Notes On Agile and Safety-Critical Development

Jakob Axelsson
SICS Swedish ICT
Box 1263, 16429 Kista
Sweden
jax@sics.se

Jaana Nyfjord
SICS Swedish ICT
Box 1263, 164 29 Kista
Sweden
jaana@sics.se

Efi Papatheocharous
SICS Swedish ICT
Box 1263, 164 29 Kista
Sweden
efi@sics.se

Martin Törngren
KTH Royal Institute of Technology
10044 Stockholm
Sweden
martint@kth.se

DOI: 10.1145/2894784.2894796
http://doi.acm.org/10.1145/2894784.2894796

ABSTRACT
Agile approaches have been highly influential to the software engineering practices in many organizations, and are increasingly being applied in larger companies, and for developing systems outside the pure software domain. To understand more about the current state of agile, its applications to safety-critical systems, and the consequences on innovation and large organizations, a seminar was organized in Stockholm in 2014. This paper gives an overview of the topics discussed at that seminar, a summary of the main results and suggestions for future work as input to a research agenda for agile development of safety-critical software.

Categories and Subject Descriptors
K.6.1 [**Project and People Management**]: Lifecycle, Systems Analysis and Design

General Terms
Management, Performance

Keywords
System, Process, Embedded Systems, Cyber Physical Systems

1. INTRODUCTION
This paper summarizes findings from a seminar on agile, safety and innovation organized in Stockholm on October 2, 2014 [2]. The main themes of the day were agile in large organizations; agile for safety-critical development; and innovation in agile teams. The context was software-intensive products, encompassing not only software but also elements of electronic hardware and mechanics. The seminar had 25 participants from 15 different companies and organizations, and consisted of 5 invited presentations, followed by group discussions on the topics of "Agile beyond software", "Agile and safety", and "Agile and innovation". We report on the results from the first two topics in this paper.

Scaling agile to large organizations is still an issue in many ways. Experience clearly indicates that many organizations are facing great difficulties in expanding agile throughout the organization effectively [1]. However, there are good examples from industry, emphasizing the need to tear down company walls, to pair business and technical people, and to involve key customers from key markets. In this work, we draw on those latter experiences and focus on particular aspects of scaling agile, which we believe will contribute to the current knowledge base in this area. More specifically, we look at the applicability of agile in domains outside pure software development. In brief, we found that applying agile for safety critical applications boils down to how to deal with the safety standards, and two approaches are possible: either agile has to be adapted to the requirements of the safety standards, or the safety standards have to be adapted to agile. These adaptations would be easier if safety standards were more goal-oriented, and less prescriptive on the methods to use. Safety-critical applications are often mechatronic in their nature, and to allow such development to be fully agile, requires better support in terms of tools allowing early validation, e.g. by automation, continuous integration, feedback, transparency and flow. It also requires focus on the technical platform for both hardware and software integration.

In the rest of this paper we provide a brief summary of the main results from the seminar discussions and conclude with suggestions for future work that will provide input to a research agenda for agile development of safety-critical systems.

2. AGILE BEYOND SOFTWARE
The group discussions focused on two issues: (a) reasons that agile have not come further in being adopted beyond software development, and (b) experiences from using agile in hardware development.

Regarding the state of agile beyond software it was concluded that hardware development is difficult to break down in iterations. Hardware development is mostly waterfall-based in principle. In addition, it was identified that the actual implementation phase can be more iterative, whereas other development phases including requirements specification, design and testing involved larger challenges for adopting agile.

Regarding the experiences, it was clear that Scrum seems to be the most popular method tried out in hardware development, at least for the implementation phase. Experiences of using other agile methods, such as XP, DSDM, Kanban were not identified among the participants in the discussion.

