ETM 549/649: Management of Technology innovation

The Role of Software Development Methodologies in the Technology Innovation Process

Payal Agrawal

Department of Computer Science, Portland State University

Portland, OR

Abstract – The software development methodologies like Scrum and eXtreme Programming (XP) are used in efficient development of software products. These methodologies follow some agile practices like communication, customer integration, feedback, collective ownership, simplicity, continual integration, daily scrum meetings, and iterative development. Further, innovation is the process which starts from conception of an idea and ends with successful development of the product. In this paper I would explain how each of the agile practices followed in Scrum and XP will support and enhance an innovation process. Thus, the agile practices which would be very handy for most of the software professionals can help them in participating in the innovation process. Various case examples will be presented in order to prove how the strengths of software development methodologies can be used to enhance innovation process.

I. INTRODUCTION

A. Introduction

With constant pressure of accelerating the product development lifecycle and produce more at lower cost, the organizations have shown a lot of interest in Scrum and XP practices. The software

development methodologies like Scrum and XP help in improving the software development quality by breaking the task into smaller units, and thus easily accommodating customer's changing requirement. Schwaber and Beedle [6] emphasized that Scrum is an approach that "reintroduces flexibility, adaptability, and productivity into systems development". Moreover, the term "eXtreme Programming" was introduced by Kent Beck in his book "*Extreme Programming Explained: Embrace Change*" [1], since then it has rapidly replaced traditional method of software development such as Waterfall model. Kent describes XP as a software development discipline that organizes people to produce higher quality software more productively. Both of these methodologies have proved their strengths in software product development from NASA's Project Mercury [7] in the early 1960 to Intel's Silicon testing tool in 2006 [14].

B. Research question

In this paper, my research would be to know how useful Scrum and XP methodologies in software development. Also, what relationship do Scrum and XP share amongst themselves. Are XP and Scrum really different? Is XP better that Scrum? Is Scrum better that XP? Are they same? The innovation is the exploration and exploitation of new idea (product, process, or service) for technology advancement and economic growth. The innovation process is broadly divided into three stages: Idea generation, Idea conversion and Idea Diffusion. Software development methodologies and innovation process are two entirely worlds. In this paper I am investigating how I can bring both of these worlds together, in fact use the strengths of software methodologies to improve the innovation process. Moreover, how the different stages of innovation process would react when Scrum and XP practices were applied to them.

C. Thesis Statement

In this paper I would explain two software methodologies – Scrum and XP, how they complement each other for successful software product development, their common practices. There are many agile practices which are common to both Scrum and XP: communication, customer integration, feedback, collective ownership, simplicity, continual integration, and iterative development ("sprint"). I will conclude the paper by suggesting that it will be in best interest if organization uses Scrum's management (process) practices and XP's development practices. Further I will be investigating innovation process

model and intend to show the *mapping of software methodologies practices and innovation process model.* In this paper, I will conclude by suggesting that agile practices of Scrum and XP can be used in improving organization's innovation process. I would substantiate my point by taking three examples where companies have used agile practices and improved their innovation process: *Oxygen Media, Intel's Silicon Testing Tool, and Honda urban car – Honda Civic.* In the end I would like to conclude that a by following these simple practices the organization management can not only improve its innovation process but can also motivate its employees to contribute in innovation process.

II. LITERATURE REVIEW

In process of writing this paper, I have done a large literature search. Beck and Andres in their book "*Extreme Programming Explained: Embrace Change*" [1] have described various agile practices followed in XP. Similarly, Schwaber and Beedle in their book "*Agile Software Development with Scrum*" [6] had described how Scrum's practices can be effectively used in software development project. Through the work done by Beedle it was evident that Scrum teamed up with XP was an effective solution for quality software production. In order to know more about the origin of Scrum practices I referred Nonaka and Takeuchi's "*The new new product development game*" [9]. They took example of various companies like Fuji, Canon, Honda and Xerox which were stuck up in a competitive situation. These companies used successfully used scrum practices to come up with new "break-through" product to regain their market strength.

Through my research I concluded that Scrum and XP are not totally different but complement each other. I studied various case studies which helped me to come to my conclusion. First, Robert and Schwaber's *"Best Practices in Scrum Project Management and XP Agile Software Development"* [12] studied Primavera System, a software company, which used agile practices and were able to make both employees and customer happy. It was win-win situation for both of them, since employees started working 8 hour a day (~40 hours a week) in a healthy environment and customer was getting good quality product in the timeline specified. Second, Rasmusson in *"Introducing XP into Greenfield Projects: Lessons Learned"* [4] presented a case example of Thoughtwork's who had to implement agile practices in Client Company, TransCanda Pipeline Limited. This case example emphasizes on why an organization should think ahead of traditional approach of software development and adapt to agile practices. Rasmusson presented XP side of agile practices where people get together, shared experiences, and collectively looked for solutions.

Next, I focused on the role of different agile practices in organization's innovation process. Baumgartner in his while paper "*The Corporate Innovation Machine*" [23], explained the various stages in organization's innovation model. Moreover, Siebra and colleagues in their research paper "*Deciphering Extreme Programming Practices for Innovation Process Management*" [22] presented the use of communication, co-operation and feedback in innovation process. Though some of stages of innovation were mentioned in details, but concrete mapping of different innovation stages and agile practices was missing.

