

Towards an Agile Requirements Engineering Process combining HERMES 5 and SCRUM

Birgit Schär
INVERSUM GmbH
Solothurn, Switzerland
birgit.schaer@inversum.ch

Stephan Jüngling, Barbara Thönssen
University of Applied Sciences Northwestern Switzerland
Olten, Switzerland
stephan.jüngling@fhnw.ch, barbara.thönssen@fhnw.ch

Abstract – The Swiss Federal Administration uses for its IT projects primarily the project management method HERMES 5, which has originally a sequential approach. With the introduction of the agile scenario ‘customized IT application (agile)’ the Federal IT Steering Unit answered on the increasing demand for agile software development with SCRUM. HERMES 5 and SCRUM are different methods and there are general as well as topic-specific challenges involved in combining them. The focus of this research is on the requirements engineering discipline and investigates how the two methods HERMES 5 and SCRUM can be combined. As such, it provides an overview of discrepancies between the two frameworks that have to be resolved by many IT projects that want to take advantage of both methods and presents some solutions that potentially could be standardized and incorporated into upcoming HERMES releases.

Keywords – requirements engineering; agility; HERMES 5; SCRUM

I. INTRODUCTION

HERMES 5 is the project management method for computer science, service and business organizations developed by the Swiss Federal Administration. The method is an open standard provided by eCH [1]. In its core HERMES 5 is a waterfall project management model. Since business changes faster nowadays, it is increasingly inappropriate to analyze the requirements at the beginning of a project followed by sometimes several years of development. Instead agile project management takes the challenge, embraces continuous improvement, scope flexibility and on-going stakeholder input to develop products that meet up-to-date requirements.

With the release of HERMES 5 in April 2013 the Federal IT Steering Unit (FITSU) introduced a new scenario called ‘customized IT application (agile)’ to react to the increasing demand for agile software development with SCRUM in the Swiss Federal Administration. Agile software development is not new, but it has taken time to become recognized as an effective method [2]. Since the beginning of the 21st century agile software development methods have gained increasing attention. The history of agile software development has its beginnings in the first half of the 20th century with the development of the PDSA- (Plan, Do, Study, Act) lifecycle by

Walter Shewhart, which became the basis of iterative and incremental software development methods [3]. The milestone for iterative and incremental software development was set 20 years later in 1950 with the project X-15 hypersonic jet [4]. After 1950, various projects in the military and aerospace areas used an iterative and incremental software development approach [5]. Much later, in 2001, the Agile Manifesto for software development was written, which defines the basic values and principles for all agile software development methods [6].

Nowadays, it is common knowledge that IT projects have an advantage, if they apply agile software development methods. But what exactly does agility mean? Derived from different definitions, as for example given by the Agile Alliance [7] and D.S. Alberts and T.M. Takai [8] we defined the term ‘agility’ in the context of IT project management as ‘the ability to react fast and with reasonable effort to changing requirements by splitting tasks into short pieces of work that allow the project team to adjust the work effort according to the new project insights and reassess and reschedule the tasks accordingly to prevent implementation of inadequate functionality and high project costs’.

In HERMES, the new scenario ‘customized IT application (agile)’ is officially available for the application in the Swiss Federal Administration since 2013. In June 2014 the FITSU published the new release HERMES 5.1 [27]. Practical experience showed that applying the scenario ‘customized IT application (agile)’ in projects raises many questions from the perspective of requirements engineering. These questions are addressed in the present work.

The paper is structured as follows: In section II we introduce the application scenario of the Swiss Federal Administration. In section III we give an overview on related work that deals with the combination of HERMES 5 and SCRUM and with agile and waterfall methods in general. In section IV we introduce the necessary basics of HERMES 5 and SCRUM. Section V contains the description of the problem. Our research methodology is described in section VI. Section VII details our approach including first evaluation results and our conclusion is given in section VIII.

II. APPLICATION SCENARIO

Traditionally, the Swiss Federal Administration uses the project management method HERMES 5 for its IT projects. For the software development part it increasingly uses SCRUM as an agile development framework. The use of SCRUM promises a fast value delivery to the customer and the developed product reaches high quality in short time by the execution of various sprints in which the customer can give continuous feedback.

The development of individual business software in the Swiss Federal Administration is mostly a big and complex undertaking. These kinds of IT projects are often influenced by political and legal factors and by constraints from other systems and running projects. Additionally the investigated business area is often very specific and unique. Unfortunately, these factors often lead to frequently changing business requirements and in worst case to a shift in scope. Agile software development deals with changes during the software development process, but it is equally important to deal with changing business requirements already during requirements engineering. A changing business requirement doesn't only refer to itself; it can also have a significant impact on other areas. Therefore, from the business point of view it is important to work and act agile from the beginning to the end of a project.

The ideas and concepts of HERMES 5 and SCRUM are different, and hence there are challenges involved in combining them. The agile scenario 'customized IT application (agile)' incorporates this combination into HERMES on a highly theoretical level by the insertion of a module called 'Agile Development'. But the interaction of the module 'Agile Development' with other modules and especially with the module 'IT System' that contains the core of the requirements engineering activities is not described in detail. Additionally a deeper look into the module 'IT System' reveals that HERMES 5 doesn't give any advice on how the requirements engineering is handled.

The Swiss Federal Administration has a diverse project portfolio. There are IT projects implemented in a single federal office, IT projects that span multiple federal offices or an entire federal department. Some IT projects are even implemented across different federal departments. The main problem is that there is no common solution that defines how to deal with mutual interactions between requirements engineering and software development. Every IT project develops its own solution and stumbles across the same hurdles that arise from the application of the agile development module in the context of the given HERMES framework.

III. RELATED WORK

Until today there is no study that answers the question how HERMES 5 and SCRUM could be combined in the field of requirements engineering in an enterprise scope. There is one study from the FITSU called 'HERMES and Agility' [9] that investigated the weaknesses and uncertainties that arise during the interaction between software development and project management if SCRUM and HERMES are used in combination. However, the study 'HERMES and Agility' is

outdated, because it refers to the old version HERMES 2003/2005. A recent study that takes HERMES 5 into account doesn't exist until today.

