Effective Release Management in Agile Scrum methodology

Submitted to

The Project Management Leadership Conference 2006
QAI India Pvt. Ltd.

Authors:
A Narasimhan – a.narasimhan@siemens.com
Niladri Sankar Ray – Niladri.Ray@siemens.com
M Sreenivasulu – M.Sreenivasulu@siemens.com

Siemens Information Systems Ltd.
Healthcare Products and Solutions group
48, Keonics Electronics City
Hosur road
Bangalore – 560 100
Effective Release Management in Agile Scrum methodology

Abstract

This paper has been broadly divided into nine sections. The first section gives a quick overview of the Agile Scrum process. In this section, the common problems of a traditional development method, especially for release management of a complex product are brought out and the advantages of Agile Scrum method is highlighted. The process of Scrum has been explained in brief giving the essential details such as the artifacts, roles involved, the development process and the important meetings. A diagrammatic representation of agile scrum process flow is given for quick understanding.

The second section talks about the project context. This gives the details of the organization, the development team structure, the project complexity etc and puts the reader in the context of the project where the release management practices were incorporated.

The next section gives an overview of the essential ingredients of managing a product release in Agile Scrum context. Bare minimum required activities involved in a complex product release management process are listed in this section.

The fourth section gives the specific technique adopted for applying Agile Scrum for our project context. The scaling of the Scrum process is explained in this section.

The importance of project management in Agile Scrum has been briefly brought out in the fifth section.

The sixth and seventh sections give details about the organization of scrum levels and typical agenda items for the meetings at different scrum levels. The detailed responsibilities for various roles involved at these levels are also given as a table in this section. The last two sections list the best practices followed at SISL and also some of the lessons learnt during the transition process to Agile.

Introduction

Effective release management has always been a challenge in traditional project management methodologies. Committing a release date for a product and meeting it is a Herculean task, as anyone experienced in that would know. When it comes to enterprise products which are huge and involve multiple distributed development teams, it is all the more crucial and most of the project management effort is directed towards making a release within the committed time. This means the customer should be involved throughout the development cycle, not at the fag end of the development cycle as followed in traditional development methods.

When the software is developed through traditional development life cycle, because of the longer release cycle, customer needs to change the business logic or user interface is quite common. The delivery of the software gets delayed because of all these feedback incorporation. Hence the traditional, contract-driven development projects are normally opposed to accepting changes, especially towards the end.

The continuous learning in traditional method is lacking because of sequential development phases where customer feedback doesn’t come from the very beginning. In general, the Project Manager drives the schedule and team doesn’t have much empowerment. Also, there is a tendency of team’s focus getting diluted during the early stages of development due to longer release cycles.

In today’s world, customers demand for quality solutions in the shortest possible time and Agile Scrum helps to achieve this efficiently. In a small to medium sized project, release management is not much complex but to manage a large enterprise solution effective release management plays a crucial role. Though Agile Scrum is scalable, it doesn’t address all the release management aspects for a large project.
This article talks about what is Agile Scrum, how Agile Scrum methodology defines the approach for software project management and how SISL has adopted it successfully by defining all the release management aspects. The focus of this article is on the release management aspects in particular and highlighting the importance of project management.

**Agile Scrum – A quick overview**

The very first statement of Ken Schwaber in his book, ‘Agile Project Management with Scrum’, is ‘I offer you Scrum, a most perplexing and paradoxical process for managing complex projects’. Scrum is indeed perplexing and paradoxical. Propounded by Ken Schwaber and Jeff Sutherland, Scrum looks like a very simple process, but it is a challenging task to implement Scrum in its entirety. Teams are highly empowered and will not be told what to do or how to do tasks. They are their own masters and are free to pick items from a list of features, which are prioritized by the Product owner and start implementing them.

Agile Scrum is software management methodology. It does not interfere with the existing engineering practices. It defines the approach for software project management. It is more suitable to use for complex projects in which requirements are ambiguous and constantly changing.

