

THE PHRENETIC PHOUGHTS & PHIL FACTOR

Confessions of an IT Manager

by Phil Factor



simple-talk.com

First published 2006 by Simple-Talk Publishing
Cambridge, UK

Simple-Talk is provided as a service by Red Gate Software

simple-talk.com

Simple-Talk.com is an online technical magazine for professional SQL Server and .NET developers and administrators, as well as enthusiasts and those taking their first steps with Microsoft technologies.

Simple-Talk combines the content quality and editorial values of a technical magazine with the interactivity of the Internet. As paid contributors, seasoned pros, such as Phil Factor, who have seen it all and lived to tell the tale, rub shoulders with Microsoft MVPs. They are joined by a collection of working professionals united by their love of everything to do with coding and database management.

While it prides itself on the quality and comprehensiveness of its technical content, Simple-Talk also discusses the everyday life and career issues faced by the IT professional. Within the virtual pages of Simple-Talk, you can find everything from articles on building a SQL Server 2005 CLR, to tracing leaks in .NET applications, to advice on the responsibilities of a cancer-suffering software developer.

Simple-Talk is sponsored by Red Gate Software as a free industry service, with no advertising and complete editorial independence. It is a growing site, currently with 110,000 subscribers, who each receive a twice-monthly newsletter providing highlights from the Simple-Talk website along with updates and announcements concerning Red Gate tools.

If you like what you read in this book, do visit the Simple-Talk 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 and .NET people.



CREDITS

Author: Phil Factor

Editor: Tony Davis

Assistant Editor: Claire Brooking

Design: 10:AM

©Copyright Phil Factor

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 which it is published and without a similar condition including this condition being imposed on the subsequent publisher.





CONTENTS

<i>Contents</i>	5
<i>Introduction</i>	6
<i>Foreword</i>	7
<i>The DBA's Demise: a recitation</i>	8
Section I: On learning to love your Manager	9
<i>Training Your IT Manager</i>	10
<i>The Septic Tank</i>	15
<i>The Yancey Men</i>	18
<i>Talking Technical</i>	23
<i>Betting on Promotion</i>	25
<i>The Devil's Manual for IT Managers: How to Prevent Initiatives</i>	28
<i>The Incident of 'The Two Johns' - an IT Manager confesses</i>	33
Section II: Software Projects: the good, the bad and the pitiful	35
<i>Secrets of successful IT projects</i>	36
<i>Confessions of an IT Strategist</i>	42
<i>The Acronym Playpen</i>	45
<i>The Writing on the Wall</i>	47
<i>The Ghost in the Machine</i>	50
<i>The Joy of IT Meetings</i>	55
<i>The Sticking Page-Down-Key Incident</i>	58
Section III: Hirings, firings and everything in between	62
<i>Bogus resumes and unblushing lies: navigating the database hiring waters</i>	63
<i>IT Agencies and the Devil</i>	69
<i>The Stepford Geeks</i>	74
<i>The Terror of Technical Tests</i>	77
<i>Two stops short of Dagenham</i>	79
Section IV: What if...?	82
<i>If IT had been responsible for the Creation</i>	83
<i>Was offshoring responsible for Hamlet?</i>	87
<i>Had Tennyson been a Technical Author</i>	92
Section V: Hiccoughs in the working day	93
<i>I could do it in my Sleep</i>	94
<i>The Pub Lunch and Programming</i>	96
<i>The Whipping Boy</i>	98
<i>Survival Tips for PowerPoint Boredom</i>	103
<i>Santa's SLA</i>	107



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 of 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 of 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.



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.

Seriously though, 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 “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 and .NET people.

Of course, the book is best enjoyed whilst relaxing over a fine pint of beer (or two).

Enjoy !

Tony Davis

Simple-Talk Editor-in-Chief,
Red-Gate Software Cambridge

Cambridge 2006





THE DBA'S DEMISE: A RECITATION

Verses written after suffering a deadlock error, whilst listening to 'The Streets of Loredó'
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*

*"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.*

Phil Factor

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]



TRAINING YOUR IT MANAGER

FIRST PUBLISHED ON 23 NOVEMBER 2006



Editor's Note: Analogies to training errant puppies aside, the underlying truth here is that when relations between a manager and his team is poor, the power to put things right is in everyone's hands: 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 lot of 'leaping up', and 'straining at the leash', and 'playful nips'. By a system of training and reward, you modify that behaviour.