Through this paper I not derived a mapping of agile practices and innovation process model but substantiated my result with various case examples. First, Judy and Beens in their paper "*Ript:Innovation and Collective Product Ownership*" [24] have mentioned that how Oxygen media's CEO used "collective ownership" (practice of Scrum and XP) to improve the organization's innovation process. Second, Kelly, and Finglas in the paper "*System Innovation Plus Extreme Programming Improves IT Software*" [14] suggested that though Intel was able to generate innovative ideas by evaluating customer's feedback they were not able to translate a great innovative idea into a product, tool or software. The use of XP practices led to the efficient development of Silicon Testing Tool. Third, Nonaka and Takeuchi in their book, "*The KnowledgeCreating Company: How Japanese Companies Create the Dynamics of Innovation*" [19], showcased a very interesting example of Honda's idea of new urban car – Honda Civic. To achieve this new theory of Automobile Evolution, they used the various agile practices like using tacit knowledge of employees, various cross-functional teams working together in short iterations. The result was creation of new market and satisfied customers.

III. REASERCH METHODS

The research method for this paper was to complete a large literature search of research papers, journal articles, and books which will give more insight into software methodologies Scrum and XP. Further, I looked into how well they can map with innovation process. I analyzed existing literature which explained various practices followed in XP and Scrum, but did not mention how to make the best possible use of those practices. I also analyzed literature which explained various stages of innovation process, but those current researches provide little information regarding the concrete mapping of various agile practices with different stages of innovation process model. I also conducted interviews with software professional that have experience working with Scrum, XP and often participate actively in the innovation process.

In order to substantiate my research I took advantages of various case studies. I tried to study different organization which have successfully used agile practices and accelerated the product development lifecycle and produce more at lower cost. I studied different organization (which vary according to their size, structure, and customer base) and yet have successfully used software methodologies in improving their organization's innovation process.

IV. DATA COLLECTION AND FINDINGS

A. SCRUM

The term Scrum was first used by Ken Schwaber and Jeff Sutherland. But the concept of SCRUM was known since long. Nonaka and Takeuchi [9] mentioned about lot of ideas which are later described in Scrum. They took example of various companies like Fuji, Canon, Honda and Xerox. Each of these companies where stuck up in a competitive situation and they had to come up with new "break-through" product to regain their market strength. The management chose cross-functional teams and asked them to come with a new product idea in a specified time frame (*time boxing*). All the team members were experts in their

respective fields. They were given full freedom to choose and implement the work-flow. The end results were very promising each and every team came up with a product which created history. For example, Fuji-Xerox's FX-3500 medium-sized copier, Canon's PC-10 personal-use copier, and Honda's car with 1200 cc engine (Honda Civic) which had great impact and visibility within the company.

The overview of SCRUM Process is shown in Figure-1 [18]. In the beginning of the product development process the prioritize list all product requirements like functionality, features, and technology called *Product Backlog* is decided. Small, cross-functional teams called *Scrum team* takes as much product backlog as they think they can deliver in 30 days iteration, or *Sprint* [6].

The Scrum team maintains a list of task to perform during each Sprint that is called *Sprint Backlog*. During the Sprint, a management representative (*Scrum Master*) enforces Scrum practices. Scrum Master helps Scrum team to deliver the Sprint goal which is to produce a shippable product increment (which includes coding, testing, documentations, code checked-in and meeting coding standards). The Scrum team meets daily for a short status meeting, called the *Daily Scrum Meeting*. The progress of the Scrum team is observed through this Scrum meetings – this gives Scrum methodology its unique *transparency* characteristic. At the end of the Sprint, the management meets with the Scrum team to inspect the product increment is delivered.



Figure-1: Overview of Scrum Process.

Role in Scrum – Chicken and Pig Role

A number of roles are defined in Scrum. All roles fall into two distinct groups—*pigs* and *chickens*—based on the nature of their involvement in the development process [16]. These groups get their names from a joke about a pig and a chicken opening a restaurant [See Figure-2].

Pig roles: The Pigs are the ones committed to the project in the Scrum process. For example, Product Owner, Scrum Master, and Scrum Team.

Chicken roles: Chicken roles are not part of the actual Scrum process, but must be taken into account. They are people for whom the software is being built. For example, Managers and Stakeholders like customers, vendors.



Figure-2: Chicken and Pig joke

Case for scrum:

Game companies have to launch new games in a small time-frame, so they use Scrum because of its fast, incremental, iterative development and use of small sprint cycles. Every game company now-a-days uses Scrum, one of the example is *High Moon Studios* (owned by Sammy Saga a Japanese company). According to Clinton Keith [10], CTO of High Moon Studios, they adopted Scrum in 2004 in the last year of development of our debut game *Darkwatch*. Major problems they faced were rigid schedule, changing user requirements, and slipping deadlines. The employees of the company were working for 12-14 hours a day but instead of increasing productivity there was 60% increase in the defects which increased the overall cost of the game. SCRUM was very handy and easy to use – they were able to put there project back on track. They were able to finish the game within six months of the original projected date. Team size of ~10, daily scrum meeting and shorter sprint cycles helped them to create new version of the game every two-four weeks. He and his organization had better understanding of the game development process and project management techniques.

