University of Applied Sciences and Arts
Northwestern Switzerland

Master Thesis

Requirements Engineering Process
HERMES 5 and SCRUM

Author:
Birgit Schär
Maria-Schürerstrasse 8A
2540 Grenchen

Date:
Grenchen, 31 January 2015

Supervisor:
Dr. Stephan Jüngling
University of Applied Sciences and Arts
Northwestern Switzerland
School of Business
Peter Merian-Strasse 86
4002 Basel

Study Program:
Master of Science in Business Information Systems

Title Master Thesis:

Requirements Engineering Process. HERMES 5 and SCRUM.

Involved Parties:

Employer

INVERSUM GmbH	Phone:	032 625 60 00
Reto Brechbuehl	E-Mail:	reto.brechbuehl@inversum.ch
Rosenweg 2	URL:	www.inversum.ch
4500 Solothurn		

Author

Birgit Schär	Phone:	078 827 10 46
Maria-Schürerstrasse 8A	E-Mail:	birgit.schaer@students.fhnw.ch
2540 Grenchen		

Supervisor

Dr. Stephan Jüngling	Phone:	061 279 18 55
University of Applied Sciences and Arts	E-Mail:	stephan.juengling@fhnw.ch
Northwestern Switzerland		
School of Business		
Peter Merian-Strasse 86		
4002 Basel		

Table of Content

Executive Summary

The goal of this master's thesis was to develop a requirements engineering process that satisfied the requirements of the project management method HERMES 5 and took advantage of the software development method SCRUM. From the outset the challenge was to combine these two contrary methods.

HERMES 5 follows partially a traditional waterfall approach, in which it is possible to execute the phases "implementation" and "deployment" iterative by the use of implementation units and releases. HERMES 5 is an open standard that allows it to adapt the single elements of HERMES 5 according to the needs of the organization. It is allowed to extend or reduce elements and the outcomes can be combined or split if the content doesn't thereby change. HERMES 5, as an open standard, does not have a lot of mandatory requirements, and these are on a high level and are not threatened by changes on deeper elements. An initial analysis of the whole agile HERMES scenario revealed some points that aren't conducive to the agile approach or that make its manner of implementation unclear.

SCRUM follows the agile approach. The advantages of SCRUM are that the customer sees a result at an early stage of the IT project. Through tight collaboration between the customer (Product Owner) and the developer, it is possible to identify problems and faults early on, which leads to high product quality. Through fast value delivery, the customer satisfaction is more rapidly guaranteed. Along with the many advantages of SCRUM, however, come just as many stumbling blocks, which may impede the achievement of the benefits mentioned. All of these stumbling blocks can be traced back to a false or insufficient understanding of the use of SCRUM.

Requirements engineering is responsible for raising, documenting, validating/negotiating and managing the system requirements that build the base for the development of the software. The literature review has shown that the requirements engineering activities are the same in traditional waterfall and agile approaches. The main difference hinges on the moment in which the activity is conducted.

In the detailed analysis, three points were investigated in detail. First, the agile HERMES scenario in general, second the requirements engineering section, and third the agile software development section of the agile HERMES scenario. The first part of the analysis covered all modules of the scenario. The analysis identified some general weaknesses, which should be adjusted independently of the result of the master's thesis. The second part of the analysis showed that the originally agile HERMES scenario doesn't define how requirements engineering is executed. It recommends that if the core organization has regulations regarding requirements engineering, that they should be taken into account. But such regulations are only present in the minority of the departments of the Swiss Federal Administration. The third part of the detailed analysis demonstrated that the agile HERMES scenario uses the SCRUM process within the meaning of the SCRUM guide (Sutherland & Schwaber 2011). The main finding was that the agile development section contains tasks that have to be fulfilled at the beginning of the project in order to prevent overlooking the stumbling blocks. An online survey has shown that the development side wishes for a close collaboration with requirements engineering. It is undisputed that a decent requirements engineering is also required in agile software development. The development has to be already involved during requirements engineering in order to achieve system requirements of high quality.

The developed agile requirements engineering process covers the elicitation, the documentation, the validation and the management of the system requirements. In the original agile HERMES scenario the requirements engineering activities are not described in detailed; only that the activity “works out system requirements” and “works out detailed study”, indicating that requirements engineering happens in the task of “design a system concept”. The developed requirements engineering process describes with a higher granularity what to do in requirements engineering. The process corresponds to the proposition of the International Requirements Engineering Board (IREB). But contrary to the traditional requirements engineering the management of the product backlog in sense of SCRUM is part of the module “Agile Development” and involves directly the SCRUM Product Owner.

The developed agile requirements engineering process meets the requirements of HERMES and takes partial advantage of SCRUM. With the focus on the development of an individual business software for a federal department in the Swiss Federal Administration, the agile requirements engineering process doesn’t force agility, which is not due to the process itself. The complexity of an IT project and the basic conditions in an organization can slow down the process execution and prevent agility.