In addition, the group shared experiences of when they had tried to introduce agile ways of working in their companies or organizations. A number of challenges were identified, e.g.:

- Testing is a challenging issue: "System testing requires integration of hardware and software", "How can that be realized in more agile ways?" Two aspects are particularly challenging and were mentioned: First, the processes for software and hardware development are very different. Hence, process synchronization is problematic. Second, ownership of the testing system is an important issue. Hence, it must be clarified who owns the test system at any given moment of the process and in consequence who has the responsibility of the testing done efficiently and effectively.
- Barriers exist due to lack of harmonization of language and terminology: An example that came up and discussed briefly was whether "sprints" was a useful term in the context of mechanics development.
- Difficulties with organizing cross-functional teams: agile states that different competences are needed, which should also be applied in the forming of development team(s). Several issues were discussed: How do you organize your teams according to the agile practices in the context of product development, involving hardware and software development? In theory, everybody should know everything. However, in practice, it is better to have some experts that are responsible for certain parts, to ensure quality. Different roles mean people with different competencies. However, one participant raised the fact that the concept of cross-functional team also enables shared responsibility, which is important to avoid different risks arising from relying on single experts only. The group agreed that there needs to be a balance between shared responsibility and allocation of responsibilities to experts. Generally, all team members should have basic knowledge of the overall development activities, and that transparency is critical to ensure flow.
- Diverse lifecycles and time-spans: The lifecycle of hardware vs. software elements is different especially in terms of lifetime and maintenance processes, which also pose challenges that are not explicit in agile.

Several other questions were also raised among the group, especially regarding scaling agile to the entire organization, e.g.: "How can you be agile throughout an entire development organization?" and "How can functions outside R&D align with an agile R&D organization to get most benefit as a whole group?"

Some solutions were suggested during the seminar. For instance:

- One person suggested that Kanban can be a better method than Scrum for a hardware team that is working with many projects in parallel. By using a Kanban board, the team can keep track of several deliveries and plan and prioritize the order and flow of development.
- The entire group also agreed that integration points are important for enabling agile in the product development process. It was suggested, based on previous experiences, that architecture views for the entire product could help in identifying critical dependencies.
- Demonstrating working functionality with regular intervals has also shown to be useful as a way to work in more agile ways, mainly because they enable faster and more informative feedback.
- Everyone agreed that a key to success is also working communication and information structures.

In addition, the participants of the seminar focused on tools for enabling the introduction of agile beyond software development. Basically, the group identified that a major difference between software and hardware development is the possibility of changing things frequently. In hardware development, functionality is difficult to change once development has started. In software development, changes can generally be accommodated throughout the development process. Hence, it was agreed that an analysis of how to deal with change in hardware development is clearly needed. In the seminar, an example was identified, discussed and suggested as a potential starting point for further investigations of introducing more flexibility into hardware development.

The example can be described as "experimentation with cyclic hardware development". One of the participating companies had experimented with the concept of dividing the traditional sequence of hardware design and construction into cyclic development activities. First, using CAD to present a basic concept, and then dividing the construction of selected hardware parts into iterations. Thereby, they were able to build-in earlier feedback in the development process enabling them to change the design later on. Typically, they developed one single frame with revision only at the end just before the actual release, which generally restricts change due to high costs.

In terms of realizing the potential of this type of cyclic breakdown of hardware development, the group agreed that support tools are undeniably needed. An idea applied the agile value of simplicity and suggested starting in small steps. For instance, one participant had used 3D-printing to print a copy with similar core qualities as the actual product. By printing a prototype they could use prototyping to simulate certain development steps prior to actual development and thereby develop the hardware more iteratively.

In summary, the examples identified and discussed indicate that experiments are ongoing in industry showing that cyclic break-down of hardware could potentially be a way to introduce agile principles and practices in hardware development. The group agreed that with the right tools, more agile principles would be applied in product development. At least, tools would enable practical end-to-end development experiments and simulations that would contribute to knowing sooner whether things will work or not, thus saving valuable development time and cost. The capability of such tools would also support the testing and adaptation of various techniques, thereby accelerating the development process.

3. AGILE AND SAFETY

The discussion on the topic of agile and safety-critical development covered a range of perspectives, from "pure agile"

and implications on safety, to "pure safety standard approach" and constraints with agile. Generally, it was concluded that there is nothing per se preventing agile processes from being applied in the context of the development of safety critical products. However, the seminar engaged the participants with a mix of questions, concerns and suggestions.