Recent studies compare other frameworks with SCRUM. For example the case study from R. Noordeloos, Ch. Manteli and H. van Vliet [10] that presents the results of two offshore projects that change from RUP to agile with SCRUM. The study shows that the benefit of SCRUM is the involvement of the customers and the developers during requirements engineering which creates a close communication and common understanding within the team. G.J. Rankins and M. Kearns compare in their article PRINCE 2 and SCRUM [11]. PRINCE2 as project management method is comparable to HERMES 5, although a bit more formal and the occurrence of changes lead to an extension of a phase with the consequence of higher effort and longer duration. In SCRUM a change never leads to an extension of the sprint. The communication in SCRUM is seen as benefit as well.

The challenge of combining waterfall and agile methods is well known in practice. In 2015 SwissQ Consulting AG analyzed the trends and benchmarks for software development in Switzerland. They interviewed 450 persons from different enterprises and different business sectors in Switzerland. The final report shows that 41% of the projects use an agile and 26% a waterfall approach as project methodology. From all waterfall approaches, 22.5% use HERMES 5, whereas half of them in combination with SCRUM. One of the main findings of the study is that requirements engineering is equally important in all project methods. But unfortunately there is a common misunderstanding that in agile projects the Product Owner role can handle it on its own. [12]

Also the International Requirements Engineering Board (IREB) focused on the question how requirements engineering changes in agile projects. It conducted a study 'Requirements Engineering and Agile Development' that investigates how the discipline of requirements engineering can be adapted to better support an agile project approach. The findings are that requirements engineering activities remain the same but in agile projects they are executed continuously. In contrary to a waterfall approach the documentation of the requirements is more dynamic and can change from time to time. [13]

In addition there exist already several books (e.g. from Christ Rupp and the SOPHISTS [14] and Johannes Bergsmann [27]) that explain the techniques and concepts of requirements engineering in an agile environment. But like already mentioned in the IREB study the requirements engineering activities itself remain the same.

M. Steven Palmquist et. al. [15] analyzed the differences and similarities between agile and waterfall methods. The study showed that the goal, the basic building blocks (scope, costs, schedule, performance) and the development building blocks (analyze, design, build, test, deploy) are the same in both methods. The difference is how the building blocks are used. With the focus on the basic building blocks, agile methods vary the scope; the waterfall methods vary the costs, schedule or performance. With the focus on the development building blocks, agile methods work through the single blocks iteratively whereas the waterfall methods go once through each

block sequentially. Furthermore the study points out that agile and waterfall methods often use different terms.

Another issue referred in literature, is the question how the agile concept can be scaled to the enterprise level. Alex Brown and Jeff Sutherland from SCRUM Inc. developed an extension to SCRUM called ‘SCRUM at Scale’, treating this topic [16]. Another well-known framework for enterprise level is the ‘Scaled Agile Framework (SAFe)’ from Dean Leffingwell [17]. The SAFe has three application layers, the portfolio layer, the program layer and the team layer. On portfolio layer the different programs are aligned with enterprise strategy and managed in a portfolio-backlog. On the program layer the topic for one program is broken down into features, which are taken as input for the release planning. The release planning is mutually agreed by all teams, to resolve the dependencies between the features that are all managed in the program-backlog. After the release planning every team has a team-backlog, which defines the tasks for each team.

The available studies in literature give a good insight into the different facets of the topic and shows that a lot of research has been done already. However, the practical application of combining agile and waterfall methods as well as the combination of requirements engineering and software development has not been addressed. In order to point out the discrepancies and potential solutions for the current uncertainties, we first give a short introduction to the project management and software development method and the relevant details focusing on the requirements engineering process.

IV. BASICS OF THE METHODS

This section gives a short introduction to the basics of SCRUM and HERMES 5.

A. Introduction to SCRUM

1) Driving Forces and Definition of Terms

SCRUM was invented by Ken Schwaber and Jeff Sutherland. Both worked independently of each other on a new method to develop software faster. The procedures and models they worked out were very similar and consequently, they aligned with each other. In 1999, together with other authors, they published the article ‘SCRUM: An extension pattern language for hyper productive software development [20]’. Sutherland and Schwaber, as well other authors, subsequently continued to publish many success stories about SCRUM, and today SCRUM is used in different fields even beyond software development [21].

SCRUM.org defines SCRUM as “a management and control process that cuts through complexity to focus on building software that meets business needs.”[14].

2) SCRUM Essentials

In the IT community, the term ‘SCRUM’ is very well known and the SCRUM guide [28] from J. Sutherland and K. Schwaber gives a good overview. For the scope of the current work, we recapture briefly the important terms, which are

necessary to address the scope of the scenario ‘customized IT application (agile)’:

- **Values and Principles:** SCRUM is based on the values and principles of the Agile Manifesto. In addition to the values of the Agile Manifesto, SCRUM has its own foundation of five values (commitment/forecast, focus, openness, respect and courage).
- **Roles:** SCRUM only has three roles. The SCRUM Master, the Product Owner and the Team. All-in-all the most of the activities in the SCRUM Process are executed by the whole SCRUM Team.
- **Process:** The SCRUM process follows an iterative and incremental approach. The precondition for initiating the SCRUM process is conceptual product development based on a vision. The SCRUM Process itself consists of the following main activities ‘Sprint Planning 1’, ‘Sprint Planning 2’, ‘Sprint’, ‘Review’ and ‘Retrospective’.

3) Requirements Engineering in SCRUM

SCRUM specifies the system requirements iteratively. Requirements engineering happens in every sprint through close collaboration between the Product Owner and the Team. Although requirements engineering is done continuously, SCRUM doesn’t define any technique that should be used for elicitation, validation or management of the requirements. The product backlog is the central instrument for managing the backlog items, which correspond to a requirement, a function, a feature, enhancements etc.[28]. SCRUM doesn’t define the form of a single backlog item.

A lot of studies recommend the use of epics and user stories. Bergsmann [26] refers to different agile requirements engineering methods (behavior driven development, specification by example and test driven development) in order to achieve a fast and easy specification of the requirements. In all these methods the necessary artifacts are drivers for software development, and can be seen as requirements.

B. Introduction to HERMES 5

1) Driving Forces and Definition of Terms

In 1970 the Swiss federal administration started with the development of its own project management method for their projects. Five years later, in 1975, the first HERMES version was published. The name HERMES stands for ‘Handbuch der Elektronischen Rechenzentren des Bundes, eine Methode für die Entwicklung von Systemen (Eng.: manual for electronic computer centers of the swiss federal administration, a method for the development of systems)’ [27].

