VFQ Value Flow Quality

Integration

by emergn

This publication forms part of Emergn's Learning Services. Details of this and Emergn courses can be obtained from Emergn, 1st Floor, Liffey Trust Centre, 117–126 Sheriff Street Upper, Dublin, D01 YC43, Ireland or Emergn, 190 High St, Floor 4, Boston, MA 02110, USA.

Alternatively, you may visit the Emergn website at http://www.emergn.com/education where you can learn more about the range of courses on offer.

To purchase Emergn's courseware, contact us for a brochure - tel. +44 (0)808 189 2043; email valueflowquality@emergn.com

Emergn Ltd.
1st Floor, Liffey Trust Centre
117–126 Sheriff Street Upper
Dublin, D01 YC43
Ireland

Emergn Inc.
190 High St, Floor 4
Boston, MA 02110
USA

First published 2012, revised 2023 - printed 27 March 2023 (version 2.1)

Printed and bound in the United Kingdom by Apple Capital Print.

CONTENTS

INTRODUCTION

Mention integration in IT and you tend to get groans. Mention integration around other departments and you get blank stares.

And yet what could be simpler?

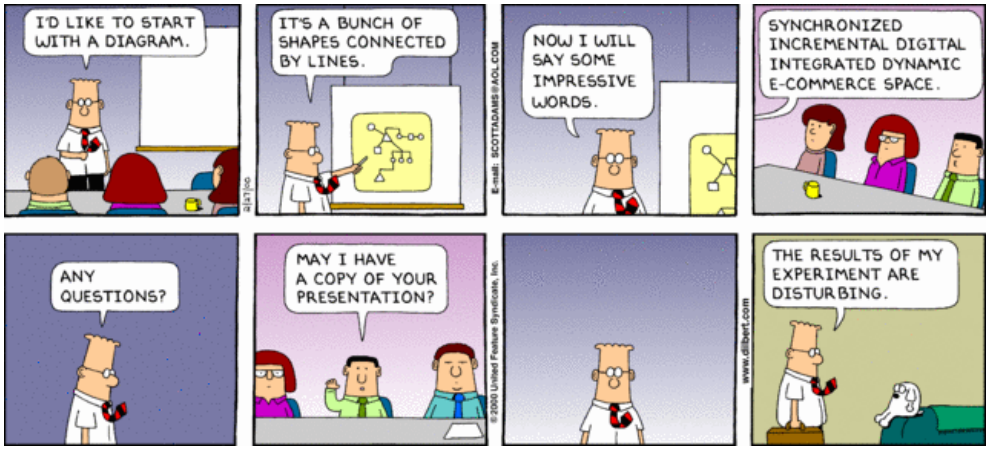


Figure 1. Dilbert cartoon

According to Wikipedia, integration is 'the bringing together of component subsystems into one system and ensuring that the subsystems function together as a system'. This definition doesn't apply only to IT – it works for any system. If you have a system like a 'professional kitchen', you want to make sure that the subsystems of 'pastry', 'fish', 'meat' and 'side dishes' all function together to produce the perfect meal.

Have you ever seen the Eddie Izzard sketch on technology?

When trying to print, he gets the error message 'Cannot access printer'.

'Cannot access printer?' demands Eddie incredulously. 'It's here.' And Eddie lifts the computer and helpfully tilts it over so it can see the printer.

Which, alas, does not help with an integration problem between the 'word-processor' and the 'printer'.

In Eddie Izzard's case, the solution (after some intensive swearing at the computer and a phone call to a helpdesk in Amsterdam) is to switch on the printer.

Figure 2. Frustration of printer not working (a scene from Modern Family on the ABC Television Network)

IT departments often take a similar approach (we don't mean swearing and calling Amsterdam for help). They see the problem as a purely technical issue: a challenge to bring differing pieces of software (and hardware) together, when in fact the problem is more fundamental. Integration difficulties often stem from not looking at the business fundamentals that underlie an apparent technical difficulty.

By the end of this session, the student will have an appreciation of:

1. Why integration is a business problem and not just a technical issue.
2. Why IT systems are naturally 'messy'.
3. The resulting difficulty of integration.
4. How business requirements drive integration.
5. The techniques that assist better integration.

1 INTEGRATION MUST BE DRIVEN BY BUSINESS

When you visit your doctor, he fills in details of your consultation on a computer. The surgery manager has probably spent a great deal of time selecting what he hopes is the perfect package. As well as being secure, affordable and robust, the manager has also had to check that the surgery database is compatible with the hospital database, can process health insurance claims and can be updated in the future as new requirements for patient data occur.

Why bother? After all, GP surgeries have been successfully holding and retrieving patient data for decades. The system called for a cardboard folder, some sheets of paper and a filing cabinet. Patients who came in a lot might merit a paperclip. The receptionist could usually pull out a complete patient file within seconds. As a system it was remarkably successful and there was no need for expensive computers, a server, a backup server, software, etc. Why would you change something that worked?

A computerised system has obvious benefits for GP surgeries and patients. Standardised notes mean the same information is readily available to all users (from a specialist consultant, to a locum doctor, to the dear old family doctor who helped deliver you); large numbers of processes can be automated, from sending out letters about your flu jab being due to ensuring your vaccination is in date. It means greater safety and better service for patients, it means an easier job for doctors, and (eventually) lower overhead costs for the surgery.

Successful businesses digitise their core processes because this makes them more efficient and quicker to respond to challenges or opportunities. Note that some elements in the core processes cannot be digitised effectively – while patients records can be updated and stored digitally, the doctor's diagnosis remains unique and human. The real benefits come from being able to discern which elements can be appropriately digitised and which are performed more effectively by people. When IT standardises and integrates processes it creates a secure operational base that leaves staff free to do other things – expand, change, innovate. This automated base is less likely to make mistakes over mundane, repetitive tasks than humans. Its rationale is to free the business, not to implement technology for technology's sake. And therefore, because integration is a business issue, not a technology one, it becomes the responsibility of everyone in the company, not just the IT department.

Figure 3. Filing of patients records

Doctors did not start using computer systems because they fancied a desktop computer rather than a cardboard file, but because the system offered real benefits.

Technology does not drive an integration project. Integration is the result of a business need, and it provides a business benefit. If it doesn't, you should stick to the paperclips and the filing cabinet.

Activity 1: The automated organisation

This activity should take about 10 minutes.

In the modern organisation, there is a seemingly relentless drive towards efficiency, with IT and particularly process automation being the tools of choice.

In some cases, the efforts are driven by the promise of complete automation, derived from the latest platform offering from a software vendor. However, in those cases, it's rare that the reality meets the expectations.

Consider your own organisation. Write a list of a dozen business processes that currently exist (regardless whether they are automated or not). Try to identify the more specific processes; for example rather than 'order fulfilment' list the constituent parts like Invoicing, Billing, Shipment, Delivery, etc. Use the terminology specific to your organisation.

