

The value that XP could add though is unquestionable and many organizations that use Scrum adopt Test Driven Development, Refactoring, and Pair Programming as practices which improve quality, fasten the release process and minimize the need to revisit work due to technical debt [5] alongside shorter iterations. Some other important things which differentiate XP from Scrum are:

XP team operates in a strict priority order whereas a Scrum team may not necessarily tackle each item in priority order.

XP team bring new items of work into an iteration and shift out items of similar size if the customer decides on a new priority order.

Regarding similarities, the role of the Client in XP is very analogous to that of the Product Owner in Scrum; in which they support to write user stories, provides prioritization order and are always accessible to the developer. The daily stand-up meeting is mandatory in both XP and Scrum [5]. Both of them stresses the significance of co-location, only XP makes it deal-breaker. Hence, XP is superior from engineering and quality perspectives.

5.1 Issues and challenges

Our software industry is at a turning point with agile. For the first initial years of agile, people embracing it struggled to convince their organizations that it worked. Now, there is little question that agile development is a highly efficient way to build software. In fact, in 2008, a critical survey found that more than half of all software development teams surveyed were using Agile principles and agile has only grown since then [4].

The following are some of the limitations of agile process:

Not much support for third party vendors as the contract is not in a clear form.

The development of safety-critical software (as having critical and complex architecture) is challenging to enhance or change in an agile process due to limited support. Also, the cost of enhancements in these systems is very exponentially high.

Systems having complex architecture, the components are tightly coupled. It is very complicated and complex to construct software for such systems using agile practices [4]. However, this is debatable and there are agile frameworks available today in the market that targets Scaling Agile for the enterprise.

6. CONCLUSIONS

The development of Agile project involves several iterations. All agile methods can accept change request at any stage of development time. In this study, we have focused on the overall agile process including its practices, methods, challenges and issues related to it. The critics of Agile are mostly concerned about Scaling of Agile [7].

Agile does not scale up to the requirements of the systems build by larger enterprises. An another way of working is needed that applies the power of Agile, but also leverages the more extensive knowledge supplies of lean product development and systems thinking. The Scaled Agile Framework (SAFe) is one of the best such approaches [7]. SAFe provides freely revealed, online, comprehensive and detailed guidance for achieving the benefits of scale Agile development. It supports enterprises to deliver more efficiently and value continuously on a periodic and predictable schedule, making them more Agile in the marketplace and more competitive in their industry. It is a knowledge base of integrated and industry best practices proven patterns for enterprise scale Lean-Agile development. It is modular and scalable and allows each organization to apply it in a form that produces better business outcomes and happier more engaged employees. It synchronizes alignment, collaboration, and delivery for a large number of Agile teams. It supports both systems development and software, from the modest scale of well under 100 practitioners to the most superior software solutions. It supports three primary bodies of knowledge: Agile development, Lean product development, and systems thinking [7].

SAFe supports those building large, integrated solutions that typically require hundreds or more practitioners to develop and maintain. SAFe is improving business outcomes for government agencies and companies of all sizes across the globe, resulting in dramatic increases in employee engagement, improved economics, and workplaces that are more productive, engaging, and fun. Benefits from documented case studies include:

- 20-50% increase in productivity
- 30-75% faster time to market
- 50%+ defect reduction.

REFERENCES

- [1] Stellman A, Greene J. (2014). Learning Agile. O'Reilly Media, Inc.
- [2] Cohn M. (2009). Succeeding with agile, Addison-Wesley Professional.
- [3] Smartsheet. <https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban>, accessed on Feb. 20, 2018.
- [4] Amir Mkkakamnk. (2013). An appraisal of agile software development process. International Journal of Advanced Science & Technology 58: 20.
- [5] Bowes J. (2015). MANIFESTO. <https://manifesto.co.uk/kanban-vs-scrum-vs-xp-an-agile-comparison/>, accessed on Jan. 14, 2018.
- [6] Ponomareff D. "SlideShare," TORAK. <http://www.slideshare.net/dimka5/introducing-agile-scrum-xp-and-kanban>, accessed on Jan. 25, 2018.
- [7] SAFe. <http://www.scaledagileframework.com/>, accessed on Feb. 28, 2018.