Agile: The Good, The Bad and The Ugly

~ By Mark Flynn

More and more organisations are adopting agile delivery methodologies in answer to demands from customers to deliver change faster and at a lower cost. However, there is much scepticism within the agile community as to whether organisations are achieving the benefits they desire. Mark Flynn of Baringa Partners argues that to achieve agile success, organisations first need to understand the core objectives of being ‘agile’, and then establish whether it’s the right approach for them.

In our experience, most of the problems that occur on agile projects are largely caused by the fact that organisations fail to understand the core objectives of being agile. It is widely believed that by using an agile methodology, projects by their very nature will require less time, less cost and will be less risky. This is an incorrect and costly assumption to make.

For many organisations, being agile requires significant culture and process changes that cannot be brought about by solely implementing an agile methodology. True agility requires that you challenge existing processes, management hierarchies and governance forums - something which many companies are often reluctant to do.

The Ugly

Ugly implementations of agile are typically driven by project teams who have a desire to jump immediately into software delivery, avoid what is seen as traditional red tape or try the next ‘big thing’. They convince their line managers of these benefits, but no investment is made in communicating them to the business users or members of other delivery teams.

As a result, communication is poor and engagement with the key business users is non-existent. Individuals in the project team operate independently (‘to get their bit done’) as opposed to committing to delivering value to the customer as a unit. Agile methodology processes (e.g. stand ups, burndown charts and scrum of scrums) are performed daily, but key issues go unresolved and the project quickly goes off track. Recovery is possible but often lengthy, resulting in the organisation returning to status quo.

Ugly implementations of agile focus solely on implementation of the methodology, thereby missing the benefits that true agility brings.

The Bad

Bad implementations of agile are driven by a business’ desire to reduce costs, timescales and traditional red tape. The organisation fails to commit to the values of the Agile Manifesto in full and thereby never achieve true agility.

Investment in communication and education is made, but the tools and techniques required to enable agility
are ignored. Cross-functional teams are formed, but they are not empowered to make key decisions in consultation with the key business stakeholders and users.

Regular project reviews are performed, but the output is never acted upon fully, and key issues persist. The project potentially delivers, but the experience has been so painful that nobody wants to do it again.

The Good

Good implementations of agile are a result of the business and IT working together to implement a project delivery methodology that meets the needs of the organisation and invest in the organisational change required to deliver the benefits.

Key business stakeholders, business users, team members and senior management are educated and truly empowered cross-functional teams are formed. Investment in the tools and techniques required to support agile (standards, continuous builds, test driven development, pair programming etc.) are made to deliver a platform for success.

Regular project reviews are treated as opportunities to improve as opposed to a necessary evil. Change is embraced and planned for, so that when it arrives, it is absorbed with ease. Good implementations of agile commit to making the long-term changes required to reap the benefits of the methodology.

Implementing good agile is not an easy task for any organisation. It requires inherent changes to the way project delivery is approached. It challenges the traditional manager - team structure and relies on the decisions of an empowered few, as opposed to a large committee. It requires project teams to constantly point out their failures and work as a team to address them.

In return it offers frequent, timely and quality delivery that is prioritised based on actual business need. It builds a community of highly enthusiastic, cross-functional and empowered individuals all focused on achieving a common goal. It establishes platforms for long-term project delivery success by implementing standards, continuous integration, test driven development and joint application design practices.

The Path to Good Agile (Baringa Top 10)

1. Don’t Rush In

In our experience most organisations rush into agile as a response to business demands for faster and lower cost delivery. More often than not, the project fails, wasting significant financial investment.

Focus on understanding the real needs of your business. Ask yourself whether agile is right for your organisation. Inspect your current processes and adapt where possible to improve, perhaps using elements of mainstream agile methodologies to support business needs.

2. Educate Your Stakeholders

Agile is about frequency and quality delivery, as opposed to cutting costs and reducing timeframes. These benefits will come with time but the initial investment must be made up front.

3. Get One Committed Product Owner From the Business

In the agile world, a committed business lead known as the ‘product owner’ is vital otherwise the project is not worth doing (i.e. why deliver something that will not be valued). In a perfect world, there is only one product owner empowered to make key decisions. In reality this is not often the case, but the individual assigned to the
project in the role of product owner must a) commit at least 50 per cent of their time to the project and b) be responsible for obtaining the appropriate decisions outside his or her remit in a timely fashion.

4. Understand and Agree Your Definition of DONE

Most agile methodologies preach the delivery of production-ready software at the end of each iteration. In most new software implementations this will not be possible until the core of the solution is built and deployed. It is important from the outset of the project to agree your definition of DONE, deliver to it and avoid perpetual development or prototyping.

5. Build Your Cross-functional Team

Agile teams don't operate effectively unless they are truly cross-functional. The team must be capable of performing all the tasks required to deliver the customers’ requirements from the Prioritised Requirements Log into operation.

Once you have identified the members of your team, it's important to spend time shaping them into a team and ensuring each share a common goal - to deliver value to the product owner.

6. Invest in the Tools and Deliverables Required to Support Agility

The first few sprints of your agile project should be invested in setting up the frameworks and process automation tools required to support agility (e.g. software frameworks, continuous build servers, automating testing, user story repository, desktop video conferencing, chat etc.). In parallel, ensure the product backlog is populated with as much detail as time permits and solution architecture is produced. These deliverables, although potentially subject to change, will serve as a blueprint for your starting point.

7. Make the Quality of Retrospectives a Priority

Agile is all about inspecting and adapting. The vehicle used to perform this activity is commonly referred to as the retrospective. It must promote transparency and self-improvement. Actions coming out of the retrospective must be given high priority, particularly in the area of estimation. Accurate estimation is crucial to achieving a constant team velocity - eradicating the overworked and unmotivated team member.

8. Do a Solution Architecture Up Front

It is an agile myth that no documentation is required. Agile values working software more than documentation. However, if documentation is required to deliver working software - it is an absolute must.

In our experience, a solution architecture or a functional requirements list is one such key document. It provides the blueprint for what the project team is going to deliver. Without it, project teams will not be able to understand the impact of a change. If it is not maintained, future project teams will struggle to understand the core components of the solution.

9. Plan for and Embrace Change

Every change comes at a cost, even in agile. The Agile Manifesto values embracing change, but does not state that change should be embraced without a cost. It is important to remember to include the cost of any potential rework resulting from a change as part of your estimation process.
10. Foster an Agile Partnership With Your External Vendors

Agile typically doesn't sit well with traditional vendor contractual agreements (fixed price; fixed outcome) and can often lead to organisations spending more time navigating the contract as opposed to delivering customer requirements.

Work with your external vendors to build a transparent relationship and dually beneficial contractual agreement. If you don’t want to use a Time and Materials model, Risk Reward contractual agreements with clearly defined KPIs, in our experience, work well in agile engagements.