Introduction

Where to start?

I remeber drawing a picture when I was a kid of about 9. The picture was supposed to be what our ideal bedroom would be - I think there was a hatch that took me down to the garage, a bed that flipped up into the wall and many many buttons. Mostly buttons were easy to draw and they meant something - gadgety.

The thing is gadgets seem easy, the perfect solution. but they are a child like reposonse

I am told that today is the modern age, the gleaming 21st Century, full of promise and hope. But I look around and I know we have not made it. I am glad and saddened in equal parts. Glad because the promise to a child was a false one, but sad because as an adult there is a better world, but it is not one where a few gadgets will get us there - it is a world where we must work harder and smarter to reach. We need people to help us convert Industrial Age organisations into 21st century versions. IT managers are to be the new heros. Like lawyers. (no honestly I am not mad - I will get there dont worry)

As a child I remember being promised a technological utopia, a paperless, robot driven school and a burnished steel workplace brimming with gadgets a-plenty, all of them helpfully making our lives simpler and easier. I even drew a picture of the future. But it has not turned out like that. And frankly, I am glad.

This vision of the future, of living my parents (1970's) life but with gadgets and machines doing the drudgery, is based on a misconception - that gadgets can replace tools and institutions, - or rather that gadgets can replace the need for a human to understand and use their tools, skillfully and well, individually and en-mass. Washing machines should have freed women from the days of toil over baths and mangles, which it did except they needed to work to pay for the machine, as well as wash more clothes more regularly.

> We have in the programmable computer created perhaps the most versatile tool in human history. But like all tools its use requires skill and understanding, and often a willingness to change what and how we work to ensure the tool can benefit us.

It is a conceit born of an unwillingness to change, but we often think that tools should alter to meet our needs, when we find that instead we need to change to meet the tools demands. Some years ago the company I worked for decided to abandon a in-house built billing system to buy the market-leading (but only two years old) system. Unfortunately we billed our subscribers in advance, a concept this system had no understanding of, but something that was only realised halfway in. Instead of, say, living with a overdraft of 2 million for a year or two, the company set about with a will to modify the billing system(s), and the usual cost overruns, delays and mishaps and lost opportunites certainly came with a higher bill than the interest on the loan.

I liken this to a factory that decides to invest in a steel stamp - basically a big hammer that falls down on steel sheets and bangs them into moulds. The factory is unfortunately on one storey only, so there are two choices - to build another storey, perhaps moving premises, or to workout a way to lay the hammer on its side. I am sure we can tell which is the “correct” choice - but in my career I have seen many companies invest a lot of clever engineering talent into working out how to make a hammer fall horizonatally (“...maybe if we used rockets?”). That engineering talent could have been building new moulds for emerging markets.

To make the right decision requires someone who understands the technology of hammers (gravity, big weight), and the nature of organisations, and that they are at the right level of influence in the company.

I am not too sure what life was like for proto-humans before fire-making, but I know that society clearly changed to accomodate the benefits of fire. The wheel enabled trade, roads created the Middle Ages, and the computer will force our Industrial Age society to change again.

I do not believe that even if it could be done, letting gadgets take care of all our needs would make our lives better - for the simple reason that a gadget embodies its designers decisions about its use, and your life. Good or even great design will always win out, but no matter how good the design, it will always be someone elses choices. Unless we can alter those choices to better suit usm we are living a life to anothers choice - a life without freedom.

Yet to shape a new society there are two types of skill needed - those who can understand and use the new tool, and those who can understand and use the new tool *in order to* change the organisations to use it most effectively. And where our organisations and institutions go, our society will follow.

This new skill, understanding how large-scale information processing will change our world, and positioning ourselves to take advantage, is currently the preserve of a curious beast known as an IT manager. But there is not a lot of discussion about this.

But there still is promise and hope. Honest.

Just like lawyers

According to Peter Drucker we are all becoming “knowledge workers” - using our brains not our brawn to manipulate and improve our world. Well the original knowledge workers may or may not agree. Cast your minds back, back into the reaches of time, say to the court of the Pharoahs.

The all powerful pharoah has had enough. Rameses the Unlikely Cockney III makes his voice ring out around the pyramids.

> “”“I send out my decrees, my word is like that of the gods, but by the time it gets to the other side of my empire the bloody word is mangled and misinterpreted, its like Chinese whispers around here. Oi you - scribe. I’ve got an idea. Write down my next decree then my soldiers can take that piece of parchment over to the other side of the empire and your mate can read it out - that way no bugger can conveniently ”misunderstand me“ - OK? ”“”

And lo, the Pharoahs word was written down, and his decrees were like the word of the gods.