Now for each of the processes listed, indicate the level of automation: fully automated, partially automated, manual. Examine the last category more closely; which of them are planned for automation? Which are most difficult to automate? Can you think of any processes that you feel can't be entirely automated? List a few reasons why.

Commentary:

As we alluded to, it is often thought that all processes can be automated and that doing so will be beneficial because of the efficiency gains. However, this can overlook the value delivered by a process which tends to depend on having the flexibility to make non-standard decisions. This is only possible when people have the freedom to make these decisions.

2 IT SYSTEMS ARE A MESS

During the Second World War the Blitz destroyed large sections of London. Instead of recreating the crowded streets of the East End and attempting to integrate new buildings into the existing structure, town planners decided to build a brighter, better future in entirely new towns. These towns would not need to fit in between medieval walls or deal with centuries of higgledy-piggledy extensions. These towns would have logically laid out streets, districts and houses. Milton Keynes was designated as the site for a new city for 250,000 people. Planners were certain that they had discovered the winning formula:

- opportunity and freedom of choice
- easy movement and access
- good communications
- balance and variety
- an attractive city
- public awareness and participation
- efficient and imaginative use of resources.

Figure 4. How Milton Keynes was supposed to look

Alas, Milton Keynes did not turn out to be quite the paradise of the planners' dreams, which perhaps points to a warning for fans of the perfect product specification... but even devotees of upfront planning will acknowledge that most IT systems are not Milton Keynes; they are London. Built up bit by bit, spreading, acquiring, changing use, gaining extensions, multiple tiers, new foundations... it is not unusual to find a company has several different websites, databases and departmental specific systems to handle ordering, warehousing, accounting and customers.

2.1. Mess is natural

Businesses, like cities, are big and complicated and they change over time, so it shouldn't come as any surprise that they tend to end up with a messy IT architecture. As a new business need arises, a new software solution will be bolted on to the old. As a technology evolves, there arises a competitive advantage for the company that is able to adopt it quickly, and eventually a penalty for any company that has NOT adopted it. Think back – if you can – to the days when companies went around asking 'so do I really need one of these website thingies then?' Now think of any company you know that doesn't trade over the internet, at least in part.

As time goes on, old systems become harder and harder to integrate with the new. Complexities grow when an add-on causes problems, so a fix is created, and then another one, and another. The result is what author Brian Foote memorably described as 'a big ball of mud'. Before very long, your IT system is filled with quirks and special processes that only the IT guy who's been there for 20 years understands. This is the opposite to the business process digitisation we described as the core of staying efficient and agile.

2.2. Natural doesn't mean inevitable

Just because cities have a natural tendency to grow piecemeal and end up messy, it doesn't mean that there aren't real benefits in putting in sewers and electrical cables, and even occasionally pulling down houses to widen streets or rebuilding flammable wooden buildings in brick. The same is true of IT systems.

3 WHY IS IT SO HARD?

More necessary to our comfort than the internal combustion engine or even electricity, Sir Joseph Bazalgette's engineering work deserves more recognition than it gets. This great man did not give us cars, trains, lights or central heating – he gave us sewers... and without them London would be quite literally drowning in effluent.

In the 19th century the population of London soared. All parts of the capital's infrastructure suffered, but none more so than the old cesspits. Cellars filled up with human waste. The whole of the Thames became an enormous open sewer. Cholera spread through the water system and killed tens of thousands. Only in 1858 after the delightfully named 'Great Stink' did Parliament decide that something had to be done. Enter Jospeh Bazalgette. His plans included six primary interceptor sewers of almost 100 miles in length, fed by 450 miles of main sewers and 13,000 miles of smaller local sewers. Construction of the interceptor system required 318 million bricks, 2.7 million cubic metres of excavated earth and 670,000 cubic metres of concrete. Bazalgette calculated the pipe diameter based on the densest population and the most generous quantity it could produce – and then doubled it. The original diameter would have led to sewers overflowing in the 1960s; as it is, Bazalgette's sewage system is still in use today.

It was not an easy project. It took 17 years and cost about £1.7 billion in today's money. Whole sections of London were closed for years at a time while the vast tunnels were dug. Trade was disrupted, while as for personal inconvenience when your street was dug up... let's just say that modern roadworks are nothing to it. Bazalgette needed to negotiate with numerous groups, all of whom had conflicting interests. He suffered a breakdown as a result.

Figure 5. Exploring sewers still in use under London (by sub-urban.com)

3.1. The size of the task

Integration projects suffer from similar issues – including the odd breakdown! In most companies integration involves linking systems that reside on different operating systems, use different database solutions, different computer languages, legacy systems that may no longer be well supported, or new systems purchased to solve problems. It links different parts of your business together, but also reaches further to include suppliers and even customers. With the number of conflicting interests to manage, the potential for disruption is huge – no wonder it's only when a Great Stink occurs that companies are prepared to begin the project at all.

And the cost and disruption of architectural change now needs to be balanced against future needs. Bazalgette doubled the pipe diameter, meaning holes had to be dug larger and hundreds of thousands of pounds extra were required for materials. What if Bazalgette's gamble had been wrong and London's population had stayed the same or fallen? His over-engineering would have cost the Victorian taxpayers very high.

Activity 2: Integrate early, integrate often

> "If you integrate throughout the project in small amounts you will not find yourself trying to integrate the system for weeks at the project's end while the deadline slips by."
>
> Integrate often from Extreme Programming

This activity should take about 15 minutes and should be followed up with actions on an on-going project.

Projects commonly consider their integration efforts a task to be performed at the end of the lifecycle and consequently suffer problems that lead to the deadline being missed.

Review at least three of the projects that have previously run in your organisation. For each of the projects draw a timeline from when the development effort started until the project was completed and released. Indicate on the timeline all integration events.

Examine the timelines. Consider how and when the projects approached integration. What benefits did they derive from integrating and, to your knowledge, what problems did they encounter?

With the information that you have gathered find a current project team that is willing to review their own integration efforts with you. Understand how they could change their strategy to increase the likelihood of successful integration. Help them identify opportunities for early integration. Work with the project team to help facilitate that change.

3.2. And then we make it harder

Yes, as if things weren't hard enough, we have processes that actually make integration more painful. Many IT projects include an integration period at the end of the project timeline, when all the branches can be merged and tested to ensure the application is working. It's as if Bazalgette dug all his tunnels separately and then decided to go back and join them up a few days before the ceremony to open the system.

Activity 3: The X Factor

This activity will take about 5 minutes and you need to do it with at least 4 colleagues. You'll need to find a vacant room, with as little furniture in it as possible – alternatively you can do it in a hallway.

The objective of this activity is to simulate over two rounds how, when a system is moving forward in development, coupling between its constituent parts can affect the ability to make change. You can experience this by moving a group of people in an "X" formation (seen below) from one side of the room to the other. To simulate change going on within the system, while moving from one side to the other, the person at the core of the formation needs to turn around 4 times.