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 with 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 by 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. 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 this 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!



THE SEPTIC TANK

FIRST PUBLISHED 28 AUGUST 2006



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



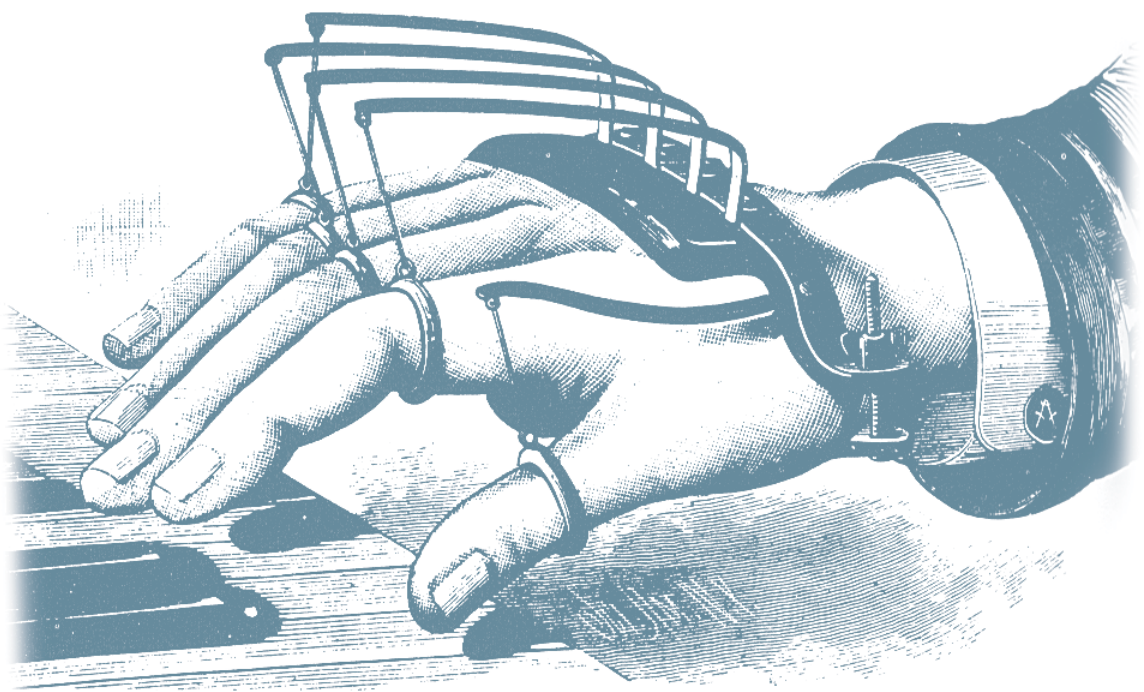
Once, whilst I was working for a large international enterprise, I received a rather disturbing "edict" from the IT director. The edict stated that *two megabytes of memory were sufficient to run all Windows-based PCs*. So firm was this edict that it became part of corporate IT strategy. PCs were not 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.





Miscellaneous electronic equipment

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.

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?"

"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 I, but which turns an ordinary person into a management X-man, with a sort of special management power only visible to another initiate?



THE YANCEY MEN

FIRST PUBLISHED 03 OCTOBER 2006



Editor's Notes: If you can't work with IT managers/directors who are technically competent, then why not just have one that could be manipulated into doing the exact bidding of the IT team? So went Phil's theory...



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.

