Agile Today
Is it time for organisations to take the plunge?

The Agile Journey
Corporations and government agencies are becoming more adventurous with their approach to software delivery and engagement models. Consequently the popularity of an Agile approach is also growing. This is understandable given the main challenges and drivers for change confronting organisations today. These include:

- Large backlogs of work
- Missing, incomplete, changing or ambiguous requirements
- Quality and rework issues
- Specialised roles leading to ‘not my task’ thinking
- Constantly changing business priorities
- Multi-tasking across several projects
- Unrealistic estimates and delivery dates set by the wrong people
- Lack of collaboration and trust between business and development teams
- Silos, handoffs, heavy processes, lack of trust and communication
- Lack of enterprise capacity measurement
- Heavy engineering and testing processes; no automation
- Overproduction and working on redundant features
- Slow time to market, too long to deliver
- Ineffective and wasteful meetings
- Lack of empowerment, low engagement and morale
- No ROI, value measurements or tracking
- Increased costs post-delivery
- Resource under-utilisation

Cornerstones of Agile: delivery focused, collaborative, self-organising, flat team structures.

In the first paper in this three paper series, we’ll discuss a wide range of organisational impediments and how Agile overcomes them. In part two, we’ll examine challenges that arise when moving from a traditional approach to an Agile approach in the software development lifecycle (SDLC). In part three, we’ll look at the “right” working software and the importance of individuals in an Agile team.

Before any organisation embarks on an Agile path, consideration needs to be given to truly understanding what Agile is, and importantly, what it is not. It’s also important to understand and align what organisations typically expect from a delivery model.

Traditional methodologies rely on extensive and predictive planning, comparative measurements based on the plan, and assumptions that what is predicted will unfold through staged releases during each phase of the Software Delivery Life Cycle (SDLC). As a consequence, opportunities to openly discuss issues in context and with the most appropriate people are often constrained.

The restriction of open communication is accentuated by a culture of baton passing (it is now someone else’s responsibility), the valve-like nature of Waterfall (everything flowing the one way), and the exclusion of the customer (‘we’ know what ‘they’ want).

Delays in delivery exacerbate these issues further. In fact, the greater the delays between deriving the solution and delivery can abstract the responsibility, accountability, and at times, the relevancy of the solution being delivered to the business. There are three areas in which these issues commonly show themselves.

1) **Schedule risk:** not everything is known upfront. Whether this is due to the tolerance of the requirement capture techniques or changes to end business practices; what was thought to be correct, may now not be correct. Traditional approaches tend to make the assumption that everything about the proposed solution has been understood, discovered and documented. This is rarely the case, and subsequently during development, discoveries tend to be ‘squeezed’ into the original, predictively-planned timeline; commonly known as ‘scope creep’. This creates milestone pressures on the delivery team; places the schedule at risk of not being achieved; and creates an atmosphere of driving to a date as opposed to driving to quality.
2) Rigidity: requirements change. Traditional methodologies typically have an inability to manage change or unknowns becoming known, particularly when lengthy requirement specifications have been signed-off and locked down early in the predictive SDLC. Change implies added time, anecdotally referred to as ‘slippage’. If the approach does not effectively manage impediments or change, milestones will come and go with no effective method of management, potentially leading to late delivery of an incompatible solution. Over the course of the delivery, ‘undocumented’ events will occur, whether in the form of unknowns becoming known, new requirements being requested, impediments arising due to system outages, or a higher than expected number of defects being detected and requiring remediation.

3) Protectionist: walls divide. Nurturing a sense of ownership is a positive trait, but not at the expense of promoting and maintaining full engagement with all stakeholders. Traditional approaches typically support a sense of ‘us’ and ‘them’ by virtue of their structure, vertical role assignments and low levels of engagement. This approach shows itself in the form of metaphoric ‘walls’ being built around the various teams to repel unsolicited engagement or interest, and in some instances to isolate knowledge or team members. Such behaviour may occur at both an intra-organisational level and inter-organisational level, and is particularly damaging when a solution’s end-users are excluded.

Delivering the right software

Some years ago, the focus was on working software, but my view is that all released software needs to work; my preference is to focus on the concept of the ‘right’ software. This term captures the reliance on continuous engagement between the business and the current state knowledge of their business practices. In this way software can be delivered that is not only working, but is also fit for its purpose and is aligned to organisational goals and practices. To deliver ‘right’ software the customer drives the backlog of work and helps prioritise exactly what features are to be included in the software product.

Studies show that lacking a continuous engagement model can result in end-users using only 20% of delivered functionality. Figure one illustrates this situation: responses ‘Never’ and ‘Rarely’ make up 20% of the answers; however the most significant figure is ‘Always’ – 7%!

Feature Usage

- Never: 19%
- Rarely: 13%
- Sometimes: 45%
- Often: 16%
- Always: 7%

Source: Jim Johnson of the Standish Group 2002
**Revolution IT - Delivering high-quality solutions for high-performance enterprise.**

At Revolution IT, we don’t just make software work, we make it perform superbly. From planning through to operations, we use proven methodologies to reduce IT spend, meet aggressive timelines and deliver outstanding application quality.

What’s made our name in the industry is our underlying focus on quality. To find out more about how our Agile services can benefit your business, talk to us today on (03) 96002566 or info@revolutionit.com.au or info@revolutionit.com.sg

---

**How the approaches differ**

Figure two shows the different success rates of Agile and Traditional approaches, and the different success factors. When Agile practices are implemented correctly, there is a 42% success rate compared to only a 14% success rate using traditional approaches. The ‘challenged’ section refers to factors outside the control of the delivery team; under an Agile model these factors are less substantial compared to traditional software development approaches.

However, the most significant section in Figure two is the failure rate. Under an Agile implementation the failure rate is less than one third the failure rates for traditional models. Also it’s important to note: the failed slice within the Agile pie was not directly attributed to the Agile approach.

The bar graphs represent factors identified as challenges to implementing Agile practices – I agree that the top-ranked challenge is the greatest impediment to introducing Agile practices within an organisation.

Organisational culture involves entrenched conventions and practices, empire protection, fear of change due to concerns about diminishing influence, reliance on traditional protocols that may not be relevant to Agile and a general comfort level with current practices.

This is reminiscent of Plato and the idea of man remaining inside the cave comfortable with his shadows cast by the fire, believing this reflects reality, rather than venturing outside to encounter new horizons, opportunities and reality.

Author, Stephen Berlyn

---

**Barriers to Agile Adoption**