First attempt

Second attempt

In the first attempt, all the people must be physically linked to each other, as close to each other as possible. Travel from one side of the room to the other, ensuring that in doing so the person at the core also turns through 4 whole revolutions. At all times, the other people must remain fully connected to the person at the core.

In the second attempt, the group should work together to make sure that the formation is maintained but the group doesn't need to be physically linked. The person at the core still needs to turn around 4 times.

Commentary:

Did you notice that there was less disruption in the second attempt? This activity illustrates the difference between a system that has tight coupling, and one that doesn't; when you reduce coupling between systems it is easier to change one of them without affecting the others too much.

Coupling and cohesion

And again like the sewage system, our IT systems are often so tightly-bounded that it is very difficult to change, upgrade or repair them without digging everything up again.

Thankfully, unlike sewage systems where over-engineering really is the only solution to future woes, IT systems don't have to work this way. True, we can build in extra capacity from the start, but we can also make our lives much easier by employing loose-coupling. Although these terms have a specific IT meaning, they take their concept from an older engineering idea. If you want to join two copper pipes together, you could weld them together into a seamless whole. But then if at some point you wanted to change one of the pipes for a bigger one, or it had corroded, you would have to saw or rip them apart. It would be much easier if you created a joint which had the flexibility to hold two pipes. Then you could swap a new pipe in, or add a second joint and attach a bigger, or smaller pipe.

Loose-coupling essentially allows us to look to the future without needing to massively over-engineer and guess what capacity we might require 5 or 10 years from now. Service Oriented Architecture (SOA) takes this one step further.

In an ideal world, each unit of code is responsible for a single purpose, all of which hang together as a logical whole. This is called cohesion. High cohesion is not only tidy, but it makes a system very easy to maintain. If you need to make a change, (presuming that the interface of the unit remains the same), you should be able to swap out the single unit without enforcing extra changes throughout the rest of the system. Instead, the single change would be sufficient and would propagate throughout the system.

Cohesion breaks down when a change is required and instead of splitting a unit to maintain individual responsibility, the unit is grown to be responsible for two or more purposes. This may often be the easiest, quickest way, but it leads to low cohesion, which makes the system hard to maintain, and brings us back to our big ball of mud.

Up until Bazalgette, the River Thames was both London's main source of drinking water and its main sewer. Bazalgette's brilliance involved separating the functions, building sewers parallel with the Thames and taking over some source rivers altogether for effluent. This meant the Thames became clean again and could once more function as a major source of drinking water. His solution was an excellent example of cohesion: by separating the responsibilities, the system became easier to maintain – and resistant to cholera.

Flexibility, structure and planning for the future comes – like Bazalgette's engineering – at a cost. Deciding on whether that is a cost your business should meet depends on your business – and not your technology – objectives.

Activity 4: Mapping systems

In the majority of organisations today you will find a wealth of applications all of which have some form of dependency on at least one other. As you understand the application landscape further, you will find that there is one system at the very core of your organisation's ability to function. That application will have a great many externally dependent systems.

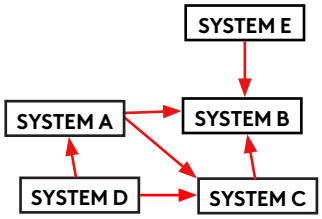
This activity should take about 45 minutes.

In this activity you will create a map that shows all the dependencies between the applications within your organisation. It's quite possible that such a map already exists in your organisation, but the purpose of this activity is to help you understand the dependencies in your application landscape and thus the process of creating the map yourself is still valuable.

Start by listing all the applications in your organisation. Consider their interrelations - look at how the data flows; examine the common interaction points. As you build this information a picture will begin to emerge of how all the applications relate to one another. When you think you have a comprehensive list attempt to draw a map which details how all the applications relate. To do this, draw a box for every system that you have identified and an arrow pointing to another system for every dependency.

In undertaking this activity you should be able to identify the application that is at the very core of your organisation. There might be more than one, that's fine too.

Keep your map, we'll use it later.



CASE STUDY: Failure of a fully integrated care records system

Rarely has there been a more embarrassing example of a massive failure of an IT project than the creation of a fully integrated care records system across the NHS (the UK's National Health Service). £2.7 billion was spent. It failed completely.

To use the words of the Commons Select Committee investigating the project, it resulted in 'many hospitals being left with little choice but to continue with out-dated interim systems that could be very expensive to maintain and to upgrade.' Or as several journalists put it more crudely: 'a couple of billion spent in order to email an X-ray.'

Integrating thousands of hospitals and primary care trusts, etc. was a challenge, yet the real failure was not technical, it was strategic. Integration depends on an ability to standardise data and processes, and the benefits of standardisation simply did not apply when considering the extremely varied and valuable needs of differing NHS bodies, from acute hospital departments to general care trusts. Each of these bodies needed specialist systems, and therefore needed to be supported with specialised, non-standard integration. The costs of that may have been high, but they were necessary. By contrast the costs of a standard integration project were astronomical because they were incapable of delivering any benefit. Failure to recognise this strategic fundamental error resulted in waste and recrimination – plus a bill larger than Bazalgette's for building London's sewage system.

Figure 6. A patient's own care record

4 BUSINESS OBJECTIVES IN INTEGRATION

Unlike in the NHS example we have just discussed, for most organisations, failing to deal with a messy, unstructured system has major penalties; seamless integration has real benefits.

4.1. Getting integration wrong has penalties

Any join within a system or between systems has the chance of failure - literally a disconnect where information can be lost. This information might be orders or records and losing such information will impact on revenue and customer satisfaction. These disconnects exist because of the very way organisations are structured. If 'finance' is responsible for billing, but 'customer service' is responsible for an address change – each department may have their own data, software and customised elements within the system. When a customer rings up to find out what's happened to an order and change their address, they don't expect the person answering to have to look in two different databases in order to process the call – perhaps changing the address in one record but not the other. Often that's exactly what happens. It's messy, it's inefficient and it has real associated business risks.

A department may know it's doing something out of sync with the rest of the company, but because staff prefer their customised solution or perceive it as beneficial, they want to keep the status quo. If change appears to be suggested merely to tidy up the IT department's life – it is unlikely to garner much support. It can be hard to find the corporate will to break down functional barriers, even if an IT manager can prove that a lack of integration is hurting business performance – perhaps through a lack of shared ideas, information or unacceptable errors in data. Departmental performance is rarely measured against such issues or potential improvements and individual functions are therefore highly unlikely to accept the case for change. Senior management need to be convinced by solid business risks and opportunities in order to change departmental goals and metrics to encourage system integration.

CASE STUDY: Extreme integration - the top down approach

Amazon CEO Jeff Bezos is unquestionably a brilliant thinker, and like many brilliant thinkers who have steered a company to success, he is not averse to setting a very clear direction – or issuing mandates that make 'ordinary control freaks look like stoned hippies', as an anonymous employee put it more colourfully.

