



White Paper

Agile Automated Combining Agile Methods With Agile Technology

July 2010

Jumar Solutions Limited
Jumar House, Pinewood Business Park
Coleshill Road
Solihull B37 7HG
United Kingdom

Tel: +44 121 788 4550
Fax: +44 121 788 4551
www.jumar-solutions.com

Table of Contents

1	Why Agile?	3
2	Eight Stages to Success	4
2.1	Budgeting.....	4
2.2	Project Initiation	4
2.3	Iterative Development (Sprints).....	5
2.4	Training.....	6
2.5	Production Launch	6
2.6	Tuning.....	6
2.7	Wrap-Up	6
2.8	Maintenance	6
3	Success is Agile Automated	7

1 Why Agile?

Rapid Application Development (RAD) first came to the attention of systems developers in the 1980s and early 1990s. James Martin, the man behind the original Information Engineering Methodology, had recognised some of the inherent problems with waterfall-type methods and determined that there was a need for an approach with less emphasis on up-front analysis and more on prototyping, iteration and getting results faster. This was RAD and it was the precursor for many subsequent development approaches such as *Scrum*, *Extreme Programming (XP)* and *Lean Development*.

In 2001 representatives of a number of software development approaches that could loosely be defined as coming under the RAD umbrella collaborated to publish the *Agile Manifesto*. This was a collection of principles and values which were to form the foundation of *Agile Software Development (ASD)*. ASD is not itself a method but a collection of methodologies and approaches which share these common principles. Radically different from traditional waterfall methods with its reduced emphasis on detailed requirements and its focus on timeboxing, iteration and regular reprioritisation, ASD was often viewed in the early years as unsuitable for 'serious' organisations or large projects. However, thanks to the ever-growing catalogue of successful Agile projects in large enterprises as well as small, ASD is no longer viewed with such suspicion and has become part of the mainstream.

As ASD has grown in popularity, software tool vendors have developed offerings to support it. The majority of these tools are method-oriented, addressing the management of the project life cycle and supporting the tasks and techniques which are part of it.

Jumar Solutions is an IT company that, amongst other things, carries out software development both in-house and onsite at customer premises. With several staff members who worked for James Martin Associates (and, post-acquisition, for Texas Instruments Software) in the 1980s and 1990s, Jumar has a rich heritage in software development methods and a strong desire to leverage the most efficient processes and techniques for the benefit of itself and its customers. Already believers in the benefits of ASD, Jumar researched the market for development tools to support Agile methods, specifically to support web application development and web services, and found the OutSystems Agile Platform.

The Agile Platform is a product which addresses the full life cycle of delivering and managing web business applications. The product includes the tools required to integrate, develop, deploy, manage and change web business applications. It is unique in the way that it automates not only the processes around Agile systems development but, through its model-based paradigm, the actual systems development itself.

Jumar has found the application of Agile methods to its software development processes, supported by the Agile Platform, to be a highly efficient and predictable way of delivering and maintaining business systems and since 2009 has partnered with OutSystems to promote this way of working.

This document describes the 8 stages of the approach that Jumar uses and how these stages are supported by the automation of the Agile Platform.

1. Budgeting
2. Project Initiation
3. Iterative Development
 - a. Sprint Planning
 - b. Development and Testing
 - c. Product Demo
4. Training
5. Production Launch
6. Tuning
7. Wrap-up
8. Maintenance