The second version, HERMES 1986, was published in 1986 after a revision. Almost ten years later, in 1995, the third version, HERMES 1995, was published after a revision based on the project management model V. The fourth version, HERMES 2003, was published in 2003 and introduced the well-known concept of flexible adaption (tailoring). The new version of HERMES, called HERMES 5, was published in

April 2013. The newest release is HERMES 5.1, which was published in June 2014 and which incorporates the agile scenario following the concept of flexible adaption as a main feature of the method [27].

2) HERMES 5 Essentials

HERMES 5 is well known in Switzerland and is the standard project management method in the Swiss Federal Administration. We assume that the HERMES 5 less known to many enterprises outside of Switzerland. For this reason HERMES 5 is described here in more detailed than SCRUM. An in-depth description of the whole method can be found in the reference handbook from G. Eicher, B. Kruschitz and H. Mourgue d'Algue [19].

To understand the main idea of HERMES 5 it is important to understand the concepts and elements that are described below:

- **Scenario:** A scenario provides a set of necessary methodological elements for different types of projects. HERMES 5 provides eight standard scenarios and the possibility to create one's individual scenario. The agile scenario 'customized IT application (agile)' is one of the standard scenarios. [19]
- **Phases:** Every project consists of different phases. As such, HERMES 5 is a waterfall methodology that uses the four mandatory phases 'initiation', 'concept', 'implementation' and 'deployment' in a sequence. [19]
- **Modules:** Every scenario consists of different modules, and each module bundles the tasks and outcomes that belong together. HERMES 5 provides 13 standard modules (e.g. 'Agile Development', 'IT System') and the possibility to define individual modules. [19]
- **Roles:** HERMES 5 names the roles of the core organization and those of the project organization. The roles of the core organization are responsible for the organization-wide project management and project steering (e.g. portfolio management, definition of guidelines). The project organization is a temporary organization, which defines the different roles for the project team. Every IT project must have at minimum the role of the 'project sponsor', 'project manager' and 'specialist'. [19]
- **Tasks and Outcomes:** Every role in a project is responsible for one or more tasks. A task consists of different activities, whereby the execution of all these activities delivers a specific outcome (e.g. the task 'design a system concept' leads to the outcome 'system requirements'). HERMES 5 defines the standard tasks that are necessary within a project. The outcomes (documents) are the central elements of HERMES 5. Therefore, HERMES 5 provides a document template for every outcome. [19]

3) The scenario 'customized IT application (agile)'

The scenario 'customized IT application (agile)' shown in Fig. 1 is one of the eight standard scenarios and is used for IT projects implementing individual business software.

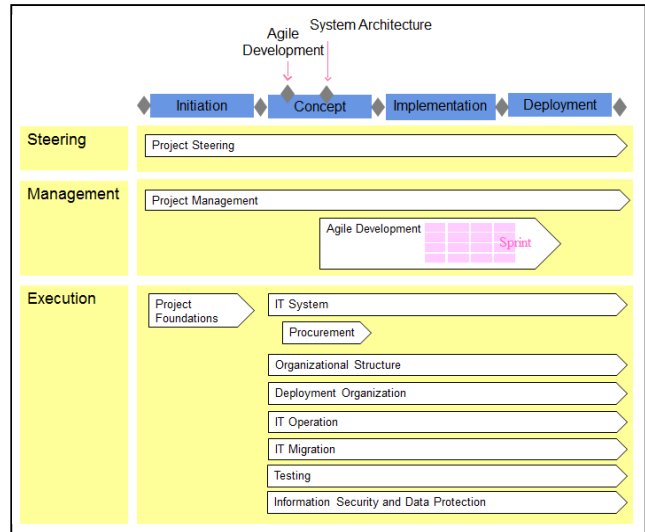


Fig. 1. Scenario 'customized IT application (agile)'

The main requirements engineering activities for gathering the system requirements take place in the module 'IT System', which starts at the beginning of the concept phase. A little later the module 'Agile Development' starts in which the gathered system requirement are prioritized and implemented applying SCRUM. New and changing requirement are tracked frequently. Such up-front gathering of system requirements is compatible with SCRUM. Especially for small projects with little complexity; the up-front gathering of the requirements can be kept short.

HERMES 5 has high focus on the outcomes. It defines the 'system requirements' as outcome of the task 'design a system concept'. The template for the outcome shows that functional and non functional system requirements are gathered, but HERMES 5 doesn't describe how they have to be documented or transferred to the module 'Agile Development' in the product backlog. The kind of documentation depends on the chosen standard/notation.

V. PROBLEM DESCRIPTION

The application scenario 'customized IT application (agile)' shows the following discrepancies between the two project management methods:

- **Procedure:** The individual procedures of both methods are clear, but how do they mutually interact?
- **Roles:** Each method defines its own roles. SCRUM has three roles; HERMES 5 defines a bigger set of roles for the creation of each outcome. Are these the same or overlapping? Do they share responsibilities?
- **Outcomes:** One value of the Agile Manifesto says that 'working software goes over a comprehensive

documentation'. HERMES 5 has the focus on the outcomes (documents). How should the outcomes be documented in order to be most valuable for the software development activities?

A deeper look into the main elements and the structure of the agile scenario as well as the practical experience show various problems when applying the agile scenario.

A closer look into the procedure of the HERMES 5 module 'IT Systems' respectively the task 'design a system concept' reveals that the scenario doesn't define how to implement requirements engineering (see section A). Furthermore the agile approach raises structural questions (see section B) and questions that refer to some explicit elements of the agile scenario (see sections C, D).

A. Requirements Engineering Activities

A big part of the requirements engineering activities in HERMES 5 is allocated to the 'concept' phase in the module 'IT System'. The mandatory outcome 'system requirements' is the output of the main requirements engineering activities. HERMES 5 defines the outcome, but neither defines how to execute the requirements engineering activities nor how to document (textual or model-based) the requirements.

In the ideal case, the core organization defines the requirements engineering process and the single activities as well as the form of documentation for all its projects. In reality, however, every project invents this part for it-self, because the core organization provides no guidelines.

B. Milestone 'Agile Development'

The decision to practice agile development with SCRUM is made at the beginning of the 'concept'-phase. The project charter, which defines the goals, the plan, the organization, the financial and non financial instruments etc. for the whole IT project, is worked out during the 'initiation' phase. To make the decision for agile development the project manager has to work out facts for decision-making based on estimates about resources. The early design of the project charter with a lack of information about the later agile development is a weakness.