> “”“ we need a good name for this system scribe. what shall we call it? Lawrence. Cant call it Lawrence - silly name. SHortend it. Call it, rence. OK? No. Thats no good. Lawren? Law?”“”

And so the scribe writes down his masters words, but over time Rameses the Cockney makes so many pronouncements, that even he forgets what he was on about, and the scribe must refer back, to different and probably conflicting pronouncements. And the scribe must then interpret the law, because obviously Rameses did not mean to contradict himself.

And the written word starts to take on more importance than the actual Rameses. I mean he cant be everywhere pronouncing on every little thing can he? So people read what he said, and work on that basis.

And some societies thrive, and some, their laws unsuitable or unchanging do not thrive.

And then someone invents a new way to write things down, a new means called programming. And society suddenly has a new way to define how it wants to run its own affairs.

And those who wrote things down, become those who can interpret what was written down. The scribes become the lawyers, arguing over precise defintions of piddling little words, whilst trying to recognise that sometimes their writings can affect millions.

Its a bit like programmers arguing over a little bug in a million lines of source code, whilst trying to remember the code is for a missile, a power plant a banking risk manager.

In fact its a lot like it ...

Bookshelves

Nowhere is this more obvious than in a large bookstore. Borders, Waterstones and so on have grown like Topsy simply to give the lie to the idea of a paperless world, the remains of millions of dead trees carefully arranged to stick two fingers up at an electronic dawn, but, but, oh irony of ironies, the heaviest tomes and the most groaning shelves are reserved for books about IT and computers. Whole shelfs creak under the weight of 300 pages promising to teach you a programming language in 24 hours even if it will take 3 days just to read the book.

However look closer, there are plenty of books about how to use Microsoft Office to improve your cashflow, how to set up Oracle in a few hours and promise in just 10 days they can make you a man, (no, sorry, thats Rocky Horror), but there is nothing on how to run the department that looks after all this stuff. Until I became an IT manager it never really struck me that there was a gap in the market.

Admittedly it is not a large gap - the number of people who run IT departments is quite small compared to the number who want to learn Excel a bit better. But in many ways that is the essence of what I am trying to say - learning Excel a bit will not make a company vastly more profitable, nor will *not* learning Excel cause it to run crunchingly into administration. However companies and countries these days rely more and more on code and data, and it is the management of these two things, and the complexities inherent in that management[#]_ that is the stuff of an IT department. Mucking up your data or code *can* drive a company into banckruptcy, and getting it right, sweetly right, can vastly improve your bottom line. As for making you a man, well ... deal with that on your own.

To be fair it was sometime *after* I started running IT teams and departments that I noticed the bookstore problem, and that is becuase the idea of running an IT department is curiously absent from the minds of most people. It is taken for granted that running a big company somehow requires some magic potion for which shareholders should be grateful they only get stiffed a measly few million a year. Running a football team is a front page job and no shortage of people looking at how you do it. But running an IT team well thats just making sure the backups are done and maybe a bit of project management.

As you may guess I think there is more to it.

What is an IT Manager / CIO / CTO / ...

In many ways this heading is the problem - what the hell do you call an IT manager. Looking at job adverts it is easy to see a creeping job-title inflation.

For the UK now, the term *IT Manager* seems to be reserved for the person in charge of the helpdesk fixing the printers and so forth. Whilst for some companies this is a role with supervision over dozens of people, it still is just supervision. *Head of IT* seems to be becoming an accepted title for what I mean - as is *CTO* or *CIO*. These job descriptions indicate that the important point is you are in charge of all the IT stuff for the company.

And to me that is the clincher, whatever the job titles, whatever the description, the person who makes the final, *informed* decision about the architecture of information data and code in an organisation is that company's IT Manager. It might be the helpdesk supervisor, it might be the accountant, but ultimately someone will decide how all the systems in a company fit together, work together and share information and share functions. Sometimes, quite often I suspect, the person who is the IT manager for a company is not even in the company. f we are looking for the final, *informed* decision maker - it is perfectly possible for that person to be a consultant hired by the nominally titled IT Manager, but that IT manager is not skilled enough to make IT decisions on their own. Frankly you need to be able to program before you can really be in the industry[#]_. And many many people who cannot program are in IT. Its like having someone in catering who cannot cook. The only job there is maitre'd.

inlcuding bespoke programming

No Decision here

Not actually having a final informed decision maker in the company seems to be getting more common, not less common. It is also one of the reasons I recommend OSS over microsoft. It is not some irrational hatred of the big bad Redmound but is a desire to see people take positive decisions over systems integration. Many years ago the default non-decision would leave a company with two systems that could not communicate, and so lead to the *systems integrators* compnaies and consultants whose sole purpose was to come in and make the new system work with the legacy computers, usually in a way that blames the old legacy computer, not the person who bought an incompatible system in the first place. However now it is easier and easier to buy Microsoft (or SAP or Oracle) and things do not break, they more or less work, more or less well. However all that has happened is that Microsoft has chosen what they think is the best way for your systems to work. Is it?

