Agile Transformation – rethinking IT strategy in an uncertain world

Contents

Introduction

Organisations all over the world acknowledge that they failed to anticipate, plan for, or react quickly to the banking crisis and the subsequent economic dislocation. Leading executives in these businesses have become progressively more frustrated by their inability to execute strategies that can handle the uncertainties of our modern world. In reality, the future has always been uncertain but one trend has become completely reliable – and is probably the main cause of this frustration. Competition and rates of change continue to increase exponentially – driven, in large part, by the rapid emergence of new technologies. However, far from being an enabler of change in their organisations, technology seems to have become a major obstacle.

Since the millennium, the world has changed fundamentally as IT has come of age and become truly accessible in every market across the world. Customers interact with businesses through their computer systems. They expect rapid responses and will move rapidly when it is easy for them to do so. Enterprises must be truly agile to keep pace in this ever changing world.

There are two problems to solve. The first is that business and technology are now inextricably linked. In Financial Services, it is virtually impossible to have a workable business strategy that is not underpinned by technology. The second is that very few IT organisations are equipped to meet the needs of the business strategy due to the way they have invested in technology over the years. In most, the heritage of business systems is massively complex and poorly integrated – so it is hard to make changes, it takes a long time and it costs too much. As the pressure for change increases, the situation is becoming unsustainable.

We believe that a fundamental change in our approach to IT strategy is needed. Not only is it needed but far-sighted organisations are already well on the way to delivering it. Enterprises that fail to recognise the trend and invest will be left in a position where they can no longer compete both in terms of speed of response and cost of operations. Agility is the key – and our IT organisations must change fundamentally to achieve the level of agility that is needed in our uncertain world.

What has to change?

The root of the problem is complexity. The things the business want to do have been pushing the boundaries of system capability for years. The way most organisations build business cases tends to drive a focus on the value of the immediate requirement – and disregard the longer term cost implications for maintainability and ongoing change. Rapid consolidation in financial services has added to the problems for the industry. Shareholders have looked for rapid release of value and this has often been achieved with only a rudimentary understanding of the complexity generated or the ongoing cost of managing it.

A fundamental change in our approach to IT strategy is needed. Not only is it necessary but far sighted organisations are already well on the way to delivering it.

The challenge this leaves IT functions is relentless. The increase in complexity is exponential rather than linear. This exponential increase has a direct impact on the size and cost of the change portfolio – and the pressure to reduce service cost each year doesn't go away. In Financial Services, more and more of the available change budget is being consumed by 'mandatory' and 'keeping the lights on' projects and this limits the capacity – both money and available capability – for value-driven business change.

The IBM 2010 CEO survey[1] reinforces this message. CEOs described a world that is increasingly volatile, uncertain and complex. Furthermore, the vast majority of the 1,541 CEOs interviewed anticipated even greater complexity in the future. More than half doubted the ability of their organisation to manage it.

Managing the increasing demand for change has been a primary issue in most financial institutions for many years. IT organisations have tried numerous initiatives to increase their capacity for change including simplification, CMMI, setting up IT shared service organisations, service orientated architectures, agile development, global delivery, outsourcing and experimentation with on-demand or cloud based infrastructure provisioning.

However, such initiatives have proved hard to justify and, consequently, have often failed to deliver their full potential. Many have been discredited by bad experiences and are now seen as 'silver bullets' providing a temporary fix rather than the complete answer.

Business and IT leaders could be forgiven for reaching the conclusion that the 'nirvana' of a truly agile IT organisation is science fiction. Not any more. There is growing evidence that organisations which take these initiatives seriously, combining the approaches in the right way, can redefine the way in which they deliver information services to the business – and be really nimble in enabling business change.

Business and IT leaders could be forgiven for reaching the conclusion that the 'nirvana' of a truly agile IT organisation is science fiction. Not any more.