An additional stumbling block is that the project sponsor often can't commit to SCRUM. The project sponsor, from a customer perspective has mostly no knowledge about the most appropriate method. It is important that the project manager decides already in the 'initiation'-phase whether to use SCRUM or not. He should indicate it in the project charter and insure that the project sponsor understands the benefits and implications of using SCRUM. Only if the customer has a deep understanding of the method and he is willing to implement it in the organization (e.g. less documentation, quick decisions) SCRUM can reveal its potential.

C. Phase release 'implementation'

According to the HERMES 5 task description the release of the 'implementation' phase is straight forward. The approval of the system requirements is a very important step at the end of the 'concept' phase because in a waterfall approach later change requests are often very expensive. In the current

framework, the mandatory result 'system requirements' has to be worked out during in the 'concept' phase, independent from the decision whether the agile or traditional scenario is used. The iterative and incremental development with SCRUM starts within the agile scenario at the beginning of the 'implementation' phase. For the business analysts and the requirements engineers it is not clear in practice what the developers need and at what level of granularity they need it in order to start with the development, nor why one should work out the mandatory result 'system requirements' when the product backlog to be developed later comprises the central instrument for managing the requirements. Why not start from the beginning with the product backlog? SCRUM itself obviously does neither assume nor define any requirements engineering before the start of the project. Therefore, in order to be able to efficiently combine HERMES 5 and SCRUM it is necessary to develop a requirements engineering process, which defines the necessary results and the level of granularity in a clear way!

The first point mentioned concerning the approval of the requirements is a problem in practice. The project sponsor first wants to approve the requirements before the development starts. When using the agile scenario, only a set of specific requirements is specified in detail at the end of the 'concept' phase. The project sponsor has to know and understand why not all the requirements are specified in detail from the beginning. Only if he understands and agrees this fact he will release the next phase.

D. Release Plan

In the breakdown of the work structure of the agile scenario the release plan isn't marked as mandatory. From the point of view of HERMES, the release plan can be integrated into the project management plan and as such is not a mandatory result. From the SCRUM point of view, the release plan is essential for planning the sprints and releases. The release plan also gives important information about which functions of the software have to be delivered first, which shows in reverse what must be specified at which point in time.

The biggest discrepancies that have to be solved are the clarification of how to do the requirements engineering and how to optimize the mutual interaction between requirements engineering and software development.

VI. METHODOLOGY

Our research objective was to develop a requirements engineering process that both meets the requirements of the project management method HERMES 5 and takes advantage of the agile software development method SCRUM. As design science research "is motivated by the desire to improve the environment by the introduction of new and innovative artifacts" [25] we applied the Design Science Research method, introduced by Hevner and Chatterjee [25]. For our approach we considered the following phases:

A. Awareness-Phase

To gather requirements and data we analyzed related work conducted several informal interviews with representatives

from the Swiss Federal Administration and made an online survey with participants (mainly developers) from the Swiss Federal Administration and the private sector. The contact to the Swiss Federal Administration was given through the practical experience as an IT consultant in various IT projects in four different federal departments. We then analysed the results to recognize patterns and to compare the practical knowledge base with the theoretical knowledge base. Additionally we conducted a detailed analysis of the available scenario ‘customized IT application (agile)’ with the focus on the following three parts:

- A general process analysis of HERMES 5, of all tasks across all phases, was necessary to point out the weaknesses and inconsistencies of the method in terms of the usage of the agile scenario.
- A detailed analysis of the agile scenario with the focus on the requirements engineering discipline. More specifically we analyzed the task ‘design a system concept’ of the module ‘IT System’ (phase ‘concept’).
- A detailed analysis of the agile scenario with the focus on the agile software development discipline. More specifically we analyzed the module ‘Agile Development’.

Based on this information we started the Suggestion-Phase for the creation of the requirements engineering process in the context of the agile scenario ‘customized IT application (agile)’.

B. Suggestion-Phase

Based on our findings we created a requirements engineering process that could be used in every IT project in the Swiss Federal Administration or any another company for the development of individual business software by using HERMES 5 and SCRUM.

The existing scenario ‘customized IT application (agile)’ formed the base for the process. For this reason the Suggestion-Phase included not only the design of the process; but also concluded in a list of proposed improvements to adapt the scenario.

C. Evaluation-Phase

As a last step we did a proof of concept of the analysis results by having reviewed our findings and proposed changes by four selected members of the eCH Standard Group - Section HERMES (including the author and co-author of HERMES 5). The goal of the proof of concept was to find out whether the developed artifact, the requirements engineering process, has the potential to be incorporated to upcoming versions of the HERMES 5 open standard.

Changes for a new HERMES 5 release have first to be discussed and approved from the eCH Standard Group – Section HERMES. This was the reason why we decided to ask selected members of the eCH Standard Group to read our research. The selection of the evaluators was done by Guido

Eicher, the author of HERMES 5 and chairman of the eCH Standard Group - Section HERMES. Together with the co-author of HERMES 5, they were checking the research according the HERMES 5 conformity. Another evaluator was responsible for checking the research from the viewpoint of the development and the forth evaluator had a look at the whole research results without any specific focus. All of the evaluators read the research independently and sent their written feedback to us, which was finally discussed in a meeting with Guido Eicher.

In addition to the evaluation of practical relevance of the developed artifact we thoroughly validated its relevance (rigor cycle), in order to guarantee that the requirements engineering process is a sound research contribution.

VII. REQUIREMENTS ENGINEERING USING HERMES 5 AND SCRUM

A. Requirements Engineering in Swiss Federal Administration

This section contains the main findings out of the awareness-phase, which build the base for the agile requirements engineering process in the context of the agile scenario ‘customized IT application (agile)’.

1) Existing regulations

The Swiss Federal Administration uses for its IT projects primarily the project management method HERMES 5, which doesn’t define how to execute the requirements engineering activities. One of the problems is that in practice every IT project reinvents the details how to implement requirements engineering in case of a later agile development with SCRUM.

2) The agile scenario and its unclarities

As already described in the ‘introduction to HERMES 5’ the standard scenario ‘customized IT application (agile)’ provides the whole structure and all the elements for IT projects for the development of individual business software using SCRUM during software development.