Figure 7. Amazon's web services

One of Bezos's most famous directives was to make every piece of work the company did service-oriented. All design had to be created so as to be able to work with external developers, even if the work were a completely internal design. Teams had to use service interface calls. There was to be no direct linking, no direct reads of another team's data store, no shared-memory model, no back-doors. And to make the point clear – not attending to these rules would be regarded as a sackable offence.

It required an enormous change for every developer in the company and strict enforcement to stop the very human desire to do this bit – just this bit – differently. The real pay off to all the pain, was when it became clear that the SOA nature of Amazon's work meant the capabilities developed for internal processes, could now be sold externally as repurposed platforms. Moreover, the embedded SOA improved their accessibility – hugely widening their potential reach. Amazon Web Services includes the Amazon Elastic Compute Cloud, the Amazon Elastic MapReduce, and the Amazon Relational Database Service.

With this kind of political will, integration projects are easy. Unless you are also a brilliant CEO with the complete faith of your board – you'll probably need to go for a more consensual style of management that builds a business case and leads by attraction.

Activity 5: It's good to talk

Using the system map that you created in Activity 4, pick a system that has more than one dependent system (it doesn't have to be the one at the core, preferably one that is relevant to your own department/business unit) and investigate how each of its dependent systems communicate with it.

Identify the core pieces of information being passed between the system and its dependents. With this information, attempt to identify areas of commonality. Could this information be useful to other systems, other areas of the business? Repeat this process with other systems until you find a subset of systems that offer information, or services that could be aggregated to provide benefit to the organisation (e.g. a single point of call for all customer related data).

In understanding the data that we have suggested you collect above, we've tried to help you identify an opportunity to create a service that could be consumed throughout the organisation. In organisations that are pursuing a Service Oriented Architecture, this would be one of the possible start points. How many other services can you identify opportunities for within your organisation? Could you make a significantly compelling case for these services to be developed? Think about the benefits that you would list to support the development of a service.

Commentary:

In this activity, we've asked you to consider the creation of a service that would provide key data to applications new and old. As we'll go on to see, making this data commonly available can aid innovation in your organisation. You may also consider creating services that offer common functionality or operations.

CASE STUDY: The disaster of a poorly integrated supply chain

In 2004, the supermarket chain Sainsbury's published its first loss in its 135 year history. The main reason was a £260 million write-off from the supply chain and IT investments – accompanied by a warning that the supermarket would need to spend an additional £200 million to fix the problem.

Sainsbury's used a number of IT systems to manage its supply chain, including outsourcing to Accenture; automated service depots; distribution warehouse management systems provided by Manhattan Associates; an Eqos alert system to improve stock availability in store, and Retek to forecast product demand in Sainsbury's stores. The real trouble did not lie in any of the individual components, but in the integration between them. The result was a lack of availability of key products in store during the busiest retail period, leading to customer dissatisfaction that was so severe that Sainsbury's had to hire temporary workers to manually order and stack shelves. Inventory held in warehouses went off, leading to high wastage costs. All of which was exactly the opposite to what they had been aiming for.

Figure 8. Empty shelves at Sainsbury's

John Bird, Marketing Manager at Manhattan Associates, said 'In any complex re-engineering initiative, the reality of productivity gains do not just come from technology. It is a combination of getting technology to work together, change management and ownership by the customer of the project. I think [Sainsbury's] recognises this.'

4.2. Get it right and the whole business reaps the benefits

Integration is sometimes presented as something you only notice when it goes wrong, as if only its failure were costly. But when done right, integration is about more than everything running smoothly, it also assists multiple units, functions, and sites to work together to increase capacity, improve performance, lower cost structure and discover opportunities for improvement. It's a virtuous cycle: integration frees your people from performing repetitive processes (required when automated systems are not in place or not working well). Instead, people can take advantage of opportunities which integrated systems have made clearer and more actionable. These can range from reducing development time to market, to invisible back-end processes, like credit checking a customer.

Disney – one of the world's most successful companies – began drawing an integration map in 1967. Their strategic realisation was that by integrating the individual businesses, they could build revenue opportunities that were more than the sum of their individual parts. By seeing that music, comic strips, publications, toys, and even the theme park all had a role in promoting the films and building on them, as well as feeding back in to one another, they could exploit the creative content. They did this not once, but seven or eight times to maximise revenue with impressive results. The Lion King, first screened in 1994, grossed $50 million at the box office, but the surrounding activity (material is still selling today), was eventually worth more than $2 billion to the company.

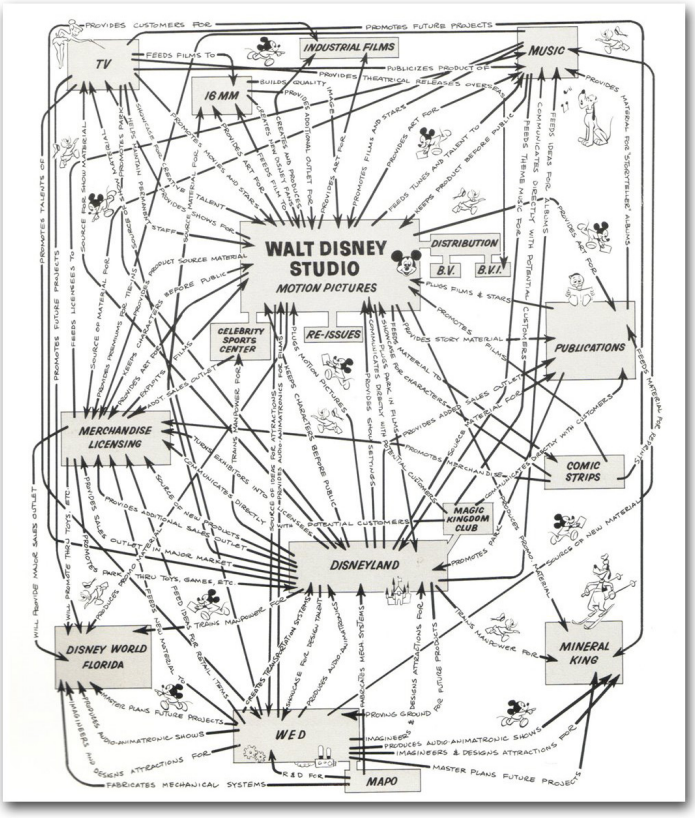


Figure 9. Disney's integration map

This is enterprise business integration with exceptional business results, but it is underpinned by an IT integration that enables assets to be moved about, used and protected. Making this closely integrated business work required IT integration and therefore was likely to garner the political will and investment needed to make it happen.

IT is often the business unit with the most experience in integrating business change programmes, such as implementing common infrastructures, shared databases, and cross-functional and enterprise systems. This is a golden opportunity for IT to take the lead in realising the benefits of closer collaboration to position the business for long-term success.