This paper proposes a new way of thinking about the maturity journey towards an 'agile IT organisation'. It argues that the journey needs to be holistic rather than single threaded, involving changes around customer management, technical architecture, application lifecycle development tooling and people. It also argues that we need to fundamentally re-think the proposition that a 'shared service' IT organisation delivers to its customers and how shared assets in IT are managed.

Where are we now?

To understand what is possible, we first need to understand where we are and where we have been. In most IT organisations the pendulum swings between two competing approaches to providing effective IT to a business – the 'Dedicated IT function' model and the 'IT Shared Service' model.

Historically, IT was delivered by a dedicated function, providing applications and infrastructure resources to a single strategic business unit or division. This approach enabled IT to deliver rapid small change, tailored to the needs of that business. It was responsive, informal and often innovative but came at a heavy cost to the wider organisation. It tended to encourage duplication and increasing complexity in the IT estate – because capabilities created in one business could not easily be shared with another. Typically, such organisations were also less effective at delivering large scale changes. Against this background, the case for sharing service – in both short term cost savings and long term strategic benefits – appeared to be compelling.

In this Shared Service model, IT resources are pooled so that they can be shared by multiple businesses. This approach is typically more efficient, gives businesses access to a wider range of skills, and promises standardisation and re-use in the estate. However, many organisations have found it hard to realise these benefits and find that a host of additional problems appear. As businesses are added to the shared service, speed of delivery and responsiveness are often impacted. Great standardisation and more formal processes can make IT feel out of touch with its customers and drive dysfunctional behaviours around demand for change.

A shared services approach should drive simplification of the IT estate but this does not always happen – and, even where it does, it happens at further cost to speed and function. Shared assets are hard to manage in the interests of more than one user group. Change becomes more complex to manage when interdependent components are shared, and local variation and innovation can be discouraged or – worse – ignored.

In most IT organisations the pendulum swings between two competing approaches to providing effective IT to a business – the 'Dedicated IT function' model and the 'IT Shared Service' model.

As delivery delays and costs spiral, many development departments have turned to outsourcing and global delivery to increase capacity and reduce unit cost; however, the results they achieve have been mixed. In many cases, cycle times have got longer and extra process steps have driven an even greater gulf between IT developers and the business they serve. Ultimately, as user satisfaction falls, IT appears progressively more insular and isolated so the business loses confidence – and starts to challenge the model.

Organisations have tended to react in two ways to these challenges, either relaxing the rules of the shared service and reverting to local variations – supported by local IT resources – or imposing a despotic governance model that drives global standardisation from the top. This 'power centric' model uses senior global 'business process owners' to drive standardisation and eliminate variation so that global applications can be implemented. Businesses get what they are given and local innovation is stifled. This approach has been used successfully for a period by a number of organisations – but can become unsustainable when power bases change. The reality is that few organisations of any scale are able to concentrate the political power needed to implement it.

The pendulum swing between these two extremes has characterised the world that most IT professionals have lived in throughout their careers. The leaders of IT organisations continually strive to resolve these issues but the pressure of day to day delivery makes it hard to achieve significant results.

Businesses need IT to achieve the benefits inherent in both models: speed and responsiveness on the one hand and sharing and leveraging across the wider organisation on the other. IT organisations struggling to make a shared service model work need to deliver a new proposition to their customers before it is too late.

Agile development pilots frequently fail to achieve their expected results and, in many cases, are just seen as an excuse for uncontrolled ways of working.

Overcoming complexity – breaking down the barriers

It has become very fashionable to talk about 'agile' as if it is a thing, a noun. There can be few companies in the world where development teams have not tried a range of rapid application development techniques including scrum, pair programming and iterative development – and quite a few have been burned by these experiments. They failed to achieve the results they expected and, in many cases, just saw the techniques as an excuse for uncontrolled ways of working – or straightforward anarchy.

