

Implementing projects the Agile way – few myths and misconceptions!

As the rate of change itself is accelerating, organizations constantly seek newer techniques to adapt to this phenomenon. Agile techniques (including its variants of SCRUM/ DSDM/ Kanban etc) have been increasingly gaining popularity to address this change. However, few myths and misconceptions tend to prevail on agile way of managing projects. In this article, we examine some of these myths and misconceptions and deliberate on how best to address them.

Myth 1: Agile is applicable primarily for Information Technology (IT) projects !

This is the most often quoted myth, undoubtedly stemming from the fact that Agile Manifesto came into being in 2001, because of the frustrations experienced while implementing IT projects using traditional 'waterfall' lifecycle model.

We should acknowledge that 'waterfall model' by itself is not a 'bad' approach to follow. However, in the context of IT projects, the 'User or the Client' becomes aware that the developer could technically keep modifying the end-product to address user's evolving or 'newly discovered' requirements. In many situations it becomes an eternal vicious cycle, with the rate of change in user requirements far exceeding the rate at which they could be addressed. Then it becomes difficult for the 'waterfall model' to cope up with this rate of change.

Application of a specific lifecycle approach also depends on the 'velocity' of the industry in which the project is getting executed. Certain industries like construction, lumbering etc have a change which is relatively slow moving – whereas in some verticals like telecommunications etc, frequent change requests are noted. And even for IT projects, there have been precursors to agile like 'Rapid Application Development', iterative implementation approaches etc., which tried to address this change aspect.

So agile is not something totally revolutionary concept by itself. But clarity is now emerging on the aspects of supporting environment (in terms of people skills, processes, collaborative techniques, change in culture etc.) which are required to make agile approach more effective.

How we can successfully use agile also depends on the ease of incremental delivery and its adoption by the user community. We have noted that service delivery projects (say, for insurance claims processing) have successfully used agile approach. Also business process reengineering projects for incremental infusion of redesigned processes have done well adopting agile. However, projects building products of a predictable nature, like developing a township, cannot be using agile naturally and waterfall model is more apt in this context.

Thus agile can instinctively be a good fit for projects where requirements keep changing and incremental delivery of outputs is feasible. Of course, these characteristics are inherent in IT projects – so agile works best here. However, even non-IT projects exhibiting the above features can use agile productively. For instance, agile has successfully been used for workflow tracking, call center management etc. 'Lean Startup' is yet another example where a variant of agile has been profitably used in the business context.

Myth 2: Agile requires the pertinent techniques to be followed rigorously

Using agile approach calls for more attitudinal change – rather than following a rigid set of tools and techniques. Creation of user stories, prioritizing the product backlog, conducting ‘standup’/planning meetings and production of ‘burn-down’ charts alone do not guarantee the success in an agile project. Change in management mindset to adopt the incremental delivery, rigorous scope prioritization, acceptance that all the scope features may not be delivered in a time-box and sticking to the time-box schedule are far more critical. Techniques alone do not guarantee success. In many companies, the structures and processes are designed to thwart any change. In such organizations, agile can become a ‘misfit’.

Myth 3- Agile projects are open to ‘unstructured’ development

Once a sprint is authorized, typically the implementation team is ‘locked’ and is focused on development of the required deliverables. It becomes the responsibility of the Product Owner to ensure that the team is not ‘disturbed’ during this sprint. This focus in fact facilitates a structured way of product development, rather than getting pulled in from multiple directions with the demands from diverse stakeholders and getting distracted in the middle of the sprint. Rigorous attention is paid to quality and testing as a part of release management. Agile is thus not a ‘code and fix’ or ‘laissez-faire’ development approach, but enforces discipline in development and release quality.

Myth 4 – Once we empower the teams, work in an Agile environment will go on smoothly by itself

Whereas empowering teams is one aspect of agile, the Product Owner should have authority and credibility to drive agile initiatives. Typically the Product Owner also may be accountable for the product roadmap and would be responsible for securing investments and resources for development. If the top management buy-in and commitment are lacking, empowering teams alone would not ensure success in an agile environment.

It is also important to develop the team members in cross-functional skills, so that they can double-up for each other in case of exigencies. Adoption of agile undoubtedly calls of breaking down of silos and greater co-operation amongst the team members, so that they become ‘self-led’. Connecting the silos to build dedicated and self-empowered teams can become a great challenge in orthodox organizations. However, empowering teams lead to job enrichment and greater motivation for the teams to strive for repeat success in delivery using agile techniques.

Myth 5- Agile is not in favor of documentation

In many projects, (especially in projects conducted in regulatory environments, such as finance and healthcare), well-structured end-user documentation is must. In agile projects, such documentation becomes yet another deliverable to be produced. If the team is stable across the product development lifecycle – the extent of project related documentation could be reduced. But since many of the projects are now executed with varying teams in different sprints, adequate documentation becomes inevitable. What agile is against is producing a ‘documentation for the document sake’. This aspect is more pertinent when the requirements keep rapidly changing.

A balance needs to be struck between detailed documentation and the extent of redundancy it can lead to due to change in requirements.

'Necessary and sufficient' principle adopted in disciplines like mathematics can be used, while determining the extent of documentation needed for smooth delivery in an agile context.

Myth 6- Agile teams jump into implementation with minimal planning

Another fallacy which gets extensively propagated. Agile teams spend considerable effort in planning, with complete team involvement. The same thing holds good for effort estimation and scheduling. One key difference here is since the sprints are relatively of short duration (say, for 4 weeks) – the extent of planning and execution spans what needs to be done for the current sprint and is relatively light as compared to what is being done in a waterfall model. So, what agile does is to avoid 'upfront' planning for the entire project duration where we usually have no full control. Agile adopts continuous and incremental planning, factoring in lessons on how previous sprints went through, by conducting retrospectives, in a 'rolling wave' fashion. 'Divide, focus and conquer' is the maxim which holds good here.

Myth 7 – Agile teams get full autonomy on what they do- which can lead to loss of control

Whereas it is true that agile teams are more empowered, as compared to say, the teams working in a traditional waterfall model, they need to work within the scope boundaries set by the sprint. There is an oversight from the Product Owner (and Project Manager in some cases), to ensure the work allocated within the sprint gets done by the defined schedules and budgets. So – agile is not a 'free for all' blue-sky approach to product development. Rather agile approach discourages micro-management and empowers teams to work with a goal oriented focus. Such a change in mind-set frees up the senior management to maintain control at the 'big picture' level, rather than losing the 'forest for the trees'.

Myth 8- Agile projects always deliver quick results

This proposition is generally true for smaller projects. For larger projects involving many functional departments, inculcation of the agile culture takes long time. Usually the pushback comes from the middle management, as these are typically the roles who would not like to move out of their comfort zones and try something new or innovative. 'We have always done this way' attitude can be detrimental.

Also in many companies, middle managers (and even some senior managers) are resistant to delegate authority and empower teams, as they fear loss of control. Agile encourages removal of impediments that cut across functional boundaries- whereas some middle managers would like to 'create' more impediments to retain control!

In this context, it takes time for the agile culture to get truly embedded in the organization. Larger projects in such organizations get into issues. Typically in corporate cultures which foster collaboration and achievement, agile becomes a 'natural fit'. And in organizations where people are not 'trusted' or self-motivated, agile approaches can fail. It needs to be recognized that it is more a problem with organizational culture rather than agile approach in such situations.

Myth 9 – Agile is the perfect antidote to all problems faced in traditional waterfall model !

Agile is not a 'silver bullet' which can solve all the issues faced during traditional development methods.

The tenets of joint application design with the Client and incremental release were established even in the waterfall model. What agile brings into prominence is its emphasis on timely delivery, prioritization of requirements, customer-centricity and cross-functional empowered teams, which focus on near-term deliverables and release.

Whereas the development teams working in a waterfall model discovered their failures much later (when the released products did not completely address the client's changed requirements), agile projects can learn quickly due to 'faster failures' and pondering over them during retrospectives for further improvements.

If the basic principles of agile are not followed rigorously, it can lead to even more chaos. What agile has clearly done is to shift the process mindset to people-centric approach and make them more productive by adapting the processes. This approach itself is easy to understand, but hard to master.

Of course, there are many more misconceptions associated with agile regarding its scalability, usability in distributed teams, capacity based development vs. feature centric development etc. We are not listing all of them.

There is a counterpoint is that agile is just a 'fad' and will 'wane' away. It is quite likely that with better tools for collaboration, variants of agile could continue to evolve. Agile approach itself can become agile! However, the fundamental tenets on which agile are based would endure. Embedding agile culture in a large scale organization is more of a journey rather than reaching a milestone!