B. Extreme Programming: Engineering Practices

XP – a method developed by Beck [1] is very well-known agile methodology. XP stresses on delivering immediate business value to the customer. According to Highsmith and Colleagues [8], "XP's 12 practices,

for example, were never intended to be all-inclusive rules; instead, they are generative rules that interact in concert when a team of individuals practices them". Although Extreme Programming itself is relatively new, many of its practices have been around for some time. Some of the XP practices were successfully used in *NASA's project mercury* [7] in 1960. Project mercury had the goal of putting a human in orbit around the Earth. They followed XP practices like short-half day iterations, TDD, continuous integration. The project was huge success and everyone saw the first successful example of using XP methodologies.

Beck came up with XP Practices:

- Incremental Planning: The user-stories are collected from the customers. These stories specify the features which are required by customer in next release. These requirements are broke down into specific tasks.
- 2) **Small Releases**: Similar to the evolutionary software development model, XP delivers a series of fullyfunctional and completely tested *small releases* leading up to the delivery of the final product.
- 3) System metaphor: The system metaphor is a story that everyone customers, programmers, and managers - can tell about how the system works [1]. This is the way through which all the people involved in the project can visualize the product development. This practice is not as important as Testfirst design and Refactoring.
- 4) Simple Design: "Simple is best" is the approach used in XP. The design is kept as simple as possible so those bare minimum requirements are targeted. According to Boehm and Turner [25], the main aim for designing is to "Designing for the battle, not the war". The motto is YAGNI (you aren't going to need it). The antimotto is BDUF (Big design Up front). Since, change is inevitable, planning for future functions is a waste of efforts.
- 5) Test-first development (test-driven development): This is one of the most important practices of XP. Here, unit-test cases are written even before writing the code. The most important *rule of XP* – a feature is not done until it passes both its unit tests and its acceptance tests [12].

- 6) Refactoring: Refactoring is the process of changing a software system in such a way that it does not alter the external behavior of the code yet improves its internal structure [15]. As an XP practice, Refactoring helps in keeping code simple and maintainable.
- 7) **Pair Programming**: The developers work in pair on one workstation. While one developer is coding another is simultaneously reviewing the code. This provides feedback on a minute-by-minute basis.
- 8) Collective ownership: This emphasize that the piece of code (software) is owned by all. All developers check in the code at central repository. This practice encourages the feeling of "whole team" amongst all team members.
- 9) Continuous Integration: Since many developers are simultaneously writing code, the code should be integrated in the shared repository on day-to-day basis. After any such integration, all the unit tests in the system must pass.
- 10) **Sustainable pace (40-hour work week)**: XP does not support the *overtime or late-working hours* of the team members. It is believed that exhausted developers would result in reduce code quality and medium-term productivity.
- 11) **Customer Integration**: In XP, the customer is the member of the development team. He specifies the product requirements and even helps in prioritizing the requirements. Customer's input helps in speeding the development process.
- 12) **Coding Standards**: Coding standard is an agreed upon set of rules that the entire development team agree to adhere to throughout the project [17]. This is especially useful in order to successfully implement "continuous integration" practice. Since, developers can integrate the code successfully if they all adhere to same coding standards.

I think the traditional software development approach of "do it the right first time" with objective of satisfying the "original contract" [3] is not suitable for projects with requirement variability. Today's scenario surrounds around changing user requirement so XP approach which emphasize on "do it right the last time"

with the objective of "delighting the customer" is best suited. XP approach to software development has constantly proved that it can be more productive than traditional approach.

C. XP @ Scrum: Common Practices

XP and Scrum share some common features like *Incremental Planning, Small releases, System metaphor, Simple Design, Collective ownership, Continuous Integration, Sustainable pace, and On-site customer.* Both of them do not support long work hours and believe that exhausted employees will produce more error and hence, productivity will be reduced. They share a common rule – a feature is not done until it passes both its unit tests and its acceptance tests. Some of the common features are:

- 1. Split the organization into small, cross-functional, self-organizing teams.
- 2. Split your work into a list of small, concrete deliverables. Sort the list by priority and estimate the relative effort of each item.
- 3. **Time boxed Iteration**: Split time into short fixed length iterations (usually 1—4 weeks), with potentially shippable code demonstrated after each iteration.
- 4. **Transparency**: Optimize the release plan and update priorities in collaboration with the customer, based on insights gained by inspecting the release after each iteration.
- 5. **Tacit Knowledge**: Establishing and updating project knowledge in the participants' heads rather than in documents (explicit knowledge).
- 6. *Embracing change*: Seeing change as an ally rather than an enemy. Change allows for more creativity and quicker value to the customer.

Are XP and Scrum really different? Is XP better that Scrum? Is Scrum better that XP? Are they same? According to me, the answer to these entire questions is XP and Scrum is mutually compatible. They go hand-in-hand. In order to make the most of both we should implement *Scrum's organizational process* and *XP's programming techniques*. According to Mike Beedle [6], who has worked a great deal in wrapping XP engineering practices with Scrum, "XP enhances the quality of the software developed and Scrum enhances the day-to-day management of the projects". He call this union **XBreed** (crossbreed).

XP and Scrum: Friends (Mutually Compatible)

XP is more prescriptive then Scrum [11]. Prescriptive mean "*more rules to follow*". XP gives you more practices to follow as compared to Scrum [See Figure-3]. It includes most of Scrum and some additional engineering practices such as test-first development, refactoring, and pair programming. On the other hand, Scrum is more adaptive, which means "*fewer rules to follow*". Scrum does not provide any engineering practice.



Figure-3: XP is more prescriptive than Scrum.

Thus, while XP focuses on engineering practices, Scrum proves its usage in management techniques. I believe that in-order to make the best of both worlds; we should follow the *mix and match* approach [See Figure-4]. I assume that it is necessary for the success of project to use Scrum features (like Sprint and daily

Scrum meeting) and combine it with XP engineering practices (test-first development, pair programming, coding standard, and refactoring). For example, early and fast feedback is needed for Scrum, and XP practices can help in providing better control over feedback.



Figure-4: Mix-and-Match approach – XP and Scrum come together.

Now I would present a case study which will show that by using a blend of Scrum and XP practices everyone in team has started enjoying working with each other, overtime has been eliminated, customers were pleased, the competition was vanguished, and bonuses started flowing.

Case one: "Primavera example"

The challenges faced by *Primavera System* [12], a software company, were that the project 3.5 release was due, employees were working overtime but then also they missed deadline. Further, the implications were unhappy management and low morale of team members. Primavera decided to try Scrum and XP practices; they used Scrum's planning and management of project and XP's development and testing practices. They formed 9 teams of about 10 people each. Primavera had used a waterfall development process where analysts analyzed, coders coded, designers design, etc. However, under the guidance of Scrum they now employ cross-functional teams. They used following XP practices: continuous testing, continuous integration, and continuous improvement of the code. The major accomplishment was *team started working together*. XP provided better control and feedback, which resulted in fast development.

Result: The release was a knockout success; it contained the functionality of two releases in one release, it was completed without anyone working overtime, and it was the result of a team environment in which marketing, product managers, developers, quality assurance, and executives worked together. Thus, the project was a success.

Case two: "Though Work's Experience"

This example shows that how agile practices were introduced by *ThoughtWork's* in *TransCanada Pipelines Limited (TCPL)*. According to Rasmusson [4], Thought Work's had to implement XP practices in Client Company, TCPL where the project team size was 13 people. The time to implement XP was six months. Rasmusson mentioned that there were many challenges because they had to train the team, learn the business, and produce a production ready legacy application in six months. Most important task was to deal with software developer's attitude towards new and changed practices like pair programming, testdriven development, Co-ordination and communication with on-site customers.

Thought Work's were successfully able to implement all of XP practices. The legacy application was huge success for all teams and was delivered on time. TCPL was so impressed that they used XP practices in their future projects as well. This shows that is an organization think ahead of traditional approach of software development and adapt to agile practices, teams were not only able to deliver on time but they started to get together, share experience, and collectively looked for solutions.

Case three: "US Strategic command's SKIWeb"

SKIWeb, or the Strategic Knowledge Integration website, is an event tracking and blogging tool used at USSTRATCOM [21]. Earlier USSTRATCOM was using traditional software development approach, which follows a strict plan-based approach. The plan-based approach makes it difficult to adapt to changing requirements. USSTRATCOM wanted to add a new advanced search capability to SKIWeb. The search capability has changing operational requirement. So team managers decided to use XP practices.

With the use of agile practices the advanced search feature of SKIWeb was of a good quality, had fewer defects, and better met customers' needs. McDonalds and colleagues concluded that USSTRATCOM management learnt two important lessons. First, the importance of team member communication. They liked the agile's collaborative style communication when compared to traditional hierarchical communication. Thus, more emphasize was placed on importance of communication in development process. Second, management involvement in the on-going development process. They realized that management involvement in innovation process helps in creating and supporting a healthy environment.

D. Innovation process and its goals

Tom Garman, in his book "Innovation" defined innovation as the art of developing a new product, service or process based upon a new idea. He mentioned that, "Innovation is the physical form of the idea new product, service or process that you can use to make money". In today's global economy innovation is the lifeblood of any organization. Innovation is a creative idea that is realized. According to *Van de Van* [26] *Innovation means inventing and implementation new ideas.* In order for innovation to take place a process model needs to be followed. The innovation process model is needed for understanding how to implement an effective, idea management based innovation strategy in an organization [23]. The innovation process model is broadly divided into three stages: Idea generation, Idea conversion and Idea Diffusion. We can further divide the innovation process model into various stages like *Creative environment, opportunity identification by studying market demand, generation of ideas, innovation validation, prototype (concept) development, product development, and product marketing (diffusion).*

When an organization or an individual thinks of innovation, some of the main goals are to have improved quality of product, creation of new product, integration of new product in existing market, user (customer) acceptance, reduced production costs, and reduced use of resources. If we take closer look on all of these goals we can see a close association with the objectives and benefits of agile practices. Here is an example: Innovation depends on creativity of individuals and creative environment to support their creativity [23]. The two of the principles of agile methodology are just to support the "creative environment" and "freedom to individual" [27]. These principles are:

- "Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done."
- 2. "The best architectures, requirements, and designs emerge from self-organizing teams."

I noticed that we can establish a strong relationship where some of the agile practices can enhance the innovation process model. I believe understanding this interrelationships is as important as practices themselves.

E. Mapping of Software methodologies practices and Innovation Process.

In this section I will show that agile practices are not just restricted to management and development of software projects, but expand their spectrum to all complex organizations [*See Figure-5*]. They help in overcoming the problems and enhancing the innovation process model. Different agile practices help in different stages of innovation process.

Creative Environment: "Is easier to enhance creativity by changing conditions in the environment than by trying make the people think more creatively." – Mihaly Csikszentmihalyi, (1997). The creative environment foster in individual the inclination to come up with new ideas or may be the recombination of old ideas. So, if we take a closer look at what agile practices can help in creating such an environment is collective ownership and communication. Collective ownership because if an individual has a sense of belonging and ownership of the product, he or she will be self-motivated to come-up with new ideas to enhance that product. Communication, because if an individual (probably a team member) is being given

an opportunity to communicate with other team members (may be customers), it gives him fair idea of what exactly customer wants and most importantly the changing market needs. The importance of communication is clearly mentioned by Khurana and Rosenthal [28] where they studied different companies and analyzed their new product development process. They showed that the company whose all team members communicated proved to be more successful.

Opportunity Identification (Market demand or R&D): According to Nonaka and Takeuchi [9], "At the individual level, all members of an organization should be allowed to act autonomously as far as circumstances permit. By allowing those to act autonomously the organization may increase the chance of introducing unexpected opportunities". Thus, *collaboration, communication, and association* (agile practices) amongst individuals are the key for introducing new opportunities, since they get direct exposure to the ideas, life experiences, and learning of staff with whom they would ordinarily not collaborate.

Further, the greatest challenge in order to identify opportunity is that market demands are not static – they evolve. The agile practice – *customer integration* comes to rescue, which lays emphasize on understanding customer need and identifying the potential technologies and applications to satisfy them. Sometimes, the new opportunity comes across by a technology breakthrough. It starts within the R&D department who generate a new technology-driven model. I believe that the *daily scrum meeting* and *transparency* within R&D department would reveal the new model as soon as it is conceived.



Figure-5: Mapping of Innovation process model and agile (Scrum and XP) practices.

Generation and Sharing of ideas: Amabile, in his book *Creativity in context*, mentioned "*All innovation begins with creative ideas* . . . We define innovation as the successful implementation of creative ideas within an organization. In this view, creativity by individuals and teams is a starting point for innovation; the first is necessary but not sufficient condition for the second". The most important source of generation of creative ideas is imagination, *tacit-knowledge* and problem solving capability of an individual. According to [knowledge creating companies], tacit-knowledge is very informal and partly consists of technical skills. It is normally hard for an individual to formalize his or her tacit-knowledge but it definitely affects their work and findings. Agile practices helps in establishing and updating the tacit-knowledge which may be result of observation, imitation, and practice.

If during the evolution of an idea, the idea is continuously changing its initial main function or concept then agile practice – *Refactoring* could be of great use [22]. Refactoring means organizing the internal structure of the code without impacting the overall behavior of the code. Thus, refactoring would organize the new idea in such a way that the overall concept would remain intact.

Innovation Validation: Once the idea is generated, individual or team needs to consider whether it is practicable. They should analyze the idea and seek for approval. Once the idea is approved then a prototype of the idea should be generated. This generally means vetting the concept brainstormed by a product development team against the grand vision and values of the company or intention [20]. This stage requires that the organization on the whole should see the change as an ally rather than an enemy. The agile practice – *embracing change* allow an individual to welcome change.

Further, validating innovative idea before implementing it is based on the *test-first development* approach. This approach helps in ensuring the correctness of idea before implementing it. Alexander Graham Bell said, "*Great discoveries and improvements invariably involve the cooperation of many minds*". Thus, innovation is collaborative achievement of pushing and riding ideas into a concrete reality [26]. Scrum and XP practices support this concept of collaboration.

Prototype (concept) development: The prototype development helps in understanding customer needs and market feasibility. According to Khurana and Rosenthal [28], "the product concept is a preliminary identification of customer needs, market segments, competitive situations, business prospects, and alignment with existing business and technology plan". *Customer integration* is very important in this stage of innovation.

Prototype design and development generates many new ideas are *iteratively*. It is most important to integrate those ideas in concrete and stable model. Another problem related to integrating could be managing the complexity and interdependency of *part-whole relationships* where the whole is greater than sum of its parts [26]. The management of part-whole relationships is one of the greatest challenges faced by any complex organization. The solution to this problem is *Continuous integration* (an agile practice). It will integrate the specialized part to recreate the whole.