In summary, the following aspects were primarily identified and discussed:

- Agile approaches are expected to benefit quality [1]; this makes it desirable to incorporate aspects of agile into safety approaches. Iterations also promote quality and focused increments should facilitate complexity management. While these practices promote quality, and should be relevant also for safety critical products, quality does not however necessarily mean the same thing as safety, therefore safety considerations must be infused into an agile approach.
- In addition, several agile techniques and practices, such as user involvement and continuous integration should be of direct benefit also for safety. However, due to lack of clear evidence it is suggested that agile practices are studied, tested and evaluated effectively to provide data about their applicability in safety-critical development.
- There are a number of agile methods and practices that are considered directly contradictory to common safety practices. For instance, agile does not prescribe thorough reviews conducted by external parties and does not explicitly require compliant documentation. However, new and more efficient approaches, tools and technologies (e.g. for the needs of documentation) should potentially resolve these tensions.
- The safety-related industry foresees an increasing need to shorten the time to market for their products. At the same time, there is a clear trend that more and more products nowadays are becoming safety-assured due to integration and increased level of sophisticated products made available as consumer products. The seminar concluded that goal-oriented safety standards leave room for adopting (more recent) development practices such as agile. This is difficult with the existing traditional strictly prescriptive safety standards. Tor Stålhane and Even-Andre Karlsson also presented practical examples of using agile methods in safety-critical development (refer to [2]). Basically, the need to emphasize software development, architecture and programming was mentioned, and also pointed out as a common denominator for achieving quality and safety.

The group discussion ended up with the following overall suggestion: *Method development and education initiatives need to be undertaken to bridge the gaps between safety and agile approaches*.

The following areas were listed as critical for further elaboration:

- Cultural integration is required to bridge the gap between safety and agile, and raise a sense of common thinking within teams. For instance, agile developers need to incorporate the "what can go wrong" perspective all the way from user stories to retrospectives.
- Architecture is seen as an important "meet in the middle" ground for bridging the existing gaps. Architecture is important to smoothly provide the preconditions for product integration (thus dealing with complexity and structuring the work), and at the same time provide safety patterns to mitigate hazards.
- Language and terminology vary between the two domains, so support for understanding across the domains is required.
- Goal-oriented safety standards are an important break-through for incorporation of evolving development practices in safety-critical development contexts. Integrating agile methods should be easier with the existence of such standards. Hence, in areas with quickly evolving technology, goal-oriented standards should be preferred.
- Close collaboration with assessors is highly recommended to understand their interpretations of the safety standards, and to facilitate method development and evaluation.

In summary, in education the gaps between agile and safety need to be emphasized and efforts towards their elimination should be made. Education efforts need to address multidisciplinary approaches incorporating safety and software development, i.e. connecting the why and the how (from hazards to design patterns). More specifically, software coding involves details that sometimes loose the direct connection to product properties. It is essential that e.g. embedded software developers maintain the connection to product properties such as safety, as one way to bridge the existing gap between agile and safety.

4. CONCLUSIONS

Many companies are placing high hopes on agile in improving their efficiency, and there is a large interest in applying it to other areas than pure software development, but that also leads to new challenges for both practitioners and researchers. In this paper we have analyzed agile in terms of its applicability in two domains; development beyond software and development of safety critical applications, respectively. A common theme in both of these perspectives is that they require more effort than just applying agile. For safety, compliance with the regulating standards is an outstanding requirement, and it is non-trivial to merge the world of agile and safety. For product development, and in going agile beyond software to mechatronic products, supplementary supporting tools and techniques are clearly needed.

Finally, it is worth noting that the seminar also covered a third topic, agile and innovation. In brief, the result showed that for innovation, the introduction of agile may actually lead to a decrease in product innovations, unless countered with additional activities focusing on longer term needs. Consequently, to fill existing process gaps we conclude that a holistic organizational perspective on agile is generally necessary, complementing the project teams with other activities for which different approaches may be more suitable. These results are all subject to further research and analysis, and thus we suggest that they are included in the research agenda for agile safety critical development.

5. ACKNOWLEDGMENTS

We thank the participants and speakers of the ICES Workshop, [2], for contributing ideas and perspectives regarding Agile and safety, some of which we have summarized in this paper.

6. REFERENCES

[1] Ambler, S. 2008. Agile in Practice – What Is Actually Going On Out There? URL: http://www.infoq.com/presentations/Agile-in-Practice-Scott-Ambler.

[2] ICES Workshop. 2014. Agile, Safety and Innovation – synergies and tensions when best practices meet in the development of software intensive systems., URL: http://www.ices.kth.se/events.aspx?pid=3&evtKeyId=752aa47809ce4885924e7525436e8f0c