WHITE PAPER

THE AGILE WATERFALL MIX – DELIVERING SUCCESSFUL PROGRAMS INVOLVING MULTIPLE ORGANIZATIONS

Amit Aggarwal
FIS Consulting Services

800.822.6758

Overview

The fintech explosion, the Internet of Things and the increased recognition that the ultimate differential is the customer experience are creating an exciting era of technology innovation within the financial sector that we haven't seen since the introduction of the Internet. Banks and financial institutions are seeking best-of-breed products and then finding themselves having to integrate across not only technology bases but integration and implementation methodologies. Existing mature applications must be integrated into the new, bringing together initiative-focused implementation teams with several supporting teams that often have competing priorities. These supporting teams usually reside in different geographic locations, and its members can work for different companies – yet all play critical implementation roles.

When deploying the Agile methodology for development in building and delivering innovative financial solutions, this backdrop causes financial institutions to reconsider how core teams and support personnel should collaborate effectively in order to succeed. This paper explores how programs should be structured and how all involved parties must be informed and committed to deliver program success.

Agile defined

An Agile methodology helps teams respond to changing requirements and unpredictability through incremental, iterative work cadences, known as sprints.[1] Agile development offers an incremental approach to software development, placing an emphasis on empowering people to collaborate and make team decisions involving continuous planning, continuous testing and continuous integration. Within Agile development the term “scrum” arises. Scrum is a subset of the Agile methodology and provides a lightweight process framework for the development team.
In Agile software development, many times a storyboard helps developers quickly understand what work needs completion. As long as the team keeps the storyboard up-to-date, anyone can see what work has been completed, who is working on which task and what work remains.

Proponents of Agile believe that software should not be developed like an automobile on an assembly line, in which each piece is added in sequential phases. In an assembly line process, every preceding phase of the project must be completed before the next phase can begin. Agile supporters often downplay traditional development approaches which dictate that developers:

- Gather all of a project's requirements
- Complete architecture and design documentation
- Develop the requisite code
- Test the solution
- Implement the solution

Agile enthusiasts believe this traditional approach often fails because of the lack of flexibility to change. Instead, Agile supporters created the *Agile Manifesto*, which describes elements that value:

- Individuals and interactions over waterfall processes
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

[1] Agile definition, agilemethodology.org web site, October 2015

Challenges of the Agile approach in a multifocused environment

The characteristics of the Agile approach to product development that make it an attractive collaborative methodology also create some challenges in large, complex initiatives. These require diverse sets of resources, not in the same location, often reporting to a completely different organization with competing priorities that have nothing to do with the program at hand. Yet their participation is critical to understanding systems and application programing interfaces (APIs), for understanding capabilities, for setup and configuring environments, and for test and production support during go-live.

The very nature of the term Agile implies extreme flexibility in a development approach. This works fine when the team is entirely devoted to the same objective. But when team members have other priorities, obtaining their time in an Agile manner becomes a daunting task.

Developers in an Agile methodology rely on dialogue over documentation. If they are not co-located and have competing priorities, they cannot walk over and ask questions of their peers; they must communicate by email or the phone. If an individual is not available at that time, this often leads to delays in the Agile program. If the team works with different supporting personnel, having to re-explain the need significantly deflates the momentum generated by the Agile methodology.

How to find a solution to enable multifocused teams to collaborate successfully?

To answer the above question, one must consider what has historically worked to deliver success across teams with multiple priorities. This involves reviewing a more traditional development approach and comparing it with Agile.

The traditional approach for developing new software is known as the Waterfall methodology. Like the name implies, events and milestones flow in a logical sequence. An initiative does not begin until its business requirements and overall blueprint are defined and documented. This approach is especially useful in large, complex initiatives in which participants are part-time team members and are not co-located. It provides defined roles for team members not necessarily dedicated to the initiative and working for different organizations and enterprises.

By defining requirements up front, each team that needs to participate knows what they will need to do and when. This allows for the members of the team to come in and out per the master plan, working on other initiatives in between while they are not needed. In these types of programs, all parties must work from one common set of design documents and specifications in order to eliminate confusion and misunderstanding. This one key factor is often lost via the Agile methodology.

