



ENHANCING SOFTWARE QUALITY

SCRUM – TAKE QUALITY TO THE NEXT LEVEL

PREPARED BY : PRASANNA.G & BRUNO SYLVESTER

TABLE OF CONTENTS

1	ABSTRACT.....	2
2	INTRODUCTION	2
3	STORYBOARD	3
4	SPEED OF DEVELOPMENT.....	3
4.1	SPEED OF DEVELOPMENT SCRUM VS. KANBAN	4
4.2	SPEED OF DEVELOPMENT SCRUM VS. XP.....	4
5	CONTROL MECHANISM	4
5.1	CONTROL IN SCRUM VS. KANBAN	5
5.2	CONTROL IN SCRUM VS. XP.....	6
6	HOW SCRUM CONTROLS RISKS	6
7	CASE STUDY – INFERENCES.....	7
7.1	SPEED – TESTING LIFE CYCLE	7
7.2	QUALITY.....	7
7.3	COST	8
8	SHORTCOMINGS	9
9	CONCLUSION	10

1 Abstract

All sports buffs know how important a formation is, to win the game of Soccer. When a formation brings success to Rugby, why not try it for software development? For every team or in software industry terms, for every customer's need, there is a need for a customized formation. Just like in the game of Rugby, the need for a formation that is highly comfortable with rotating pieces, one that is ready to handle all the quick moves formed the basis for inventing Agile's Scrum.

As we all know, Scrum was defined as holistic or Rugby approach by Hirotaka Takeuchi and Ikujiro Nonaka in the "New product development game". Later in 1995, Jeff Sutherland and Ken Schwaber of Easel corp. merged their experiences and industry best practices into what is now known as the Scrum methodology. The principles of Scrum not only give Rugby but also software development teams the chance to huddle around and plan for frequent deliveries. Scrum also wraps several existing engineering practices and development methods.

Heading on with Scrum and its global rollout in both large and small software companies, an organization always looks for software cycles that provide visibility into early error detection, a delivery method for frequent releases and a real good solution that actually fits their customers. The one shot solution for all of these business needs comes with Scrum and in the near future, Scrum will grab the trust of stake holders by its empirical process control.

2 Introduction

With several applications from different genres invading the market and ever changing technologies and techniques, methodologies have to take large strides to keep up to them. Traditional models like waterfall and v-models seems to be no longer in the

– Take Quality to the Next Level

picture. Agile and its processes have the limelight now. It is sailing past the others and, in particular, Scrum is what is trending in the software development lifecycle. Scrum and its formations have attracted several software development organizations and it has lived up to its origin proving that formation is what it is all about.

Software companies that want to boost their operating efficiency need to advance and streamline their software development practices. Agile has helped leading industry players overcome dependencies and steep overheads. This has helped to meet industry regulations providing an opportunity to improve collaboration and enhance user experience.

3 Storyboard

Supported by a case study, this paper helps measure speed, performance, quality, etc. across some of the popular methods in Agile such as Scrum, Kanban and XP and it illustrates how Scrum is better in its own way therefore ultimately leading to the most important aspect – Focused Releases – thereby winning over also in cost.

Requirement	A Point Of Sales web application in development needs testing as new features are added.
Test team size	2 manual testers
Timeline	6 weeks
Total Story Points in Backlog	50 story points
Modules (Assigned points)	License (9); EDI (6); Order Entry (8); Report (13); Pricing (7); Maintenance (7)

Figure (i) – Scrum Board

4 Speed of development

When it comes to development, Scrum, XP and Kanban are on the same lane. Scrum plans and assigns prioritized backlog to releases and teams, using a simple drag-and-drop whiteboard planning environment. Scrum is based on having a fixed iteration cycle. You can choose the length of the iteration, but the general idea is to keep the same length of iteration over a period of time and thereby establish a cadence.

- **Beginning of Iteration:** An iteration plan is created, the team pulls out a specific number of items from the product backlog based on the product owner's priorities and on how much the team thinks it can complete.
- **During Iteration:** Team focuses on completing the items they committed to.
- **End of Iteration:** Team demonstrates working code to the relevant stakeholders, ideally this code should be potentially shippable (i.e. tested and ready to go). The team then does a retrospective to discuss and improve their process.

So Scrum iteration is one single cadence combining three different activities: planning, process improvement, and (ideally) release. Because the length of the iteration cannot be altered, with Scrum one knows how much time he has exactly at hand (we have a 30 day cycle which will not be altered) and the Scrum master carefully prioritizing the story points, a majority (about 75% - 37 story points) of them in the backlog will get into the iteration grouped based on modules.

4.1 Speed of development Scrum vs. Kanban

Unlike Scrum, Kanban iterations are not prescribed. You can choose when to do planning, process improvement, and release. In a typical Kanban team, the scenario would be mostly event-driven. They would ideally plan over a meeting when they run out of items to work on. This process cuts the speed because it involves

- Self-limiting of work
- Less governance over the system

4.2 Speed of development Scrum vs. XP

XP starts with pair programming on design phase and carries over with test-driven development and continuous integration. The possibility of bottlenecks is high with this process having continuous integration. XP teams build multiple times per day, keeping the system fully integrated at all times. Infrequent integration leads to serious problems on a software project.

Integration is critical to shipping good working code. Infrequently integration often results in buggy code. Also weak integration process leads to long code freezes. Code freezes lead to long waits that holds back developers from working on important features, thus delaying the process. This weakens your position in the market, or with your end users. In addition to time consumption for integrating the system often, XP also believes in revisiting their estimates therefore building uncertainty in finish dates.

5 Control Mechanism

In Scrum the items for a sprint are chosen by the Scrum master and he controls the product development and manages the team. The Scrum master knows the output of his product therefore there wouldn't be a problem in selecting the right item at the right time. Scrum also closely tracks progress by making use of its board or wall as it is called, keeping everyone on the same page. This not just helps to see where we are but also helps anticipate risk.

In Scrum, the sprint backlog is just one part of the picture – the part that shows what the team is doing during the current sprint. The other part is the product backlog – the list of things that the product owner wants to have done in future sprints. The product owner gains an insight

– Take Quality to the Next Level

into the sprint backlog but he cannot alter it when already made part of a sprint. Here the Scrum master has complete control on what should and should not get into a sprint. Referring back to our cases study the 4 modules picked - Order Entry, Pricing, License, Report (75% of the backlog) - are made part of the committed list. Once this is picked, there is no change in the list. It moves from Committed to Ongoing to Done.

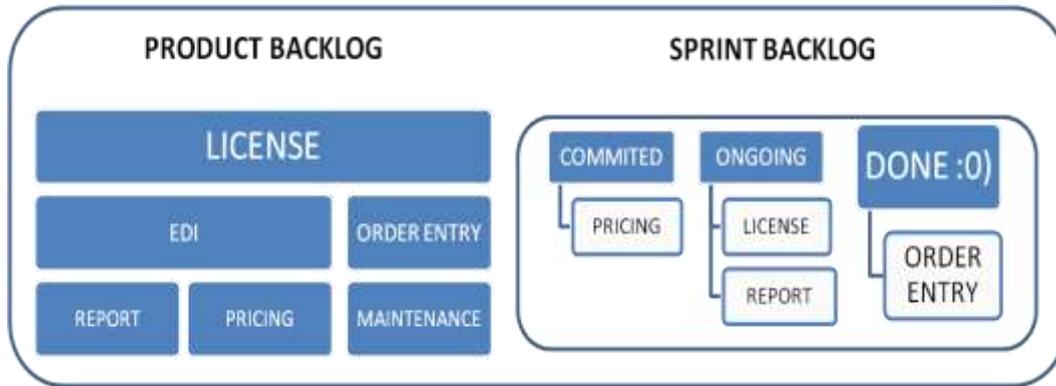


Figure (ii) – Scrum Board

When the sprint is done, the team “delivers potentially shippable code” to the product owner. So the team finishes the sprint, does a sprint review, and proudly demonstrates features Pricing, License, Report, and Order Entry to the product owner. The product owner can now decide whether or not to ship this.

5.1 Control in Scrum vs. Kanban

In a Kanban setup the “Backlog” column is just a general wish list of the entire team, in no particular order and no one is specific has complete control over it. Under a similar scenario, a Kanban board might look something like this:



Figure (iii) – KanbanBoard

Here the whole work flow is on the same board. The “selected” column contains the high priority items, with a Kanban limit of 2. So there may be only 2 high priority items at any given moment. Whenever the team is ready to start working on a new item, they will take the top item from

Selected. The product owner can make changes to the Backlog and Selected columns any time he likes, but not the other columns. Switching focus back and forth can lead to quality dips.

5.2 Control in Scrum vs. XP

The main disadvantage here in XP is the ownership of selecting items from backlog is in the developers' control. This practice may lead to trivial selection of product backlog and customer may not get the desired product at the right time.