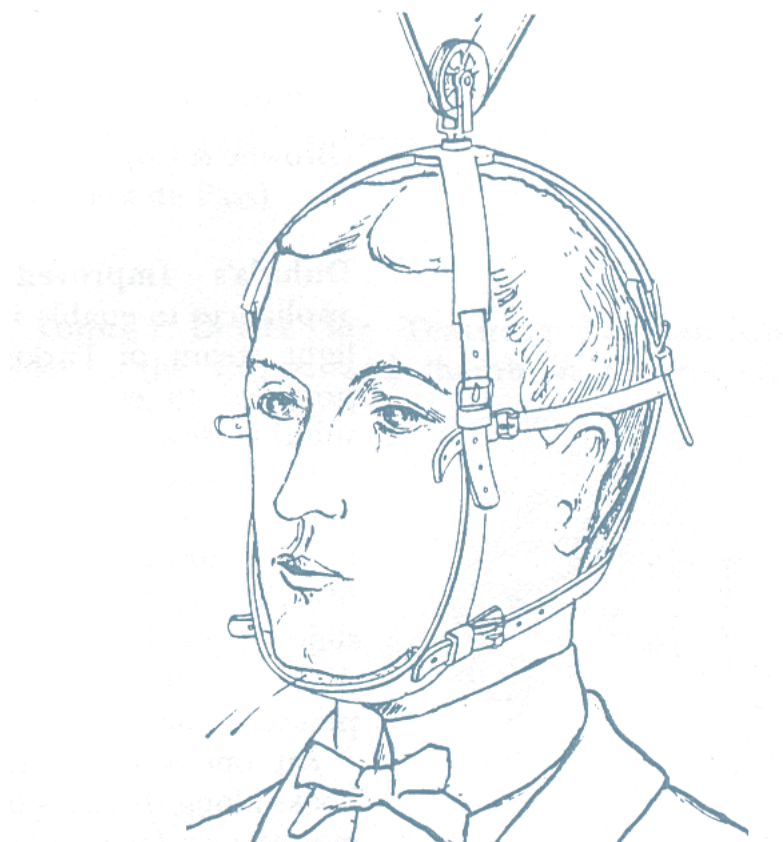
The IT director of a large company actually has little work, so long as the members of his management team know what they are doing. He does, however, need to look smart, sober, and dignified and to behave at all times with a natural 'gravitas'. His only duties are to arbitrate, convene meetings properly, 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, tall, slim, well-dressed, with hair turning slightly silvery on the temples. One evening, this very consultant, whilst being plied with drinks in





They named him Yancey

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.

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 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.



TALKING TECHNICAL

FIRST PUBLISHED 26 MAY 2006



Editor's Note: Phil exposes the darker side of having non-technical people in positions of power. To find out what's really going on you have to be prepared to fight through the 'sandstorm of waffle' and talk technically, with technical people.



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 biro's 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.

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.



BETTING ON PROMOTION

FIRST PUBLISHED 09 MAY 2006



Editor's Note: The forging 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 department 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 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.



THE DEVIL'S MANUAL FOR IT MANAGERS: HOW TO PREVENT INITIATIVES

FIRST PUBLISHED 09 MAY 2006



Editor's Note: In his career, Phil has sat on both sides of the management fence. Here he offers an essential guide to curtailing initiatives, and generally dampening high spirits, in your IT department. You will definitely recognize one or two of these...;).



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 ones' 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.

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.

