Best of Simple Talk

Confessions of an IT Manager

Phil Factor



Second Edition
ISBN: 978-1-906434-18-2
www.dbooks.org

Confessions of an IT Manager

2nd Edition

by Phil Factor

Copyright Phil Factor 2009

ISBN 978-1-906434-19-9

The right of Phil Factor to be identified as the author of this work has been asserted by him in accordance with the Copyright, Designs and Patents Act 1988.

All rights reserved. No part of this publication may be reproduced, stored or introduced into a retrieval system, or transmitted, in any form, or by any means (electronic, mechanical, photocopying, recording or otherwise) without the prior written consent of the publisher. Any person who does any unauthorised act in relation to this publication may be liable to criminal prosecution and civil claims for damages.

This book should not, by way of trade or otherwise, be lent, sold, hired out, or otherwise circulated without the publisher's prior consent in any form other than that in which it is published and without a similar condition including this condition being imposed on the subsequent publisher.

Editor Tony Davis

www.dbooks.org

Contents

Contents 3

Introduction 7

Foreword 9

The DBA's Demise - a recitation 10

Section I: On Learning to Love your Manager 12

- Training Your IT Manager 13
- The Septic Tank 19
- The Yancey Men 23
- Bunnikins! 29
- Betting on Promotion 35
- How to Prevent Initiatives 39
- Irregular verbs for IT managers 45
- Phrases with which to discourage ideas 47
- Initiating a project with a Strategy one-pager 51
- Looking Busy 55
- Doing things - The Manual 59
- The Incident of 'The Two Johns' – an IT Manager confesses 61

Section II: The Strange Business of Software 64

- Going It Alone 65
- The Walrus and the Manager 69
- Click-A-Crematorium 75
- A Chilling Prophecy 79
- Talking Technical 81
- The Diary of a Microsoft Marketing Man 85
- Microsoft Boy announces his School Homework 87
- The March of Technology 91

Clinging to the Flotsam. A Survivor's Tale 97

Section III: Software Projects: the Good, the Bad and the Pitiful 100

Secrets of successful IT projects 101

Confessions of an IT Strategist 109

The Technically Minded Subclass and the Fog of Misperception 113

The Data Dialog 117

The Joy of NAD 121

Smoke and Mirrors 127

The Acronym Playpen 131

The Writing on the Wall 135

The Ghost in the Machine 139

The Sticking Page-Down-Key Incident 145

The Escape from Developer Hell 149

The Time Bomb 153

On the Trail with the Cowboy Coders 159

When the Fever is Over, and one's Work is done 163

Section IV: Hiring, Firing and other acts of Villainy 174

Bogus resumes and unblushing lies: navigating the database hiring waters 175

IT Agencies and the Devil 181

The Stepford Geeks 187

The Terror of Technical Tests 191

The Interview with the Psychometric Test 193

Brown Shoes Don't Make It 195

Technical Interviews and Tests Have Got to Stop! 199

Two stops short of Dagenham 203

The March of Time 207

Fired With Enthusiasm 209

The Whipping Boy 213

Hens that Crow 219

www.dbooks.org

The New Man 223

Section V: What If …? 228

Had IT been responsible for the Creation 229

Had offshoring been responsible for Hamlet 233

Had Tennyson been a Technical Author 239

Had God been a Technical Author 241

Had the Sistine Chapel been Created by Committee 243

Captain Codd and the Simple Proposition 249

Section VI: Hiccoughs in the Working Day 254

I could do it in my Sleep 255

The Pub Lunch and Programming 259

At half-past three, it's time for tea... 261

Cha: Tea-drinking for IT Developers 265

Survival Tips for PowerPoint Boredom 269

The Joy of IT Meetings 275

Why do we call them 'Bugs'? 279

The Ballad of the Tuple Relation 281

Common Law, and the Need for Restraint 283

Phil Factor on the Law 287

Santa's SLA 291

Tomorrow will be our Dancing Day 295

The Data Center that Exploded: A Halloween Tale 301

www.dbooks.org

Introduction

During my career as an IT professional, strange things have always seemed to happen to me. The true incidents I describe in the chapters that lie ahead are those that are intended to illustrate a point about working in IT.

There are many such incidents that I cannot talk about because, in doing so, I would reveal my true identity. Many cannot be used because they are unrelated to the themes of this set of articles, which are about the IT workplace, and the predicament of those who work in the business. Some stories have to remain untold as they are so outrageous that nobody would believe them. A few would be unsuitable for a public airing on grounds of decency.

Even as a student, I was dogged by bizarre events. I once called in, out of curiosity, to a fashionable restaurant and nightclub in Central London that was decked out in a 'Satanist' theme, all black walls and plastic skulls. My wife and I sat down to a tasty and rather expensive meal amongst the rather dodgy tasteless decorations.

After a while, a large chandelier fell on my head.

I mistakenly decided that I was the target of a joke or a television prank, so I pretended that nothing had happened and carried on eating. Blood trickled down from a cut on my forehead, but I munched on my meal with true British Backbone. Nervous waiters suddenly were fluttering around apologising and mopping up. Thinking the joke had gone on rather a long time, I ignored them.

After a while peace was restored. It suddenly seemed too quiet. I looked up to spot other diners backing out through the restaurant door, eyeing us nervously. From behind the bar, I could just see the top of the head of a barman as he stared like a frightened rabbit at me from behind a row of bottles. The kitchen door was slightly ajar, and two pale faces were dimly visible from behind.

Oh dear, I thought, they think I'm the Antichrist.

I stood up. The kitchen door slammed, and I then caught site of my bloodied face in the mirror behind the bar. Hmm. I could understand why they'd thought that.

Nobody charged us for the meal, then or on any of the subsequent occasions that we visited. There is joy to be had in a case of mistaken identity.

www.dbooks.org

Foreword

Whilst steering the Simple-Talk.com website, I was on the look out for an experienced IT professional who could add tone and dignity to the site by making weighty pronouncements about industry trends, and provide thoughtful comments on events of interest to those working with SQL and .NET.

Instead I found Phil Factor.

While his articles deviated alarmingly from the brief, shot through as they were with a wild seam of anarchic humour, it was really no surprise to me that they went down well with our readers. In all of his work the humour is a sugar coating for a pill of sage advice, wrought from years of rough-and-tumble in the IT industry.

It is an honour and a pleasure to present to you his Full Confessions of an IT Manager – a selection of some of his finest writings for the Simple-Talk site.

If you like what you read here, please do visit the website (http://www.simple-talk.com) where, alongside the pepper-in-the-mill that is Phil, you'll find all manner of technical tutorials, opinion, workbenches, and scripts for SQL Server, .NET and Exchange people.

Of course, the book is best enjoyed whilst relaxing over a pint or two of real ale. Enjoy!

Tony Davis

Editor, Red-Gate Software Cambridge

The DBA's Demise - a recitation



Verses written after suffering a deadlock error, whilst listening to 'The Streets of Loredo' on the Internet Radio.

On a doorway in Dover
I once stumbled over
a programmer wrapped in a sheet.
This poor prostrate nerd
didn't utter a word
as he lay there half-blocking the street.

www.dbooks.org

"From your face I can see
that you work in IT
what has brought you to this sorry state?"
This poor wretched geek
then started to speak
and told me his terrible fate

Cruel fortune had picked him,
as sad deadlock victim.
The conflict had left him for dead;
for wise men don't mess
with a deadlocked process
an incident DBAs dread.

"When I'm dead, on my harp
I shall play in C#"
on my terminal, happy I'll be.
I'm going offline
to a site that's divine
where all pizza and coffee is free."

"Insert into grave
select * from poor Dave
And put on my gravestone 'Raiserror!
a victim, one time
of a deadlocking crime"
Then he died with a look of sheer terror.

Section I: On Learning to Love your Manager

During Phil's time in software development he has encountered all manner of oddballs and eccentrics … and most of them have been IT managers. There are two basic routes into IT management. You can learn your trade through blood, sweat and tears and work your way up the ladder gradually, based on the credibility you've gained from hard-earned technical know-how and successful projects. Alternatively, you can don a sharp suit and tie, learn the lingo, and smooth talk your way to the top.

Both routes seem equally effective. Dealing with the latter breed can certainly cause some moments of dismay and incredulity … despair even … and some of that is documented in these stories.

However, it's easy to become sad and embittered when one encounters technical incompetence in positions of power, and to tar all managers with that same brush. Phil advises against it. Most managers work hard and contribute well to the company, and even poor managers can be trained up to the required standard, if you just follow a few simple guidelines. It's part of your team responsibility to help your manager function in a way that will benefit all.

Ultimately, if you can't train them, get them promoted, or avoid them, maybe you can learn to love them just for their unintended contribution to the rich comedy of the workplace. [TD]

www.dbooks.org

Training Your IT Manager

As Mark twain once said, "Training is everything. The peach was once a bitter almond; cauliflower is nothing but cabbage with a college education." But, in the workplace, who is training whom? Phil demonstrates that even the most clueless manager can be brought up to the required standard; it just requires method.

A while back, I got a most agreeable job working as a DBA on a gaggle of Sybase systems for a Financial Services company. Compared to my previous white-knuckle ride in a start-up IT Company, it was tranquil and stress-free. I had plenty of time to study my surroundings.

After a while, I noticed that the Comms team seemed to be enjoying their working life far more than the rest of us. There was always laughter in their group. Regular as clockwork at midday on Friday, their manager would slap his wallet and shout, "Come on guys, first pint is on me!" And off the team would go the local hostelry. The rest of us ate soggy sandwiches in front of our screens, while our managers scowled morosely at piles of memos, reports and emails. It also had to be more than coincidence that the Comms team always seemed to have special 'offsite meetings' at a local hotel with a widescreen TV whenever an important football match was on. Despite all this jollity and pub time, the team were, to everybody's intense annoyance, highly productive and always met their commitments. Overtime and bonuses rained on them like manna from on high.

Eventually, their manager got promoted and his replacement came with a reputation for being tough, humourless and difficult. As we had become institutionalised into being miserable and unpleasant people, we gleefully awaited his arrival. Fireworks were expected and the whole open plan office was awash with a sort of pre-emptive Schadenfreude. People moved their PCs around to get a better view of what would happen next. One or two members of staff hung mirrors on the wall in strategic places.

At first, it seemed that the situation was developing much as we had predicted. The new manager would storm about the office, and into his pigpen, barely acknowledging the existence of his team. Instructions would be relayed

via curt emails. The visits to the pub, and gaiety in general, were curtailed and the Comms group went very quiet; studious even. Puzzlingly, however, the team seemed unperturbed and the expected uprising failed to materialize. They were often seen to confer quietly amongst themselves, as if planning something.

After a few weeks, the more observant among us detected that something was different. We noticed an increase in the frequency of conferences between manager and team. Soon, in his journeys about the office, the manager started to adopt a strange, clearly unpractised, facial gesture that was like the grimace of a sallow Transylvanian aristocrat. After a few more attempts, however, his facial muscles settled into these previously unfamiliar duties, and the grimace transformed into a sincere and rather fetching smile. We were stunned. One or two members of his previous team visited on a pretext, just to gawp.

After about a month we were startled by the once-familiar cry of "Right lads, it's Friday lunchtime, what the hell are we doing here?" And off they went to the pub, laughing and joking, and watched by a sea of astonished heads furtively peering over the tops of cubicles.

In the weeks that followed, peals of laughter emanated once more from the Comms area, and good-natured jokes flew around the group. It was extraordinary. It was like being in Santa's grotto. The group worked hard, like gnomes, often into unpaid overtime if a job required it, but seemed to know how to make the most of life, even whistling as they worked. In and around them was their new beaming, genial manager.

A short while later, the manager took the whole team out for an evening meal, as a special thank-you for finishing ahead of schedule a tricky network-replacement project. At this point, curiosity got the better of me and I approached one of the Comms team members. He was a veteran who had chosen a 'technical' career path, which meant that he always reported to managers with less knowledge and experience than himself.

"Come on, this didn't happen by accident" I said, "How did you do it? Did you dope him or something?"

"I'd love to tell you all, but my throat goes dry when I have to talk."

I took the hint and off we went to try out the guest bitter in the local pub. I lined up a few glasses along the table and, as we sipped on the amber nectar, he explained as follows.

"Managers have to be trained to your ways. They always arrive in a state that makes you whistle to yourself and wonder who'd had charge of them before you. It is very similar to the training of a puppy. At the start, there is a

www.dbooks.org

lot of 'leaping up', and 'straining at the leash', and 'playful nips'. By a system of training and reward, you modify that behaviour.



Managers need to be trained to your ways

All we've done is take that basic behaviour modification process and adapt it for IT manager training. I didn't invent it, oh no" he added, "I learned it ages ago from a grizzled old analyst in ICL."

At this point, we had a fascinating conversation that lasted a couple of hours but which I have condensed into the following IT manager training manual:

Stage 1: Observation

When a new manager arrives, observe his or her behavior closely. Start making a list of all of his, or her, good qualities. Many systems analysts find this hard, and protest that their manager hasn't got any. One often has to struggle, but after a while the process of listing them becomes intrinsically fascinating and one often ends up startled by the length of the list. In order to 'take away the sting' some teams counterbalance this with a list of the manager's faults. However, I advise against this as it can be counterproductive to the task at hand.

Stage 2: Establishing a baseline

Once the list is complete, set up a database that all can access. Log all occurrences of the manager's 'good behavior' in the list. If your manager acknowledges your existence as human being one morning, then record the date and time that this occurred. Likewise, if he thanks you for doing something, log

it. Perform a frequency analysis to find out how often the good behaviours occur. This is his 'baseline' behaviour.

The goal then is to develop a strategy that will adapt the nature of this 'good behaviour' and increase its frequency above your established baseline, and up to some predefined target level.

Stage 3: Developing a training strategy

With behaviour modification, punishment is out. It simply doesn't work. This is a great disappointment to many an IT person being introduced to the technique. They anticipate, with some relish, wielding a large cattle-prod. Anger must be defused early on.

Instead, this technique works on a reward system. After a certain number of occurrences of 'good behaviour' a reward must be administered. It is important to identify the reward. Flattery works well with managers: they can't resist it. Everyone believes they are immune to flattery, but my goodness, most of us can take it by the skip-load. There seems to be no upper limit, and this is where Stage 1 pays off.

Alternative and supplementary techniques could include showing an interest in listening to how cute and clever a manager's children are. It is hard work but most managers like nothing more than to tell you of all the clever things their children did. Another good reward is to work flat out on a job and get it done ahead of schedule. Alcoholic drinks and cigars should only be used as a standby, as a successful training session could have side effects to health.

As well as devising a reward system, you must also identify and use a 'token' system. The reason for this is that it is not always possible to administer an award directly – for example the manager may merit a reward during an external meeting. In such circumstances, one must administer a 'token' in lieu of a reward. With dogs, a simple pat on the head, a chocolate drop or a spoken 'good boy' is often enough; but this can be conspicuous with a manager. Instead, try to use a key phrase that the manager will come, by 'Pavlovian' training, to subconsciously associate with a forthcoming reward. Often, a meaningless word such as 'synergistic' works well.

Stage 4: Deployment

This is where good planning is put into effect. The team takes it in turn to take the lead, becoming what is known as the 'dog handler' to the manager. All other team members key in the occurrences of good behavior and they are logged immediately in the database. Did he say good morning? Tick in the box.

www.dbooks.org

Ten ticks, and a reward has to be administered by the dog handler, or a token if the reward cannot be immediately given. The dog handler is usually notified of this by an email or SMS message

If all is done properly, the frequency of good behavior should increase greatly, and the manager should become dimly aware that his good behavior is being noticed. He will begin to realise that, if he smiles and is polite, the work gets done quicker, or team-members start saying how cute his children look in his framed family photo.

Stage 5: Appraisal

This part of the project cycle involves reassessing the objectives or, in other words, working out if the desirable behavior has increased to the target levels, and re-defining the triggers that warrant reward.

For example: you might have recorded that, at one point the manager made a vague promise that at some undefined moment in the future the group might be allowed out to a nice country pub one Friday lunchtime. Your reward system for such desirable behavior may have increased the frequency of vague promises, along with the occasional insertion of a clear commitment to go to a pub lunch. You then shift the target behavior to 'actually going for a pub lunch' and determine the reward system needed to reach that target. Then it is back to Stage 4, unless you determine that you've reached all your objectives.

At this point the grizzled veteran sat back and took a satisfied pull on his pint.

"So you work on this system continuously?" I asked.

"Funnily enough, no" he replied "After a while, we genuinely start to like the manager as his behaviour and attitude becomes more agreeable. And because of all our hard work, flattery, and so on, he begins to like us. If Stage 5 is deemed successful we generally go back to background observations. We continue the reward system but at a more relaxed and instinctive level."

"And then …?"

"And then everybody notices how successfully the group is working. The manager gets the credit and he gets promoted."

"Is that fair?"

"Well, it pays for all the drinks he's bought us. And we start again with the next manager. It may seem hard work but it gives us an interest, and we get good management for a few months before the inevitable promotion."

Of course, I have simplified the techniques as described to me, as I have no wish to bore you. Wherever I have worked ever since, I have passed on knowledge of these techniques to anyone who will listen. If it is done in the right way, it works like a charm. One programmer I instructed even left the profession to become a dog trainer!

Now that I am, myself, a grizzled systems analyst I pass on the knowledge with the plea: Please don't tell the management. However, fear not. You can be assured that they have secret techniques of their own!

www.dbooks.org

The Septic Tank

It's the few IT managers who are promoted far beyond their talents that spoil the perception of the whole breed. But how did they manage to progress so far? The 'septic tank' analogy has stuck in Phil's mind.

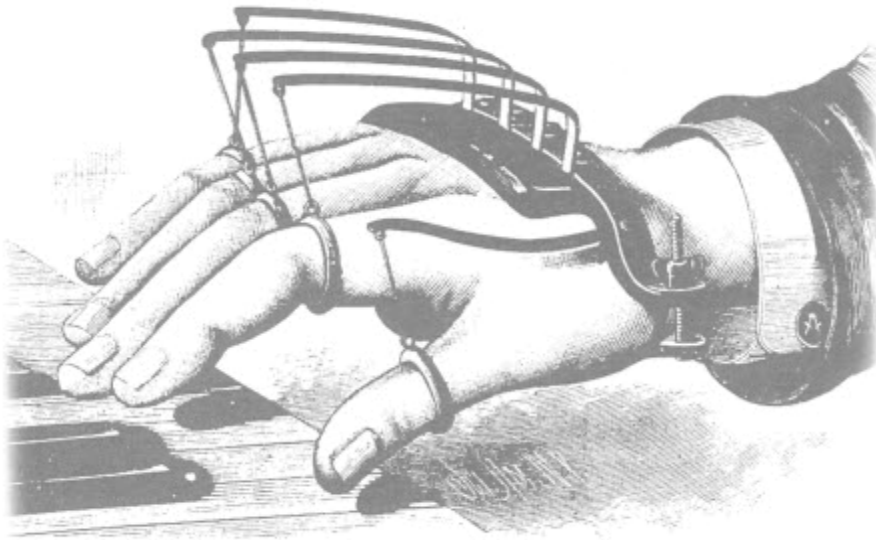
Once, whilst I was working for a large international enterprise, I received a rather disturbing 'edict' from the IT director. The edict stated that **two megabytes of memory were sufficient to run all Windows-based PCs**. So firm was this edict that it became part of corporate IT strategy. PCs were not to be purchased with more than two megabytes of memory. It was some years ago, but even then it was absurd.

I phoned the departmental manager responsible for setting the standards for the configuration of PCs, and suggested to him that there may have been a clerical error in typing the edict. He was somewhat apologetic, but said it had come directly from the IT director who, against all advice, was extraordinarily keen on the idea and was already proudly calculating the money he would save the company.

"We have no choice but to 'go through the motions' of implementing it" he noted, glumly, "Also, it's the first idea he has come up with since he had taken on the job, and it seems wise to encourage and support him."

The edict caused me little more than inconvenience. All I had to do was to install some PCs in the purchasing department to the new regulated memory size. There was much laughter when I showed them the edict … they knew what was required. When more memory was subsequently purchased, it was simply listed under 'computer sundries', or 'miscellaneous electronic equipment' by these good people, and supplies then flowed in as usual.

However, the incident did serve to suddenly awaken my interest in the IT Director, Tony, who had issued the edict. By all accounts, he lacked any of the qualities necessary for such a demanding role. His social skills were so poor that only the DEC salesman would willingly talk to him. He had to call his secretary over to access and read his email (it was usually printed out for him). He was helpless in any challenging practical or intellectual task. He apparently spent his days buying and selling his own shares on the stock exchange, with little success.



Miscellaneous electronic equipment

How, I wondered, had he progressed so far through the company?

I pulled aside another manager, who had known Tony for many years, and asked him how on earth he had reached the exalted ranks of IT director. He shook his head sadly, and spoke coarse words that have stayed with me ever since:

"Ah well, they say that the biggest turds always rise to the top of the septic tank."

At the time I was taken aback by this coarse phrase (apparently of army origin), which I apologise for repeating verbatim. However, it … err … stuck in the mind and does, I think, contain an element of truth.

Subsequently, I spoke to as many people as I could and a consistent story emerged.

"Oh yes, poor Tony. He was a bit of an oddball he was. I was persuaded to take him into my team by a colleague. I was told he was a good worker if he got the right opportunities. Was he hell! He sat around dreamily, and he had such repulsive personal habits that it used to upset the team. Nobody would work with him. It was getting on my nerves too. There seemed no good way to get rid of him, so I got him promoted."

"You did what?"

www.dbooks.org

"Yup. You don't imagine it is possible to sack a member of staff for incompetence do you? We'd have almost nobody left. What do you imagine the unions would say? The only viable way to get rid of an unwanted team member is to promote him. Onwards and upwards, as they say … I phoned a colleague, and told him he was a good worker if he got the right opportunities."

"Do you ever recommend people who are good at their job for promotion?"

"And lose them immediately? Are you kidding?! I need to retain good hard-working people in my team".

And so it went on. Everybody I spoke to gave me the same story. Tony made remorseless progress through the tiers of IT management by the simple technique of picking his nose whilst staring at the screen, and grunting sporadically like a sow in heat.

Fortunately for the IT industry, not every manager within a large company, or government department, has achieved his promotion this way. I've met many talented managers who have made enormous personal contributions to the well being of the enterprise for which they work. It is, however the minority who are promoted far beyond their talents that interests me the most, for their unintended contribution to the rich comedy of the workplace. How did they get there? Is there a whole range of group processes that cause unlikely people to be propelled way beyond their level of incompetence? Is there a particular talent that goes unrecognised by you or me, but which turns an ordinary person into a management X-man, with a sort of special management power only visible to another initiate?

www.dbooks.org

The Yancey Men

An IT director with strong views and opinions is a walking time bomb. It is much safer to install a 'Yancey Man', whose actions and words can be minutely controlled by the team. However, as Phil learns to his cost, a puppet might bow to your bidding; but only when it is you who is pulling the strings.

I was once part of a team of managers in a large corporate IT department who hired a 'retired' actor to be the IT Director.

This statement requires some elaboration.

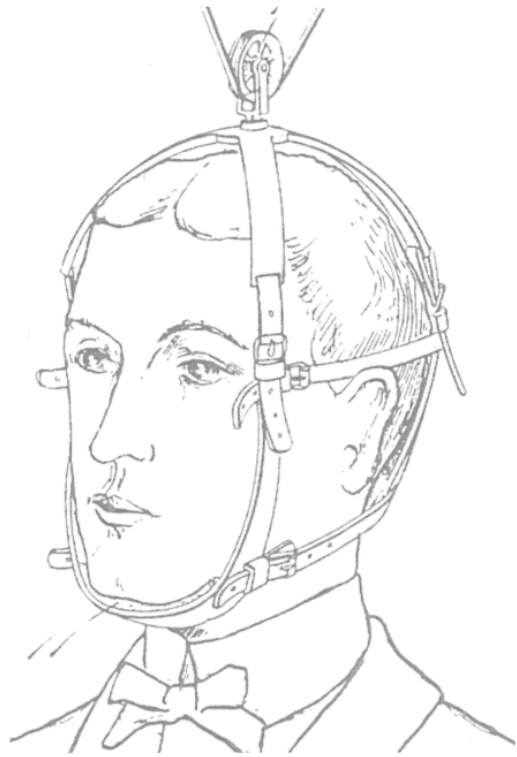
The IT director of a large company actually has little work, so long as the members of his management team know what they are doing. He does, however, need to look smart, sober, and dignified and to behave at all times with a natural 'gravitas'. His only duties are to arbitrate, convene meetings properly, and have the knack of being able to summarise a mass of detail into an overall picture. He must also work with complete integrity to reassure nervous stockholders and board members. His final task is to translate complex technological issues in such a way that everyone thinks they understand them. He can then spend the rest of the week on the golf course, or staring out the window, meditating. The last thing he should think of doing is 'getting involved'. 'Smartness' is not to be encouraged in an IT director. He should never have ideas of his own, but merely encourage others to have them; he can later turn them into a departmental initiative if the consensus favours them.

This may come as a surprise. To outsiders in a large organisation, the IT director seems a man of inestimable importance. 'I wonder if Tony can come up with the goods for the new project', they will say, or, 'Let's ask Tony what the long term strategy should be with the reorganisation of the logistics'. It is as if Tony is rushing around personally doing everything in a muck sweat. Actually, if the IT department is working as it should, he has almost nothing to do besides looking serious, and promising to 'get back with an answer'.

Once an IT department is running in this harmonious way, it causes disruption if the IT director leaves. There is always the temptation in other parts of the enterprise to jolt the IT department out of its complacency by hiring a

new person with strong views and opinions; a chief who will lead from the front. I have never seen an IT department benefit from a new broom. On the contrary, the consequences are generally messy and unfortunate.

The inspiration for our idea of hiring an actor to undertake the role of IT director comes from a wonderful novel by Philip K Dick, 'The Penultimate Truth' in which a group of manipulative but powerful people in government decide to construct an animatronic puppet, called **Yancey**. The Yancey puppet is subsequently elected by them as president. The inner coterie, called Yancey Men, write all the scripts uttered by the puppet, in order to delude the population. The novel was a clever satire of the Eisenhower government and the 'invention' of the Cold War as a political device, but also eerily reminiscent of the Blair regime in the UK.

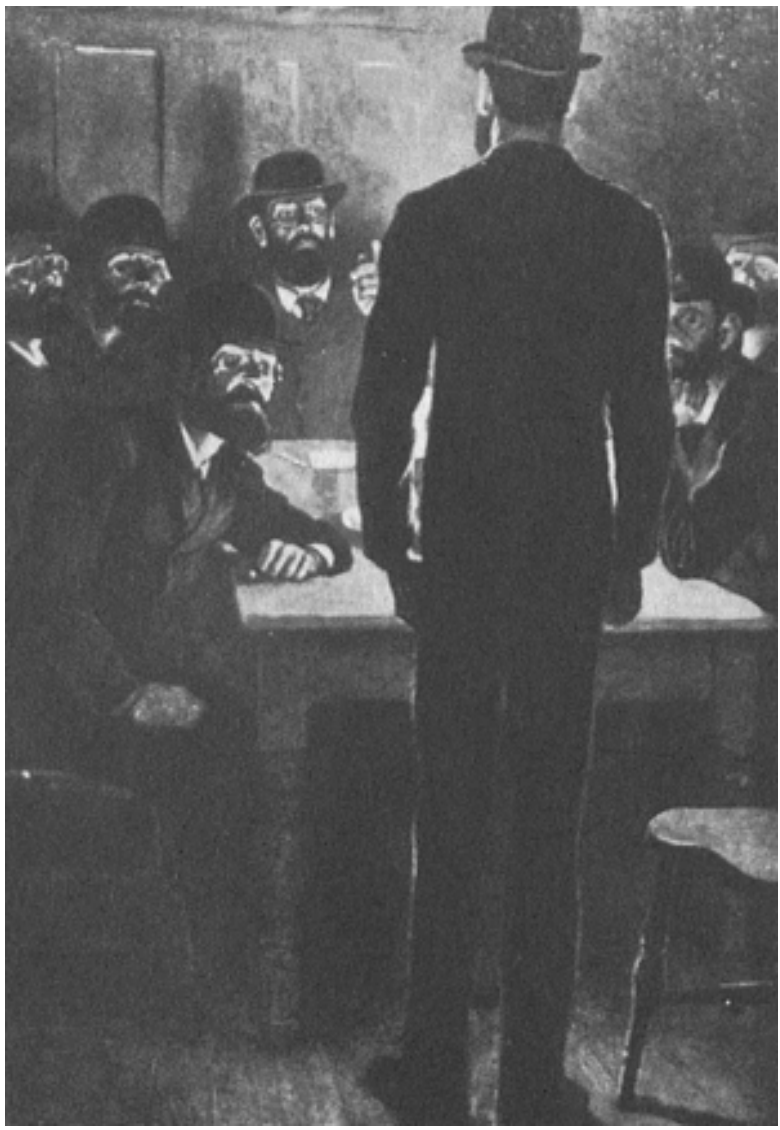


They constructed the animatronic puppet

Installing an actual puppet as the figurehead of the IT department seemed beyond us, but the idea in principle was a good one. As luck would have it, a firm of consultants we'd employed to advise us on workflow systems included a man who embodied the archetypal bishop, with a grave and thoughtful face,

www.dbooks.org

tall, slim, well-dressed, with hair turning slightly silvery on the temples. One evening, this very consultant, whilst being plied with drinks in the local pub, became garrulous and confessed to me that he was actually an actor, whose days as a matinee idol were long past, but whom the consultants had taken on to lend tone and dignity to their team. He was entirely innocent of any in-depth knowledge of IT.



They named him Yancey

Once I'd got over my initial indignation at the effrontery of the trick, I began to watch him closely and my admiration grew. Every word he spoke, every gesture, radiated wisdom and measured courtesy. He could have recited 'Baa Baa Black Sheep' and made it sound like a corporate mission statement.

On speaking to the other managers, I discovered that we'd all been struck with the same thought. This man was a gift from a higher entity. It would be a sin not to headhunt him as our new IT director. We'd tell him what to say, and what to do, and soon we would have our very own, living, breathing Yancey. We'd be left in peace to run our departments, whilst our Mr Yancey did all the boring things that are the lot of an IT director. We knew he'd bond well with the board and the shareholders. It was an easy decision to make, and we made it.

Well, it all went like clockwork. We appointed him, and he did what he was told. Of course, we had to give him some training in corporate IT-speak. One must address concerns, not solve problems (this came about after a lawsuit in the motor industry), manage expectations rather than lie. Admit to confusion rather than accuse someone else of lying. One should refer to things as sub-optimal rather than cr*p, non-final rather than unfinished. He had, however, already been well-drilled in the phraseology by his consultancy, as it was part of their initiation training.

We had weekly meetings at which he'd brief us, and we'd decide on how he should react. He'd come up with all the questions he'd been asked and we'd tell him how to respond. We were in awe of his talent at presentation; we felt no jealousy at his exalted status, only respect for his thespian profession. A most harmonious interval ensued. We 'pulled the strings' and our puppet wowed the audience. We got on with the real work of our departments, secure in the knowledge that we were being ably represented in meetings and in the interface with other parts of the corporation. The IT department's reputation improved mightily and budgets increased, as the board gazed spellbound at our profound and empty-headed leader.

The inevitable storm appeared like a 'cloud no bigger than a man's hand'. At one of our meetings he suddenly asked why we hadn't considered 'Snibbo' in our IT solutions. We laughed genially and explained that the reason was that 'Snibbo' may as well not exist; its influence would be so small. He shrugged and changed the subject. (I use the word 'Snibbo' instead of the real word, in order to avoid legal issues: practically every possible name in IT has been used up.)

As the next meeting drew to a close, he said, 'I'm surprised you haven't considered a forward-looking solution to our workflow issues, such as 'Snibbo'. I'm told that you are inclined to be very conservative in your thinking'. We

www.dbooks.org

blushed prettily at the unintended compliment but puzzled about where 'Snibbo' had come from. Our Yancey was not normally given to independent thought.

Over the next few days, I quizzed other members of the management team but they all looked innocent. Nobody could remember feeding these lines to our puppet. We met together at the pub to try to puzzle out what was going on, but drew a blank. Then the thought struck us with horror: was someone else putting thoughts into his head?

It quickly got horribly worse. He went to the MD of the company and fired him up on the necessity of getting IT solutions based on 'Snibbo'. He explained to him that the only thing holding him back was his backward-looking management team in IT. Such were his presentation skills that the MD became putty in his hands.

At our next meeting, we decided to turn on him in a concerted way and explained to him the dangers of overselling a product-type as a panacea for the IT requirements of the business. He shrugged and merely said that he had received expert advice on the subject. He surprised us all by telling us that we would do well to become properly acquainted with 'Snibbo' and its repercussions for the rollout of IT solutions in a fast-moving business; this tirade promptly ended with some techno-babble too arcane to be repeatable. He'd learned it off by heart, just as he had when wowing the matinee audiences during his theatre days.

I was now desperate to find out who had gained control of our 'Yancey' and was using the device against us, the true owners. I was most indignant, though I couldn't quite work out what crime had been committed against us. After much ferreting about, and closely questioning his secretary, I uncovered the new puppeteer: a business-analyst who was supposed to be preparing the ground for an IT initiative in the Purchasing Department. Somehow, in the course of reporting progress on his project, he had inserted the idea of 'Snibbo' as 'the glue between all our disparate systems' into Yancey's fertile and impressionable mind.

It proved difficult to ask for our puppet back, especially when the new puppeteer had insinuated himself into an impregnable position, being seen as a fount of wisdom by both the MD and the IT director. We couldn't explain to the board what we'd done. It might be taken the wrong way.

We did all we could to counter the tide of misinformation. We hired a well-known expert in workflow systems who was able to say that Snibbo was not even on the candidate list for a workflow solution and had no track record of success. We later discovered highly suspicious links between the business

analyst and the publishers of the Snibbo Software system. We also found out that the system was not yet even working on any reference site.

Skilfully, our Yancey puppet raged through our defences, like Godzilla lumbering towards Tokyo through the electric fence. The siren call to the business was irresistible: 'There is really no need to reorganize the business to make it more efficient, when all you need is 'Snibbo' to make current processes quicker and less error-prone'.

I can leave the wider consequences safely to the reader's imagination. All I need to relate is that a vast new initiative introducing 'Snibbo' was launched. When the reality of the product became apparent, teams of programmers from the Indian subcontinent arrived, all claiming to be experts in the subject, and eager to add 'the missing five percent' to the product and integrate it into the company's existing systems.

I didn't hang about to see the final act of the tragedy, like in Hamlet, with bodies littered around the stage. Our actor-turned-consultant was one of the corpses. When the curtain was pulled down, a new, slimmer IT department emerged, with no appetite for radical IT solutions. Even now, people wince when the word 'Snibbo' is mentioned.

As for me, I should, I suppose feel guilty for my part in creating the monster that reared up on its hind legs and destroyed all around it. It has certainly modified my conception of the perfect IT director. An empty head is never a good idea, even if it is well-groomed and dignified. Wisdom and experience cannot, it seems, be applied by remote control.

www.dbooks.org

Bunnikins!

When an IT manager is selected as a victim of office politics at a large corporate, it is time for him to engage in intelligence-gathering. Phil Factor exposes a ruthless streak when called to account over an Engineering Workflow system.

"Hmm. Should one really put the Bishop of Llandaff in the same bed as Lady Kerkrade I wonder?"

"I wouldn't. The Bishop tends to attract flies, which will infect Lady Kerkrade. The Bishop is a bit loose in habit as well, and will flop all over Lady Kerkrade. Also Lady Kerkrade is a gross feeder and needs plenty of top dressing. I tend to fork manure in around her."

"But the colour combination works well surely? I've always wanted a hybrid and the Bishop of Llandaff is a great cross-pollinator."

I looked thoughtfully at the flower catalogue.

"Nah, the colours clash. Lady Kerkrade is a rather strident 'bathroom pink'."

Keith, the Director of Engineering, sighed and stared dreamily out of the window. He and I shared a common interest in dahlias, and I often slipped up to his office in the lunch hour to discuss the finer points of gardening. Even an IT manager must relax occasionally.

As we thumbed through the flower catalogue, like two schoolboys over a stamp album, Sharon stared at me lustfully through the glass partition. A remarkable woman, Sharon. Her mind seldom seemed to stray far from thoughts of sex, and her undemanding job, as Keith's secretary, didn't provide her much distraction. I blamed the air from the Essex salt marshes, or perhaps her diet of E-numbers.

Suddenly, her demeanour changed and she tapped the partition. We hastily put the flower catalogue under a nice thick IT strategy report. A moment later Dan Stepford marched into the room, looking keen, dynamic and intelligent. Difficult for a man that looked like a poodle. 'Desperate Dan', he was called amongst the managers, due to his frantic efforts toward promotion.

Desperate Dan was a bit of a problem to both of us. His current tactic in raising his profile and visibility was to lambast IT for its failings. This was seen by the rest of the organisation as amusing but unsporting; a bit like shooting a sitting duck.

"Keith, may I have the file for the BCF27 Variant project please."

"Be my guest, of course."

Dan glared at me and marched briskly out. Keith looked worried.

"Surely, he's not going to start putting the dagger into the engineering workflow project now? I can't tell you how much that has helped us hit our 'dates'. OK we cut corners, and the IT Kremlin are hissing through their teeth, but it does the job."

"He can't" I answered flatly. "It's watertight. A few bugbears here and there of course but they'll be ironed out, and the users love it."

Keith sighed. "May the lord strike him. I don't like the way he looks at me as though measuring me up for my coffin."

"Did you know that before Admiral Byng's execution, they used the well-tried method of sending in four men into his cell of roughly his height, and starting a pretend argument about which of them all was the tallest."

"Shut up Phil."

"Yes. Sorry." History is a bit of a hobby and I occasionally get carried away.

"Will no man rid me of this turbulent executive", muttered Keith, picking up the historical theme. We fell into a brooding silence. I removed the flower catalogue from underneath the strategy paper and began idly flicking through it. It was no good, though; the light mood had been replaced with a weary gloom.

Through the partition I caught sight of Dan engaged in intimate conversation with Sharon. As Dan began to move away, Sharon winked coquettishly and mouthed a word at Dan. His cheeks turned a shade of pink strikingly like that of the 'Lady Kerkrade'. He shuffled on the spot for a moment before grinning awkwardly and striding off.

In the everyday life of an IT manager, information is everything and something told me that I needed to know what Sharon had said that had made Desperate Dan blush so.

Any customer-facing IT manager who wants to survive more than 6 months needs to harness the vast office intelligence system, fuelled by email, text, instant messaging and gossip.

www.dbooks.org

It is important only to have a contact who keeps abreast of it. Under the pretence of having some figures checked for an IT project I was costing out, I had a quiet conference with Gemma in Accounts. Although hardly someone who you'd want on your side in a pub quiz, she was a walking encyclopaedia of company gossip. Nothing escaped her attention.

I returned to my cubicle after my chat with Gemma, smiling smugly. However, I immediately saw that Tony was quivering with alarm. Tony was an able but highly-strung programmer. He was a genius at producing office-automation systems in VBScript and SQL Server. Whilst most development teams would have still been busy planning ambitious architectures, Tony had the entire engineering workflow application hacked out and working to the goggling amazement of the users. I had gained so much credibility in the business from harnessing his skills that I kept him close by in my cubicle, just as a gamekeeper keeps his best ferret in his sitting-room.

"Dan Stepford has just sent us an email with copies to half the human race criticising the engineering workflow system for being inadequate for their needs. Evidently, it is not 'strategic'."

"Let him" I spat. I read the Email. "Cor. A bit harsh. Ouch. Dammit."

Having finished reading the email I felt what I imagine Wellington felt after having just managed to withstand the attack by Marshal Ney. All right, he asked for it.

"Come Tony." I said. "Off we go to see Desperate Dan."

"Bbbbbbbbbut …"

"None of that! All you need to do is observe."

We found Dan at his desk in the open office area. The vast room was a sea of faces bent over desks, or staring into terminals.

"I haven't read such a stupid email as yours for a year. That is seven years in your time-scale I reckon." I declaimed loudly. My years in amateur theatricals and operatics came in handy.

Dan looked up. He knew what this meant. Only one man would be left standing after this conversation.

The sea of faces turned too. Nobody actually got up and moved, but you sensed that everyone had somehow slithered into ear-shot.

I allowed Dan his first riposte. He did well, I grant him. With vicious forensic precision, he tore our wonderful workflow project to pieces. We tried to look dignified as he spoke eloquently about its shortcomings, the lack of

scalability, production robustness, and maintainability and so on. I could hear Tony gently whimpering beside me but I moved not an inch. Like a vast game of grandmother's footsteps, the occupants of the room inched closer. Everyone likes a fist-fight in the workplace.

I waited until he'd finished, my thoughts focussed on my next move. Dan was looking at me smugly. The reputation of the IT department depended on my ability to repudiate his tirade.

"Dan, you are entirely wrong in your analysis of the workflow system. It has been signed off as satisfactory for use by your director, after full user acceptance testing." In the blink of an eye, I switched to a soothing, placatory voice. "Look, I'm sure we can sit down and work out these differences, Dan ..." I paused momentarily, filling my lungs much as I did when I was the star of the local operatic Society, when my voice filled the auditorium ... "… or should I call you Bunnikins ...?"

There was a slight pause and, like the spear-carriers of the operatic chorus, the assembled occupants of the office looked at each other quizzically. The fat man had stopped singing.



"Or should I call you Bunnikins?" The fat man had stopped singing.

www.dbooks.org

Rather poetically, Dan again turned that same interesting shade of bathroom pink that Keith and I had admired in the Dahlia. He stood up as if to speak, his mouth flapped open uselessly and he slumped back down into his executive swivel chair.

He managed a spluttering parting salvo about the incompetence of the IT department, but he was a beaten man, and everyone could see it.

The room gaped in astonishment as I led Tony from the room to the lift, and back to the safety of the IT office. After a soothing cup of tea, Tony had recovered well enough to ask the obvious question.

"Ah well, Tony," I started, "it is all down to information gathering and, intelligence. And lack of intelligence. There are two rules in corporate life that are inviolate. Desperate Dan forgot the first rule, never to be forgotten by an ambitious manager; that a torrid bout of after-work lovemaking on top of one's boss's desk, with one's boss's libidinous secretary, may end up providing the wrong sort of visibility."

"And the second rule?"

"Sharon forgot the second rule: if you really must indulge in a torrid bout of after-work lovemaking with a …er ... rising manager, never describe the nuts-and-bolts of the experience in an email to your closest girl friends at work. I'm sure she swore them to secrecy but this sort of email is dynamite – especially as in it she revealed her pet name for Dan, a playful reference to his sexual desktop stamina. From that point, it was inevitable that the mail would be forwarded on and on through the Email system and eventually turn up in the inbox of one of my contacts in the office intelligence network. Such is the power of Email in our lives."

And so it was that, the next time I popped up to see Keith, he shook me warmly by the hand, and gazed at me in fond gratitude through eyes moist with emotion, before we settled down to our customary chat about gardening. We didn't even bother to have an IT Strategy paper to hand, as Desperate Dan had asked for a transfer, and left for another part of the company in Holland.

It was difficult to be a thrusting, dignified, ambitious executive with everyone calling him 'Bunnikins'.

www.dbooks.org

Betting on Promotion

The forging of a successful career in IT management often has as much to do with smoothness of manner and sharpness of suit as any real technical expertise. But on the plus side, this does make the management appointment process satisfyingly unpredictable and therefore fertile territory for the gambling man ...

The process of appointing managers in large IT departments is shrouded in mystery. The bewilderingly random and illogical nature of the process makes it a gambler's delight, and it's often difficult for workers to resist the odd wager on the outcome. This can bring much excitement to the dull working life of the system analyst.



Twenty Pounds on the one with the good hair!

At one stage in my career, I worked in the Engineering department of a large multinational company, tending a range of Oracle systems. Whenever a management position became vacant, a thrill of excitement would sweep the department. The IT managers tended to mistake this enthusiasm for genuine interest in management comings and goings and in the career progression of their bosses. In fact, of course, the interest was purely in those parameters that affected the odds on the outcome.

At some point, the post of supervisor came up in the adjacent IT area. A book was opened and the candidates for the job were appraised like racehorses. Tips from the stables were considered, and reference to fetlocks, nobbling, front-runners, coming up on the canter and coughing in the stables were passed between the members of staff.

One of my colleagues, Dim David (not his real name), was an amiable chap who tinkered with a whole range of engineering applications written in Fortran. He worked alone, largely, but always bought his round in the pub and was pleasant but dull. When this post came up, he wasn't one of the likely candidates, as he had none of the required academic qualifications. However, he had one great asset: he looked the part. His brainpower was unremarkable but he was tall, had good hair and whatever he did, he did with a natural 'gravitas'.

Whilst chiselling away at my Oracle databases, it suddenly occurred to me that Dim David would be a perfect outside bet for this supervisor post. I immediately called in to see Pedro the bookie, the local trade union rep, who held an undemanding role in the department that gave him plenty of time for other activities. He was amazed when I ignored the hot favourites and put a good sum of money on Dim David. He had a loud laugh, and on that occasion it jiggled the ceiling tiles. I was unperturbed. The decision making process for the management appointment was so extended and diffuse that I believed I could influence the outcome by viral means. After all, I hadn't been the first to tamper with a promotion race in an unsporting way. In order to avoid a loss on a wager, Pedro himself had once nobbled the favourite for a junior management post. Whilst purporting to give the candidate good interview advice, he had maintained that the managers who were conducting the interview had a grand sense of humour and would appreciate a few jokes about the candidate's previous bosses. They did not.

On doing my rounds for the next fortnight, I joined in every conversation I could about the forthcoming appointment. After a while I'd interject a phrase such as "Ian tells me that Dim David's name has come up for that supervisor job" Or, "I was surprised to hear that Dim David is being considered for that job but, thinking about it, he is due for a break." On other occasions, I'd argue

www.dbooks.org

against his appointment with equal vigor, "What are they thinking of, putting Dim David's name forward. He's struggling in the role he has!" Slowly, but surely, his profile increased to the point that his manager asked him to apply for the post, which he happily did. Pedro the Bookie observed the progress of this rank outsider with initial fascination. This turned to mounting alarm as David cantered along the rails, overtaking more fancied center-field candidates. He felt sure I was up to something but was not sure what.

I groomed David carefully for the interview. We went through all the right answers, and made sure he was neatly turned out with the correct colour of suit and so on. It's worth bearing in mind that if you are a candidate for promotion and an analyst starts giving you tips on interview techniques, it is probably only because he has a few dollars on you each way .

You will have guessed the outcome by now. Dim David passed the finish line head and shoulders in front of the next candidate. To all except Pedro, the victory was a cause for a great deal of pride and celebration in the department.

What happened next, however, is painful to recall. Poor Dim David had to supervise friends with whom he'd worked for years. They resented it slightly, and Dim David reacted with bad tempered authoritarianism. He'd learned his management style from a bad role model, and his 'school monitor' approach was ridiculous in the workplace. After a while, his senior managers noticed that he was floundering and shook their heads, wondering who on earth had promoted him in the first place. They decided that his difficulties were largely due to having to supervise his old workmates, so they decided to move him to a different team. Suddenly, Dim David was my boss.

I'd had a most harmonious relationship with my previous boss. He understood the technology. He understood his role. He only ever intervened to assist. For Friday lunch he always insisted on buying the drinks In short, he was ideal in every way. Unfortunately, Dim David clung to his autocratic techniques in his new team and felt that I was too rough a diamond to be left alone for very long. There was little left of the amiable Dim David I'd known in the old days. Promotion had turned him into a parody of a manager, and I couldn't help feeling that it was largely my fault. Soon afterwards, I left the company.

Even now, I find it hard to believe that I affected a decision process as serious as career progression just by viral campaigning. IT workers are largely just flotsam on the tides and storms that are typical of the industry, and I try to console myself with the thought that it was just coincidence. This particular storm, characterised by the usual confusion and unpredictability, had simply swept David into a post to which he was ill-suited. Maybe my meddling actually had little impact in the grander scheme of things. Nevertheless, I wish I

could at least finish by saying that I never bet on promotions ever again. Not a bit of it. I am always ready for a flutter. So when, as an IT manager, you gaze out of your glass box at the sea of heads, bent intently over screen and keyboard, don't be deluded into thinking that all is diligence and harmony. A good proportion of those heads will be gazing keenly at the latest odds in the sweepstake on who will get your job when you lose it.

www.dbooks.org

How to Prevent Initiatives



No fate should be avoided more assiduously than being party to someone else's initiative. Here, Phil offers managers an essential guide to curtailing initiatives, and generally dampening high spirits, in their IT department.

The IT Manager must do all he can to prevent initiatives happening within his department, whilst, at the same time, maintaining a general atmosphere of innovation. Your staff may not understand the subtleties of this, and will badger you with good ideas. The problem with ideas and initiatives is that they cause alarm and disturbance amongst other managers and force you to make a decision one way or the other. If you decide to block an initiative you are then at risk of being exposed in a negative way as a 'Blocker of Initiatives'. This must be avoided at all costs. Initiative avoidance techniques are therefore necessary before they reach the decision stage.