XP does not have sets of specific development process that gives importance to defined roles, conducts frequent meetings and monitors the progress constantly. While poor communication between the team is one of the main reasons for poor outputs in XP, good communication is stressed between all the members but this is very hard to achieve because there is no set role specified. There also does not seem to be a specific tracking system such as a board or a wall.

6 How Scrum controls risks

An unforeseen risk is always dangerous. Sizing risk is a little harder than sizing a feature. With a strict timeline for Scrum, it is only wise to have the estimates right. Estimation does not stop with adding points to a requirement but this should also include buffer for these unforeseen risks. Risk assessment fits most naturally into the release planning and product backlog grooming sessions. Here is an approach in Scrum that will help mitigate risk:

- As the team grooms the product backlog, pick items that represent risks
- For every risk or risky product backlog item:
 - ✓ Assess the probability of the risk occurring during the course of the iteration
 - ✓ Assign points to the risk based on the impact it would have on the application
- Measured total risk in unit points
- The risk and the product backlog items are tracked in a similar manner

During these estimation sessions, there can arise challenges like:

- The team finds it difficult to come up with a size estimate on a product backlog item. Let us say the EDI module is new and there is still more to understand before we can estimate it. To arrive at an estimate as close as possible the Scrum master should encourage “what if” questions during the estimation sessions.
 - How one understands a feature may not be another's perspective. As a Scrum master encourage an open discussion. Be sure the feature is understood right before the end of the session.

The overall release plan is still viewed as an optimistic projection even though the team has been careful to base it on their actual performance.

7 Case Study – Inferences

Here is the inference of the case study. The project followed Scrum and based on the above collective pointers, an attempt has been made trying to reason out how Scrum was better than Kanban and XP.

7.1 Speed – Testing Life Cycle

In Scrum the requirements are clearly defined in the Backlogs therefore reducing the chances of deferring. In the case of XP and Kanban since there are chances of deferring a requirement in the middle of a sprint, time spent on it may not be effective resulting in extended timelines not being able to deliver on time. XP's refactoring and test driven approach consumes time which can also impact timelines. Also XP requires Acceptance and Integration test during the sprints. This practice may lead to bottlenecks and poor test coverage.

Kanban which is more relaxed and since estimation & team commitment to a specific amount of work for an iteration becomes optional, the team may fail to deliver the shipment to the customer on time.

Finally since the WIP in progress is complete at the end of the Sprint in Scrum, it ensures good test coverage.

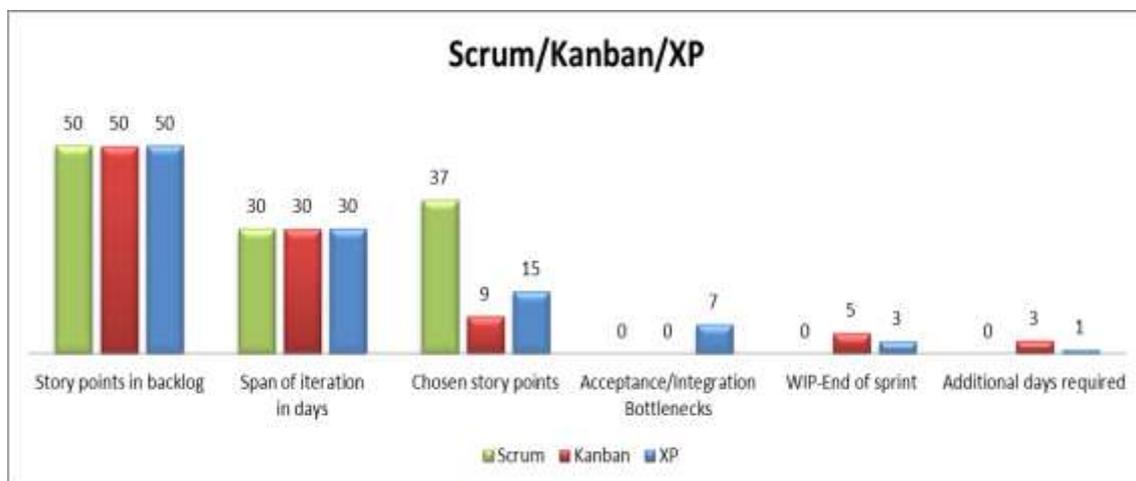


Figure (v) – Scrum/XP/KANBAN

7.2 Quality

Usually Agile kick starts with a bunch of backlogs. And since it is the Scrum master who picks and chooses, there is complete control and related requirement that will make a good combination to develop is chosen. When it comes to early defect detection Scrum hammers XP and Kanban. Since requirements are changing with Kanban and XP testers find it difficult to maintain the focus. Scrum will have the focus till the end of the sprint.