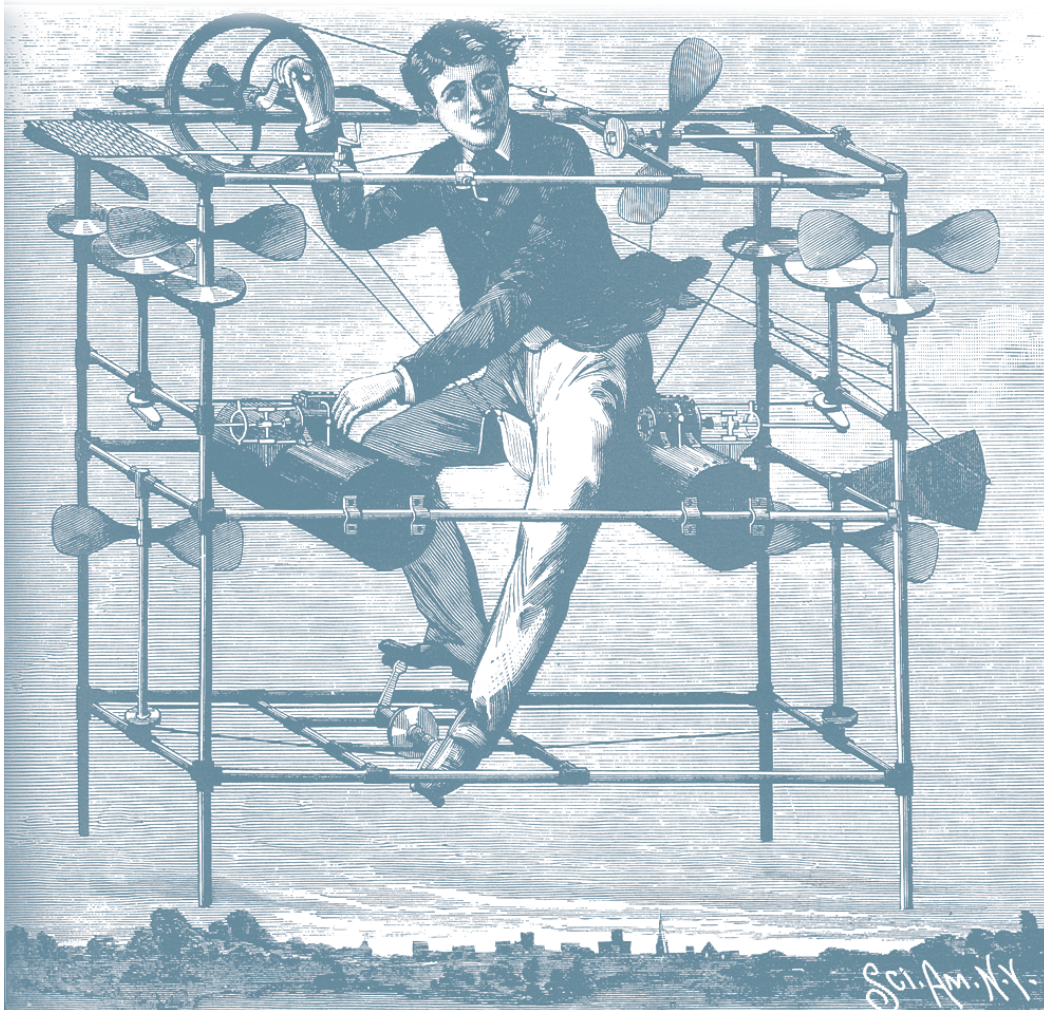
THE 'NO-PROBLEM-EXISTS' METHOD

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!

"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.

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 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.



THE INCIDENT OF 'THE TWO JOHNS': AN IT MANAGER CONFESSES

FIRST PUBLISHED MARCH 20, 2006



Editor's Note: Finally, in this section, Phil exploits his cloak of anonymity to confess to a grievous mix up in his managerial appraisal of two of his 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 in my chair with despair

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.

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:

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 were 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 no position to preach ;). [TD]



SECRETS OF SUCCESSFUL IT PROJECTS

FIRST PUBLISHED NOVEMBER 25, 2005



Editor's Note: How do know you're in a successful development team? More of them than average will be wearing tweed.



"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.

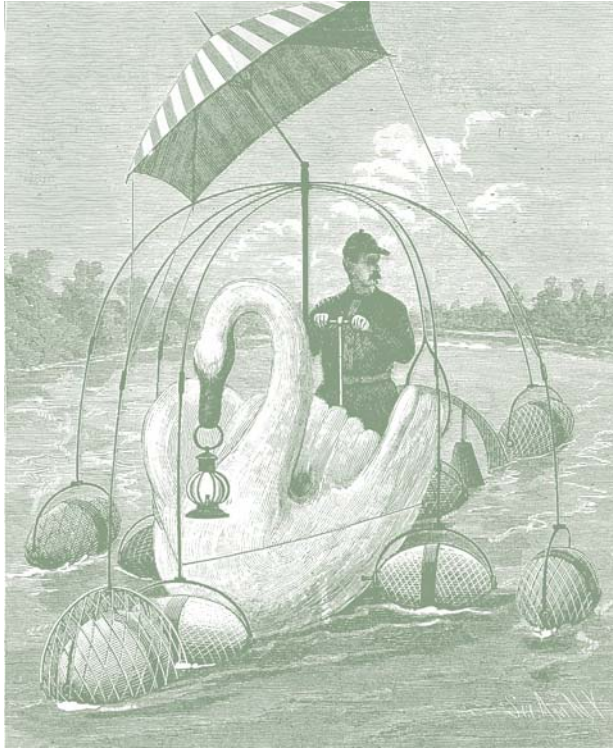
Some time ago, I worked on a team that was attempting to define an IT strategy for a large multinational company. Management assumed, incorrectly, that the company was unique in that it had a high number of failed IT projects. In fact, only one in four of all IT projects can be termed successful, if 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!