The resulting challenges of the Agile approach with in a complex multiorganization team environment include:

- Dedicated teams can be difficult to staff, especially with knowledgeable business owners and product experts.
- Daily meetings and scrum sessions are challenging to manage unless all participants are located in the same space.
- Continuous changes to requirements and the subsequent code may frustrate nondedicated team members.
- Lack of documentation of process and results can leave gaps in regulatory and audit reporting.
- Collectively, these challenges could lead to delays and cost overruns.

The traditional approach for implementing new products at banks works through an evaluation process that defines and budgets for new system deployments. High-level requirements gain definition at this point in the deployment process. After that effort, the solution moves into an implementation phase. Often, when multiple parties are involved during the implementation, multiple visions of the final solution develop. This creates the need for one common set of business requirements and one solution design document – deliverables not supported in an Agile approach. But this unity of design and purpose results in the negation of flexibility within the development life cycle. In addition, a lot of the up-and-coming fintech providers deploy the Agile methodology for rapid development and deployment, which conflicts with the waterfall approach.

What, then, is the answer?

The proposed approach

In the onset of a complex and multilocation program, the leadership of the initiative must devote time to the following elements of the upfront definition:

- High-level business requirements
- High-level blueprint
- High-level, integrated planning

The program owner must ensure the level of detail in the above documentation doesn't impede the creativity and flexibility gained through the Agile approach. In setting up the program, one must:

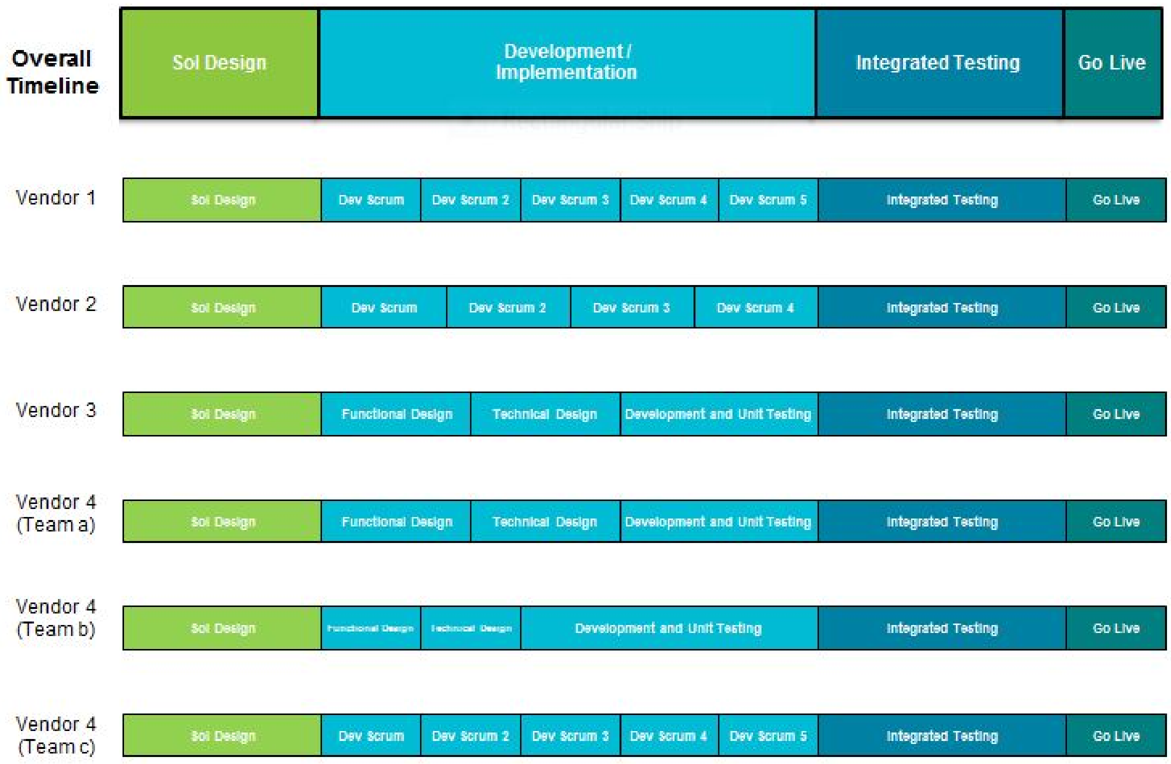
- Have a general overall plan
- Secure external resources via funding
  - Ask for a bucket of hours, but this will still cause delays if an individual is not available.
  - Obtain fixed hours per week.
  - Plan for the weeks that are peak workloads to ensure dedicated support.