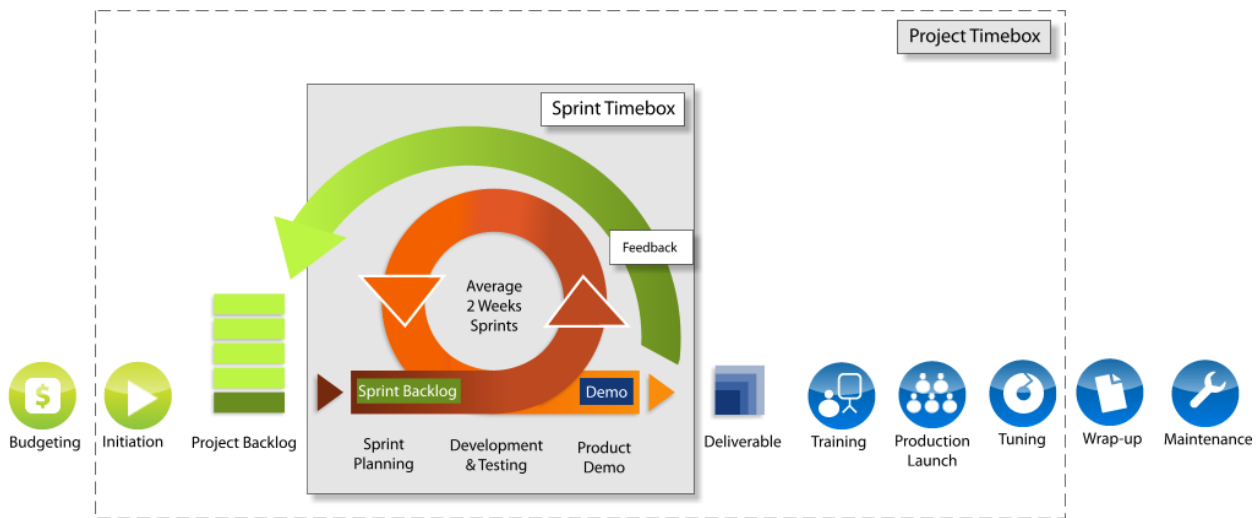


Figure 1: Stages of the iterative Agile approach

2 Eight Stages to Success

2.1 Budgeting

The most important concepts applied during the budgeting stage are *User Stories*, *Sizing* and the *Project Timebox*. User Stories define the project scope; they are high-level requirements defined in business terms. In the Agile approach these descriptions of what the application should do are not defined in the great detail that traditionally would have been demanded for sign-off. They are likely to be collected in days as opposed to the more traditional requirements gathering processes that can take weeks or even months.

Each User Story is decomposed into a set of high-level features, which are then sized to determine the effort required to deliver them. The effort or size of each feature is based on its planned pattern of implementation; for example, listing, sorting, searching, editing, etc. Once all features are sized, the Project Timebox can be defined. The timebox for the project is based on the set of User Stories in scope, the estimated time to complete each feature and the projected project team composition. The Project Timebox defines the overall budget for the project in terms of resources and number of days to deliver the defined functionality.

The timebox principle is the key to predictability. Once they have been set, the budget and the timeline are fixed and non-negotiable. This is true for the project and each sprint that makes up the project. To be truly Agile in the budgeting stage, we use the Agile Sizing tool. The Agile Sizing tool allows us to scope a project based on heuristics accumulated from over 600 Agile projects. This tool automates the sizing calculations based on patterns, as well as project and application parameters. It provides an independent estimate, free from the influences of individual agendas. When the sizing is complete, the Sizing tool creates an initial project plan.

2.2 Project Initiation

In the Project Initiation stage the User Stories and features are loaded into the Agile Projects component. This is called bootstrapping the project and creates the initial *Project Backlog*. The Project Backlog is the list of requirements to be delivered. Once the project is bootstrapped a kick-off meeting is held to communicate and resolve any issues with project scope, roles, responsibilities, risks, sprint demo session dates, etc. At this point the project scope may be analysed in more detail. Scope analysis involves a closer look at requirements and external interfaces to establish dependencies. Scope analysis also involves prioritizing the requirements based on the value that they will provide to the business.

Typical questions asked at this stage are: “What are the most critical business user profiles?”, “What are the most frequent User Stories or use cases that involve those profiles?”, “What are the use cases and user profiles that contribute most to the application’s ROI?” The result is a Project Backlog containing a prioritised list of requirements. As part of the project bootstrapping process, the project sprints are defined in the Agile Projects

component. A sprint is an iterative cycle usually lasting 2 or 3 weeks in which a portion of the application is developed. At the end of each sprint the project team delivers a working, vertically integrated portion of the final application. The length of a sprint is defined in the original plan provided through the Sizing tool. Throughout the remainder of the project the Agile Projects component is the primary tool used for managing the day-to-day tasks of the project. This component is actually composed of various tools to facilitate project initiation, planning, execution, controlling, reporting and closure. It includes specific tools for managing Agile projects like backlog management, sprint planning and requirements management.

2.3 Iterative Development (Sprints)

Sprint planning is the first major activity of the Iterative Development stage. The sprint planning activity involves reviewing the Project Backlog and collaborating with the business team members to decide which user requirements will be included in the current sprint. This process is called Sprint Backlog Settlement. During this process, feature negotiation and prioritization takes place between the Engagement Manager and the Business Manager to establish priorities for the sprint and balance the overall project timebox. The goal of Sprint Backlog Settlement is to identify the high-value requirements to be included in the sprint's backlog without exceeding the sprint's timebox.

Once the sprint backlog is defined, a detailed analysis of the requirements is conducted. By further detailing the requirements to define their design and implementation approach, it may become evident that some may be impossible to implement or alternative implementations may need to be identified. The impact of these changes will be assessed to determine if these requirements need to be shifted to the next sprint or lesser value requirements be removed from the sprint backlog. Remember, the timebox is deemed unmoveable for each sprint and for the overall project. All through the sprint planning activity, the Agile Projects component is used to record the sprint backlog, change requests, issues and risks that arise because of the continuous planning process.