The proof of concept of the results has led all in all to a positive feedback. The master thesis will be handed-in official to the eCH Standard Group Section HERMES, at which the members from the public and private sectors will discuss the master thesis and the already identified evidences again. They will take the decision if the master thesis can be added as best practice guide to the Standard eCH 0054 and if it is necessary to make some changes on the HERMES reference book based on the results of the master thesis. The secondary goal of the master thesis to add a value to the community can be reached finally with this next step, in which Guido Eicher, the chairman of the eCH Standard Group Section HERMES, already confirmed that the master thesis delivers input for the next HERMES release.

Foreword / Acknowledgements

In my daily work as a business analyst, I'm directly confronted with the topic of the master thesis. The question of how to use requirements engineering as best as possible for a later agile development is a question that I have met several times in my practical work in the past. The master thesis has given me the opportunity to deal intensely with the topic. The time for creating the master thesis was beside the job challenging but very instructive. The result of the master thesis has shown me that it is not as easy as I had thought to work out one common solution, which confirms the values and principles of the Agile Manifesto. One personal goal with the creation of the master thesis was furthermore to bring a benefit to the HERMES community. I'm very happy that it was possible to make an assessment of the master thesis results through experts of the Federal IT Steering Unit (FITSU), which are partially members of the eCH Standard Group Section HERMES. The assessment lays the first foundation stone for a valuable contribution.

A special thanks goes to my supervisor Stephan Jüngling (University of Applied Sciences Northwestern Switzerland), who has supported me active in every phase of the master thesis. Furthermore, I would like to thank Guido Eicher (Federal IT Steering Unit), who is very engaged in bringing the result of my master thesis to the HERMES community. And finally I would like to thank the people that were available for an interview, the people that participated in the online survey, the people that clarified the question about available requirements engineering regulations in their Federal Department, and finally, I would thank all the experts, who took the time to read my work and to judge it. Without their help, it would not have been possible for me to reach to the findings contained in this work.

1. Introduction

This chapter describes the content and scope of the master's thesis. The background information (chapter 1.1) provides a brief insight into the topic and describes the problems that motivated the choice to write about it (chapter 1.2). The introduction describes the thesis statement (chapter 1.3) as well as the research questions and objectives (chapter 1.4) that have to be investigated and answered in the master's thesis. Furthermore, it describes limitations imposed on the scope (chapter 1.5) and their rationale, as well as the contribution of the study (chapter 1.8 and 1.9). The last section gives an overview of how the master's thesis is structured (chapter 1.10).

1.1 Background Information

According Wang and Conboy (2009), the agile software development methods have attracted more and more attention since the start of the 21st century. The history of agile software development has its beginnings in the first half of the 20th century (1930) with the development of the PDSA- (Plan, Do, Study, Act) Lifecycle by Walter Shewhart, which is the basis of the iterative and incremental software development methods (Perring 2010). The milestone for iterative and incremental software development was set 20 years later in 1950 with the project X-15 hypersonic jet (Basili, Victor & Larman 2003). After 1950, various projects in the military and aerospace areas (e.g. Project Mercury, USS Ohio Submarine command and control system, Army Site Defence missile tracking software, Light Airborne Multipurpose System, Space Shuttle avionic software) used an iterative and incremental software development approach (Casali 2012). Much later, in 2001, seventeen people working in the area of software development came together to discuss the topic of agile software development. The result was the Agile Manifesto for software development, which defines the basic values and principles for all agile software development methods (Beck et al. 2001).

The topic of agile software development was raised in 2013 in the Swiss Federal Administration with the release of the new HERMES version. The HERMES project management method, which is the main one used in the Swiss Federal Administration, is an open standard developed by the Federal IT Steering Unit (FITSU) of the Federal Department of Finance (FDF). HERMES is applicable to all kind of projects (both IT and non IT). The last HERMES version, "HERMES 2003/2005", provided a traditional project management method for software development or software adaption using a sequential procedure. The new HERMES version "HERMES 5" was published in April 2013, and provides different application scenarios for various situations. One application scenario ("customized IT application (agile)") is specifically designed for the agile development of individual business software.

Today, everyone talks about the fact that IT projects have to achieve agility using agile software development methods. But what exactly does agility mean? According the Oxford English Dictionary (2014) the definition of "agile" is *"to be able to move quickly and easily"*. The business requirements in IT projects can frequently change according to internal and external influences. A high agility allows the project team to react quickly and easily to changing business requirements, and, as a consequence, to prevent high costs in later stages of the project. Nearly every book and study about agile software development describes the importance of agility and the benefits of using an agile software development method.

The agile HERMES scenario "customized IT application (agile)" combines HERMES 5 with the agile development framework SCRUM on a theoretical level. From a practical standpoint, however, it is unclear if the scenario delivers the desired agility.

Personal experience has shown me that an IT project that introduces a new and complex individual business software in a Federal Office begins independently from the chosen scenario, mostly with forms of business analysis followed by extensive requirements engineering activities. The software development isn't harmonized with previous requirements engineering - which, particularly in an agile environment, is not an advantage. