

Agile Project Management vs Traditional – Questions raised

a) **Resourcing Agile Projects:** Agile teams can be strongly skilled for the production of some features on a project, but relatively weak for the production of others. It is difficult to generalise like this but one of the key "must have's" for Agile to work is that you are working with Subject Matter Experts in their field. There is no real opportunity to "ramp up" new recruits!

b) **Team Location:** Where Agile teams may be geographically dispersed, "osmotic" communication can be difficult and there may be insufficient communication within the team. I have not seen an Agile team working geographically dispersed as all the teams I have led have been on site and usually with the customer in frequent attendance. Although I can accept it is by today's technology quite possible. The key for the team is communication often and in short bursts, also to be able to problem solve as and when necessary. It is possible to set up the daily "stand-ups" via video link/telecom and the access to development code drops can be done daily by having say Visual Studio Team Server on a centrally accessed server.

c) **Agile Requirements:** Stakeholders influence in the production process is managed by the Product Owner/Sponsor. This can be a source of tension where there is not a systematic tool or process in place for deciding the order of delivery. In Agile the "order" of priority is decided through the "Backlog" and a discussion with the Product owner (often the customer sponsor) as to which "User Stories" are to be prioritised for each Iteration. Thus it is often in the use of the MoSCoW rules being applied that this becomes clear. However, having said that it is sometimes the case that the priority of any requirement (described in the User Stories) may change during an iteration in "discovering" hidden requirements and thus a decision is made as to what drops out of the iteration in order to include the new requirement. Thus the dropped requirement is considered for inclusion in a later Iteration, if still a priority.

d) **Risk, quality for Agile:** Even where features may be relatively complex, Agile does not support the production of management products such as a quality management strategy or risk plan. **Not true.** Agile project methods adhere to the "continuous improvement" principle of quality management. Adjustments for quality are made throughout every Iteration at key decision points and reviews. Although it is true that Agile methods "give up" the need for lots of documentation in favour of "working code", I have found that the quality management in software deliveries is covered by a rigorous testing process which often takes place over night in a separate shift in complex deliveries. Risks are indeed identified, analysed and responses planned and executed in an Agile project but becomes less documented in detailed process documents and more visible on flipcharts and wall visuals with owners of actions identified and is normally covered as part of the daily "stand up" reviews. Part of the reason SMEs are chosen for Agile projects is that they bring their experience and skill and therefore their passion for producing a high quality product.

e) **Meeting Business Benefits in Agile:** Agile plans do not include management of the expected benefits (or dis-benefits) of each feature. **Not true.** If you are delivering an Agile project your business benefits are covered in the Project Charter as for any project. The benefits (and dis-benefits) of each feature are discussed and outlined in the User Stories as a matter of course and form part of the prioritisation decision. Those features delivering highest benefit will often be prioritised over less beneficial ones. At each End of Iteration Review you would expect the question of the "project delivering expected benefits against the Business Case" to be reviewed and Go/No Go decisions made. This might also lead to changes in requirements and thus User Story priorities.

f) **Quality standards in Agile:** Even if there is close collaboration between developers and users, the lack of quality planning on Agile projects can cause confusion on quality standards. **Not at all.** In fact the developers do not usually "collaborate" with the users directly, that would inject confusion! The quality is maintained intact by the developers and testers being tasked to work in pairs in smaller projects or separately in separate shifts on larger projects (the tester basically being tasked to break the code of the developer to ensure quality

Agile Project Management vs Traditional – Questions raised

of feature sets!). The Product Owner manages the overall feature sets described in the User Stories and the User Experience SME will be tasked with ensuring all "User Stories" cover the Stakeholder (and thus User) requirements. The main collaboration with the customer is where the project management team discuss priorities and plan for next Iterations, and of course during the demonstration sessions held for progress reviews.

g) **Risk Management in Agile:** The lack of risk planning can expose Agile projects to unforeseen issues. Planning in these circumstances would probably have improved the delivery of the product. **Not true.** Agile requires planning and executes planning on a "rolling wave" approach. The first few days of the Agile project involves some very focused "bottom up" planning over the first two Iterations ahead and "top down" planning for those beyond. Every Iteration involves a parallel activity where planning for future Iterations take place alongside the production of code for the current Iteration. That is the beauty of the Agile method. Risk planning takes place during these sessions as part of the planning process.

h) **Lessons Learnt in Agile:** Agile production does not consider lessons learned from previous attempts at the production of features. **Not so,** the Agile approach is to work on continuous improvement as built into the approach. I always look for lessons learnt during each Iteration and thus there is the potential with Agile approach to make significant improvements from one iteration to the next as well as for the next project.

i) **Change Control in Agile:** Frequent, on-the-fly changes requested by the client are not all accounted on Agile projects. **Not so.** All changes in requirements are "weighed" by the Product Owner and Project Management team during each Iteration. This is the main strength of Agile methodology, it can and does respond to "on the fly" changes! The impact of that change of course may be to change the prioritisation of User Stories in each Iteration but that is acceptable with the usual project constraints.

j) **Documentation:** Agile produces no documentation to describe how the development was conducted. **Not true.** Usually Microsoft Agile teams make use of Microsoft Visual Studio Team Server. This allows for extensive reporting on development products and testing results. It is also not true to suggest that Agile produces no documentation; it produces *sufficient* to ensure requirements are known, iteration content is clear and the project "rhythm" is known by all (i.e. Communication, reporting, reviews, team stand ups and demonstrations) and of course the customers communication requirements (Conditions of Satisfaction) are also documented. The rest of the traditional documentation is covered by visual representation around the team room normally and is organised by the Scrum Master.

Further information on Microsoft Agile methods available from
Simon Robertson PMP®

+44 (0) 7967 300344

simon@robertsonconsulting.co.uk

+44 (0) 7967 300 344

PMP is a registered mark of the Project Management Institute, Inc.