In reality agile is an adjective, describing what you are – or, for most of us, what we aspire to become. It encompasses a whole set of capabilities and approaches which, when combined, can redefine the way that IT services are delivered, enabling a new level of responsiveness in the IT organisation, the delivery model and the IT estate. There is now a body of evidence which indicates organisations that embrace agile techniques and tooling are able to drive much higher levels of change productivity and development quality, and are able to do things in shorter and more targeted cycles. When these approaches are combined with other techniques such as componentised architecture, collaboration tooling and more consultative and responsive people skills, it becomes possible to start thinking about the 'product' that IT delivers in a radically different way.

'Waterfall' versus 'Agile at Scale' delivery

Many organisations have made a huge investment in 'Waterfall' based change. In a Waterfall process, change goes through a structured lifecycle including feasibility, requirements analysis, detailed specification, build, test and implementation. A central process repository, quality control gates, practice structures and CMMI certification have frequently been adopted in the quest for more consistent and higher quality change. The aim is well structured, controlled change with the minimum need for 'heroic' activity. Everyone can be trained to repeat their part of the process, factory-like. Build costs can be lowered by 'handing over' documentation to offshore or outsourced build teams.

Structured project practices tend to be rigidly enforced centrally and controlled by reviews to ensure that the mandated process documentation is completed and to demonstrate Sarbanes-Oxley compliance.

So the question becomes 'if this is the most effective way of developing software, why has IBM, one of the leading software development organisations in the world, systematically replaced Waterfall delivery with agile delivery techniques in all of its major software products over the last five years?'

The answer is quality, risk and responsiveness. In large projects and releases, Waterfall can give a false sense of security and control. Progress is measured using a number of 'input' and 'process' deliverables but, in truth, these are not robust indicators of delivery progress or quality. Often, in these large scale deliveries, a small under-estimation of effort can lead to real problems of timeliness and build quality that only become apparent at the back end of the change cycle. As time runs out, either testing is squeezed – with a consequent impact on quality – or the delivery is delayed. The only way to be sure of success in complex Waterfall projects is to over estimate delivery effort and plan for long testing cycles. The result is often lost productivity and sub-optimisation of the change capacity.

Furthermore, detailed analysis effort 'mandated' early in the change cycle can be wasted by misunderstandings, late rework or lack of understanding of the detail of the technical platforms and applications. Sometimes 'mandated' deliverables are even ignored or unused later in the change cycle.

None of these concepts are really new – and many have already created disappointment when used on their own – but together they have the potential to reconstruct the fundamental elements of the proposition that IT brings to its customers, enabling really significant increases in responsiveness and cost effectiveness.

We believe that the secret lies in the way that the techniques come together to create levels of capability that allow an IT organisation to break out of the 'pendulum dilemma'. At IBM we have charted this capability journey in a Maturity Model that shows how the different elements combine to change the game.

Together, these techniques have the potential to fundamentally change the proposition that IT brings to its customers, enabling really significant increases in responsiveness and cost effectiveness.

Finally, feedback from the 'customers' or 'sponsors' of the change tends to be either non existent, theoretical (hundreds of pages of documents presented for signoff) or during user acceptance testing where it is too late to influence the delivery safely. Put like this, Waterfall is less attractive.

The key difference with agile techniques is the ability to break up a release into many small chunks or iterations each delivering value. (See figure 1). Releases in IBM software group are now delivered by a series of small scrum teams located around the globe, building four week iterations of product increment (sprints), enabled by continuous build, test and integration tooling and effective collaboration tools. Product increment can be demonstrated to product managers to gain feedback as the release progresses and customers can be actively involved through 'beta' style programmes.