Once the overall plan is determined, an overall timeline for the program becomes part of a joint discussion between key stakeholders and program participants. All parties need to agree on the high-level plan and timing of the milestones. Then, the solution architect creates a single cross-vendor solution design. This design documents the business scope and high-level requirements, identifies all solution components, assigns major responsibilities and defines the integration components.

Once this cross-vendor solution design is completed, individual plans are constructed for every third party (or vendor) in the program. These individual plans must then align to the overall timeline. In this planning process the aspects of the development methodologies are discussed, high level dependencies noted and the overall plan is adjusted as needed based on the needs of each participant.

The following graphic depicts the outcome of this planning and coordination, showing how the individual vendor timelines incorporate into the overall program effort.

Joint-Methodology Sample Timeline



The Solution Architect in action

Within the planning and execution of a large, complex program, the solution architect must address many needs and can provide the foundation for distinct development teams (following different methodologies) to coexist and thrive. Primary duties of this critical role include:

- Defining roles and responsibilities
- Understanding and documenting the current and future states of the existing and future solutions
- Creating the overall initiative scope and high-level requirements
- Conducting a gap analysis
- Consolidating estimates and the overall timeline
- Ensuring integrity of design and ultimate solution maintained throughout the program
- Conducting post-implementation meetings and identifying lessons learned

At the onset of the program, the solution architect's role is to understand high-level business needs and construct a blueprint and high-level plan for all parties to follow. In the process, the solution architect identifies the different development methodology and incorporates it into the plan as described in the previous section.

Once a program is in motion, the solution architect's knowledge of the business needs and what changes need to occur, along with a deep understanding of banking and related technology, allows the solution architect to provide the rapid input and feedback required by an Agile team as well as other teams that may be using other methodologies, such as Waterfall. As the different teams progress through their development cycle (Agile with a series of short development cycles and Waterfall with deeper requirement gathering and design), the solution architect can provide insight into the integration requirements of their component of the program, speaking to nuances of middleware configurations, API features and options while keeping an eye on the big picture. In this manner, the two methodologies can coexist while driving a singular focus and achieving a greater chance of success for the overall program.

Two different scenarios in the table below provide examples of how Agile and more traditional teams can interact within the same initiative. In the first example the solution architect provides knowledge and guidance to an Agile team operating under the governance of a Waterfall approach. In the second example the Agile methodology provides the governance, and the solution architect, quality assurance (QA) staff and other experts are drawn into the planned sprints of the Agile team, following the guidance of the Agile facilitator.

Development Team Methodology	Guidance to Development Team	Resources to Development Team
Waterfall	High-level solution design that feeds detailed requirements and technical design	• Solution architect provides vision and high level requirements. • QA, SMEs, etc. understand when to provide assistance.
Agile	High-level solution design that that feeds one common storyboard	• Solution architect provides vision and high-level requirements. • Solution architect primary knowledge resource for sprints and other iterative efforts • Solution architect the facilitator to communicate details across nondedicated resources, as needed • QA, SMEs, etc. understand when to provide assistance.

Throughout execution of all the projects that comprise the program, the solution architect serves as a focal point for team members of the initiative, regardless of which methodology is used.

Summary

With the proliferation of new technology in financial services, bank customers expect innovative new products. The development, implementation and integration of these new products may require coexistence and blending of Agile with a more traditional methodology. The solution architect can serve as the unifying force that can deliver the best of both worlds, contributing to successful launches of large and complex initiatives.

Contact us

For more information on our experience with Agile teams interacting with FIS solution architecture and design consultants, contact FIS Consulting Services at 800.822.6758 or visit www.fisglobal.com.