Are you sure you do not want to think about that?

If you run your IT the same way as millions of others, where does your competitive advantage come from? Some companies can answer that happy in the knowledge their competitive advantage lies outside of a CPU... but in the 21st Century, more and more competitive advantage will come from the zone between the real and the virtual.

> What you need is someone who can sit down and think about how to get all your systems to work together in the most beneficial way for you and your organisation. Like an IT Manager.

Cook well on your own

Well, firstly there needs to be ability. Technical skills are the bottom line, no matter what else is waved around, technical ability is vital. It is not that we expect the CIO to write all the code or configure all the databases. But we do expect them to set the standards, spot the glaring design faults and ensure that what is produced is the best quality we are willing to pay for.

An example that recently made my blood boil was a job ad for 'Head of IT', which clearly stated:

```
A non-technical background is not considered a bar-
rier, as there are 3rd party providers who
can supply technical assistance.
```

Dear God, thats not "Head of IT". Thats a supervisior. At 80K a year. I mean would we accept a job that ran

```
Wanted -
Head Chef. No need to able to cook, as the providers of frozen pro-
cessed foods
will be able to supply all assitance necessary.
```

Well actually we do get quite a few adverts like that - they explain the parlous state of our childrens school meals, which have moved from something actually prepared and cooked on site to, well, Turkey Twizzlers.

And lets face it, the company above (a law firm of some sort) will get the same kind of nutritious and profitable IT systems as the patrons of the restaurant above will get as meals. All the value of a frozen breadcrumbed reconsitituied chicken. And yet, yet look at the salary. It is based at 80K, a clear indication that the partners expect something special, something nutirtious and tasty from their 'Head of IT'. They are willing to accept a dinner lady, but pay for Gordon Ramsey. I wonder which they are mostl likey to get applying?

so the first line of the job ad should read:

```
wanted: Head Chef. As a *sine qua non* You must be able to cook well on your own.
```

This really emphaisises the most obvious point about IT management and one that I feel is ignored and even quite controversial when I state it - A good IT manager should be able to do every job under him. FOr me the simple answer is that the IT manager should be able to run a small office (say 3-5 people) entirely on his own. THis is the equivalent of asking, if I hire Gordon Ramsey as a cook for my family (yeah right) will he be able to shop and cook from scratch all alone? FOr GOrdon we can happily say yes. At that point we can worry if he can scale it up to a restaurant or a chain. For an IT manager, we need to ask, can he set up and run all the email, web, printers and so forth for a 1 or 2 person office? If not, then on what basis do we think they can scale up?

The section of the book entitled *SoHo Solo* is about how to actually do this - to run a small office / home office (SoHo) as if you were on your own - from the initial setup of servers, to DNS, email virus scanning and database management. It is quite a tall order, but then let us face it, buying quality fresh foods and knowing how to boil a lobster, make fresh pasta and raise a souffle is quite a tall order too.

The Head chef sets the tone of a kitchen, sets the standards, and is the gateway through which all finished products must pass. He can only do this if he knows what good fresh pasta looks like and what is needed to prepare it.

In short you need a boss who can do the job he is in charge of. You wont hire a newspaper editor who cannot write, you wont hire a chef who cannot cook. So why hire an IT manager who cannot program?

Necessary but not sufficient

I freely and happily admit that *technical competance is a necessary but not sufficient condition* for being a successful IT manager.

overview

- most flexible tool
- programming is vital, but not sufficient. we are just scribes. Asked in to "Do some writing down"
- sysadmin/soho is "scraping your own vellum"
- project managmeent, release management, source control
- taking care of the scribes, mentoring, story telling, escaping the industrial mind set
- Town planning, not architecture, the issues over architecture and building, code is the design broken windows
- managing upwards and outwqards - specifications and reliability, escaping the support trap
- real and virtual - programmatic access is the linking area. RFID, epidimeiolgy, search etc.