The outcome has been impressive. Quality becomes endemic, as a by-product of the way of working rather than somebody else's challenge at the back end of the lifecycle. Productivity and asset re-use are significantly higher and cycle times are a lot shorter. The results were so impressive that IBM's internal IT organisation adopted a similar set of principles (called GenO) for the next wave of its transformation. We may be at the beginning of the agile revolution in software delivery, but it is here and here to stay.
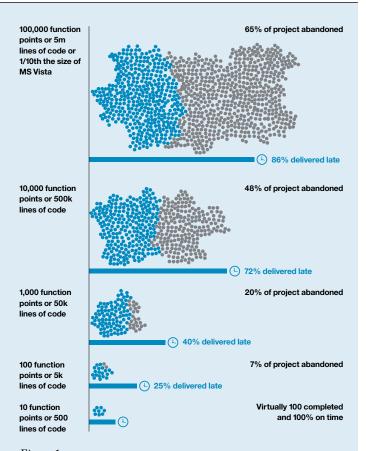
100,000 function points or 5m lines of code or 1/10th the size of MS Vista	65% of project abandoned
	86% delivered late
10,000 function points or 500k lines of code	48% of project abandoned
	72% delivered late
1,000 function points or 50k lines of code	20% of project abandoned
	40% delivered late
100 function points or 5k lines of code	7% of project abandoned
	25% delivered late
10 function points or 500 lines of code	Virtually 100 completed and 100% on time

Figure 1: Small is the new big: statistics that convinced IBM to explore the merits of more agile development approaches.

Source: Capers Jones, 'Patterns of Software Systems Failures' and IBM development statistics

The implications of the Maturity Model

The right hand side of the Maturity Model, in figure 2, describes a way of delivering IT services that is radically different from the left-hand side. It describes a position where IT is on the front foot, innovating in the right places, providing the right set of functions and delivering change quickly and effectively. Key shared assets in the IT estate are thought of and managed as 'products' rather than 'applications'.

Central IT provides a 'platform' where local variation and innovation can be facilitated and collaboration between users, IT and the organisation's business partners can take place. The core services of the IT 'product' are tightly controlled and managed under structured product release.

The solution delivery supply chain has been integrated through development and collaboration tooling, such as IBM's Rational Jazz Platform[2]. Organisations at the right hand side of the model become focused on outcomes and value, not just the delivery process. The benefits of this approach can be substantial –

reduced cycle time for delivery, higher delivery quality and higher asset re-use. Innovation is encouraged and is easier to leverage across the wider business. The Maturity Model recognises that progress is needed across all five dimensions for an organisation to become truly more agile.

This maturity journey has significant implications for the IT operating model:

1. IT must develop the capability to advise the business on how to use technology to leverage the organisations information assets and transform business processes; the way IT engages the business will change fundamentally.

2. A shared target architecture must provide a clear understanding of the role and value of each IT component, how much it is expected to change, and whether it is standardised or localised.

3. Adopting a 'Product Management' approach to managing and supporting technology assets, will drive a move away from monolithic change – driven by point projects affecting individual applications – to a world where shared assets each

Maturity level	Low			Medium		High		
Global maturity dimensions	Level 1 Legacy	Level 2 Foundation	Level 3 Intermediate	Level 4 Advanced	Level 5 Proficiency	Level 6 Excellence	Level 7 Best in class	
Customer management	Order taking	Requirements management	Structured demand	Requirements centre of excellence	Value aligned business service levels	Advisory partnership	Collaborative capability extension	
Product management	Project centric	Product definition	Product investment framework	Product release scope alignment	Service catalogue based charging	Integrated product development	Offering value optimisation	
Architecture	Monolithic	Architecture framework definition	Modularised code	Multi-tenant code base	Service Access	Data access	Integrated utility services	
Application lifecycle management & tools	Siloed application development	Federated configuration management	Automated unit testing	Continuous build and collaboration	QA integration	Performance integration	Support integration	
People and organisations	Siloed body shop	Flexible resourcing	Performance value measures	Service based delivery	Real time collaboration and sharing	Talent led innovation	Research integration	

Figure 2: IBM's Global IT Shared Service Maturity Model

have their own investment strategy and release programme. Agility at scale is created systematically at a 'platform' level not just through a few pilot projects.