The Trick is to kill the initiative whilst giving the impression that one is actively encouraging and fostering initiatives from ones staff. Most of the following techniques owe their popularity to the way that they make it seem

that it is the apathy of one's staff that caused the initiative to fail, rather than one's own inadequacy as a manager.

One of the following methods will effectively restore the Status Quo.

The 'Thicko' method

Adopt a bluff, avuncular attitude. When the initiator comes up with a technical proposal, pretend not to understand it, and get him to explain it repeatedly, and unsuccessfully. This is an even more amusing technique if you can form a double-act with a confederate. The initiator will eventually lose patience and storm off. Much job satisfaction can be gained from the parting sally "Perhaps you will find it easier to explain it once you've really understood the technicalities yourself."

The 'Tantrum' method

Pretend to be highly stressed with far more pressing issues. When the initiator offers a proposal, feign incredulity and then throw a tantrum. This is crude, perhaps, but effective, especially if you have developed a high-testosterone management style. Follow up the tantrum by feigning indignation that they should have their head in the clouds when they should be fighting fires.

The 'Death By Alternatives' approach

In this method, you ask the initiator for the list of all the alternatives that he has considered and rejected. If he has already prepared several, ask for a more detailed analysis and demand a discussion paper on the alternative technical platforms, with lots of diagrams. You can be sure that fatigue and apathy will set in.

The 'Daddy Knows Better' approach

Receive the initiative as if greeting an old friend and say something like "Yes, we've thought of that at various times, but it simply wasn't practical/ economic/ policy/ effective." If you have the paperwork, then produce it and demand that the initiator ploughs through the white papers, background correspondence and discussion documents. Otherwise, launch into a rambling personal reminiscence, involving the squeezing of vast databases onto floppy disks and 8 megs of RAM, using punched cards.

www.dbooks.org

The 'Is This Wise' method

When the initiator comes up with the idea, flinch as though involuntarily, and glance nervously about. Introduce vague concerns, without being too specific. It helps if you can mention similar initiatives that have failed. If cornered, mention various technical projects sponsored by the government, as everyone can believe that these fail. If the initiator persists, get him to do a highly-involved risk analysis, and to prepare a detailed paper on the security issues, data protection act, or the EEC regulations.

The 'More-Details-Please' method

This is an unsubtle method, but is well-tried within the industry. Pretend to be keen and interested –"Yes, this is great! We could extend it to take in X and Y, or even Z!" However, keep asking for more and more details, plans and justifications. Ask for 'a 'matrix of users against functionality', and a 'detailed analysis of maintenance issues'. The initiator will sooner or later abandon his proposal when the whole initiative eventually collapses under its own weight.

The 'Management Babble' method

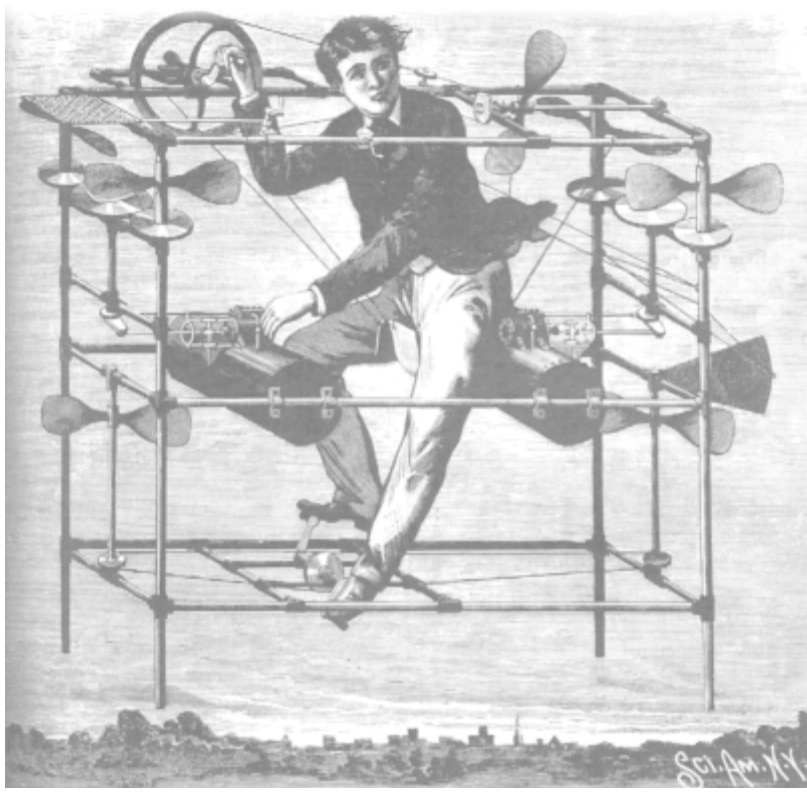
Use management jargon to confuse the initiator. Make sure that he does not know more about management than you: otherwise embarrassment could ensue.

The 'No-Problem-Exists' method

Deny the very existence of the problem. Look quizzical and say that you're not sure about the 'Business Drivers'.

The 'Let's-Make-It-Happen' method

This technique takes its effectiveness from the apparent enthusiasm of the decision-avoider. You welcome the initiator, and involve your whole team, burbling phrases such as "Guys, let's make this fly!" You then carefully diffuse responsibility so much that no one knows who is supposed to do what. The scheme should disappear without a trace.



Guys, let's make this fly!

The 'Displacement Activity' technique

This technique is well known in IT Systems. If a decision threatens, you simply introduce a complete diversion such as 'Tidy Friday', 'Embracing Diversity', or 'Green Wastepaper Baskets'. Training staff in the correct way of picking up heavy equipment, or seminars in the legal repercussions of the latest Health and Safety legislation, are highly effective. I have seen whole IT departments brought to a standstill by this sort of HR 'decoy initiative'.

The 'Better Mousetrap' technique

This technique involves embracing the idea wholeheartedly, and with apparent gusto. Pretend that the idea is the best thing since sliced bread or, at least, XML. Talk of 'embedding it into our strategy' whilst simultaneously promising that all due credit will go to the initiator. The initiator will be completely thrown off-balance by the novelty of the response and withdraw in

www.dbooks.org

confusion. Beware, though, that this will be counter productive if you have misjudged the initiator. If he misguidedly takes encouragement, then revert to one of the other strategies.

The 'Depressed Manager' approach.

Use this technique only if cornered. You start by saying that the initiator's idea addresses only one of the many key issues that you have to deal with, and that your life at work is intolerable due to the pressures. References to personal problems are an effective reinforcement at this point. Medical problems are also useful, especially if they are disgusting. The initiator will be completely distracted, or will recoil at the thought of having to listen, and will beat a retreat.

The 'There's No Budget' method

Explain blankly that there are no funds around to implement the proposal. Blame this on the obtuseness of the Finance Department, or the short-sightedness of the Board. Alternatively, just say 'We haven't the manpower/resources/head count'.

The 'Call in the Management Consultants' method

Management consultants are consummate masters of producing reports that make decision-avoidance almost necessary for organizational survival. When things get really bad, and you are likely to be forced into making a decision, call in the management consultants. Expensive, but foolproof.

www.dbooks.org

Irregular verbs for IT managers



Phil's essential guide to interpreting IT Manager-speak.

The way you describe events, decisions and actions in IT development depends on who you are. The following chart should give the general idea on the way things can be spun. I feel sure I have left a lot out, so I'd be most interested to hear of omissions.

I understand the broad implications of the technology.	You are trying to keep up with the technology.	He is playing with the technology.
My development project succeeded in broad principle.	Your project was not totally satisfactory.	His development project was a complete disaster.

I am rightly indignant.	You are annoyed.	He is making a fuss about nothing.
I conform to existing technical standards.	You are hidebound in your thinking.	He is in the technological dark-ages.
I understand the broad business vision.	You are not technology-focused.	He does not understand what is technically possible.
I write business papers.	You are required to read business papers.	He puts business papers in his budgie's cage.
I have convictions about the most suitable technical platform.	You are prejudiced about the technology you use.	He is totally biased towards/against Microsoft Software.
I am fastidious in my work practices.	You do your job in a fussy way.	He is an old woman.
I am firm.	You are obstinate.	He is pig-headed.
My project has met the revised timescales.	Your project is later than the client was originally told.	His project has overrun its schedule.
I respond to changes in the industry.	You have changed you mind.	He is always vacillating .

www.dbooks.org

Phrases with which to discourage ideas



No IT Manager really likes having to deal with ideas. They tend to destroy the harmony and equilibrium in the workplace. Never fear, Phil's list of put-downs will drop even the liveliest idea dead in its tracks.

This is the third in my series of hints for aspiring IT managers. Here, we give you put-down phrases for discouraging ideas. No IT Manager really likes having to deal with ideas. They tend to destroy the harmony and equilibrium in the workplace. They might help the company, but will bring nothing but odium to your department.

This list should also be useful for anyone determined to get ideas acted on, as all you need to do is to go through the list and make sure you have an antidote, solution or riposte ready.

Basically, you're just technology-driven.

That wouldn't fly.

Do me a one-pager with a facer.

We need a matrix of users against functionality.

Does it need to link into our existing systems?

Is it compliant with SOX/ISO9000/FSA?

Get the cost and weight effect.

We need to get our existing suppliers to do it for us.

Give me a timing plan.

Has anyone ever tried it? What are the reference sites?

How many 'function points' does it have?

What sort of headcount-reduction can we make?

I think we need to discuss this further.

I'm not prepared to fight that battle.

Is it in the Budget?

It is not supported.

It's not strategic.

Let's form a steering committee.

Let's pass that on to the Purchase department.

OK, but what about Manufacturing/Marketing/Finance.

Please document the 'next steps'.

That's contrary to policy.

That's ridiculous.

That's too radical.

That's too superficial; it does not tackle the underlying problem.

There would be too much network loading.

We haven't the manpower.

We haven't the resources/head count.

We need a user survey.

We need proper documentation for your idea.

We need to get 'buy-in' from the group.

We need to get the users to fund that.

We'll have to get the Security Team involved.

What about the maintenance issues?

Write me a paper on the subject.

www.dbooks.org

Yes, we thought of that one a while back.

You just don't understand the business issues.

www.dbooks.org

Initiating a project with a Strategy one-pager



Many new IT managers fall into the trap of regarding strategy documents as the cornerstone of an organization's understanding of the business problem, and its most apposite solution. Dearie me, no. In order to be successful a strategy one-pager must have a beginning, a muddle, an end, and more bullets than an Al Capone movie.

This is the fourth in my series of hints for aspiring IT managers. Here, we give you the template for the one-page strategy document required for initiating an IT project in a corporate setting. This will turn a lengthy process, that hitherto has required much thought, into a simple cut n' paste exercise.

Before we give you the templates, I should make a few points clear. 'One pagers' are not written to inform the reader but to protect the writer, so, in order to understand the reality of what is being proposed, you have to read between the lies. 'One-pagers' should be obfuscated. To this end, they have a beginning, a muddle, and an end, and should have more bullet points than the St Valentines Day massacre.

The 'One-pager' should not mention anything that could possibly arouse anxiety. There are no problems, only benefits. Costs should not be mentioned, and infrastructure, maintenance, and resilience planning should not be alluded to. The paper should present a soft-focus flattering picture, much like a bra advert in a women's magazine.

A special word about Grammar. Sentences should be clipped of pronouns, prepositions and other trivia. Every paragraph should start with a bullet point. These are special IT Management Bullet Points, with a completely different meaning to the normal: it signifies that the paragraph has an implied ending '... if that is all right with all you managers' and, if read aloud, should be done so in a placating tone of voice with rising intonation.

Factor's Law on writing One-pagers is that the time required to write a one pager with a facer is inevitably two more days than you've got.

Generic IT Project One-Pager

Background

A key business requirement has been identified to improve business processes, increase profitability, increase efficiency, and improve customer satisfaction by developing an all-purpose one-pager.

The IT Systems Office recognizes that, since all papers will be completely rewritten by each tier of management (before reverting to the first version) the actual content of the first draft doesn't matter.

Achievements

A cross-functional team has been put in place to gain buy-in to the principle of forming a sub-committee to supervise the drafting of a straw-man by an Intern or Industrial Trainee.

The users/sponsors have committed to come up with the name of the new system and develop a video simulation for Executive review.

XML and Web 2.0 technology opportunity given a bold heading in Management review.

Issued this paper for review to huge email distribution list

Preliminary system scope identified potential savings of

0.01% of warranty costs and three equivalent supplier heads.

www.dbooks.org

Increased the rate of decrease in the year-on-year profit reduction in European markets.

Plans

An outline process, complete with Executive and Steering committees, working groups, task forces, will be launched in the First Quarter of next year.

A detailed plan for a plan to schedule some work amongst the management reviews will be issued.

Rotate all senior analysts on the team to other jobs and replace them by enthusiastic newly trained recruits who identify the 'quick fix' to that 2 year problem, involving the use of Java and EJBs on a LAMP platform.

Support low inventory objectives by designing on-line BI balanced schedule functionality for all sites - link to SOE, Bivalve Ballcocks, rapid flame thrower technology, Orgone accumulators and other non-specific complex-sounding downstream equipment.

Identify a means whereby the project helps to save the planet by using low energy, recycling, and carbon neutrality. Failing that then use Open Standards, and offer rich business performance information.

Issues

Multiple issues required to avoid the impression that the task is simple.

Need to ensure system delivers benefits in line with current fashionable flavour (cost, quality, health and safety, ISO9000, lead-time etc.).

Planned actions

Define phase one scope for quick launch (year after next) and increase project team from two people, consider bland issue which no-one really understands, but is generally considered to be a big deal.

Read last year's business plan and copy the listed benefits, but take care to juggle the order.

www.dbooks.org

Looking Busy



The art of looking busy and efficient, even when one is really just arranging one's social life, or power-napping, is an essential skill for any IT manager.

This is the fifth in my series of hints for aspiring IT managers. Here, we tell you how to look busy and efficient. The most important thing for an IT manager is to look busy. In fact, it is a technique that dominates the management of the public sector, where whole office blocks are filled with people earnestly engaged in nothing more than sending emails to each other full of positioning papers, strategy documents, directives and plans. Looking busy is one of the primary skills that is required in a manager, and one looks in vain for sage advice on this in the manuals on management that are sold in airports. To fill a crying need, I offer the following tips.

Be creative in staggering your hours: I once knew a manager who arrived early at the office and always was the last to leave. It impressed everyone, and nobody seemed to realise that he, legitimately, took four hours off in the middle of the day.

Spend your time at a PC. The PC is a wonderful invention because it is almost impossible to detect whether the user is doing productive work or not. An entire office can be seen bent over terminals engaged in work of mysterious

complexity, when they are, in fact, doing nothing more than updating their facebook profile or organising their social life.

Project-management software is essential. This is because just a small amount of data will produce reams of print-outs with esoteric graphs, calculations and reports. When ever anybody queries the status of one of your projects, just hand them the folder with all the printouts. The more data you put into project-management software, the more wildly impressive it looks.

Attend lots of meetings. It is important to note that if you call lots of meetings with other people in your own company, they eventually get bored and don't turn up. Suppliers, and potential suppliers, are different: they always turn up, especially if you are buying from them. Even if you don't, you can string them along for a long while, and then you can just choose a different set.

Engage in 'crisis management', pursue just the most visible parts of the job. It always helps to drop into conversation a mention of the feats of management you regularly accomplish.

Make sure you get lots of SMS, phone and mobile calls (you will achieve this by putting ads in the local papers).

Make lots of visits to other offices within your organisation. There's no need to do anything when you're there, but it looks good, and is useful for dropping in the conversation subsequently, thereby paying a double dividend. Once everyone is aware of your peripatetic habits, nobody will even realise when occasional visits are to the golf course.

Keep a box full of files which you strew all over your desk in neat piles when you get in. Before you go, just stack them away again, thereby getting a reputation for efficiency as well. A successful modification of this technique is to dump the files into your in-tray, thereby impressing all with your workload as well.

Revise all the documents that your subordinates submit to you for approval. You will enjoy this too, since there is no passion like the passion to revise someone else's document.

Spend lots of time organizing your time. That way, you look busy and efficient, even though you will not have the time left to do the things you have organized yourself to do.

When challenged as to why you have not done something you chuckle confidently, say 'that would have been too simple, what we are doing is studying the underlying processes, and understanding the fundamental requirements of the users', and lean back in your chair with a smug smile.

www.dbooks.org

Never miss a chance to (loudly) complain how many mail items arrive whilst you are away from your desk (this is a major indicator of your importance). It pays to turn the sound up on your PC so others can hear the emails arrive.

Engage in a vigorous Email correspondence with a large number of other managers, making sure you rush round telling everybody about the huge number of unread Emails in your mailbox. Achieve this by putting a question at the end of every email you send out.

In meetings, be prepared with a number of Powerpoint presentations. The quality is immaterial as these are devised to reduce your audience to a state of induced catatonia. Always aim to say as much as possible. If there is a white board, be sure to get control of the felt-tips.

www.dbooks.org

Doing things - The Manual

Phil reflects on the 'good old days' when the only steps between deciding to do something and actually doing it, were rolling up your sleeves and loosening your tie.

In the past, in order to get something done, you did something. This normally involved taking off your jacket; possibly even loosening your tie. It could be you needed gloves and goggles, maybe a mate to hold your tools, but generally, you got on and did it.

Now, dear me no. Now there are so many steps between deciding to do something and actually doing it that by the time you're halfway you've generally forgotten what it was you wanted to do in the first place.

Decide on need to do something.

Attend 2 day training course.

Submit signed statement of commitment.

Engage Stakeholders, communities, agencies, and other staff .

Establish a task group.

Carry out an audit and risk assessment.

Identify priorities and targets in line with key policies.

Collate Baseline Data.

Complete and submit an action plan, itemising key objectives.

Obtain management signoff for action plan.

Meet with management team as appropriate.

Attend network meetings.

Review and evaluate.

Return monitoring forms to your supervisor.

Return updated action plan as appropriate.

Prepare for doing something.

Do something (practice run).

Compare new data with baseline.

Final report to management teams for signoff.

Actually Do Something.

Of course, whilst doing something, one has to consider the 'people side' of the change you're implementing. I offer the following as a "best practice framework for identifying and managing the five critical success factors in the 'people side' of any proposed change".

Thinking, planning and doing the right 'people things' throughout your change initiative.

Building and sustaining the right levels of commitment to make your change work.

Understanding what people issues could be enablers or blockers.

Focusing on those issues that are critical for the success of your change increasing the capacity for benefits realization.

Applying sound change management principles.

All clear? No? Then back to Step 2 for you.

www.dbooks.org

The Incident of 'The Two Johns' – an IT Manager confesses

Keep your friends close and your enemies closer; but try not to muddle which is which. Phil exploits his cloak of anonymity to confess to a grievous mix up in his managerial appraisal of two team members.

As an IT Manager, I used to love doing staff appraisals. These rituals come by different names in different companies, be it 'performance review', 'annual report', 'staff assessment' and so on. There are periodic crazes for tinkering with the process, and pinning a different name on it, but the tradition itself remains basically the same.

I appreciated the value of standing back and taking stock of how ones 'direct reports' were shaping up. To mull over the events of the year, advise them as best one could, and to do what one could to help the advancement of their careers. I looked forward to the process … until the incident of 'The Two Johns'

In my development team, I had two programmers who couldn't have been more different. They were both called John. Clever John found the world of work easy. He ploughed through every job he was given with supercilious ease. He was a useful member of the team, though he seemed to get bored with mundane jobs and often overcomplicated his code, just to keep himself amused. Dull John, by contrast, was a plodder. He didn't seem to read enough to keep up-to-date and he was inclined to ask for help at every turn. He was content to do just enough to tick along but stay invisible in the team.

When appraisal time approached, I put together reports for every member of the team, as usual. Because human nature dictates that it is easier to remember negative things about people, I used to keep a notebook with me to record every good thing that happened in the day-to-day work of the team. By the end of the year, the appraisals flowed naturally from these notes.

Clever John's appraisal was a work of unstinting praise, emphasising his readiness to take on more responsibility and the maturity of his attitude to the work. I liked the way he took on the 'ownership' of problems and issues when they happened and told him so. Dull John's report was more of a struggle. I

rather wanted him to realize that he had a lot more potential and that he should take a bit more control, and drift less. Although the report was as bland and complimentary as such things have to be, I put the message pretty strongly, and wrote that everybody was hoping for more from him, as we'd all benefit, both he and the company.

When the day came, I issued the reports and worked through the interviews. Dull John was first in. Although I had been as careful as possible, I was still worried that my report might have offended him. To my complete surprise he was all smiles, and expressed his entire agreement with the report. He had no issues at all and shook me warmly by the hand.

As he stood up to leave, I glanced down and noticed that he had Clever John's appraisal. Somehow I'd mixed up the two reports. Working late, I suppose. The confusion of the same name, I shall never know. The brain can play tricks when one is tired.

I sank down despairingly in my chair wondering what to do next. Suddenly I realised the implication of Clever John getting Dull John's report and I was gripped by a sudden panic. At that very moment there was a tap on the door and in walked Clever John.



I sank down in my chair with despair

www.dbooks.org

Again, to my intense surprise, Clever John's face was wreathed in smiles. Once more I was warmly shaken by the hand. The report had given Clever John a jolt, but, thinking it over and chatting to friends about it, he realized how correct I was. After the hard graft of University, he found the work here too easy; he'd slid into complacency and was just drifting. It had taken my annual appraisal to make him realize that the time had come to focus on developing his career.

Once more he complimented me on my insights, and left me a gibbering wreck completely at a loss to know what to do. Of course, I decided to do nothing. Suddenly, both Johns were the stars of the team and began to get noticed within the company for their drive and initiative. Dull John drew me to one side and confessed to me that I was the first boss he'd had that really appreciated his work and his true potential. I visited his home, and was delighted to meet his wife, now pregnant for the first time after several years of marriage. I became good friends with both Johns and it was not long before they both left the team with promotions, and they both began their meteoric rise through the company and beyond, which continues to this day.

I suspect that there is a moral to this story but I don't know quite what it is. A random mistake did more good than all the carefully-crafted appraisals I ever did. I'm just glad that the cloak of anonymity allows me to make a full confession.

Section II: The Strange Business of Software

"*Have you ever played a computer game where you stand there and ghastly slavering monsters come charging at you out of nowhere, one after another, making blood-curdling noises? You are required to stand there killing them one after another. I can't play them: it is just too much like the real life experiences of a successful software vendor*."

So begins Phil's reminiscences on the strange business of software. After a few decades in the industry, Phil has lived through many of its mad ideas and cyclical phenomena, and is still around to tell the tale. From the halcyon days when marketing your software meant buying advertising space in one of the two PC magazines, through the wild west dot.com boom days, when IT men wore sharp suits and winning smiles and business plans were optional, to modern day Microsoft with its software bloat, and marketing-speak that often defies parody.

Out of his tales fall a few eternal truths about the software business that people ignore at their peril, but ultimately do anyway.[TD]

www.dbooks.org

Going It Alone

Phil has a 'Madame Mim' moment when a valued programmer leaves to start his own software company.

One of the best programmers that I ever recruited dropped the bombshell after a year's harmonious association, announcing that he was leaving to start up a company, selling software.

It wasn't the first time such a thing had happened but I became distressed, not just because I was losing a good programmer, but because the guy had become a friend and I could see his fate just as clearly as if I were Madame Mim and her tea leaves.

The more I pleaded, the more obdurate he became.

"This software you're selling; is it selling well at the moment?" I asked.

"We haven't written it yet, but we're sure it is going to be popular."

"Have you got a business plan?"

"Eh? What's that?" Pause. "Phil, are you all right?"

(Cue strange music, the screen goes watery)

I was not dead or apoplectic, just lost in reverie. It was twenty-seven years ago that I first became a 'software tool vendor' as it is now called. I'd built one of the early CP/M-based microcomputers. They came as a 'kit' - just a whole lot of PCBs, components and a pile of chips – because assembled, working micros were too expensive for an individual to buy. When the beast finally fired into life, a few months later, I knew a great deal about the way it worked. CP/M was the only operating system around and Microsoft Basic about the only language. I was a typical user: I filled the floppy drive with a great deal of work and didn't do backups properly (backups with only one floppy disk, and nothing else, are problematic). I then accidentally deleted the contents of the disk.

In a frenzy of despair and worry, I wrote a fairly simple utility that 'undeleted' the files I'd deleted. After a long struggle with the machine,

I had it working, and was pretty handy with assembler code. I was so pleased with it that I gave it away free to the CP/M users group.

I quickly realized that it was *very* popular.

I had stumbled on a simple secret of success. This program had all the virtues of successful utility software in that it solved the problem of guys in a crisis with a thick wallet. I was also the first to turn this particular, simple process into a product that anyone could use. Quickly, I tidied up my public-domain code and started to market it via a friend who owned a local computer franchise. In those days, it was a simple matter of buying advertising space in the PC magazines; there were only two of them in the UK at the time.

The resulting success of this code kick-started me into an IT career, and very soon I had a software house with a staff of twenty, writing and selling all manner of software products such as payroll systems, customer databases, and accounting packages. It all seemed terribly easy.

It wasn't.

Have you ever played a computer game where you stand there and ghastly slavering monsters come charging at you out of nowhere, one after another, making blood-curdling noises? You are required to stand there killing them one after another. I can't play them: it is just too much like the real life experiences of a successful software vendor.

These slobbering, warty monsters come with ghastly inscriptions such as VAT Inspector, Solicitor, Competitor, Employee, Journalist and Taxman tattooed on their receding foreheads. Nobody really likes success in others, least so the public-sector employee. These creatures employ subtle malice to cause stress and create havoc.

On top of that, one has to contend with the notoriously cyclical nature of the IT industry. Everyone remembers the good times in IT. The good times are wonderful. I shall never forget, at one Computer Exhibition, looking up from our stand at the sea of faces, cheeks aglow, firmly intent on buying our software. We had to press friends and relatives into manning the stand to take all the money.

Few people, however, remember the bad times. But they come with the inevitability of the grim reaper. At the same exhibition centre, a year or two later, we had to pretend to talk to each other on the stand to maintain the illusion of vibrancy and cheerfulness; the occasional

www.dbooks.org

punter who drifted past could not even be persuaded to accept the software for free.

One friend of mine, an IT contractor, came up with an interesting solution to the 'bad times'. He trained as a butcher as well as a programmer, and just switched careers on every downturn. However, it is far from easy for a Software Vendor, with a large wages bill, to ride the wave.

"Oh, I'm OK," I said coming back from my reverie, and blinking at my best programmer, and friend, who'd just told me he was leaving.

He looked at me expectantly. "Well, what do you think?" he asked. "After all, you've done it."

"Bless you and good luck" I said, shaking him by the hand. "There will be a job here for you if it doesn't work out."

I almost said "…**when** it doesn't work out", but checked myself just in time.

I knew from experience that this was something that any good programmer had to get out of his system. There is a deep-set fallacy that if software is well-written, and full of features, then it will sell. This is like saying that a well-performed song will get to the top of the singles chart by sheer merit. The analogy is fairly close, because in both cases one has to hit the zeitgeist, the spirit of the age. Quality helps one's chances, but is, by itself, insufficient.

I keep in touch with some of my contemporaries, who once ran successful companies, writing and selling software tools. One breeds racehorses with only moderate success; another is trying to get a local television station off the ground; another is a consultant with a large ISV. None of them are still in the business. One of my friends, who once made good money from writing and selling software tools, confessed to me that once he counted up all the hours he'd spent, he worked out that he'd have earned more money babysitting.

I keep wondering why I even think about developing software 'on spec'. I suppose it is the giddy, unforgettable, excitement when that one application in the ten that you write takes off. But easy it isn't.

www.dbooks.org

The Walrus and the Manager

Why do Phil's eyes water whenever he hears the poem 'The Walrus and the Carpenter'? Is it the voice of his conscience or memories of struggles selling software services to large companies? Why does he identify so strongly with the eldest oyster?



'... and shed a bitter tear'

I once worked for an international company who did a study to find out why so many of its suppliers had disappeared. The subsequent report made disturbing reading. The report revealed what could best be described as the 'Walrus and the Carpenter' effect. Even today I can't read Lewis Carroll's poem, The Walrus and the Carpenter, without a lump in the throat.

Much of the work of a manager in a manufacturing company is to create and foster relationships with suppliers. It is true of many IT departments too, where many applications and services are bought in, rather than created in-house. Who'd want to write and maintain their own payroll package, for example? However, in such a relationship, there is a lot of work to do to make sure both sides obtain maximum benefit. If you get it wrong, it can cause real suffering.

The poem introduces the walrus and the carpenter walking along the beach at night. To me, the Walrus represents the complacent corporate organisation. The lethargy, and the layers of blubber, speak volumes. The Carpenter is obviously a project manager.

The Walrus and the Carpenter
Were walking close at hand;
They wept like anything to see
Such quantities of sand:
"If this were only cleared away,"
They said, "it would be grand!"
"If seven maids with seven mops
Swept it for half a year.
Do you suppose," the Walrus said,
"That they could get it clear?"
"I doubt it," said the Carpenter,
And shed a bitter tear.

Note the subtlety of the dialogue; the naïve trust of corporate reasoning that projects that are 'slipping their dates' just need additional headcount to ensure that they come in on time, and the wise riposte of the Carpenter. In IT, such a practice usually leads to tears.

I learned very early on in business that it is safest to enter into commercial relationships only with companies of a similar size to your own. This is why IT salesmen who sell applications and services to corporates always trot out their revenues and projected growth at the start of their PowerPoint Presentations. An experienced manager will be nervous of buying from a small company - 'The Oyster': But not these two characters, it seems.

"O Oysters, come and walk with us!"
The Walrus did beseech.
"A pleasant walk, a pleasant talk,
Along the briny beach:
We cannot do with more than four,
To give a hand to each."

The corporate is, suspiciously, encouraging the 'oysters'. For a small supplier, the idea of forming a relationship with the IT department of a large company is beguiling. Once you are in place, with a good maintenance contract, the future looks rosy. Many IT managers in large businesses will encourage this idea of 'synergistic' business opportunities with smaller partners, as they know that it will generally give them the 'upper hand' in most of their dealings. However, the oysters should be very wary of placing their fortunes

www.dbooks.org

and future completely at the mercy of the company. Suddenly we meet the old 'Phil Factor' oyster. He has already seen the consequences.

The eldest Oyster looked at him,
But never a word he said:
The eldest Oyster winked his eye,
And shook his heavy head--
Meaning to say he did not choose
To leave the oyster-bed.

Tie up the resources of a growing company in a commercial relationship with a single large one? Get distracted from developing other commercial links by the promise of easy money from this single source? What happens when the contract is renegotiated, and you've let other revenues atrophy? The oyster will want to neglect the natural predatory instincts of the walrus, but he does so at his peril. Sadly, the temptation is just too great for the less experienced.

But four young Oysters hurried up,
All eager for the treat:
Their coats were brushed, their faces washed,
Their shoes were clean and neat--
And this was odd, because, you know,
They hadn't any feet.

Out of their natural element they go, beguiled by the false expectations of doing business with a large company, and deaf to the warnings of the old oyster, Uncle Phil. From the description, it is the sales reps who make the running. Wiser counsels are usually forgotten in the stampede for corporate sales.

The problem is that small companies often require a large investment just to survive the ups and downs of negotiation over a contract, which can be protracted. They generally don't have the funds readily available and so take out loans in expectation of the revenue streams to come. Even if they succeed, and win a contract, they often have to expand just to service that contract.

Four other Oysters followed them,
And yet another four;
And thick and fast they came at last,
And more, and more, and more--
All hopping through the frothy waves,
And scrambling to the shore.

And how the talking can go on … and on. Meeting after meeting, with always the promise that this last presentation will close the deal. Then the manager is changed and you have to start again.

"The time has come," the Walrus said,
"To talk of many things:
Of shoes--and ships--and sealing-wax--
Of cabbages--and kings--
And why the sea is boiling hot--
And whether pigs have wings."



'The time has come to talk of many things.'

The weeks turn to months. Then come the changes in requirements, the changes in scope and the delays. Sometimes these delays are for no better reason than the manager has lost the file or is distracted by things that seem more important at the time. I knew a sad senior manager who called in salesmen from potential suppliers for meetings, merely because they were the only people who were ever pleasant to him.

It is fine for a medium sized or large business to take on a large account. The resources are there, they have people specially trained to do account management. The small businesses, on the other hand, are forced by their size to field people without the training for the ordeal.

"But wait a bit," the Oysters cried,
"Before we have our chat;
For some of us are out of breath,
And all of us are fat!"
"No hurry!" said the Carpenter.
They thanked him much for that.

www.dbooks.org

And then there is the awful moment when it dawns on an over-extended small company that a synergistic relationship was never on the cards. The large company has beguiled them into doing a free business analysis, received a great deal of specialised information, and will often have got them to write all the documentation for them.

"A loaf of bread," the Walrus said,
"Is what we chiefly need:
Pepper and vinegar besides
Are very good indeed--
Now if you're ready, Oysters dear,
We can begin to feed."
"But not on us!" the Oysters cried,
Turning a little blue.
"After such kindness, that would be
A dismal thing to do!"
"The night is fine," the Walrus said.
"Do you admire the view?"

Eventually, the banks who have loaned the money to the small company realise that a deal isn't on the cards and call in the loan. The company will have put so much energy into the deal that it will make them very vulnerable. Even if the deal goes through, the Oyster suddenly realises the dangers of shedding all those fixed assets that have been put in place purely for the project, if the deal is terminated or renegotiated.

"I weep for you," the Walrus said:
"I deeply sympathize."
With sobs and tears he sorted out
Those of the largest size,
Holding his pocket-handkerchief
Before his streaming eyes.
"O Oysters," said the Carpenter,
"You've had a pleasant run!
Shall we be trotting home again?"
But answer came there none--
And this was scarcely odd, because
They'd eaten every one.



'You've had a pleasant run'

There was, certainly a great deal of hand-wringing amongst the senior management of the company when they discovered why 'answer came there none' from their suppliers. The company was suffering for the sins of its own management, who had exploited their positions. A great deal of regret was expressed. Practices were tightened up; even the one of using one's suppliers as unofficial banks by being late in payments was banned.

The carpenter seems surprised at the demise of the oysters. Indeed, many project managers in government or large corporates have no idea of the damage their delays and dithering can do to a smaller company. The walrus, however has deliberately planned their destruction even whilst protesting his regret and compassion. It isn't entirely a mistake.

Oysters may be eager to join the big boys at the dinner table but responsible organizations will only dine with those of a similar size to themselves. On the seashore of commercial life, it is harder, after all, to absent-mindedly eat a walrus than an oyster.

www.dbooks.org

Click-A-Crematorium

The dotcom boom was the Wild West days of the IT industry. Anyone with half an idea and a winning smile could secure backing; no business plan required. Phil saw many of these ideas come and mostly go, but wonders if it's finally time for 'Click-a-Crematorium'.

During the last dot.com boom, I had a most agreeable job advising investment bankers on the value of some of the propositions that were put to them for funding. The company, an 'Internet Incubator', specialised in getting viable internet-based enterprises off the ground. City Investors were approached by many of the organisations that believed that they had moneymaking schemes that would revolutionise commerce.

Those days were very like the hysterical phase of the South Sea Bubble of 1715. That far-off time abounded with such schemes as the building of floating mansions, distilling sunshine from vegetables or the manufacture of square cannonballs to be used against Muslims. Many of these schemes attracted investment from a public that had seen vast fortunes made. Like the dot.com boom, or the current house-price rise in the UK, it all seemed so easy.

A variety of propositions that had caught the eye of the investment bankers were paraded in front to us for our assessment. Our job was to adopt, and turn into reality, any that looked like being a 'flier' and my purported role was to define and implement the technology. It was very hard not to laugh at the many entrepreneurs who presented their proposals to us. They had forgotten the awful truth that the Internet was merely a useful extension to existing business practices. An experienced retailer was far more likely than anyone else to run a successful Ecommerce business. Most of these people had not even studied the area of commerce and here they were hoping to revolutionise it. They were, without exception, utterly convinced that their ideas were wonderful and that they were on the brink of a fortune. They swore us to secrecy and made us sign ridiculous non-disclosure agreements before they would breathe even a word of their splendid ideas.

My favourite presentation was the wonderfully-named 'Click A Crematorium'. The idea was to allow you to select from any of the

crematoriums in the country for your departed loved one. If, one supposes, one felt the angels calling, one could surf the brochures extolling the sylvan glades of the memorial parks, or read with wonder the technology of the furnaces, and book one's spot. There were others too, such as a curry home-delivery service wherever you lived in the UK.

We made the mistake, at first, of either collapsing into giggles, or of telling them that they had more hope of becoming the next pope than of getting their ideas to work. We were upset at the thought of the wasted money and work that would be the inevitable consequence of their determination to go ahead with their schemes, and we tried to dissuade them. They were extraordinarily offended and immediately ascribed our contrary opinion to our stupidity, or to a cunning scheme by us to steal their ideas. The bankers reproached us for our lack of tact, especially as they'd thought the ideas to be the best of the bunch.

We needed a more subtle approach. One of our fellow panel members was completely caught up in the zeitgeist and seemed to genuinely believe in most of the ideas put before us. He positively glowed with enthusiasm at every presentation and usually had to be drawn to one side for a dose of reality-serum. I hit upon the idea of following his lead. From that point on, we aped his enthusiasm, and expressed our unqualified praise and admiration for the ideas presented to us, however bizarre or misguided they were.

At the end of the meeting, we would reiterate the deep impression their ideas had had on us, shake them solemnly by the hand and ask to see their detailed business plan so that we could move on to the next phase

Business Plan? Invariably, they agreed with slightly glassy smiles. Without exception we never heard from them again. Mostly, the effort of the task repelled them so much that they abandoned their schemes, but a few did their plans and discovered with shock that, even if all went well, there were some noughts missing in the profit figures.

I was, of course, careful to double-check with the subsequent progress of all the schemes that we'd reviewed. To get it wrong would have been unforgivable. They had all disappeared within a year.

However, I still have nagging doubts about that 'Click a Crematorium' idea. Maybe it *was* a good idea, just before its time. My aunt arranged her own burial once she knew that her death was imminent. It was a wonderful event, with a wicker coffin, in romantic woodland in the depths of the country. She found the Internet perfect for her task, and the planning of it provided her with much interest in her declining days. An elderly neighbour of mine, when arranging his mother's funeral, discovered that he could get a discount if he did his at the same time, and he has proudly showed me his tombstone in the churchyard,

www.dbooks.org

complete with his name on it. It had everything on it except the actual date of his death. He keeps a photo of it on his mantelpiece just to remind himself to pack as much fun into life as possible.



Click a Crematorium

As the moths are now getting into the post-war-bulge generation, mortality is now chic. Now is the time, perhaps, for that final frontier of the Internet age to be reached. Hmm. Time to phone my chums in Investment Banking. But firstly, I shall do a business plan.

www.dbooks.org

A Chilling Prophecy

'Integrity is the rock on which western business was founded'. Could it be true? Phil hopes not for all our sakes.

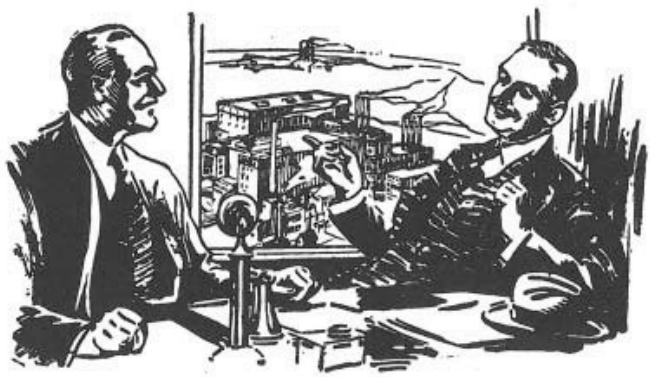
When I was a schoolboy with fantasies about being a successful businessman, I went to a lecture given by one of the most successful entrepreneurs of the time.

The white-haired gentleman who gave the lecture was the complete antithesis of what I had expected a successful businessman to look like. He was quiet, courteous, respectful and reflective. He proceeded to spell out the qualities he looked for in young people aspiring to make a success in the rough and tumble of the City. He emphasized over and over again that a lot of talents were required but they were as naught without integrity. It was integrity which formed the rock on which western business was founded.

Integrity? I had to go and look the word up.

Various awful experiences have happened to me in business, the Business of Software, that have convinced me that this seeming platitude is of overwhelming importance. Being utterly straightforward in all ones business dealings is such an embarrassingly boring piece of advice that one hesitates to give it. The problem is that everyone has to do it. In fact, the industrial revolution of the Victorian era couldn't have happened without the ethical constraints of the close-knit trading community based on mutual trust, with the catchword 'My word is my bond'. It is an unnatural state for any human group barring a successful army. You only have to study the way business was conducted in London, Manchester or Bristol in the 1860s, without any information technology to see that there was a magical ingredient. Despite what you might have heard, Chicago, Paris, Tokyo, and New York all prospered under a similar cartel of traders who chose to be painstakingly honest with each other and ruthless with anyone who broke the code. No amount of regulation, clever IT, or policing can substitute for this in today's business community, as recent events have proved. The current recession is based not on a breakdown of the monetary system, but of the essential honesty of the management of the banking system, and of the businesses that depend on it.

Some time ago, when I had an office in the City of London, there was an old firm of stockbrokers downstairs. The three partners were well past retirement age, in a dusty, poorly-lit office; all mahogany bureaus, typewriters and manila files. As they made good money they carried on, cheerfully trading. We'd pop down to borrow tea-bags and get tips on stocks and shares. When they finally closed the business we bade them goodbye in the middle of a suddenly-bare office, clutching classes of sherry. They were adamant that destruction of the 'old boy's network' of brokers and traders, the deregulation of the stock exchange and banking would lead to a heady boom in trading, leading eventually to a catastrophic collapse.



The Old boys' network

They saw the introduction of Information Technology as being a mixed blessing as it made dishonesty easier to conceal, and made it easier to avoid looking your victim of fraudulent dealing in the eye. We nodded quietly at the time, but laughed afterwards at their old-fashioned caution. What a silly idea, the collapse of confidence in the banking and trading system, we thought.

www.dbooks.org

Talking Technical

The only way to get through the sandstorm of waffle that blows over the desert of the IT industry is to talk technically with technical people. When you do, the sometimes-unpalatable truth will out. Phil plays 'one of three' to a developer's Ancient Mariner, and uncovers a tale of corruption and fraud.

Should IT managers understand technical issues, or is a full appreciation of the technology somehow unworthy and unnecessary for senior staff?

In such a rapidly changing industry where fortunes are made and lost by attempting to exploit gaps in the market too thin to see with the naked eye, the answer would seem obvious. However, I have repeatedly come across amazing gaps in the technical knowledge of managers.

Developers, programmers and designers tend to be deeply geeky types, immersed in the minutiae of the technology. In stark contrast, many IT managers appear to have difficulty working the remote controls on their television. This cultural gap between the foot soldier and officer classes of the IT industry can have interesting consequences. It can also, of course, be exploited. I have done so several times in my career. Sometimes, it's essential to find ways of bypassing the management level in order to find out what's really going on …

I once suddenly found myself appointed to be the IT director of a City of London company developing internet-based trading systems. Like many such appointments, it was rather a suicide mission. I arrived at a crisis point: a sizable part of the IT effort had been outsourced to a software house that didn't seem to be delivering. After going through the paperwork, I asked the CEO to delay announcing my appointment as IT director for a week whilst I investigated. I gave him a brief outline of what I planned to do, which while broadly accurate was slightly economical with the truth. Once he understood my strategy, he gave out a vague and minor announcement that I'd 'joined the company'.

I was determined to tell no lies, but merely to encourage the wrong assumptions. Borrowing an old car from my secretary's daughter, and wearing a shabby suit that I was using for gardening, I made a visit to the software house,

explaining that I'd started work for the company and wanted to get up to speed on what they were developing for us. I stuck a row of biros in my jacket pocket, put a notepad under my arm, and shambled in to meet them. Their office was one of those strange glass constructions near Staines, beloved of IT companies. The MD, looking immaculate in charcoal grey with gold specs, nodded distantly at me, his lips curling in a slight grimace of contempt, and hurriedly passed me on to an amiable programmer. I didn't see the MD again, or any other supervisory staff, which suited me just fine.

We talked happily for a couple of hours about the intricacies of messaging in a distributed environment, and the architecture of the application that they were writing. The programmer then ran through the work that had been done – and they hadn't done very much. Considering the man-hours they'd charged for, things just didn't seem to add up.

Over a rather uninspiring canteen lunch of machine tea and egg sandwiches, he waxed garrulous, and I responded at my cynical and jaundiced best. We reminisced about the ups and downs of a career in IT and swapped tales of our years spent in front of terminals. Suddenly he was singing like a canary, and leaking like a sieve.

He explained how the expensive database designer for whose services we were paying was actually just a friend of the MD's wife who was doing a Maths postgraduate project. It was her first database. Although we were being charged for the full-time services of five programmers, the team of three were being constantly pulled away to do other work.

What I was hearing confirmed what I had begun to suspect earlier: the expenses bore no relation to reality, and the project was likely to slip disastrously. After a short while, I'd heard enough. No triumph, just sadness and revulsion. I tried to steer the conversation back to safe technological topics, but once he had started, he was like the Ancient Mariner who 'stoppeth one of three' to tell his epic tale. I discovered that the technical platform was chosen because the MD's chum had just got the dealership for the hardware. Also, that they had taken open source modules that specifically forbade free commercial use and had deleted all copyright and authorship messages. They'd then charged us as if they'd been written for us.

And so it went on. Like the Ancient Mariner he had to get the whole story out to assuage the guilt-by-association he felt. It was just rather unfortunate for the miscreants that I, the IT Director of the company they were defrauding, was the 'one of three' that he stopped.

www.dbooks.org



He was singing like a canary

What happened next is tangential to point I'm making, which is that the only way to get through the sandstorm of waffle that blows over the desert of the IT industry is to talk technical to technically competent people. And that, once talking at the technical level, the truth will out. Suffice it to say that the information blurted out to me proved to be entirely true, and extraordinarily useful to know. I confess that I will cherish the memory of the look I saw on the face of the MD of the software company when he walked into my office, with his hand held out in hearty welcome, to 'meet the new IT Director'. I must admit, too, to savouring the process of introducing him to our corporate legal team.

www.dbooks.org

The Diary of a Microsoft Marketing Man

Phil suffers reality distortion, or maybe just a senior moment, while reading an article on Microsoft's business website and ends up penning the 'Diary of a Microsoft Marketing Man'.

It seems to me that some of the literature on the Microsoft business site has become unhinged from its purpose of *communication*. In its place is a kind of hypnotic repetition of marketing mantras, designed to appeal to the emotions rather than the intellect. I don't blame Microsoft in particular; software documentation and instructions, in general, seem to have retreated into a strange private language, and taken on a life of its own, quite divorced from its original purpose.

People-ready business …*gubble gubble*… Brittle business processes …*gubble gubble*… Maturity continuum …*gubble gubble*… Monolithic line-of-business (LOB) applications …*gubble gubble*… Complex development tools and processes. …*gubble gubble*… Agile and adaptive paradigm shift …*gubble gubble*… True partners of a business. …*gubble gubble*… Interoperate…*gubble gubble*… Core Solutions …*gubble gubble*… Scalable data infrastructure …*gubble gubble*… Scalable management processes. …*gubble gubble*… Seamlessly interact …*gubble gubble*… Feature-rich …*gubble gubble*… Attack surface …*gubble gubble*… Per-service basis …*gubble gubble*… Transform valuable enterprise data …*gubble gubble*… Can leverage the capabilities …*gubble gubble*… A true corporate asset …*gubble gubble*… Ongoing business value …*gubble gubble*… A scalable and extensible architecture …*gubble gubble*… An adaptive application platform …*gubble gubble*… Strong business partnerships …*gubble gubble*… Aligning IT with the business …*gubble gubble*… Empowering people, …*gubble gubble*… Driving the business forward.

I sat up and blinked, shaking myself out of the hypnotic state that reading these phrases invoked. Do Microsoft people really speak and think like this? Maybe they do. It was an interesting thought, and shortly afterwards, I had the beginning stages of … **The Diary of a Microsoft Marketing Man**.

I woke up, in a 'one stop' process requiring a subtle paradigm shift. I was in my bed at home, a complete integrated platform. The Sun™ was streaming through the Windows™. The world outside seemed people-ready. To drive decisions on getting out of bed, I enumerated multiple insightful, timely reasons for aligning with domestic goals in a familiar and powerful way.