An IT manager must play many roles, and I try and list them below, and what follows will be discussions on the roles and ways to play them out successfully.

first must be able to cook well on his own then project mgmt - handling more than one order town planning

- Technician and geek.
- Architect and Town Planner
- Project and Programme Manager
- Editor, Reviewer and Publisher (release mgmt)
- Accountant and Auditor

- Geek Philosopher and Storyteller (affecting culture (how we do things) and managing expectations)
- Modeller, monitor and mentor
- feedback, shout triumphs, sailing round storms
- refactoring code over time is like scraping off barnacles, broken windows

Guiding principles and philosophy

There are several guiding principles

- If you do not design, build or sell the product, I will do all in my power to automate you.
- build from scratch, from scripts.
- release well
- everything can have programmatic access. Everything.
- Wrong and not wrong
- everyone will be able to program, but not everyone will earn a living as a programmer. Just as everyone can write but not everyone will earn a living writing. think lawyers not journalists.

programmatic access

Wrong, and not wrong

Most people think the world is a place of grey, uncertain moral and value judgements where somethings depend greatly on your point of view. Of course very few people think everything depends on your point of view (its very hard to have a grey area of paedophilia for example) and equally few people feel nothing is realtive, that God or who ever has already laid down all you need to know.

Given this is it strange that most businesses seem to think that a project is a success or a failure, a contributr to the bottom line or not.

I prefer, instrad of the right or wrong judegemtn to have 4 categories instead of two

- Right
- Not Wrong
- Wrong
- Not even wrong

Right is the best possible choice and outcome. It is the superlative option, the Van Gogh cornfields, the Da Vinci sketch. It is also nearly impossible to judge at the moment - the passage of time, comparison to many other alternatives is usually needed to be sure it was Right.

Not Even Wrong is a phrase coined by the German Phyisicist (xxx) - he was commenting on the work of a student with a certain lack of tact, and made it clear that the student had not simply got his answer wrong, but that he had been working in the wrong directions. Its a little like my Maths O levels, where I was expected to ’show my workings’ so that even if my long division failed me I would still get points for trying to divide the right fractions.

THis concept gives us a look at the ideas of Not Wrong. If right is the artisitically correct, perfect version then there are others that are also close to perfect - constables sketches, Turner as a scultptor. THese are things that will not be world class, but are the poutput of talented professionsal with true skill. Not wrong is where we have tried to divide the correct fractions, and got our division correct. Its just that we could have used calculus.

Wrong is where our long division failed us. We can at the time see that worng is wrong. We, or perhpas a professional IT manager would say ' THat is not how I want it because that will lead to problems'

anlaogies

driving writing cooking architecture/building/construction

Ranting about non-technical requirements

Driving in 20C Writing in 15C

Both changed the world, but ONLY when everyone[1] started to do it

[1] for a given value of everyone

technology makes possible everything you do, it surrounds you, feeds and clothes you shelters you and you could not survive without it

not understanding technology makes ou an ELoi, ignoring the real world whilst the morlocks toil away beneath you

we must understandintechnology, how we go from energy to use, how we can use it to survive, grow and do less damage.

less damage - the pest contorl version - cynaide pellets, DDT, GMO - each less violent.

Thinking *is* innovative. And lets face it, in the stone age would you like to have a book on emaotuions or a book on science

if we think about it, the writing analogy is truly forceful - would we consider it sensible that todays business hires a team of writers everytime they want to write something down.?

THis in many ways used to be the stenopool, the bookmaker/accountant/dataentry, or the scribe. What happened to them, what technological or social shifts moved them out? THe hancock replaced by Lyons, the stenopool, replaced by cost of typist vs cost of PA. The stenopool became PA, organisers, project mgrs.

The scribe. They became lawyers.

What will become of a society that has rules codified in a way only the computer can understand. We will need a profession that can be trusted to write things down, the rules of society.

Understanding technology

Its all about programming

Computers are there for programming, for computing the codified, straitjacketed expression of human intelligence known as source code. These computers are for programming as paper is for writing. We do not hand our teenagers pen and paper and congratulate them as 'paper-literate' if they are able to fold an aeroplane, or doodle a recognisable tree. the very nature of writing is its value, its ability to express and communicate across time and space. A program can do the same and more. Its expressiveness can be greater, as it allows a recursion and growth that makes infinity seem podgy and small.

Books can take us out of ourselves and into the world, whereas a program can take us inside itself.

Where this new world will leave us all I do not know. Its up to our children to find out. But we must ensure they are truly programming-literate.

I used to use driving as a metaphor for programming - that we would look at someone who could not program in same way as someone who could not drive. Askance, unusual. But I believe the change is deeper. being able to use a car, let us say drive safely, is similar to being able to use Microsoft Office, the current standard for computer-literate the European COmputer Drivign licence.

However it seems to go beyond that, someone is not considered literate if able to fold paper.

Footnotes

malformed hyperlink target.

malformed hyperlink target.

Docutils System Messages

Too many autonumbered footnote references: only 0 corresponding footnotes available.

[A] http://www.quotedb.com/quotes/1563

[B] Socrates was forced to commit suicide by drinking Hemlock. http://en.wikipedia.org/wiki/Socrates#Trial_and_Death