The detailed analysis of the agile scenario with the focus on requirements engineering (module ‘IT System’) has led to the following main findings:

- The task ‘work out system requirements’ doesn’t define how to execute requirements engineering.
- It is not described in which form (document or tool based) the outcome ‘system requirements’ has to be documented.
- The outcome of ‘system requirements’ does not define particular delivery objects
- It is not defined which role is responsible for the creation of the outcome.

Overall, the requirements engineering process in agile software development project isn’t covered explicitly in the agile scenario. For the agile development the scenario uses SCRUM. The detailed analysis of the agile scenario with the

focus on the agile software development (module 'Agile Development') part has led to the following main findings.

- The SCRUM process is used correctly in relation to the procedure.
- The tasks 'decide on agile development using SCRUM' and 'introduce SCRUM' should be preconditions of the module 'Agile Development', and as such, be decided upfront.
- The task 'design a release plan' is not part of SCRUM. Anyway the task should not only focus on software development but also on the requirements engineering discipline that has to be incorporated to the release plan.
- The roles, which are responsible for the tasks of the module 'Agile Development', have not the same naming like in SCRUM. In other words the SCRUM roles are not addressed explicitly in the agile scenario.
- The single activities and outcomes of the task 'work in sprints' are not named as they are defined in SCRUM.
- The task 'keep a product backlog' contains the creation of the product backlog based on the gathered system requirements and the prioritization of them. The requirements engineering activities during a sprint are not described.

3) Expectations from Development

The lack of harmonization between requirements engineering and software development and between the methods HERMES 5 and SCRUM are well known in practice. To develop a common solution it is important to establish mutual understanding. In order to get an appropriate insight, we executed an online survey to find out what the developers expect from requirements engineering. All participants together have practical experience from around 50 agile software development projects. The following list points out the main findings:

- Close collaboration between requirements engineering, software development and testing.
- Central management of the requirements (same base for all) or continuous integration.
- Involve the requirements engineering team till the completion of the project.
- Sensitize the Product Owner and the whole project team for the agile discipline.
- Define together clear regulations for requirements engineering (e.g. procedure, delivery objects, depth of specification).
- Do early technical verification of the requirements by the development team and the customer.

- Do reasonable requirements engineering before the start of the implementation of SCRUM. On time completion of the requirements (at least before the sprint).
- Only describe the "what" in requirements engineering (not the "how"). Professional and clear formulated requirements (business focus, no technical focus).
- Use epics and user stories. And use visual artifacts alongside textual requirements like screen mock-ups (inclusion of a usability expert). Describe the normal path, alternative path and errors.

B. Agile Requirements Engineering

Due to the application scenario, the agile scenario 'customized IT application (agile)' builds the frame for the agile requirements engineering. Based on the findings from the analysis it was necessary to adjust the agile scenario to harmonize with SCRUM and to treat the practical stumbling blocks.

1) Changes in Module 'Project Management'

The task 'decide on agile development using SCRUM' was originally assigned to the module 'Agile Development' through the Project Manager. In fact, the decision for agile development not only has an influence on the activities of implementation, but also of requirements engineering. For this reason the task 'decide on agile development using SCRUM' must be done earlier. The task should be moved from the beginning of the 'concept' phase to the end of the 'initiation' phase. At this point of time it is already possible to take the decision into account for the creation of the project charter, which in turn causes the customer to commit to agile development with the sign-off of the 'project charter'.

The task 'introduce SCRUM' was originally executed in the module 'Agile Development'. But after a decision for agile development using SCRUM has been made, defining the tools/instruments, procedure etc. is not only relevant for the development side. It is equally important to do the same for requirements engineering and to define the collaboration between requirements engineering and development. This includes common understanding of the artifacts and the 'definition of ready' for these single artifacts as well as the definition of how to document the requirements (in a requirements engineering tool, in a document, as product backlog etc.).

The task 'introduce SCRUM' doesn't intend to train the project team, but SCRUM is with a high probability unknown to the customer. By this reason it is important that all involved parties in the IT project know what SCRUM is and how it works. At the very least we recommend deleting the activity 'conduct first sprints' in the task, because it corresponds to the task 'work in sprints'. Furthermore, it is recommended that the 'Project Management Plan' be defined so that the first sprints have the goal to test the procedure.

The third task that has been readjusted is the task 'design a release plan'. This task was originally part of the module

‘Agile Development’. Practical experience from the development teams showed that the requirements engineering activities are often executed too late during the sprint. The release plan should not only consolidate the basic plan for the deployment- but also for the concept-phase. From the point of view of requirements engineering it is important to know which parts have to be developed first and which, as a consequence, have to be delivered first for development. For these reasons the task ‘design a release plan’ has been moved, along with the associated activities and outcomes of the module ‘Project Management’. In a first step the release plan could be on a high level. But the task must be a continuous task to be adjusted in every sprint of requirements engineering and in every sprint of development. The creation of the release plan is the responsibility of the Project Manager, but he has to involve the Product Owner, the Business Analyst, The SCRUM Master and the Developers.

2) Changes in Module ‘Agile Development’

The changes that affect the module ‘Agile Development’ were already described in the previous sections (shift of the tasks ‘decide on agile development using SCRUM’, ‘introduce SCRUM’ and ‘design a release plan’ into the module ‘Project Management’).

It should be noted that the task ‘keep a product backlog’ still belongs to the module ‘Agile Development’, but also supports the new module ‘Agile Requirements Engineering’.

3) New Module ‘Agile Requirements Engineering’

The biggest change is the new module called ‘Agile Requirements Engineering’. The module consists of the same tasks like the module ‘IT System’.

The task ‘implement prototype’ covers the specification, implementation and evaluation/documentation of the prototype. In an agile software development project, the creation of a prototype should also use the agile procedure and selected tasks from the agile scenario respectively. For this reason specifying the prototype should be handled equally with the task ‘design a system concept’. The implementation component is covered in the whole module ‘Agile Development’, the documentation component in the activity ‘update documentation’ in the task ‘implements system’. Therefore, it would not be necessary to list the separate task ‘implement prototype’ as part of the module.

The activities of the task ‘implement system’ and those of the task ‘implement prototype’ are partly covered by other tasks. The activity ‘work out the detailed specification’, in particular, takes part in agile development already in the task ‘design a system concept’, or later during the sprints.

The last point mentioned regarding the creation of the specification is handled in the task ‘design a system concept’. This task handles the initial requirements engineering, which should also not be neglected in the later agile development with SCRUM. The challenge is to adapt requirements engineering as best as possible to the agile approach. The current task only contains the activity ‘work out system requirements’. The analysis showed that this activity is a black box with no further information on how to do it.