On reaching the kitchen, I achieved unparalleled connectivity in the kettle by plugging it in, and leveraged the capabilities of the toaster by switching it on in real time. Today, I felt I had a flexible and connected infrastructure around me. I empowered the agile and adaptive cat at every level by opening the cat-flap

I drank multiple cups of Feature-rich coffee, a default slice of familiar and powerful toast followed by an optional one on a per-egg basis, and adopted best practices in the bathroom to support daily bowel operations. I seamlessly interacted with the bathroom mirror to transform the valuable face and exceed expectations. I interoperated with the toothbrush.

To support the bottom line I got into my trousers, I added a belt in order to benefit from higher trouser reliability, improved manageability, and enhanced support. I felt I had reduced my attack-surface.

Once dressed, I could focus on the high-value task of getting to work and thereby maximize my interoperability, visibility and insight for more accurate resource allocation, prioritisation, and risk management. How optimal, I thought, to once more speed up delivery of business-critical solutions that deliver higher growth and comprehensive profitability, to achieve visibility, collaboration, and control.

So it was once more down the garden path to start the journey to my office at Microsoft ….

www.dbooks.org

Microsoft Boy announces his School Homework

Phil continues his journey into the twisted mind of the Microsoft marketing person to give you a glimpse of Microsoft Boy at school, before the start of his splendid career at Redmond.

Scene: The History lesson in school. The teacher wearily calls Microsoft Boy to his desk to try to discover where his homework is.

Teacher:

"Well, young William, (*looks over his glasses severely*) where is your homework? It should have been handed in today, I'm afraid."

Microsoft Boy:

(*with a smug ingratiating smile redolent of sincerity*) "The past week has been an amazing time for me as I geared up to announce the delivery of my essay. The response to my announcement from friends and parents has been overwhelmingly positive – in fact, even my aunt Edith wants to read it. What is catching users' eyes? Legibility, correctness, conciseness … the list goes on and on. Simply put, this history essay is a significant release for me – one that builds on all of the great things that I was able to deliver last year in the Lower fifth. I see it as a critical step forward for my academic life here, and the foundation of the broader vision for my school career. Based on what we are hearing from people who have seen the current version of my essay, it seems that everyone agrees."

Teacher

(*impatiently)* "Well, that may be the case, but you haven't actually handed your work in. Where is it for heavens sake? The others have managed to hand their work in!"

Microsoft Boy:

(*earnestly*) "Not surprisingly, one of the top areas of focus for me is always to deliver high quality homework, and in a very predictable manner. This is **vital** for my dazzling school career – which is why I've frequently discussed my goal of releasing my history essay within three months of the last one. I am on track to reach this goal." (*folds his arms with a smile of achievement*)



This is vital for my dazzling school career ...

Teacher: (whilst rustling about, searching on his desk)

"I don't see it, I really can't find your essay on my desk. It was supposed to have been handed in today."

Microsoft Boy: *(sensing something not quite right in his relationship)*

"To continue in this spirit of open communication between us, I want to provide clarification on the roadmap for my essay. Over the coming months, you, and the other teaching staff here can look forward to significant milestones in the delivery of my homework. I am excited to deliver a release candidate of the essay in a month's time, at Scout Camp, with final Release of the entire homework expected in another couple of months. My goal is to deliver the highest quality History essay possible and I simply want to use the time to reach the high bar that you, my teacher, have set."

www.dbooks.org

Teacher: (Head in hands, dispairingly)

"I really don't understand. Have you handed in your homework or not?"

Microsoft Boy

"I have not, in any way, changed my plans for launching the essay today. What I have done today is to announce to you the delivery of my essay, and I'm proud to have met this target. Please keep the great feedback coming and thank you again for your ongoing support of my 'best-in-class' academic work!" (*Proudly walks out of the classroom)*

www.dbooks.org

The March of Technology

In which Phil, under the influence of a surfeit of lobster, some fine wine and a large book, conjures up an imaginary Bill Gates, lets him know why people aren't flocking to SQL 2005, and offers him a turning off of the path to ever more complex and bloated database software.

It must have been the lobster we had for supper, but I dozed over my book, and slept fitfully. In my dreams, I bumped into Bill Gates.

"Hi Bill", I greeted him amiably. "Long time no see".

"It was 1982 to be precise", he replied, with that lopsided, curiously adolescent, grin I remembered so well. "And, I've been thinking, I was wrong to shout at you when we had adjacent stands at that conference."

Blimey, I thought to myself, the lobster must have been off.

"My fault entirely", I replied magnanimously, "for explaining rather too enthusiastically to that senior executive from Ford Motor Company that PCDOS was just CP/M with a bit of make up and frilly knickers."

"I didn't mind that so much as your long and amusing speech about Concurrent CP/M being years ahead of PCDOS."

"It was", I protested, "but you caught up eventually with NT. A shame about Xenix and OS/2 though."

"Never mind all that", said my old friend colouring up remarkably, "I need your advice on SQL Server".

"I must be dreaming!" I expostulated.

"You are", he replied. "You can't really imagine that the real Bill Gates wants your advice, can you?"

"I suppose not. Even in 1982, you weren't too keen. So you really are just an imaginary Bill, dreamed up after drinking too much Vouvray 2002 with my Lobster ..."

"Just be thankful it wasn't a large glass of Chardonnay, otherwise you'd have gotten Larry Ellison."

"… and if I had finished the bottle, it would have been Beelzebub himself, in a haze of sulphurous smoke."

"OK, enough pleasantries" said Bill, adding after a thoughtful pause. "I have a problem. I spent five years with one of the biggest development teams in history, creating SQL Server 2005. It is now 2007 and a good-sized chunk of the industry is still using SQL Server 2000." He paused again. "So, we started giving it away as SQL Sever Express, thinking that PostgreSQL and MySQL would just vanish as a result, and that existing SQL 2000 users would happily convert over to Express, and then up to the full SQL 2005. We were wrong on both counts. Everybody admits SQL 2005 is the most comprehensive database in history. It has everything. So what is wrong?"

"Simple. Look at this book I fell asleep over. This is typical. A book on administering SQL Server and it was written by nine people. The entire story of the "Decline and Fall of the Roman Empire" was written by one man, and here it takes nine people just to tell the poor souls who have the unfortunate task of administering the behemoth of a system how to do it. It doesn't quite say whether they wrote it serially, collapsing exhausted with writers cramp and passing on the metaphorical baton to the next volunteer, or whether they just partitioned off the immensity of the task between them. I find this deeply scary."

"Scary?" asked Bill, puzzled.

"Yup. I've got to plough through all seven hundred and thirty-five pages before getting back to re-reading Beauchemin and Sullivan's 964 pages, and that's before I get stuck into all the changes in NET 2 that I missed. And I have a stack of books on PHP5 to tackle. If this trend continues, Microsoft's next offering will require each administrator to read an entire bookshop. Your SQL Server 2005 Administrator's Pocket Book might fit in the pocket of a Behemoth, but I would walk with a strange limp if I managed to stuff it in my trouser pocket. Authors wrote the great books on SQL Server 7 single-handed. Today's tomes are as thick as doorstops and are written by about nine authors. By simple mathematical extrapolation, books on the next version of SQL Server will be 2000 pages long and written by a small army!"

"Well, you have to keep up-to-date in any job …"

"Well, I used to think that doctors have the worst job of keeping up to date, but they don't find the entire human body changing its plumbing every five years. Sure, the drugs change, but aspirin is still aspirin. IT people are finding

www.dbooks.org

the entire field of knowledge changing radically every five years. We're getting information overload."

"Come now, Phil, it is just that you are an old fart, who can't keep up any more. You should be growing roses and knitting antimacassars, thinking about nice comfy chairs and joining the bowling club."

"There may be some truth in what you say … but I don't think *anyone* is keeping up. Nobody likes to admit it. It is like owning up to having pitifully inadequate secondary sexual characteristics, or the brainpower of a two-toed sloth. We all like to keep up the myth of our mental adequacy, but it is a myth. It took man 100,000 years just to progress from the first stone tools to discovering how to make metal objects – and in evolutionary terms, we are complete strangers to the computer keyboard! It is a tall order for us to suddenly become the priests of the mystical power of Information Technology. It would be all right if we were developing extra brain-lobes to keep up with the galloping complexity, but, if anything, our brains have shrunk slightly since civilization began."

I warmed to the subject. "The huge breakthroughs in Information Technology were breakthroughs because they were simple, and they reflected the way that business was already being conducted. When the relational database was introduced, people found that it was uncannily like the existing ledgers, and the systems put in place by the nineteenth century logistical experts after the Crimean war, and refined ever since. The Spreadsheet is simply an accountancy summary table, and could be recognised and comprehended by any accounting clerk from the time of the Industrial Revolution."

"SQL was a staggering advance, because it was so simple. The clever and difficult stuff happened under the bonnet. For one, you told the system what you wanted but not how to go about doing it; for another, you didn't have to keep a huge number of details in your head. It was just the same with Email or the Internet; dead simple, and easy to understand. The real skill of technology is to keep it conceptually simple. So what have you been doing for the last fifteen years, Bill, but adding complexity to a system whose chief virtue was its simplicity? It requires no thought to create a tower of Babel, a vast beast riddled with Rampant Featuritis; just effort."

We fell into a gloomy silence for a while, before I continued.

"Whatever anyone asks for in SQL Server is put in, until there are eight different ways to export a query to a file, four ways of extracting data from a spreadsheet, and a Byzantine intricacy in the handling of XML – which, by the way, started out as a dead simple way of representing small databases, before it

got the complexity treatment. It is so easy to be obliging and add stuff in. The great skill is in keeping the product simple at the same time. One can imagine it happening. Some persuasive guy contacts Microsoft, suggesting a feature and, without checking if it is already there, a team gallops around in all directions developing a splendid do-it-yourself technology with a new TLA, that ends up being little more than an invitation to program the feature yourself: when all that was really required was a button."

"And there, frantically rushing around, trying to keep up are the DBAs and Database Developers. You give them the choice between a database like SQL Server 2000 that is complicated, but works and does the job they want, and one that is wildly complicated, but works and does the job they want, and then are surprised when they want to stick with the one that is merely complicated."

Bill looked at me impatiently. It was obvious that he'd heard this line of talk many times before. "OK," he said wearily, "So where next? What can one do about it?"

In a blinding flash, it all became clear to me; the vision of a product that hid its complexity behind a simple logical interface to the data. Here was a product where Microsoft no longer needed to tell us to do our own programming to get the features we needed. Here was a SQL engine that could be described in a short book, probably written by Joe Celko, or Bill Stanek.

I was about to tell Bill how to go about it when I felt myself being shaken, and the veil of sleep lifting. "Stop! Stop!" I shouted, "I must tell Bill what to do!"

The next moment I was awake in my chair, eyes glazed, textbook still open in my lap. My wife was standing over me, looking at me severely, and asking who this 'Bill' was, who ought to be told what to do. In panic I tried to recall what it was I was going to tell him. Surprisingly, I managed to recall at least a part of the advice that I had to offer.

All data objects, whether Word files, Excel files, XML, relational databases, text files, HTML files, registry files, LDAPs, Outlook and so on, are really relational databases, spanning the spectrum of metadata complexity. All we need to do is to build on all the ideas of ODBC and JDBC to provide the same logical interface to all of them. If we can do that, then SQL Server can treat them all the same logical way, as linked databases, and extract the metadata, create the entities and relationships in the same way, and use the same SQL syntax to interrogate, create, read, write and update them.

The complexity would be in the implementation, hidden from the poor DBA and developer. He would just know that he could create a Word, Access, XML or Excel file in the same easy way he could create a SQL Server database, and

www.dbooks.org

transfer data between these esoteric media, being obliged to do no more than specify the medium. Now, there's a start …

The rest of my advice got lost in the struggle to regain consciousness. Its loss is not important; as I'd be much more interested to read your suggestions on how SQL Server could be made simpler.

www.dbooks.org

Clinging to the Flotsam. A Survivor's Tale

Still holding his head above water after numerous years being buffeted by the storms of the IT industry, Phil offers some survival tips.

I sharpened my metaphorical pencil and brooded. I decided to write some simple tips on how a software company could survive in a recession, and make best use of it. As always, I sat down in front of a blank screen and stared at it. I waited for the muse to whisper inspiration in my ear.

"Phil," came the old familiar voice of inspiration in my ear. "This subject is a dead cert. Your company was founded in the nineteen seventies, and has been in business ever since. You've done well, generally speaking."

"Don't listen to her," whispered a voice in my other ear. Damn it, I thought, my conscience has woken up. "A lot of your efforts were spent advising other companies how to do it, or in writing software for other people. You even copped out entirely on occasion and went contacting or acting as a consultant. OK, I'll admit that you always did it via your company but it is a cop-out."

"Phil! Your conscience is being too harsh." replied my inspiration. "You have a contribution. After all, very few software companies have even survived that long. All your friends and colleagues have simply vanished into the night, joined the Civil Service, retired, or become wizened and burned-out senior execs in multinationals."

A slightly desperate edge was detectable in my conscience. It could see it was losing the argument. "Look, Phil; these guys might take your advice. Beware. Here they are, putting on their snorkels and goggles and waiting, for the first time, for the coming flood of recession. No bullshit now."

"OK guys," I thought, "I'll try to behave, but because the real survivors such as Bill Gates and …er… Larry Ellison aren't going to blog on this site I'll have to do something. Maybe as I've been clinging to flotsam and holding my breath for so long now, someone might enjoy reading a survivor's tale."

Recessions come fairly regularly, and one adapts. There are actually many tactics that you can use to survive. One tactic is to cut and run. A chum of mine

disappears every time there is a recession since he has an Undertaker's business on the side. He was always a bit quiet about this, but I'd always wondered why all his cars were black Mercedes with huge backs. I'd supposed he had underworld connections, but in fact, whenever the software business went quiet, he'd put his company on ice and he'd busy himself with burying the dead. The next boom time, he'd bob back up, put away his black frock-coat as if nothing had happened, and continue selling software. Another friend of mine developed payroll software when times were good, but after he once fell victim to a recession, he trained as a Butcher. Now, whenever there is a recession, he is out in his van delivering sausages. Some jobs are recession-proof.

My own tactic has been different. I've always worked rather like a commercial plantsman. Recessions are like the seasons. Whenever the phone stops ringing, and the icy blast of economic slowdown happens, I prepare frantically for the next summer. Each software boom tends to be different from the last, and the plants put out as soon as possible in the early spring and cherished under glass are the ones that yield the most.

Recessions are really the only time to develop software. The fact that the phone has stopped ringing means one less distraction as one works frantically to ensure that, when the next boom happens, you have software products finished, tested, and ready to sell. If a plantsman spent his winter idling or moping about the lack of sun, rather than rushing around planting seeds, taking cuttings, splitting crowns, dividing roots, and so on, his business wouldn't survive. The plants he sells in the summer were started off in the harshest part of the winter when the coming of the heat of summer seems to be impossibility.

In the freemasonry of the profession of software developers, I hear stories of new exciting startups inspired by the recession. I hear of friends who have dropped everything to build new applications. The other day I called in to see a friend who has a new startup company developing a 'cloud-based' business suite in Adobe Air and Flex. We marvelled at the application. "A perfect time to develop radical new software!" he said, chuckling.

More or less; if only the economy had the regularity of the seasons. A couple of times I've been caught out being on the point of publishing an application that has taken a year to develop only to find that the buyer has disappeared, as a victim of the tide of recession. One has to adapt, and find new customers. One day, I shall tell the story of the application that started out as Multimedia encyclopedia of Opera, morphed into a CDROM of naked ladies, and finally underpinned the customer-facing application of a major retail bank. Such is the real world of software. One adapts, and rides with the blows. The important thing is to realize that the world economies are subject to wild

www.dbooks.org

variations in growth and recession, and your business plan must always acknowledge that fact.

Section III: Software Projects: the Good, the Bad and the Pitiful

Throughout his career, Phil has encountered many reasons why software projects struggle, drag on interminably, or just fail horribly. And funnily enough, none of them was due to the team failing to use XP or ICONIX or whatever the latest hot new development methodology happened to be at that time.

The main culprits, in no particular order of merit, are human stupidity, reliance on overly complex and unproven technology, unfathomably bizarre and outdated processes and procedures (stand up almost every Government-funded project he ever worked on), lack of long-term planning, and plain old, simple lack of communication.

This section starts with a dissection of some of the common sense principles and practices that experience dictates should be applied to every software project. This is followed by a few tales of projects that have trampled over some or all of these principles, and their consequences.

I think Phil's aim is purely to help people avoid some of these mistakes in future. As he freely confesses, he has, over the years, either 'watched or participated in almost every mistake that can be made in IT', so is in no position to preach. [TD]

www.dbooks.org

Secrets of successful IT projects

Over the years, Phil has either watched or participated in almost every mistake that can be made in IT. He's come to realize that if you boldly go where no man has gone before, you will probably just fall flat on your face.

"The project was two years late and three years in development," he said. "We had 30 people on the project. We delivered an application the user didn't need. They had stopped selling the product over a year before." – Sid, quoted in the 1994 Standish Chaos report.

Some time ago, I worked on a team that was attempting to define an IT strategy for a large multinational company. Management assumed, incorrectly, that the company was unique in that it had a high number of failed IT projects. In fact, only one in four of all IT projects can be termed successful, if the benchmarks of success are adherence to a scheduled completion time and budget, and realization of the project goal, be that saving money, growing the business, or what have you. I have to explain this because many of you may never have come across a successful IT project.

We spent a long time looking at current and recently completed projects within the company's facilities. As the company had shied away from a single corporate strategy, I had an education on the different ways an IT project could be undertaken. One common thread was that, despite their innovative designs and use of pioneering technologies, projects were almost invariably late and over budget.

Later I got wind of a group of projects that had bucked the trend. They weren't flashy but they did what they promised, met all the criteria for success, and the company's business users universally loved them. All of these projects had emerged from the same 'out-of-the-way' IT department.

When I got inside the building and was being shown around this particular department, I noticed something curious. The programmers were slightly plumper than usual, and had a tendency to baldness. Many wore tweed jackets, even the ladies. A software house ran the department under contract. I eventually discovered, to my astonishment, that the staff was programming in

COBOL with a Sybase back end, and using a fancy windowing interface to disguise the fact!

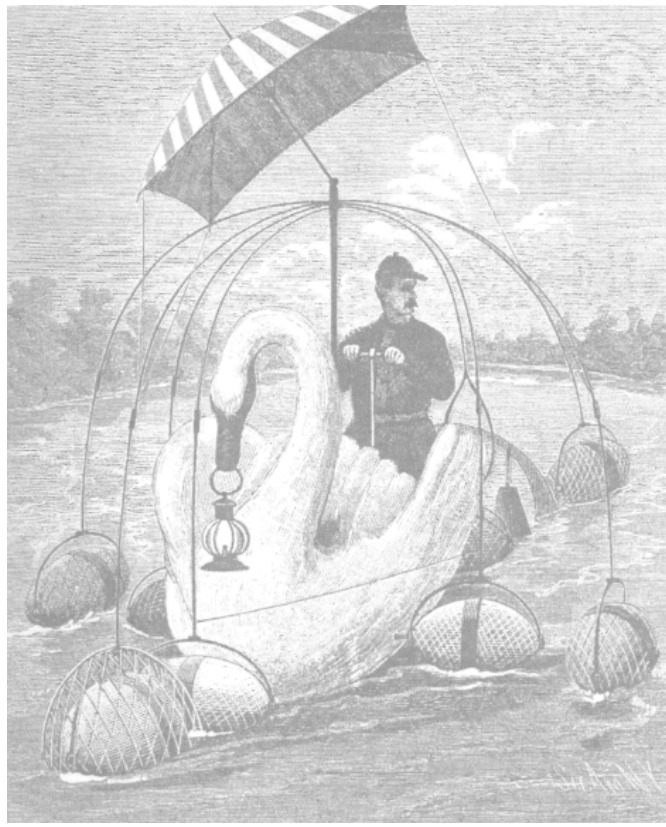
They implored me not to give up their secret, and I obliged. I can happily report that when I left the company some years later, they continued to churn out successful applications with almost no visibility to senior management. For all I know, they are still there.

I discovered during that exercise that there are a number of reasons given for the failure of IT projects, most of which are bogus and designed to distract attention from the real causes. In contrast, there are just a few reasons why they succeed. The experience taught me that, instead of attempting to improve our success rate by identifying what had gone wrong with certain projects, we should look at how and why other projects succeeded.

Arguably, COBOL should not play a real part in a modern IT project, although, like SQL, it is grounded in and inspired by the processes of commerce. There are, however, certain rules for software development that consistently hold true:

- Decide what needs to be done – sit on your hands until everyone buys into what they have to do and how long they have to do it. Then operate rigid change control.
- Don't be a pioneer – insist on, and stick, with technology you know will work.
- Code, then recode – when you've got the code working, throw it away and rewrite it properly.
- Discourage virtuosity – never let a programmer do something you cannot understand.
- Encourage results – reward work that meets scheduled completion dates and specifications. Politely ignore excuses as to why things that should have happened didn't.
- **Don't overload the lifeboat**

www.dbooks.org



Don't be a pioneer

Let's take a look at each in turn.

Define your scope and stick to it

Scope-creep is a euphemism for the stacking of requirements on a development project that looks like a runner in the corporate environment. Until a project gets sign off, it is generally ignored within the enterprise, but afterwards it becomes the focus of frenzied attention. From a management viewpoint, it could be the company's last chance to achieve business efficiencies through technology.

From the IT management perspective, resisting the pressure to turn a molehill into a mountain is difficult. Imagine yourself being in charge of a lifeboat, beating away the drowning with an oar to prevent the boat from sinking. When you try to argue that the addition of functionality cannot be achieved without resources, you have a similar reaction to deliberately bashing someone's fingers with an oar. The rejection of new changes into your project

plan is necessary, however, since an overloaded, constantly changing, project will go nowhere.

Don't be fooled by an impressive-looking requirements document from the business side. The true nature or complexity of the company's business processes is not often shared, even when management knows what it is.

I was once involved in a corporate re-engineering exercise that spent roughly $17 million on a requirements-gathering exercise. For months, well-groomed young graduates in dark suits earnestly identified processes and procedures. The resulting report filled a large steel filing cabinet. I was the first to, err, skim through the document, and I believe I was also the last.

I took one department's requirements back to them and challenged the staff to admit, without blushing, that the document reflected reality. After a bit of blustering, they confessed that their contributions were mostly fabricated to make their role in the project - and the company - seem more important. For them, re-engineering was nothing more than redundancy.

Determining what the IT development staff should do is the hardest part of a project, and no code should be cut until that job is done.

Use boring, proven technology

I always hear gasps of shock from J2EE enthusiasts when I mention this rule, as if using a proven technology is somehow radical and dangerous. Time and again I've seen IT departments fall prey to the idea that 'perhaps, this time, if we try something new, things will be better'. It is also comforting to blame old technology for the failure of IT systems.

Beware of misguided attempts by business management or the project sponsor to decide on the technology before the project starts. This is like choosing the wedding dress without the benefit of a suitor. Invariably, this happens when management is successfully seduced by the technology supplier's sales claims.

If the customer is the enemy without when defending the due process of selecting technology, the development team is the enemy within, because developers have their own agenda.

Many developers have mixed feelings about a conservative choice in technology. The industry values programmers for their current skills, so their prime objective is to hone those skills to increase their market worth. If the project they are working on doesn't require a so-called hot skill, they will use their spare energy to make sure it does.

www.dbooks.org

It may not be logic that fires the hunger for the latest technology. One of the strange side effects of testosterone is an irresistible urge to play with the latest technology, and it is hard to distinguish real productivity from unashamed self-stimulation.

It takes courage and willpower to adopt a relatively conservative approach to development work and select the technology appropriate for the application. When I plan a project, I generally construct a 'playpen' area in which all the latest technologies are used to develop a non-critical project component, and then let each team take turns developing it.

The team gets to put all sorts of skills and technologies on their CVs, and the pressure is off to prematurely adopt a fancy technology for the project's serious deliverables. We can then inform the project sponsor that we are using glitzy technology just like in the advertisements, and everyone is happy.

Code, then recode

Throwing away code, just when you've gotten it to work, may seem unkind or unnecessary but is actually a deeply cleansing experience, like sloughing off dead skin. I discovered this by accident when the bad luck fairy struck and no one had made backups.

Initial feelings of despair were followed by a curious lightness of spirit, as all the false turnings and cul-de-sacs of everyday coding were forgotten. We knew what needed to be built, having done it once, stumbling in the half-light of the systems analysis.

When we keyed it all in again, it was half its previous length and ran twice as fast. It didn't take long to do it either, and we hummed as we worked. I've since discovered that the same applies to any creative work. Tentative work should always go on the fire.

Discourage virtuosity

The principle that you should never let a programmer do anything you cannot understand is a classic one, first articulated by C.A.R. Hoare, inventor of the Quicksort algorithm. It is ignored at your peril.

My worst experience with this behavior was when I was supervising a freelance Sybase programmer who created a reporting system for a financial services company. He used dynamically compiled stored procedures that were created in response to the exact slice-and-dice query required by the financial analyst.

He almost got the code working to the satisfaction of the business, and then demanded a doubling of his contract rate. We parted on bad terms, and I was

left with one of the scariest and most opaque bits of application code I had ever come across in my waking hours. To make matters worse, the business sponsors were baying for impossible improvements, bug fixes and extensions to the scope of the project.

Reward results

The idea of rewarding a programmer's productivity may seem unfair and unreasonable in a dignified profession. On the contrary, it is as difficult for programmers to gauge their own productivity as it is for them to test their code, and a monetary reward sends a clear signal of a job well done.

There is always one programmer on a team who works hard and does heroic coding, but achieves very little. The objective of producing an application that satisfies the business requirement is lost in the abstractions and obfuscations of the code.

It is as if they get sucked into a Tolkien-esque universe. There have been times when I've half expected to go to work and see nothing but a pair of feet sticking out of the screen. Now that physical violence against programmers is frowned upon, one way to jolt them back to reality is to be selective but generous in the distribution of bonuses.

Been there, done that

Unless it is thought that I speak with the sanctimony of virtue, I must confess that over the years I have either watched or participated in almost every mistake that can be made in IT. One of my worst experiences occurred when the company I worked for got a contract with a government department embedded deep within what was then called the Ministry of Agriculture.

Our sales rep got overexcited after a meeting with government officials and priced out the job as a trivial application involving the automated sorting and grading of vegetables with its associated simple accounting. Instead, it turned into a monster application that was to be released nationwide as a major government initiative.

The scope of the project grew faster than the development work could be completed. Interfaces and dependencies grew like field mushrooms. To make matters worse, the hardware, selected by government dictat, was unreliable, and the macros used for process control were almost impossible to debug. The software, a version of BASIC, was proprietary to the hardware and innocent of database functionality.

The day of the demonstration loomed. The government mandarins and representatives of the farming community were soon scheduled to come and admire the efficiency with which the application could read barcodes; grade,

www.dbooks.org

weigh and sort produce; feed the data into the computers; and produce some impressive paperwork. We were up to our armpits in unfinished code and non-functional interfaces. Then we were summoned to an important meeting.

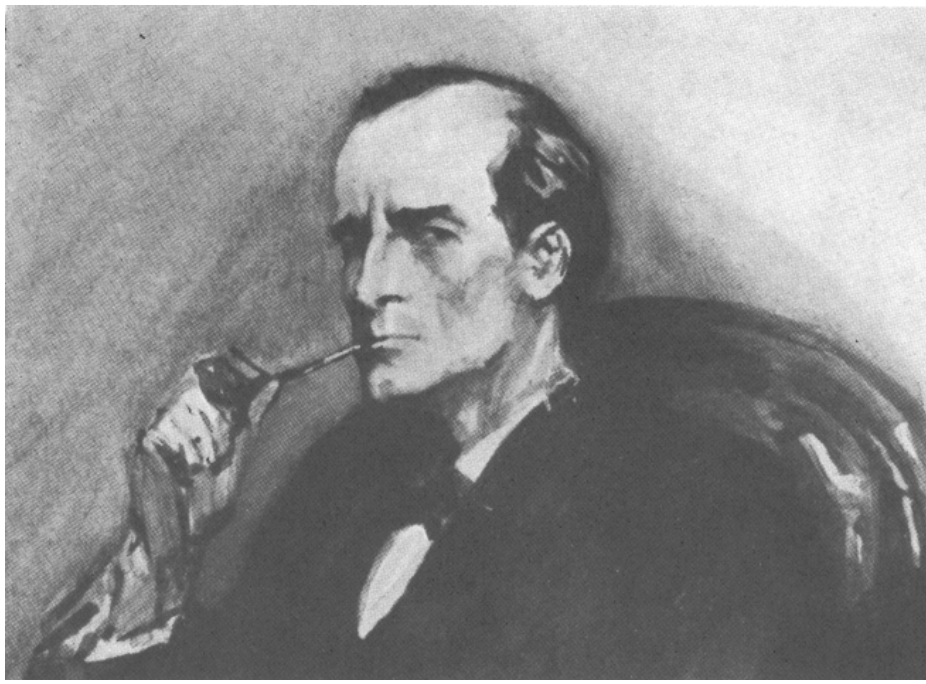
With trepidation we put on our best suits, as if going to our own execution. To our surprise, the ministry representatives were extraordinarily apologetic. They broke the news to us that their department was to be closed down in a Government reorganization and that the project would be canceled.

They mistook my flushed face with crazy and wet eyes for signs of sadness. "Don't worry, Phil," I was told. "You'll be paid in full, so you won't suffer too much."

Perhaps there is a patron saint for software development after all.

www.dbooks.org

Confessions of an IT Strategist



IT Strategy is held in the iron fist of an incompetent 'Kremlin' and Lobster Lunches are the only way to get things done. Anarchy ensues.

In the new Simple-Talk spirit of 'coming out', I have to confess I used to be a practicing IT strategist. However, I gave it up and went back to being a database developer after a number of almost identical conversations with IT Agencies.

"OK, you've sent me your CV, how would you sum up what skills you possess?"

"Ah well, I'm an IT Strategist."

… (embarrassed pause) …

"Yes, great, but what skills do you have?"

"Well, IT strategist is one of them."

… (nervous laugh) …

"Come on, no kidding please. What sort of marketable skills do you have, like VB, Sybase, Oracle, .NET?"

"Oh hell, I give in. You win. Yes, SQL Server, C#, etc … etc …"

Why should any IT department bother with an IT strategy when there is Microsoft, which one can follow like a sheep after a bell? As a constant reminder of the answer to this I keep a complete set of Xenix, created by Microsoft as a successor to MSDOS, on the bookshelf. I used to have a copy of Microsoft's OS/2 as well, but it irritated me so much I threw it out. The message is simple: Microsoft gets it wrong sometimes. Others sometimes get it right.

The trouble is that long-term planning is so neglected that most organisations are unable to think far enough ahead even to employ a strategist to do the work. Please do not confuse a strategist with a project manager. A project manager cannot care what happens to the rest of the universe as long as his particular project comes on in time and under budget. A strategist has a broader perspective, and tries to see further.

Nowadays, it is rare to come across a well-thought-out long-term strategy plan in IT departments. A lot of the work is now either neglected entirely, or done by a hybrid race usually known as 'Technical Architects'. It is a very cosy idea that, having chosen the general direction of one's IT developments (now either J2EE or .NET, it seems), it is all subsequently a fairground ride with no steering wheel, often with everyone screaming.

A while back, I was asked to investigate the IT problems in a large national organisation whose identity I am not allowed to divulge as I signed a piece of paper saying I couldn't tell. They should have made me sign one that said I couldn't laugh either. It was irresistible.

At one time, this organisation had a strong, centralised IT department, nicknamed 'The Kremlin', that ruled with a rod of iron. It carefully maintained an IT strategy, which plotted the narrow path between the icebergs of the IT marketplace.

Unfortunately, the gifted intelligent people who used to run it had long moved elsewhere, leaving a runt of 'jobsworths' and knaves. In their hands the careful IT strategy had subsided into an ossified orthodoxy, which could be changed only by the technique known as the 'Lobster Lunch'. This involved the

www.dbooks.org

salesmen of various IT companies vying with each other to take the Kremlin IT executives to the best lobster restaurant on the Maine Coast. I've always worked on the premise that, if I can eat it, or drink it, in one sitting, then it isn't a bribe, but 'Lobster Lunch' soon became a code word for all sorts of other, more criminal, inducements.

The inevitable then happened: the Kremlin was overturned. This event was greeted with cheers from almost everybody except the rather plump IT executives. The various regional groups within this organisation started to create their own IT systems, in much the spirit of the French revolutionaries. With cheeks aglow, off they went to PC world to purchase whatever development system, 4GL, or database caught their eye.

At last, the 'Kremlin' was gone, and they could build the systems they wanted! And so they did. Powered by a sort of frantic euphoria, each region raced to produce its own, often very expensive, systems.

The problem was that, at the point I was called in, the organisation was being reorganised. It was no longer to be regionally based. The IT applications now had to synchronise and inter-communicate. Unfortunately, by the time I came on the scene, there were seventeen different systems, all of which performed exactly the same role, all written in different ways in different applications. None of them could even share data.

Calling in a strategist at this stage was like calling in the fire-prevention officer when the building was a smoking ruin.

However, the design of a new architecture was a pleasant and rewarding task. Due to the fact I had not signed a 'no laughter' clause, it was a happy time too. We had to reverse-engineer some of the systems to soothe ruffled feathers, and find out what they did. In the rush to get the systems launched, some managers had unfortunately employed their teenage sons to do the work in the thoroughly mistaken idea that they were computer geniuses. There were some database howlers that, had they been furry creatures, I'd have had stuffed.

Even when professional organisations were commissioned to do the work, there were disasters due to the lack of any strategic thinking. The funniest one was the elaborate Sybase database system, commissioned at great expense from a highly esteemed software house, which I would dearly love to name and shame. It recorded the entire years trading in every detail. It was installed, and eventually worked fine.



This must have been a fine building once ...

At the end of the first year, it stopped working. Everybody scratched their heads wondering what had happened until they suddenly realised it was written only to do one year. Simple, they concluded; all one needed to do was to install a new system every year. By the time I got there, five identical databases existed on the server, one for each year, and any serious reporting was a nightmare.

To cut a long story short, I had a happy and remunerative time mopping up, to the point in time that the reconstruction could be handed back to the permanent staff. I popped back a year later to see how things were going. The Kremlin was once more in place, staffed, I hoped, with earnest, conscientious types. It was all very cordial and I felt that we'd set the organisation to rights.

In the pub after the meeting I nearly coughed on my beer when I heard that the organisation was going back to a regional structure. I hope the 'Kremlin' stood firm this time.

www.dbooks.org

The Technically Minded Subclass and the Fog of Misperception

Most IT projects are shrouded in a 'fog of misperception'. They fail time and again because nobody actually bothers to talk to the people who understand the business processes and the technical details of the product.

I spent several years in a team that advised a large international manufacturing company on their software-purchasing strategy. It always amazed me how far the software companies misunderstood our core requirements, but never really took the time to find out what they really were. All their information about us was gleaned from their own non-technical sales force, whose scientific knowledge stopped short of ruling out the existence of fairies. And even these guys talked only to our senior management who understood nothing of the details of what went on in their empires, but were happy to bluff their way through in return for a hearty lobster lunch.

The technically minded subclass who actually *understood* the business processes and the technical details of product development were treated with general contempt. They were usually kept concealed from visitors due to their bad-tempered candour and their propensity for blurting out embarrassing details of management mistakes.

One hardware and software supplier, in particular, based their continued prosperity on supplying the company. We were visiting their development Labs one day when they excitedly guided us into a large room where a number of developers were working on a wonderful and exciting top-secret project. They'd been at it for two years. Basically, it was a system for modelling complex castings on the computer. It put all sorts of stresses on them to see how the structure would flex, based on finite element analysis. For two hours, expert after expert presented aspects of the application. It would, they told us, save huge sums of money in building prototypes, and test equipment. It would revolutionise our manufacturing processes. At the end, all eyes were on us and an expectant hush descended as they awaited our response.

I said that any company who purchased the system would have a handy tool, but that it would be no use to us. There was an amazed silence. I asked, in some alarm, who on earth had they had in mind as customers for the product? 'Why you, of course', they replied pathetically. We then had to explain that we already had a CAE system that was far more sophisticated than the one that they'd shown us. Our one was able to take development of components from concept through to development, prototyping, test, costing, release and manufacture, all from the same basic wire-frame data model. It was designed to allow multidisciplinary teams in different organisations and countries to work together on complex design processes. Their application could never be made to fit into the way we did business.

There were emotional scenes. They'd spent two years, and enormous expense (investment they called it, somewhat optimistically), doing this work with only us as customers. They had created purpose-built graphical workstations. They had relied only on the occasional remark by senior management of our company, but had never thought to penetrate through to the people who were actually doing the work. We left rather hurriedly, and they were immediately on the phone to those managers in our company who had lulled them into proceeding with the project. Their initial ploy in maintaining that we were mere running dogs, incapable of understanding corporate strategy, didn't wash. In the end, everyone had to admit that this was another project that would bite the dust.



There were emotional scenes

This story is re-enacted over and over again, with different details, in government and industry throughout the world. Almost without exception, the same mistake is made every time. It is not entirely the case that a development team doesn't do their research first, although cutting code is much more fun than understanding in detail what the application needs to achieve. More

www.dbooks.org

commonly, they do not have the experience to identify where the real knowledge about business requirements and processes lies.

In other words, they never penetrate the fog of misperception that exists in every large company. At an intellectual level, it is obvious that IT initiatives have to be fit for purpose, but emotionally, there is always the siren voice saying 'This time it will be different, we can evolve prototypes and elicit the full requirements along the way; we can re-engineer business processes, rather than to fit with the existing ones.' It never is different. Sadly, history just goes on repeating itself.

www.dbooks.org

The Data Dialog

Is your company ready for the truth on why most data systems take so long to implement, go so far over budget, and end up failing anyway? If not, best to misdirect them with IDC (Initials Designed to Confuse) and ADH (Acronyms Designed to Humiliate).

Boss: Phil, this strategy paper that you've written on our corporate data integration policy. Look, I'm the IT director and even for me it's pretty opaque. Damned difficult to read, in fact!

Phil [*blushing modestly*]: Why, thank you. I do my best.

Boss: But damn it, Phil, we can't present this to the board for signoff! What is it actually trying to say?

Phil: Well, in essence, it says that our new data systems will cost twice as much as we predicted, and take twice as long to implement. And, by the way, our last project that promised to take an integrated approach to the problem of corporate data management crashed and burned without trace.

Boss: What!? I never read about any of that!

Phil: [*pointing*]: It's right there on page 3, tucked into that long paragraph near the end …

Boss: Damn it again, Phil. How often do I have to tell you that we senior staff only like to read the first page of these documents?

Phil: Precisely! We just need to make sure they can't claim that they were never informed about the outcome of our IT initiatives. We don't want them to actually read it … that would be disastrous.

Boss: … and what about all these terms in the document … MDM, BI,

CDI, PIM, ERP, ETL, EAI, EII, EIS, ST, SOA and SDM?

Phil: They are all examples of **IDC** (**I**nitials **D**esigned to **C**onfuse), also known as **ADH** (**A**cronyms **D**esigned to **H**umiliate). They are very useful ammunition against the Geeks when they try the same tactics with their .NET and J2EE jargon but, as is the case here, they are mainly used to browbeat Business managers with our knowledge and intellect.

Many IT initiatives are so complex as to be mystical quasi-religions. They are difficult to criticize because their proponents can always sorrowfully shake their heads and protest that you simply don't understand enough about them.

Boss: [*sighing wearily*]: What about this part here … the analysis of the current system. What does this mean? [*pointing*]

Phil: [*peering over his shoulder*]: *'Diverse system architectures*' means that the bloody computer systems can't communicate; '*diverse business functions*' means that even the business can't communicate; a '*meta-data abstraction layer*' is the database equivalent of Harvester Tape. Referring to a '*consistent information framework*' means that at least the column names in the databases are the same and mean the same thing. Where I refer to people as '*stakeholders*', it means I don't know what the heck they contribute to the business, or can't remember the right word for it. If an IT system '*gives micro and macro business process capabilities*', it means that it 'does stuff.'

Boss: You are going at it pretty strong on the past failings and mistakes of IT! Aren't you overdoing it?

Phil: Curiously, any business is only too ready to believe that IT makes mistakes; I sprinkle in these confessions to give the whole document verisimilitude.

Boss: OK but why then leave out the two biggest mistakes of all … the *Business Re-Engineering* project and the *Data-Warehousing* project?

Phil: Because the business management still remember how much they paid IT to implement them. It may have a negative impact if we confess they were blind alleys. Also, if we mention those, they may also spot

www.dbooks.org

that this new MDM (Master Data Management) strategy is just a re-arrangement of the furniture to sell the same old apartment.

Boss: So what is the purpose of this document? It doesn't provide information.

Phil: [*shocked*]: Thank goodness! IT business documents aren't there to provide 'information'! They are there to provide cover and concealment for our real activities.

Boss: Come on Phil, don't be so silly, and just rephrase the entire document so that ordinary mortals like me can understand it. Tell me what it means as though I was a chum at the golf club.

Phil: [*apprehensively*]: Are you absolutely sure you want that?

Boss: Of course, get on with it man!

Phil: [*gulping nervously*]:

OK …

Once upon a time, IT departments used to appoint someone to keep tabs on what data was held by the company, what the business understood about that data and the sort of processes that happened to that data. This person, usually a grizzled *Systems Analyst* (or *Data Architect*), would ensure that there was a common understanding of all the real entities involved, such as customers, products, employees, and the processes that acted upon them, such as releases, recalls, invoices, and so on. He would maintain a document, usually called a *Data Dictionary* or *Data Model*, which could be read and understood by anyone, and provided a basis for each individual IT project. Any misunderstandings were soon ironed out, and the activity was an essential part of strategic planning.

When the cyclical downturns in IT happened, it became tempting to get rid of this chore. The Data Architect, or whatever he was called, was packed off with an ornamental mantel clock under his arm, and there seemed to be no dire consequences. It all seemed painless. The integrity, and shared understanding, of Corporate data didn't seem to be an important asset after all.

However, with mergers, acquisitions and restructurings, many enterprises, including our own, have struggled to keep a consistent

culture and nomenclature. The first sign of the disease is the increasing number of anomalies that seem to slip in to the company reports. Often, the root cause of such anomalies is laughably crude, such as the case when one division of a major bank counted a customer as a person who had one or more accounts, whereas another counted a customer as an account-holder.

The problems quickly became endemic and the only solution was to perform a *major business-re-engineering.* This suited the hotheads, but led to several high-profile disasters. The idea of 'Data Warehousing' was an attractive panacea, as it meant that nobody had to confront the essential problem of 'data-confusion' within the business. One could fiddle with the messy data and pummel it into a logical state, from which one could gather all that vital business information and reporting. Most bought into the dream.

Many a silver-haired ex-Data Architect will have smiled as he sat in his deck-chair, reading about the vast cost and wasted effort involved in implementing a dream that couldn't live up to its promise. If data is fundamentally unsound, no manipulation can help it. As they say in Essex, 'you can't turn turds into plum pudding'.

With this company's adoption of distributed architectures and complex messaging, the opportunities for confusion have increased. Whereas, one once had just to keep a consensus on what business entities comprised at the data layer, now one has to browbeat anyone who has the wit to nail an XML message together.

So we now come to the point where what this paper is suggesting is a type of *Master Data Management* initiative. We once more realise the importance of understanding the 'corporate metadata', the consensus on how the corporate entities are understood and accounted, and the nature of the processes that act on them. We understand a service-based approach to data and we now have the technology to integrate this model into our IT systems. And, ironically, it is only now that we have come to realize the importance of that ancient Systems analyst poring over his obscure diagrams of boxes and arrows. His prolonged absence is the reason why this new data system will now cost so much, and take so long to implement. And we still might not get it right …

Boss: Hmm. On second thoughts, I doubt if the business is really ready for the truth. Perhaps we'll go back to your original three-pager.

Phil: [*relieved*] I think it will have the effect that we want.

www.dbooks.org

The Joy of NAD

Phil Factor convenes a short meeting to discuss the data architecture for storing a client's contact details ... how painful could it be? He discovers that even the simplest database can be a major cause of grief and complexity.

The Scene: A meeting room in the IT department of a large company in London. The meeting is in its third hour and the participants lie slumped in their chairs, like discarded puppets. The walls are covered with crumpled ER diagrams full of boxes and crow's-foot arrows. An air of doom permeates.



Phil Factor

(*fending off black despair*) Right. It looks as if we can't agree on the data architecture for the trickier parts of the new application, such as the back office functionality. Let's take a step back and do something

easy. What about looking at the NAD side of the application (Ed: *NAD=Name and Address*). This'll be nice and simple, hopefully. *(Thinks: if I can catch the early train, there's that nice pub in Aldeburgh ....)*

(Developers look up from their notebooks and stare bleakly at Phil)

Phil

(*with strained enthusiasm*) Well, how about a nice simple table with the customer's name, phone number and contact address in it?

Dan the Data Man:

Wait a second, I can see problems already. People can run accounts under different names. Single customers will have many names.

Developer:

Surely, we can just tell them they can't have more than one name?

Phil Factor:

(Thinks: ... and people wonder why we are reluctant to let the developers out of their cubicles)

Dan the Data Man:

(*patiently*) It is perfectly reasonable for our customers to have accounts under several different names. Disallowing this might even be seen as discriminatory but in any event it would certainly be inefficient. If we treated them all as different customers we'd end up phoning or visiting them several times in a row. It happens occasionally and we know that our customers don't like it. And besides, this is real life – a married lady can use her own surname or her husbands, and even call herself something like 'Mrs Phil Factor'.

Developer:

(attempting humour) But that could also mean that Phil has had a sex change?

Phil Factor:

I say, dash it chaps!

Dan the Data Man:

Ah, quite. You've identified another issue. Oh dear, the complications whenever that happens. Sex changes are date-related so

www.dbooks.org

that if we print out transactions before the sex change they'd have to be Mr Phil Factor, and after the date they'd be Ms Philippa Factor.

Developer:

OK. So we'll have to put the names in a different table and have a many-to-many relationship with the Customer table

Alan Analyst:

The application developers will never understand the SQL for the CRUD (Ed: Create, Read, Update and Delete). Their embedded code will cause havoc and the DBAs will demand an interface based on stored procedures.

Phil:

(*to himself*) Yes!!! Result!!!

Dan the Data Man:

You want to store contact addresses too? (*Sucks through teeth and shakes his head sorrowfully*).We'd need to record all of the addresses associated with each customer. What about billing addresses, Bankcard addresses, work addresses and so on?

Dan the Data Man:

Are we recording the customers' contact numbers as well?

Phil Factor:

Of course. Sales have put in the requirement and it would seem a good idea to recognise the CLI (Ed: *Caller's phone number*) when a customer phones into the telephone banking system …

Dan the Data Man:

A customer will have a home phone-number, a work number, a mobile number and so on, some of which will be current and others which will have expired at a certain date. This means that we'll have to record the type and termination date of each entry.

Phil Factor:

(*with a pathetic attempt at bringing the meeting to a close before his train goes*) Well, is there any way we can let the code-jockeys create their own Customer object and just 'persist it' (Ed: *store it*) as XML in the database?

(Various developers bob up and down in their seats with excitement)

Dan the Data Man:

The last time we did that, it was pandemonium. We ended up with six mutually exclusive definitions of what a customer was and none of them recognisable by the business.

Derek DBA:

(between clenched teeth) … and no ordinary mortal could maintain the production system when it finally emerged.

(Developers relapse into surly quiescence)

Developer:

Yes … so we'll have to put the contact numbers in a different table and have a many-to-many relationship with the Customer table. The linking table will need the date stamp and the type of contact number *(thinks: is this Groundhog Day?)*

Alan Analyst:

Well, also, marketing want a whole lot of other customer details stored, so they can work out the demographic profile of purchasers. They're dead keen but rather unclear as to what these details will be.

All:

'One True Lookup Table'!

Phil Factor:

OK! OK! So someone think of a better way of making changes to an entity on a production server without needing to do any change management procedures? We're going to need an EAV (Ed: *Entity Attribute Value*) table just to bridge the gap between marketing and reality, by allowing 'soft attributes'.

Dan the Data Man:

We generally find that marketing have very short memories anyway, so we can time-bomb the additional 'soft' attributes. We call it the 'Goldfish' mechanism.

www.dbooks.org

Phil Factor:

Excellent! We seem to have a broad agreement on the general principles of the design. There's no sense in getting into to much detail at this stage. *(Thinks, if I run part of the way to Liverpool Street station, I'll get that train)*

Colin Compliance:

(clears throat) I'm a bit concerned about the auditability of this NAD.

Phil Factor:

(Thinks: Two million years of human evolution just to get a Compliance officer! Nature can be cruel. It doesn't seem right.)

Colin Compliance:

This design isn't going to get sign-off by the audit-compliance team. Do you realise that we have to be able to reproduce invoices, advice notes, returns and such paperwork exactly as first done? We have to audit when an address changes. Any change of name has to be logged with date, time, the database user and workstation. There have been many frauds in the past that have involved changing a customer's address, so the audit-compliance team will have a feeding-frenzy on the design as it is so far.