The prototype development would be an iterative process because it will incorporate the suggestions and feedback. For the success of an organization this process should be very fast, thus *Sprint* and *Daily Scrum meeting* would be very useful. Sprint would provide speed to prototype development and Daily Scrum meeting would provide transparency and early feedback to the process. *Simple design* practice would also be very useful because its target is to provide bare minimum requirements.

Development: From the experience gained from prototype production and trials, the development process will begin. In software projects this is the main phase where agile practices find their usage. Here many practices like *iterative development, continues integration, Sprint,* and *daily scrum meeting* used during prototype development will find their use. Some other practices like *code refactoring* and use of *cross-functional teams* would be useful. Use of cross-functional teams will promote the concept of dividing the labor amongst specialist who is best qualified to perform the unique tasks. Moreover, Scrum and XP practices support *sustainable-pace* and no over-time. This practice would motivate an individual and result in better development of product.

Production and Marketing: As production begins it will be necessary to take steps to ensure that a product or service finds adequate customers. Marketing is at least as important as production. Main

motive of marketing is to build strong customer relationship and satisfy the customer. Well-managed companies are not only close to their customers they search out and focus on their most demanding customers. XP and Scrum practices stresses a great deal on *customer involvement and feedback*.

Thus through this mapping I have proved that if agile practices are used properly; not only improve organization's innovation process but also motivates its employees to contribute in innovation process. Moreover, I believe that since now-a-days major companies are adopting these agile practices for software development projects; many software professionals are already well accustomed with these practices. I discussed this aspect with Rohit Garg, a senior software developer in Tata Consultancy Services. He mentioned that the environment where all team members interact amongst themselves and with customers, get to know the complete domain knowledge. He said his work involves working in close collaboration with cross-functional team; this motivates him to come up with new creative ideas. The responsibility of the product (or the code ownership) makes him more involved in the overall product development process. Amit Singh, a former software design architect in Mahindra British Telecom, noticed that with the use of agile practices he and his team were better able to meet changing customer requirements. They were not just getting frequent feedback from customers but were also able to implement those changes at a fast pace. He also mentioned that these agile practices have given him lot of creative freedom and better understanding of overall innovation process.

F. Case Studies

i. Case one: Collective ownership in Oxygen Media

Oxygen Media's Chairman and CEO, Geraldine (Gerry) Laybourne, is an innovator in the television industry largely responsible for the overwhelming success Nickelodeon achieved in the 1980s and 1990s [20]. She is a strong believer of the creative potential of her staff and value of tacit knowledge within an organization. She teamed up with Scrum and XP development teams in order to give new mission and spark innovation in the organization. Gerry believed that self-directed agile team will share more authority and responsibility while doing technical work. They will have broader idea of the product and will create new opportunities.

The agile team comprised of four developers, two Certified ScrumMasters (CSMs) and a manager. The XP practices used were pair programming, test driven development (TDD), and continuous integration. The agile team would brainstorm to explore opportunities across different departments (crossfunctional teams). They had thirty-day iterations (sprint), will meet for 30 minutes to prioritized backlog. Gerry herself will meet with representatives of the team and development management to get update on sprint health, non-development issues related to the project, and to pre-stage priorities for the next sprint.

Changes in innovation process: Gerry's experience with Scrum and XP practices helped in better understanding and enhancing the innovation process in Oxygen Media. The main focus of the organization shifted to developing deep mutual trust amongst employees. Scrum and XP practices gave upper management the tools to engage with team face to face and at a level of detail. It was realized that tapping the tacit knowledge of the front-line staff and middle management would be the most important source of innovation. Moreover, collective product ownership was dealt with new perspective. It was understood that the great product owners does not dominate, manage or control the development team but rather lead them. Judy and Beens [24] observed that in 3M and Toyota, product leaders inspired technical teams because they strongly believe that leveraging the talents of a large pool of experts is far more effective than trying to control the work.

Result: Gerry was able to establish a track record of consistently meeting or exceeding expectations in mission critical internal projects. Using Scrum/XP practices, Oxygen Media's development team completed a product release for early usability and focus group testing at 45% effort in eight team work weeks. Oxygen Media's senior leadership encouraged innovation and their team of software developers responded by using and expanding their agile practices.

ii. Case two: Intel's Silicon testing tool

According to Corrigan, Kelly, and Finglas [14], IT division in Intel wanted to create a "*Silicon Testing Tool*". The *two major challenges* they were facing was long development cycles and lack of flexibility to adapt to changing requirement. They were using systematic approach to analyze customer needs but in-order to create they needed fast development approach. They used following agile practices: cross-functional team, customer involvement, prototyping, Incremental Planning, sustainable pace, and Daily Scrum meetings.

The team was easily able to implement changing requirements with minimal impact to the product. If they would have used traditional development methods, the need for changes would not have become apparent until later in the development process. With the use of agile practices *transparency* was induced, agile team was able to implement changes as soon as they were requested. On the other hand, implementing changes at the very end is always a very tedious and time-consuming work. In that case, customer would not have been able to use the tool. With the use of agile practices the customer of Silicon testing tool were very satisfied and tool was a huge success.