The next chapter treats the gap of the missing requirements engineering and shows how the requirements engineering could be handled in an IT project that uses the scenario ‘customized IT application (agile)’. The agile requirements engineering activities take place in the new module ‘Agile Requirements Engineering’ in the task ‘design a system concept’. Today the existing task knows only the activity ‘work out system requirements’ as requirements engineering discipline that creates the outcome ‘system requirements’. With the new process described in the next sections we would like to focus on the following points:

- The entire requirements engineering discipline as well as its single main activities that are applied in an agile software development projects of the Swiss Federal Administration that uses the scenario ‘customized IT application (agile)’. The process addresses how the requirements engineering is matched to the later development with SCRUM.
- The roles which are responsible for the execution of the activities of the process and other additional roles and how they can be combined with the roles in SCRUM.
- The mutual interaction between requirements engineering and agile software development with SCRUM.

a) Process and Activities

The preconditions for starting the process are as following:

- The Project Manager has decided on agile software development with SCRUM.
- Both sides, requirements engineering and software development were involved in the creation of the release plan and have committed to it. It is clear what to develop and accordingly what to specify first.
- Both sides, requirements engineering and software development have agreed on the delivery objects and their documentation form (SCRUM is introduced). The ‘definitions of ready’ for each delivery object are clearly defined.

The process starts with three initial activities that are executed one time when starting the process at the beginning of the ‘concept’ phase.

• Check basis conditions of the project order

The project order defines the basic conditions for the project. The Business Analyst has to check in collaboration with the Project Manager the basic conditions, which affect the requirements engineering activities (e.g. available resources, time frame, and project plan resp. release plan). When the conditions are clear he could continue. If not he has to clarify the problems with the Project Manager.

• Check situation analysis

The situation analysis was created in the phase ‘initiation’ and builds the base for requirements

engineering. The situation analysis describes the current system and the direction for the further development.

The Business Analyst has to check the situation analysis at the beginning to know the basic conditions and secure that he doesn't overstep the crash barriers later in requirements engineering.

- **Work out situation analysis**

If the situation analysis isn't detailed enough the Business Analyst has to complete/concretize the analysis.

When the situation analysis is complete and the basic conditions of the project order are clear and given, agile requirements engineering starts. It is worth mentioning that parallel to requirements engineering the IT Architect needs to work on the system architecture, which has to be in line with the requirements engineering (this collaboration was not part of the investigations).

- **Select topic(s)**

The Business Analyst selects in consultation with the Product Owner one or more topics for the first requirements engineering sprint. It is comparable with the 'Sprint Planning 1' in SCRUM.

The release plan provides the set of topics, which have to be done in the current release. The Business Analyst and Product Owner decide together the prioritization of the topics.

- **Plan RE activities for selected topic(s)**

After the selection of one or more topics for the first requirements engineering sprint, the Business Analyst plans the single steps of requirements engineering. The activity is comparable with the 'Sprint Planning 2' in SCRUM.

It is possible that the detailed planning of a topic discovers a lot of hidden subtopic. Hidden subtopics can lead to a higher effort than planned. In this case it is important to discuss this immediately with the Project Manager and adapt if necessary the release plan.

After selecting the most important topics the first requirements engineering sprint starts. The detailed analysis has showed that the development side expects a prompt delivery of the system requirements (due to the short development sprints). The requirements engineering discipline for the development of individual business software includes often the clarification of bigger issues, especially in the public environment. The following list shows aspects that can prevent short requirements engineering iterations:

- Political factors (e.g. open political decision, which influences project scope)
- Legal factors (e.g. changing law, guidelines, standards)

- Dependencies to other projects (different prioritization of these projects → different project progress)
- Dependencies to foreign systems
- Size and local distribution of project team
- Complexity of business area / project topic
- Uniqueness of business area / project topic

Due to this fact it is mostly not possible to conduct as short requirements engineering sprints as development sprints. Reasonable requirements engineering has to be done before the start of the first SCRUM sprint to clarify bigger uncertainties, manage dependencies and their impact on the project; especially in the case of complex unfamiliar project topics. To still use the power of SCRUM it is necessary to feed bigger blocks of consolidated system requirements to the development team to be split up into sprints according to the priorities of the product owner. The integration happens in bigger blocks, whereas one block contains topics for more than one development sprint.

- **Execute elicitation**

After the planning of a topic the Business Analyst executes the elicitation of the system requirements. The used elicitation techniques (e.g. document-centric techniques, observation techniques, creativity techniques) depend on the topic. It is possible that the elicitation is handled in multiple executions.

It is very important to already involve the Product Owner, the developers and the superiors of the User Representative in the elicitation procedure.

- **Document requirements**

The documentation of the requirements can happen in parallel to the elicitation or afterwards. As mentioned before, delivery objects should be clearly defined including a 'definition of ready' for each delivery object and how it should be documented. How the single delivery objects are documented depends on the project and the preference of the developers. The conducted survey (cf. methodology chapter) showed that developers not only expect epics and user stories in agile environment but also ask for use cases and visual artefacts. If the development sides use a tool for the product backlog it is recommended to document the system requirements directly into the product backlog. This should be mutually agreed with the Team.

- **Work out detailed study**

Depending on the topic it is necessary to work out a detailed study. It is important to create the detailed study only in case the development or customer side asks for it. Avoid creating too much documentation that wouldn't be read.

- **Validate results**

After the elicitation and documentation of the system requirements the Business Analyst is responsible to validate the results with the customer and with the

development team. The validation can be a written examination of the documented requirements and/or a validation meeting with a written minute. Through the validation, errors in requirements engineering should be detected and corrected at an early stage before handing it over to the development.

The goal of the validation with the development is to prove the technical feasibility and to check the 'definition of ready', which defines when the system requirements are ready for the handover to the development. Additionally the validation through development should secure that the development clearly understands the specification and uncertainties and obstacles from the perspective of development are discussed

- **Approve topic(s) for use**

If the validation was successful the topics are going to be approved through the User Representatives and the Superiors of the User Representatives to approve the topic 'ready for use. 'Ready for use' means that development can use the system requirements in the module 'Agile Development'. However, it is still possible that the requirements can change based on discussions or decisions of the Product Owner.