Phil Factor:

(thinks: OK, next train in an hour's time. Might as well spread the anguish) Good thinking, Colin. So that means we'll definitely need to have all CRUD done via stored procedures, with a separate log for all changes.

(Developers slump, despondent)

Colin Compliance:

(warming to the subject) Yes, the design will have to have a full audit trail. We will need to make all tables time-based so that all versions are kept, with an insertion-Date and termination-date, and the SQL picks out the current one only due to its null termination-date. We'll need a trigger to enforce the rule of having only one record with a null termination-date where one would otherwise have a unique constraint. We can then cross-check it with the separate audit log.

Phil Factor:

(recovering from a near-death experience) Quite so, some interesting ideas there.

Derek DBA:

Someone is going to ask for an independent Data Protection audit report on how we conform to the Data Protection Act. Marketing always sail close to the wind. Maybe we'll need to encrypt some fields, over and above the usual security precautions.

Dan the Data Man:

Security issues aside, we'll probably need to allow customers to check all data we hold on them for accuracy. Has anyone studied the Freedom of Information Act and the Human Rights Act for compliance issues?

Colin Compliance:

Yes, and we haven't even started discussing Sarbanes-Oxley, FSA, HIPAA, HSPD-12, BASEL II, SEC, FSA, FASB (Financial Accounting Standards Board) and Gramm-Leach-Bliley …

Phil Factor:

(Thinks: I have an hour to kill, a concealed iPod and a thirst for revenge) Yes, I think we should bottom out all these fascinating compliance issues first and then move on to identify any potential performance concerns. But regrettably we must be brief, as the room is only booked for the next fifty minutes.

www.dbooks.org

Smoke and Mirrors

It seems hard to imagine that a presentation on a new IT application to the directors can go too well. Phil doesn't need to imagine; it happened to him. He winces at the memory.

The directors of the Imperial Bank came into the room, taciturn and solemn, each of them accompanied by a young, brisk, charcoal-suited 'minder', a manager in gold-rimmed specs and neat hair. It immediately reminded me of the occasion, many years before, when I was persuaded to sing some songs, accompanied by a banjo, for the local Old Peoples Home.

No. These august gentlemen gave the impression that they ran more to dignity than brainpower. There is a popular myth, amongst those who have never worked in the corporate setting, that the higher up a company hierarchy one goes, the more clever and cultured the managers become, until at the very top one gets to meet a higher entity rather like the sentient beings from another planet one finds in fifties B Movies ('My name? It is not important', they invariably say when asked). As I had an uncle who ran one of Britain's largest and most profitable companies for a while, I had no such illusions. He was a jolly chap, the perfect company for a day out at the races, and one of my dearest relatives, but an intellectual lightweight.

"Right," I started, beaming at the assembled dignitaries, "I shall take you through the features of the new application." This was a 'Smoke and Mirrors' presentation, a preview of how a new application was going to look. It was a crucial new initiative for the great bank, to provide telephone banking. I'd been appointed to design the application that would support the operation.

'The customer phones in …"

I clicked on a button on the screen and a little animated phone bounced up and down slightly, and the computer gave a ring.

The minders frowned, shocked by the frivolity. The directors were suddenly fascinated. I paused to let the effect sink in.

"Ee' Lad, answer it. Don't keep him waiting."

"No, actually it isn't a real customer; I'm just showing how the system will work."

He blinked in surprise.

"… so now I answer the call, and look, we've already recognised the CLI of the caller and got all his banking details and we've even taken a guess at what he's likely to want, based on his previous phone calls."

The screen showed an impressive form filled with lists, grids, images, and directories. Oh God, how I love designing these demonstration systems.

They goggled in wonder. "I knew it really was a customer!"

I ignored him.

"… and now we find out what the customer wants. If we're stuck for words, a prompts appears on screen …" They stared, as if hypnotised. They read the prompt, their lips moving slowly. They were hooked.

I blossom in front of an appreciative audience. I went through the carefully scripted presentation. The 'Smoke and Mirrors' demonstration had enough functionality to take one through a couple of typical scenarios. As I navigated through the banking procedures, the directors occasionally whistled in appreciation. I warmed to the task, like Widow Twankie before a lively matinee audience.

At the end, I paused for questions. The minders looked glum, since they felt that their charges had somehow slipped the leash. The directors twittered amongst themselves like excited schoolchildren on an outing. "Young man," said one of them importantly. "I'll be frank; we like what you've done. No one can accuse us of being indecisive. We'll take it as it is. No point in shilly-shallying around. I'm a plain man and speak as I find."

"Excuse me," I said, suddenly panicked "It isn't a completed system, just a demonstration of what the system will look like!"

"Aye, you'll need to take them bugbears out lad." Said one proudly, airing his IT knowledge; all his IT knowledge.

"We'll also have to write it. What you saw was just a simulation of the planned application. There are months of work ahead."

They smiled like children being shown a card-trick. "That's as maybe" one said, winking, "and you'll no doubt be charging for consultancy too!" He nodded knowingly at his fellow directors.

These guys could have done the voice-overs for a Wallace and Grommit film.

www.dbooks.org

I explained at considerable length the process of building an application. First, we created a series of simulations from which we could get all sorts of data including usability; then we hardened up the look-and-feel until it reached the point that we could get the training and documentation started. Only at that point did we build a production quality system. (This was before 'Agile' was re-invented.) They seemed to take it in, and nodded wisely, but I thought I could still detect an amused twinkle in their eyes, as if they were 'on to my game' and were just playing along.

It was all decided. We would build the system. When we were three months from release, we would invest in the expensive parts of the project such as buildings, training and staff. The minders gravely took notes with gold biros.

A week later, I discovered that the directors of the bank had spent over ten million pounds buying a building to house the telephone banking operation. Recruitment and training were already swinging into action. My head sunk to my desk and I groaned pitifully. Clearly, the central message of my demonstration hadn't *quite* got across. An email message dropped into my inbox. The IT director wanted to see me urgently. The next few months were not easy.

www.dbooks.org

The Acronym Playpen

A rose by any other name smells even sweeter when it is called a buzzword. Most programmers find hard to resist the lure of the latest, greatest ADH technology, but it's the dull, tried-and-tested technologies that work best for many projects. Phil's compromise is the acronym playpen.

A major high-street bank once offered me a completely impossible design brief. Since I was being paid by the hour, it was one I gladly accepted. The brief was to design, and help to build, a telephone-based retail banking system for a new call centre. This was to take over many of the functions traditionally carried out at branch level. By the time I'd arrived on the scene, they had already decided that it was to be a client-server application built in Visual Basic for the client side with SQL Server at the back end. They didn't know quite what they wanted but they knew what it was to be built in. This is akin to the wedding-guests deciding on the venue, and hymns, before the bride and groom have met.

A systems analyst was sent to interview the branch staff to find out what constituted the retail banking processes. The Retail Bankers suffered a strange amnesia, and professed ignorance. They understood that replacing the traditional functions of the bank with a call-centre would not be good for their careers. High-Street banks would be 'rationalised'. They wanted no part in it.

Fairly quickly, I realised that I was the technical architect on a doomed project. The scope was undefined, the system analysis hadn't been done, and the interfaces with the existing banking systems hadn't been started. It was little comfort that at least they'd decided on Visual Basic. What made it worse was that the corporate regulations for IT systems insisted on a six-week test cycle for each software release. The Call Centre was to go live in nine months, by which time the staff had to be fully trained on the new system.

The solution to this intractable problem occurred to me fairly soon after trawling through the corporate computer manual. If we could identify all the component banking tasks, we could represent the processes in a high-level (4GL) retail banking 'workflow' language. We would create our own scripting language, which would then be executed by an interpreter written in Visual

Basic. I checked with the high-priests of the IT Department, who confirmed that the actual scripts would not need separate test cycles. It was enough to test the script interpreter. In effect, testing could then be done in parallel with the development of the scripts.

I soon had a development system in place. We checked it out with simple scenarios such as a customer calling to check how much was in her account. I handed it over to the development team who promptly refused to have anything to do with it.

Normally, with developers, one can quickly find out what is irking them. In this case, there was diffuse moodiness and petulance. They hated the code I'd developed, hated the project, and hated the solution I'd thought up for survival; but they couldn't articulate an alternative.

After a few days of hell, the real reason popped out. They felt that the work wouldn't look good on their CVs. They were VB programmers who wanted experience in all the new buzzword-technologies that Microsoft's marketing department could come up with. A programmer could wear these TLAs (Three-letter Acronyms) on his CV to enhance his status, like a tribal elder with his shiny beads. Worse, they felt that developing workflow systems in a unique scripting language wasn't VB programming, which was their 'raison d' être'. All the clever stuff was closely bound to the workflow interpreter that executed the scripts. Everyone was fearful of going near the interpreter because it smacked of Computer Science: anathema to the jobbing programmer.

Once I realised the problem, I felt enormously sympathetic. Were the IT industry a rational place, it would seem a silly anxiety to have, but one must live by the bizarre rules of a nutty game.

We reached a compromise: we would create an acronym playpen. We would design and build a non-essential module that would adopt all the latest, CV-friendly technologies. We'd all take it in turns to work on it. In return, each team would do a stretch sweating away on the hard work of creating the scripts for the banking processes, and developing the interpreter, and underlying database. This meant that we were all able to claim real practical experience in the hot technologies, but do some real work too. I took great pleasure in designing a dazzling module for printing out the correspondence (confirmation letters and so on) that were generated by the banking processes. It was TLA heaven. As a backstop, we also created a much simpler module that duplicated the functionality in a less elegant way.

The 'playpen' idea worked perfectly. The senior managers were ecstatic. They could boast that the project was using all the latest Microsoft

www.dbooks.org

technologies. The programmers could brush up their CVs until they glowed. We could progress on with creating the application

The ruse of using scripts worked perfectly. The VB script-processor scraped through testing, almost on schedule. Even with the project live, with a vast call-centre humming with people, we were able to modify the scripts 'on the fly', without a frown from the bank's IT department. This was all very fortuitous, since, at the last minute, the Retail Bankers realised that passive resistance would not stop the project, and their amnesia evaporated. They were falling over themselves to help with scripting all the banking processes, and being identified as one of the key participants in the project. The great enterprise got its call centre on time by a simple study of the corporate IT rules, to find the loopholes.

www.dbooks.org

The Writing on the Wall

Complete with braces and gelled hair, the new management team descended on the hapless enterprise, full of ideas for introducing dazzling efficiencies and dynamic new processes. A forty-volume report and several million pounds, all that was left of the project was the writing on the wall.

There is a world of difference between the business processes that really take place in a company and the management's understanding of those processes ...

After several years working as a systems analyst I intuitively understood this fact, but its truth was brought home to me forcefully only after I went to work for a well-known company that specialised in financial services. I'd had quite a bit of retail banking experience but nothing really prepared me for the strange, archaic work-practices of this company. Little had changed since the late-Victorian peak of their enterprise, except that the workforce no longer wore starched collars and ties.

Two years before my arrival, the management had been swept away in one of those whirlwinds that periodically affect such firms. The workforce barely looked up from their ledgers as they scratched away at their timeless rituals, but a new dynamic team of managers, complete with braces and gelled hair, had descended on the hapless enterprise. They were determined to enact a dynamic business re-engineering exercise that would 'transform the company into best-in-class providers of financial services for the new millennium'.

These human dynamos, square of jaw and decisive of action, resolved to introduce startling efficiencies that would, in their words, 'reshape and redefine the whole enterprise in a new and vibrant way'. They decided to call in the management consultants. Suddenly the corridors were full of earnest, humourless young men in gold-rimmed spectacles, neat hair and smart suits. Solemnly they interviewed the staff to determine, and document, the business procedures of the company.

At this point the employees woke up to the reality of what was going on around them. They recognized 'introducing efficiencies' as a euphemism for a drastic slimming of the workforce, 'reshaping the enterprise' as a variation on

the same theme. With the creative energy of people in fear of their livelihood, they gave the serious-suited gentlemen procedures by the bucket load. Every department was able to describe in vivid detail their intrinsic importance to the working of the enterprise, the vital nature of their role. They conjured processes out of thin air.

The management consultants accepted everything at face value. They did not seem to even consider that old Joe in dispatch, whose job was a sinecure that an energetic man could complete in half an hour, might have lied to them when he described the elaborate and skilled nature of his contribution to the company. It was all noted, analysed, diagrammed and documented. The final report was delivered in forty volumes and placed in a large filing cabinet, along with a final bill to the company of several million pounds.

It has always perplexed me that management never question the rule that whatever is said by a serious young man in a suit, gold-rimmed specs, and a smart haircut, must be hot stuff. As these particular smart young men were also from a London firm of accountants and charged stratospheric fees, then their contribution must, they knew, be profound.

As part of the team of IT consultants hired to implement the "business reengineering", I'd been tasked with turning the documented business procedures for the Shipping part of the business into an application. It seemed simple. One just had to start at page one of the documented procedures, implementing everything as described until tired but happy, one reached the final page. I unlocked the filing cabinet and gazed upon what must have been one of the most expensive books in history. The IT director told me, with a hint of admiration in his voice, that I was the first to attempt to read it. With mounting panic I took the relevant volumes to the department and talked to the team that ran it.

"Do you really do all this stuff?" I asked. The response was silence and sheepish looks all round. My second question, "If we created a computer system that did all this stuff, would it work?" was greeted with muffled laughter. At this point, I was shaken by a wave of panic so violent that I stood up abruptly and swept my hand across the wall to steady myself. Their laughter turned to looks of horror, and wails of dismay.

Thinking from their reaction that maybe my arm had dropped off and the severed stump was spraying blood around the room, I glanced around. I had, with my sleeve, wiped off a considerable section of the writing on a whiteboard fastened to the wall. To cut a long story short, the real, simple story of their mundane working lives, had been enshrined in a series of simple charts on a whiteboard, consisting of checks, crosses, dates and names. My carelessness

www.dbooks.org

had erased a section of the true procedures that coordinated their work activities.



What is this strange writing on the wall?!

This, I pondered, was a bit of a give-away. An hour later we had restored the chart and, in doing so I had bottomed out the simple routines and paperwork that governed their working lives. It bore no resemblance to the lies told to the management consultants. Even more alarmingly, it was refined and honed by a hundred years of evolution into a system that was hard to better.

Three months later, they had their computer system. I'd written it, by myself, during my lunch hours. It was based on their whiteboard, in a comforting and reassuring way. It was simple, it worked well, and it required no retraining. The IT director fussed about it as it seemed to embody none of the profound knowledge imparted by the expensive management consultants, but I told him, without blushing, that the essential procedures from the report had been adopted. He sighed with relief, and signed off what proved to be the only successful part of the subsequent business reengineering project that was ever completed.

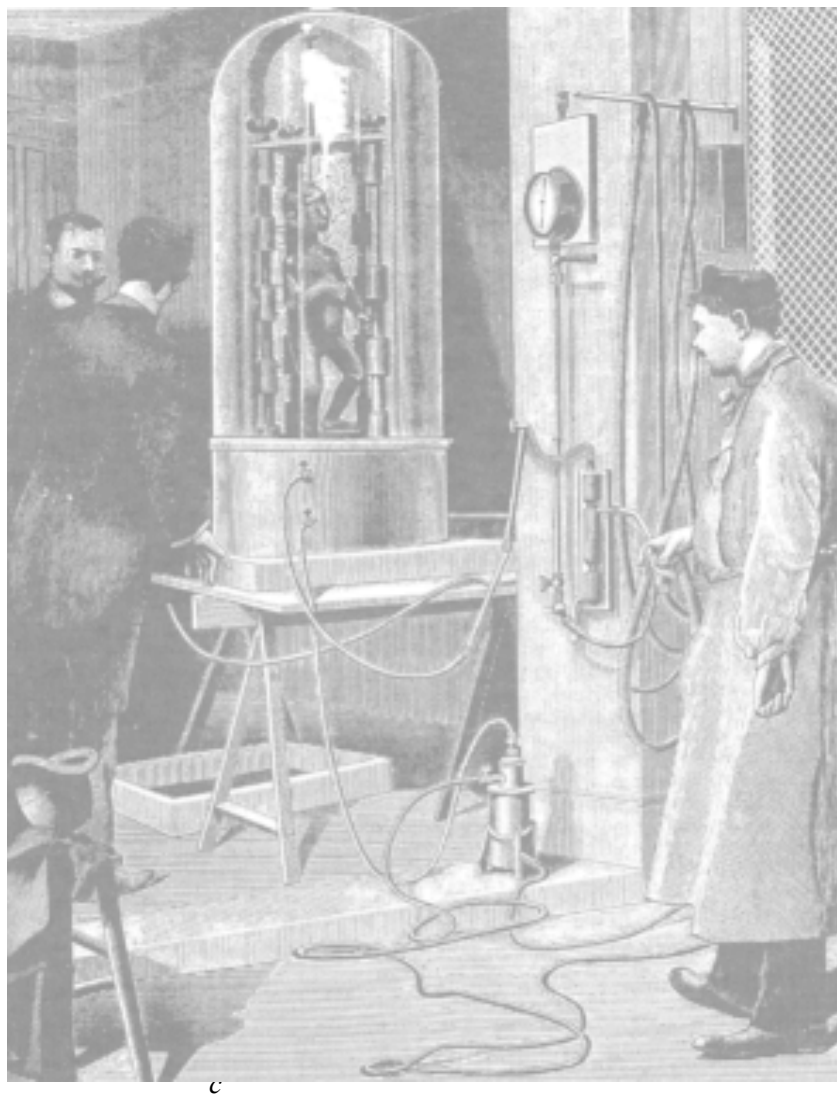
As with all enterprises, the tide of management revolution turned, and the dynamic management in braces and striped shirts all disappeared. Against all expectations, the old traditional parts of the business flourished and the radical new initiatives failed to achieve their promise. The workforce once more settled down to their old ways without the threat of radical reform of their work practices. The only thing that had changed was that visitors were no longer liable to wipe out the organisational system of the shipping area with their sleeves. Their cherished diagrams flickered on computer screens around their department.

The vast filing cabinet with its precious contents was quietly removed, like the Ark of the Covenant, to a convenient cellar. The business reengineering and its colossal expense were quickly forgotten.

So when one hears, with monotonous regularity, that enormous government computer systems fail, could it be that it is not the fault of the IT people who implemented the systems at all? Maybe it is the fact that the government employees puff up the complexity and importance of their roles in the business and systems analysis to the point at which one is implementing a complete fantasy, designed to protect cushy jobs rather than to serve the public? Could it be that, once these specifications and architectures are complete, nobody has the power or the inclination to challenge the veracity of what has been produced?

www.dbooks.org

The Ghost in the Machine



c
olossus, for on that mighty ruin, a thousand flowers can grow.

I have always felt rather second-rate as an IT pundit as I have yet to introduce my own TLA (Three-letter Acronym) into the IT industry. I would therefore like to suggest a new branch of IT: solving IT problems by **Antiquarian Research in Technology,** or **ART**, to aficionados.

Time and again in my working life, I have come across some truly bizarre applications. At first glance, the existence of such a system can seem inexplicable. When called upon to upgrade one of them the natural tendency is to re-write it from first principles. This is time-consuming and often involves more work than is strictly necessary. On occasions when an application is so complex that rewriting it from scratch is unfeasible, one may simply throw one's hands up in defeat and allow the bizarre application to keep trundling on, often way past its natural expiry date. I offer a third way: progress through ART.

I was once called in to help with the decommissioning of a large VAX mainframe. It was getting expensive to run and all but one of the database applications that it was hosting had been superseded. This last application was, I was told, doing engineering calculations of such a specialist nature that it was likely to be understood by only a handful of postgraduate academics with extra lobes to their brains. I was asked to find a home for this application, either by re-writing the application or porting it to a different host, so they could then switch off the VAX. It seemed an impossible task.

In some awe, I visited the computer centre, and was guided round by the manager who proudly showed me the enormously expensive mainframes, the Star-Trek network boxes with their twinkling lights, and the fire-prevention systems that, in a moment, could fill the place with carbon dioxide and exterminate every life-form within.

The analyst who tended the VAX had never had to maintain the system in any way, and had never probed the inner workings of the application, but he did express his puzzlement at the huge amount of disk access that it required. A quick look at the source code was enough to convince me that no human had spawned it. It was completely opaque FORTRAN code with numeric labels and no comments. It looked like the machine-generated output of a 4G Language. I did the only sensible thing, which was to recoil in horror.

When I had calmed down, I mulled over the problem. This was a complex and technical application. The 'real' source code for it was missing and the machine-generated code was un-modifiable. Rewriting it from scratch was not an option. Further, the FORTRAN code was completely machine-specific, so a straight port was out-of-the-question too. Refusing to be defeated, I turned to ART. I felt that if I could find out how on earth the application had wound up where it was, in the state it was, I would be much closer to a solution.

www.dbooks.org

Since all I had were the names of the people using the application, this seemed like the best place to start. Unfortunately, as far as their knowledge of the application went, they merely turned the handle, and got the results they wanted so they proved to be of little help on that front. However, they did provide me with the name of the manager, long since gone, who had originally been responsible for the application.

I met the grizzled old business manager, a plain-speaking engineer, as he tended the roses in the garden of his retirement bungalow. Over a couple of pints of a very fine beer (Mauldon's Blackadder Stout, in case you're interested), he told me his tale.

In those far-off days, he'd been responsible for the design of a vital part of the product that the company manufactured. It was a fiddly and difficult part that was prone to fracture due to the stresses put on it. Over-engineering the part wasn't the solution because of weight restrictions. He decided that the only way to solve the problem was to calculate exactly how the part should be designed purely according to the forces placed on it.

The project was opposed by corporate management, whose 'accountancy culture' failed to understand the requirement. It also was subject to all manner of blocking activities by the IT department, who had a visceral revulsion toward any IT initiative that they did not initiate and control. They used the regulations within the Corporate IT Standards Manual as a stick with which to beat the project. Such an important application had, they argued, to be developed to run on company-standard mainframes in a computer centre.

Nevertheless, he persevered. Since the project was completely unofficial, there was no funding. Armed only with a redundant CP/M-based computer called an 'Intertec Superbrain' that he found in one of the departmental cupboards, and the covert help of a PhD student from a very fine university engineering department, he began work on the project.

Intertec Superbrain had no hard disk and only 64K memory, so the application had to do its own paging onto one of two floppy disks. During its calculations, Superbrain would grind and buzz like a demented coffee machine but the results it spewed out were pure gold dust. They described a design that would enable the department to produce engineering solutions that were cheaper to make and of vastly increased quality.

This caused enormous upset.

Both corporate management and the IT department had to find a way to pretend that they had actually supported the project all along. This was achieved by a process known as 'Grandmothers Footsteps' (or GF), which allowed them to effect a complete about-face, via a series of small and subtle

changes of position, barely detectable to their colleagues. As a face-saving exercise, the IT department agreed to port the successful application to a mainframe in a secure data-centre. This enabled them to say that it then conformed to corporate standards for a 'critical' application that was essential for the prosperity of the enterprise. In this way, peace was restored, though it was peace at a price: the port was expensive, and the application was consequently lost from sight to the people who had designed it, and frozen at that point. It became a black box.

With the knowledge gained from the retired manager, I managed to dig out from the files the address of the software company that had been commissioned to port the application to the VAX. I drove over to see them. The management looked tense and waffled me a blizzard of defensive gobbledegook. I suspect that they knew nothing of the work or whether the proper records had been kept of the job, and they didn't want to risk their reputation with our company. Fortunately, they caved in once I started asking scarily technical questions, and introduced me to the programmers. Cheerfully, and with little guile, they added the next chapter in the story.

When the application had arrived, they had managed to find an identical Intertec Superbrain computer. They observed with fascination the whirring of the disks and the flashing of the access LEDs as it ran the application. They couldn't understand what on earth the application was supposed to be doing. It was written in interpreted BASIC, had its own primitive database and did its own paging to disk.

Faced with an application that they would never be able to fully understand, or reverse-engineer, they did the only sensible thing they could and bought in a product that was able to convert the BASIC code into Fortran, which was then compiled and put on the VAX. The floppy disks were emulated on two small areas of the VAX's mighty drives. Lo, inside the VAX was a little cocoon which held a virtual CP/M application: the ghost of a 'Superbrain' inside the machine. Whenever the application was used, the duty operator in the data centre operator would look up from his newspaper in surprise at the sudden increase in disk activity.

With a tremble in my voice, I asked if they still had the original BASIC source. A programmer bounded down into the cellars and, after a tense ten minutes, returned with the CP/M SuperBrain disks (a special format that we eventually managed to read without major difficulty) and a printout of the source on fanfold 'music' paper. Eagerly we pored over the runes like Egyptologists over a sand-worn inscription. It was all there, with liberal comments, in the old Microsoft CP/M Basic.

www.dbooks.org

So, after a bit of research and some pleasant days out, we were able to take the original BASIC source and convert it to Visual Basic for DOS, pulling out all the paging code in the process. The application then ran on every machine in the department sweetly and silently. So pleased were the engineers, and so cheap was the solution to the decommissioning of the VAX, that the IT Department felt it prudent to forget their stipulation that it had to run in a data centre. The mighty VAX was finally turned off for the last time, and the data centre was able to reduce their staff headcount. Once the code was accessible, the engineering graduates leapt on it and were able to improve it in several subtle ways.

As they became more familiar with the application, one of the graduates realized that, since those far-off pioneering days, the whole science of computer-aided Engineering had moved on remarkably and that a vastly superior application could now be purchased very cheaply and run on a Workstation. At last, old SuperBrain could be fully laid to rest; the ghost in the machine led to the light.

When the engineers got their Workstation application, they invited me over to see its graphical wonders. In glancing through the manual for the new system, my eye was caught by a name: there was a credit to the University engineering department that had helped to develop the original SuperBrain application and the name of the original PhD student (now a professor, I noticed).

Perhaps, I mused, the soul of that old application lived on after all.

www.dbooks.org

The Sticking Page-Down-Key Incident

He who promises more than he is able to perform is false to himself and can do damage to his friends.

Most software projects seem to go on an emotional roller-coaster voyage. Initial euphoria soon fades into confusion, quickly followed by desperation. If a project is going to succeed in reaching port, then out of this desperation must come a consensus decision to save as much as possible from the wreckage of the initial 'vision'. Maybe I'm unlucky, but I've never worked on a project where the voyage didn't involve a storm and a near-shipwreck. In projects that are going to fail, desperation is followed by despair, resignation and a state of apathy, as the rigging crashes around your ears, and the rats desert the sinking ship.



The rats desert the sinking ship

Quite often, in a development project's stage of apathy, I've been hired to report on the problems affecting the project and to try to suggest rapid ways of 'turning the ship around'.

The problem that confronts any 'consultant' faced with providing a simple picture of the state of a project, and the solution to its ills, is to turn a mass of conflicting, and sometimes irrelevant, information into something sensible and intelligible. Most problems besetting a software development project are complex and diffuse. Very rarely, however, one gets the delightful experience of accidentally stumbling on the simple nub of the problem straight away.

It was one of these experiences that I remember as the 'Sticking Page-Down-Key incident'.

It was a telesales application that was being developed by the IT department of a well-known company. The distraught manager showed me the ranks of telesales people in the vast open-plan office, trying, and failing, to use the initial release of the software. Most had reverted to the previous system.

I talked to all the team members individually, and read through the project documentation. If I were to believe everything that was written there, then the project was in wonderful shape. I didn't, because it wasn't.

I was beginning my rounds once again when I settled on the team writing the client application. I started to talk to one of the team members, using my usual trick of pretending to be even more stupid than I really am. I asked him about the technology; He gave me a 'Word Salad'. 'Word Salad', I should explain, is a technical term that describes stringing phrases together in a random order. It is an attempt to bluff the user into believing that the words convey subtle meaning too intricate for the listener to comprehend. It is much used in IT to cover-up a lack of knowledge.

I find being the recipient of a 'Word Salad' particularly irritating. Whilst fidgeting around, willing him silently to stop, I accidentally leant on the keyboard and the 'Page-Down' key.

I stared transfixed at the screen. Initially, there was a flurry of movement on the screen and then nothing, just static text: Nothing, that is, excepting for a number changing at the top of the screen. I pulled myself off the keyboard. The number stopped moving. I pressed the Page-Down again: a twitch of the number, but the rest of the screen static. After what seemed like an age, the screen suddenly burst into flickering life.

This programmer had been making a large number of SQL calls to the database, just changing one parameter every time. It just so happened that each iteration took up exactly a screen-height when the editing window was maximised.

I scrolled back to the start of the source-code file and pressed the Page-Down button again. Normally, when you hold the Next-Page button down

www.dbooks.org

whilst viewing source-code, the scrolling is so rapid as to be a blur, creating, on the screen, an animated line on a white background. It wiggles from the left slightly as code gets indented. In this case, it wiggled, and wiggled, across the width of the window, and on until it disappeared off the right. Page after page flashed by; occasionally it wiggled provocatively into view from the right hand side, like a fan-dancer from behind a curtain, only to wiggle off-screen again.

I sat mesmerised.

After what seemed like an age, the blurred vertical line of code wiggled firmly back into view and gradually across the screen until finally it laid itself to rest once more on the left hand side. End of file. No procedures, no subroutines, just stream-of-consciousness coding.

A few straightforward questions elicited the obvious fact that he knew less about programming than I did about Chinese ballet. It was a curious thought that his code had been peer-reviewed, a thought that led me to look at the code for the whole team.

When I broke it to the project manager that the team probably had many talents, but he was not to rely on any programming skills from any of them, he was nonplussed and showed me the CVs. They were so good I had the sudden uncontrollable urge to hire them on the spot before I shook myself fully awake.

With everyone's consent, I arranged for the CVs to be checked out by a specialist. It is, perhaps, little known that it is very easy, for a modest fee, to arrange for a CV to be checked for its veracity. When my researcher phoned back, he was gurgling with laughter. They were almost completely fictional. Qualifications and experience were completely invented. Nothing checked out at all beyond the basic personal information. They were all contractors and all hired from the same agency. They were all appointed at the same time. Funnily enough, they all left at the same time, leaving a project headed once again on the right course.



They all left at the same time

I'll be the first to admit that this was an unusual occurrence, and it was strange that the developmental methodology, or team processes, had not rapidly highlighted the source of the problem. However, real life seldom goes quite like the book says it ought to, and the reality of the software industry is, just occasionally, more absurd than one would dare to imagine.

www.dbooks.org

The Escape from Developer Hell

Where the instrument of intelligence is added to brute power and evil will the developer is powerless in his own defence. Had Dante lived today, he would have dreamed up a special place for the sinful soul of the dead developer, ruled with fiery menace by the incompetent DBA.

"C'mon Phil! You can't blame developers all the time! You must have encountered a really bad DBA at least once in your working life!"

The developer flushed angrily as I rambled on, recounting some of the hilarious mistakes that developers make when tackling databases. Somehow, my SQL horror stories didn't strike him as amusing.

"Well, strangely enough, yes. Once, the boot was on the other foot, entirely. There was a time when I directly experienced the sort of hell that Dante would have dreamed up for the sinful soul of the dead developer."

I leant back in my chair, and put down the soothing cup of Darjeeling tea.

"But we escaped relatively unscathed in the end …"

The scene goes watery, and there is weird music …

It was during one of those sporadic downturns in the industry that I was ejected from a rather cozy job with an international Financial Services Provider. I'd been a Technical Architect specializing in documenting and maintaining the corporate data model. A CV that mistakenly lists this sort of esoteric skill is anathema: the phone never rang, and my CV hung limply on all the jobsites. Eventually, I did what everyone does in these circumstances and reverted to being a VB developer. I'd used the language now and again, though I'd never previously admitted to it. It was like confessing to unnatural practices.

Soon, I was sitting at a desk in a vast office-block in London, heading up a newly formed Dev Team of six developers and ten testers. We were tasked with writing a Database-driven ECommerce application.

As there was not a line of code written, the testers busied themselves scanning social networking sites on the internet and trying to look important. I

looked around for the Development DBA. There wasn't one. The Project manager told me in hushed tones that they had just acquired an expert in SQL Server. Apparently, he used to be a big cheese in Microsoft but was now an independent consultant; one of their top people, I was told. I felt it prudent to keep quiet about my many years of Oracle and SQL Server experience in view of the fact I'd majored more on my VB skills in my job-application. No lies, you understand; just balderdash.

By the time Benjie finally arrived, the team had worked itself into a fever-pitch of anticipation. A real Microsoft expert! Wow! I phoned a few of my Microsoft contacts but they couldn't place his name. Benjie, a most affable and charming Nigerian, was introduced to the team by the deferential project manager. After a month of work, he produced a ER diagram of the database for the application. I was slightly apprehensive, since it looked wildly complicated. 'What,' I remember asking 'is an EventHospitalityGiftItem table for?' I gave the diagram a bleak, but thorough, inspection. It was scarily awful. I was full of forebodings.

A couple of weeks later, he had created the database and provided a stored procedure-based Interface. I'm all in favor of stored procedures, except that in this case the spCustomer procedure had 92 parameters, all of which, he said, had to be supplied on every call. The devs hadn't been consulted, and they started to give vent to their discontent about the endless problems this interface was going to cause them.

Unfortunately, the project manager was still hypnotized by the aura of Microsoft magic surrounding Benjie. Under this zealous protection, Benjie basked in the glory of his facile dictatorship, and the wacky interface was imposed on us, despite all our protests.

And so began a period of Developer Hell. For a month, we suffered The Curse of The Incompetent DBA. Nothing in the database worked reliably, so the Developers became very adept at their error handling routines. The design was so mad that even the simplest data sets were tedious to extract. All our requests for extensions to the interface were refused on grounds that it wouldn't scale, offended against the relational model, was against the corporate data model, was a security risk, would overload the server, or was unnecessary. We had, of necessity, to add business logic on the client that should have been done on the server. The database became a no-go area. It was worse for me, as I could see simple solutions for problems that kept us programming into the evenings, and, besides, I had used all of Benjie's excuses myself in the past. There is nothing worse than having to program to a cockeyed database schema.

www.dbooks.org

Eventually, the team got so restless that we decided that something radical had to be done to save the situation. In desperation, I phoned a friend who worked for Microsoft in Ireland, at their support center.

"Keith, I'm working on an ECommerce project with a guy here that used to be with Microsoft. His name's Benjie. Do you know anything about him?"

"Benjie!" trilled Keith, "Give him my love. What a nice guy! We took him on as part of our first response team at the call-centre, as he had such a good phone manner. The trouble was he knew almost nothing about the technology, so we had to let him go in the end."

"Well, he's turned up here as our database expert." I told him, dejectedly.

There was a strange wheezing sound from the other end of the phone.

"Database expert you say? Gimme a break! My pet terrier 'Buster' knows more about databases than Benjie!!" Keith hooted with laughter. "He's a VB programmer by trade, I believe: well that's what he told us. He's quite a character though. I remember when he offered to show us his Mapouka Dance at our Christmas party. Epic: we still talk about it to this day. "

I immediately set off in search of Benjie. When I found him, I could see immediately that he was already in a state of considerable panic. It seems that, through canteen-gossip, perhaps, the Production DBA team had got wind of the news that a database-maniac was on the loose. The Production Manager and senior DBA had just marched into our project manager's office and demanded a formal review of Benjie's work.

I added to his misery by telling him that I'd rumbled his CV. His eyes swiveled in guilty remorse. He looked so dejected and morose that I felt a surge of pity for the chap. Anyway, my own CV as a VB developer wasn't exactly whiter than white. In fact, there was a strange symmetry here. Here was a VB programmer masquerading as a DBA and there was I, a DBA, pretending to be a VB programmer. Damn, there must be some way out of this. There was a quiet pause as I tried to think of something and Benjie blinked sadly and stared miserably out of the window. Suddenly the solution came to me. "Benjie, " I started, "I have a solution, but it will hurt; you, not me …"

The following week, the Production Managers arrived, with a team of DBAs in tow, to perform the review. He gave Benjie the look that a huntsman gives to a cornered fox. However, in contrast to the mayhem and confusion that had defined the previous weeks, today all was sweetness and light in the office. The developers all looked relaxed, contented, and busy. Had any bird been able to do more than cough asthmatically in that part of London, you'd have heard sweet birdsong and Elysian harmony.

The server was given a thorough review. No stone was left unturned. The data model, documentation, build scripts and interface specification were all scrutinized in great detail … and pronounced to be near-perfect! The DBA team left, frowning in puzzlement.

I'd completely re-written the database over the weekend, with the help of a couple of the Developers who knew the ropes.

Benjie had happily agreed to my terms. This included the condition that he must reveal to us developers the secrets of the the quivering Mapouka Dance. Also, from now on, he was to be just a figurehead and wasn't allowed to touch the database without my approval. However, he had squeaked with delight when I showed him how to rewrite his 92-parameter stored procedure. He seemed amazed that one could provide default values to stored procedures in SQL Server, and that one could also detect what parameters had been supplied.

From that point on, I and an accomplice administered the database and did all the database dev work, while Benjie happily settled into his new role.

We kept the secret without problems. Any seasoned DBA knows how to look busy without really doing anything, but I was able to give Benjie a few good tips such as …

If you are seen to be studying an ER diagram, nobody thinks you are idle.

Always have animated performance graphs on your screen, displaying as many criteria as possible.

Have several open DBA reference books on your desk.

Never automate any of the routine tasks such as scripting out, doing backups, testing restores, BCPing out tables, running and monitoring data feeds.

The project proceeded harmoniously, and met its dates. As the recession eased, I was able to crawl out from my agreeable shelter and get back to my primary career. Benjie left at the same time, unsurprisingly.

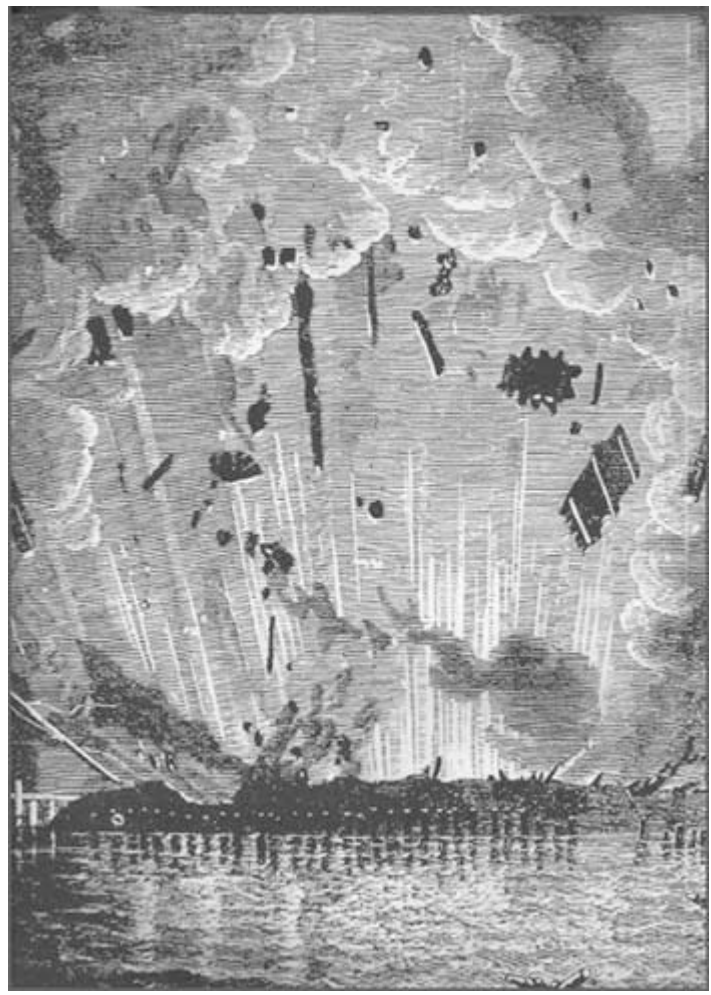
That should be the end of the story but, to this day, I still keep hearing stories about Benjie's subsequent career. His CV positively glowed after his success at our project. He is such an engaging character that he continues to carve out a career for himself around London. Last I heard, he was working as a DBA for a media company.

Long after his IT contributions are forgotten, I will still cherish his descriptions of the quivering Mapouka Dance, and of the musical language of the Yoruba.

www.dbooks.org

The Time Bomb

Phil cheerfully admits to his fair share of accidental IT disasters. However, his software 'Time Bomb', the explosion of which caused civil unrest in a far-away country, was deliberate.



The bolt from the blue

The phone call, when it came, was memorable. Over a crackly line, with all sorts of strange delays and clicks, came a panicky voice …

> "Pliss, Help! Your softworr is, how say, flaked, err … flooked. Iss noggod. It helps me please. I used his software of the sheet of payment and I do not encase any more long payment of the employees his software is broken, they starts agitated."

I was puzzled. The voice was strained. I could have sworn I could hear raised voices in the background followed by breaking glass and a scream. I doubt if this sort of catastrophic failure happens to many software developers. It was something that, I have to admit, was entirely the consequence of my own actions.

I'll have to explain …

The Gap in the Market

The independent software publisher survives by reacting quickly to small gaps in the market. It is always a happy time when one manages the trick. At the dawn of the PC, there was a battle between many manufacturers of electronics to design the definitive personal computer, and thereby dominate the fledgling market in affordable computers. Each machine that came out was different. Usually they had a primitive but serviceable BASIC interpreter, generally in ROM, and you got the ubiquitous disk operating system, CP/M. Each computer would be able to run Microsoft Basic and usually provided hooks into its hardware via extensions to BASIC. Manufacturer-after-manufacturer would release beautifully-designed computers onto the market. Each had only one minor flaw: no useful software.

Such is the overriding power of marketing that many computers were sold without any useful applications that could run on them. Even simple spreadsheets such as VisiCalc and Multiplan were unknown. However, it was usually not long before these vast multinational companies, big in Hi-Fi music players but innocent of any knowledge of computers, were casting about for some expertise.

It was at this point that my company would come to the rescue. After admiring the beauty of the sculptured desk-jewellery that was shipped to us from the panicking company, our first job was to crack the format of the disk so

www.dbooks.org

that we could get industry-standard software onto the machine. Then we would install Wordstar, the only serviceable word-processing software that existed at that time. We'd then write drivers for each of the small nucleus of applications that were on the market for doing standard office functions. We'd ship all that work to an associated company who would then sell the 'ready-to-run' software, in the correct format, to anyone stupid enough to buy the computers. We grew plump and prosperous on the work, and got to be on first-name terms with the UK directors of a number of Japanese and American multinational companies.

Once the initial crisis was over, we'd start to introduce special business management software, providing accounting and payroll functions to smallish businesses, each one 'individually tailored' to the computer in question. Well, that was what we'd tell everyone. Actually, the trick would be to maintain a complete range of accounting software packages, each written in the anodyne generic Microsoft Basic. It assumed nothing of the hardware it had to run on beyond the ability to emulate a teletype. This was the same Basic that Microsoft would license to these hapless and inexperienced companies, but with hooks into the graphics and other I/O. This generic software was maintained by a small team of programmers. Whenever a new computer was launched in the industry, we would be up most nights converting it into a 'custom' version for the machine, complete with graphical, sound and I/O features 'specially adapted' for that particular piece of hapless hardware.

This always went down big with the marketing arm of the manufacturers. Even if the computer bombed in the marketplace, we would still rake in a healthy amount of cash by consulting with them until, eventually, a wider sanity prevailed and they would extract themselves from the PC market, licking their wounds.

The only major downside with this enterprise was the fact that the BASIC was interpreted, and Microsoft Basic had no effective encryption. Software had to be provided to the customer in source. We considered drafting a legal document, stating that whenever you hit the BREAK key, and the source scrolled out across the screen, you had to promise to close your eyes, put a paper bag over your head, or turn your back on the computer. However, we doubted whether we'd be able to enforce it.

Instead, we resorted to a number of tricks to make sure that the source of all our prosperity was not copied. It was vital that nobody else realised how easy it was to make a killing from the battle for supremacy amongst the PC manufacturers.

The code was therefore full of tricks, traps, and unpleasantness to thwart even the most determined software pirate. It might be my vindictive nature, but

I made sure that these tricks were extremely subtle and difficult to detect. Only the core team that maintained the code, and I, knew them all.

Now, back to our panicking caller …

The Elbonian Payroll package

In broken English, a lady from a country we'll call 'Elbonia' was pleading for help. She had one of our payroll packages and it had stopped working. We'd never sold any software to Elbonia, and had no knowledge of the labyrinthine laws governing payroll in that country. She seemed reproachful that we should have written software that had failed, and was in an awful predicament as she couldn't use the system to pay the staff. A quick check revealed that it was, indeed, our software. It had been altered, but it still had our copyright message and contact details in place (just try removing them! Hah!). We were completely bemused but, eventually, we took pity on her predicament and gave her enough information to enable her to bypass the copy-protection device that had kicked in, in order to do that particular payroll run.

As soon as the call ended, one of us had to rush out to buy an atlas to find out where Elbonia was. As it turned out, this was just the first of a flood of phone-calls from this small South American country, all of them characterized by the same atmosphere of barely suppressed mayhem and panic. They were generally fielded by others, as I was trying to find out what had happened.

I quickly managed to establish that the marketing director of the local office of the computer's manufacturer in Elbonia had naively got in cahoots with a local Elbonian software house to 'alter' our payroll software to meet local needs and had 'forgotten' to ask our permission. How unfortunate.

Elbonians had taken to the computer with a gusto unmatched by its reception in Britain. I suspect that the marketing director had also 'forgotten' most of the Ten Commandments in his commercial zeal to sell the computer to Elbonian commerce. Sadly, they walked right into one of our 'traps': a time-bomb that misled the pirate into believing that he had successfully filched the code, only for it to subsequently 'blow up' in his face.

The software had been adopted with such zeal that the time-bomb, one of my best, kicked in more or less simultaneously across many companies throughout the country. On this particular Friday, a lot of Elbonians didn't get paid and they were understandably miffed. It was causing disturbance on a national scale.

www.dbooks.org

An agreeeable cure for a Senior Moment

My meeting with the boss of the vast international manufacturing company, in an exclusive Sushi restaurant, near Covent Garden, was a most agreeable event. He was a cultured, urbane Japanese; a Harvard graduate. We bonded instantly. He was in excellent spirits, despite the jet-lag from his sudden trip to London.

"The sudden amnesia of our employee in Elbonia was a sad and unfortunate event," he confided. "He has, sadly, had to leave the company on medical grounds. We are left with an unfortunate problem that requires us to come to you in the hope of your assistance."

"The memory can play terrible tricks, as you are only too aware." I replied. "Unfortunately, none of us can quite remember where the anti-pirate device is located."

He nodded vigorously. "I quite see the problem, but we Japanese have an ancient remedy that is guaranteed to stimulate the failing memory, which I can now offer you." He produced a cheque-book, and wrote out a cheque for an amount so large that even now the thought of it makes me quiver with excitement. "My God, it's worked! I've suddenly remembered where it is! Excuse me whilst I make a phone call ..."

I soon returned to the table, after contacting the programming team to authorise them to release the patch to Elbonia, and to let them know that their annual bonus would include a new car.

The rest of the evening was rather a blur, but gave me a new respect for the ability of senior executives of multinational companies to consume alcohol and get thrown out of Soho Clubs. It was a revelation.

Suddenly, the threat to the political stability of Elbonia was averted and the computer was left to pursue its mercurial fortunes.



For us, it was soon time to look for a new gap in the market, that brief but profitable opportunity, as the industry re-invented itself once again.

www.dbooks.org

On the Trail with the Cowboy Coders

Testing code before you release it? Who would have thought of such a thing? Is there no end to what Agile Programming can do? Phil is, once again, left shaking his head in disbelief.

One of the signs of increasing age in the IT industry is a frequent and acute sense of 'déjà vu'. New things that are laboriously explained to you ring all sorts of bells. The past flashes before one's eyes.

I recently visited a company developing Internet-based applications. They were proud of their progressiveness and explained to me that they developed database systems using the radical new Agile XP (Extreme Programming) methodology. Wow, I thought. That must have impressed the shareholders. Perhaps they don't realize that Agile XP is known in the industry as 'Cowboy Coding'.

I feigned ignorance, which is not usually hard for me to do.

> "Well," said my genial and pleasant host, "One of the great things about Agile XP is that you test every part of your code as soon as you've written it. XP uses what it calls 'Unit Tests'. These are automated tests that test the code. The programmer will try to write as many tests as they can think of that might break the code they are writing; if all tests run successfully then the coding is complete, and the programmer can then go on to develop more code. Flaws in the system are easily communicated by writing a unit test that proves a certain piece of code will break."
>
> "Gosh! You mean you test your code rather than just write it?"
>
> "Ingenious, isn't it? When writing code, the unit test provides direct feedback as to how the system reacts to the changes one has made. If, for instance, the changes affect a part of the system that is not in the scope of the programmer who made them, that programmer will not notice the flaw. There is a large chance that this bug will appear when the system is in production."