Result: Satisfied customer, 20% faster completion of project, organization-wide collaboration among developers, testers, users, customers, and the project governance team. Earlier, Intel's innovation process was able to explore and develop innovative ideas but was not able to sustain it. They were not able to translate a great innovative idea into a product, tool or software. With the use of agile practices they found a mechanism to legitimize and sustain creativity throughout the innovation process. The result was fast development of good quality product.

iii. Case three: Honda's innovative urban car – Honda City

In 1978, Honda came up with a slogan "Let's gamble" [19], they were aiming at new-concept car. This is an example of one of earliest application of agile practices especially Scrum practices. Honda's goal was simple – they had to come up with new "break-through" product to regain their market strength. The management chose cross-functional teams and asked them to come with a new product idea in a specified time frame (*time boxing*). All the team members were experts in their respective fields. The team

comprised of young engineers and designers was formed. The team leader *Hiroo Watanabe* came up with the slogan "*Theory of Automobile Evolution*". The team was given all possible facilities, freedom and approximately two months of time. They had to come up with new concept for car taking into consideration to cost of the car. The communication and daily scrum meetings became an essential part of the process. The results were amazing; team came up with "*Tall Boy*" the new car with 1200 cc engine specially designed taking care of urban needs.

According to Nonaka [19], "The car inaugurated a whole new approach to design in the Japanese auto-industry based on the man-maximum, machine-minimum concept, which has led to the new generation of tall and short cars now quite prevalent in Japan." *Lessons learnt*: Innovation is an organization process but its core strength lie in the individual. A brilliant researcher has an insight that leads to a new patent. A middle manager's intuitive sense of market trends becomes the catalyst for an important new product development and a shop-floor worker draws on years of experience to come up with a new process innovation. The positive results of using agile practices are evident since Honda has become famous for the ability to respond quickly to customers, create new markets, rapidly develop new products, and dominant emergent technologies.

G. LIMITATION AND FUTURE RESEARCH WORK

Like any other approach XP and Scrum are not perfect. They share some common challenges. The impact of those challenges may vary between XP and Scrum. These challenges faced by XP and Scrum's practices prevent them to be used in enhancing the innovation process. Wood and Kleb [5] emphasize that we should be ready to deal with several *cultural conflicts* before implementing there practices.

1. Both approaches specify that team should be small (~10), so there are still doubts on how they are suited for large-organization where there are large teams, more complexities.

- 2. Traditional approach which highly focuses on *up-front* design. The traditional software development wants to start with well defined requirement gathering, design documents, and design. Both, XP and Scrum best suits to situation where requirement changes more often but when market and user requirement are very stable then use XP and Scrum would not be of much use.
- 3. Traditional approach which focuses on *big specifications*, where as XP and Scrum both emphasize on *incremental development (small releases)*.
- 4. Changes are always hard to implement. Beck [1] observed that "Really smart programmers sometimes have a hard time with XP". So, for average individual it can be hard to adopt XP and Scrum practices.
- 5. It is not just large organization where adopting new practices is a problem, even in small organization or small teams can have problems like interpersonal conflicts, cross-fertilization [5].
- 6. XP and Scrum has not been proven to work with systems that have *scalability issues*. Some research still needs to be done on how well new applications can integrate into existing systems.
- 7. The most important feature of both XP and Scrum is small, interactive, synchronized, and crossfunctional teams. It will be tough to implement XP and Scrum in an environment where programmers are separated geographically.
- Since, both XP and Scrum require constant feedback and customer involvement so according to Beck
 [1], "Another technology barrier is an environment where a long time is needed to gain feedback".

XP and Scrum both emphasize a great deal on testing. The definition of "*Done*" means fully tested and integrated component [10]. But sometimes an overloaded team compensates for the heavy workload by dropping some of its lower priorities. Iterations cannot be delayed; therefore they free developers to complete the higher priority goals within the time frame originally established. This problem is commonly faced by teams who are implementing XP or Scrum or both.

According to Boehm [13], "Real-world examples argue for and against agile methods". The use of agile practices has been proved to be critical technical success factor in the Internet browser battle between Microsoft and Netscape. Also, the use of these practices has been cited as the source of many software disasters, such as the \$3 billion overrun of the US Federal Aviation Administration's Advanced Automation System for *National Air Traffic Control*.

H. CONCLUSION

Scrum and XP practices has proved to be very useful in accelerating the product development lifecycle, increasing productivity, lower the development cost, and improving the overall quality of product. Scrum and XP being agile methodologies share many common features and rules. Despite being sharing so many common features they have some basic difference and one major difference is: Scrum's inclination toward organizational and management techniques and XP's inclination toward software development techniques. In this paper I have tried to represent the relationship between these two methodologies. Various companies have successfully made use of this combination of Scrum and XP like Primavera Systems, Trans Canada Pipelines systems, and Intel to name the few. Further, the mapping of agile practices (like customer integration, communication, tapping tacit knowledge of employees, and embracing change) and product innovation practices appears to be a potentially promising approach to reduce the risks of product failures, target resource spending more precisely, and respond to the challenges caused by changing customer requirements throughout industrial product innovation projects. In this research paper I have studied examples on how Honda, Oxygen Media, and Intel took advantage of this mapping. Use of agile practices will not only boost up individual's involvement in innovation process but also broaden the management involvement and understanding of innovation process. I know that management cannot ensure innovation success, but can definitely influence its odds. Similarly, a healthy interactive trustworthy working environment where individuals tacit is tapped and given right direction form the basis of innovative organization.