4. Organisations will need to learn how to break down silos, encouraging and facilitating collaborative working. This also implies a new approach to the sourcing of services from third parties, better managing and integrating their work as part of the end to end value chain.

5. People will become more flexible and widely skilled, building on their traditional functional IT roles and becoming more aligned to the business function they support.

6. Tooling is critical in the transformation of the application lifecycle. Continuous build and integration will enable agile development methods to flourish. When combined with a structured release management approach, these can substantially increase delivery quality and reduce the risk inherent in very large change projects.

7. Infrastructure provisioning must become fast and dynamic – with services provisioned in minutes not weeks or months.

Making it happen with Agile Transformation

Becoming more agile implies several things:
- Breaking down change into smaller pieces or 'increments'
- Creating IT components that are easy to share and are decoupled from each other (making it easier to make change and control the scope of change projects)
- Changing the organisation's mindset to value different things in change delivery: responsiveness, collaboration and quality from the start.

This is not an easy journey but clarity can be gained by using the Maturity Model to help focus and shape an IT transformation roadmap and by adopting 'agile' principles for the management of the transformation initiatives themselves – breaking the journey down into a roadmap of small steps that create real increments in capability. Not just becoming agile, but transforming in an agile way.

Rethinking the business case for SOA

Like-minded developers looking for a better way of working created the Agile Manifesto (www.agilemanifesto.org) in 2001. Like agile development, Service Orientated Architecture (SOA) and componentised based design approaches have been around for many years. Some of the key principles behind SOA are to construct technology in ways that 'de-couple' component IT functions (in other words reducing the interdependence between two or more parts of the system) and to allow these component functions to be leveraged as services more easily by other applications. The reason this is important for agility is that it allows IT components to be changed without the risk of affecting other parts of the systems landscape, significantly reducing the change and testing effort. Not to develop like this would seem to be institutional suicide in today's fast changing market.

However, most SOA business cases focus on re-use as the primary benefit – a benefit that in reality never seems to appear. This has not been a compelling argument. When seen through the narrow lens of a business sponsor responsible for a single project, the perceived overhead of developing like this in the first place is not attractive – it risks delaying the benefit. Change business cases rarely include lifecycle costs in their baselines, and when they do the focus is typically on run cost. Finance functions have great difficulty in accounting for future change cost so it is often conveniently ignored in many project business cases. As a result the organisation's future change agility depends on the strength of business vision amongst change sponsors, not on institutional financial logic.

Looking at the architecture in the context of the Maturity Model implies that there may be an alternative way of building a compelling financial business case. SOA, business process management and componentisation approaches are easy bedfellows of agile development: in fact they might have been invented for each other.

There is potential to build a business case around a broader set of change delivery metrics, including both asset re-use and a wider range of change performance measures. These time based performance and change productivity benefits are potentially significantly larger than the savings that can be achieved by labour arbitrage alone. Realisation of these benefits requires a parallel maturing of the application development and organisational change capabilities to demonstrate the real business value of SOA agility.

Disruptive platforms

Developing the capability to deliver change at the right hand side of the Maturity Model has some interesting strategic implications. We believe there is a huge opportunity for companies that can do this to enter new markets or re-define their existing markets by using disruptive platform thinking. IBM's European CTO Rashik Parmar says:

"The concept is best explained exploring companies like Apple. They were able to conceive, build and create a collaborative platform (i.e. iTunes) that leveraged their unique core competencies to enter and dominate an entire market in a relatively short period of time. The platform has allowed an ecosystem of customers and third parties to collaborate and innovate in entirely new ways, ways that could not have been foreseen or planned by a single company. They are not alone; other organisations are thinking about break through in the same way."

The strategic question becomes how can organisations leverage their core differentiated capabilities and combine these with new technology to re-define fundamentally their propositions to existing customers or open up new markets? Answering this question puts IT right at the heart of creating and enabling business strategy.