If the Superiors of the User Representative were already highly involved in the elicitation and validation activity it is recommended to make the release through the responsible Business Analyst (author release).

- **Prepare handover to development**

After the release of a topic the Business Analyst prepares and coordinates the handover to the development with the Product Owner.

The handover includes the transfer of the system requirements into the product backlog of the development. Later the Business Analyst can support the Product Owner on demand in the module 'Agile Development' (e.g. participation at sprint planning 1 or review). But in general the Product Owner should be able to clarify upcoming questions quickly by himself.

- **Check for further topic(s)**

After the handover the Business Analyst checks the release plan for next topics and restarts the procedure again with the selection of one or more topics.

Besides the execution of the requirements engineering activities it is recommended to conduct a weekly meeting during the requirements engineering process. This is especially important with big and complex projects where more than one Business Analyst is involved, which leads, as a consequence, to a parallel execution of requirements engineering activities for different topics. The weekly meeting builds the place where the Business Analysts, the Product Owner and the representatives from development inform each other about the current work and occurring problems.

b) Roles

The process declares two roles responsible for executing the activities. The IT Architect for the system architecture activities and the Business Analyst for all other activities. The other HERMES 5 roles that are involved in the task 'design a system concept' like the User Representative, Business Process Owner, Project Manager, Operations Manager support the Business Analyst or IT Architect and have only subordinate responsibility.

Often the question by using the agile scenario 'customized IT application (agile)' is how the roles of HERMES 5 and SCRUM could be combined. The problem is that the module 'Agile Development', which represents the SCRUM process, doesn't use the SCRUM roles. In other words the SCRUM roles aren't explicit addressed in HERMES 5. From the point of the requirements engineering discipline a combination of the roles is not necessary. But it is important to already involve the Product Owner and the Developers from the SCRUM organization during the execution of the activities to create a common understanding. As soon as the development starts agile development with SCRUM the SCRUM organization has to be activated. At this point it is important to understand that all the HERMES 5 roles can be subsumed under the SCRUM Team. The columns of the table below show all the roles that are involved in the task 'design a system concept'. The rows represent the SCRUM roles. The matrix shows which HERMES 5 are involved later on in 'Agile Development'.

TABLE I. HERMES 5 VS. SCRUM ROLES

SCRUM Roles	HERMES 5 Roles							
	Business Analyst	IT Architect	User Representative	Business Process Owner	Developer	Product Owner	Operation Manager	Project Manager
SCRUM Master								
SCRUM Product Owner						x		
SCRUM Team	x	x	(x)	(x)	x		x	

TABLE I. shows that most of the HERMES roles can later on be involved in the SCRUM process as part of the SCRUM Team. This means concrete when the Developers, as fix part of a SCRUM Team, need consultation they can invite any other role (e.g. for the review). It is to mention that the interests of the User Representative and Business Process Owner should be represented through the SCRUM Product Owner. By this reason the affiliation (x) is set in brackets. The HERMES 5 role Project Manager is not directly involved into the SCRUM process, but it is to keep in mind that the SCRUM Master should report to the Project Manager. The Product Owner is involved in the requirements engineering discipline as well as in the SCRUM process. Being Product Owner is a fulltime job, why it is important that he only has this single role. In relation to the single process we have the following main responsibilities:

- In the module ‘Agile Requirements Engineering’ the Business Analyst is responsible for the execution of the task ‘design a system concept’.
- In the module ‘Agile Development’ the SCRUM Team respectively the Developers are responsible for the execution of the task ‘work in sprints’. And the Product Owner is responsible for the execution of the task ‘keep a product backlog’.

c) *Interaction with agile development*

This section shows how the task ‘design a system concept’ interacts with the tasks of the module ‘Agile Development’. After the validation and approval of the system requirements for a topic the Business Analyst prepares and coordinates together with the Product Owner the handover to the development.

Assuming that the documentation of the requirements is done as a product backlog, the handover occurs in a meeting in which the Business Analyst discusses the requirements with the Product Owner. Later on, the Product Owner is responsible for the further management of the system requirements in the product backlog. This means that he is responsible for the main activity ‘management’ of the requirements engineering process, as well as the management of the product backlog in terms of SCRUM. Both of these have the same goal, namely, to keep the requirements or set of requirements up to date. After a tranche is handed over to development and is recorded correctly in the product backlog, the SCRUM Team can start with the implementation in the agile software development process. If the SCRUM Team discovers a need for change in the activity ‘conduct review’ (sprint review), then the SCRUM Product Owner creates a new backlog item for the correction in the task ‘keep a product backlog’. It is to mention that it is absolutely possible that the development need some professional clarification of a system requirement. In this case and at this point of time the Product Owner should be able to clarify on its own without any excessive elicitation and documentation procedure.

VIII. OUTLOOK

The evaluation of the requirement engineering process, performed by four selected members of the eCH Standard Group (cf. methodology chapter) proofs its usefulness but also reveals some aspects that need further investigations:

- The integration of requirements engineering into HERMES 5 is very valuable.
- The detailed analysis of requirements engineering is specifically focused on the task ‘design a system concept’, but there are further requirements engineering activities to be investigated.
- There is further potential to clarify how requirements engineering cooperates with testing

Overall, the evaluation of the requirements engineering process through selected eCH Standard Group members was very positive and the findings of our research will officially be

submitted to the eCH Standard Group - Section HERMES for further evaluation and potential next steps.

IX. CONCLUSION

In Swiss Federal Administration, conducting IT projects using agile development methods is already implemented in current practice. However, although the current HERMES 5.1 release recently included the ‘customized IT application (agile)’ on a very high level, many practical issues remain unresolved. In this paper, we investigated research into practical problems that arise from the combination of HERMES and SCRUM with specific focus on the requirements engineering discipline. Based on an in-depth analysis of the tasks as well as roles and responsibilities, of the current agile scenario a new agile requirements engineering process has been proposed, that includes feedback from over 50 projects, that applied SCRUM within the overall HERMES framework, which is a given standard and provides a lot of help from the perspective of managing a large project portfolio within a larger organization or enterprise. Main discussion points are addressed and resolved, that are major pitfalls in the mutual interaction between requirements engineering and agile software development teams. The different set of roles in the two frameworks and the process of handling the outcomes of the requirements engineering activities in a form that fits into a SCRUM based development process.