Don't be a pioneer

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*



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 afterward 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. 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 get 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 it 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, 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.

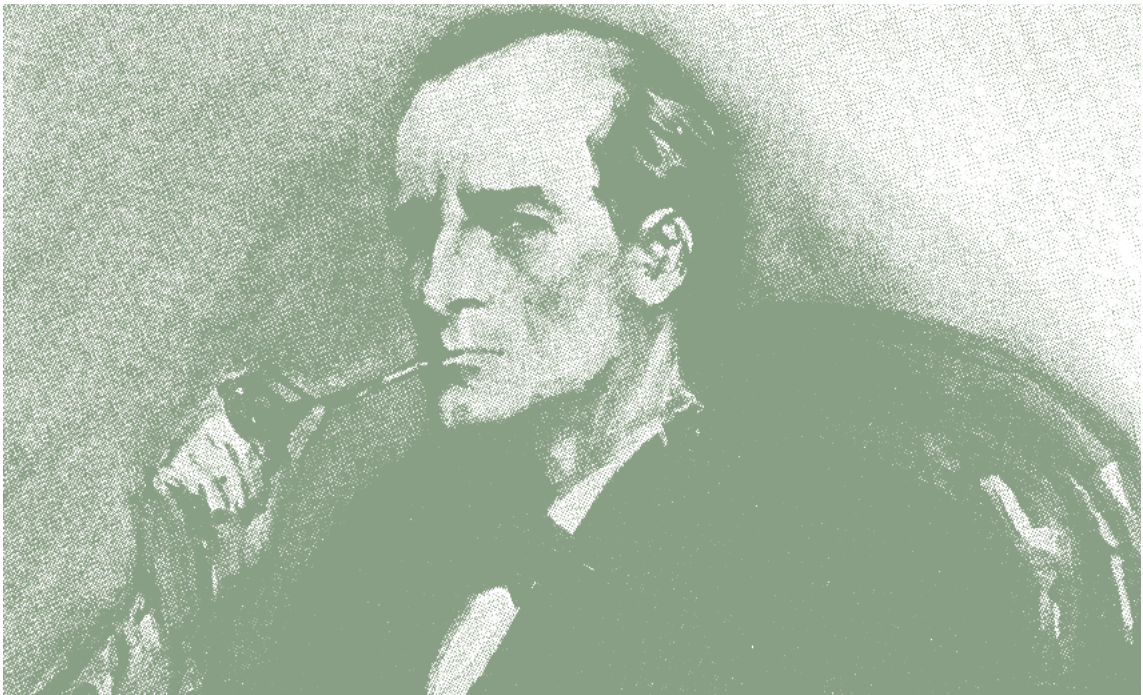


CONFESSIONS OF AN IT STRATEGIST

FIRST PUBLISHED NOVEMBER 16, 2006



Editor's Note: 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 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. 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.



THE ACRONYM PLAYPEN

FIRST PUBLISHED SEPTEMBER 13, 2006



Editor's Note: Most programmers find hard to resist the lure of the latest, hottest Three-Letter-Acronym 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 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.



THE WRITING ON THE WALL

FIRST PUBLISHED APRIL 27, 2006



Editor's Note: This tale highlights a truth hinted at in the previous story: that people are often not that keen at all to help you uncover the true processes that underpin their working lives. They fear, sometimes with good reason, that "rationalization" projects are just another name for "redundancies".



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.



What is this strange writing on the wall?!



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

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?



