Scrum and IEC 60880

Tor Stålhane
Norwegian University of Science and Technology
+47 73594484, stalhane@idi.ntnu.nol

Vikash Katta
Norwegian University of Science and Technology and OECD Halden Reactor Project,
+47 45464323, vikash.katta@hrp.no

Thor Myklebust
SINTEF ICT
+47 95779869, thor.myklebust@sintef.no

Abstract

Agile development has already proven to be a big success in several areas of application. It started in areas like web development but has now even moved into safety critical domains – e.g. air traffic management, automotive. Companies working with industrial automation – e.g. ABB – are considering using an agile development process. The main reason for this is that requirements changes are more frequent than before plus acceptance of the fact that requirements seldom are finished when the application development starts. To quote Daniel M. Barry "…it might be that the only solution is to identify requirements, to carry out a design sufficient to get a black-box description of the system, to identify and analyse hazards, and then to begin the lifecycle again with changed requirements".

NTNU – IDI has, together with SINTEF ICT, defined a process called Safe Scrum plus a process to handle the challenges posed by relevant standards. This process has already been applied to agile development using ISO 9001 and IEC 61508. For the proposed paper we will apply the same process to the standard IEC 60880 which is used in the nuclear power plant domain.

Important issues discussed in the proposed paper will be documentation, planning and proof of conformance. These three areas are important in the development of all software that shall be certified. In addition, they are the three areas where agile and plan-driven development is most different.

1. Introduction

Agile development is an idea, not a method. It is summed up in the agile manifesto as follows:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

The agile movement can be seen as a reaction to the strong focus on documents and plans that was, and in many cases, still is prevailing in software development organizations. Many developers saw this as a straight-jacket and that it focused on fulfilling the plans at the expense of satisfying the customers.

2. What is Agile development and what is Scrum

There is no such thing as “The agile development process”. People who implement the agile manifesto have met this challenge in different ways. Examples of processes are XP, Lean Software development and Scrum. Based on what we have seen in industry we will focus on Scrum since this method already has been implemented or is about to be implemented in several large companies – also in the safety critical area. The standard Scrum process is shown in figure 1. Avinor and Autronica already use Scrum in safety-critical application development while ABB is starting to consider it.

In addition, several companies are evaluating Scrum for possible, later implementation.

There are two main views on agile development - the believers’ view and the sceptics’ view. The following list is taken from a presentation by Geir K. Hanssen [1] and the items are summarized below.

- The believers:
  - Agile development is cheaper because:
    - Only what is needed will be developed
    - Misunderstandings and errors are discovered early
    - Communication is more efficient
  - Better conditions for creativity
  - Changes cannot be controlled. Thus, it is better to emphasize change responses and change control
  - Self-organizing groups perform better

- The sceptics
  - Customer attention is luxury
  - Customer will not accept ”no plan – no estimates”
  - Small releases will only fit small problems and small projects
  - Agile development does not fit in traditional project management framework
  - Compliance with important standards such as IEC 61508

The Scrum process is one way to realize an agile process. The process is described by a Scrum process, a set of artefacts and three roles - see [1].

- The Scrum process
  - The Sprint planning meeting – select requirements for the next sprint
  - Sprints – also called iterations, where the implementation and testing is done. A sprint most often takes one week to one month. The work in each sprint can be viewed as a mini-waterfall development project.
  - The daily Scrum meetings – what did we do yesterday, which problems did we encounter and what will we do today?
  - Sprint review meetings. What did we achieve in this sprint, showing real, working code?
  - Sprint retrospectives – what went well in the previous sprint and what should be improved? How shall we change the process to realize the improvements?
- Artefacts
  - The product backlog – the requirements that have been identified but not yet implemented
  - Sprint backlog – the requirements to be implemented in the coming sprint
  - The wall – a set of charts (e.g. the burn-down chart) showing the current project status.
- Roles