As XP and Kanban hold no clarity with requirements, it may lead the tester to look at the application in a different perspective and invalid bugs will be posted from test side. Also if you go back to “Figure (iv) – Scrum/XP/KANBAN” ‘s WIP-End of Sprint, the stories yet to be completed have to either be rolled back introducing bugs because of reduced time available for regression OR an extension of time will have to be requested resulting in delayed delivery.

The above reasons explained, the following chart projects defect detection with Scrum, XP and Kanban for the 6 week iteration.

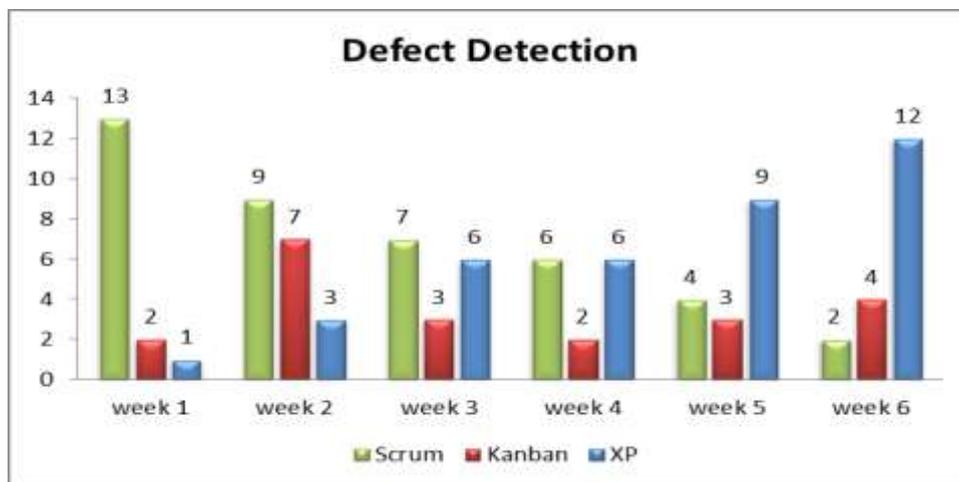


Figure (vi) – Defect detection

7.3 Cost

In first few weeks, Scrum will find bundle of defects and gets done with it. As we know, the earlier a defect found is less costly it is to fix. Also when quality starts improving the tester can be allotted to the next sprint if it is a leap frog approach. XP may have unclear requirements which may lead to spending more money. Kanban cuts short in committed work resulting in need for additional time and money.