Some of the salient features of Agile Scrum are:

- Team is highly empowered. Sprint success responsibility lies with the Scrum team, not with an individual.
- Customer involvement is more - Customer owns and prioritizes the product features and also accepts/rejects the product increment at the end of a sprint and hence is involved right through the development process.
- Effective and early feedback mechanism through sprint review and retrospection meetings helps the team to learn quickly and continuously.
- Allows effective management of unknown or changing product requirements.
- Using the sprint backlog, focus can be given to the most needed requirements.
- Effective communication through small sized teams.
- Build and release products in 30-day cycles called sprints while maintaining quality.
- Intelligent and Active Management
- Involvement of Product Analysts, Designers, Developers, Testers, DB team members in a team

Diagrammatic representation of the Agile Scrum process flow is as shown below (Fig 1).

![Fig 1: Agile Scrum Process flow diagram](image-url)
The below figure (Fig 2) talks about the essential aspects of Agile Scrum – roles, artifacts, key meetings and the development Process. All the relevant terms used have been explained under the Appendix section.

### Process- SCRUM Development Process

<table>
<thead>
<tr>
<th>Roles</th>
<th>Key Artifacts</th>
<th>Key Meetings</th>
<th>Development Process</th>
</tr>
</thead>
</table>
| Product Owner | Product Backlog:  
- List of requirements  
- Owned by Product Owner  
- Anybody can add to it  
- Only Product Owner prioritizes | Sprint Planning Meeting:  
- Hosted by ScrumMaster; 3-1 day  
- In: Product Backlog, existing product, business & technology conditions  
1. Select highest priority items in Product Backlog; declare Sprint Goal  
2. Team turns selected items into Sprint Backlog  
- Out: Sprint Goal, Sprint Backlog | Sprint Planning Meeting  
- 30 days each |
| ScrumMaster  | Sprint Goal:  
- One-sentence summary  
- Declared by Product Owner  
- Accepted by team | Daily Scrum:  
- Hosted by ScrumMaster  
- Attended by all, but Stakeholders don’t speak  
- Same time every day  
- Answer: 1) What did you do yesterday? 2) What will you do today? 3) What is in your way?  
- Team updates Sprint Backlog; ScrumMaster updates Blocks List | Daily Scrum |
| Scrum Team   | Sprint Backlog:  
- List of tasks  
- Owned by team  
- Only team modifies it | | |
| Stakeholders | Blocks List:  
- List of blocks & unmade decisions  
- Owned by ScrumMaster  
- Updated daily | | |
| Increment    | Increment:  
- Version of the product  
- Shippable functionality (tested, documented, etc.) | Sprint Review Meeting:  
- Hosted by ScrumMaster  
- Attended by all  
- Informal, 4-hour, informational  
- Team demos Increment  
- All discuss  
- Hold retrospective  
- Announce next Sprint Planning Meeting | Increment |

Fig 2: Agile Scrum process roles, artifacts, key meetings and development process

The Fig 3 shows the difference between the traditional Waterfall method of development and the Agile Scrum methodology. In the Waterfall method, the various phases are mostly linear in fashion while in the Agile Scrum; they are iterative as can be seen from the figure.

Fig 3: Difference between waterfall and Agile Scrum methods
Project Context

The current article talks about the implementation of Agile Scrum with specific focus on effective product release management practiced in Siemens Information Systems Ltd (SISL), Bangalore. SISL has a team of over 450 professionals working on a global enterprise healthcare product for Siemens MED, US. The product, Soarian Clinicals along with the departmental modules for Cardiology, Oncology and Operating Room Management, has installations worldwide and the recent version has been released successfully in the month of June 06. The product handles over 1000 batch transactions per minute, is scalable to 10,000 concurrent users and has the best clinical functionality in the world. The product owners are located in the US and development teams are distributed across Bangalore, Sweden and the US. The entire team has been transitioned to Agile Scrum and using Scrum for more than a year.

SISL had always been using the traditional methods of development, mostly waterfall model and it was a big challenge to move to Agile Scrum. The transition was done in phases for various modules of the product and the entire team was trained on Agile Scrum. Today SISL has over 100 certified Scrum Masters and has also been successfully operating at CMMI Level 3 with Agile QMS.

Managing a release using Agile Scrum

Release management in an agile world is more challenging as has been highlighted in earlier sections. Release management includes project management aspects in an agile scenario and specific release management activities.