> "Funny how nobody has ever thought of this before!"

My face goes red when I'm desperately trying to stop myself guffawing with laughter. What I love best about working in IT is its rich, unintended comedy. My host misunderstood my visceral difficulties as excitement. Seeing how interested I was he proudly produced an elaborate Java-based module he'd written that provided a test harness for SQL queries and procedures.

> "It is great, you can change the parameters in the query (Klop Klop Klop on the keyboard) just type in what results you expect", (Klop Klop Klop Klop), "and then Run the test harness." (Klop Klop Klop Klop Klop Klop)

Various windows flashed on the screen in front of me, full of inscrutable commands. It was all very 'Seventies'. Finally a message popped up from the depths saying 'OK'. He was so proud; he could have been showing me a photo of his baby.

'Look, all we have to do is to change the SQL code a bit in order to simulate an error and you'll see what happens!'

(Klop Klop Klop Klop Klop Klop)

An error message popped up saying there was an error in the code. Wow!

> "To think that, for all these years, we've been doing it so stupidly." I sighed, shaking my head in mock-regret.

Before I left, I think I did enough to persuade him that perhaps there had been some sort of thought, however primitive, given to testing code, before the genesis of Agile XP. However, I didn't get the chance to actually show him how an old SQL Programmer like myself would go about doing exactly the same thing, in their humble rudimentary way. To make up for this, I thought I'd include an example here.

Before you write code, you write a test-harness. It is often best to get someone else to do it, preferably someone with malice in their soul. When you write the piece of SQL code, such as a function, you append the test harness for unit-testing purposes. In some places I've worked, you'd be given a rough time at code-review if these were missing or inadequate. Here is an example … (normally, you comment out the test harness when the code is 'put to bed' and the GO moved to the end, so that the harness stays with the code, or you can, in SQL Server, save it as an extended property.)

www.dbooks.org

```
-- ============================================
-- Author:                      Phil Factor
-- Create date:
-- Description:                 Reverse Charindex. Gives the
index
-- INTO the string OF the LAST occurence OF the
-- substring
-- ============================================
ALTER FUNCTION [dbo].[rCharindex] --reverse Charindex
(
   -- Add the parameters for the function here
   @needle VARCHAR(80),    --the substring
   @HayStack VARCHAR(8000) --the string
)
RETURNS INT
AS
BEGIN
   IF @needle+@Haystack IS NULL RETURN NULL
   IF CHARINDEX(@needle,@haystack)=0 RETURN 0
   RETURN COALESCE(LEN(REPLACE(@haystack,' ','|'))
             -CHARINDEX(REVERSE(@needle),
                        REVERSE(@needle+@haystack)
                        )
              -LEN(REPLACE(@needle,' ','|'))+2,0)

END
GO
DECLARE @Test VARCHAR(80)
SET @Test='First test'----------------------------
IF dbo.rcharindex ('this','This can NOT be a thistle ')
   <> 19
   RAISERROR ('Failed the %s',16,1,@test)
SET @Test='Second test'---------------------------
IF dbo.rcharindex ('pr',
  '  May the lord preserve me from the priests of the IT
industry  ')
   <> 37
   RAISERROR ('Failed the %s',16,1,@test)
SET @Test='Third test'----------------------------
IF dbo.rcharindex ('l',
  '...who maintain an evangelical following.. ')
   <> 35
   RAISERROR ('Failed the %s',16,1,@test)
SET @Test='Fourth test'---------------------------
IF dbo.rcharindex (NULL,
  '...by waving their hands a lot.. ')
   IS NOT NULL
   RAISERROR ('Failed the %s',16,1,@test)
SET @Test='Fifth test'----------------------------
```

```
IF dbo.rcharindex (' without their ties on.',NULL)
   IS NOT NULL
   RAISERROR ('Failed the %s',16,1,@test)
SET @Test='Sixth test'--------------------------
IF dbo.rcharindex ('','')
   <>0
   RAISERROR ('Failed the %s',16,1,@test)
SET @Test='Seventh test'------------------------
IF dbo.rcharindex ('','and looking sincere')
   <>0
   RAISERROR ('Failed the %s',16,1,@test)
SET @Test='Eighth test'-------------------------
IF dbo.rcharindex ('but it adds to life''s rich humour','')
   <>0
SET @Test='Ninth test'--------------------------
IF dbo.rcharindex ('Klop','so perhaps one should be
tolerant')
   <>0
   RAISERROR ('Failed the %s',16,1,@test)
SET @Test='Tenth test'--------------------------
IF dbo.rcharindex ('Klop','KlopKlopKlop ')
   <>9
   RAISERROR ('Failed the %s',16,1,@test)
SET @Test='Eleventh test'-----------------------
IF dbo.rcharindex (' Klop ','Klop Klop Klop ')
   <>10
   RAISERROR ('Failed the %s',16,1,@test)
SET @Test='Twelveth test'-----------------------
IF dbo.rcharindex (' ','      ')
   <>6
   RAISERROR ('Failed the %s',16,1,@test)

SET @Test='Thirteenth test'---------------------
IF dbo.rcharindex (' ! ',' ! !!!  !  !!!     ')
   <>8
   RAISERROR ('Failed the %s',16,1,@test)
```

There are many cleverer ways of doing it but I blush to think of the stupid bugs that this sort of system catches, that slip through casual testing.

www.dbooks.org

When the Fever is Over, and one's Work is done

Like the mumps, 'EAV' disease strikes the afflicted only once, but the memories of suffering and delirium last forever.

There comes a time in the life of anyone working with databases when they are struck with a peculiar mental aberration. For want of a better term, it is usually called 'One True Lookup Syndrome' or OTLS. It is in the same league as the rather more severe affliction 'EAV disease' (Entity Attribute Value). It is like measles or mumps in that it afflicts the sufferer just once and he, or she, is thereafter immune. The memory of the suffering and delirium is retained, however, and the feelings of embarrassment.

> The sufferer is easy to recognise. He shakes his head in wonder, saying "This is like, such a cool idea, how come nobody has thought of it before? It is like wow! I mean it is so neat." A classic symptom of the disease is that they will sidle up to you and ask how to patent software ideas. We grey-muzzles will shake our heads sadly, with a pitying smile, and just mutter solemnly 'One true lookup table'.

Usually, OTLS comes about when one is forced to create a number of simple lookup tables which contain a code and a name. One gets the sudden flash of inspiration that one could roll them all into one lookup table, with a unique identifier for each row, and a type field, thereby saving the chore of creating and maintaining a number of tables. EAV disease, like OTLS, both tend to be the result of pressure from programmers to push their OO designs back into the relational database. There is loose talk of 'persisting object data in a data store', as though one was storing apples in a shed. It also comes about when a development team attempts an 'Agile' development before they've fully understood the data model.

I'm a recovering EAV sufferer and I remember the first symptoms well. I was under stress, working with a systems analyst whose analysis skills were at about the same level as a dead sheep's. The database was for an internet system for car dealerships, selling second-hand cars. It was supposed to value your stock, and to enable groups of dealers to supply a punter with the exact car he

was looking for. The systems analyst was making very heavy weather of understanding the business processes of the dealers. He kept changing his mind, yet wanted a system that he could demonstrate to the dealers. It suddenly occurred to me that, instead of having a normalised and rational data model, one could store the 'soft' attributes of each object in a single table, the 'Entity Attribute Value table'. I could then accommodate a changing schema without changing the structure of any table.

It all seems so easy. You have an 'object' table that stores the metadata of each object. If you are severely stricken, you make this hierarchical, so that you can implement inheritance. A swift lookup will tell you what attributes there are for any particular object and what their data type is. If you are heavily into Dynamic SQL, then you'd add rules, constraints and defaults. You then have one instance table that is shared by all objects. The value is stored as a string, just as it is in SQLite. When a new object is created, it receives a unique identifier, and its attributes are written into this table.

With this system, you can make changes to your data structures 'on the fly' without any apparent need for a-priori database design. You then will wish to record the relationships between your objects. In my case, I simply popped in a relationship table. I could then do fairly complex queries with remarkable ease:

- How many different models of car were manufactured by Chrysler in 2003?
- What is the current value of all the cars on the various forecourts?
- What Ford estate cars are currently for sale on any of the forecourts?

All of these questions were answered in the blink of an eye. I loaded the system with all the current data from 'Glass's guide', so that I had all the necessary information for all the cars that had been in volume production for the past twenty years. Everything worked fine. I created test data by inventing several subscribing dealerships, and filled their imaginary forecourt with many imaginary cars. I then created an application interface based on stored procedures. I was pleased with, and rather proud of, my work.

The application programmer who'd been assigned the task of creating the Internet site hated the interface. He was accustomed to having direct SQL access to tables via Cold Fusion. His first step was to create a scrollable listbox containing the details of all the cars on the forecourts. At first, he claimed that Cold Fusion didn't support stored procedures; then he decided that my stored procedure wouldn't work. After I'd tired of trying to explain how to go about it, I supplied a view for each object, which was created dynamically from the Object table every time a change was made. He was much happier for a short period but then decided he wanted to update the 'Table'. The view wasn't

www.dbooks.org

updateable. I wrote a stored procedure and showed him how to use it. He began to feel resentful that his programming knowledge had been 'shown up' and began to drop hints around the office that the database was no good.

To the Systems Analyst, who was out of his depth and struggling to explain the slow progress he'd made in coming up with the business and process model, the rumour of problems with the database was an unexpected lifebelt, which he grasped gratefully. When pestered by the project manager, he put on his best 'Mr Sincerity' face and reported that it would have all been finished had it not been for the confusion caused by Phil Factor's wild and wacky database design.

Things were getting difficult. A string of anxious managers demanded that I explain the database design to them. As they had no grounding in either object methodology or relational databases, this was always going to be a doomed mission.

It must have been around the time that I devised a meta-language to describe the objects and their relationships, that I had a sneaking suspicion that all I'd done was to write a database system in SQL Server. My tables were beginning to look eerily like the system tables in SQL Server.

Another problem loomed. I still needed to implement the checks, constraints and rules that are the bedrock of maintaining the data integrity. It is really tempting fate to say that, if there is an interface based on stored procedures, then bad data never gets into the system. One law of Relational Databases is that, if you put in all the necessary rules, checks and constraints, you never see bad data, and if you don't, then it will somehow insinuate itself in by supernatural means. If you have mixed entities in one table, then the business of implementing these checks becomes very difficult.

I was reaching crisis point with the database, when my guardian angel stepped in and produced a buyer who wished to purchase the system. My database, flawed as it was, had served its purpose, which was to provide a slick demonstration of a system before the hard work of understanding the real business processes had been done. The appearance of the buyer opened the purse-strings, and I bade farewell to my beautiful, elegant, complex data model, and replaced it with something more conventional.

I did not pine for it, since the fever of OTLS and EAV had passed.

www.dbooks.org

The Computer that Swore

Database Developers occasionally get crazy ideas into their heads. Phil should know; he's had a few completely loopy thoughts that he's pursued to destruction. The one that still makes him wince is the time when he caused a production server to swear at customers.

Ordinary people have great difficulty in remembering numbers, even in the short-term. Computer people find this hard to appreciate, and often insist that end-users must remember them so as to identify themselves, their bank accounts or their invoices. These numbers often end up being written around the edge of the monitor, or on bits of paper around the desk, which rather defeats the object. Errors abound. On websites, there was a time when we tried giving out numbers that represented the primary key for a customer, but we've all now abandoned that idea as being hopeless, and now use the email address to provide a unique ID for visitors.

People can remember words far more easily than numbers and it occurred to me that, whenever we wanted customers to remember an ID, we ought to translate it into a memorable nonsense-word, and give them that instead.

My idea was to create a program that would translate numbers into words, and vice versa. It meant that, instead of a number, you could give someone a nonsense word, rather like the name of an alien space-traveller in a Science-Fiction film, which they found they could remember. When it was fed back through the routine, it produced the original number. Genuis!

As it happens, it wasn't such a good idea after all, but we'll come to that in due course.

A base-245 numbering system

All you need to do is to create a table with the main consonant/vowel combinations of most languages:

```
CREATE TABLE [dbo].[Syllables] (
 [TheIndex] [int] IDENTITY (1, 1) NOT NULL ,
 [Syllable] [varchar] (4) COLLATE
```

```
SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO
```

Then, a simple stored procedure can be written that stocks it:

```
   CREATE PROCEDURE CreateSyllableTable
AS

DECLARE @Possibilities VARCHAR(255)
SELECT @Possibilities='B C D F G H J K L M N P R S T V W Z
SCSKKNSNSPSTBLCLFLGLPLSLBRCRDRFRGRPRTRVRSHSMGHCHPHRHWHBWCWSWT
W'
DECLARE @ii INT
DECLARE @jj INT
DECLARE @iiMax INT
DECLARE @Consonants VARCHAR(2)
DECLARE @Vowel VARCHAR(2)
SELECT @ii=1, @iiMax=DATALENGTH(@possibilities)
WHILE @ii<=@iiMax
   BEGIN
   SELECT @Consonants=RTRIM(SUBSTRING(@Possibilities,@ii,2))
   SELECT @jj=1
   WHILE @jj<=5
       BEGIN
       SELECT @Vowel=SUBSTRING('AEIOU',@jj,1)
       INSERT INTO Syllables(Syllable)
           SELECT @consonants+@Vowel
       SELECT @jj=@jj+1
       END
   SELECT @ii=@ii+2
   END
```

Execute this stored procedure so as to fill the **Syllable** table. The result is a list of around 245 vowel/consonant combinations:

```
   BA, BE, BI, BO, BU, CA, CE, CI, CO, CU, DA, DE, DI, DO, DU,
FA, FE, FI, FO, FU, GA, GE, GI, GO, GU, HA, HE, HI, HO, HU,
JA, JE, JI, JO, JU, KA, KE, KI, KO, KU, LA, LE, LI, LO, LU,
MA, ME, MI, MO, MU, NA, NE, NI, NO, NU, PA, PE, PI, PO, PU,
RA, RE, RI, RO, RU, SA, SE, SI, SO, SU, TA, TE, TI, TO, TU,
VA, VE, VI, VO, VU, WA, WE, WI, WO, WU, ZA, ZE, ZI, ZO, ZU,
SCA, SCE, SCI, SCO, SCU, SKA, SKE, SKI, SKO, SKU, KNA, KNE,
KNI, KNO, KNU, SNA, SNE, SNI, SNO, SNU, SPA, SPE, SPI, SPO,
SPU, STA, STE, STI, STO, STU, BLA, BLE, BLI, BLO, BLU, CLA,
CLE, CLI, CLO, CLU, FLA, FLE, FLI, FLO, FLU, GLA, GLE, GLI,
GLO, GLU, PLA, PLE, PLI, PLO, PLU, SLA, SLE, SLI, SLO, SLU,
BRA, BRE, BRI, BRO, BRU, CRA, CRE, CRI, CRO, CRU, DRA, DRE,
DRI, DRO, DRU, FRA, FRE, FRI, FRO, FRU, GRA, GRE, GRI, GRO,
GRU, PRA, PRE, PRI, PRO, PRU, TRA, TRE, TRI, TRO, TRU, VRA,
VRE, VRI, VRO, VRU, SHA, SHE, SHI, SHO, SHU, SMA, SME, SMI,
```

www.dbooks.org

```
SMO, SMU, GHA, GHE, GHI, GHO, GHU, CHA, CHE, CHI, CHO, CHU,
PHA, PHE, PHI, PHO, PHU, RHA, RHE, RHI, RHO, RHU, WHA, WHE,
WHI, WHO, WHU, BWA, BWE, BWI, BWO, BWU, CWA, CWE, CWI, CWO,
CWU, SWA, SWE, SWI, SWO, SWU, TWA, TWE, TWI, TWO and TWU
```

This then gives to a base-245 numbering system where each syllable represents a digit.

Encoding and decoding the numbers

Encoding a number is then easy:

```
CREATE PROCEDURE encode
/*
   Don't forget to execure CreateSyllableTable before
   use
eg
Declare @TheCode varchar(100)
Execute Encode 694852357567584,@TheCode output
Select @TheCode
*/
@TheNumber DECIMAL(18,0),
@TheCode VARCHAR(100) output
AS
DECLARE @Dividend INT
DECLARE @BigDividend DECIMAL(18,0)
DECLARE @BigNumberSoFar DECIMAL(18,0)
DECLARE @Code VARCHAR( 255)
DECLARE @Mod INT
IF @TheNumber<2147483647
   BEGIN
   SELECT @Dividend = @TheNumber
   SELECT @Code=''
   WHILE @dividend >0
       BEGIN
       SELECT @Mod=@Dividend % 245
       SELECT @Dividend = @Dividend / 245
       SELECT @Code=Syllable+@Code FROM Syllables WHERE
TheIndex=@Mod+1
       END
   END
ELSE
   BEGIN
   SELECT @BigDividend = @TheNumber
   SELECT @Code=''
   WHILE @Bigdividend >0
       BEGIN
       SELECT @BigNumberSoFar=@BigDividend
```

```
        SELECT @BigDividend = FLOOR( @BigDividend / 245)
        SELECT @Mod= @BigNumberSoFar-(@BigDividend* 245)
        SELECT @Code=Syllable+@Code FROM Syllables WHERE
TheIndex=@Mod+1
        END
   END
SELECT @TheCode=@Code


GO
```

The results are hardly longer than the original numbers, and generally much easier to remember. They can then be quickly decoded.

```
   CREATE PROCEDURE Decode
/*
   Don't forget to execure CreateSyllableTable before
   use
eg
Declare @TheNumber int
Execute Decode 'BA HEBO DRACRO',@TheNumber output
Select @TheNumber
*/
@TheCode VARCHAR(100),
@TheNumber INT output
AS
DECLARE @Num INT
DECLARE @ii INT
DECLARE @iiMax INT
DECLARE @TheSyllable VARCHAR(4)
DECLARE @TheMultiplicand INT
SELECT @Num = 0
SELECT @TheCode=REPLACE (@TheCode,' ','')
SELECT @TheCode=REPLACE (@TheCode,'''',''')
SELECT @TheCode=REPLACE (@TheCode,'-','')
SELECT @TheCode=REPLACE (@TheCode,',','')
SELECT @ii=1,@iiMax=DATALENGTH(@TheCode)
WHILE @ii<@iiMax
   BEGIN
   SELECT @TheMultiplicand=TheIndex-1,
          @THeSyllable=Syllable
      FROM Syllables
      WHERE SUBSTRING(@TheCode,@ii,100) LIKE Syllable + '%'
   IF @TheMultiplicand IS NULL
       BEGIN
       PRINT 'Your code is corrupted and cannot be decoded'
       RETURN 1
       END
   SELECT @Num=(@Num*245) + @TheMultiplicand
```

www.dbooks.org

```
    SELECT @ii=@ii+DATALENGTH(@TheSyllable)
    END
SELECT @TheNumber=@Num
RETURN 0

GO
```

So far, so good: you can then give your customers nonsense words, and these can either be used as the foreign key (they will be guaranteed to be unique) or translated into the original Identity field. In situations where every object in your database, whether it be customers, products, purchases or whatever, has a unique name, then it is even better; you can just key in the nonsense word, and the system knows what you are referring to and can navigate straight there.

Testing the App

I then tested it out on an application. It worked perfectly. The users of the system found that they could get the correct invoice onscreen just by typing 'Bupris Lona' (actually 69485235), or a purchase order called 'Bifu Glofro' (actually 30586703). They could actually remember these names too, and the requirement for sticky notes and pencils plummeted. The users were delighted and decided that I was 'on their side' against the alien force of geeks in IT. In conversation. I'd hear people from accounts refer affectionately to a product or salesman as 'Gaci Skofo' or 'Bub Wivro'. They were able to remember these names and write them accurately into the application.

Excellent, I decided. I've really discovered something here. I then put a similar routine into a commercial website I was writing. At first, the business was puzzled by the system because they expect impossible numbers from computer systems just as they used to expect flashing lights from computers. Then, when I'd demonstrated what I meant, they took to the idea with enthusiasm. As I'd tested the routine at length, I felt completely confident this time.

Swearing at Customers

My undoing was this: instead of seeding the tables from a large number, over four million, as with the previous application, this time they insisted that I seeded the customer table from 1 because the CustomerIDs started from 1.

Generally, I always choose a good healthy number to start public-facing identity fields because I dislike letting anyone know how many customers we actually have, or the number of purchases there have been; but at the time, I

was too engrossed with other things to think through the potential consequences.

When I am nursing a new website, I check things every night before going to bed, just to peer around and see what is going on, and make sure that all is well. What I do is to maintain a copy of the website on a local server. I make a comparison using SQL Compare and use the application to see what has changed or been added during the day. Then I re-synchronise. When a website is just starting out, one can see almost at a glance what has happened during the day and one can pick up all sorts of problems that way, before they become a crisis.

All seemed well this particular night. We'd done good business and had got to customer no 4660. I sleepily looked at the email message queue to see the message go out for this customer's purchase:

'*Dear Mrs xxxxx*

Thank you for purchasing from the Kamakaze Laxative Company (Not the real name I hasten to add). Your customer ID is Fuca and your password…. (blah blah)'

Suddenly, I was jolted awake. Was I seeing this correctly, or had I fallen asleep at the keyboard and was suffering a ghastly dream. The computer had suddenly been possessed with 'Gilles de la Tourette syndrome' and was sending obscene words to the customer!

No! In a flash, I realised that it was my fault for insisting on memorable IDs – coupled with the fact that because I'd had to start at 1, the names were very short. I managed to stop the message going out and sat back in my chair in relief. Then I thought to myself 'I wonder what else we've sent out?'

The result was not a pretty sight. The English language has a number of four-letter words that are entirely innocent but look awful. For some reason, the eye seems to convert them to the nearest politically inappropriate word. Customer 1001 had been sent an Email informing him that his user ID was 'Buger'. Customer 4415 had been sent an email assigning the name 'Foca'.

If I hadn't spotted the problem when I did, there were a range of offensive words that might have gone out. Heaven only knows what would have happened when we got to N. I did not sleep well that night.

In the morning, I went to see the Boss, with the proverbial wet newspaper down the back of the trousers. *(Ed: aged Boarding-School joke.)* I told him we'd assigned the userID 'Foca' to one of his customers. He pulled a face and got on the phone to the customer. A lady answered. She was very considerate but said that she had been rather surprised to receive the ID. 'It is all right you

www.dbooks.org

people swearing like that in the office, but we really don't appreciate it in the home' she laughed. My boss was most charming to her; it was a revelation, as it was a side to his character I hadn't seen.

I hurriedly reset the identity fields to a higher number, and made the routines more complex so that they couldn't come up with the commonest offensive syllables. I didn't know many of them myself, despite having gone to a boarding school, but a half hour in Dispatch gave me an encycopaedic knowledge, which I then used to cleanse the routine.

The system soldiers on, cured of its propensities for bad language, and still much appreciated by the users. The staff of Dispatch still talk of the strange guy from IT who rushed around with a notebook, excitedly recording all their foulest language. The incident has made me even more obsessional about 'nursing' my websites when they are newly launched because computer systems can fail in unexpected ways.

Most of all, however, I am far more cautious of my own big ideas and less eager to bully them through to implementation.

Section IV: Hiring, Firing and other acts of Villainy

In his murky tales of IT recruitment, Phil presents us with a bleakly comic world where applicants falsify their CVs and recruiters sometimes take advantage of their position. Between the two are the recruitment agencies, roundly condemned to Dante's inferno for crimes against innocent contractors.

Phil has been both candidate and interviewer and plucks stories from his experience to illustrate the potential pitfalls of the process. Phil himself shrinks from giving advice on how to be a successful candidate; he claims that somehow he has managed his career despite failing almost every interview he has ever had. [TD]

www.dbooks.org

Bogus resumes and unblushing lies: navigating the database hiring waters

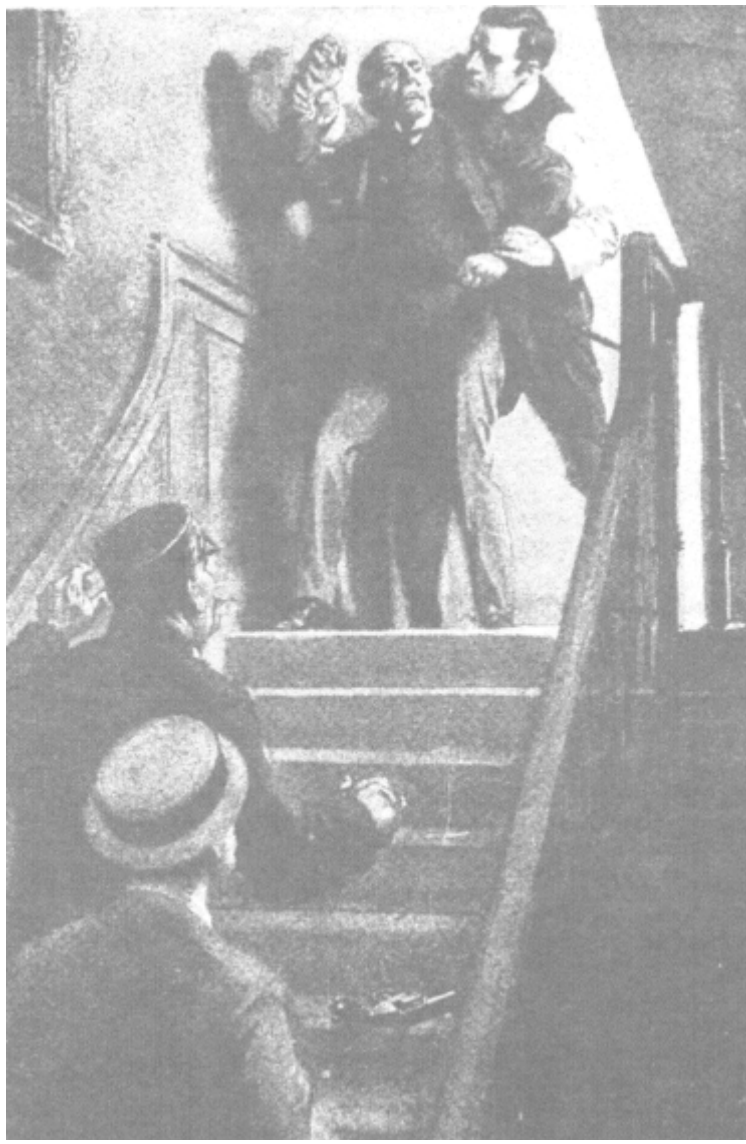
I have mixed feelings about selecting a team for a development project. I've been so long in the industry and have played so many parts – from blushing young novice to hard-bitten contractor to harassed employer – that, in theory, I should find the process a simple, straightforward affair. In selecting the right people to work with though, I've learned that the more I know, the more I know what I don't know.

One problem with selecting candidates to undertake an IT role is that so many of them write bogus resumes, and then during the interview lie unblushingly about their skills and experiences. In a well-ordered universe, all references would be contacted, qualifications checked, and previous employers phoned to check the story, but I have yet to inhabit this dream world. And even with these precautions, errors are made.

Making false statements regarding qualifications on a resume and thereby obtaining a job is a serious criminal offence. I've only heard of prosecutions against teachers or doctors, but if the police were to turn their attention to the IT industry, our prisons would be overflowing. Quite often, when employment agencies call about a job they want me to apply for, I'm asked to rewrite my resume so it fits the job requirements more closely. Occasionally, they want to do it for me.

The unworthy Wunderkind

I once got a job as a SQL Server developer for a telecommunications company. The pay was good, and it was a restful job after the helter-skelter of the dot-com boom. A month after I started, the IT director rushed in excitedly, holding a resume, saying he had just interviewed, and hired, a most excellent fellow to be my team leader. Would I move my desk over to give him a cubicle commensurate with his qualifications? The name sounded vaguely familiar, but I shrugged and thought fondly of my hourly rate – the perfect panacea for stress in any contractor.



Experienced programmer required. Must have extensive knowledge of programming in T-SQL, Visual Basic and Croydon.

The wunderkind arrived a fortnight later. I recognized him immediately. I'd recently employed him as a C++ programmer, before his reincarnation as a database expert. He had, in fact, been a mediocre C++ programmer with a poor grasp of databases. Then he emerged, like a moth from its cocoon, as a SQL Server expert. For three months I stared at him like Macbeth's ghost of Banquo

www.dbooks.org

as he messed up catastrophically. I had, of course, told the IT director, who assumed I was motivated by professional jealousy.

Programming in Croydon



Have you ever programmed in Croydon?

In another position, I interviewed a programmer for some tricky database work that required interfacing with a variety of languages. Was he familiar with Pascal, I asked? Had he used Visual Basic? Having worked my way through the list of languages and been assured with great effusions of sincerity that he had programmed in all of them, I moved on to other questions.

Our office was located in an out-of-the-way borough of London, and, with my radar on alert from his overenthusiastic responses, I thought to ask the candidate if he minded programming in Croydon, which was, of course, the name of our town. To my surprise, he burst into a speech about his expertise with the Croydon language!

To make matters worse, when employment agencies get wind of an opening in the IT field, there's often a feeding frenzy and the hiring manager is called repeatedly and bombarded with resumes. Over the past 20 years, I have known a good number of agents, some of whom are honest and check the resumes of their candidates. To these I send Christmas cards every year. Three extra cards is not a great expense, after all.

Some candidates go too far. One resume I reviewed stated that the applicant had two PhDs and had worked for IBM, Microsoft and a host of other large

employers. I was intrigued enough to interview the man, at which time it was immediately apparent that his resume was a complete fabrication. He wasn't old enough to have attained his alleged qualifications, let alone his work experience. His ignorance of the elementary principles of database theory was startling as well.

Questions from a wily nincompoop

After countless experiences like this, I determined that the way to weed out the poor candidates was to ask some searching technical questions. I have lost count of the times I have suffered these myself, and I know only too well that one's brain can be reduced to jelly when having to remember a few facts that are so familiar that, outside of an interview setting, they are like remembering the names of one's parents.

It's monstrous to confront someone with an examination without warning, like some human resource departments did in the 1990s with their intelligence and personality tests. I always warn candidates, via their agents, that they may be asked a few relevant technical questions, but that these would not take too long.

When conducting an interview, I generally adopt the persona of a kindly, middle-aged nincompoop. I gaze at the candidate with the bonhomie one imagines Santa would adopt. Other employers adopt a sullen approach, but I try not to upset the candidate just in case he subsequently lies himself into a job from which he might one day interview me. It has happened. The following approach never seems to have caused resentment in a candidate.

The first part of the interview consists of a rather optimistic account of the company and the pleasure one would derive from working for it. After this softening-up exercise, I gaze at the candidate benignly and ask how he would rate his SQL Server skills. If I have done the first part of the interview correctly, he will tell me his skills are superb.

"Oh, good, so you wouldn't mind answering a few simple questions about SQL Server?" I ask, with a kindly, paternalistic smile. "Nothing particularly technical, you understand, but I would like to see how you would approach a simple problem in any of the major SQL databases."

"Of course I wouldn't," the candidate inevitably replies, in anticipation of being tickled with a feather duster.

The metaphorical baseball bat that I then use consists of a number of simple questions that anyone who had done serious work in SQL Server would easily tackle but which immediately seem to sort the wheat from the chaff.. In no

www.dbooks.org

particular order, below is a small selection of the questions I use, just to give the feel for what they are like.

You have two tables of identical structure, with some identical entries and some different entries. How would you list out the rows from one table that were not contained in the other? How might you list out all entries in either table that were not common to both tables?

Not hard, eh? There are several neat ways of doing this, all of which are valid. I'm pleased when the candidate gets the first part of the question right, and overjoyed if he gets both. More often I hear replies such as: "One would never have two identically structured tables in a database," or "It can't be done."

Imagine you are in charge of a database that has a customer table with an identity field used as a primary key. You find out that this table has duplicate entries. How would you go about finding them? What strategy would you use for eliminating them? What might you need to watch out for?

A good candidate will rattle on about the various tactics that could be adopted and the pitfalls of any rash attempt at de-duplication. It must be difficult to have had experience at the sharp end of commercial database work without being faced with the task of mopping up. I get a little flush of pleasure if the prospective candidate mentions the possibility of using the 'group by' clause.

You are asked to produce an accounting report that lists credits and debits in each row on a number of customer accounts, and requires a column that gives a running total for that particular account holder after the transaction. How would you tackle that?

It is always nice to hear the words 'correlated subquery' in the answer, but that's not often the case. Maybe simple financial accounting skills are not taught any more. But even if the programmer is not familiar with subqueries, it is fascinating to see how he wrestles with the problem.

In the interview, I spell out the problem in more detail, and often pull out the report so the candidate can see what I'm talking about. Many times I'm told with complete assurance that it simply isn't possible to do what I've requested in SQL, and I'm given an elaborate account of creating a Java module to do it. How's that for insight into the candidate's work?

There's nothing too technical in my questions. A smart database developer will do as much as he can without specific details of a particular database system or version, as this general knowledge has a long shelf life. It's more important to know where to get the information to do the job in the shortest possible time.

Also important is an appreciation of set theory, an understanding of SQL, and insight into non-procedural programming. You would weep to know how few candidates who claim to be professional database programmers even understand the questions, let alone come up with answers. If they can tackle one question, they can generally tackle them all. And even if one question is, by a quirk of past experience, unfamiliar to them, they will come up with wonderful solutions that illustrate their essential understanding.

Works well with others?

Before one runs away with the idea that these questions represent the royal road to the ideal database developer, remember that knowledge is nothing if the candidate can't work with the team.

At one multinational company I worked for, the resume and erudition of a new Sybase programmer dazzled the IT director. He seemed wonderful, and at least technically, he was.

Soon after he had been shown his desk in our crowded, open-plan IT office, he began laughing out loud sporadically and inappropriately while reviewing code. It wasn't my code, so I didn't mind. But then his strange mannerisms began, complete with bizarre, jerky arm movements. His digestion seemed to be shot to pieces, and foul air issued noisily from him. Work in the department slowed to a crawl as every neck craned to see what he would do next.

I tried to fit the man into the team. He confided in me that fools and idiots surrounded him. He mercilessly criticized the work of his colleagues. He definitely lacked personal skills. Finally, after he had cleared an impressive space around his cubicle, his indignation at the incompetence of his co-workers proved too much for his self-restraint.

"Look at this code," he spluttered. "It is outrageous! I shall go and speak to the IT director!"

"Hmm. Not a good idea."

"He must be made aware!" he said firmly and with great zeal. So off he went. The IT director made it clear to me that the man had to go. Shortly thereafter, we reclaimed our space and settled back to our former mundane harmony.

There is something to be said for top-notch technical skills on the development team, but only when combined with an affable personality and appropriate social skills.

www.dbooks.org

IT Agencies and the Devil

The hottest places in hell are reserved for those who in times of great moral crises maintain their neutrality. The second hottest are reserved for IT agents. Harvey Furtz has his revenge.

Dante was being shown around Purgatory by the Devil, who was explaining the various fates awaiting the souls of the sinful who had failed to atone. After viewing a range of horrendous ways of punishing the wicked, they came to a large pool filled with liquid manure in which a host of people were floating. "Ah," said the Devil, noting Dante's curiosity, "the IT pool is a new feature. The level of immersion of each IT professional is proportional to the extent of their sinning during their lives."

Dante recognised many prominent executives and senior managers, some up to their waists, some up to their necks. There were a host of spammers, salesmen and plagiarisers, up to their ears. Various development teams floated disconsolately around along with technical authors, business analysts, trainers, engineers and so on. In fact, it was pretty crowded. Dante couldn't help noticing, however, a large group of IT Agents apparently standing on the pool surface with hardly their polished shoes immersed in the mire, chatting imperturbably amongst themselves or taking calls on their mobiles.

Dante drew the Devil's attention to this group, and expressed his surprise that they weren't up to their eyebrows in the mire. "Ah, yes," said the Devil, "a bit of a problem, I'll admit, but even here they are standing on the heads of the contract programmers."

How to start your own IT agency

Whenever I worry about how I shall earn a crust when my mental powers start failing, I always console myself with the thought that I can go and start an IT employment agency. Many before me have done so. It is remarkably easy, as the profession of 'Employment Agent' seems to be almost free of regulation.

There are only really two steps involved:

1. Gather CVs

The best way of accumulating CVs used to be to advertise entirely fictitious jobs in the IT press, something like this:

'A Developer with good interpersonal skills is required. No qualifications necessary – all training provided. Highly competitive salary. An excellent employer with offices throughout the South East of England'.

I've never met anyone who doubted the quality of his own interpersonal skills, so the agency should be swamped with CVs from hopefuls. Nowadays, of course, you can get as many CVs as you want just by buying them from one of the major job sites. Nevertheless, it is always best to do a few 'fishing trips' so as to build up a pool of fresh potential candidates.

These adverts are a nuisance because the people who don't realise that they are fake subsequently acquire completely unrealistic expectations about potential salaries in the industry, and of the effort required to find a decent job. Unfortunately, this practice still goes on; the same bland adverts promising employment nirvana keep reappearing, with the wording changed only slightly.

2. Find and contact companies that are recruiting

One might assume that the easiest method is to poach the client list of other agencies by poaching their agents. Agents seem to regard the contacts they make during their salaried employment as their own personal property. To expect otherwise would be like expecting crocodiles to be vegetarian. The worst problem with this approach is that an agent who would do that to their previous employer is likely to pull the same stunt on you. Also it is unnecessarily expensive.

There are much easier and cheaper ways of building the client list. The 'Employment History' sections of the CVs garnered from your spoof adverts provide you with a directory of potential employers. Now all you need is some contacts within these companies. The 'Referees' sections on the same CVs are a joy in this regard: a ready made list of people (usually managers) to whom you can sell your services! (If you think I'm being wicked and cynical, then plant a CV on one of the major job sites quoting, as a referee, a spoof person with a real contact phone number in a registered company. Then sit back and see how many phone calls that number gets from agents touting for custom).

Still need more contacts? Then, once again, your potential job candidate will often prove an excellent, if unwitting, accomplice. The technique goes as follows: you phone up each candidate and tell them what magnificent CVs they have and how hopeful you are of getting them one of the splendid jobs you

www.dbooks.org

have in the offing. You then go through each candidate's recent jobs saying something like "Ah, I see you worked for the Kamikaze Laxative Company. I used to know the head of IT there … Bob … what's his name? Err, Bob Weasel. Is he still there?" The candidate, soothed into a state of placidity by the unexpected compliments about his CV, generally lets slip "No, it is Arthur Stoat now." He may even provide you with a phone number although failing that there's always the company switchboard …

The nefarious deeds of IT agencies

To anyone who has not experienced IT agencies, my antagonism towards them might seem unreasonable. Ask any agency and they will tell you, with tears rolling down their cheeks, of the beastliness and duplicity of IT staff. However, the perfidiousness and greed of agencies is a topic of conversation that seems to unite all experienced programmers. I once worked for six years as an on-site contractor. The agency took 22% of my salary. They contacted me once in all that time, and that was to complain that I hadn't sent my time-sheets in quickly enough. When I left the job it was as though I had never existed. I never heard from them again.



Agents with morals of ferrets …

I have encountered several agencies whose staff appear to have the morals of ferrets. Several times, I have had agents try to persuade me to alter my CV to

incorporate skills and experience that I don't possess. "It's just to bring out those aspects of your CV that are important to the client," they explain soothingly. In a couple of cases, they actually altered my CV themselves before passing it on to the client. Unfortunately, I only discovered this when the interviewer homed in on a golden nugget of my fictional IT experience.

However, it's not all 'mutual loathing': on occasion, the relationship between agency and job seeker can be surprisingly synergistic. I once did a contract job for a company that involved building a complex SQL Server reporting system for a Telecomms Switch. After a year had passed, I'd done everything necessary, and arranged with the IT department to recruit a permanent member of staff to maintain the system. Although I favoured a rather boring candidate who had the required skills, an alternative candidate was suddenly presented by one of the agencies. He was bright, talked the talk, had polished shoes, a suit and good hair. He had everything, in essence, to attract the typical IT manager. To me, he seemed too good to be true. I couldn't work out why the job would attract him. In real life one doesn't get candidates like that. However, the suit and the good hair won the day and I was over-ruled.

Once this splendid fellow had settled in, I started the handover process. Although he was amiable, he showed no more than a polite interest in the technology or application. He would often sneak off to do deals on his mobile. It was all very hard work. Eventually, just before I left, I challenged him as to what he was really up to. Surprisingly, he offered to tell me on condition I was sworn to secrecy. For every candidate the agency placed, they got a percentage of his or her first year salary as a 'finding fee', on condition that the candidate stayed in the post for at least three months.

Having discovered this, along with his own talent for interviews and for impressing IT managers, he had negotiated an arrangement with the agency whereby, over and over again, they submitted him as a candidate. He stayed in each job for exactly three months, at which point the agency split the finding fee with him. I was amazed. I ran the calculations through a spreadsheet: it was a good living. Furthermore, the lack of communication between IT departments is such that he could happily go on working the scam for years and years without detection. Naturally, he left the company a month after I did.

I don't know if there is a name within the industry for this agency trick, but I suspect it ought to be termed 'the tethered goat', were it not for the fact that the original goat was often sacrificed.

www.dbooks.org

If you have to deal with the devil ...

Many have tried to bypass the need for agencies, without great success. They seem to be a necessary evil, much like wasps in summer. However, I do have a couple of suggestions for how to make the best of a bad lot.

I stumbled over the following excellent way of at least ensuring a more even playing field: I designed and built a SQL Server Database that trawled the Internet job-sites on a daily basis, scraping off all the job adverts, and storing them all in a searchable form. It proved an entirely useless way of finding the ideal job, but an unexpectedly good way of freaking out agents. As I got better at doing complex aggregate reporting on the agencies, contact names at a given agency, job descriptions, locations and salaries etc, I was able to tell agents, when they phoned up, exactly who their staff were, where they were recruiting, what their spoof adverts were, when they published them, and so on. One could give them the names of the people at their agencies and even make a guess when they joined and left the company.

This technique offers a delightful way of frightening agents into being reasonable and amenable, particularly if you are the manager who is doing the recruiting. On several occasions, I managed to negotiate their rate downwards before they regained their emotional balance.

My second 'getting even' technique is unlikely to change the behaviour or attitude of agencies in any appreciable way, but has brought much joy to many a bored IT department. The opportunity arises when you detect that an agent is phoning you up merely to determine who the IT manager was at your previous company, so he can phone him and tout for custom. A trick I have successfully worked in the past is to offer an invented character called Leonard Fuchs, or Harvey Fertz. It helps if one leads up to this with some convincing corroborative material, and to tantalise the caller with the extent of recruiting going on in the company. The effectiveness of this trick depends on warning the agent that Leonard (or Harvey) is sensitive to the way his name is pronounced. You must get the agent to pronounce it properly. Then, you simply wait for the agent to phone the company. The trick is always funniest when one can tip off the receptionist at the company concerned. The resulting recorded phone conversations, though probably illegal, can be highly entertaining.



Harvey Furtz is sensitive about his name

www.dbooks.org

The Stepford Geeks

If you expose your ideas to a charlatan, don't be surprised if they are stolen.

I was getting stuck into writing an article on interviews for IT jobs the other day. You know the sort of thing: what to wear, how to prepare your self, what to say when asked the typical questions, and the sort of questions to ask at the end. It suddenly struck me that I had failed in almost every interview I had been on, and therefore had no right to pass on any tips. I once even failed an interview for a job I was already doing, and doing to everyone's satisfaction. (Perhaps one day I should write up that particular ghastly experience.) On another occasion, I went for an interview with a bank in the Docklands, and thought I'd done pretty well. When I phoned the agency the following day, they passed on the feedback they'd received, which was that, whilst I seemed to know my stuff, they felt that I was the most arrogant and conceited person they'd ever interviewed, and they were sure that the department would be in open revolt if I were appointed to the job. "Does that mean I didn't get the job?" I replied, jokingly.

After several similar experiences I began to believe in a God whose primary mission was to teach mankind humility even if it killed them, which it invariably did.

By far the most curious failure I ever had was with a large insurance company in the Midlands of the UK. Database work was a bit slack at the time and I had morphed into an expert on 'Business R-engineering', which was the current craze. The agency told me that this company felt it was missing out on this wonderful re-engineering lark that they'd read so much about, and wished to recruit an expert to their team to guide the process.

An agency phoned me up asking if I'd like to go for an interview with this company, as my CV seemed impressive (all CVs 'seem impressive' when agencies want you to do something for them). So it was that, looking every inch the part in standard sober charcoal-grey suit, with subtle pinstripes and black shoes, I strode into the IT department for the interview.

I was greeted by a very pleasant young chap, who gave me a preliminary interview and, when I started to explain about my experiences with the whole

process of 'Business Re-engineering' excused himself to bring a couple of his colleagues into the room.

Dammit, Stepford Geeks. By some strange Darwinian process, young IT people have evolved a curious sameness, like lizards in the Galapagos. They beamed across the table at me in unison. They were pleasant and very keen to find out more about my Business Re-engineering expertise.

I love a receptive audience, and all my caution flew out the window as I launched into an exposition on the entire IT involvement in a Business Re-engineering. As I had recently written a book-length training document on the subject it wasn't too hard.

I was in full song, and going down strong. Two more colleagues came in after another quarter hour and the five scribbled away furiously. My masterly exposition complete, they launched into a series of penetrating questions that I fielded like WG Grace at the crease. The whiteboards in the room were soon filled with process diagrams as, with my face flushed with the enthusiasm that only comes with an appreciative audience, I gave them the benefit of my knowledge with both barrels. I remember vividly that, when I'd finished, one of them began to clap before being shushed by a colleague.

After the interview they lined up by my car, and shook my hand warmly whilst mustering a heartfelt thanks to me for coming for interview.

The following day, the agency phoned.

"How do you think it went?"

"Well, that's tempting fate; but I think I went down pretty well, actually."

"Good, but it's odd I've heard nothing yet."

"Early days yet. They're probably wondering if they can afford me."

After a fortnight, the agency phoned in a rather off-hand way to say they weren't going to proceed with considering my candidacy. Before bounding irrepressibly after the next job, I must admit to being a bit puzzled by their change of attitude

I might never have known the end of the story had it not been for the strange way that the flotsam of the IT industry seems to circulate just like it does in the Sargasso Sea. The insurance company eventually got taken over, or went bust, and its IT people were scattered to the four winds.

At that time, I was a senior manager in London, responsible for recruiting staff. One day, as I sat down to interview a candidate, I spotted the name of this insurance company on the CV. I looked up, and stared at the candidate long and hard. Dammit, it was one of the Stepford Geeks who had interviewed me

www.dbooks.org

on that day. After exchanging pleasantries, I couldn't help but ask why on earth they'd rejected me after seeming so keen to employ me on the day. I was half expecting to hear more about my conceited manner, but instead he said,

"We'd never had any intention of employing you then. We had no budget to employ you."

"Why waste a day of my life then?!"

"Well, the company had raised the idea of a Business Re-engineering project with us and so we felt the best way of coming up to speed was to interview some experts in the field. We had no budget to actually hire these experts, you see, so it seemed a good idea."

"So I just had my brains sucked out then …"

"Well in a way. After we'd discussed and written up what you told us, which was brilliant, we went back to the company and managed to discourage them from going forward with the Business Re-engineering Project at all. So you would never have been needed."

"But, but … had you ... err ... considered the ethical dimension of this? What about my wasted day?"

He looked genuinely puzzled, and blinked at me for a few seconds in bewilderment.

"Well, when the IT director found out the whole story he was very shocked and told us to damned well put it all right. We felt pretty bad about it I can tell you. We phoned the agency, and explained that it wasn't your fault that we couldn't make the appointment, but that you'd been so helpful that we'd pay you two days consultancy fee plus travelling expenses. The Agency billed us and we paid. A few weeks later we asked the agency for two weeks consultancy from you but they said you'd got another job."

Needless to say, the agency had long disappeared when I tried to contact them later that afternoon.

www.dbooks.org

The Terror of Technical Tests

Phil has a fear of technical tests. Not because of his lack of IT knowledge, more a humble recognition of how much he still doesn't know after all these years. He laments an IT industry that has got in such a mess that unqualified and untrained people can bullshit their way into a position of responsibility armed only with a bogus CV and a string of false references.

After all the years I've spent working with databases, I am continually shocked by how little I know. The power and facilities of relational databases have increased enormously, and we struggle to keep up. One has, of necessity, to spend an increasing portion of the working day in retraining. To a hard-pressed project manager, time spent on familiarising oneself with current ideas and practices looks like wasted time because it can't be fitted logically into the chart. He therefore discourages it. In consequence, one can waste time creating a procedure that can be done much simpler with a new feature or third-party product, simply because one hasn't had the time to check it out.

It is the knowledge about my lack of knowledge that gives me a terror of Technical Tests.

I was recently persuaded, against my better judgement, to apply for a well-paid job as a DBA/Developer. (They pronounce it as deebeeaye stroke developer, which sounds like a closet office liaison). I was sat down in a room, given a brief synopsis of the company, and then given a technical test.