We began by looking at the improvements doctors made when they automated systems. These systems are far more sophisticated than our simplified discussion about filing patients' notes. An implanted cardiac medical device, for example, not only monitors and regulates a patient's heart, but can wirelessly communicate any cardiac event to a specialist cardiac unit. The data the implant provides allows doctors to remotely monitor the patient, diagnose the severity of the event and then take appropriate action – often before the patient is aware of any serious symptoms.

CASE STUDY: Nike's introduction of an integrated database

Nike introduced an integrated database within its SAP ERP system for every employee in North America and EMEA. It meant that everyone agreed on business practices and common data definitions before the software went in. The idea was to bring the manufacturing cycle for a pair of trainers down to three months – which would align it with retailers' ordering schedule and permit Nike to manufacture to order, cutting inventory holding and reducing risk.

Figure 10. Nike advertisement

To do so Nike implemented ERP software, specifically SAP's R/3 software, with i2 supply, demand and collaboration planner software applications and Siebel's CRM software integrated into the overall system using middleware from STC.

One element of the project went very publicly wrong – the i2 planner had a glitch that ramped up production on an unpopular model and massively under-ordered on a bestseller. It cost the company $100 million in lost sales. Astonishingly, the CIO kept his job and the project continued – why? Because Nike's senior managers understood the overall business benefits of an integrated supply chain system and backed the project. A forward planning tool was a minor element – the business decision to include it was a larger error than the technical glitch. Nike moved the planning function back into their ERP SAP which had the additional benefit of allowing them to simplify their integration requirements.

As well as the big prize of inventory and manufacturing control, a single database provides financial visibility, cash flow management, revenue forecasting, and the ability to take advantage of shifting exchange rates (a significant revenue stream given that Nike operates in so many countries). Success in reducing manufacturing cycle time makes a $100 million loss look like loose change.

5 HOW DO WE DO THINGS DIFFERENTLY?

This section focuses on behaviours and changes that IT departments can make to ensure they meet the business objectives of integration and don't end up with the enormous white elephant projects that make such embarrassing headlines in trade publications.

5.1. Concentrate on value not just cost

We've discussed why an IT system's history and organisational structure can create poor architecture. Other reasons are tight deadlines and an emphasis on getting the cheapest possible code or software, meaning that planned architecture and integration take a back seat. It's a business mantra that doubling volumes reduces costs by 15 to 25 percent, and doubling variation increases costs by 25 to 35 percent. The ideal low-cost model for IT, therefore, is maximum standardisation and maximum volume. However, this needs to be weighed against the potential benefits to the whole business of higher IT investment in fulfilling unique needs.

Sometimes what you've got (like the manual filing system in the GP surgery) is just fine for your enterprise. The problem is that it doesn't scale up. Every time you want to add an application, managing the connections becomes more difficult. For large or complex organisations, architecture (often SOA) that is built to permit new growth or variation is almost essential because the business benefit value outweighs the IT cost.

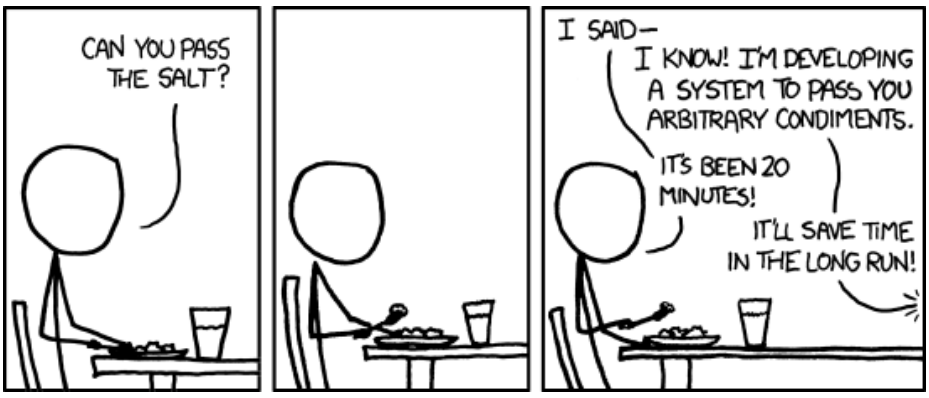


Figure 11. The General Problem (used with permission from xkcd.com)

This investment in architecture and integration is one that pays dividends in the long-term. Of course, long-term investment is no good if you've gone bankrupt before you can take advantage of it.

Architecture must also provide some benefit in the short to medium term that can be readily grasped by people other than those doing the building and designing. Otherwise it is doomed to failure – not only will the necessity for good architecture not receive buy-in and investment, but its potential will also never be realised.

CASE STUDY: From cost-cutting to company transformation

BT's technology group developed a strategy to create a platform that would enable them to sell services. They had big plans about how this could transform a company still half-trapped by its history as a telephone provider into a dynamic telecommunications services provider in the broadest sense. By creating an Internet Protocol (IP) that would permit identical service from the network in up to 167 countries, they gave themselves the foundations with which to provide a host of different products and services externally, and manage a flowering of ideas and entrepreneurial drive internally. Yet this ambitious aim was presented primarily as a cost-cutting and efficiency exercise to shareholders and the media – a short-term goal that felt unthreatening and provided instant value.

It certainly yielded some impressive immediate savings: almost 20% in unit costs, amounting to £240 million over two years. BT had 14,500 IT workers split across the enterprise in departmental silos. Together they worked on over 4,000 projects, few of which were aligned to one another or to the company's overall goals. The team set out to reduce the number of systems to 100, running on 14 core platforms built around a service oriented architecture, but allowing BT to build its own services on top.

Once this was done, BT could begin to capitalise on its newly integrated systems and develop the 'New Wave' projects that would bring it revenue – the long-term and ambitious reward for getting integration right.

5.2. The joy of standards

Engineering standards are so obvious to us in our daily life that we barely consider them. We know that drill bits, rawl plugs and screws all come in matching sizes. We know that plug sockets and plugs are designed to fit one another. We get cross when we remember that these standards do not apply everywhere and we try to plug a phone charger into the socket in a Bangkok hotel room only to realise we have forgotten our travel adapter. These standards make life easier – not only for customers, but also for companies. You don't have to spend lots of time designing your power plug, you can put all that energy into the device itself.

Standards can apply in every area of our businesses. We can standardise the way in which we take information. For example, it may be that the finance and the customer service team hold two entirely different sets of information on different databases. Integrating these can be a real headache. One of the simplest ways to do so is to standardise the type of information held – name, surname, address, bank details, etc. The interface does not have to be the same – you may not want the help desk to see a customer's bank account information, but by ensuring the information is all taken in the same format and that the core content is the same, you'll make your integration task much easier.