– Take Quality to the Next Level

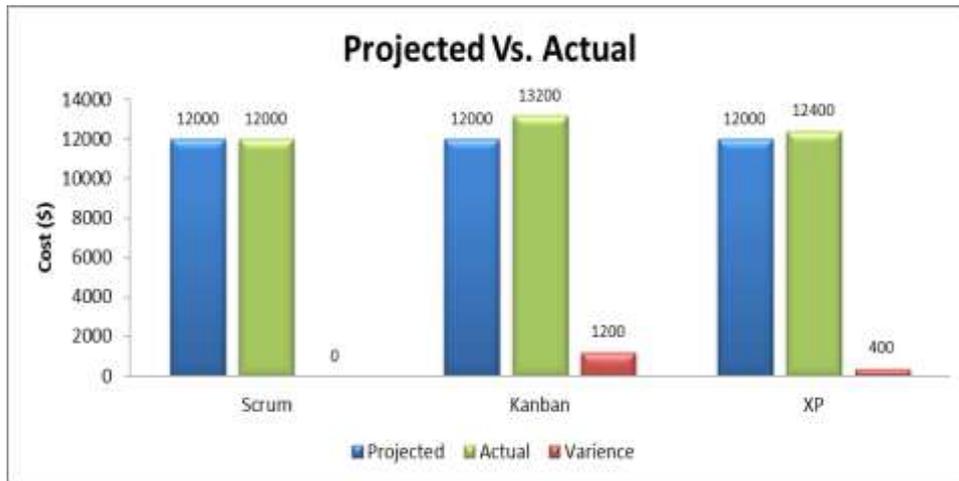


Figure (vii) – Cost – Projected vs. Actual

Key Lever	Scrum vs. XP / Kanban	Reasons
Test Execution	Faster by 25%	<ul style="list-style-type: none"> Scrum master's complete control Focused requirements
Defect Detection	Increased by 30%	<ul style="list-style-type: none"> Invalid defects count is minimal saving time on review
Quality	Increased by 35%	<ul style="list-style-type: none"> Regular catch-ups preventing de-railing
Backlog	Reduced by 35%	<ul style="list-style-type: none"> Stories for the sprint from the product backlog is selected by the Scrum master which assures the right requirements are picked
Cost	0% Budget Increase	<ul style="list-style-type: none"> Work is done earlier can help release resource from current Sprint

Table (i) – Scrum vs. XP and Kanban

8 Shortcomings

Scrum does have a few disadvantages:

- Unlike Extreme Programming (XP), Scrum teams do not allow changes into their sprints; XP welcomes changes to the current iteration. XP team would use velocity at the project or release level but not as a metric from iteration to iteration.
- Although Kanban teams don't need iterations, Kanban does suggest having a cadence of input and delivery. Most will not require a planning day so performance in terms of velocity has got nothing to do with Kanban but Scrum rigidly uses velocity as a key performance indicator.
- Maintenance work (bug fixing): Unless the Scrum team is quite small, the team will rarely work one bug at a time. Teams doing full time maintenance work should probably look at Kanban rather than Scrum although the daily stand-up meeting should

still happen.

- When dissimilar projects are very short: If the Scrum team is working on dissimilar projects, i.e. projects where technology or product domain are not related, and with short sprints, it's not likely that the velocity would ever stabilize or the velocity established on the previous project would be applicable.
- Roles in Scrum: If the Scrum master, product owner, developers and testers all on one team are not measured the same then this can lead to either perceived or real conflicts of interest. Product owners write the user stories, developers and testers implement and test against the Scrum team's definition of done, and the Scrum master is responsible to teach the Scrum team by coaching and by leading it to be more productive and produce higher quality products. All these factors affect velocity and everyone needs to be equally accountable.

9 Conclusion

Several software development companies and organizations have adopted Scrum as the preferred method to produce functioning, deliverable products in an environment where system requirements are not fully defined at the beginning of the development process. Scrum achieves this flexibility and adaptability through a well-defined development process designed to recognize and respond to changes in the environment. Practicing Scrum can therefore assure confident delivery taking quality to the next level.

About Indium:

Indium Software is exclusively focused independent software testing services firm since 1999. Over the years, Indium mastered objective methods that minimize the risk of failure of applications and software products. With a global headcount of over 300 employees, Indium works for a mix of marquee Enterprise and ISV clients spread across the globe. Indium is aggressively pursuing the Social, Mobile & Cloud agenda to make these the core of our next wave of service specialization.

Contact us:**USA - Indium Software Inc.****Corporate Office
California**

Suite 210, 1250 Oakmead Parkway
Sunnyvale, CA - 94085.
Phone: +1(408) 501-8844
Fax: +1(408) 501-8808

Atlanta

Crown Office Suites,
1870 The Exchange, Suite 100,
Atlanta, Georgia 30339. USA
Phone: +1(770) 989-7302,
+1(678) 608-0809

INDIA - Indium Software (India) Limited**Global Delivery Headquarters
Chennai**

VDS House
41, 2nd Floor
Cathedral Road
Chennai - 600 086.
Phone: +91-44-2811 6330
Fax: +91-44-4210 4033

Prince Arcade Building
22-A, 3rd floor
Cathedral Road
Chennai – 600 086.
Phone: +91-44-4347 7200

Steeple Reach,
Old No. 25, New No. 39,
Cathedral Road, Chennai –
600 086.
Phone: +91-44-4347 7122

Bengaluru

Salarpuria Adonis, 2nd Floor
Binnamangala, 3/1, Old Madras Road
Kadranapalya, Indira Nagar
Bengaluru – 560 038.
Phone: +91-80-6784 7500

New Delhi

F-1/5 Okhla Industrial Area,
Phase-I,
New Delhi – 110 020.
Phone: +91-11-6613 0400

Sales Inquiries

americas.sales@indiumsoft.com
apac.sales@indiumsoft.com
emea.sales@indiumsoft.com
india.sales@indiumsoft.com
sales@indiumsoft.com

General Enquiries

careers@indiumsoft.com
info@indiumsoft.com