A release has several well-defined activities in a product life cycle. It starts with planning the scope, requirements elicitation (scope could evolve from this too) and ends with live run at the customer site. Not all activities in this can be run as sprints in a large enterprise product environment, that too involving large distributed team. We at SISL found that it is better to use the scalability feature of Scrum and define the activities at different levels. A typical set of activities (depends again on life cycle, product size etc.) which was needed for our product is listed below:

- Scope planning
- Requirements analysis and creation of product backlog
- Architecture definition and framework development
- High-level design
- Creation of release backlog
- Development sprints
- Database activities (new database or ported database creation)
- Data porting
- Workflow testing
- Systems testing
- Bug fixing sprints in parallel with testing
- Beta activities
- Customer training
- Any customization at site
- Testing with customer database
- Parallel runs
- Live run and support

This is not an exhaustive list but has several important activities to be done in a typical release.
Scaling the Scrum process for Release Management

Ideally, any sprint should have some feature development. But in large product development work and large teams, it is practically not possible to implement this way. There are many activities in a release that might have to be done outside of sprints. The main development activity, which includes detailed design, coding, unit testing, code reviews, integration, sprint acceptance testing etc are always done within the sprints. Also, the bug fixing activity can be done in sprints. Other non-feature-developmental activities such as architecture definition, requirements analysis, high level design etc. need to be planned, executed and monitored by teams outside the main development scrum teams.

Scrum is scalable to several levels using the structure of Scrum-of-Scrums (SOS or S2), Scrum-of-Scrum-of-Scrums (SOSOS or S3) and so on. Using this structure, we defined processes and team structure to handle all the release activities along with the sprints. The activities were separated into sprint level (S1), module level (S2) and release level (S3) responsibilities.

Importance of Project Management in Agile

Every project needs a leader. Agile methodologies free up the project manager from the drudgery of being a task master thereby enabling the project manager to focus on being a leader – someone who keeps the spotlight on the vision, who inspires the team, who promotes teamwork and collaboration, who champions the project and removes obstacles to progress. Rather than being an operational controller, Agile expects the project managers to be an adaptive leader.

The best project managers are not just organizers – they combine business vision, communication skills, soft management skills and technical savvy with the ability to plan, coordinate and execute. So they are not just managers, they are the leaders. While this has always been the case, agile project management places a higher premium on the leadership skills than ever before. It has been proven that strong management is absolutely critical to the successful adoption and application of agile methodologies. It has also been observed that many established project management practices still apply to agile development projects with some adaptation and a strong dose of leadership.

The basic phases of an Agile development project are really no different from those of any other project. It is still required to define and initiate the project, plan for the project, execute the plan, and monitor and control the results. But, the ways in which these steps are accomplished are different.

In Agile Scrum, the Scrum Master actually plays the role of a project manager to some extent at S1 level. For a large project, the project management activities at the S2 and S3 levels gain a lot of importance for effective release management. Keeping this in mind, we at SISL have defined the responsibilities of a project manager as shown in Fig 5.

Organization of Scrum teams

The recommended team size for a scrum team is 6-8 members while our total team size exceeded 450. This team of 450+ includes managers, architects, designers, developers, testers, DB team, quality team and technical writers. So the first task was to form the right scrum teams for development with the right mix of all these skills. Then the second layer (SOS) to manage the inter-module interfaces, inter Scrum team dependencies etc had to be formed. The third layer (SOSOS) to manage the release level activities was formed.
The broad structure and roles/responsibilities of persons in all these levels and teams is depicted in Fig 4 and Fig 5 below:

Fig 4: Agile Scrum scaled to three levels with participants list

Now that we have defined the Scrum levels and the participants at each level, we need to define the responsibilities for the participants at these levels. This was a bit tricky as the existing traditional structure was strongly in place. Considering various roles and based on the different activities at all levels of Scrum, we defined the responsibilities as explained in Fig 5. This clear definition of responsibilities has really helped in increasing the effectiveness of the release.
<table>
<thead>
<tr>
<th>Scrum Levels</th>
<th>S1- SCRUM Teams (Features)</th>
<th>S2- SCRUM of SCRUM (Module/Component)</th>
<th>S3- SCRUM of SCRUM of SCRUM (Release)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Owner</td>
<td>Creates and prioritizes the product backlog Defines sprint goal(s) Participates in sprint planning and review meetings Accepts / rejects sprint output</td>
<td>Participates in S2 Meeting Resolves S2 level issues/risks</td>
<td>Participates in S3 Meeting Escalates unresolved issues/risks to S3</td>
</tr>
<tr>
<td>Scrum Team</td>
<td>Self-organized cross-functional team Accepts and commits to achieving sprint goal(s) Works to complete sprint backlog items to achieve goal(s) Participates in sprint planning, daily scrum and sprint review meetings Communicates issues/risks to scrum master Responsible for demonstrating sprint output</td>
<td>Not Applicable</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Project Manager</td>
<td>Participates in sprint planning, daily scrum (optional), and sprint review meetings Infrastructure issues - Hardware, Environment, etc. Promotes teamwork and collaboration</td>
<td>Resource planning/optimization to form multiple scrum teams Coordination of sprint planning/review meetings Identify, document and track dependency matrix across Scrum teams / Modules Coordination to resolve module level technical issues S2 level Issues/Risks management Participates in S2 meeting Merge and Build related Plan, if applicable Monitor the trend of burn down charts and advise corrective actions accordingly Corrective actions based on the retrospection/lessons learned from previous sprints for further improvement</td>
<td>Participates in release level planning Creates project plan Ensures execution of project as per the plan Project control to take corrective action, if required Participates in S3 level status meeting S3 level Issues/Risks management Release level status tracking and reporting</td>
</tr>
</tbody>
</table>

Fig 5: Roles and responsibilities at various levels of Scrum

**Typical agenda for S2 and S3 meetings**

In order to make the S2 and S3 meetings more effective, we also standardized the agenda items for discussion in these meetings. A list of typical agenda items for these meetings are given below which would help an organization to be effective right from the start while transitioning to Agile Scrum.

**Agenda for an S3 meeting**

- Update on status of various releases
- System test readiness and environment requirement
- Decision on Beta release to customer
• Working on securing beta resources
• Setting up the data by DB team
• Impact of release schedule change (if any) on development
• Any training requirements for team
• Update on Beta Mgt - involvement in sprints / priority
• Resources availability issues if any
• Status of any external dependencies across releases
• Collecting customer details for Beta management such as Current product version with customer, OS, environments, etc.
• Database schema code freeze status
• Status of porting of data for various releases
• Discuss open issues/risk items for each release
• Release backlog formation/changes
• Architectural/design issues

**Agenda for an S2 meeting**

• Open Technical issues across scrum teams
• Status of interface points across scrum teams
• Any open inter-module dependency related issues
• Any requirements/issues with Environment being used by the scrum teams
• Planning for next sprint
• Any resource issues in any of the Scrum teams
• Specific expertise (like architects, DB team) required by any Scrum team
• Any integration issues in the sprint
• Any issues with commitment of shared resources
• Any points to be escalated to S3
• Decision on database to be used, data porting etc. taken from S3
• Any build related issues
• Issues/risks in any Scrum team impacting other Scrum teams
• Lessons learnt from earlier sprint

**Lessons Learnt**

An exhaustive list of lessons learnt was captured through various retrospective meetings at various levels and some selected lessons which helped us specifically towards making an effective release management are listed below.

• Resource sharing across multiple releases/scrum teams should be avoided to the maximum extent.
• Number of parallel releases controlled in order to reduce resource sharing and thereby increasing the effectiveness of a release.
• Stable build should be available at least by middle of the sprint.
• Well-defined build process is required and it should not be changed frequently.
• Actual effort should be captured irrespective of methodology used.
• Include effort required for Builds, setting up environment, sprint review preparation, etc.
• Well-defined scrum process was not available before start of the project; instead it is evolved during the project.
• There should be clear plan about which testing is part of sprint and which is outside of the sprint.
Best Practices

Agile being new to the organization, there was a continuous focus on process improvement and identification of best practices. Again, some specific best practices relevant to this paper are listed below.

- S2 and S3 teams are very essential to release management when large teams are involved in development and product is complex.
- Scaling the Scrum team structure and clear definition of roles and responsibilities at various levels of Scrums.
- Schedule periodic meetings for S2 and S3 levels – preferably once a week in order to track and monitor the release level activities and resolve any inter-module dependency issues.
- Setting the typical agenda for S2 and S3 level meetings so that it becomes easier to conduct focused meetings towards effective release management.
- Availability of Scrum Server and Laptop for each scrum team resulted higher productivity
- Daily build helped the scrum teams to identify issues immediately and fix the same
- Collaboration with Testing and Product Analyst teams, as part of scrum team, helped in understanding requirements quickly.