THE GHOST IN THE MACHINE

FIRST PUBLISHED JUNE 27, 2006



Editor's Note: When a developer inherits third party code, how often do you do hear "Wow, that's great. Not much I can do to improve that!"? I'm guessing not very often. More common is the whistling through teeth, groaning in disbelief, and raising of eyebrows to heaven that precede a complete dismantling and rewriting. Before you grab the mallet, however, Phil suggests you try a little ART...



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 involved more work than is strictly necessary. On occasions when an application is so complex that rewriting it from scratch is infeasible, 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 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.

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 consequentially lost from sight of 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 VAXs 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.

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.



THE JOY OF IT MEETINGS

FIRST PUBLISHED AUGUST 03, 2006



Editor's Note: 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 in 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.



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 lawnmower, 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.



THE STICKING PAGE-DOWN-KEY INCIDENT

FIRST PUBLISHED SEPTEMBER 01, 2006



Editor's Note: Sometimes the reasons for a project's failure are many and diverse. Occasionally, though, the single cause is just sitting there right in front of you, waiting to be discovered.



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 doing 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 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.



They all left at the same time

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 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.

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.



SECTION III: HIRINGS, FIRINGS AND EVERYTHING IN BETWEEN

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]

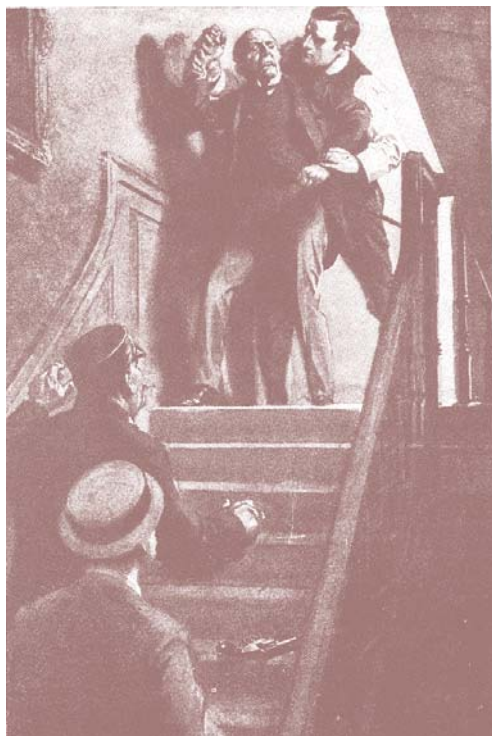


BOGUS RESUMES AND UNBLUSHING LIES: NAVIGATING THE DATABASE HIRING WATERS

FIRST PUBLISHED OCTOBER 18, 2005



Editor's Note: Experienced programmer required. Must have extensive knowledge of programming in T-SQL, Visual Basic and Croydon...



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.

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 as he messed up catastrophically. I had, of course, told the IT director, who assumed I was motivated by professional jealousy.



Have you ever programmed in Croydon?



PROGRAMMING 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 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.



IT AGENCIES AND THE DEVIL

FIRST PUBLISHED MARCH 15, 2006



Editor's Note: Bored of your current job? Need a new challenge? Read Phil's easy, two-step guide to starting your own IT employment agency. Highly competitive salaries on offer.



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 at getting



them one of the splendid jobs you 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 Stroat 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.

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.

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.



*Harvey Furtz is sensitive about
his name*

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.



THE STEPFOORD GEEKS

FIRST PUBLISHED JUNE 10, 2006



Editor's Note: Having your brains sucked out at interview and more nefarious deeds by IT agencies...



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 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 Reengineering 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.



THE TERROR OF TECHNICAL TESTS

FIRST PUBLISHED NOVEMBER 03, 2006



Editor's Note: Technical tests are a necessary evil. Imagine there was a chance that you could be operated on by a surgeon who had managed to "bullshit his way into a job armed with a bogus CV and a string of false references". Now imagine you're a database and the surgeon is a DBA. Scary.



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.



TWO STOPS SHORT OF DAGENHAM

FIRST PUBLISHED FEBRUARY 09, 2006



Editor's Note: Don't be put off by the obtuse title. Once you're past it what you get is 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.