But standards destroy creativity

Throughout this course we remind you that although variation costs, it often has value that makes it worth bearing with. Some people assume that standards means eliminating variation. This isn't true – the trick is in picking which elements to standardise, and which to keep variable. Consider the Apple App Store. This runs an extremely clear standard of what will integrate with the platform. Developers know exactly what will work and they know exactly which devices their app needs to work on (only a handful of models for just three devices at the time of writing). This does not restrict the hundreds of thousands of applications that can be written, from a graphic of a beer glass emptying as the user tips the phone, to mobile banking applications. Indeed, the imposition of standards actually assists creativity by making it clear what to focus on. A good comparison is the Android market where a developer needs to write an application that will work with over 5,000 devices – most of the time the differences won't be too extreme, but the differing screen size (for example), would make applications like the emptying beer glass significantly more fiddly to achieve. Small loss, you might think, but since graphic imagery is an important part of many mobile applications the relative lack of standards can prove extremely time-consuming.

Standards often work best when we are able to break them down. If we create a high-level, over-arching standard which tries to encompass everything, it proves too restrictive and loses the value that exists within variation, but if we manage to create tightly-defined low-level standards, we can make our work much easier.

This is the way computers use the Internet – the layers are separated in such a way that that standards can exist at a bit and byte level and remain entirely distinct from the content or its presentation. We keep our data standard separate from our transfer standard.

Which standard to pick

In fast-evolving industries, standards take a while to form depending on which innovation proves the most successful and how many companies can reach consensus. Standards can fall out of date quickly and have to be revised if an industry-changing new norm emerges. Occasionally the standard is so convenient or entrenched that it may remain – even if it doesn't provide the best results. There is always a risk in being an 'early-adopter' because you might select as 'standard' a format, language or hardware that does not then become the eventual industry standard.

When Sony launched Betamax, they positioned it as the highest quality picture recording format available – most people still agree that it was a superior product. But in the 'video format war' with VHS, Betamax lost. Far more consumers wanted the cheaper prices and longer recording time than wanted perfect picture. Consumers who selected Betamax video cassette players and recorders lost out since the product was eventually discontinued and few films were made available on the format.

It's rather similar to the 'standards war' that happened between George Stephenson (inventor of the steam locomotive) and Isambard Kingdom Brunel (arguably the greatest engineer of his age) over railway gauges (the gauge being the distance between the rails). Stephenson copied the wagons and carts that people already used and set the rail-track gauge at 4ft 8 ½". Brunel, by contrast, calculated that a broader gauge would allow carriages greater stability when travelling at speed. He was looking ahead to a time when locomotives could go even faster than the dizzying 24 m.p.h. they travelled at in 1825, and he designed the Great Western Railway to work on a 7ft gauge. Brunel lost in Britain, and the GWR was converted in 1854, but Brunel won the gauge war in India and Russia, which is one reason why sleeper cars are far roomier in those countries.

The point of both stories is that the best standard is not always the one that wins. Organisations have to balance the risk of adopting a standard early before it has gained critical mass, with the opportunities offered to ease integration pain by plumping for a single standard. It also means that organisations need to pay as much attention to what is popular as to what is the highest quality.

Doing it anyway

Just like with drill bits and plugs, having standards makes integration and portability easier for software and technical hardware. With a standard communication protocol, for example, rather than creating code especially to allow two databases to talk to one another, you can take this as a given, and spend your time, effort and creativity on doing something valuable with the data itself. IT departments often lament that they have to run harder and harder just to maintain creaking systems as they are – standards are one of the best ways of making maintenance work easier, allowing teams to develop new functionality instead.

Sometimes applications may require something 'extra special', probably from proprietary services, but when combined with an architecture designed to keep these elements as separate modules, it provides an excellent base to help ease integration problems.

CASE STUDY: The USB interface

The USB (Universal Serial Bus) is one of the greatest examples of a successful interface ever created. It allows users to instantly plug in a host of peripheral devices to their computer – import and export data, recharge batteries, and use the common capabilities of a device without installing a special device driver. From mobile phones and cameras to external hard-drives and even bicycle lights, billions of devices now connect to our computers via USB. Where would spy films be without the humble memory stick?

This standard interface did not appear without effort. Before the USB, devices each had their own installation procedure which often involved adding a card to the computer and rebooting it. In 1995 Intel set up a formal company with major industry players to try and fast-track the creation of a standard interface. They recognised that consumers – and thus they – would benefit in the long-run, but in the short term it involved giving away impressive quantities of their own research and development under a royalty-free licensing scheme. Crucially, they created a standardisation specification so that manufacturers could develop hardware that would be consistently compatible with the USB system.

Activity 6: The benefits of standards

There are luvvies out there who could bore you senseless about how beautiful the English language is, and if you'll let them, even go on tell you about medieval poetry too.

Let's have some fun with English and while doing so, understand how standards are used within it. We have three exercises for you to do:

1. Standardised letters

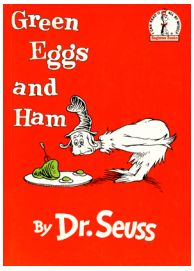
Given the phrase "SCIENTIFIC METHOD" write as many English words as you can using the letters that it contains. How many words can you construct? Try to write at least 20, perhaps 50. We know that you could write as many as 72.

2. Standardised words

Given the following 50 words:

> "a, am, and, anywhere, are, be, boat, box, car, could, dark, do, eat, eggs, fox, goat, good, green, ham, here, house, I, if, in, let, like, may, me, mouse, not, on, or, rain, Sam, say, see, so, thank, that, the, them, there, they, train, tree, try, will, with, would, you"

How many different sentences could you write? Try to write at least a few. Do you think you could write a whole story? No? Well, in fact using the words above, is just what Theodor Seuss Geisel did in 1968. He wrote an entire children's story called Green Eggs and Ham.



3. Standardised punctuation

Ask a few colleagues to punctuate the following words:

> "woman without her man is nothing".

There are multiple ways that it could be punctuated. Indeed, you might find that women punctuate the phrase differently from men, preferring "Woman. Without her, man is nothing", whereas men tend to punctuate it "woman without her man, is nothing".

Commentary:

We've just given you three examples of how a standard set of artefacts - letters, words and punctuation marks - have been provided in the English language. All of them have a discreet purpose and there are standards for how they should be used together too. What's interesting to observe though, is the freedom any user finds in them for expression. In fact, this standard is so good, that it enables as many as 1.8 billion people today to communicate freely on any topic or subject they choose.

5.3. Keep on integrating

There's a great adage that in software, when something is painful, the way to reduce the pain is to do it more frequently, not less. Rather than viewing integration as a phase that occurs at the end of the project, it is done as part of every change to the system. A new working piece of code must be integrated. If it breaks the system or does not fulfil the customer's acceptance criteria at the time it is introduced into the system, then the teams must work to fix the problem before they can move on and add a new piece of code. The benefit is that at any given time, the piece you are building can be released.