This research explains the problems in detail and makes a proposal for the adjustment of the agile scenario with a focus on requirements engineering. The adjustment includes an early execution of the task ‘decide on agile development’ at the end of the phase ‘initiation’ and shift it together with the tasks ‘introduce SCRUM’ and ‘design a release plan’ from the module ‘Agile Development’ into the module ‘Project Management’. The reallocated tasks insure an overall topic-based planning that spans requirements engineering and software development. The product backlog still mainly allocated to the module ‘Agile Development’ becomes important much earlier during the requirements engineering activities.

In the module ‘Agile Requirements Engineering’ respectively in the task ‘design a system concept’ the research shows how the requirements engineering discipline can be handled in agile IT projects. The proposed process adapts the iterative and incremental approach from SCRUM from software development to the activities in the requirements engineering discipline. The single activities remain basically the same, but the sequence of tasks and activities is adapted to work consistently on the product backlog from the beginning of the requirements engineering until the end of the implementation.

The requirements engineering activities are not very different from IT projects that use an agile approach from those that work in traditional waterfall projects. The main difference is the time point in time when they are executed and the amount of communication, that is involved between the Product Owner the requirement engineers and the developers. In addition to the ‘definition of done’, ‘the definition of ready’ should build the common understanding of the description of

the different product features. A further consequence is that decision to decide for agile software development should be taken earlier, so that the requirements can be established using early feedback processes as well.

Depending on the complexity of the IT project and the basic conditions in an organization, however, it is not always possible to force agility through the application of agile concepts. Especially the requirements engineering in complex IT projects in the Swiss Federal Administration cannot be agile in such a way like software development. The project topic is mostly very complex and unique. The first dealing with the topic and the possible solution needs time. Internal and external dependencies and influences have a high impact on the project schedule and needs often longer clarifications and general project-oriented decisions. Up to our knowledge our contribution is the first post of its kind and closes the gap requirements engineering and software development in the agile environment. The research shows how HERMES 5 and SCRUM can work together in the field of requirements engineering. The findings make a contribution to the Swiss Federal Administration, which uses HERMES 5 as federal standard, but also to other public administrations or private companies that use the project management method HERMES 5. The FITSU as responsible unit for the further development of HERMES 5 has recognized the findings and will make a request to the eCH Standard Group (Section HERMES) to add the findings in form of a best practice guide to the eCH Standard 0054.

REFERENCES

- [1] G. Eicher, B. Kruschitz and H. Mourgue d'Algue, "eCH-0054 HERMES Projektmanagement-Methode", 2014, available at: <http://www.ech.ch/vechweb/page?p=dossier&documentNumber=eCH-0054&documentVersion=2.0>
- [2] N. Abbas, A.M. Gravell and G.B. Wills, "Historical roots of agile methods : where did "agile thinking" come from ?", 2008.
- [3] R.J. Perring, "Lean meets agile software development & project management", 2010 , p.12. Available at: http://www.aspe-sdlc.com/offers/pickups_0386572/lean_meets_agile.pdf.
- [4] Basili, R. Victor (University of M. & Larman, C. (Valtech), "Iterative and incremental development: a brief history", (June) 2003., p.10.
- [5] D. Casali, "A brief history of agile methods", 2012, available at: <http://intenseminimalism.com/2012/a-brief-history-of-agile-methods/>.
- [6] K. Beck et al., "Manifesto for agile software development", 2001, available at: <http://agilemanifesto.org/>.
- [7] Agile Alliance, "agile alliance", 2014, available at: <http://www.agilealliance.org/>
- [8] D.S. Alberts, T.M. Takai, "The agility advantage: a survival guide for complex enterprises and endeavors", 2011
- [9] Federal IT Steering Unit, "HERMES und Agilität", 2010.
- [10] R. Noordeloos, Ch. Manteli, H. van Vliet, "From RUP to SCRUM in Global Software Development: A Case Study", 2012
- [11] G.J. Rankins, M. Kearns, "Integrating PRINCE2 and SCRUM for successful new product development", 2008
- [12] SwissQ Consulting AG, "Trends & benchmark report 2015", 2015
- [13] B.R. Grau, K. Lauenroth, "Requirements engineering and agile development - collaborative , just enough , just in time , sustainable - content", pp.1–18, 2014
- [14] C. Rupp, SOPHISTen, "Requirements engineering und management", 6th Editio., Carl Hanser Verlag München Wien, 2014
- [15] M. Steven Palmquist, Mary Ann Lapham, Suzanne Miller, Timothy Chick and Ipek Ozkaya, 'Paralleld worlds: agile and waterfall differences and similarities', 2013
- [16] A. Brown, J. Sutherland, "SCRUM at Scale. Go modular for greater success", available at: <http://www.scruminc.com/wp-content/uploads/2014/07/Scrum-at-Scale-A-Modular-Approach.pdf>
- [17] D. Leffingwell, "Scaled Agile Framework. A proven, publicly available framework for applying lean/agile practices at enterprise scale", 2015, available at: <http://www.scaledagileframework.com/>
- [18] SCRUM.org, "What is SCRUM?", 2015, available at: <https://www.scrum.org/resources/what-is-scrum>
- [19] G. Eicher, B. Kruschitz and H. Mourgue d'Algue, "HERMES 5 – Referenzhandbuch", 2014.
- [20] M. Beedle, M. Devos, Y. Sharon, K. Schwaber, J. Sutherland, 1999. SCRUM : An extension pattern language for hyperproductive software development,
- [21] R. Dräther, K. Holger, S. Carsten, "SCRUM - kurz & gut", 1th Editio., Köln: O'Reilly Verlag, 2013
- [22] iSport, "The rules of the scrum", 2014, available at: <http://rugby.isport.com/rugby-guides/the-rules-of-the-scrum>
- [23] B. Schär, "Requirements engineering process. HERMES 5 and SCRUM", 2015
- [24] DAS SCRUM Team AG, "SCRUM reader", 2012
- [25] A. Hevner and S. Chatterjee, "Design research in information systems, integrated". Springer, 2010.
- [26] Bergsmann Johannes, "Requirements engineering für die agile softwareentwicklung", 1st ed., dpunkt.verlag GmbH, 2014
- [27] Federal IT Steering Unit, "Die Geschichte der Projektführung mit HERMES", 2013, available at: <http://www.isb.admin.ch/themen/methoden/01661/01666/index.html?lang=de>.
- [28] J. Sutherland, K. Schwaber, "The scrum guide", (October) 2011.