My initial horror, which stems from the fact that the more one knows about SQL Server and relational Databases, the more aware you are of your ignorance, turned to delight. I was back in primary school. What is Normalisation? What is the use of an index? How would you pass output variables via ODBC? What is a transaction? Eh?

It was like waking up after a knock on the head, and the doctor asking you the name of the president of the United States to check that you are Compos Mentis. After rattling through the answers, I asked the interviewer what sort of creature wouldn't know the answers to this sort of question. Bleakly, he replied that they'd already interviewed five candidates for this specialist role and two of

them hadn't even known what the normalisation process was. I was the only candidate so far who had even understood all the questions, let alone answered them all correctly (which I had, I hasten to add).

So, it seems that agencies are putting forward candidates for important roles who profess to have database skills and yet are entirely ignorant of even the basics.

I then sat a second, practical, test, using a sample database, that involved writing a stored procedure or two and a few SQL expressions. This was harder, but only because there was a fundamental mistake with the design of the database that confused the quantity of items purchased with the amount of money charged. After I completed the tasks I'd been given, I pointed it out to the interviewer. He was amazed. It turned out that none of the other candidates had got far enough with the task to trip over the error.

After the tests, the interviewer filled me in on the nature of the role. He was a chap of extraordinary charm and tact, who put a fine spin on the problems they had, but it was immediately obvious that the company was in considerable peril from a database that was being operated recklessly and in breach of all financial and procedural guidelines. Could nothing be done? Sorry, our hands are tied. All we can do is to support and maintain what exists. Like the legendary reporter, I made my excuses and left. I'd rather be poor, but sleep at night.

As I stumbled back out into the light of the car-park, I wondered, for the life of me, how the IT industry had got in such a mess that unqualified and untrained people were in responsible positions within organisations all over the country, managing the databases that are the lifeblood of the enterprise. What would happen if the same state of affairs infected Surgery, so that there was a chance of being operated on by someone whose only qualification was that he'd cut open his teddy bear as a child, using his 'My Little Doctor' kit, but managed to bullshit his way into a job armed with a bogus CV and a string of false references.

www.dbooks.org

The Interview with the Psychometric Test

He knew there would be trouble the moment she walked in the room. Like many IT people, Phil has an instinctive aversion to HR people and little but disdain for the 'psychometric test'.

The HR person looked briskly at me and waddled importantly to the desk. I'd come for an interview with a firm in Cambridge running an IT consultancy, from an office quite near where Red Gate now is based. They said in the advert that they wanted a 'Database expert with Oracle, Sybase and SQL Server experience'. I'd mistakenly thought I'd talk about consultancy, and IT issues, eye-to-eye with a like-minded technologist. Instead, I faced this creature.

After some anodyne pleasantries, she came to the point

"Right. First, I'd like you to complete this test. You will have three-quarters of an hour."

She banged the form in front of me. I glanced at it.

"This is an NFER test for schools, designed to assess the academic skills of fourteen year olds." I said.

She tut-tutted. "This is a standard test that we give to all candidates for jobs here. We are familiar with the results and so it is invaluable in our candidate-selection process."

"Impossible. This is a test standardised on a school population, and designed to be administered in a certain way by a trained teacher. It is used to try to predict academic progress of pupils up to the age of 16, and it was never much good at that. It will tell you absolutely nothing about how effective someone will be in the workplace."

There was a strained silence while she administered her best 'we've got a right smartarse here' look. I took advantage of the lull.

"Another problem is that I know all the answers, since I've had to administer and mark the wretched thing in the past, when I was a teacher. You have

additional problems in the ethics of giving someone a psychometric test without prior notice or proper consent. It is actually not permitted in the NHS."

There was a pause as she tried to come to grips with this. "I don't agree," she said, finally, "All our candidates are happy to take the test and we find the results very useful."

I gave up. "OK," I said. "I'll take the test." She looked intensely relieved, set her stopwatch and left the room.

I fell into a reverie. When I worked so hard to get my Masters, I thought to myself that it would be a generally recognised badge of educational achievement. How times change. Why did I bother? Why didn't I just backpack around the world for a couple of years, and then create a completely bogus CV claiming experience in all the hot technologies. I'd seen it work many times.

I wondered if I could show her how silly this was. I looked down at the once-familiar test. I never thought I'd see it ever again. Within fifteen minutes, I'd filled it all in, with perfect answers since I'd done so many markings of the wretched test. I was gazing out of the window, and meditating on the comedy of human frailty when she minced in after three-quarters of an hour to pick up the completed test.

I was then taken to another room where a manager started to interview me. As the interview progressed, I became increasingly disillusioned and disappointed. He gave the impression that the entire business was run by the rule book, and that the consultants were mere cannon fodder in a large consultant army. This was no place of contentment for a free-thinking technologist. The HR lady brought the test results into the room, glancing at me with a new-found respect. The manager did a double-take, immediately followed by a complete u-turn from his previous arrogant demeanour. I had, of course, achieved a perfect score. One doesn't forget the correct answers after all that marking.

For the remainder of the interview he became repulsively ingratiating, until the moment came …

"… and why, in particular, would you like to join our company?" Wide smile.

This is it. "Because of my experiences this morning, I'm afraid that I don't."

I rose to leave, shook him by the hand and was soon, flooded with relief, in the fresh air once more.

"Did you have a good interview?" asked my wife brightly, as I returned home.

www.dbooks.org

Brown Shoes Don't Make It

'... Be a loyal plastic robot
For a world that doesn't care
Smile at every *ugly*
Shine on your shoes and cut your hair
Be a jerk and go to work Be a jerk and go to work
Be a jerk and go to work Be a jerk and go to work
Do your job, and do it right
Life's a ball! (TV tonight!)
Do you love it, do you hate it?
There it is, the way you made it (WOOOooow) '

-- *from Frank Zappa's song 'Brown Shoes Don't make it'*

I once made the terrible mistake of going to an interview for an IT job at the head office of a bank in London whilst improperly dressed. I forget what the job was exactly; some IT management role that was customer-facing (in other words, it meant talking to people other than geeks).

I wore brown shoes. OK, laugh, readers, but how was I to know? What book, with a name, I imagine, like 'Knock 'em Dead at Interview', ever warns you never to wear brown shoes in front of Bankers. I suspect it is just too obvious to be stated amongst the *cognoscenti.*

I was doing quite well, I thought. Dressed in my regulation pinstripe suit, white shirt and sober tie (slight hint at a good regiment), I was explaining the finer points of Catalyst, or some other development methodology, when I absent-mindedly crossed my legs. In consequence, a highly-polished brown shoe swung into view of the interviewers. Two of the three saw the offending shoe immediately and stared at it, as if hypnotized. I knew immediately that I'd lost two-thirds of my audience. All their attention was focused on my well-polished brogues. The third interviewer was still following my splendid train of thought until nudged, by one of the others, into looking at the offensive footwear. After that, it was downhill all the way.

After the interview, I sat miserably on a bench in a nearby park while my agent gave me the interview feedback. She reproached me for failing to take enough care of my personal appearance. It was as if I'd attended the interview in a gorilla suit. Apparently, they would have considered me if only I had 'turned out properly' but, in the circumstances, were forced to assume that I had 'little or no idea of what was required of someone who will be customer-facing

in this enterprise'. I imagined the three interviewers shaking their heads in disbelief whilst debriefing themselves over a soothing cup of coffee.

I cursed my naivety in assuming that my technical abilities would take precedence over my fashion sense. I must have looked even more dejected than I felt, because she suddenly stopped in mid-flow and promised to buy me a nice pair of regulation pattern black shoes, perfect for IT work in a bank. This she then did, explaining that she was only temporarily working as an agent and hadn't yet been infected with that callous indifference to their job candidates that is so prevalent in the profession.

I have to say that it worked like a charm: within days I was 'aceing' job interviews, knocking 'em dead. Occasionally, I would deliberately cross my legs just so they could admire how closely my footwear conformed to the sartorial standard. 'Hmm - footwear good, must be sound' I could almost hear them think. I still have the original pair as a sentimental souvenir and, when shoe shopping, I always take them with me and insist that the new pair are exactly like them.

In the city, there is much discussion about the type of shirt to wear at interview. I always advise white; you can't really go wrong. However, there is some debate over the matter because the liberal wing, rather recklessly in my opinion, will suggest that any shirt with sober vertical stripes will do. The suit, it goes without saying, should be charcoal grey, though you can try pinstriped for interviews for jobs of senior grade.

All this knowledge and expertise is nugatory when interviewing for Linux jobs. It is part of the exaggerated informality of organizations that operate a Linux policy that ties are symbols of the beast, and suits are the uniforms of the army of Baal. I once had the misfortune to turn up to an interview for a PostgresSQL job in such an organization, having come directly from an interview for a banking job. I was greeted by a Java programmer dressed in a full-length black cloak like a high-church rector, though the black skulls and deaths-head chain rather spoilt the effect, and I'm not sure if the church is supportive of plaitted hair in male priests (I later learned that he was a "Goth"). He was very charming, but I saw him glance nervously at my black shoes.

I surreptitiously took my tie off and slung the jacket over my shoulder. My interviewer and I stared at each other as a lobster and a squid might do, and decided instantly to keep the conversation on the common ground of creating distributed applications in J2EE. I suspect he had half his mind on the pleasures of chasing squealing elves through woodland at weekends, and he probably thought I was dreaming of blasting grouse out of the sky whilst sneering at the beaters. We cast our cultural differences to one side and basked in the shared excitement of nailing squealing EJBs and 'stateful session beans' as they ran for

www.dbooks.org

cover, chuckling over 'thrash-tuning antipatterns' and other oxymoronic neologisms.

This time I got the job, despite my sartorial *faux-pas*. Maybe the open source world is just more forgiving than the Microsoft one or maybe, despite himself, the interviewer just couldn't help wondering wistfully where I'd gotten those splendid shoes.

www.dbooks.org

Technical Interviews and Tests Have Got to Stop!

Instead of employing developers based on their responses to a series of trivia questions, culled from obscure facts on MSDN, you should see how good they are at Table Tennis, Table footie, or guitar.



'Technical Interviews' have got to stop. They are a disgrace to the IT profession. Two MVPs who I asked the question 'Have you ever passed a technical interview' have admitted 'Never'. I'd like more successful developers to confess their inability to remember much more than their name under the pressure of a technical interview. The most extreme geeks all have brains that blue-screen with a temporary aphasia under stress.

I have a new proposal to make. Instead of employing developers after asking them a series of trivia based from trawling obscure facts in MSDN, you should see how good they are at Table Tennis, Table footie, or guitar. I might include pool/Billiards too. I intend to justify this apparently ludicrous assertion.

There are many theories of what makes a good developer. In the 1960s, when the shortage of people with IT skills first became apparent, people were selected on their mathematical skills. This proved to be disastrous, and IT departments soon filled with strange geeky folk devoid of social skills, or any understanding of real business applications. This puzzled the recruiters who consulted the psychologists. The psychologists pondered over this and, in the fashion of the time, came up with a set of problem-solving tests based on Hans Eysenck's intelligence tests. In Britain, these seemed so obviously appropriate that they even appeared on adverts 'If you can solve this puzzle in five minutes, call xxxxx'. Subsequent exercises in validation showed that they were no more successful than chance in successfully predicting a good IT programmer. In fact, it just told you that the person was good at puzzles; it also suggested that they didn't get out enough in the evenings. The psychologists then took the more sensible approach of testing real successful IT programmers, and seeing what their skill-set consisted of. It turned out that good programmers were very articulate. Otherwise, they had no obvious dominant skill in common.

Selecting programmers who were good at language skills just didn't look right. It had no 'Face Validation', and it didn't impress the brash new profession of 'HR'. They wanted tests like they had at school. Where there is a demand, there are always people prepared to supply it, and soon a lucrative market in technical tests developed. The validation of any psychometric test of ability is a highly technical subject which I won't bore you with. Please just remember that, because these many tests of technical competence have never been scientifically validated, they are no more effective in selecting the good candidates than testing to see if the applicant can sing 'Somewhere over the rainbow' in a high voice. The classic Microsoft interview questions, such as 'Why are manhole covers round?' shouldn't be used until you can prove that successful programmers believe (wrongly, it turns out) that it is to stop the covers dropping into the hole, more frequently than do the duffers.

Besides fluency with spoken language, there is another essential component, a personality trait that is necessary for a good developer. This is a cussed stubbornness that won't let go of a problem until they have solved it. It is that they will to go through pain, boredom, hunger, and lack of sleep in pursuit of doing something that is almost impossible, such as playing guitar properly, solving advanced physics and mathematical problems, playing badminton or table tennis well.

The best programmers I've ever met are astonishingly alike in this respect. If they can't get something in their lives to work as well as they want, it turns into a man-machine struggle of epic proportions. If I give someone a programming puzzle to solve in an interview, the good programmers don't necessarily know the answer; the good programmers refuse to finish the interview until they've

www.dbooks.org

got the answer right. It suddenly rears up in their minds as being even more important than the interview. The whole objective of getting a job is temporarily put to one side in pursuit of a solution.

The same personality, if she, or he, finds herself, or himself, unable to play table tennis properly, or play 'tiptoe through the tulips' on the ukulele, or beat the regulars at pool, will bloody practice until she, or he, can or renders herself or himself incapacitated in the attempt.

If you think that this is an absurd theory, you will be surprised to hear that the standard American test of mathematics, which is prefaced by a rather long-winded form for name, address etc., has an almost perfect correlation between the Mathematics score and the ability to fill in the preliminary form accurately. If the applicant has the mental stamina and obstinacy to apply to fill in the preliminary form you can be certain that he or she will be good at Maths. That same cussedness turns out a good IT programmer, or a good table-tennis player, or any skill that required dogged determination.

So, if I ever have to interview you for a technical development job, have no fear of being taxed with knowledge of obscure parameters to DBCC functions you would never dream of using, or obscure behavior of Transaction rollback in linked servers. No, beyond establishing a sound ability to understand questions and solve general programming problems, I shall be there to challenge you to a game of table football, croquet, pool, or any other skill that can only be achieved by bone-headed stubbornness and practice beyond the patience of an ordinary mortal.

www.dbooks.org

Two stops short of Dagenham

A bittersweet tale of how the IT industry has changed to the point that it now tends to exclude the maverick, the eccentric – in short, the sort of person who has historically made huge contributions to technology and to the workplace.

==*==

Ed: Don't be put off by the obtuse title; here is my short explanation:

- On the District line of the London Underground Tube System, Dagenham was originally two stops away from Barking.
- 'Barking' is also a shortened version of the English colloquialism 'barking mad'.

So to suggest that a person is 'Two stops short of Dagenham' is to imply that they are 'barking' i.e. 'mad, oddball, crazy'.

==*==

Nowadays, when you see someone wandering down the corridor at work with his hand to his ear talking, apparently to himself, and laughing sporadically for no apparent reason, you know he's on his mobile phone. In the days before mobiles, one used to shrug and assume he was a programmer.

The short history of computer programming has been highly enriched by a number of deeply eccentric people. They were rarely psychotic I hasten to add, although I recall that one of the most successful of all PC applications was originally written by a talented programmer who occasionally took it into his head that he was a teapot, and would subsequently disappear for an extended holiday, to brew maybe. They were unusual, odd, eccentric people; even some of the great pillars of Microsoft's current product range started out life as the creations of maverick programmers.

I sometimes wonder why all of the great applications around today, such as spreadsheets, email, word-processing and relational databases, were invented in the seventies and eighties. Our greatest achievement in the past fifteen years seems to have been to refine the visionary approaches of a previous generation – one that cherished and valued the creative spark, and tolerated eccentricity.

Every programming team used to have its eccentrics. Society once valued their contribution highly. I've worked in development teams with characters whose saintly innocence and quirky world-views were valued. One of the best programmers I ever met was referred to, in Essex jargon, as being 'two stops short of Dagenham' (Barking). This was once no obstacle to the world of work.

When Fred, who we'd probably now tag with the "Asperger's syndrome" label, was given a computer problem to solve, he would go into an extended reverie, staring out of the window in an apparent dream state. A fortnight would pass, with him frozen in thought, muttering, and showing strange 'ritualized' idiosyncratic mannerisms, hand movements, tics and such. We paid no attention because we knew there would always be a next stage, when he would suddenly rush to the keyboard and type in the complete source code from start to finish. His code always compiled first time and one never needed to test it. It always worked flawlessly, and was brilliantly conceived. Naturally, he was useless for group work or collaborative projects, but set him a clearly-defined task and we were all happy. Fred was an oddball, but he was one of us, a team member, and we loved him. In the pub after work, he was always with us, glass in hand, beaming in slight puzzlement at the social banter.

However, times were changing and eventually the spectre of 'Human Resources', with their quasi-scientific theories, fell like a dark shadow across our happy world. Fred carried on as before, but the time came when a young feminist graduate sat at a desk nearby. She misinterpreted those ritualised 'tics' and hand-movements that signified Fred exercising his magnificent brain and wrestling with invisible database entities (she wrongly assumed he was masturbating).

Quick as a flash she phoned the 'Human Resources' department. These people, whose collective job had once been the mundane execution of personnel processes and payroll runs, leapt to their new task: weeding out the oddballs. They summoned the security men who lifted Fred bodily, one on either side, and dragged him off to 'Human Resources' for a 'disciplinary procedure', just as he had almost constructed in his marvellous brain the most beautiful error-processing module.

The entire IT department froze in horror at what had been done. 'Our Fred!' went the cry. In a rare act of cohesion and speed we had the IT management shoulder-to-shoulder with the Union in getting our Fred back. We filled the corridors with angry shouts. He soon re-emerged apparently unscathed, surrounded by some highly embarrassed and apologetic HR bosses. Although Fred was apparently un-flummoxed, for the first and only time the program that then emerged refused to compile.

www.dbooks.org

Time was against Fred and his ilk. An era that flattered itself to be 'inclusive' soon weeded out the eccentrics from corporate IT departments. I believe that with them went the creative spark and intrinsic stubbornness that is so vital to development work. One persistent myth in the IT industry is that, in development work, we need team players, not mavericks. It depends on what the team is doing. Harmony is fine for picnics and team-bonding exercises, but is not necessarily the precursor to productivity. The best development teams I've worked on embraced the whole gamut of humanity – a wild mix of cultures, sexuality, motivation, age and beliefs, spiced with maverick spirits and eccentrics. The result being that everyone was jolted out of their complacency, both professionally and personally, and learned to challenge assumptions rather than accept them. The trick is to harness the resulting energy and to share out the task to match each person's skills and temperament.

Nevertheless, 'Human Resources' practitioners persist in many enterprises, with their 'personality' screening. This is an evil practice. Many will have experienced job interviews for a technical role where one is required to engage in 'role play', or is asked silly questions like 'if you were an animal, which animal would you like to be', and so on. Any trained and qualified psychologist will tell you that the tests are bunkum, culled from Girlie magazines. Even voluntary, properly administered personality tests, validated on large populations, and given in un-stressful settings, are notoriously unreliable. When they suddenly pop up in an interview, without prior consent, and the candidate has to give the 'right' answer when a potential job is at stake, they are a gross impertinence, and probably illegal.

So cast a look around any IT development team. Typically, they will be all of about the same age, usually all-male, with very similar backgrounds and education. All good team players, supportive and wholesome, I'd guess. All well and good, but I've been privileged to meet quite a few of the people who made the great leaps forward in programming, and I suspect that the ones I've met would have got weeded out by the modern interview process. Nowadays, in this world of team players and well-adjusted, politically appropriate, vegetarian programmers, the idea of the wild and the eccentric in the midst of development activities would strike terror into many hearts.

www.dbooks.org

The March of Time

Phil, in atypically sombre mood, reflects on the built-in prejudices of the IT industry, which means most IT programmers are twenty- or thirty-something males, and a wealth of other talent is neglected and lost.

Tony was a systems analyst and programmer. For many years, he'd worked for a large insurance company in London. He'd spent all his working life there, in fact, seeing their IT systems grow from simple Cobol-based accounting systems to their current sophisticated distributed architecture. He'd grown to become a veritable repository of IT knowledge but, because he was eventually the only one who understood the legacy systems which still powered the heart of the commercial activities, he became indispensable there, and was never given much of an opportunity to get first-hand experience of the latest technologies. He was particularly interested in J2EE and EJBs. Longingly, he pored through the J2EE and Java manuals, surreptitiously creating Java-based applications when nobody was looking, whilst tending the aging accounting systems. Youngsters joined the department knowing only the Java world, whilst Tony understood the breadth of the corporate IT activities, and was relied on to provide the skills to maintain the production systems. The interesting jobs went to others. Tony occasionally shrugged regretfully, but pressed on, with the consolation that he was useful to the enterprise.

Then, Tony's mother got sick. There was nobody else to look after her. Tony asked for a sabbatical so he could be her carer, but even after decades of service, the company regretfully refused, on the reasoning that everybody else would ask for one too. He therefore resigned.

When his mother died, he decided he'd get a proper training in J2EE. He found a comprehensive course that guaranteed a job at the end of the training. The course was very easy for him, as he'd worked in IT all his life and had taken the trouble to keep up-to-date. He sailed through with top marks. However, halfway through the course, the organisers drew him to one side and told him that, in his case, they would not be able to honour any guarantee of employment at the end of the course. Tony, being a placid chap, accepted this with some puzzlement. He assumed that, because the market required skills, and he had them, that he'd be OK.

Things weren't OK. Armed with a certificate of competence in J2EE and Java, a CV that glowed, fulsome references, and a track record of steady employment, he clocked up an enormous collection of failed applications. He was called for interview twice, but wasn't chosen. Rather despairingly, he reapplied to his previous company, but they told him that the legacy systems were now gone and they had enough J2EE skills now.

Tony eventually realised that his effort was pointless and so he answered an advert for a salesman. It was a cut in salary, but Tony didn't care. Surprisingly, his new employers were appreciative of his work, pleasant, and soon promoted him to a position of responsibility.

The organisers of his course were right: He was too old to be a programmer. It was not that his skills or knowledge had diminished, just that, culturally, we seem to expect software developers to be twenty- or thirty-something males. Poor Tony was fifty-something. 'Ah well,' thought Tony, 'probably prejudice had worked in his favour in the past, so what is the point of grieving'. For Tony, being a resilient and likeable person, things ended happily with a new career that got him out from the bowels of a corporate data centre, blinking into the sunlight and fresh air, and a new career chatting to people and getting paid by an appreciative employer.

It is, perhaps, the IT industry, with its misplaced prejudices about recruitment, which is the loser.

www.dbooks.org

Fired With Enthusiasm

Phil has met more than his fair share of idlers in his IT career, mainly nestling in the crooks and crannies of large international enterprises and government departments. He calls them 'light arms': low calibre, small bore, need to be fired.

Kishor, the IT director, looked up from his notes. "We don't have many people working for us here." he said, and flung the file on his desk with some exasperation.

I had learned rapidly, in my new career as an underling, never to contradict anyone senior to me. Instead, I mutely stared out over the vast sea of cubicles in the huge open-plan office, and scratched my head in puzzlement.

"I suspect it depends on your definition of the word *many*." I suggested placatingly.

"No it effing doesn't," he spat out "it depends on your definition of the word *work*."

I had come to the IT department of a major London Bank as a refugee, having run my own IT business for some years, and I was enjoying the increased pay and tranquillity. It had, however, come as a complete surprise to find myself surrounded by members of the IT staff who didn't do a stroke of useful work all day, and became animated only when the magical hour of 5PM approached.

I'd never come across such a thing before. In the IT world of the SMEs, we were all caught up in the never-ending struggle to survive. We worked all the hours we could in order to deal with impossible workloads. I had to agree with Kishor that we had more than our fair share of idlers in the staff.

We called them the *light arms* (low calibre, small bore, needed to be fired). One such character confessed to me how he was able to catch up on his sleep while at work. His amazing talent was being able to doze off with his eyes open, whilst staring at the screen and tapping the CAPS key.

If resting after the exertions of ones night-life wasn't to your taste, then there were other diversions to while away the time. The Finance Director happened

to be looking through the small-ads in the local paper and came across a phone-number that looked curiously familiar. It was for a professional accountant. Puzzled, he phoned the number and was put through to one of his managers a few desks away who was running a firm of accountants 'on the side'.

Therefore, I sympathised with Kishor, and helped him to refine his list of the 'light arms' members of the department. It wasn't hard to do, as the programmers leaked like sieves on the subject of the 'time-servers'. However, it seemed to me a bleak and hopeless task to motivate such people, and I knew immediately how difficult it would be to sack them. I felt that the only way forward was to try to encourage these people by a mixture of flattery and job role-change.

As it happened, Kishor never asked for my advice, but I suspected he had a plan. Kishor always had a plan. For some time before I came to work for him, I'd heard of his exploits, and his cunning. I once met him in the course of business, when he first started out in IT; a young serious-minded Systems Analyst, working for a software house. He'd come a long way since.

Within a few days, the department secretary announced that some of the 'high fliers' in the department were to go on a training course at the company's expense, to learn C#. Various people were picked to go on this course, and there were dark mutterings of discontent from some of the excluded technical die-hards, mainly about how they'd had to learn the language themselves, in the evenings. Kishor told the few managers who protested to him at the choice of several 'light arms' candidates that it was a way of inspiring them to change their attitude to work, giving them a fresh start with a different technology. The C# course came and went. As I suspected, I didn't need to arrange cover for any of the participants in my department. Neither did anyone else.

Soon afterwards, the participants started to hand in their notice. The trickle increased to a rush, and it soon became apparent that they had all been head-hunted by a rival bank, The Imperial Bank (not it's real name).

Kishor conducted a stormy staff meeting soon afterwards. The gossip had swept the city that the Imperial Bank had scooped up the entire group of high-fliers from our department. Evidently, a list of the participants in the course had somehow got into the hands of the Imperial Bank. Kishor tearfully railed at the iniquity of our rivals and we all shook our heads in sorrow at the disloyalty of our staff, and the chicanery of the person who had leaked the list. We were particularly incensed that we'd trained them in valuable skills at our company's expense.

www.dbooks.org

When the extent of the duplicity became apparent, we became objects of pity in the wider world of IT. It even got a mention in one of the computer papers that we'd lost an entire 'year' of graduate entrants to the other bank.

Still mulling over this dramatic turn of events, I retreated to one of our favored City Pubs, where I was sipping a pint of fine ale, and enduring a certain amount of good-natured ribbing from one of the managers from the Imperial Bank's IT department.

It was at this exact moment that it dawned on me what Kishor had done. I suddenly recalled, with complete clarity, walking through the City streets one evening that summer, just after the training course was announced, and catching sight of Kishor in this very same establishment. The grey-suited customers spilled out into the street clutching their beer-glasses and talking loudly. In the thick of the mêlée was Kishor, surrounded by IT people. Here and there, like jackals around a campsite, I had noted the sharp attire and haircut that marked out an *IT employment agent*.



Kishor had 'accidentally' let slip to one of these IT agents the fact that he was sending all his most talented staff on a training course, knowing full well that the agent would subsequently inveigle the list from one of the more 'malleable' of the light arms candidates. From there, it was but a short hop for the list, into the hands of the agent's best client – the Imperial Bank! I smiled in awe and wonder at the subtlety of Kishor's plan, and its implications. Suddenly, the pint of Landlord Bitter in my hand, bought for me by an Imperial Bank manager whose conscience had been troubling him, tasted twice as sweet.

After a few months, of course, the IT department of the Imperial Bank realized that all was not quite as they'd hoped. They'd expected a swift recouping of the expense of their golden handshaking. It didn't happen like that and they began to detect the deadening lethargy of their new recruits. I was even drawn to one side by a senior manager at the Imperial Bank and asked what on earth our *low*-fliers were like. I merely smiled and said that so much employee productivity was bound up in the whole ethos of the workplace and our skilled management techniques.

Not only had Kishor cleared out all the dead wood from our department but he had transferred it to the rival bank in such a way as to inflict maximum expense and damage to them. We had, as well, all emerged as the innocent victims. It was at that point that I realised I would never make it to the top rank of IT managers. Something else was necessary; a talent I didn't possess. Kishor had it.

In subsequent jobs, I learned quickly that the *light arms* brigade tend to nestle in the crooks and crannies of *all* large international enterprises and government departments. In one particularly memorable case, I spent three months in a government department working on a database audit, and sharing a desk with a Systems Analyst who was running a society magazine for Rural Field Sports. He was supposed to be maintaining several 'legacy systems' but this took only the occasional five minutes out of his busy life amongst the ferrets, grouse and salmon.

It's at these times that I remember the lessons I learned from Kishor. His talent, and his encyclopaedic knowledge of the goings-on in all the City of London IT departments, had seen him rise from the shy, serious programmer I'd first known, to the youngest IT director ever in the City of London. It was comforting to me to reflect that at least some of the knowledge that had driven him there had been gained in one of our beloved City of London pubs, where IT people congregated, and where Kishor gathered intelligence over a glass of orange juice.

www.dbooks.org

The Whipping Boy



Whenever harsh words are heading in Phil's direction, he does what all seasoned contractors do in the circumstances, which is to calculate, on the hourly rate, how much he is being paid to be there in the room listening to the tirade.

When working in IT, one often makes erroneous assumptions about one's role – assumptions that are not borne out by experience. Like a dog that mistakenly believes his true role in life is to rush up and down the garden fence barking furiously, I have always foolishly assumed my true role in an organisation is to develop IT systems that meet their needs.

One experience particularly sticks in my memory. I started work at a small start-up telecommunications company, at a time when the data-side of the enterprise was seriously deranged. The company had been a startling success and usage of their service for making phone-calls had shot up. It had all been too sudden, and the small, inexperienced IT department were bewildered by the demands being placed on them. They were producing invoices for their

customers but little else. Corporate management was defined by the sort of instinctive, high adrenaline, caffeine-fuelled style that I dub the 'Smouldering Underpants' technique.

The IT director was a charming, urbane Indian who read poetry constantly, and smiled sweetly whenever possible, which was not often. When I first met him, I was slightly puzzled by the way his eyes lit up as I walked into the interview room. I have to admit that I assumed at the time that he had somehow instinctively recognised my splendid abilities where so many had not. Now, I'm not so sure.

On my appointment as 'team leader' in the department, I was shown into the open-plan office. The team, who were tenderly referred to by the IT director as 'My Boys', gazed with frightened eyes from behind their terminals, like small feral creatures spotting a potential carnivore. I soon realized that his was due in no small part to the MD's habit of sporadically rushing into the IT department with his face flushed with righteous anger, gesticulating, and haranguing one of the poor IT staff for being an idle and incompetent.

Despite this atmosphere of palpable fear, I settled down to the work with gusto. My first task was to get live monitoring of the call traffic working as soon as possible, and tie up a number of loose ends. The salesmen weren't being monitored properly, fraud couldn't be tracked, and usage couldn't be counter-checked. Even the production of monthly invoices, and therefore the revenues, was on a knife-edge.

I spent a wonderfully hectic few weeks getting an emergency system in place. I crawled around the data centre in the semi-darkness, looking for that vital serial port with the live traffic reports. I got a SQL Server system in place that succeeded in getting estimates of the costs and revenues on a daily basis, to check for fraudulent traffic, and to check the output of the billing system. I managed to wade through the 'stream-of-consciousness' source-code to the billing system, written in MS Access, and correct the most obvious mistakes.

The first time I was called into the MDs room, I thought it was so they could make a little speech of thanks, tearfully pat me on the shoulder, and shake my hand in gratitude. I was somewhat surprised to receive, instead, a glowering and grim-faced dressing down. The MD waved his hands excitedly and the IT director nodded soberly and looked suitably crushed. Various managers and visitors looked solemn. I did what all seasoned contractors do in the circumstances, which was to calculate, on the hourly rate, how much I was being paid to be there in the room. It never fails to cheer, especially if you visualise five-pound notes floating lazily down from the ceiling.

www.dbooks.org

After the meeting, I apologized to the IT director for letting him down, though I admitted to not quite understanding how, or in what way, I'd done so. I offered to waive my notice period if he wanted to be rid of me. He was horrified and begged me to stay. To mollify me, he showed me his holiday snaps from visiting his parents in India, and even read me one of his own poetic compositions.

The IT department was a shambles and the working day of a DBA was defined by a frantic, desperate urgency. Management were snatching systems out of our hands before we could finish developing them. The sacred division between development and production ceased to exist in the scrabble to keep things afloat and get that vital information. I remember literally running down the corridor grasping the latest report.

Despite my best efforts, however, my mauling at the hands of the MD proved not to be a one-off. I'd be summoned and, as soon as I stepped into the room, accused of all manner of professional inadequacies. I would never argue. I would just adopt a vaguely contrite demeanour and drift off into my usual reverie of working out how much they were paying me in order to harangue me. The IT director would look solemn and agree with the MD with discreet nods of his head. After the meeting he would be paradoxically friendly, and appreciative of my efforts. The only upside to this, I noticed, was that the MD no longer practised his bizarre and distasteful incursions into the IT department to harangue the team over their shortcomings.

I soon realized, moreover, that the theatrical aspect of my dressing-downs was much more pronounced when visitors or particular directors, or managers, were in the room. It slowly dawned on me that I was being used as a ritual 'whipping boy' for all the sins of IT. Of course, this being an IT business, I was effectively the whipping boy for the whole enterprise. When royalty went to school, they were not free from punishment. However, in view of their exalted rank, the punishments were carried out not on the little princes of princesses but on poor surrogate children whose job was to accompany the royal scholar and receive random and unjustified punishments in their place. As the pay and conditions were, for the most part, good there were no shortages of volunteers.

Once I understood my role, I hammed it up a bit more, much to the delight of the MD and IT director. We would head off to a restaurant together after a session and the MD would buy us expensive meals in gratitude. He used this 'whipping boy' technique for impressing visiting shareholders, distracting prying government officials, withdrawing bonuses from the sales force, and sacking staff. Somehow, the targets of this neat psychological device, both visitors and staff, found it a huge comfort to see the representative of the IT department getting a dressing-down, with a look of suffering on his face.

Meanwhile, we continued the battle to get the relationship between the IT department and the business on a rational footing. We fought desperately to achieve order and rationality but as soon as we had determined the business structures and processes, management went and changed them. We knew what we wanted, but were forced to make compromises in the face of this grim every-day reality. And reality meant occasional glitches and bugs.

It all came to a head one day when I was summoned to the boardroom to be confronted by the MD, the IT director and a lot of Americans in sharp suits. The MD broke into his usual tirade, and we settled into our usual roles. Suddenly, however, the MD shouted "Phil, you're fired!" I was mildly intrigued by the unusual passion in his voice and glanced at the IT director. He looked puzzled and alarmed. I shrugged and the meeting carried on.

After the meeting we met, in the usual way, at a French restaurant around the corner.

"Hey," I said to the MD, "you were going a bit strong then. You had me worried for a moment."

"Not a bit of it," he protested "I meant it! It is about time we had a competent man in your position."

The IT director looked at me like a drowning man. I knew instantly what was going through his mind: if I left, he would once more become the company's whipping boy, the goat sacrificed as an atonement to appease the angry gods of commercial life. After a pause, in which all one could hear was the rattle of his teaspoon against the cup as he held the saucer in his trembling hand, the IT director spoke.

"Quite right of course, but it would make for a far better hand-over if Phil could work his notice and perhaps a little-bit longer."

The MD thought over the suggestion. The truth was that he had got over-excited by the importance of the occasion and had perhaps improvised too freely.

"What would you have in mind by a little bit longer," he asked slowly.

"Well, things are very busy, and Phil would be hard to replace; perhaps we can leave it a bit vague … a month … a year maybe?"

"Excellent plan," he snapped, "and we'll give you due notice of course." He then bought us a very nice meal.

After that, the usual routine was re-established with sporadic and theatrical dressing-downs in the boardroom interrupting what I believed was my real job:

www.dbooks.org

that of getting the IT department on a footing where it could grow to meet the requirements of the business.

Before I knew it the white-knuckle ride of this young telecommunications company had kept me busy for a year. I hadn't really intended to stay that long and, finally, I'd had enough. As it happens, it was really not so much the conditions as the travelling that had worn me down. In any event, a better job offer came up and I took it.

I was worrying over how to break the news to the MD and IT director, when I was summoned once more to the board room. This time, when the MD waded into his usual hostile speech, I stood my ground and told him my viewpoint. I gave what I still believe to be a fair and frank assessment of the progress the department had made and the difficulties we'd operated under. I did not mince my words. The IT director backed away into a corner as if I were a dangerous grenade from which the pin had just been pulled. The MD's face reached parts of the colour spectrum that I hadn't realised were possible in a live human. When I'd finished, a tense silence filled the room. However, the angry "you're fired" reaction that I was predicting failed to materialize. After a while, the MD's colour returned, and he nodded soberly. In a flash, the other managers in the room picked up the altered zeitgeist of the meeting and nodded amongst themselves like a theatrical chorus.

We had been sipping our post-meeting cappuccinos in silence for a while when the MD suddenly announced that the IT director had, for some time, wanted to move to another part of the company … and offered me his job! With a rueful smile, I politely refused and announced my departure. They put up a spirited fight to try to persuade me to stay but I gritted my teeth and held firm.

Still, it made me wonder whether I'd misjudged the situation. A short while later, I walked out of the building for the last time, fortified by a most generous leaving party and clutching a book of poems, pressed into my hand by a moist-eyed IT director, and still unsure as to whether I had been loved for my IT skills, and team leadership abilities, or was merely a convenient and able Whipping Boy.

www.dbooks.org

Hens that Crow

Short on results, but long on superficially plausible technical excuses, Phil Factor describes his dealings with the hen that crowed. The second time's the charm.



Robert, a programmer, was a fairly new recruit in my company. He seemed pleasant enough and got on well with the others, but I was uneasy about him because he seemed compelled to give me unsolicited lectures on technical subjects.

I'd been so dazzled by his slick interview technique and impeccable references that, against my better instincts, I gave him a job. He was now late with his various assignments but was very willing to give me long, superficially plausible technical explanations.

There is a saying that is popular among Development Managers:

'Hens that crow are never going to lay eggs'.



He seemed compelled to give me unsolicited lectures on technical subjects

Robert was heading inexorably towards the proverbial meat-cleaver, but I wanted to do my best to help turn things around for him. Out of desperation, I decided to phone one of his referees for advice on how I should be getting the best from him. I quickly realized that it was something that I should have done long before. Both of his references were faked.

It was clear that that I needed to find a replacement, but I wanted to find one without prematurely alerting Robert to the fact that his destiny didn't include a career in my company. I didn't tell the agency that had sent Robert to me about the CV, as I didn't want anything fed back to Robert. It also meant that I couldn't advertise his job directly, without raising his suspicions.

The solution came to me: I would advertise the position through, Batterbase Ltd, one of my other companies. You have to understand that the IT business can get pretty complicated, and I was not alone in having several small enterprises and a number of deals and projects going on at any time, only one in ten of which ever came to anything. Batterbase was a company I'd once set up to import and distribute a keyed ISAM database system from the States, but which had recently lain dormant as a plaything for my accountant.

Batterbase was obligingly hosted by an old friend of mine called Bernard (I performed a similar role for him). It wasn't a very onerous job, as the Batterbase phone rang only very occasionally, and it was usually only a hopeful salesman selling water-coolers. Bernard and I were members of that freemasonry of software developers that used to exist. We knew how random the nature of success was in the industry, and there was a fair degree of mutual support.

www.dbooks.org

I wrote out the basic job description for a programmer with knowledge of dealing systems and, in the hope of attracting a slightly higher class of candidate, I upped the salary a little. The description of the vacant post at Batterbase went out to the agencies, listing Bernard as contact.

A few days later Bernard phoned.

"Darlllling!" he trilled.

"My Love," I cooed back, nervously glancing around the office in case anyone was overhearing. Bernard was an extraordinary chap, an ex-actor who hid his unreconstructed pre-modernist masculinity behind an exaggerated theatrical camp.

"You'll never guess it, but you've had someone apply for that job at Batterbase." Bernard rather liked the clandestine nature of this assignment. "Let me do the first interview, darling, as I'm here already and it will save you a trip. I know more about your business than you do, so I can flannel it without a script."

I found it impossible to refuse his request, as it was very convenient even if it meant missing an amusing lunch with a great companion. A week later, he phoned again in some excitement. I'd rather forgotten about the interview in the distraction of sorting out various crises, and his first sentence was rather a shock.

"I've given the job to your candidate. What a treasure he was, he'll do you proud!"

"That was supposed to be a first interview, Bernard!" I cried, reproachfully, "I haven't even seen the CV!"

"Well, I know but he was a real cracker. He had terrific references and recent experience in dealing systems. You'll love him!"

"Bernard!"

"Oh dear. Maybe I got a bit carried away."

I was in the process of replacing a person whose references I hadn't checked properly with someone who I'd never met and who's CV I hadn't even seen! I felt that things had got slightly out of control.

On the morning that Robert returned from a bout of flu I guided him into my office in order to give him his cards. He must have sensed that I wasn't going to give him a pay-rise, as he beat me to it and handed in his notice. We decided on an immediate departure, and he shook hands and disappeared off down the

street. I spent the rest of the morning skipping around like a spring lamb at the joy of such a clean and harmonious parting. Real life rarely ran that smoothly.

Later on that day, the CV arrived from Bernard. I opened the envelope and immediately discovered that … I'd re-hired Robert. I could only assume that his agency had alerted him to a vacancy in the same field with better pay. Ah.

Immediately, I phoned Bernard and explained through gritted teeth what he'd done. Bernard was contrite but I couldn't help thinking that he found something funny about the whole thing. I didn't. Moodily, I glanced through the CV and saw immediately that it was a work of considerable fiction. According to the CV, he was not the junior programmer I knew, but a team leader with an almost unique overall understanding of our dealing system and its technical ramifications. He'd also extended his employment time here. Understandably, he'd kept his bogus referees.

Suddenly I saw the funny side of it all. As Bernard and I laughed, and mulled over the capriciousness of the Gods, the perfect solution to the whole matter presented itself to us.

Bernard played his part well. Robert was summoned to his office and told that the senior director, a fierce and fussy man prone to sticking to protocol, had insisted on checking references before confirming the appointment. Robert bristled and protested that the offer could scarcely be withdrawn at this stage. Bernard merely picked up the phone and checked the two referees, only to be told that no such people existed in the organisations, or ever had.

Robert's truculence had evaporated. "Never mind, " said Bernard cheerfully, "I know that your time working for Phil Factor will be more easily tracked." Robert looked startled. Bernard winked, "An old chum of mine, Phil. I'm sure he will be lavish in his praises of your work."

> It wasn't necessary. Robert was on his feet. Rapidly, he shook hands, scooped up the CV from the table and left, for the last time.

www.dbooks.org

The New Man

Working in IT, one meets all sorts of people, but rarely quite so odd, or creepy, as a 'New Man'. Phil Factor relates how a brush with such a character got him the sack on the second day in a new job.

I once got the sack on the second day of a new job, on grounds of incompetence. I had had no forebodings of what was to come; I was simply undone by my natural curiosity. Even if I had seen it coming, it wouldn't have bothered me too much. As with much of one's life, when apparently bad things happen, one just dusts oneself down, shakes one's head in puzzlement and wanders after the next cornucopia.

I joined the company in question on the rebound from the messy collapse of an Internet company, set up during the dot.com boom to trade commodities online. I'd started out as a lowly programmer but rapidly shot up the hierarchy and popped out at the top, as IT director. The brief glories of the boom inevitably turned to bust and the company collapsed in a fog of ill-informed recriminations.

In any event, there I was, older, wiser, and with several years experience developing Internet Applications under my belt, settling in to the relatively tranquil existence of developing websites for a software house.

My first task was to create a simple trading site for fabrics. I'd bottomed out the requirements with the client and cut the normalised database, so that I had a clear idea of the data model. As it happened, it all fitted well and there were few uncertainties besides the usual problem of having to trade in the metric system, whilst the industry still thought in terms of imperial measurements. I was just stocking the database with test data when my concentration began to waver slightly. I came back down to planet earth, from my normalized, rational fantasy world, to realize that the chap who'd joined the company at the same time as me was talking to a client, in an irritatingly loud and pompous voice. He was pontificating on the technology behind the website he was about to construct.

He was droning on about the advantages of Dynamic HTML. At first, I didn't really listen as I was engrossed in my own work. Shortly, however, I realised with a shock that the guy was talking with immense authority on a

subject he knew absolutely nothing about. In fact, it was hard to work out the various misconceptions he possessed, as he had got it so wrong. The client was nodding wisely and agreeing vaguely that DHTML was the way to go with his website. I suspected, however, that his response would have been similar if it had been suggested that 'friendly bacteria' was the way forward for websites.

After his client left, I was intrigued enough to wander over to his desk and engage him in conversation. I felt convinced that my colleague was a 'New Man' and was curious to see if my hunch was correct.

A 'New Man' is a term you may not be familiar with. A 'New Man' is one who, in response to economic or psychological pressures, reinvents himself as someone else, usually as an expert with qualifications and experiences. He takes advantage of any upheaval in society to become someone he isn't. The classic examples are the men who managed to insinuate into the post-war French government on account of their heroic, and entirely fictional, feats in the French Resistance movement. Such people appear in every society and rarely get found out, because of the energy they expend in establishing their false credentials. Occasionally one hears of surgeons who are unmasked as school dropouts, without any medical training at all; head teachers with a string of fake qualifications. There has recently been a case of a computer specialist, with a bogus degree, whose testimony as an 'expert' witness had jailed several people. I once had an IT director who had succeeded so well, as a 'New Man', with an entirely bogus identity, that he had established enough real experience to launch a legitimate career. Freed from the normal moral restraints that generally guide our lives, these 'New Men' are capable of deceptions that few of us would imagine possible. Their daring is their greatest asset.



Working in IT, one meets all sorts of people, but rarely quite so odd, or creepy, as a 'New Man'.

www.dbooks.org

My usual technique with IT 'New Men' was to exploit my biggest natural asset, namely my 'Nincompoop' face. Some people have a face that lends them an effortless dignity, a natural gravitas that adds weight to even their most fatuous statements. My face has quite the reverse effect. This can be a problem on any occasion that demands the caste of authority, but is perfect for this sort of 'undercover' work, since people tend to drop their guard.

It would never have occurred to my colleague that I'd had years of experience using DHTML. I asked him a few simple questions, to which anyone with even a passing acquaintance with the science of browsers and websites would have known the answer. He was all at sea, though he flannelled with such authority and fluency that I was momentarily dazzled enough to doubt my own knowledge and understanding. I thanked him and went back to my desk. Yes, a 'New Man', if ever I saw one. Of course, it was really none of my business, but curiosity can sometimes get the better of you.

That evening, after I got back home, I got a very apologetic call from the Agent, telling me not to turn up again at my new place of work. The Director had phoned him up and told him that he'd been advised on some authority that I knew nothing of the work I was supposed to be doing, and he was refusing to let me back on-site. It seems that the 'New Man' had gone to the IT director as soon as I'd left the building, and regaled him with the story of how I'd had to ask him a whole lot of very simple questions about website construction. Questions that I ought to have known the answer to if I was to be trusted with their website development.

It could have been that my 'nincompoop' questions really had backfired on me, and convinced him that I knew nothing, but I'm pretty sure that he realised I'd tricked him and that I was on to his secret. He'd therefore opted for a pre-emptive strike. In any event, he was convincing enough for the IT Director, as most 'New Men' are. Pursuit of your own survival goes hand-in-hand with acutely developed powers of persuasion.

God, I find, rewards you for placidly accepting your fate without rancour, and gives you bonus points for seeing the funny side. I felt a bit foolish about overusing my 'nincompoop' trick but shrugged the incident off and a second phone-call that night was from another agent, with the offer of a better-paid job nearer home.

I have to confess that, despite my good humour, I sent a letter to the Director, with a copy to the client for whom I started to write the website, pointing out a few facts that, in his haste, he had forgotten to establish before asking me to leave, such as my proven track record in the industry. I also gave a brief summary of the various employment laws he had breached. The result of this missive was explosive, and the agent, now angry rather than apologetic,

told me the Director was threatening to sue, on the grounds that I'd undermined the good commercial relationship he'd had with his client. He didn't, of course, and the agent wasn't even able to blacklist me as he threatened. Funnily enough, the new man's supercharged DHTML website never appeared, and the company too vanished within three months. There was, evidently, only one client.

I forgot the whole incident, until several years later. I'd gone to visit a relative in hospital and was discussing his progress with the hospital doctor. In the midst of the conversation, a man came up looking very important and had a brief conversation with the doctor. The face looked oddly familiar. Was my memory playing tricks, or was this the 'New Man' who had been responsible for my abrupt sacking? I tried to look nondescript. He strode off. The doctor apologised.

"I'm so sorry, I just had to have a quick word with the consultant; now where were we?"

www.dbooks.org

Section V: What If ...?

Most of Phil's writing is concerned with slightly oddball tales of the victories, defeats and absurdities of a life in IT. All of the stories up to this point in the book are true. Names have been changed to protect the guilty, some artistic license is taken with character descriptions, but all of the events described actually happened.