But...if you're still worried about that title, here is my short explanation:

1. *On the District line of the London Underground Tube System, Dagenham was originally two stops away from Barking.*
2. *"Barking" is also a shortened version of the English colloquialism "barking mad"*
3. *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.



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 *Girly* 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.



SECTION IV:

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]

IV



IF IT HAD BEEN RESPONSIBLE FOR THE CREATION

FIRST PUBLISHED AUGUST 09, 2006



Editor's Note: 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 angel executive 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 manger 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 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..."



WAS OFFSHORING RESPONSIBLE FOR HAMLET?

FIRST PUBLISHED SEPTEMBER 05, 2006



Editor's Note: 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 Shakespeare. 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, Johnson, 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.

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, whist 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]: These small miscommunications 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 inult me, Thou inult 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 assure you that the team which worked upon this project were all experienced versifiers and qualified playwrights, sensitive to thine business requirements, able to bombast out a blank verse as the befit 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 t's 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 stiff! 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...

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 Prince 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)



HAD TENNYSON BEEN A TECHNICAL AUTHOR

FIRST PUBLISHED JULY 16, 2006



Editor's Note: 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*



SECTION V: 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].

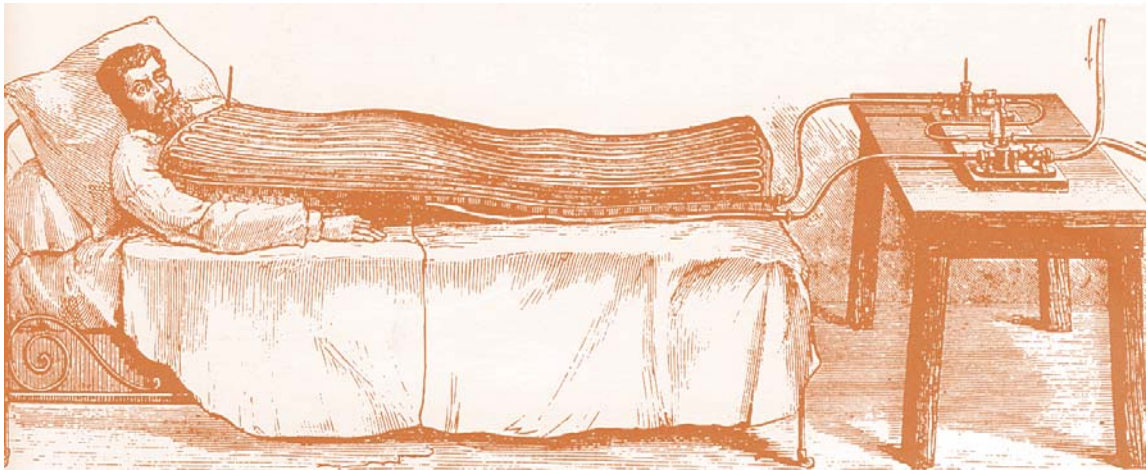


I COULD DO IT IN MY SLEEP

FIRST PUBLISHED FEBRUARY 23, 2006



Editor's Note: 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 cyclic 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 ones 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 Huguenot 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.

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.



THE PUB LUNCH AND PROGRAMMING

FIRST PUBLISHED MAY 04, 2006



Editor's Note: A fascinating, scientific investigation, plotting the variation in coding errors with number of pints imbibed.



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.

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 ones' 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:



<i>Consumption</i>	<i>Errors per Stored Procedure</i>
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?



THE WHIPPING BOY

FIRST PUBLISHED JULY 21, 2006



Editor's Note: Whenever IT life throws Phil a curveball, and harsh words are heading in his 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 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 this 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 MD's 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.

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 scramble 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 practiced 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: 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.



SURVIVAL TIPS FOR POWERPOINT BOREDOM

FIRST PUBLISHED JANUARY 17, 2006



Editor's Note: 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.



You have to fight it!

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 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 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 10 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.

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.



SANTA'S SLA

FIRST PUBLISHED DECEMBER 05, 2006



Editor's Note: Finishing the book on a seasonal note, we have a cheering festive tale of the SLA that didn't deliver all it seemed to promise.



SANTA CLAUS

- a la mode !

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 an 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.

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.

THE END