Safety Stories – A New Concept in Agile Development

Thor Myklebust, Tor Stålhane

Safety Stories – A New Concept in Agile Development

Thor Myklebust,
SINTEF ICT
Trondheim, Norway
thor.myklebust@sintef.no

Tor Stålhane,
NTNU IDI
Trondheim Norway
stalhane@idi.ntnu.no

Abstract - Safety stories is a new practice, developed to ensure that the agile requirements management process encompass the important safety requirements together with required measurements and techniques. The safety story card will create a common problem understanding between the software developers and the safety stakeholders. Safety stories are user stories that include one or more safety requirements. If you do not get the safety right it does not matter how well you execute the rest of the project.

Keywords-user stories; safety stories; IEC 61508, communication; assessor

I. INTRODUCTION

The term "user stories" were introduced by K. Beck in the XP-1999 conference [1]. A user story is a short, simple description of a feature, told from the problem owner's perspective. The essence of user-centric requirements elicitation is to focus on what the user wants or needs and not on what the system shall do. User stories are intended to shift the focus from writing about features to discussing them. The discussions are often more important than the written text. It is common practice to also document the acceptance criteria together with each user story.

A typical user story follows the following pattern:

As a <type of user> **I want** <to perform some tasks> **so that I** <can achieve some goal, benefit or value>

While the user stories usually are simple, the acceptance criteria are often more complex. Unfortunately, there is no agreed-upon common format for the acceptance criteria. Consider the following user story and acceptance criteria example:

As a <customer> **I want** <to withdraw cash from an ATM> **so that I** <don't have to wait in line in the bank>

Acceptance criteria: Given that the account is creditworthy and that the card is valid and the dispenser contains cash: When the customer requests the cash, then ensure that the account is debited, and ensure that cash is dispensed and ensure that the card is returned.

II. THE SAFETY EPIC

An epic is a high-level user story. It must be refined and split into several user stories before it can be realized. The epics are important for two reasons – they define the high-level requirements of the system to be developed and it identifies the system's environment. A typical safety epic could be

To satisfy <the overall safety needs> **the system must** <always be able to reach a safe state>

To keep <the operators> **safe, the system must** <stop the robot when someone enters the tool cell>

Through having safety epics and discussing them right from the start we make sure that safety is on the agenda throughout the project and prevent it from being an add-on considered too late.

III. THE SAFETY STORY

We would like to keep the patterns of the safety stories close to the user story pattern. It is, however, important to bear in mind that safety requirements – and thus, safety stories – have two sources (1) standard requirements and (2) system hazard analyses. A final challenge when it comes to safety stories is that they often will be linked to a functional user story, either directly or indirectly.

The safety requirements introduced by the applicable standard can be formulated as follows:

To satisfy <a safety standard requirement> **the system must** <achieve or avoid something>

It is also possible to describe process-related requirements in this way, but the value of this is doubtful since a user story is supposed to be taken from the backlog and then be finished. A process requirement, on the other, hand, should be on the team's agenda throughout the project. The safety requirements that are not related to the standard should be related to either a user story or to the results of a safety analysis – e.g. a HazId or an FMEA (Failure Mode and Effect Analysis). The FMEA may also include an evaluation of the diagnosis. It is then named FMEDA (Failure Mode and Effect Diagnostic Analysis). The result of a safety analysis is usually either a requirement change, an architectural change or the introduction of a barrier – e.g. a watchdog. The safety story should refer to the analysis that introduced it. This is achieved by linking the safety story to the system, subsystem or function FMEA or FMEDA – e.g.:

To keep <function> **safe, the system must** <achieve or avoid something>

Note that linking safety needs to software design is already required by the IEC 61508:2010, part 3.

IV. WHY SAFETY STORIES AND SAFETY EPICS

User stories are written throughout an agile project. A user-story workshop is often arranged at the start of the project. Everyone in the team should participate and the goal is to produce a product backlog that fully describes the current understanding of the functionality to be delivered. As a lot of things in agile development, creating user stories is also part of achieving efficient communication in the project. Thus, who writes the user story is far less important than who is involved in the discussion.

Even though the product backlog is in some senses the equivalent of a requirements document in a non-agile project, it is important to bear in mind that a user story or safety story is not finished before it is discussed both within the team and with the customer representatives.

Writing safety stories will have several benefits. The most important ones are:

- It will create discussions about how to realize the safety stories, both when writing them, during the sprint backlog refinement discussions and when they are to be implemented. Discussions lead to communication and a common metal model of how to realize the story which will improve system quality.
- The safety stories and the improved communication will help to create a safety culture in the team.
- We will discuss how to realize the safety story when it is taken out of the backlog. Thus, our decision will be based on more information that if we had decided how to realize it at the start of the project.
- Since the safety stories are about "what" and not about "how", they are a good starting point for discussions with the assessor or the safety responsible person. E.g., will the assessor accept that we solve this safety story by implanting this barrier?

V. CONCLUSION

The introduction of safety stories into agile development will enable us to involve developers, assessors and customers into realizing the system's safety requirements, help to build a safety culture and base our safety-related decisions on all currently available information.

ACKNOWLEDGMENT

This work was partially funded by the Norwegian Research Council under grant #228431 (the SUSS project) and SINTEF SEP project Safe Software.

REFERENCES

[1] K. Beck. Extreme programming explained: embrace change Page 190 Publisher: Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA ©2000 ISBN0-201-61641-6