Some teams choose to do this weekly – all new code must be committed by Tuesday lunchtime. The system then compiles – things go well, or they don't... In order to make the process as painless as possible, all developers should have already built and run all their own unit tests and passed the appropriate acceptance tests. Because the build is done so frequently, it is easy to trace who did what, or who caused a problem. In general, this makes people far more careful about the quality of the work they commit. Since they know that they will be the ones fixing any errors, they are motivated to fix them in advance.

The practice works so well that some teams don't integrate once a week, but all the time. As soon as a piece of code is complete, it is committed and then built. This is an automated process, run by one of several build-tools designed to make it easy. While some may simply indicate a problem or break, others go further, deploying the existing software into an environment intended to mirror or simulate the real production environment in order to check for further errors and problems when integrating with other existing systems.

This process requires investment from managers who may be anxious to see a different kind of progress. It requires investment from developers who are pushed to focus on Acceptance Test Driven Development, involvement in the careful writing of acceptance tests, and also into a more modular kind of code. While these have obvious benefits, the process isn't popular with everyone. Furthermore it requires up-front investment both in set-up, the tools to assist the team, and in hardware to make the build fast.

Another investment is the artificial deployment environment. Unless you have sufficient budget to build an exact replica, this will almost certainly be a scaled down version. But all this investment will be for nothing unless there has been intelligent thought regarding the testing design at every stage from acceptance criteria to models. Continuous integration tests won't be successful unless you're asking the right questions. The valley of IT is littered with the sun-bleached skeletons of projects which were tested with 50 users, only to find problems in the real world of millions of users.

In essence, continuous integration is a means of speeding up feedback on how your system is working – and that feedback is as much about insisting on good communication as much as it is about a technical solution. We discuss the subject again in the Feedback session.

CASE STUDY: Heathrow Terminal 5

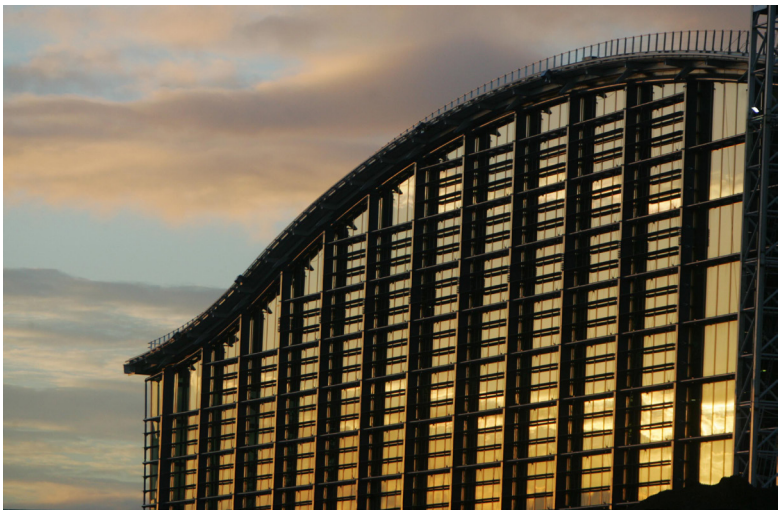


Figure 12. Heathrow Terminal 5

Terminal 5 cost British Airways (BA) and the British Airports Authority (BAA) £4.3 billion to build and outfit. BA said around £75 million of these costs were for technology, while BAA invested at least another £175 million in IT systems. The work involved 180 IT suppliers, installing 163 IT systems, 546 interfaces, more than 9,000 connected devices, 2,100 PCs, 175 lifts, 131 escalators, 18 kilometres of conveyor belts and "enough cable to lay to Istanbul and back".

The potential for something to go wrong was enormous. And it did.

The baggage handling system, to be precise. On its first day in operation the system was hit by numerous failures, from baggage handlers being told by the system the flight had left and returning bags to store, to losing bags in the system (including a foreign government VIP's). Bags are supposed to be automatically read, screened and sorted to their final build location via two electrical HELIXORTERS™. Early bags are stored in a warehouse with 4,000 positions, allowing individual access to each bag any time. Late bags should be transported at high speed directly to the head of stands where the awaiting aircraft are parked. Despite the time and investment in the system, under actual conditions, it went wrong. Later, baggage handlers claimed the system had failed during tests, but that managers had not released the resources to fix the problems – resulting in an embarrassingly public launch failure.

Although BAA claimed to understand that 'integration is not just plugging technology together; it's about people, processes and systems working together'. Sadly that was precisely what didn't happen.

The Emergn View

When the IT industry gets a new tool or methodology, it tends to act like the fitness industry, rushing to new workout routines, diet plans, and supplements with the promise of changing lives forever. But just as with any new solution in the IT industry, it's important to take a step back and evaluate whether it aligns with your objectives and values before making a commitment.

SOA is one of the biggest buzzwords in the industry, and yet a surprising number of business leaders seem not to know what it is – except that it's going to solve every integration woe there has ever been.

Service Oriented Architecture (SOA) is simply a design concept whereby you take the core business processes that aren't likely to change very often and ensure they're kept separate as independent services. Between these monoliths run presentation programmes which can accept and present the data from any user, and processes which describe differing routes in which to access services.

When you check out your favourite supermarket's online store, the web pages change from week to week. The marketing manager is trying different positions for the promotional offer, responding to customer feedback about how to make the site more navigable and experimenting with special offers that are introduced at checkout. This is a highly changeable presentation programme, but the backend IT services that search for prices and handle payment, stock and delivery remain the same. Booking a delivery slot will involve a complex business process based on calling up the customer's postcode, checking vehicle availability at the nearest store depot – a process which will evolve as the business does but which still uses the same assets.

By keeping elements separated, change becomes easier and less risky. The method of communication between these assets – and hence integration – all at once seems a simpler proposition. However, carrying out the separation and differentiating between a service and a changeable piece of information is not always straight-forward. Intelligent understanding of the business processes involved is essential – and because we are human, our judgement is fallible.

As is the case with the latest fitness trend, it's easy to be lured in by the appeal of SOA as a promising integration solution. However, it's important to remember that our needs are different, so it's essential to carefully evaluate its benefits and limitations before committing. There may be other integration solutions out there that are better suited to our specific requirements.

6 CONCLUSION

The Agile Manifesto states that 'the best architecture, requirements, and designs emerge from self-organising teams, not from top-down, rigid design requirements.' The world of technology changes and changes fast. Customers and requirements change too. That means that your plan for world domination is more likely to come true than your 5-year strategy to meet all your company's IT needs.

Figure 13. Dr. Evil seeking world domination

There is a constant and sometimes creative tension that arises between the integration benefits and savings that come from employing standards, and the competitive edge achieved by producing something unique. Incremental growth is often essential to address local and changing needs, and the burden of integration it carries with it must be borne. The trick is to make this burden as light as possible, but also to accept that even with the best architecture standards like SOA – there will be constant integration challenges to overcome. Recognising this, and seeing both problem and opportunity as applying to the company as a whole, and not just IT, is almost as valuable as the solution.