- Product owner – customer representative
- Scrum team – those who do detailed design and coding. The team typically consists of five to ten persons who work full time on the project
- Scrum master – “project manager”. His main jobs are to facilitate development and to remove impediments

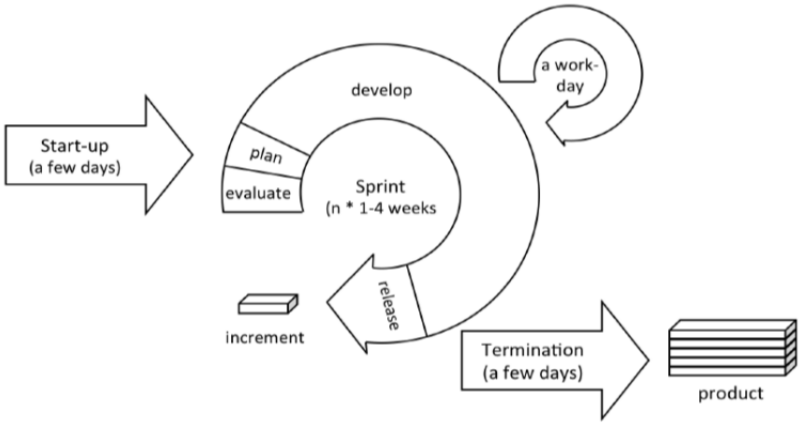

Figure 1: The standard Scrum process

3. Why should the Safety-Critical Industry consider Scrum

The part of the industry that develops safety-critical software has for a long time been plan-driven and methodically conservative. Several changes in the environment have, however, affected this:

- The tempo with which new technology is introduced in the marketplace. This holds both for new products (what we develop) and for new components (what we uses) – e.g. sensors.
- Increased focus on flexibility. This is partly a consequence of the first bullet point.
- There is a growing realization that the plan-driven development paradigm is too much focussed on writing and rewriting plans that are not used and on producing documents that are not read.
- The industry’s general focus on lean development and production. Whatever that does not contribute to the product’s final value should be removed.

There is no reason to believe that the tempo of inventions and innovations will slow down and those who do not follow will quickly get into trouble. In addition, more and more developers are using agile development. It remains to be seen if these programmers will be interested in working in a development environment based on plan- and document-driven development methods. To quote from our IEC 61508 paper [2], the key benefits that comes from this combination of a safety-oriented approach and a process model for agile software development are that the process enables

- Continuous feedback both to the customer, the development team and the independent test team.
- Re-planning, based on the most recent understanding of the requirements and the system under development.
- Mapping of functional and safety requirements.
- Code-requirements traceability.
- Coordination of work and responsibilities between the three key roles; the development team, the customer and the assessor.
- Test-driven development of safety critical systems.

All of these points will help us to get a more visible process and thus better control over the development process, which again will help us to deliver on time and within budget.

4. Challenges when using Scrum

First and foremost – Scrum is a software development method. As a consequence of this, we need to single out software development in a separate activity. This does not mean that the software should be developed in isolation from the rest of the project but that it should be organized as a separate activity.

The nuclear industry and any other industry that is dependent on safe operation of complex control systems will have one or more standards that shall help the industry in focussing on and achieving safe operation. These standards, however, mirror a traditional, plan- and document-driven view on software development. Changing these standards to also accommodate the agile development paradigm will take considerable time – e.g. five to ten years. There is also a real risk that the changes in software development paradigms will outrun the standards ability to change in order to incorporate such changes.

Thus, in order to start using an agile development method – in this case Scrum – we need to explore two options:

- Changes to Scrum – e.g. add-ons to cater to the traceability requirements
- Alternative interpretations of requirements in the applicable standards – e.g. what should be accepted as proof of conformance for an activity?

