# SCRUM: AN AGILE PROCESS

**K Selvi[1], Rana Majumdar[2]**

*Amity University, kselvi@nic.in, rmajumdar@amity.edu*

## Abstract

*Developing flexible, robust and scalable application is the purpose. Current IT industry is growing in the fast pace and have to survive in the competitive world by providing quality and time to market software. In order to sustain in the competitive world, requirement changes very frequently Agile methodologies today provide mechanism to handle growing expectations of customers with sufficient nimbleness to respond to changing market needs. Software development companies with its focus on innovation and value-centricity has adopted and matured its software development and maintenance methodology around agile development techniques, namely, Extreme Programming, Feature driven development and SCRUM. The purpose of this paper is to briefly introduce best suitable agile process like Scrum and Extreme Programming for different kinds of projects depending the attributes.*

-------------------------------------------------------------------***-------------------------------------------------------------------

## 1. INTRODUCTION

Traditional software development methods strict to all the phases of the SDLC cycle and also baselines the documents in every phase which are not that much important to meet the customer requirements and documentation

IT companies today have to overcome several challenges on behalf of these organizations:

- End customers of such organization today have growing expectations around product, services and customer experience adding to **complexity of software development** and management
- Customer need products and/or services yesterday and expect consistent change around such product and services, hence there is significant **time to market** challenge.
- Enterprise today have to demonstrate continuous innovation with new product and services, hence they suffer from **unclear and evolving customer requirement**
- Enterprise need **agility** today to respond to changing market needs, IT organization's today have to be prepared to handle portfolio of applications, realign initiatives, shelves an existing initiative etc.
- With growing cost models, IT outsourcing companies have to provide a way to demonstrate **value centricity and innovation** on customer engagements.

Agile methodologies like Extreme Programming (XP) and Scrum introduce a new approach to manage development processes in a flexible fashion. Agile methodologies differ from traditional way of development like it promotes communication, self-organization, flexibility and innovation instead of extensive planning and standard process.

## 2. AGILE METHODOLOGIES

**Delivering IT Product consistently at rapid pace**
Traditional methodology focuses on baseline requirement specifications and its implementation in distinct phases, resulting in limited customer involvement in development life cycle and high cost of change. Agile overcome most of the challenges by logically dividing the requirement (and devise their specification) as per the customer and market need and continuous seek the feedback resulting in improved product life cycle.

Agile methodologies (SCRUM, Extreme Programming etc.) are best suited for customer centric (or end user centric) application sustenance and extension. Agile promotes a learning driven, short interval, co-teamed and automated software development. Agile focuses on delivering continuous build to production in short iterations or sprints of requirement through implementation, resulting in a customer feedback every 3-5 weeks and better alignment to customer expectations and market needs. Agile also promotes automation in software development thereby improving time to market of product and services.

**Table: 1** Comparison between Traditional and Agile methodology

| Traditional | Agile (Scrum & XP) |
|---|---|
| Plan driven | Feature driven |
| Release frequency, commonly 6 to 12 months | Release frequency, commonly 1 to 2 months |
| Frozen requirements | Just-in requirements |
| Develop project in phases based on process | Develop project with less documentation |
| Integration at the end of the code development | Continues integration of the features |
| Separate testing phase | Continues testing |
| High cost of change | Low cost of change |
| More client involvement | Less client involvement |

## 3. SCRUM METHODOLOGY

**Scrum Definition**

Scrum is a process for a project team which is based on collaborative approach with incremental and iterative development which always focuses to deliver a business value.

**An Adaptation Iterative/Incremental Development**

SCRUM is an adaptation of incremental and iterative development technique; it deploys an empirical development model where a system is build as series of incremental steps called as Sprints. Usually sprints are development increments of 2-5 weeks duration, involving all the steps of software development life cycle – requirement to implementation.

SCRUM initially plan the context and broad deliverable definition, and then evolve the deliverable during the project based on the environment. SCRUM acknowledges that the underlying development processes are incompletely defined and uses control mechanisms to improve flexibility.

**Characteristics of SCRUM methodology are:**

- The first and last phases (Planning and Closure) consist of defined processes, where all processes, inputs and outputs are well defined. The flow is linear, with some iteration in the planning phase.
- The Sprint phase is a practical process. Many of the processes in the sprint phase are unknown. It is treated as a black box that requires external controls. Accordingly, controls, in risk management, are identified each iteration of the Sprint phase to avoid chaos while maximizing flexibility.
- Sprints are nonlinear and flexible. Where available, explicit process knowledge is used; otherwise tacit knowledge and trial and error is used to build process knowledge. Sprints are used to evolve the final product.
- The project is open to the environment until the Closure phase. The deliverable can be changed at any time during the Planning and Sprint phases of the project.
- The project remains open to environmental complexity, including competitive, time, quality, and financial pressures, throughout these phases.
- The deliverable is determined during the project based on the environment. [30]

## 3.1 SCRUM - CONCEPTS

Scrum is a software development process which focuses on delivery the project with short span of time. Scrum process has mid-level deliveries called sprints. Scrum consists of a basic unit called sprints. Each sprint is an intermediate release. Each sprint has resource and task plan.