The benefits of this approach are twofold. Firstly, it can reduce the risks of transformation by breaking the change into a series of short 'sprints' delivering incremental value and capability, and driving a closer alignment between progress and capability creation. Secondly, it encourages a more holistic view of the capability needed to be successful and helps to reduce the dependencies that typically derail such initiatives.

The starting point is to make a realistic assessment of current maturity against the five dimensions of the Maturity Model and to build the organisational case for change. Secondly, the target needs to be defined. IBM's view is that level four of the Maturity Model is a critical 'pivot point', at which it should be possible to establish sufficient change management and delivery capability to break out of the 'pendulum dilemma' described above.

Thirdly the improvement route map and timescales need to be defined. The Maturity Model greatly simplifies this process as it charts a focused journey to some of the most critical capabilities needed. Individual organisations will have additional capabilities and changes that are essential for them but nevertheless a focused transformation journey can be constructed by using the dimensions of the model as a spine.

Finally, the organisation itself needs to be aligned behind the journey. It is easy for existing development silos and power politics to derail or ignore the change. Adjusting the way that IT change is delivered is a bit like trying to re-wire the house with the power still on and it is easy for those who manage key areas of the IT estate undergoing the biggest delivery commitments to plead exception. However, it is those very 'target' and 'strategic' parts of the estate that need to become a priority for transformation if an organisation is going to become truly agile. Managers of these areas need to be realigned towards implementing and leveraging those new capabilities and practices that the transformation programme will create.

Conclusion: rethinking IT strategy

Agile IT organisations are not science fiction. They are starting to happen. In our uncertain world, there are huge risks for business leaders who focus on IT cost reduction rather than the value that can be derived from transforming IT capability – enabling effective realisation of business strategies (ahead of competitors) and driving business performance.

In many ways, the biggest challenge lies with IT leaders themselves. A relentless focus over the last few years on standardisation, industrialisation of change, centralisation, offshoring, shared service management and cost reduction has created a collective mindset, culture and patterns of thinking that could cause leaders to ignore, deny or dismiss what is happening in software engineering.

A second and even more dangerous risk for IT leaders is to adopt agile clothing and to pretend that the organisation is agile when it is not. This will only lead to failure and disappointment and run the risk of delaying the creation of the true capabilities that the organisation needs to become more agile.

All the evidence we have seen indicates that the right approach is for IT leaders to plan a capability journey across all five dimensions of the Maturity Model and recognise that becoming more agile is a key part of the future IT strategy; a strategy which is about how to put IT right at the heart of advising on – and delivering – effective, high quality and rapid business change. Delivering this strategy is not just 'no regrets' investment, it is arguably essential for survival.

For more information

To learn more about IBM Global Business Services, contact your IBM sales representative or visit:
ibm.com/gbs

About the authors

Jon Brock is Technology Strategy Leader for Financial Services in the UK, IBM Global Business Services. He is a transformation leader with a strong track record of shaping and delivering transformation initiatives for his clients operating at an executive level. He has extensive knowledge of the retail banking, general and life insurance industries. He can be contacted on +44 (0)7710 820755 or by email at jon_brock@uk.ibm.com.

Ewan Puckle Hobbs is a Senior Managing Consultant in IBM Global Business Services. He specialises in transformational change across business operations and technology in the Financial Services industry – and has a particular interest in how services can be effectively shared across both national and global organisations. He has travelled extensively, delivering change programmes in the UK, Europe, the Middle East and Asia. He can be contacted on +44 (0)7802 244781 or by email at ewanph@uk.ibm.com.

We would like to thank the following contributors who played a major role in the formation of the ideas behind this paper: Max Bocchini, Julian Ashley, Anubav Rajput, Mark Dixon, Clara Challoner Walker and Rashik Parmar.

1 http://www.ibm.com/services/us/ceo/ceostudy2010/index.html
2 http://www.ibm.com/software/rational/jazz

Please Recycle

VEW03001-GBEN-00