Both approaches are useful. The first and third author have used them both successfully in two cases – (1) Scrum and ISO 9001 [3] and (2) Scrum and IEC 61508 [2] – and we will use parts of both options later in this paper

5. A Method for Scrum adoption

We present a method for Scrum adopted to address the needs of development of safety system. This method was used in [2] and [3], it is simple and our experience so far is that it is highly efficient.

1. Collect a team containing a software expert, a domain expert and an assessor for the standard(s) under consideration
2. Identify all requirements in the standard related to software development
3. Go through all the requirements, asking the question “Will this requirement be fulfilled if we use Scrum?” This delegate each requirement to one of the following categories:
    a. Is fulfilled also if we use Scrum as is
    b. Is partly fulfilled if we use Scrum as is. Will need adding extra activities to the Scrum process
    c. Cannot be met if we use Scrum as is.
4. Use the two strategies identified in section 4 to sort out the problems – con-compliances

This approach leaves us with two challenges – (1) have we identified all relevant requirements and (2) different assessors have different opinions of what should count as proof of conformance. Especially challenge (2) is problematic since it has no final solution. One possible way out of this is to involve the assessor from day one and ask questions such as “If we use approach X here, will this be accepted?” This approach must, however, be used with care so that we do not hold the

assessor hostage to out choice of development process. If we need to ensure assessor independency, we can use one assessor as a “sparring partner” during the project and another one – preferably from the same organization – for certification.

6. IEC 60880 and Scrum

6.1 Relevant standard requirements

We have taken sections 5 to 10 of IEC 60880 [4] as our main starting point. In addition, we have consulted IEC 62138 [5] and tables 2 and 3, section 1.11 in the document “Licensing of safety critical software for nuclear reactors” for guidance. From the diagram below, taken from IEC 60880, we see that the software implementation only concern a small part of the total process. It is only this part that is touched by Safe Scrum, the rest is Scrum independent.

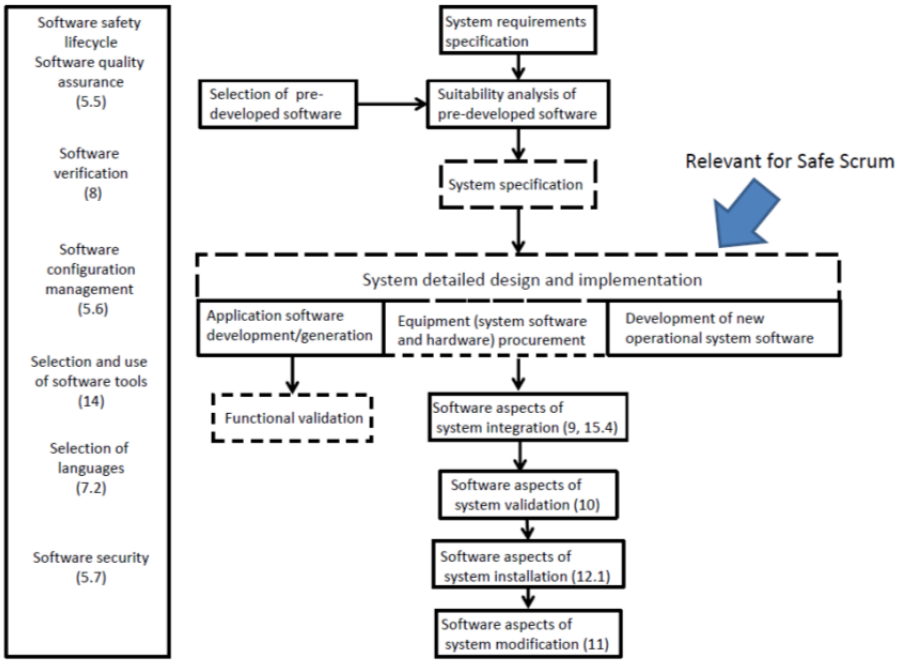

Figure 2: Activities in the system safety life cycle

We studied sections 5 to 10 of IEC 60880 in details. Based on the requirements stated in these sections, we selected the following areas for a closer scrutiny:

- 5.3 – Software development approach. This section has no references to processes or procedures and is thus by default, also applicable to Scrum
- 5.4 – Software project management. In this section, the standard states that the development process may be iterative, provided that certain requirements in clause 6 in IEC 61513 [8] are fulfilled. This part of IEC 61513 makes requirements to the system safety lifecycle, which have to be fulfilled outside Scrum. In addition, this section in sub-section 5.4.9 states that each phase should generate a set of documents according to annex F in this standard.
- 5.5 – Software quality assurance plan, which also include security assurance and safety assurance. This section says that there should exist a quality assurance plan. This plan may, however, be “adapted for individual product phases or particular software components…provided the principles defined in this standard are addressed “, and that “Any

deviation from the requirements of this standard and its normative annexes shall be identified and justified". Thus, it is up to the assessor what he is willing to accept. It is practical to have a company QA plan which can be reused, in whole or in parts, from one project to the next. In addition, other company specific standards will influence our solution.

- 7 – Design and implementation. The most important provisions in this section are that there should be (1) a program structure based on decompositions, that this structure should be simple to understand, (3) that a top down approach should be preferred to a bottom-up solution and that (4) a conceptual model of the software architecture should be adopted at the beginning of each software project. In addition, the two clauses 7.1.2 and 7.1.3 are discussed in some more details below – see sections 6.2 and 7.2.

Since Scrum and all other agile methods are specifically made to handle problems related to the specification and changing of requirements section 6 is an important part of the standard when we want to adapt to agile development. We have thus selected 6.1 – Specification of software requirements – for a closer look.

As a consequence of the reference to annex F in IEC 60880, we also include sections 8.2.2 and 8.2.3 for a closer look.

6.2 A closer look

We have singled out the following sections of the standard for a closer look:

- 5.4 – Software project management
  - 9 requirements are OK
  - 2 requirements need a closer look – 5.4.9, which invokes annex F and 5.4.10.
- 6.1 – Specification of software requirements.
  - 13 requirements are OK
  - 2 requirements need a closer look – 6.1.4 and 6.1.5, which invokes annex A
- 7.1.2 – Implementation of new software in general-purpose languages
  - 4 requirements are OK
  - 1 requirement needs a closer look – 7.1.2.5, which invokes annex B
- 7.1.3 – Implementation of new software in application-oriented languages
  - 4 requirements are OK
  - 0 needs a closer look
- 8.2.2 – Design verification
  - 7 requirements are OK
  - 0 needs a closer look
- 8.2.3 – Implementation verification
  - 11 requirements are OK
  - 1 requirement needs a closer look – the intro, which invokes table E.4.2

This gives us a To-Do list of six requirements. The rest – 48 requirements – do not need any special treatment or consideration when we use Scrum – 11%. To put these numbers into perspective, we had to have a closer look on 15 of 183 requirements when assessing Scrum for IEC 61508 – 8%. We found no requirements in the standard that could definitively not be fit into the Scrum process.

7. A workable solution

7.1 Safe Scrum

We have observed that the safety requirements are quite stable, while the functional requirements can change considerably over time. The most important sources of changes for safety requirements are changes in relevant standards, which happen only seldom, and the discovery of new hazards during RAMS (Reliability, Availability, Maintenance and Safety) validation. This is taken care of in Safe Scrum with the possibility for revising the backlog after RAMS validation – see figure 3 below.

Development with a high probability of changes to requirements will favour an agile approach. Usually, each *backlog item* also indicates the estimated amount of resources needed to complete the item – for instance the number of developer work hours. These estimates can be developed using simple group-based techniques like 'planning poker', which is a popularized version of wideband-Delphi [6].