The Development and Testing activity begins in each sprint once the sprint backlog is settled. At the start of this activity, the Delivery Manager and each member of the delivery team works together to detail the design, identify the tasks, make delivery commitments and finalise the expected effort for each of the assigned work items. As the team progresses in developing and testing the features, they do so using the visual design and development approach provided by the Agile Platform's core components – Integration Studio, Service Studio and Service Center. On a daily basis, the Delivery Manager conducts a 15-minute standing-only Scrum meeting to synchronize the team's direction. At least once a day (often more frequently) all team members upload their work to the Agile Platform and integrate with the other team member's work, ensuring that all work completed is made available to the rest of the team. The project's Delivery Manager uses the Agile Projects component to manage the team's workload. Similarly, each team member uses the Agile Projects component to record completed effort against each work item on his or her task list.

At the end of each Sprint, the delivery team provides a demonstration of the working application to the key business users. During each demo session, the end users and QA team members will provide feedback using the Agile Platform's Embedded Change Technology (ECT) so that all input goes directly to the Agile Projects component for immediate assessment and possible addition to the project's backlog. Beginning with the initial sprint, the formal demo signals the start of the *Continuous User Acceptance* process. Because sprints are short and frequent, the users and Quality Assurance engineers will be continuously testing the delivered application functionality and identifying issues, enhancements and functional errors. This shift in testing greatly enhances application quality and user adoption.

Once the demo is completed, the set of features accepted by the Business Manager are signed off and closed. Any changes or new requirements are added into the Project Backlog and discussed during the next sprint planning session. Whether or not these are incorporated into the next sprint backlog is discussed during feature negotiations. It should be noted that new requirements or changes of higher priority will inevitably displace lower priority items that are in-scope. This is to be expected and is actually one of the benefits of the Agile approach. By constantly aligning with the business, we can be assured that the application delivered meets the needs of the business.

2.4 Training

The training stage is where ownership of the application is passed to the business. In an Agile training programme the focus is not only on making sure that users learn how to use the application but also on obtaining extended feedback.

Key business users and front-line individuals are trained to a level at which they can assist all other users in learning to use the application. If the application is well designed and intuitive, users should be able to learn the application with minimal to no formal training.

The same key individuals will receive and provide feedback so that the application can continue to evolve as the business evolves. Feedback is encouraged through the Embedded Change Technology (ECT) mechanism which is automatically built into every Agile Platform application. Using ECT all user feedback will automatically be logged and fed back for evaluation and, if appropriate, adding into the Project Backlog.

2.5 Production Launch

During production launch the application is published to production using the Agile Platform's 1-click publishing technology. Depending on the nature of the application, the production launch might be targeted to a limited set of users and be running in parallel with any replacement systems. As soon as the application is in production, the delivery team begins tuning the application. This stage is critical in preparation for full application rollout and end-user adoption.

2.6 Tuning

The tuning stage is key to project success. During this 'special' sprint, two key activities are conducted – application performance tuning and functional tuning. During functional tuning outstanding issues are resolved and any remaining low-priority Project Backlog items are assessed for delivery. These items are assessed to determine if they can be considered as 'adoption boosting' functionality. Adoption boosting features increase the probability of the users liking and immediately using the new application. From the perspective of the delivery team this stage can be very intense. There will typically be multiple production deployments each day as the team conducts functional tuning.

While the developers are busy pushing new versions to production, the application, system and network administrators are conducting performance tuning to ensure that the application and the environment meet or exceed specified service levels. In performing the tuning tasks, the delivery team continuously uses the Agile Platform delivery-related components – Integration Studio, Service Studio and Service Center – as appropriate.

2.7 Wrap-Up

The wrap-up stage officially closes the project (or at least this release of the project). During wrap-up the tuned application is fully signed-off in production. Using the Agile Projects component all remaining Project Backlog items are discussed and either archived, prepared for the next release of the application, or identified as items to be actioned during the maintenance stage.

2.8 Maintenance

The objective of the maintenance stage is to ensure that the application continuously supports the business through evolutionary maintenance. This means that as the business changes, so should the application. Therefore, this stage consists of a series of 1 to 2 week sprints depending on the changes or new features identified on an ongoing basis. These sprints, although they may not be contiguous, continue to follow all the primary activities in a regular sprint and leverage all the necessary tools required to keep the delivery team Agile.



3 Success is Agile Automated

OutSystems define project success as being on time, on budget and having 100% user adoption. Jumar fully endorses this but would go further and say that real excellence has only been achieved if the same set of requirements could not have been implemented with the same quality any quicker or any cheaper. To make this kind of claim it is necessary to automate to the maximum extent possible; to use a combination of efficient development processes and efficient enabling technology.

The Agile approach described above, supported by the Agile Platform, delivers that winning combination of efficient development processes and world-beating automation. Together they can shorten delivery cycles while improving project predictability, responsiveness to business change and overall development productivity.

Further Information

For more information on Jumar Solutions and the OutSystems Agile Platform please follow these links:

	http://www.jumar-solutions.com
	http://www.outsystems.com