References

- 1. Beck, K. and Andres, C. 2004 *Extreme Programming Explained: Embrace Change (2nd Edition)*. Addison-Wesley Professional.
- 2. Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., May, J., and Kahkonen, T. 2004. *Agile Software Development in Large Organizations. Computer* 37, 12 (Dec. 2004), 26-34.
- 3. Laurie Williams, Guest Editor's *Introduction: The XP Programmer--The Few-Minutes Programmer*, IEEE Software, vol. 20, no. 3, pp. 16-20, May/June 2003.
- 4. Rasmusson, J. 2003. Introducing XP into Greenfield Projects: Lessons Learned. IEEE Software 20, 3 (May. 2003), 21-28.
- 5. Wood, W. A. and Kleb, W. L. 2003. *Exploring XP for Scientific Research. IEEE Softw.* 20, 3 (May. 2003), 30-36.
- 6. Ken Schwaber, Mike Beedle, Agile Software Development with Scrum, Prentice Hall, 2001.
- 7. Craig Larman , Agile and iterative development, Addison-Wesley. 2003
- 8. Jim Highsmith, Alistair Cockburn, *Agile Software Development: The Business of Innovation, Computer*, vol. 34, no. 9, pp. 120-122, September, 2001.
- 9. Takeuchi, Hirotaka, and Ikujiro Nonaka. 1986. *The new new product development game*. Harvard Business Review, January, 137–146.
- 10. Keith, Clinton. 2007. Scrum Rising. Session presented at Game Developers Conference, San Jose.
- 11. Kniberg, Henrik. 2009. Kanban vs Scrum how to make the best of both. Crisp, June, version 1.1.
- 12. Martin, Robert C.; Schwaber, Ken. *Best Practices in Scrum Project Management and XP Agile Software Development*. July, 2004.
- 13. Barry Boehm, Get Ready for Agile Methods, with Care, Computer, vol. 35, no. 1, pp. 64-69, January, 2002.
- 14. Tony Corrigan, Jim Kelly, and Keith Finglas. *Systemic Innovation Plus Extreme Programming Improves IT Software*. Intel Corporation. July 2006
- 15. Martin Fowler, *Refactoring: Improving the Design of Existing Code*. Dorling Kindersley Pvt Ltd, 2008.
- 16. http://en.wikipedia.org/wiki/Scrum_(development).
- 17. Kolawa, Adam; Huizinga, Dorota (2007). *Automated Defect Prevention: Best Practices in Software Management*. Wiley-IEEE Computer Society Press. p. 75. ISBN 0470042125.
- 18. http://www.mountaingoatsoftware.com/scrum
- 19. Ikujiro Nonaka, The KnowledgeCreating Company: How Japanese Companies Create the Dynamics of Innovation. New York: Oxford University Press, 1995.
- 20. Ken H. Judy, Ilio Krumins-Beens, "Using Agile Practices to Spark Innovation in a Small to Medium Sized Business," *Hawaii International Conference on System Sciences*, pp. 275b, 40th Annual Hawaii International Conference on System Sciences (HICSS'07), 2007.
- 21. Fruhling, A. McDonald, P. Dunbar, C., " A Case Study: Introducing eXtreme Programming in a US Government System Development Project," *Hawaii International Conference on System Sciences*, pp. 464-464, Proceedings of the 41st Annual (HICSS'08), 2008.
- 22. Siebra, C.A. Filho, M. Silva, F. Santos, A., "Deciphering extreme programming practices for innovation process management," Management of Innovation and Technology, 2008. ICMIT 2008., pp. 1292-1297, 4th IEEE International Conference, 2008.
- 23. Jeffrey Baumgartner, the Corporate Innovation Machine. http://www.jpb.com.
- 24. Judy, K.H. Krumins-Beens, I., "Ript: Innovation and Collective Product Ownership," AGILE, 2007, pp. 316-316, Washington, DC. 2007.
- Boehm, B. and Turner, R. 2005. Management Challenges to Implementing Agile Processes in Traditional Development Organizations. *IEEE Softw.* 22, 5 (Sep. 2005), 30-39. DOI= <u>http://dx.doi.org/10.1109/MS.2005.129</u>

- 26. A. H. Van de Ven, "Central problems in the management of innovation," *Management Science*, vol. 32, no. 5, pp. 590-607, 1986.
- 27. Beck, K., et al, "Principles behind the Agile Manifesto,", September 2006 < http://agilemanifesto.org/principles.html>.]
- 28. Khurana A, Rosenthal SR (1997) *Integrating the fuzzy front end* of the *new product development* process. MIT Sloan Mgmt. Review 25(4):35–49