All the risk and safety analyses on the system level are done outside the Safe Scrum process, including the analysis needed to decide the safety level. Software is considered during the initial risk analysis and all the later analysis – on per iteration. Just as for testing, safety analysis also improves when it is done iteratively and for small increments – see [7].

Due to the focus on safety requirements, we propose to use two product backlogs, one *functional product backlog*, which is typical for Scrum projects, and one *safety product backlog*, which is used to handle safety requirements. Adding a second backlog is an extension of the original Scrum process and is needed to separate the frequently changed functional requirements from the more stable safety requirements. With two backlogs we can keep track of how each item in the functional product backlog relates to the items in the safety product backlog, i.e. which safety requirements that are affected by which functional requirements. This can be done by using simple cross-references in the two backlogs and can also be supported with an explanation of how the requirements are related if this is needed to fully understand a requirement.

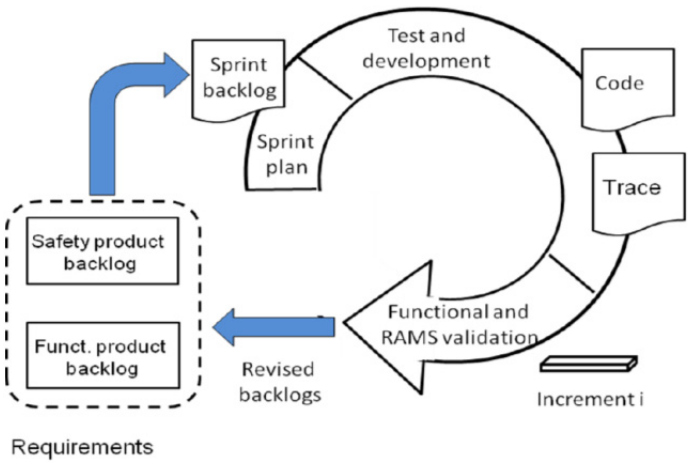

Figure 3: Safe Scrum process

7.2 Scrum adaptations

- 5.4.9 – Annex F in IEC 60880, lists a recommended set of documents to be generated at the end of each phase. Since each sprint is a miniature V-model, each sprint will contain activities from several phases. The relevant points in the annex F table are 8.2.2 and 8.2.3 – both related to verification. Each phase will need a set of documents generated at the end of a phase to prove that the activities are done according to the standard. It is up to the assessor what he will accept as proof of conformance. This thus needs to be discussed and agreed upon at the start of the project. Note that these documents will have to be (partly) rewritten when one or more requirements changes. The three documents needed at the end of each sprint are
  - Software test specification document. Given the dynamic nature of requirements handling in Scrum, this report has to be written when handling each requirement, based on the tests designed for this requirement.
  - Software code verification report – written at the end of each sprint
  - Software test report – written at the end of each sprint.
- 5.4.10 – "Each phase shall be systematically terminated by a review..." Based on the way a project phase is described in section 5.4, we will consider the end of a sprint as the end of a set of phases. Scrum already has a review at the end of each sprint but this may need to be extended in order to meet the assessor's requirements.
- 6.1.4 – "...the process of laying down software requirements shall be rigorous". It is up to the assessor to decide what he will accept as rigorous. As a minimum the organization needs to propose a definition which should be company-wide, not project specific. It is common in Scrum to elaborate the requirements when they are taken out of the sprint backlog. This will, however, not be sufficient in our case and we must adapt Scrum as follows: all requirements must be rigorously defined when they are
  - Inserted into the backlog
  - Taken out of the sprint backlog
  - Revised and reinserted into the backlog
- 6.1.5 – Annex A is related to the software safety life cycle. This annex describes how to handle requirements. Part of this – e.g. A.2.1: Description of constraints between hardware and software – is handled outside Safe Scrum, while other parts – e.g. A.2.2: Self-supervision – is taken care of by requirements in the safety product backlog.
- 7.1.2.5 – Annex B is related to requirements handling. The first part handles the design process, which is outside Scrum – often called Scrum iteration 0. Part 2 handles the software structure, which should be part of the coding standard and will not affect the choice of development process. The same holds for part 3 – Self-supervision and part 5 – Language dependent recommendations.