The maximum life cycle of a scrum is 3 to 4 weeks. In a short span of time customer is able to view the product which makes to gain confident from the customer.

Scrum development method has a small team with combination of technical, business and management people.
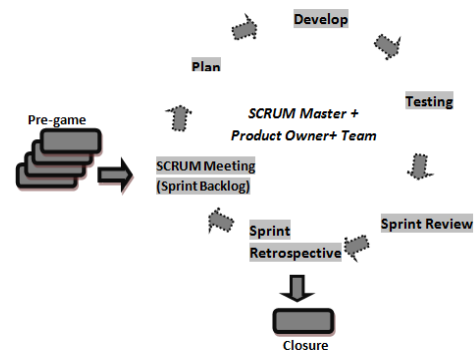


The phases of Scrum process
1. Pre-game
2. Game
3. Post-game/Closure

Scrum consists of many smaller releases called sprints. Sprint is a part of the scrum release and it has the following phases

1) **Sprint planning meeting**: In this meeting scrum product backlog is analyzed and product backlog (functionality) is identified for the release of the scrum.
2) **Scrum sprint meeting**: Each day of scrum sprint, a meeting is carried out to discuss the task carried out the day before and the task to be done on that day.
3) **Sprint development**: In this phase design, development and testing is carried for the identified product backlog.
4) **Sprint review**: In this phase testing is done on the incremental functionality of the product and the decision is taken based on the customer's feedback.
5) **Retrospective sprint closure**: Virtual closure of the sprint is done and the identified sprint backlog is added to the scrum product backlog.

Each sprint is carried out by following all the above phases of the sprint. At last when all the sprints are done then final scrum release takes place.

## 3.2 Team – Roles & Responsibilities

SCRUM team is generally managed by the SCRUM Master, the team overall is responsible for estimation, reviewing product backlog list and suggesting impediments removed from project.

**Table: 2** Roles and Responsibilities of SCRUM

| Role | Key Responsibilities |
|---|---|
| Customer | Participates in tasks related to product backlog list |

| Role | Key Responsibilities |
|------|----------------------|
| Management | Makes final decision making along with charters, standards and conventions to be followed in project<br>Participates in setting of goals and requirements |
| SCRUM Master | Ensures that project is carried according to the practices, values and rules of SCRUM and monitor the project's progress<br>Interacts with projects team, customer and management during the project & ensures that impediments are removed. Selects product owner |
| Product Owner | Responsible for managing and controlling the project and making visible the product backlog list<br>Makes decision on task related to product backlog<br>Estimates the development effort for backlog items<br>Turns the issues in the backlog into features to be developed |

## 3.3 Challenges & Benefits of SCRUM

SCRUM is very intuitive and extremely demanding methodology and poses following challenges to product development process.

*Co-Location*: SCRUM requires business and technology teams to work together and closely jointly owning the product and its development. In typically, onsite/ offshore mode, it is quite challenging and requires strong communication processes.

*Enhancement centric*: SCRUM is more apt for incremental development or enhancement to base product, hence in typical product development scenarios, a baseline architecture and key infrastructure components needs to be developed before practicing SCRUM methodology

*Quality of Deliverables*: SCRUM deploys time-boxing with fixed date of deployment with trade off scope of the sprint. In typical mindset, developers end up compromising the quality of deliverable than impacting the scope to meet the timeline. Hence, it is quite taxing and without sufficient checkpoint can impact the quality of deliverables

*Refactoring/ Design improvement*: SCRUM preaches simple design and refactoring subsequent releases, sometime in mission critical applications/ products, this approach is not well suited.

*Automation*: SCRUM can be best leveraged with extensive automation and tools for deployment, build and testing, in absence of tools/ automation, some of the best practices cannot be deployed to its completeness.

While these challenges needs an effective management and discipline, there are several advantages of SCRUM deployment.

- *Customer focused*: With disruptive transformation in the industry, customer are challenged to respond to change quickly to maintain competitive advantage, SCRUM provide a native capability to managing changing requirements effectively.
- *Time to Market*: SCRUM definitely score in quickly produces something useful, keeps making useful enhancements and hence very well suited to R&D initiatives and high risk projects where continuous customer feedback is critical to the success of project
- *Higher Team Productivity*: With continuous integration, use of automated tools, avoidance of rework, SCRUM usually improves the team productivity.
- *Shorter Release Cycles*: Hence minimal risk, very well suite for R&D initiatives and innovative products that needs customer feedback and insight on regular basis.
- *Continuous feedback*: And communication, avoids rework, reduce the risk and improve the development productivity.
- *Team unification*: Promotes one team, one goal between the business and IT teams, hence joint accountability/ enhanced acceptance and ownership of the product from business.