In this session we looked at the problem of integration as it is often experienced in software today and how large its impact is on the wider organisation. We also discussed the key behaviours that enable better integration.

Learning outcomes

Now that you have completed this session, you will have an appreciation of:

Why integration is a business problem and not just a technical issue

- Successful businesses digitise their core processes in order to free resource for other tasks
- The need to integrate is the result of a business requirement and thus decisions regarding integration should be taken by the business as a whole

Why IT systems are naturally 'messy'

- Systems build up over time rather than being planned from the beginning
- Natural doesn't mean inevitable

The resulting difficulty of integration

- Making numerous different systems work together means addressing conflicting interests
- Systems are often tightly coupled with low cohesion
- We need to decide how much planning for the future we intend to build in to our system

How business requirements drive integration

- Getting integration wrong can cost an organisation dearly
- Getting it right provides benefits that go beyond cost-saving

The behaviours that assist better integration

- Concentrate on value not cost
- Adopting and developing standards where possible
- Evolving towards continual integration

BIBLIOGRAPHY

Agile Manifesto, 2001. Principles behind the Agile Manifesto. [online] Available at: <http://agilemanifesto.org/principles.html>. [Accessed 3 November 2011].

Answers.com. Difference between Coupling and Cohesion. [online] Available at: <http://wiki.answers.com/Q/What_is_the_difference_between_coupling_and_cohesion>. [Accessed 26 October 2011].

Appelo, J., 2010. Management 3.0: Leading Agile Developers, Developing Agile Leaders. Addison Wesley.

Beck, 2009. Coupling and Cohesion. Three Rivers Institute – Thoughts On Programming, [blog] 7 April. Available at: <http://www.threeriversinstitute.org/blog/?p=104>. [Accessed 28 October 2011].

CIO, 2003. Business and IT Integration: The Key to Vanguard's Success. [online] Available at: <http://www.cio.com/article/29893/Business_and_IT_Integration_The_Key_to_Vanguard_s_Success>. [Accessed 3 November 2011].

CIO, 2004. Nike Rebounds: How (and Why) Nike Recovered from Its Supply Chain Disaster. [online] Available at: <http://www.cio.com/article/32334/Nike_Rebounds_How_and_Why_Nike_Recovered_from_Its_Supply_Chain_Disaster>. [Accessed 26 October 2011].

CIO Insight, 2008. How IT Transformed BT. [online] Available at: <http://www.cioinsight.com/c/a/Case-Studies/How-IT-Transformed-BT/>. [Accessed 26 October 2011].

Commons Select Committee, 2011. The National Programme for IT in the NHS: an update on the delivery of detailed care records system. Public Accounts Committee. [online] Available at: <http://www.publications.parliament.uk/pa/cm201012/cmselect/cmpubacc/1070/107004.htm>. [Accessed 3 November 2011].

Cummins, F., 2002. Enterprise Integration: An Architecture for Enterprise Applications and Systems Integration. John Wiley & Sons.

Computer Weekly, 2004. Sainsbury's Writes Off £260m As Supply Chain IT Trouble Hits Profit. [online] Available at: <http://www.computerweekly.com/Articles/2004/10/26/206226/Sainsbury39s-writes-off-163260m-as-supply-chain-IT-trouble-hits.htm>. [Accessed 29 October 2011].

Foote, B., Yoder, J., 1999. Big Ball Of Mud. [online] Available at: <http://www.laputan.org/mud/mud.html>. [Accessed 29 November 2011].

Harvard Business Review, 1998. Disney's "The Lion King" (A): The $2 Billion Movie. [online] Available at: <http://hbr.org/product/disney-s-the-lion-king-a-the-2-billion-movie/an/899041-PDF-ENG>. [Accessed 1 November 2011].

Harvard Business Review, 2008. Teaming Up To Crack Innovation and Enterprise Integration. [online] Available at: <http://hbr.org/2008/11/teaming-up-to-crack-innovation-and-enterprise-integration/ar/1>. [Accessed 4 October 2011].

Harvard Business Review, 2011. Learning to Live With Complexity. [online] Available at: <http://hbr.org/2011/09/learning-to-live-with-complexity/ar/1>. [Accessed 26 October 2011].

Hohmann, L., 2003. Beyond Software Architecture: Creating And Sustaining Winning Solutions. Addison Wesley.

Hohpe, G., Woolf, B., 2003. Enterprise Integration Patterns: Designing, Building and Deploying Messaging Solutions. Addison Wesley.

Humble, J., Farley, D., 2010. Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation. Addison Wesley.

IBM, 2010. Four Key Strategies For Enabling Innovation In The Age Of Smart. [online] Available at: <https://www14.software.ibm.com/webapp/iwm/web/signup.do?source=swg-rtl-spsm-wp&S_PKG=wpnt_IPM-4keyStrategies>. [Accessed 13 October 2011].

The Independent, 2004. Sainsbury's Supply Problems Lead To First Loss In 135 Years. [online] Available at: <http://www.independent.co.uk/news/business/news/sainsburys-supply-problems-lead-to-first-loss-in-135-years-533660.html>. [Accessed 29 October 2011].

Intel. Intel and USB: Make It Easier to Connect Devices to PCs. [online] Available at: <http://www.intel.com/content/www/us/en/standards/usb-standards-case-study.html>. [Accessed 26 October 2011].

Kuhn, D., 1990. On The Effective Use Of Software Standards In Systems Integration. [online] Available at: <http://csrc.nist.gov/staff/Kuhn/kuhn-sys-integ-90.pdf>. [Accessed 3 November 2011].

Reinertsen, D., 2009. The Principles of Product Development Flow: Second Generation Lean Product Development. Celeritas.

Ross, J., Weill, P., Robertson, D., 2006. Enterprise Architecture As Strategy: Creating A Foundation For Business Execution. Harvard Business Press.

Seddon, J., 2003. Freedom from Command and Control: A Better Way To Make The Work Work. Vanguard Consulting Ltd.

Schmidt, J., Lyle, D., 2010. Lean Integration: An Integration Factory Approach to Business Agility. Addison Wesley.

Wikipedia. Integration Competency Centre. Available at: <http://en.wikipedia.org/wiki/Integration_Competency_Center>. [Accessed 29 November 2011].

Yegge, S., 2011. Stevey's Google Platforms Rant. [blog] Available at: <https://plus.google.com/112678702228711889851/posts/eVeouesvaVX>. [Accessed 17 October 2011].

ZDNet, 2008. IT failure at Heathrow T5: What really happened. [online] Available at:<http://www.zdnet.com/blog/projectfailures/it-failure-at-heathrow-t5-what-really-happened/681>. [Accessed 29 October 2011].

VFQ Value Flow Quality

valueflowquality.com

emergn.com

by emergn