  Part 4 is about subroutines and goes a long way towards recommending test-driven development, albeit without actually using this term. The important challenge is found in B4gc which requires that "A formal description of the test inputs and results (test protocol) should be produced." This is an extension of the common way of doing testing in Scrum and we thus need to insert this into the develop-and-test part of the development process.
- 8.2.3 – Table E.4.2: Testing methods. This table describes types of tests that should be performed – e.g. path testing, data movement testing and timing testing – and should thus be included into a test procedure description.

8. Threats to validity

When discussion the relevance of our conclusions, four things are important – have we understood certification, the standard, have we touched all relevant parts and have we understood agile development in general and especially Scrum? We will briefly discuss each of these requirements below.

- Have we understood the right way to do certification? One of the authors has been working with safety certification of safety-critical systems for a long time. His experience and insight give confidence that our discussions and conclusions are sound.
- Have we understood IEC 60880? One of the authors has been working in the nuclear industry for a long time and knows the relevant standards and how they are used in nuclear-related software development. Thus, this part is OK.
- Have we touched all relevant items? We have scrutinized the standard and identified all parts where the term “software” has been used. The requirements handled in sections 6.3 and 7.3 thus cover all relevant items.
- Have we understood agile development – especially Scrum? We have already done extensive research on the application of Scrum in two other standards – ISO 9001 and IEC 61508. IEC 60880 contains no problems that have not already been discussed in relation to the two previous standards.
- Our claim to validity is based on the three preceding bullet points. Based on the discussion above, we are confident that the adaptation described in section 7 will make it possible to use Scrum as a development process and still be IEC 60880 compliant.

9. Conclusions and Further work

First and foremost; while there are problems with the application of Scrum-as-is together with IEC and IEC 60880 – e.g. traceability – these and all other problems identified in section 6 above are taken care of when using our Safe Scrum.

In addition – the assessor need to be involved from day one of the development. We have already identified several sections in relevant parts of the standard where the standard itself leaves the requirements open to interpretation. It is the development organization’s duty to make a relevant interpretation of each requirement in the standard but it will ease the final certification process tremendously if these interpretations are accepted by the assessor before development starts.

As should be expected, the majority of Scrum problems handled by Safe Scrum are related to the handling of software requirements and to testing – test specifications, the test methods used and the final test report for each requirement.

The next step in this work is to try and identify a small but real project in the nuclear industry where we can try Safe Scrum in a real environment. As always, the proof of the pudding is in the eating.

10. References

[1] Hanssen, G.K.: Agile software development and safety critical systems. Presentation at Autronica, January 9, 2013.

[2] Stålhane, T., Myklebust, T., and Hanssen, G.K.: The application of Safe Scrum to IEC 61508 certifiable software. ESREL 2012, Helsinki, Finland, June 2012

[3] Stålhane, T. and Hanssen, G.K.: The application of ISO 9001 to agile software development. PROFES08, Monte Porzio Catone, Italy, June 2008

[4] CEI/IEC: IEC 60880 Nuclear power plants – Instrumentation and control systems important to safety – Software aspects for computer-based systems performing category A functions. Second edition, 2006

[5] CEI/IEC: IEC 62138 Nuclear power plants – Instrumentation and control important for safety – Software aspects for computer-based systems performing category B or C functions. First edition, 2004

[6] B. Boehm: Software Engineering Economics. Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1981, ISBN 0-13-822122-7

[7] R. Morsicano, R. and Shoemaker B.: Tutorial: Agile Methods in Regulated and Safety-Critical Environments. ShoeBar Associates

[8] CEI/IEC: Nuclear Power Plans. Instrumentation and control important to safety. General requirements for systems. Second edition, 2011.