## 4. EXPERTS VIEW

"Good people will make good software (even without agile, SCRUM, or whatever)... mediocre and lower people will churn out similar software even with their home-grown variety of agile. However people doing agile as it were meant to... will result in better products."[29]

"Scrum introduces two really valuable differences from unregenerate Waterfall, testing both ideas and partial results with stakeholders like customers and end users, possibly via use of surrogates, and very short cycles which take advantage of seeing where you are going by facing reality early and often. That is, collaborate with your customers, and watch what you're doing and how well it is working." [32]

**Top 10 Reasons to use Scrum instead of Waterfall**
10. Scrum enables rapid reaction to changing customer requirements
9. Scrum teams possess all the required skills to get the job done
8. Scrum teams incur less technical debt
7. Scrum improves communication
6. Scrum leads to better client relationships

5. Scrum improves personnel satisfaction and commitment

4. Scrum reduces time taken to get product to market

3. Scrum produces higher quality product

2. Scrum succeeds by giving the customer what they need

1. Scrum increases productivity and lowers costs[31]

## CONCLUSIONS

SCRUM is systematic, structured and robust methodology – for small to enterprise project, it stretches the project management and engineering practices to deliver the best, in shortest time possible and continuously, hence avoid the rework and improve team productivity. It is highly customer centric with joint ownership, early planning and feedback to keep the IT team aligned to business objectives and respond with market needs appropriately. Automation is key to success for SCRUM and some of the best practices like continuous integration and test driven development can only be suited in automated mode. SCRUM today provide extensive leverage to development initiatives by improving the time to market and productivity.

## REFERENCES

[1] Ken Schwaber and Mike Beedle, Agile Software Development with Scrum (Prentice Hall, 2001).

[2]http://en.wikipedia.org/wiki/Scrum_(development)

[3]S. W. Ambler: Software Development

[4]J. Highsmith, Agile Software Development Ecosystem. Addison Wesley, 2002

[5]Elisabeth Hendrickson.: 2005: Agile Testing

[6]Somerville, I. (2006). Software Engineering. London, Pearson Education.

[7]Schwaber K and M Beedle (2002). Agile Software Development with Scrum. Upper Saddle River, NJ., Prentice Hall.

[8]Computerworld (2007)Extreme Programming Inventor talks about agile development ComputerWorld.

[9]P. Abrahamsson, O. Salo, J. Ronkainen, J. Warsta, Agile software development methods: review and analysis, VTT Technical report, 2002.

[10]K. Beck, Extreme Programming Explained: Embrace Change, Addison -Wesley, 2000.

[11]P. Checkland, J. Scholes, Soft Systems Methodology in Action, Wiley, Chichester, 1990.

[12]W.S. Humphrey, PSP: A Self-Improvement Process for Software Engineers, Addison-Wesley, 2005

[13]P. Meso, R. Jain, Agile software development: adaptive systems principles and best practices, Information Systems Management 23 (3) (2006).

[14]L. Williams, A. Cockburn, Agile software development: it's about feedback and change, IEEE Computer 36 (6) (2003).

[15]V. Rajlich, Changing the paradigm of software engineering, Communications of the ACM 49 (8) (2006).

[16]SCRUM Development Process, Ken Schwaber, Advanced Development Methods, 131 Middlesex Turnpike Burlington, MA 01803

[17]S. Jalali and C. Wohlin, "Agile Practices in Global, software engineering - A systematic map," 5th IEEE, International Conference on Global Software, Engineering, pp. 45-54, 2010

[18]Michael Coram and Shawn Bohner, "The Impact of Agile, Methods on Software Project Management," in 12th IEEE, International Conference and Workshops on the Engineering of Computer-Based Systems, pp.363-370, 2005.

[19]M. a-Vanzin, M. B. Ribeiro, R. Prikladnicki, I. Ceccato and D. Antunes, D, "Global software processes definition in a distributed environment," 29th Annual IEEE/NASA Software Engineering Workshop, pp. 57-65, 2005.

[20]Stapleton, J.: 'DSDM: business focused development' (Addison Wesley, London, 2003)

[21]Outi, S.: 'Improving software process in agile software development projects: results from two XP case studies'. Proc. 30th EUROMICRO Conf., France, 2004, pp. 310–317

[22]Laurie Williams, "A Survey of Agile Development Methodologies", 2007.

[23] Naresh Kumar Nagwani,Pradeep Singh, "An Agile Based Model for Change-Oriented Software Engineering," IEEE International Journal of Recent Trends in Engineering, vol. 1, no. 1, May 2009

[24]Pekka Abrahamsson,Outi Salo & Jussi Ronainen, "Agile Software Development Methods Review and Analysis," VTT Publication, vol. 478.

[25]Boehm, B., Turner, R.: 'Management challenges to implementing agile processes in traditional development organizations', IEEE Softw., 2005, 22, (5), pp. 30–39

[26]http://scrumalliance.org/articles

[27] Observe-Mine-Adopt (Oma): An Agile Way To Enhance Software Maintainability Jane Huffman Hayes, Naresh Mohamed And Tina Hong Gao

[28]Agile Software Development [Paperback] by Alistair Cockburn.

[29]Reference: Scrum smells on Mike Cohn's http://www.mountaingoatsoftware.com/ topics/scrum

[30]http://gowegian.5gbfree.com/scrum Development Process.pdf

[31]/www.webgateinternational.com /2012/05/top-10-reasons-to-use-scrum-instead-of-waterfall

[32]Stackoverflow.com/questions /343162/is- Scrum-evil