References

1. Agile Project Management with Scrum – Ken Schwaber
3. Agile Software Development with Scrum – Ken Schwaber, Mike Beedle

Authors’ Biography

A Narasimhan, Chief Consultant, SISL – HPS, Bangalore
a.narasimhan@siemens.com

Academic Qualifications: B.E., PGDIM (IISc), PMP (PMI-USA), and Certified Scrum Master

24 years of total experience in the IT Industry.

Worked as Development Manager for Soarian MedSuite product and PMO-Head for DC6 product line. Soarian MedSuite is a complete healthcare product for the international market from Siemens Med USA. It is an internationalized product with over 20 modules. DC6 has three departmental products for the Cardiology, Oncology and Operating Room Management. Also conducted several training programs internally on Project Management, Agile and SCRUM Master training areas.

Niladri Sankar Ray, Chief Manager, SISL – HPS, Bangalore
Niladri.ray@siemens.com

Academic Qualifications: M.Sc., MBA, Ph.D in Management (USA) and Certified Agile SCRUM Master.

16+ years of total experience in the IT Industry including Analysis, Design, Development and Project Management experience of application and tools development related to various domains such as Health Care and Telecom involving variety of technologies and platforms. Experience on implementing Agile-SCRUM methodology and CMM/CMMI process.

Currently working as Development Manager for Healthcare Product - Soarian Clinicals.
Sreenivasulu M, Senior Manager, SISL – HPS, Bangalore
m.sreenivasulu@siemens.com

Academic Qualifications: Masters in Computer Science, Certified Project Mgmt Professional from PMI, USA, and Certified Agile Scrum Master

16+ years of total experience including Analysis, Design, Development and Project Management experience of machine control and stand-alone applications related to various domains such as Healthcare, Semiconductor, Telecom involving variety of technologies and platforms.

Experience in implementation of Agile-SCRUM methodology, CMM/CMMi process compliance.

Currently working as Project Manager/Release Manager for Healthcare Product - Soarian Clinicals. Also conducted several training programs internally on Project Management, Agile-SCRUM.

APPENDIX

Definitions

Scrum – An Agile method for project management, which is an iterative, incremental process for developing any product or managing any work. It produces a potentially shippable set of functionality at the end of every sprint.

Sprint – The scrum version of iteration. A sprint has a specific beginning and end. The beginning is the planning meeting and the end is the sprint review. A sprint is usually 30 calendar days long. During the sprint-planning meeting, the scrum defines the sprint goal that drives all activities within the sprint.

Scrum Team - The scrum team is expected to be 100% dedicated to the sprint. The scrum team is expected to be composed of all members needed to execute the activities required to achieve sprint goals.

Daily Scrum (S1) Meeting - A fifteen-minute daily meeting during which the Scrum team discusses the activity of the past day and the planned activity for the next day. This meeting primarily helps to ensure that the team is always focused on the sprint goal and any obstacles are identified and cleared. There is a set of standard questions each member of the team answers. NO PROBLEM SOLVING OCCURS DURING THIS MEETING.

Scrum of Scrum (S2) Meeting –This meeting helps to ensure that the Scrum teams are collaborating properly and all interactions and dependencies are tracked and managed. Scrum of scrums is a forum where issues requiring escalation outside the Scrum are raised. Could possibly meet as often as twice a week if there is a need.

Scrum of Scrum of Scrum (S3) Meeting –This meeting looks at the release level issues, ensuring that the release teams are collaborating properly and that all interactions and inter-dependencies are tracked and managed.

Product Backlog – Product Backlog consists of all work that can be foreseen for a product. Backlog consists of product features, functionality, infrastructure, architecture, and technology work.

Release Backlog – the portion of the product backlog being selected for a specific release.
**Sprint Backlog** – the list of tasks created by a Scrum team to achieve a specific sprint goal. The sprint backlog is first created during the sprint-planning meeting and is modified throughout the sprint by the Scrum.

**Product Owner** – The person who owns and prioritizes the product backlog.

**Scrum Master** – The person responsible and accountable for 1 – building a Scrum team based on the features intended for the team, 2 – facilitating the sprint planning, sprint review, and daily scrum meetings, 3 – ensuring the proper execution of the Scrum process, and 4 – identifying and clearing obstacles from the path of the Scrum team. He or she is charged with the success of the sprint.