However, when the mood has taken him, he has attempted a few pieces that are more whimsical in nature. Pieces that still hold recognizable truths about the actions and behaviour of certain IT types, but transposed into more fanciful circumstances.

The inspiration for these can be something trivial, such as a certain song on the radio, but getting them down is anything but. Since they are 'pretty much written in blood' they are in scanter supply, and are well worth your attention. [TD]

www.dbooks.org

Had IT been responsible for the Creation

In which Phil takes the liberty of retelling the Creation story to make it 'relevant' to the modern IT professional.

God looked up wearily as a two-headed donkey wandered into the amphitheatre. It had a baffled look, and rolled both pairs of eyes before bleating pathetically, raising its tail vertically, giving a squawk of alarm, and exploding into a myriad of quivering chunks of meat. Whilst distractedly wiping off the stains from his toga, God cast an anxious glance at the assembled IT managers.

"Art thou sure of meeting the aggressive six-day schedule for this project, verily??"

The smartly-clad angels looked nervously around for a spokesman. After a moment, a well-groomed executive angel stepped forward.

"We have, er … hath …, a total commitment to quality delivery of an effective solution."

"… meaning?"

"Our mission is to achieve total excellence in meeting the timescales for delivery of the project to the defined and agreed scope".

"Quality … excellence ...", harmonised the assembled IT angels, upon hearing this confirmation of their mission statement, and rustled their feathered-wings to signal their commitment and solidarity.

"… and we're all clear on the project deliverables?" asked God. "Being more of the instinctive sort of executive, I realise I may have been heavy on the overall project vision … you know 'let there be light', and that sort of stuff … and light on the practicalities. I'm not really a detail person. That's why I delegate that sort of thing to you. I can appreciate that this is really a logistics and facilities-management issue but, we're 5 days in to a 6-day project and exploding two-headed donkeys at this stage make me rather nervous."

"You're not implying that maybe we need to invoke the change-control procedure, surely? The project definition is carved on tablets of stone and signed off by you."

"Did I sign it off? Does a lightening bolt signify assent?"

"Well, there is nothing in the project specification about donkeys having only one head, and I'm sure there is nothing about lack of donkey-explosions. It could be a very effective means of regulating their population." Observing God's rising eyebrows, the executive angel gestured in a conciliatory way. "Look", he said, "we can hammer out these details, but surely the most important point is that we are all synergistically working together in total commitment to achieving the project schedule and exceeding your expectations." He waved his hands elegantly, in a meaningless way. The assembled angels all took the cue, and looked innocent and keen.

"You're not bullshitting me are you?" asked God, wryly.

"Why no," cried the spokesman in exaggerated, affected, indignation at this idea, "Merely, managing your expectations. Besides, we have a bit of a slippage on bulls due to technical issues."

"Ah, technical issues? What a relief. Nothing important, then?"

The assembled angels looked at each other in mock confidence.

Meanwhile, behind this group, young technical angels were picking up the quivering pieces of donkey and attempting to piece them back together. Standing over them was a belligerent, pot-bellied angel in sandals and bottle glasses and with wings that looked as if they'd been through a shredder.

'Bloody shambles I call it", mumbled the pot-bellied one, "I'd love to meet the character who insisted on this bloody useless technical platform. Create a whole world with this? Who is He kidding? Thank goodness most of it is made of mud."

God, catching the faintest murmur of a foul oath, glanced in his direction. "You there!" he barked, "Art thou one of the angelic host?"

"No, he isn't", the manager angel muttered with marked distaste "he's a contractor angel. And of course the technical platform is in total compliance with specifications and will be, ahem, fully documented in the post-project Genesis report."

However, the manager angel's attempts to draw back God's attention were thwarted by the contractor angel's loud and bitter laugh. "Me, a permie!? You're kidding, mate! I'll get this lot done and then I'm off. What a shambles, I'm telling you. The bloody angels are so keen on promotion they've got no spare

www.dbooks.org

energy for the real work. We know that the devil has got the only good operating system, but why you couldn't at least have opened the wallet for a proper database I can't imagine."

"Proper database?" asked God

"Yes, we're using SQLite. Fine for managing your bleeding stamp collection, but not for this sort of work!"

"I said 'Let there be Light, not Lite'!"

"What you sayeth at one end and what cometh out the other, having filtered through six levels of management, are two very different things. I suppose you weren't insisting on Sun workstations either?"

"I had more of a celestial object in mind, actually …"

The executive angel became increasingly agitated as this conversation continued and finally interjected, "I think it would be better to allow me to manage the customer interface, if you don't mind. You need to concentrate on the technical issues" he said, huffily, and with an angry glare at the contractor angel. "And concentrate on them pretty hard too." he added menacingly.

"If you guys didn't keep meddling with the architecture then we'd have finished the job already and knocked off early on Friday for a beer break. The Sahara was a tropical paradise before you insisted on open-source components. And don't get me started on the animals. We thought the Alpaca was bad but that was before we had to use UML. It was then that we had to go down the two-headed-donkey route. Why, I'll never know but I suspect some manager wanted to put in his CV that he had 'done' a distributed architecture. And then you've got the user-acceptance testing. What confusion! This is the only project where bugs are actually specified. More species of bugs than anything else, and the test teams keep thinking a cockroach is a bug with a bug ..."

"Enough!" thundered God. "Yea verily this soundeth like an utter shambles. Let there be a project review!"

"Don't think we haven't been there already. At the last one, there was a long argument about re-designing an American song thrush and, by the time everyone had finished interfering, we had created the turkey."

God suddenly felt very weary. His fears and premonitions, which had gradually grown over the course of the 5 days, seemed to be confirmed. "Oh woe unto this project." he muttered.

"Cheer up" said the contractor, before the angelic manager could intervene once more, in order to effect a 'reconstruction of the perceived user-experience'. "There may be a way out."

"And what is that?"

"An insurance job."

"An insurance job? I see only through a glass darkly."

"Well, the project is insured against acts of God, isn't it?"

"Well yes, they forgot to take that clause out."

"OK, then. So what about a big accidental fire, just after the end of the project sign-off? We all get paid and you get the insurance money."

"Fire? Hmmm … not sure about that."

"Well it doesn't have to be a fire. The world is a bit big to accidentally drop down the stairs but you could always engineer a flood."

"A flood, eh?" mused God. "Shh! I'm thinking …" he snapped, as the manager made to interject yet again. "I think I'd better give Noah a call …"

www.dbooks.org

Had offshoring been responsible for Hamlet



After struggling with an outsourcing project, Phil wondered how Shakespeare would have coped if forced to outsource writing plays. Recalling the 'tragedy' that is Hamlet, it suddenly didn't seem quite so far-fetched ...

'*The most Lamentable Comedy, and cruel death ...*' (A Midsummer Night's Dream)

The Scene: A large oak-panelled meeting room of the Globe theatre on a cold September morning in 1600. The evening before had seen the disastrous dress rehearsal of 'Hamlet'. The audience had boo-ed the cast off stage. Rotten fruit had been thrown. Will Kemp, the actor, has resigned from the Globe in protest at the play being 'Not Funny Enough'. A crisis meeting is in progress. Present are Shakespeare (writer); Marlowe, Jonson, Beaumont and Fletcher (in-house development team) and a Moorish Project Manager.

Amidst the amorphous drone of the meeting, Shakespeare looks bored, and possibly hungry. He is idly scrawling his name on his pad. Wil Shakspure he writes, but is not quite pleased and tries again: Will Shakespare. He leans back in his chair with one eye closed in order to examine his work. Still he looks discontented …

Shakespeare [*soliloquy*]: Writing is a mysterious process. Just when you think you have the hang of it, experience proves you wrong. Thank goodness I've always had such a good in-house development team. Marlow, Jonson, Beaumont and Fletcher have helped me turn out some excellent plays. A team with a proven track record is surely worth its weight in gold! But no, some cream-faced loon in the accounting house says we need to shave a few doubloons off the expense sheet and the next thing I know, the board of The Globe are 'persuading' me to offshore the whole play-writing process.

It started well enough, I suppose. The proposal broadsheet was beautifully printed and promised to [reading from document] 'spearhead our business growth and place the Globe Theatre on the threshold of a new future.' Having the plays written by Moorish experts would, they argued, 'strengthen our competitive edge, not only by unmatched cost reduction, but also by a flexible integrated approach to a variety of playwriting challenges'. Sounds very impressive, no? And there is more.

'We are dedicated to developing and supporting custom scripts for new plays – tailored solutions that will meet thine particular target market'. And how about this: 'The Moorish company's dialogue-development systems art based on methodology already proven to the international playwriting industry'. I could go on, but needless to say, it all went down pretty big with De Vere and the rest of the Globe. In theory, all I have to do is specify the plot and they deliver the finished play, far cheaper than the in-house development team could have done it. Well, so much for theory …

Shakespeare is jolted out of his reverie by the raised voice of Beaumont.

www.dbooks.org

Beaumont [*glaring around the room, challengingly*]: Look, we've had a string of hits with our Shakespeare-brand comedies. 'Much Ado About Nothing', 'Love's Labour's Lost' and 'The Merry Wives of Windsor' all resulted in plenty of cod pieces on seats thanks to our quality initiatives. Do I have to remind everyone of my pretty strongly-worded broadsheet on the dangers of offshoring our new 'Prince' project? Without wishing to crow about it, we've all now seen that this offshore team of Moorish writers have turned it into a complete disaster!

Jonson: It's a bloody tragedy.

Fletcher: A flaming joke, more like.

Beaumont: The four of us may get a higher hourly rate, but we get the job done! At least the audiences laughed. Thou shouldst have heard the cackles when we called the hero of Midsummer Night's Dream 'Bottom'.

Marlowe: Quite; Masterly!

Beaumont [*fixing the nervous-looking Moorish Project Manager with a glare*]: Let's take you back to the project specification, shall we? This was supposed to be a simple comedy of misunderstandings! Hamlet, fresh from college, takes an unreasonable dislike to his stepfather Claudius, and Polonius tries to stop Hamlet's romance with Ophelia. He gets his son to dress up as Ophelia and trap him, but Rosencrantz takes a fancy to him. How difficult is that?!

Marlowe: Absolutely! All good family fun. Hamlet, Rosencrantz and Guildenstern together, college chums. A few jokes about putting swords in scabbards, cross-dressing, false bras getting dislodged, a few dropped doublets. And then they all laugh and make up at the end, whilst Hamlet and Ophelia kiss to a rousing chorus from the assembled company.

Jonson [*sounding pained*]: It should have been our best comedy yet! That scene where Hamlet thinks he sees a guest on the battlements and it turns out to be Horatio relieving himself over the parapet. He unwittingly urinates over Polonius who is peeping through the window, trying to chaperone his daughter who has taken a shine to Hamlet. Now that is a funny scene. Or at least it was until thou got thine hands on it …

Moorish Project Manager [*looking uneasy*]: Now that was a simple mifunderstanding. We thought it said 'ghost', not 'guest'. It really wafn't very clear …

Jonson [*laughing bitterly*]: Well, the sudden introduction of a ghost certainly stopped the fun dead in its tracks, didn't it?! And as to the rest of it … painful! [Shouting now] When Polonius gets caught red-handed spying on the

couple, instead of a playful whack on the arse, he gets stabbed to death behind the Arras. Where's the belly-laugh there?

Moorish Project Manager [*cheeks coloring*]: Thefe small mifcommunications are unfortunate, but...

Beaumont [*rising angrily from his seat*]: Small?! The devil damn thee black thou swivel-eyed loon. Rosencrantz and Guildenstern were supposed to be dead drunk in England, not just plain dead! And as for killing off the love interest, Ophelia … words fail me! 'Floating down the River' was meant to be a song and dance routine, not a stage instruction!! This is the work of a bunch of amateurs!

Moorish Project Manager [*flinging the broadsheet on the table with temperamental petulance*]: What is this thou art saying to me? Thou infult me, Thou infult our work! Thou thinkst we know nothing but we know damn all!

The Project manager stands, looking round challengingly and clutching the scabbard of his sword.

I can affure you that the team which worked upon this project were all experienced versifiers and qualified playwrights, sensitive to thine bufiness requirements, able to bombast out a blank verse as the beft of you, and eager to exceed all thy expectations ….

Shakespeare: But we were expecting a romantic comedy …. and you didn't even finish it off properly. The finale was meant to be a series of duets culminating in the whole company singing a rousing rendition of 'Friends to the End'.

Moorish Project Manager [*snatching the specification sheet back from Shakespeare*]: Duets!? No, no!! It definitely says duels!

The Project Manager waves the paper around, a look of triumph on his face.

Shakespeare [soliloquy, staring at the specification]: Bother, I've forgotten to cross my ts again. Bloody writing!! Isn't it about time somebody invented some sort of writing machine?

Beaumont [sarcasm rising in his voice]: Well, at least we had the audience in fits of laughter with that last scene! People drinking poisoned wine, swords with poison on them, duels! It is one of the stupidest scripts I have ever read. The play ends with Horatio weeping noisily on stage surrounded by four stiffs! A child could have written a better ending. I'm only surprised that the actors weren't giggling too when they were supposed to be playing dead …

www.dbooks.org

Jonson: That was easy. They were rigid with embarrassment at having to act this nonsense!

Jonson, Beaumont and Fletcher all burst into gales of laughter. The Moor, trembling with anger, storms out of the room muttering dark oaths. Silence falls on the meeting. Shakespeare moodily returns to his doodling. He idly scrawls 'Will Shakespeare' on his pad, and suddenly his face brightens.

Shakespeare: I think I have an idea …

[*Suppressed laughter around the table*]

Marlowe: A better one, I hope, than making Henry VI into a serial. It never got past the third episode!

Shakespeare: [*waving his hand dismissively at Marlowe*]: How's this … we shelve this disaster somewhere in the attics of the Globe. And for heaven's sake, let's make sure it doesn't go in the folio. I'd hate to think that posterity would believe I actually wrote such rubbish. We then play for time by telling the Board that we are ironing out a few technical issues that emerged in user-acceptance testing.

In the meantime we rework it back towards its original specification, with juggling oranges, amusing cross-dressing incidents, dogs on wheels, inflated pigs' bladders … the lot! Now lads! [Looking at his old ghost-writers – or was it guest-writers – with renewed warmth] I think we have a play to write! And we only have two weeks to do it, so we'd better get going!!

Beaumont [*excited*]: Yes! We can call it 'Hamlet, the dunce of Denmark'.

Shakespeare: Hmmm, I had in mind something more like 'All's well that ends well'. How does that sound to you?

All: [**raising their flagons of ale with renewed enthusiasm**]: Where there's a Will, there's a way!!

The End

'I saw Hamlet Price of Denmark played, but now the old plays begin to disgust this refined age, since his majesty has been so long abroad' -- John Evelyn (1620-1706)

'Was there ever such stuff as great part of Shakespeare? Only one must not say so. But what think you? What? Is there not sad stuff? What? – What?' -- George III (1738-1820)

www.dbooks.org

Had Tennyson been a Technical Author

Phil gives reign to his inner poet, documenting the SQL Server index in the style of Lord Alfred Tennyson.

I have always wondered why nobody has written a book on SQL Server in verse. To correct this lamentable gap in the market, I have been penning some stanzas. Here, as a sample, is a short verse on indexing.

The Index: An Elegy

An index is used as a short-cut to data
a table will warrant one sooner or later
Because only one can be clustered, beware
and ponder the index you cluster with care.

The issues are clearer than you might suppose
this index determines the order of rows
so searching the index requires less I/O.
Selecting the column on which it should go
depends on the way that the rows are selected,
which should become clear if the Schema's inspected.

One problem, however, I think you should know,
retrieving a range can be horribly slow.
A non-clustered index is almost as good
once ordering keys can be well understood
make sure that the columns you use are selective
for if too few values, it's most ineffective.

If data is changing or updating too

with frequent insertions, keep indexes few.
From 2000 on you can index a view
(but then there's restrictions on what you can do)
and even on computed columns as well
but only if deterministic as hell.

For reasons which often are misunderstood
a non-clustered covering index is good
when composite columns are used with some care
they outperform anything else that's out there.

www.dbooks.org

Had God been a Technical Author

Phil was contentedly re-reading Exodus, when he suddenly found himself imagining God in the role of modern-day technical author with Moses as his long-suffering editor.

'And The Lord spake unto Moses face-to-face as a man speaketh unto his friend' Exodus XXXIII: 11 JKV

It wasn't the cool wind on top of Mount Sinai that caused Moses to shiver, it was panic. As the smoke that engulfed the summit briefly cleared, Moses had anxiously looked at the stone tablets. They were blank, just as he'd left them.

He'd had to make all sorts of promises to the stiff-necked multitude, who were pitched below in the wilderness, pining for the fleshpots of Egypt. They'd started getting more and more attracted by Aaron's Open-source Golden Calf project, and so He'd countered by committing to the publication of a definitive prestige guide to the true religion, Mosaic Law in a Nutshell. With great rapidity, Aaron was able to outmanoeuvre him with the announcement of the imminent release of the rival GravenImagesTM Visual Quickstart.

When Moses, in some desperation, had first mooted the idea for a Dummies Guide to Monotheism, God had been so enthusiastic. "Yes!" he said, "I've always fancied myself as an author. I've had several ideas floating around for a while. I'm sure I could bash out a book in no time." But then there was delay after delay, with several different plausible excuses. When reality kicks in, the art of instructional writing doesn't look quite so easy.

"So" shouted Moses, clapping his hands nervously, "What hast thou got for me?"

"Aaaaah, well, sorry, but the new chapters aren't going to be ready in time, they're going to slip. I've got all sorts of pressing commitments right now; fallen angels causing me hassle; also, for some reason, the creation just didn't happen in Utah. Helluva mess. We've had to sort of evolve a solution there as we went along. Still, we all felt better once we'd established that it was the developers' fault."

"But … thou hast made some progress, right? Listen, God, if thou canst at least give us the Mosaic code, then we can ghostwrite the commentary. If we don't respond quickly, we'll have lost the initiative".

"Tell you what … I'm giving a couple of presentations over the next few weeks to the Hosts of Midian, so I've got to work on the material anyway. Based on the feedback I get, I can pull it all together into a couple of really top notch chapters, say next week?"

Moses sighed. "I knowest not, God. I promised the multitudes a book that wouldst cover all of the big issues. They'll feel short-changed. Baal hath already got a publication list as long as thine arm."

They lapsed into a ruminating silence.

"Mo! Let's turn it into a Crib-sheet! You know, one of those hyper-condensed 'Top Ten Tips for this, that and the other. We could turn that out in no time'?"

As Moses started shaking his head, largely out of habit, God's suggestion began to sink in. "Sort of 'Top Ten Reasons to smite Jericho'?" he mused out loud, "or 'Ten False Idols and How to Destroy them' verily?" His voice was rising in pitch now.

"Yeah, or perhaps we could put it a bit stronger than that even. What about 'Ten Best Practices for the Children of Israel'? Nice, short, easy to do."

"Great idea, that. I liketh the 'Best Practices' phrase. It soundeth keen and efficient."

"Maybe it's not quite prescriptive enough. You know how the Children of Israel are, you have to put it straight or they'll be looking for loopholes. Could we make it 'Ten things you're not allowed to do' ?"

"Hmm. Great, but it isn't snappy enough. It has certainly got to be 'The Ten Something'. What about …" There was a thoughtful pause. "Oh hang it, let's have a quick omer of manna and maybe inspiration will strike."

www.dbooks.org

Had the Sistine Chapel been Created by Committee

Phil wonders if the wonderful frescos of the Sistine Chapel would ever have got painted, if the type of corporate 'best-practice' and management now seen in IT development projects were in place then. Phil Factor imagines the pressures on Michelangelo and the Pope, from strategists, MVPs, project managers, analysts, architects, sponsors, and bureaucrats, and conjures up two visionaries in a very modern predicament.

"Once a development project looks likely to be a success, it's amazing how many people consider themselves essential for its completion." Pope Julius scraped a drop of paint from his immaculate cape and studied it frowningly for a moment. Then he continued "For is it not said, (1 Kings 9:23), *'These were the chief of the officers that were over Solomon's work, five hundred and fifty, which bare rule over the people that wrought in the work*'."

Pope Julius shifted his attention to Michelangelo's packed lunch "I say, you don't really want all those pickled gherkins do you?"

The two men sat high in the scaffolding in the Sistine Chapel, their legs swinging in the air. Below them draped netting, designed to protect the chapel furnishings from plaster and paint. Above and around them blazed the half-completed frescos. Up here, they felt closer to God, rank was forgotten and their friendship felt simpler, more like the companionship of two schoolboys who have climbed up a tree.

Michelangelo shrugged. He generally hated interruptions, especially when the plaster was drying. Still, he always looked forward to a visit from Pope Julius, and it was nice to straighten his back. He sighed and scratched his head before replying with *'So and more also do God unto the enemies of David, if I leave of all that pertain to him by the morning light any that pisseth against the wall'* (1 Samuel 25:22). Help yourself to a Gherkin."

The Pope munched reflectively. "I'm getting a lot of pressure from the cardinals to put this project on a proper footing. I've just had a very difficult

meeting with them. They kept raising tiresome issues. In the end, I had to concede that I would at least introduce you to a Project Manager. His function will be to oversee the work and ensure that it meets its targets."

The pope gestured to a group of befrocked clerics standing in a respectful group far below them in the nave. Michelangelo pulled a face and muttered *"If a wise man contendeth with a foolish man, whether he rage or laugh, there is no rest."* (Proverbs 29:9.)

Wearily, the two men descended ladders, from scaffold to scaffold, until they reached the nave. A solemn looking young priest moved forwards to them.

"Antonio," said the pope, "I believe you've some interesting observations to make about this project."

"Yes," he replied, earnestly studying a large wadge of manuscript paper, "I have performed a time-management study and calculated that Michelangelo actually does productive work for only 40% of the time. On an average day, he spends 20% of his time staring at the designs in an apparent trance-like state, 10% of his time gazing out of the window, scratching, 15% of his time chatting, particularly to youthful choristers, and 15% of the time showing the work to interested officials."

Michelangelo flushed with irritation. "That apparent 'trance-like state' is the process of planning out the grand design!" he snapped. "Every figure has to be painted in fresh plaster. There is no time for preliminary sketching; the design has to be there in every detail before the plaster fully sets, which it is doing at the moment, incidentally. If there is any mistake in the overall design, then the plaster has to be stripped off again. A figure looking in the wrong place, or just a few inches to one side, or God forbid, in the wrong scale …"

"Yes, I'm glad you mention the designs" interrupted Antonio, curtly, pulling out another manuscript, "because it has come to our attention that they are not in source control. Some of the figures have been painted without any prior design at all, or agreement with the steering committee. One can only imagine the consequences of an earthquake or fire. We've also had a number of complaints from the specialist craftsmen who expand and transfer the approved plans onto plaster ready for painting. Evidently you're not drawing exactly to the specification but painting freehand. How are the testers going to check that it conforms to the spec?"

"'Agile', they call it. It is the latest thing," muttered Michelangelo.

"Latest? Hardly … you're still working in two layers of plaster when the fresco hotshots are all using N-layer Architecture! What I suggest is that we get a couple of these fresco hotshots on board – Perugino, Botticelli and

www.dbooks.org

Ghirlandaio are all available. They will do the designs and leave you to do the painting. We won't insist on you coming in and painting at weekends of course, but it is customary when projects are in overrun."

Michelangelo stared at the netting draped from the scaffolding. The vast fresco swirled up over the ceiling, like a Technicolor tsunami, awesome even in its partially-completed state.

"I have, here in my heart, a vision of the entire work, in every detail," he said. "It would pain me beyond endurance to see it misinterpreted. Besides which, I employed four contractors from Florence and, as soon as I'd got them trained they tried to double their rates. They were MVPs too. Mercenary Vatican Painters."

"You're not much of a team player are you, Michelangelo?" said the Pope, chuckling fondly.

Antonio scented support from the Pope. "Look here Michelangelo, I hope you don't mind me asking, but why aren't you standing back and managing this project rather than getting involved in the actual development processes. After all, at your age it's time you were moving up the promotional ladder."

Michaelangelo frowned. "You forget that the game is called 'Snakes and Ladders', not 'Ladders'. There's a reason for that, I'll wager. I'll play safe and carry on painting and carving just so long as someone pays my invoices."

Antonio sighed wearily. "Well, someone has to manage this project," he said. "It is all taking far too much time. Talk about Scope Creep! When the cardinals submitted their strategic business proposal for this fresco, it was for a 2-year project based on pictures of the twelve apostles, fully clothed. Instead, you have felt obliged to deliver a vast panoply of naked and half naked bodies…"

"It came out strongly in the Focus Groups," said Michaelangelo, delighted to trump Antonio with his own jargon. "People insisted on a naked Adam and Eve, and, anyway, I charge extra for clothing. Those folds in fabric don't come cheap. Besides, the 'ignudi' are very popular with the visiting monks."

"Yes, incidentally," muttered Antonio, pulling out the appropriate form. "Cardinal Carafa has put in a change-request to have breeches painted over all those reproductive organs. He finds them distracting when he is in prayer."

Michaelangelo opened his mouth to protest but was interrupted by the officious Antonio. "This whole project is a mess. We're now talking at least four years to completion if things go on at this rate. Half the chapel roof is full of scaffolding and netting, which is both unsightly and obstructive.'

Michaelangelo was exasperated. "Since the project got the go-ahead, every cardinal in the place has insisted on getting their pet components in as part of the overall design. I never wanted to paint all those blasted ancestors of Christ. I even ended up having to put pictures of the cardinals' families in the triangular spandrels above the chapel windows. How on earth can I be expected to get the project finished on time, when it keeps being expanded!?"

Antonio looked up and spotted his own boss, depicted in a spandrel, above the east window. He shifted uneasily and decided to switch the focus of his attack. "Let's look at the other issues. We have done a health and safety risk-assessment. We were shocked at the risks, and our employee insurance won't cover it."

"Oh, come off it! How many Fresco-painters have ever been injured at work?"

"That's not the point. Risk assessments evaluate the risks, rather than look at accident statistics. Besides, there are many compliance issues. Did you study Church law and best-practice before putting in all those Humanist elements into the grand design? This is going to cause all sorts of difficulties when we audit the expenditure."

"Hardly. The Pope commissioned this and the Pope is infallible: Humanist elements or not."

Antonio looked appealingly at the pope, but Pope Julius avoided his gaze by affecting to scrape a bit of plaster from his robes.

"And who do you think you are to tell me what I ought to be doing? Have you ever painted a fresco?" cried Michaelengelo, his patience finally at an end.

"I have a proven track record in organizational effectiveness and project management," said Antonio proudly, "which continue to be the discriminators that the Vatican wants to see. My company's deep experience in managing the decoration of church architecture will allow the church organization to achieve the efficiencies and increased productivity they desire. It's a win/win relationship."

At this, the Pope finally raised his hand and interjected. "Thank you Antonio, I'd like a word with Michelangelo in private."

Antonio had worked himself into a state of righteous pride. He glared disapprovingly at everyone as he gathered up his papers and flounced off, importantly.

"... and all you others, too," said the Pope, looking round at the assembled cardinals. "Please go. Michelangelo and I have some things to discuss."

www.dbooks.org

The two friends sat together in a pew, staring reflectively at the high altar as the clatter of clerical shoes died away.

"Your holiness, I suspect you want to talk about sponsorship."

"It is a delicate matter, but we have to be conscious of the business plan."

"Everyone knows that the ceiling glorifies God, demonstrates his might, and instructs the faithful the bible story."

"Quite. However, we both know why Libica, Daniel and Cumala are all carrying enormous books – the printers' and papermakers' Guild paid a handsome contribution to this project. You did very well to paint all those prominently positioned hops and grapes. The Brewers, Vintners and Publicans have all been most generous. As the book says 'And thou shalt make bars of shittim wood' (Exodus 26:26). Joel seems to be unravelling a roll of toilet paper - a useful and lucrative reminder to the worshippers. We must look at all revenue sources where we can: For is it not said 'Take ye from among you an offering unto the Lord: whosoever is of a willing heart, let him bring it, an offering of the Lord; gold, and silver, and brass' (Exodus 35:5) I don't understand why you were so rude to the Medici's marketing manager, though."

"I merely pointed out that a large sign, pinned over Adam's crotch, announcing the Medici's Banking Services, might militate against the theological message."

"Hmm. He is walking strangely and muttering about being assaulted with the handle of a paintbrush."

"Well, it was his insistence of an oscillating notice board, surrounded by candles, saying 'Congratulations, you are our 10,000 visitor at the Sistine Chapel. Call into Medici Bank for your cash prize'. Even I can be provoked."

The two men fell into a moody silence. They stood, staring at the vast ceiling, and lost in thought. Here, in plaster was a complex, intricate message. For the faithful, it was a map of their beliefs and dogma. For everyone, it was a depiction of faith, and a blueprint of civilized life, preserved so that future generations could see, and understand the glory of the Christian era.

"Damn it." said Michelangelo, "If ever we succeed, it will be through your vision, and my brush-strokes, not through management teams, and broad consensus."

The Pope did not answer. He was lost in a reverie as he stared at the almost-completed design for the creation. His lips twitched as though, in his imagination, he was talking to someone.

"Michelangelo," he said at last, "God has been laughing at us. See him in your fresco creating the universe: A single creator. Is he surrounded by bleating configuration managers, compliance officers, health and safety officials, strategists, project managers, accountants and architects? Oh no, the act of creation is compelling, and yet it is too easily stifled. *'Go to the ant, thou sluggard; consider her ways, and be wise: Which having no guide, overseer, or ruler, Provideth her meat in the summer, and gathereth her food in the harvest'* (Proverbs 6:6-8). Paint away, and paint alone. It will upset Antonio and the cardinals but I will pay for it myself, and you shall not be bothered by distractions."

Neither of them spoke for a while. They stood, running their gaze over the work, two figures alone in the vast space, with no movement but the gentle billowing of netting under the scaffolding.

"Y'know," said Michelangelo eventually, "it must be my astigmatism, but I could have sworn that God winked at us from the fresco just then."

www.dbooks.org

Captain Codd and the Simple Proposition

In which Phil presents a comic strip script for Database Professionals. The artist charged with providing the illustrations reported sick, and has not been in contact since.

Scene 1

In the church vestry. Polly Morphism looks despairingly at Rev. D. Composition. The elderly priest is bent over his desk in a canonical form (he is a cardinal type). The candle flickers; Polly is tearful.

D Composition

(shaking his head sadly) Hah, that base Relvar! The wicked Relvar Predicate is entirely without any integrity constraints. He is an insensitive curser and a primitive operator. Do anything to avoid him.

Polly:

It is no good; he has my pet tuples in his nidalic view. I must retrieve them with my tuple extractor or they will be subject to lossy decomposition. I will have to bargain with him.

D Composition:

I have to admit that he is a star schemer. You'll have to negotiate, maybe offer buffer cash. Blackmail is rank Relvar's characteristic function.

Scene 2

A smoky dive, called the 'HotSpot'. Rough types abound. Relvar and Polly sit hunched over a table. Her faithful SPID stands behind her, arms folded impassively, scimitar tucked in his belt. Relvar's SPID stands, sneering wickedly.

Polly

I've come to negotiate, to make a simple proposition.

Relvar

Your n-Place or mine? Your persistence is monadic. I'm glad you have seen reason, because your tuples are in my power. Soon, the Tuple Unwrapping will begin. Their internal predicates will be mine *(Nasty laugh)*. I will be idempotent!



Unhand my pet tuples, you fiend!

Polly

(flushing) You swine! Haven't you heard of De Morgan's Laws?

www.dbooks.org

Relvar

(with ACID sarcasm) Thanks for the Complement. You scare me! Hah! That flat relation? Pshaw! *(audience boos noisily)*

Polly

(thinks: Maybe this is a bad transaction scenario) Surely we can be sensible? I'm here to see if we can do a deal to retrieve my tuples. I have an atomic proposition that could lead to a Transaction. My SPID can start the process with a triggered procedure.

(Suddenly, an Argument starts at a nearby base table. A fight breaks out. A lunging body falls perilously near to Polly and knocks her SPID flying. Inverted Tables everywhere. The scimitar skids across the floor. Quick as a flash, Relvar's SPID leaps onto Polly's SPID and pins him to the ground.)

Relvar

(gloatingly) So fate has chosen your SPID to be a deadlock victim. Soon he will be a bound variable.

Captain Codd bursts in on the scene. A clash of Cymbals.

Codd:

Not so fast, Relvar. You have committed a fatal error and made a hash of it. Polly is under my protection!!

(An immediate conflict ensues. Wham! Bam! Captain Codd delivers a lightning Sheffer stroke to Relvar's jaw, Soon, Captain Codd has Relvar in a granular lock followed by a restriction, an immediate constraint, finishing with a deadly embrace.)

Relvar

Aiee! You have me in a Multi-Relvar constraint. I repent!! Let me go! Release my constraint! From now on I shall lead a Normal life.

Captain Codd:

If so, it will be your first Normal Form, Relvar. I suspect you will need a unique constraint. Any base manipulation, and you will be a bound variable in a loop structure.

Polly

Thanks, Captain Codd! I rely on you but it is a multi-valued dependency, though I am irreducably dependent.

Scene 3

An Arcadian meadow. The sun is shining. Polly Morphism and Captain Codd admire Polly's flock of tuples.

Polly

(surrounded by her bleating tuples, nuzzling up to her adoringly) Oh Captain Codd, now my SPID is released and my tuples safe, I can enjoy a tuple union. How can I repay you?

Captain Codd

(looking serious) It isn't over until we can engineer some sort of total Relvar constraint, with an identity restriction. He could yet strike again.

Ominous music ...

Will Relvar Predicate strike again?

Look out for the thrilling Sequel to ... Captain Codd and the Simple Proposition.

Heroic music ...

Ends.

www.dbooks.org

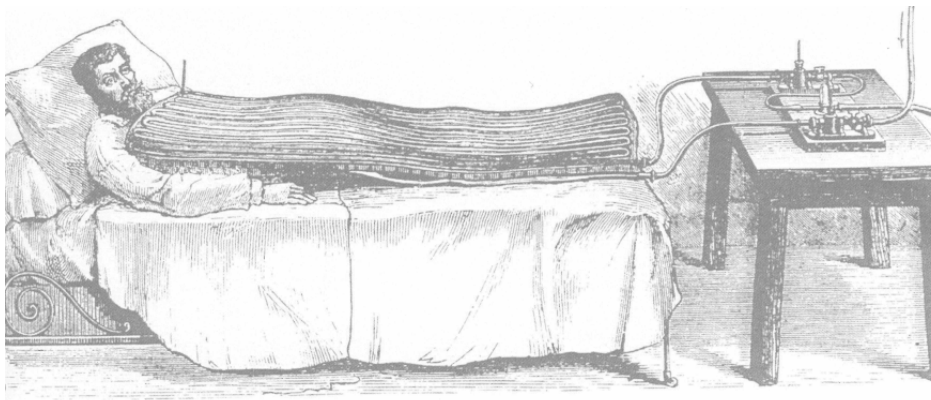
Section VI: Hiccoughs in the Working Day

Sometimes, to maintain his sanity, or just to get through the working day, the seasoned IT hack just has to step back from the chaos and confusion around him and find diversion by whatever means are close to hand.

From working out how to charge for time spent asleep, to playing the 'waffle game' during PowerPoint presentations, to counting five-pound notes floating lazily down from the ceiling while being castigated, Phil has a few useful tricks up his sleeve. [TD].

www.dbooks.org

I could do it in my Sleep



We sleep, but the loom of life never stops. High-pressure projects often cause Phil to start programming in his sleep, even being heard to cry out 'Save, save!' when shaken awake in the morning. Could he possibly start charging clients for time spent in slumber?

Like many other programmers, I have adapted to a rather cat-like lifestyle of intense activity followed by relative languor. This coincides with the cyclical nature of the industry: Either there is too much work or too little. At the moment, there's too much work, so I'm tending to work around twelve-hour days, often more.

Of course, insomnia isn't a problem; not with my copy of 'SQL Server 2000 with XML' on my bedside table. Two pages, and I am insensible, even if I pick it up at noon, after drinking strong coffee. Every doctor should prescribe this book. I recommend it unreservedly.

Definitely, insomnia isn't my problem when I'm tackling a high workload. The worst problem is being hardly able to see the screen by the end of the day. Almost as bad is the fact that I carry on programming in my sleep. In this strange state of unconsciousness, I'm not entirely sure what I'm programming, or why. This is, I suppose, not far off the everyday work experience of the

average developer. What is more alarming though is that, after putting in eight hours of sleep-programming, the whole lot gets lost on waking. Sometimes, I am even heard to cry out 'Save, save!' when being shaken awake in the morning.

I can normally remember nothing of what I've written in my sleep, but recently I seem to have been working out procedures in Transact SQL for zipping or 'Huffmanising' text. This was triggered off by my penultimate blog, on Teddy. I was hoping to entice other people in doing so, in order to munificently issue a glittering prize, but nobody took up the challenge. I don't blame them, but I tried it myself to make sure it was possible and, despite my complete ignorance of the algorithms, I reduced the storage requirements of ASCII art down by 80%. Even ordinary text boiled down 60%. It is an idea that, for some reason I can't leave alone, and I keep getting ideas for improving the method that I then have to tap in hurriedly after breakfast before the memory fades.

I've always wondered why it is that a programming problem that seemed so daunting the previous night has a simple solution when the morning comes. 'The phrase 'I'll sleep on it' could describe the strange way that the brain keeps doggedly chipping away at problems. Maybe one actually does remember the conclusions of all that unconscious programming effort. Now all I have to do is to work out a way of charging all that sleeping time to my clients.

Some programmers I know actually manage to charge hours spent asleep to their clients. One of the strange techniques one can acquire when being paid by the hour is to be able to sleep whilst giving the appearance of staring in intense concentration at the screen. I'm hopeless at this. Whenever I fall asleep whilst programming, I wake up with the neat impress of the keyboard on my forehead. Alternatively, I end up with my head flung back, mouth open, snoring loudly. Neither type of shut-eye inspires the confidence of one's employers.

There was a time in the IT industry, before laptops and PowerPoint, that salesmen used to employ slides and slide-projectors. The wonderful thing about these devices was that the room had to be plunged into darkness for the slides to be seen on the screen. Many a salesman must have wondered why there were so few questions about his splendid products, or why nobody laughed at his jokes, when he was giving his pitch to a roomful of developers. Once or twice, I have suddenly switched on the lights to be met by a scene reminiscent of the aftermath of the Hugenot Massacres. Still bodies slumped everywhere. Sales presentations seemed to be a universal trigger for sleep amongst IT staff. Ah me, the cruelty of technical progress.

www.dbooks.org

To be fair, all this is the reaction to the demands made on developers to work ridiculously long hours. I suspect that, out there, there are others who have adapted in ingenious ways to the pressures of the job, and of course we'd love to hear from you.

www.dbooks.org

The Pub Lunch and Programming

Alcohol is like oil. A little of it makes the brain run more smoothly: too much, and your thoughts skid off track.

One of the great pleasures of programming in a team is the pub lunch. It is something I always relish, particularly on a Friday, for the refreshments, the gossip and the companionship. I thought that this ancient ritual was an international one until I worked for a while in Japan, and discovered to my horror that it was soft drinks only until the sun went down. At first I thought this was due to their abstemiousness until I saw the propensity of the Japanese engineer or programmer for consuming the stuff once the sun had dropped.



Some of the best code we ever wrote!

Nevertheless I always wondered whether there was a moral dimension to celebrating the end of the working week with a couple of pints of best Bitter. Is

it true that one's capacity for work drops after the Amber Nectar has hit the bloodstream? I have always doubted it. Do careless errors slip into one's code after the beverage hits the bladder? My own experience tells me not. Once the warm glow of the hops descends on my soul, I feel calmer, more confident, and less distractible. My best ever code has, I have always considered, been written with a jar of Stout in one hand and a cigar in the other.

There will be some, particularly those with the cold grey ice of Puritanism in their souls, who will click through their teeth and protest that I would be better off drinking water and breathing in fresh invigorating air.

This should be put to the scientific test. I have of course done the best I can but my results are merely indicative. I have conscientiously plotted the average number of times I have hit the compile button before a stored procedure has compiled without errors, against the number of half-pints of Beer I have consumed. The data was collected over a period of a month. The results, for what they are worth are:

Consumption	**Errors per Stored Procedure**
Sober:	4
Half a pint	3.2
One pint	2.87
One and a half pints	3.2
Two pints	4.8

The problem with these results is that they take no account of the 'Placebo Effect'. My error-rate may be more closely related to my subconscious will to come up with the result I want. For a controlled experiment, a whole range of programmers, selected by random numbers and balanced so as to represent a true sample of the population of programmers, should be given both real beer and an alcohol-free 'Placebo' beverage that is indistinguishable from it. It should be done at various times of day with a full range of database tasks. It is an indictment of the poor state of academic psychology that such a task has not yet been undertaken. Of course, for all I know, a PhD thesis may already lie neglected in some university library with the truth already firmly established

Should the community of database developers and DBAs unite to determine this issue once and for all? If it can be proved beyond all doubt that a good beer actually improves ones programming skills then maybe one can claim the cost of purchase against tax?

www.dbooks.org

At half-past three, it's time for tea...

According to Phil, any company that doesn't understand the importance of providing its developers with first class coffee and tea is one doomed to under-achievement, and ultimately self-destruction. He may have a point.

"Look, see what I mean? " the manager said, as a dishevelled DBA stumbled past his pig pen grunting inconsequentially to himself.

"Good morning, " said the manager to him, forcing a beam from ear to ear, with a rather strained bonhomie.

The DBA gave an extra-loud grunt, his pale puffy face turning to stare balefully at the manager, as he shambled off to his flickering screen.

I knew the problem. Sometimes, in personal references, one reads comments such as 'excellent technical skills but not ideally suited to a customer-facing role'. In the case of this particular IT department, the problems were more acute. Everyone was quietly working away, but nobody was talking to anyone else, except to complain about 'lack of communication'. The inevitable consequence was a strange increasing social isolation. Everyone was working hard but without shared purpose. Productivity was awful.

As the manager and I mulled over the problems, an email arrived 'ping' in his inbox. It was from the grunting DBA. It read 'Morning'.

The manager looked pleadingly at me, much like a dog desperately paddling for shore, and asked my advice.

Complex problems sometimes don't require elaborate solutions. Fortunately I knew enough about the department at this stage to come up with a solution.

I wish I'd thought of this clever strategy, which I originally came across a long time ago in a college of psychiatry. They were suffering the same problem. No discussions, no testing of ideas by bouncing them off colleagues, no coordination, just practitioners working in isolation. These guys were supposed to know how to tackle problems like this but the disease of the

community had crept up on them and they'd been unaware of their own dysfunction until an outsider pointed it out to them. This is the solution they came up with, and I've always been in awe of it.

Someone from outside the organisation was hired solely to prepare coffee for them at ten thirty, exactly, in the morning and tea in the afternoon at three-thirty. This was no ordinary person, but one of the few who knew the exact science of preparing such beverages. The coffee and tea were so good, and smelled so beguiling when they were prepared that everyone left their desks to share in it, smile, chat to others and socialise. From all over the department, like rats to the Pied Piper, the staff appeared. The timing was exact, and announced by the ringing of a 'Pavlovian' bell.

The genius of the task of preparing the beverages would call on the coffee shop on her way to the college. She would buy fresh, lightly-roasted Kenyan Peaberry coffee still warm from the roaster, and only the best and freshest tea. 'If you can smell it at all strongly, it is too late, the flavour has gone from the bean' she told me. She would grind the coffee by hand so as to ensure that the coffee beans did not heat up to the point where the volatile essential oils left the beans. She would infuse the coffee with water that was near, but not at boiling point.

When the urn of coffee was ready every cell of our bodies yearned for the magical brew. The final touch was to heat up some of the spare beans until the corridors were filled with the aroma of fresh-roasted coffee.

Believe me, it was impossible to resist.

You may never have experienced the effect of a properly prepared fresh coffee hitting the blood stream, the subtle blend of caffeine and essential oils; that surge of confidence, concentration, and bonhomie. It is said that coffee was discovered when a sheperd in the Yemen, in around 500AD noticed that his sheep, after grazing on some unusual shrubs, started leaping about oddly, and inventing novel applications for XML. Hmm, perhaps I exaggerate. Certainly, the effect was striking. The coffee room resounded with happy members of staff, talking, discussing, relating, and sipping the magical beverage. The same was true of afternoon tea, served properly using similar principles. The bell sounded the time at which the staff would meet, talk, and function as a team. We all planned our day around these happy occasions.

And so it went. The strategy worked perfectly as it always does. Where I have suggested it, and management has refused on grounds of expense, or the ridiculous idea that their staff don't deserve good coffee, then I know that the organisation is doomed to self-destruction.

www.dbooks.org

FAQ

“OK Phil. How should I prepare coffee then?”

The beans should be rapidly and evenly roasted for between eight to twelve minutes at 390 degrees Fahrenheit (it helps to add just a little sugar). If you get a strong coffee smell at this point, the best essential oils are lost. Peaberry, being round, gets a more even roast than a normal bean. Coffee must never be under-roasted. If you taste a bitter taste in the coffee, it has been over-roasted. The moment roasting is completed; the beans must be cooled rapidly. It should be consumed as soon after roasting as possible, whilst it is fresh.

Coffee must be ground coarsely immediately before the coffee is prepared. Once grinding is done, coffee flavours will oxidise. Use electric grinders cautiously, as they can heat up the beans enough to lose flavour.

Coffee should be prepared with water that has boiled and has cooled slightly. Fanatics will suggest that only earthenware vessels should be used, and one should certainly filter the coffee. Take your time with the infusion, but never let the water cool too much.

“So, where did coffee come from? I don't buy the idea of sheep using XML.”

Coffea Arabica comes from the mountains of Yemen or Ethiopia, and probably was not discovered before AD 800. Its use was local to Arabia and the first coffee shop in Cairo didn't open until 1550 AD. It got to Turkey at around 1600 AD. In around 1650, the first coffee houses opened in London and Paris. In 1688, Lloyds coffee shop opened in London and quickly developed into a centre for the insurance business, and went on to become the world-famous Insurance Underwriters. Some historians refer to the Age of Reason as the Age of Coffee.

“Surely we should drink Espresso and Cappuccino?”

Within the coffee trade, it is well known that the Italians and French will happily buy the lowest-quality grades of coffee that the Americans and Northern Europeans won't touch. Once they have had their brutal way with the beans, all the flavour has gone anyway, giving you a shot of caffeine that merely serves to make you rush around getting flushed and angry, with your temples thumping. The real caffeine buzz depends on the subtle blend of essential oils.

“Why hire someone just to make coffee?”

When stressful events happen, people forget the importance of the Coffee and Tea ceremony in the scrabble to do stuff. It is important to entrust the task to someone who can't be 're-prioritised', and who understands the importance of what they're doing. For people to make the time to communicate, it is essential for the coffee/tea break to happen invariably at the correct time. Also, I've never yet come across an IT professional who can make coffee or tea to the required standard.

“Hmm. OK, how should Tea be made?”

Tea relies on its blend of caffeine, tannin, and essential oils. To prepare tea well involves extracting the caffeine without too much tannin, whilst conserving the aroma and flavour. This is not easy. One should use fresh water, freshly boiled, and a three to five minute infusion, perhaps longer in hard-water areas. Use one rounded teaspoonful of tea for each cup required. If it is to be drunk with milk, a six-minute infusion is recommended because the casein in the milk reacts with the tannin (Mitchell 1905). You should stir the pot during the infusion. The type of tea that works best depends entirely on the ph of the water supply, with soft water being the best. Lancashire water is legendary for tea-making due to its softness. My own preference is to use a good-quality Ceylon Tea and to use a brown glazed earthenware teapot for the preparation of tea.

www.dbooks.org

Cha: Tea-drinking for IT Developers

Phil considers and explains the importance to any development team of drinking tea, and doing it properly.

Tea drinking is important to developing software. It matters how you drink it, as well as how you prepare the tea.

It was a very long time ago, whilst working in a development team with a well-known computer company in Japan that I first realized there was more to drinking tea than dumping a typhoo teabag into a mug. One day, we'd reached that point where we were all completely jaded, when someone suggested going to see a Geisha for chakai. I had no idea what to expect and I was slightly apprehensive as I didn't know the Japanese for 'Excuse me Madam, but I'd rather keep my clothes on if you don't mind'. We piled off into a bus and ended up in a beautiful wooden house where an elderly Japanese geisha, dressed in exquisite traditional kimono, entertained us to tea. We sat cross-legged around a table as she acted as hostess with both skill and dignity, entertaining us with a stream of jokes which had all the Japanese speakers convulsed in laughter. Even I couldn't help giggling, though she could have been reading the telephone directory for all I knew. It worked like a charm. We piled back to work in a completely changed mood. I was amazed and delighted; and discovered that it was standard practice to 'clear the brain' in this way when things got stressed and out-of-proportion.

For Britain, the ceremony is quite different, but just as important. It is usually known as 'cha', the Chinese name, taken from army slang. First, the teapot; this must be brown earthenware, and has always been known as 'Brown Betty' for some reason. Brown Betty's shape has been refined over hundreds of years. It is never dribblesome. (There is a book called 'The Dribblesome Teapot' by Norman Hunter, purely about a 'Brown Betty' that dribbles.) It is brown because that color conserves the heat best. The tea is of great importance. I favor a high-quality Assam, full-bodied, black as sin, unblended. Just to describe it make me tremble with excitement. A Sri Lanka Kanoy can be wonderful, producing a luscious bright-golden infusion. Darjeeling is excellent, with its rich, fruity taste. For variety, I'd always include Keemun or Oolong. Of

course, on a hot day, a Celon tea is wonderful, served as Ice tea. People who claim they don't like tea have probably drunk only disgusting reject leaves sent by puzzled producers in India over to the Midlands of England where it is mistakenly considered a delicacy.

I have a rather unconventional liking for Gunpowder tea, rolled in pellets, and unfermented. This Chinese green tea can be sipped for hours without the caffeine effect becoming unpleasant. In London, you will always find me in Soho, in an excellent Chinese restaurant where they happily bring you endless Brown Bettys full of piping hot Gunpowder tea.

I'm very much on the liberal wing as far as the preparation of tea goes. As I have previously pointed out in a blog, the art is to extract the caffeine without too much tannin; whilst conserving the aroma and flavor. The aroma relies on very volatile essential oils. You make a mistake and the flavor has gone. Warm the pot first. Always boil fresh water (soft water is best) and infuse the tea for three to five minutes, perhaps longer in hard-water areas, stirring the pot occasionally. Use one rounded teaspoonful of tea for each cup required. You will find that a six-minute infusion is required if tea is to be drunk with milk because the casein in the milk reacts with the Tannin.

Tea should be served in a quiet room away from the phones and screens. I have never been able to get hold of a professional Master of Ceremonies, such as a Geisha, but one can take this role in turns. The Master of Ceremonies works very like a Chairman. Nobody is allowed to dominate the conversation, or to get overly technical. The ceremony ends in fifteen minutes. Any mention of post-it notes, pigs or chickens is banned.

The subject of biscuits is contentious. I have even heard tell of chocolate biscuits being served, but I think that the line must be drawn here. If biscuits are served, they should be plain, so as not to detract from the delicate flavor of the tea. Dunking is anathema.

A final word about milk and sugar: Milk is allowable, so long as the tea is prepared specially for the addition of milk. Sugar ruins anything it is added to, and destroys the taste of tea. The Health Stasi who interfere with our natural god-given wholesome human right to the occasional Sumatran Cigar would expend their beastly energies to far greater good by pursuing the disgusting habit of adding sugar to tea.

Ever since I was initiated to the idea of Tea Ceremony in development teams, I have introduced it many times with great success. Unlike Agile Scrums, the practice leads to peace, harmony and a clear head. I once encountered some resistance when introducing the Tea Ceremony to a development team of rough Essex people near Southend. They called the

www.dbooks.org

glorious Formosa Oolong that I bought for them 'Poofy Tea', a name that somehow stuck. They took a long time to come to terms with the idea that dunking was an unnatural vice, and that sugar was a vile drug. They knuckled down eventually, became converts, and ended up enthusiastically trying all sorts of way-out teas. One day, I shall tell the story of the tranquilizing effects of elderflower tea, hop tea and apricot tea in the workplace.

www.dbooks.org

Survival Tips for PowerPoint Boredom

There are occasions in all of our working lives when sitting through a PowerPoint presentation is inevitable. Fortunately, there are techniques for feigning interest, many of which have developed over hundreds of years. All you need is a handful of like-minded colleagues with a sporting attitude.

When Cardinal John Henry Newman wrote 'The Dream of Gerontius', his renowned poem about the death of an old man, he penned these lines:

'Tis this strange innermost abandonment,

This emptying out of each constituent

And natural force, by which I come to be.

Newman's description bears an uncanny resemblance to the near-death experience of sitting through a PowerPoint presentation by an IT salesman. The longer you listen, the worse it gets, until you feel like heading for the light the moment the company growth charts hit the screen.

While the first few presentations can be tolerable, the brain soon reaches a point of allergy and gives up trying to wrest meaning from the droning stream of waffle. The harder you struggle to stay awake, the more a torpor steals over you, as the brain attempts a body-wide shutdown due to lack of stimulation.

Once the IT salesman has the laptop and the screen, nothing will deter him from his speech. He will attempt to soothe you with smooth, bland phrases while, by gesture and eye contact, he aims to induce a visceral liking for his product, even if the logical part of your brain is screaming "It's shite!"

There is no alternative but to fight back.



You have to fight it!

You may think that the most effective way to do so is to sit quietly and attentively until the first PowerPoint page appears on the screen. Then, you and your colleagues would put your fingers in your ears and shout "LA, LA, LA!" until the salesman stops talking.

I thought the same myself until a team I was in actually tried it. Unfortunately, the salesman's manager had just taken our IT director to a very expensive lunch, and the atmosphere in the department became very tense as a result.

In reality, there are occasions in all of our working lives when sitting through a PowerPoint presentation is inevitable. Fortunately, there are techniques for feigning interest, many of which have developed over hundreds

www.dbooks.org

of years. All you need is a handful of like-minded colleagues with a sporting attitude.

The games people play

The first, most traditional, game is called 'J for Jesus', so-named by the choirboys who developed it to overcome the monotony of sermons. Each participant contributed a sum of money to a pool or tote, and listened during the sermon for a two- (or more) syllable word beginning with A. The boys then listened and remembered a word beginning with B, then C, and so on.

The first boy to get to J blew his nose, which was why there were so many handkerchiefs whipped out when the vicar said the word Jesus halfway through his sermon. In the vestry, if the winner could remember every alphabetical word and there was general agreement that the words were spoken, he collected the money. If his memory faltered, he doubled the tote for the next sermon.

This game was easily adapted to the IT industry, although there are few multi-syllabic words that begin with J. So whenever the participants are waiting for a word beginning with J, the tension in the room becomes palpable. I remember one occasion in which the tension got so great that the winner gave out a whoop and punched the air when the salesman mentioned 'the justification process'.

In some versions of the game, it is legal to ask a leading question such as "Would it be fair to say that it is the juxtaposition of your product with our business environment that is important?" In general, however, this is ruled out of play.

Speaking in tongues

On a number of occasions I was part of a team asked to adjudicate between suppliers over the choice of a 'strategic' product for the business. Many PowerPoint presentations ensued. On one team, we quickly tired of the Jesus game and went on to analyze the words and phrases used by the salesmen.

IT salesmen spout gibberish of a special sort, a queer language in which meaning has little place. The phrases themselves can be fascinating, particularly when you stop trying to clutch at their meaning:

Fourth-generation environment, function hierarchy analysis, structured business analysis, inevitability of amelioration, attenuation of subsequent feedback, relational flexibility, strategic framework, dynamic systems strategy, technical coherence, high-leverage area, internal resource capability, separate roles and significances of the formal strategic direction, interactive concern-

control system, calculus of consequence, functionality matrix, conceptual baseline …

The Jesus game thus evolved into a more complex sport of picking out waffle phrases. At first we tried an alphabetical system, but eventually decided that in addition to quantity, phrases would be judged on quality as well. To 'own' a phrase, you had to ask a question that included the phrase that the salesman had just used.

Using one of the examples above, a participant might ask: "Would you consider that the separate roles and significances of the formal strategic direction had a particular value in the context of your product?" This signaled the others that the phrase was 'yours'.

The rating process involved writing down the phrases and getting participants to rate them on their absurdity. As time went on, we collected the best and seemingly most powerful phrases until we had a phrase bank that could be used to generate automated PowerPoint presentations. This was the famous 'waffle generator,' which spread like a virus through the industry.

Use your assets

The waffle game electrified the presentations. The salesmen would be surprised to see a row of keen faces in the audience, eyes bright like buttons, lapping up every word and asking intelligent questions. Unfortunately, the very nature of the game made it go sour, because our apparent interest only stimulated the salesmen to talk more.

The great virtue of the Jesus game was that, after the J word had been spoken, the audience lost interest in the presentation. Sensing that something had gone wrong, the salesman truncated his presentation and shut up. The waffle game had to be modified so that only the first ten minutes of the presentation were allowed.

I have come across other games too. The 'situation' game offers considerable scope. The audience divides into two teams that take alternate presentations and try to induce the speaker to say the word 'situation' the most number of times. There are few rules, and the ingenuity of the cheating can be fascinating.

My best team ever included a young graduate trainee named Louise who was quite well endowed. We worked out a technique in which Louise would lean forward slightly in her chair every time the salesman said 'situation'. Like one of Pavlov's dogs, the salesman would unconsciously increase his use of 'situation', until his speech was peppered with the word.

www.dbooks.org

Even in the age of the iPod, when no one need suffer from PowerPoint presentations, you will find me chuckling appreciatively in my seat, taking lots of notes and listening attentively. The collection of phrases of complete techno-babble can become an intrinsic joy forever, even without the money and the competition.

www.dbooks.org

The Joy of IT Meetings

One of the real pleasures of any IT project are those endless meetings that drone on without mercy or action points and leave you even more confused about the project's direction than you were when you went in. It doesn't have to be like that…

As the concentration span of the average adult human is less than forty minutes, I have always puzzled over the reason why meetings last any longer than this, particularly in IT where the concentration span strains to reach even these giddy lengths.

You will be astonished to know that there is actually a correct way to conduct business meetings, as you have probably never been told about it, and never experienced a properly conducted meeting. I will therefore explain.

When the Second World War was imminent, Britain was totally unprepared, being in one of its occasional utopian dream states we now recognise as Blair-ism. Fortunately, the government of the time was out of tune with this zeitgeist of torpor and urgently put in place the means of developing technological breakthroughs to aid the forthcoming war effort. A number of scientific projects were initiated by a group of scientists grabbed from the universities, nicknamed the 'Boffins', and given government positions to develop the means to win the war. Radar, code-breaking, and aero-engine technology were the most visible achievements of the 'Boffins'.

Operational Research was, however, the one that had the most immediate effect, and which the Americans wanted as much as radar and the jet engine. The science spawned 'time and motion', the modern science of Project Management, and a huge methodology of coordinating enterprises, whether commercial or military. Its effect was more far-reaching even than the jet engine.

The conduct of meetings was an early success of Operational Research. The boffins noted that, within less than an hour in a meeting, managers or army officers entered a semi-hypnotic state almost the same as REM sleep, but still enabling them to pour out waffle like 'we are addressing your issues and concerns'. They noted that the productive work of a meeting was inversely proportional to the length of time taken by the meeting, past this magic time

period. After a great deal of research, they came up with the following rules, taken from existing best-practice ...

- No meeting should ever take more than an hour, and its length must be determined and published beforehand.
- Nobody should be allowed to speak for more than three minutes at a time.
- The meeting should have an itemised agenda.
- All documents to be discussed or produced should be 'discovered' (i.e. shown or copied to all attendees) in good time before the meeting.
- No decision can be reached by the meeting on any item that is not on the agenda.
- The meeting must have a predetermined quorum A meeting must always have a Chairman (man in the sense of human, girls).
- The Chairman is responsible for the good conduct of the meeting.
- For the course of the meeting the Chairman represents the organisation and has the authority and respect of that organisation.
- All actions and decisions of meetings must be minuted, and previous minutes must be explicitly approved.

These simple rules expedited the decision-making process in British industry and the Services to such an extent that, by way of example, we had more serviceable fighter planes after the Battle of Britain than before, and were able to achieve the evacuation of Dunkirk, the scale of which was previously unimaginable.

Now cast your minds back and try to think of a meeting in any IT department anywhere that conforms to any of these rules. They are generally pandemonium, lasting several hours whilst all the 'Alpha Males' wrestle for control of the WhiteBoard or witter in a stream-of-consciousness way about anything that drifts lazily into their brains. After forty minutes of this, we still seem to be awake, but in fact we have as much power of original thought as a Zombie. After a couple of hours we lose the will to live. The reason that this is ever allowed to happen is because this sort of meeting puts power in the hands of people ready to exploit this foolish way of working.

If a thought cannot be expressed in three minutes, it has not been properly formulated, and human groups have the uncanny ability to find solutions within seconds before or after the end of a meeting, whatever its length.

www.dbooks.org

I have been accused of being an old and cynical man, but I can assure you I used to be young and cynical. It was whilst still in this blessed state that I had the good fortune to be tutored by an expert on the best ways of influencing an organisation.

I had almost accidentally ended up as a district councillor, when I met my tutor. It was an efficient and well-run organisation. This puzzled me greatly as the councillors were like an animation of a Hogarth cartoon, self-serving, idle and venal. They never 'read their papers' and only became animated when sensing a new trough into which they could put their metaphorical snouts. They would have struggled running a pair of tights let alone a tier of government.

I went to the Chief Executive officer of the council and asked him why he allowed council meetings to go on so long. After a great deal of prevarication, he realised that the game was up and explained the technique, which I have found extremely valuable in getting my way in any organisation.

Imagine you have a public-sector IT project going over-budget that needs more funds to complete. Fortunately most people do not really understand the magnitude of difference between £100 and £1000; in adding a 'nought' to a figure. Meetings therefore started in the morning with a good long agenda that has to be finished in the morning. The meeting starts with a series of trivial decisions on whether the council should purchase a new lawn-mower, or a few wheelie-bins. Councillors can understand such things, and are full of blood-sugar from their hearty breakfast. They want to cut a dash in front of the reporters from the local paper. You give them their head, caring little for their decision. The morning drags on and you introduce decisions on more expensive and abstract items. They lapse into oxygen debt. There is a glorious time as lunchtime approaches when the meeting has reached that joyous deep-hypnosis, and their souls yearn only for lunch. It is then that you slip in the request for an extra £1,000,000 for your recalcitrant IT project. The money will be voted through with a minimum of fuss, in favour of completing the agenda and getting out to lunch. The news reporters will already be in the pub.

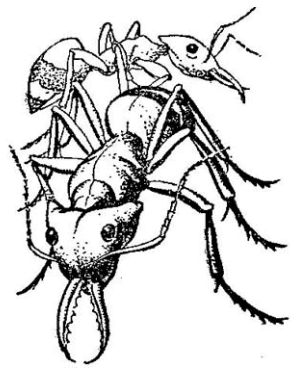
So simple, and it really works

www.dbooks.org

Why do we call them 'Bugs'?

In which Phil uncovers the true origin of the word 'bug' and offers up a celebratory verse to mark the occasion. In the process, he spots another potential gap in the IT market.

I've read a great deal about the origin of the word 'bug' in computer software. You'd have thought the argument was settled ages ago when everyone agreed that, in 1947, the Harvard Faculty at the Computation Laboratory traced an error in the Mark II computer to a moth trapped in a relay, thereby coining the term 'bug'. This bug was carefully removed and taped to the logbook, which still exists.



Nasty bugs ...

Actually, the incident was recorded only because it was the first time a bug had been caused by a real bug. It was amusing at the time only because the term was in common usage. In fact, the word had been common in the telephone industry for many years. There is an apocryphal story that the word was coined after the noise on the telephone line, which sounded like a cockroach. Sadly this is all nonsense. The word has been used in engineering since the nineteenth century.

The word 'bug' actually is short for Bugbear, sometimes found as Bugaboo. Its meaning is much closer to 'Gremlin', where the people who worked on engineering prototypes often grew to suspect that the problems were due to malicious spooks. I sometimes still hear it said that a particular piece of

software is cursed with malicious spirits. The 'Bug' or 'Bogey' part of the word is traceable back to the fifteenth century in the meaning of 'Hobgoblin', devil or ghost. In East Anglia particularly, the word 'Bugbear', first recorded in the sixteenth century, is still used in referring to problems with machinery.

A Bugbear is a malicious spirit, and the word has nothing whatsoever to do with insects. In celebration of the finding of the real meaning of the word 'bug' I offer the following verses

I always hope there is no place
for bugbears in my database
and wish that no-one ever sees
a bugbear in my foreign keys.

So bugbear go! for I don't need'ya
bugging up my stored procedure
causing table locks, and things
suddenly truncating strings,
endless loops and other terrors
like rogue spids and rounding errors.
Bugbears causing sudden death
in a tested UDF.
Routines that have worked for days,
fail in unexpected ways

It's the bugbears at their game
surely I am not to blame.
Is the database offline?
It's the bugbears' fault not mine,

I suspect that there is a gap in the market for a new type of utility software, one that drives out the malicious spirits infecting a piece of code. To this end we announce the availability of two software applications: the first one removes the curse of bugbears from a piece of software, and the other one curses other people's software with bugbears. Any volunteers for helping with Beta Testing in both categories should contact me.

www.dbooks.org

The Ballad of the Tuple Relation

Phil turns post-modernist poet in order to seek out the true meaning of a particularly twisted explanation of Grouping.

I was peacefully pottering away at my workstation, contentedly spawning a daughter thread or something similar, when Robyn Page popped in. She'd been wrestling with the Grouping Workbench and trying to come up with a copper-bottomed explanation for what the GROUP BY statement does. I'd been trying to help out in my usual arrogant, paternalistic way. This time she innocently asked me to interpret the following passage, which is an explanation of Grouping in O'Reilly's Relational Database Dictionary. ('*Helping to ensure the success of your database projects*' gushes the back cover.)

Grouping

Let r be a relation and let the heading of r be partitioned into subsets {X} and {Y}. Let the attributes of {Y} be Yl, Y2, ..., Yn; also, let {X} not contain any attribute called YR. Then the grouping r GROUP ({Y} AS YR) is another relation s. The heading of s consists of {X} extended with an attribute YR of type RELATION {Y}. The body of s is defined as follows: first, let z be the result of r WRAP ({Y} AS YT). Second, for each distinct X value x in z, (a) let yr be the relation whose tuples are all and only those YT values from tuples in z in which the X value is x; (b) let t be a tuple with X value x and YR value yr (and no other attributes); then, and only then, t is a tuple of s. Note: Given a relation r and some grouping of r, there's always an inverse ungrouping that will yield r again; however, the converse is not necessarily true. Example: The following expression denotes a grouping of the relation that's the current value of relvar SP:

SP GROUP ({ P#, QTY } AS PQ_REL)

That grouping is a relation spq of type RELATION {S# S#, PQ_REL RELATION {P# P#, QTY QTY}}. Relation spq contains one tuple for each distinct S# value currently appearing in. relvar SP, and no other tuples.

You might be thinking 'Phew, hot stuff', but I couldn't make much sense of it. In fact, were I ever to be abducted by extraterrestrial aliens, I will expect them to address me in similar unfathomable clicks and whirrs before conducting their foul experiments upon my person.

It was whilst I was fruitlessly re-reading the definition for the fifth time, trying to decipher some meaning from it, I began to wonder whether, in fact, it was a poem. I give the first two resonant stanzas

> Let r be a relation and I will tell you why;
> let the heading of r be partitioned into subsets {X} and {Y}.
> Let the attributes of {Y} be Yl, Y2, ..., Yn; also
> let {X} not contain any attribute called YR though.
> The heading, now, of s consists of {X} extended by
> an attribute YR of type RELATION {Y}.
>
> The body of s is defined as follows: See!
> first, let z be the result of r WRAP ({Y} AS YT).
> Second, for each distinct X value x in z,
> (a) let yr be the relation whose tuples be
> all and only those YT values, as anyone expects
> from tuples in z in which the X value is x;

... and so on. Quite the sort of output one might expect of a post-modernist poet.

But then, after having extracted as much amusement out of the paragraphs as I could, I felt a certain twinge of sadness. Robyn and I are both firmly of the belief there is nothing in relational database theory or practice that cannot be explained to an interested child. SQL is merely a formalised dialect of English that describes action. I firmly believe that a relational database can be made so simple that any educated person can use it. Codd formalised the theory for the automation of existing processes that existed long before the first computers. It is the language of information retrieval.

www.dbooks.org

Common Law, and the Need for Restraint

> *Mastery of the database, knowledge of the underlying OS and network, good communication skills, relentless attention to detail ... these are all admirable skills to have in a prospective DBA. Yes, yes, but what Phil really wants to know is ... can he do the Vulcan Nerve Grip?*

I'll never forget the best DBA I ever appointed, because he thoughtfully taught me an effective technique for removing wheel-clamps from cars. It was after the interview, I seem to remember. We walked together to his car to get a file. There in the boot was some interesting equipment collected together to meet the needs of the urban DBA. He described the technique. I worried slightly over certain aspects. 'Is this legal?' He looked at me pityingly, and gave me a stirring speech about our duty to fight oppressive legislation, including the line, quoted from the Magna Carta.

> *'No free man shall be seized or imprisoned, or stripped of his rights or possessions, or outlawed or exiled or deprived of his standing in any other way, nor will we proceed with force against him, or send others to do so, except by the lawful judgement of his equals' ...*

Hmm, he'll go far, I thought. I was right, too; he proved to have an unparalleled zeal for mastering corporate regulations and compliance issues.

The other DBA who proved to be the master of any crisis was once a specially trained riot policeman. Bill changed career when the police force 'went soft'. His highly trained skills of subduing rioters were no longer prized, and a future as a desk-bound social worker didn't appeal. A career as a DBA beckoned.

Bill commanded instant respect amongst the developers. He never had to do any more than give 'That Look'. This is because he was an instructor in martial arts and a champion weightlifter. He could lift the heaviest server in and out of the racks without effort or assistance. All the recalcitrant servers were soon pacified with the ease with which he once subdued over-excited Trotskyites

Once, whilst chatting in the server room, he showed me a grip which could restrain anyone so effectively that even talking, let alone struggling, was impossible.

One dreams wistfully of using this real-life Vulcan Nerve-Grip.

Phil: I gather that the development team has issues over the interface between the database and application layer?

Developer: Well, yes. I believe strongly that we should be using Hibernate, and should have table-level access to the databases so we can 'persist' our objects in the 'data repository'. We can spray un-optimized dynamic SQL all over the database without any clear idea about what is actually being generated and can Nnnngggh! … (thud)

Phil (wiping his hands) Any other clever ideas?



It looks like he used theVulcan nerve grip

www.dbooks.org

I've never had to put either technique to the test. For the exceptional DBAs, the important skill is to plan for every conceivable problem and have a strategy in place to deal with it, even if the strategy involves crowbars and martial arts. For some reason, preparedness for an event makes it less likely to happen.

www.dbooks.org

Phil Factor on the Law

Taking the adversarial system to task with vigour and courage, Phil Factor gives us his view on how to handle the trials of litigation if all cannot be solved by a hearty pub lunch.

I was once taught the correct way to halt the attack of a charging bull terrier. Stage one is to **show no fear**; to stare straight into its eyes and dare it to proceed. If that doesn't do the trick, then in stage two you grab its lower jaw so it cannot bite, and roll it on its back. I once had cause to put this technique into practice and I can vouch for its efficacy.

It is a useful life-skill and a technique that can just as easily be applied to your dealings with lawyers. In my dealings with the legal profession, I have never had to proceed to stage two, though I am yearning to give it a try and am confident that it would be effective. It helps, with both dogs and lawyers, to handle them regularly and to engage in play fights so that you do not experience fear.

In my IT career, I have had several involvements in litigation, usually on the receiving end. If I've learned one thing, it's that that there is a yawning gap between law and justice. My advice to anyone thinking of engaging in litigation is simple: **don't**. Unfortunately, however, frustration and anger can sometimes override commonsense. In the grip of adrenaline, the litigant can easily convince himself that, in his trusty lawyer, he has found an instrument of sweet and righteous revenge. In fact, all he has found is a highly effective drain on his bank account.

Some time ago, I wrote a very fine commercial communications package. It was for the secure encrypted communications of corporate data via X25. It included a special PCI card with a slave processor for managing the actual nuts and bolts of the rather complex synchronous serial communications, and a PC application for sending stuff and setting the various parameters. It worked very well and I was enormously proud of it.

Then the company I wrote it for went bust, and the trouble started.

Some of the staff of the late-departed company took the source code and the design of the PCI card, started up a new company and began to market and sell

my application under a new name. This probably wouldn't have mattered much, had it not been for the fact that a third company bought up the assets of the dead company, including the designs and code that I'd written, and then engaged me to develop the product.

Very soon, there were two virtually identical products on the market. The new owner was understandably aggrieved, especially after I obtained a copy of their product, disassembled it, and was able to prove that it was identical to mine. It looked like an open-and-shut case and so off we went, full of confidence, to the solicitor's office, armed with a file of documents. He smiled brightly, took our instructions, and briefed a leading London barrister.

After a while, we were summoned to the barrister's chambers, in Lincoln's Inn. It was like a mediaeval cloister of Oxford College. The barrister was articulate and affable, just as anyone would be on his hourly rate. We went through the evidence, and he absorbed the technical intricacies as though he had been programming X25 all his life.

"Oh yes," he admitted once we'd finished, "you have a case, a good case. But…" he paused for a few seconds, thereby adding a few more pounds to his charges.

"But what? "

"We could ask for an injunction to stop them selling the package … "

"Yes, yes," we agreed eagerly.

"… but the problem with that is that you might lose the subsequent court case, and the other company would have unfairly lost all that business; it would have suffered loss as a result of the injunction, for which you would then be liable."

"But we won't lose!" we protested.

"I'm just telling you the system," he explained. "There have to be checks in place to prevent malicious and unreasonable injunctions being requested. If you ask for a preliminary injunction to stop them using your product then, as a security, the Court will require you to deposit a sum of money equivalent to the amount of their subsequent losses, just in case the action goes against you. The other party is not going to underestimate their losses are they? They will suffer enormous financial hardship, and might even go into liquidation. I suspect that they will ask for around a million pounds. That is money you have to find before you are awarded an injunction. "

"But it is an open or shut case. Surely, it is unfair to assume we are going to lose?"

www.dbooks.org

He looked sheepish, "Oh dear. The legal system tends to decide fairly, of course. But then, a court action is always a gamble. Even the best case can go against you. "

We looked at each other, hurriedly shook his hands and left the chambers. Somehow the idea of gambling a million pounds didn't seem such a good idea.

We have an adversarial legal system. That means something like trial by combat. It is a slight advance on dipping the plaintiff and defendant in the river to see which one floats the longest, but not by a large margin. It is more like hiring a mediaeval knight to bash it out with an adversary hired by the other side. If your knight is not on top form, or has neglected his sword training skills, then you will lose, however just your cause.

Our only option was to curse the wretched company that stole our property, and shake our fists at them. This proved agreeably effective, because within a year they took their company and their unfortunate investor's money across the river Styx, and this time the liquidation was final.

As I write this, I am locked in a legal dispute with a large government organisation. The details don't matter beyond the fact that (of course) I am entirely in the right. However, any legal battle would be like entering a bleeding competition with a blood bank. Legal victory generally goes to the party with the deepest pockets. Legal Aid used to be a wonderful instrument in allowing a small minority lucky enough to get it to win actions against those who didn't. Nowadays, however, the only mediaeval knight you can hire with legal aid is desperate, slow-witted or suffering mental hernias. For the rest of us, guerrilla warfare is the only possible response.

Once, an IT company was foolish enough to attempt litigation with me over a dispute about a contract. It could all have been settled over a beer in the pub, but tempers flared and they attempted a court claim. I put in a counter claim and we settled down to trench warfare. The only difference was that they hired an excellent but costly solicitor, and I settled down contentedly with a pile of legal textbooks.

I started to send in a steady stream of questions, clarifications, requests for discovery of documents, and so on. This delighted the solicitor, as every request had to be acted on, and each letter he sent out in response meant another charge to his client. We soon became good friends on first-name terms.

At one point, I got a bad bout of flu and was out of action for a few weeks. He anxiously contacted me, and even gave me a few tips on the sort of questions I should be asking, or points that should be clarified. After a while, the initial passions cooled, and any court action was still months or years away.

Meanwhile, their bills mounted steadily and alarmingly. They began to put out peace feelers, which I ignored.

Then, out-of-the-blue, another company expressed an interest in taking them over. Of course, nobody is interested in purchasing, or investing in, a company that has court proceedings against them, or who are engaged in litigation. It is an unquantifiable. They looked through the file, saw the legal expenses, and told the company that they had to settle before any deal could be reached. Their solicitor was delighted and leaked the information to me.

It is a rare but delightful experience to sit around a table, smiling affably at people who are grinding their teeth with bottled-up anger, at being forced to offer me a substantial sum of money to settle a claim.

The whole experience did, however, convince me that in the world of commerce, litigation is just a means of feeding lawyers. And they are always hungry. Negotiation is the answer. Much can be achieved at a pub lunch; and if that fails, then a vile, blood-curling, curse can, in my experience, be most effective.

www.dbooks.org

Santa's SLA



A cheering festive tale of the SLA that didn't deliver all it seemed to promise.

For reasons too personal to be interesting, Christmas brings out the worst in me. My happiest Christmas was spent covering for a DBA colleague. I was on double time, and sitting in a nice quiet spot in the Server Room where only swipe-card holders could enter. (We somehow neglected to issue swipe-cards to the management. It must have slipped our minds.) I was contentedly sipping a sherry and gazing at the performance monitor, as one does with a Lava Lamp, relishing its calming, hypnotic, qualities. 'God rest you merry' ... A deep peace was all around me.

Just then, a disk failed in a SCSI array on a critical database.

I knew that, if another one went, it meant big trouble. I was working for a Telecommunications company, and Christmas Day generated one of the peak

revenue days for the company in the entire year, as people phoned home. Putting down my sherry-glass, I rummaged around the store cupboard to slot another one in. There wasn't one. I phoned the DBA at his home. "Hello, it's Phil here. I'm at work." There was a scream of overexcited children in the background, blowing off an e-number binge by quarrelling noisily.

"Ah yes," he said wistfully.

"18 Gig SCSI drive gone tits," I told him, lapsing into esoteric jargon.

"Ooh," he exclaimed hopefully, "do you think I ought to come over?"

"No need." I responded, cruelly. Christmas, as I've already explained, does that to me. "Just tell me where the spare 18 Gig SCSIs are kept. Even I know how to hot-swap a drive in a RAID array."

"Hmm. No spares, we don't carry them any more. We were hoping to get a new RAID array with larger-capacity drives after Christmas."

"So what is the solution?"

"We have a service-level agreement with the supplier that guarantees a two-hour response time. Just give them a ring and let them do the worrying".

After a long wait, ringing the supplier, then probably the largest providers of Enterprise-level servers in the world, a voice answered.

"Hello," said the voice irrelevantly.

"I'm with xxxxx Company, and we have a maintenance agreement with you for our servers." I quoted the reference numbers.

"Mo," There was a tapping on a keyboard at the other end after which he came back and asked what the problem was.

"18 Gig SCSI drive has handed in the dinner pail. It has croaked."

"Oh, I'll just put you on hold while I put you through."

Then the 'calming' music ...

... When it snows ain't it thrilling
Though your nose gets a chilling
We'll frolic and play
the Eskimo way
walking in a winter wonderland ...

Aaargh!! Any residual pining to be celebrating a festive Xmas in the bosom of my family died at that moment. A 'Bob Cratchitt' chill entered my heart.

www.dbooks.org

The music suddenly stopped. I was through to someone technical just as the life force was starting to drain away.

"Sorry to trouble you today." I started.

"No trouble at all. I'm on double time. It's quiet here. I'm only too pleased to speak to a fellow human."

I wondered briefly who he had been speaking to before. A salesman? I explained the problem and read out the full product code.

He sucked through his teeth. "Sorry, we can't do you. There isn't another one of those in the country. They'll have to come over from the States. It'll be ten days, I guess."

"I thought we had maintenance contract with a two-hour response time with your company."

There was a pause. I assume he was drawing himself up to his full height. "Well," he said rather huffily, "we did respond within two hours. We responded rapidly, and efficiently, by telling you it was going to take ten days."

I sighed and put the phone down. It was the season of goodwill after all, and I assumed the guy I was speaking to had overdone the Office Party. I had a brief dream of trudging through the Bleak Mid Winter snow like good King Wenceslas on the feast of Stephen, looking for 18 GB SCSI drives in PC World.

However, the solution proved to be simple. The test server, luckily, was an identical twin of the production one, and kept synchronised with it. A drive was taken from it and popped in the production RAID array. Within an hour I was back at peace with the world, sipping sherry and listening to Rammstein through headphones, as I ministered to the server.

Ten days later, a dispatch rider solemnly ran up the steps of the company into reception, carrying an 18 Gig SCSI drive, and proudly made us sign for it as if it were a holy relic. We dropped it straight in the bin, because the new Raid Array from our next IT supplier was clicking away happily in the rack.

I've never since retained any faith in maintenance agreements that promise rapid response times, unless they spell out exactly what constitutes a response. Ours didn't.

www.dbooks.org

Tomorrow will be our Dancing Day

Phil provides a classic IT ghost story.



"Come on," Great Uncle Phil, tell us a ghost story!"

I'd settled down in front of the log fire in the library one evening, and was staring gloomily at the embers, whilst sipping an agreeable port. For me, Christmas is a bittersweet time, a time to contact friends and relatives, and reflect on those no longer with us. The children had burst in on my solitude from the light and noise of the drawing room.

"Ghost Story? Hmm. I can tell you a few IT horror stories. TJX and the 45 million customer records, the NHS, or the missing Child Benefit database."

Outside, there was a sudden clap of thunder and a squall of rain beat on the windows. The faces of my little great nieces and nephews went pale, and they trembled, despite the proximity of the blazing fire.

"No," said little Eve nervously, acting as spokesman, "it has to be a ghost story, and one about computers. We always have ghost stories at Christmas. You do believe in ghosts, don't you?"

"Well, we all occasionally experience things that we find hard to explain." I paused for a moment to collect my thoughts. "It happened to me once at work, in the early nineties, when I was an IT manager in an international corporate near London."

The children sensed a story, and began to settle down by the fire.

"David was one of that breed of programmer one doesn't often see nowadays. When he was trained, there was much less peripheral stuff to learn, and so undergraduates concentrated hard on the basic craft of programming, on algorithms, on getting to the heart of business processes. He was a good programmer."

"Was he a ghost?" asked little Oliver, looking slightly bored.

"I'd never worked with him, but I'd seen his work. It had fantastic style. He could use Cobol or Fortran to create programming structures that were effective, simple and robust. Unlike lesser talents, he didn't rush in, but took time to plan, to see the big picture, to spot what was important and what wasn't."

"… but he wasn't a ghost." commented Oliver with some disappointment.

"Ah, you may say that," I retorted, "but one day, he was hit by the 5.47 train from Liverpool Street."

There was an appreciative silence from my descendants. "Was it horrible?" asked Rupert, eventually.

"It was, as a matter of fact." I said. "Very messy." I paused, flinching at the memory of the inquest. There was a frisson of excitement in front of the fire. "David had been in the team that developed one of the most important projects we'd ever undertaken. He worked well with understanding managers, but for this job he had the misfortune of working under one of the worst managers in the business - a man called Nigel Savage.

Nigel had no idea how to make a team work well. You can train a manager, but some people will always revert to black arts instinctively. Nigel was one of these. He saw that the key to the success of the project was David - so he put him under pressure in many subtle, but ruthlessly-unkind, ways."

www.dbooks.org



The 5.47 from Liverpool Street

"Shouldn't Nigel have been hit by a train?"

"We'll come to that. The inquest hinted at David's suicide. I knew more than most about the vindictive spite and black politics that Nigel had used against his team, and David in particular, but somehow I still didn't think it was. Nevertheless, we were all very shocked, and all wondered if we'd done all we could to try to prevent it. His computer stood unused at his desk for a couple of months. For us, it was like an extended wake. David had spent so long at his computer that it seemed to us as if part of his soul had transferred into the inanimate object and its circuitry. Nigel sensed nothing at all besides the rank injustice to himself of having a key programmer in his team dispatched by the grim reaper. He felt that it was most inconvenient."

"Not a twinge of guilt?", asked Eve,

"None."

The children gave an involuntary shudder. A gust of wind howled in the chimney, as if nature itself reared up in revulsion.

"Eventually, we got some industrial trainees in the department. They were undergraduates, working under supervision. One of them joined Nigel's team and took over David's computer. Things seemed to return to normal and we got on with life.

It was a few weeks later that I was in the canteen, chatting to one of the university lecturers who'd come to supervise the students. "Our student has come on a lot," he said. "She used to be a bit of a plodder with her programming; much better with business analysis. What do you think, Phil?" I cast an eye over the printout. It was good; very good; too good; and I recognised David's style immediately.

"Ooohhh!" said one of the great nephews, "automatic writing?"

"It gave me quite a turn I can tell you. My first thought that it was something of David's on the hard drive they'd missed, but it was written in response to a change request that had come in only a couple of weeks ago. All the quirks were there; the way he named subroutines, the style of comments. I gulped, but merely said 'She has learned very quickly from a great tutor'. I thought it best not to elaborate."

"It was as if that small part of David in that machine was still striving, as he always did, to finish the project; that determination to succeed, transcending mortality."

"Is that all?" asked my great nephew Tom, disappointed.

"If only it had been." I sighed. "The whole department knew that something was horribly wrong but Nigel sensed nothing. His project was suddenly on-schedule and he was going to get a promotion out of it. That was all that mattered to him. And so it was. When the project finished, we put David's PC in a cupboard in the Computer Room. None of us could explain why."

"And …?"

"Well, over the coming days, things started going wrong for Nigel. The word spread from the surviving members of the development team about Nigel's petulant and bullying manner, and his insensitive management style. Nigel's acts of vindictiveness, spite, and black politics against his team, and David in particular, became ‘Water-fountain' gossip. After a while, Nigel ate alone in the canteen, a pariah. An ambitious manager must never have a reputation as a martinet. 'He gets his projects in on time, but at what a cost'! All the managers smelt the stench of a career in freefall, and avoided him.

Despite the ultimate success of the project, Nigel gradually underwent a striking change in demeanour. Gone was the cocky, ruthless action-man, replaced by someone who seemed … err … haunted by anxiety. He seemed

www.dbooks.org

distracted, looking over his shoulder, as if fearful of shadows, and dimly seen shapes in the periphery of his vision.

"Had he seen a ghost?"

"One morning, we received the news that Nigel had been mown down by the 6.45 Yarmouth train. It was a ghastly shock, even if a wit in the department was heard to remark 'It couldn't have happened to a nicer man'. Two deaths, even in a large IT department, was a shock. The Company Directors decided that the pressures of delivering such an important IT project were probably to blame. There were many repercussions.

The train driver was the man to feel most sorry for. His testimony at the inquest made one suspect that the shock of what had happened had unhinged him, for he swore to the coroner that, as he rounded the end of the Marks Tey Viaduct, he saw two men on the crossing, one struggling as if to get away, and the other grey, impassive, almost translucent: As if, he added, locked together in a macabre dance. But he swore that he hit only one man.

It was my sad job, as Nigel's colleague, to clear up his desk, and his PC. I sorted through his emails to delete or archive everything. It was then that I came across…*the emails*." I gave a pause and looked at the audience. Their pale faces looked up in delighted anticipation.

"There were five of them; Simple in their content and sent to to Nigel in the days leading up to his death. Apparently they were from David, though that was, of course, impossible. Can an email somehow get stuck for months in a relay somewhere? The first email read simply: *'We meet in a week.'*

Another one read: *'Three days to go, Nigel*'. The final, haunting email read:

'Tomorrow will be our dancing day'...

The room lapsed into a hushed silence.

"So what happened," asked Rupert finally "did you bury the haunted computer?"

"Bury it? Of course not. When the department disposed of it, I bought it from facilities, as a memento. It is in the attic here if you'd like to see it."

Eve gave a squeak of fright. At that moment, Great Aunt Jenny burst into the library. "Now there you are, children, time to wash your hands for supper. And I hope that Great Uncle Phil hasn't been telling you one of his silly stories." The children streamed out into the light, hubbub, and colours of Christmas Dinner.

www.dbooks.org

The Data Center that Exploded: A Halloween Tale

A while back, in a Simple-Talk editorial meeting, someone bet Phil that he couldn't come up with a Halloween story. To our surprise he said he could, as long as he didn't have to keep to the strict literal truth. In the end, he came up with a story about a story.

"There's always a good reason why Database Applications go wrong. It is always down to human error or mechanical failure."

The older I get, the more I feel compelled to contradict confident assertions.

"You might think that," I replied, "but you probably don't remember that time back in 1992 when the overnight clearing system for the Imperial Bank crashed, and couldn't be restarted for twelve hours."

"No," he said firmly.

I wasn't discouraged.

A Prank is Plotted

Have you ever had a garrulous colleague at work? If you have a pig-pen you can escape; but in an open office area it is torture. Alistair hadn't enough work to occupy his day. Out of work hours, he lived alone; at work, he chattered unceasingly. To make things worse, Alistair was cocky. He was a contractor, and like many of his breed, he spent too much of his energies telling us how clever he was. When you are working ten hours a day in close proximity to a guy like this, it can get wearying. To make matters worse, he was too prone to playing 'practical jokes' on people, and there was more than a whiff of malice about them. I loved to escape occasionally to the Data Centre to call in on 'Bill the Minder', the manager in charge.

It is hard to describe the sheer immensity of the data centre of the Imperial Bank at that time. There was floor upon floor of expensive steel cabinets, weird robotic tape systems, and curious islands of obsolescent hardware, all cast in

subdued lighting against cream-painted concrete. High up on a mezzanine floor shone the windows of the operators offices. There, in a strange middle-earth grotto lived a typical breed of operator; opinionated, slightly belligerent, but single-mindedly devoted to the health of the systems in their care. It was there that Bill had his office and ruled his kingdom. For me, it was paradise; Technology-play. Bill and his team could check and control everything. Computers, lights, air conditioning, the power to the systems: Screens were set up to monitor everything that SNMP could reach.

Out in the cavernous hall of the old data centre, there were countless indicators for alerts and activity, so that wherever you were, you could be aware of a sudden problem. In the semi-darkness of the windowless building, one could imagine being in a strange satanic cathedral.

It was while I was standing on the tiled floor, staring up at the lights of the office, that the idea came to me of a Halloween prank. I thought that, maybe, it would earn us all some peace from Alistair's prattling.

I went up the stairs to see Bill. "Would you be able to set up a bank of disk lights and make them go off when I give a signal?" He looked contemptuous. "We already have them. We set some up for the last visit of one of the directors of the bank. They were worried that there wasn't enough so see so we lashed them up." I told Bill of my plan for Alistair. He agreed to do everything he could.

Bugbears

"Y'know," I started "how the word 'Bug' came about don't you?" I asked Alistair innocently. He and I were working away in the third-floor switch room where the test database was kept.

"Yes, everyone knows that. An insect was found in a computer. There was a fault and a moth had to be extracted from between the contacts of a relay. They wrote in the log that it was the first actual time that a computer had been 'debugged'. They stuck the moth in there. That was where the term originated."

"Not at all. They were just amused at the time because they all used the word debugging already, and they wanted to record the first time that a real bug caused the damage. In fact, Radar Electronics used the term during WW2. And before then, any mysterious problems in machinery were called 'bugbears'. The Welsh referred to them as 'Byg'."

"Oh yes?" Alistair sneered. One of Alistair's tireless themes was his militant scepticism. He railed against all superstition.

www.dbooks.org

"It is an interesting word. You can track it back hundreds of years. People suspected that any machinery, even a cart, could be infected by Bugbears, which were a sort of mischievous spirit like a gremlin. 'Hobgoblins and buggybeares'. Even the millers in the old watermills would talk about 'buggybows' getting into the mill works. We have references to the word in the oldest printed books in English."

"Nah. It's an American word. I think Thomas Edison used it."

"Well, so did a scholar in the twelfth century, writing about devils and spirits: There was a Celtic god, Bugibus, who was associated with avenging or malicious spirits."

"Phil, you make up the most incredible rot, you really do."

I shrugged. "Look it up yourself, it is all in the Oxford English dictionary."

He looked doubtful, which was a shame as I could have won some money on a bet.

"I've been doing some work on the subject myself at the Harry Price Library at the Senate House." I added. "The custodian ferreted out several boxes of miscellaneous papers, mainly bits of old books from Powys Castle library that had never really been sorted since their bindings fell apart. On one of them was a poem; an 'Elegy to Buegibos'. It was written on what seems to have been the flyleaf of a fifteenth century printed book. I photocopied it and had it transcribed, and partly translated, by a friend at the British Library. Buegibos seems to have been that vengeful Celtic spirit."

"OK," interjected Alistair, challengingly, "what did the poem say?"

"It starts off 'Darkest Dark, and deepest deep, Bugbears rise from deaths last sleep, Come, with the funeral's tolling bell, from vampire's lair and shrieking hell' …"

At that moment, the lights went out. It is a simple trick, which is done by leaning against the wall and tapping the switch with your shoulder-blade. "Oh blast, the condenser in the overhead lights is a bit dodgy, I must get it fixed." I lit a cigarette lighter, and the room became suffused by an eerie glow. "I'll try switching on and off again." Before he had time to look, I had the lights on again.

"All rot, this is", spoke Alistair, slightly hotly, I thought.

"It could be." I shrugged. "I'm a historian, not a spook-chaser. The poem is genuine enough. Let's try it out in the data centre and see if it really does infect big computers."

"You must be crazy! We'd be sued for millions if we caused damage there."

"By reading an old translated poem in a server room? What would we be convicted of? Witchcraft? It would be simple, anyway, because 'Bill the Minder' isn't around after seven in the evening and the production team are watching 'Big Brother' on one of the monitors then. I'm allowed access to the server room, and I have a swipe."

Alistair looked doubtful. "Worried by a silly poem now are we?" I commented, tantalisingly, "not quite certain are we?"

There was silence while he balanced the pros and cons. He was imagining his humiliation if I was able to tell the office that I'd tricked him into believing that it was possible to cause a computer failure by reciting a poem. The pros proved weightier. "Oh, all right, if you want to make a fool of yourself, I'll be there to witness it, and maybe pass on the story to the IT department."

Oncely, Twicely, Cast the Runes ...

It was Halloween. As rare as snow at Christmas, there was a thunder storm during Halloween. It was a perfect adjunct to the surprise I planned to get the wind up Alistair. Alistair followed me into the dimly lit data centre. We gazed all around at that huge cathedral to technology, clicking sighing and flashing in its restless routine activity. This would be perfect theatre. The office where Bill the Minder and his team lived and worked was in semi-darkness, lit only by flickering screens. Somewhere in the room was Bill, waiting to animate my little Halloween pantomime.

I got the script out, and cleared my throat.

"Darkest Dark, and Deepest Deep, Bugbears rise from Death's last sleep." I intoned. The poem was vile, and deeply obscure. The mediaeval mind was full of '*memento mori*' details that are not for the squeamish. Then it got down to business. I finally came to the crescendo, reading the poem aloud in a melodramatic voice, so that it echoed around the hall;

"….
By the restless ghouls that haunt by night,
While Buegibos laughs with a grim delight
Leave your torture and your pain
Taunt those living souls again
Let lightening flash, let thunder roar
Let the clouds sweat gobs of gore
Come through the night, come creep, come crawl,
To Bugbears wail and vampire's squall,
Oncely twicely cast the runes,

www.dbooks.org

Tear the wind with ghastly tunes,
Strip the apple from the Yew
Drip the venom in the brew,
Arise Buegibos spirit of the night
Infest with Bygs in this, thy ageless rite."

I shouted out that last sentence in a loud, declamatory voice. It was a prearranged signal to Bill the Minder. To my great excitement and delight, the banks of LEDs went out. Beautiful! Alistair bent forward in surprise, as if punched in the stomach. The next moment there was an immensely loud bang high up in the building, and a shower of sparks rained down. There were strange snapping and popping sounds all around. I was amazed. Bill had surpassed himself with this. Then I became aware of an eerie silence. The huge background hissing, humming, sighing and clicking had suddenly diminished. The computers were falling silent. Alistair gave out a screech.

Bill rushed in through the entrance, looking wet and dishevelled. 'Hmm. That was a neat trick' I thought.

"What the hell is going on?" he shouted; his face white. A good actor Bill was. Alistair made another squeak and ran for the door. I gave a chuckle.

"Oh, yes, Bill, you surpassed yourself there. That was magnificent. How did you do that extraordinary explosion?"

Bill looked at me oddly

"I was stuck in a tube train for twenty minutes between stations. You were the only two people here. My pager went off just as I was walking up to the Data Centre through the rain."

------||------

"Oh come on, Phil: Another of your fine stories. We know about that incident. There was a huge direct lightening strike on the Data Centre at the Imperial Bank that almost melted the steel frame and blew just about every piece of electronic equipment in the place. It was famous, it is in all the text-books, and led to the practice of twinning data centres at two different geographical locations."

"Fine", I countered, getting off my bench in the Data Centre where I'd just told the story, "What really happened remained a secret between Alistair, Bill and I. If you're entirely sure that I've made the story up, then I just happen to have remembered the entire poem, and I'm going to recite it here and now. '*Darkest Dark and Deepest Deep, Bu……*'". The rest of the poem was lost as they wrestled me to the ground.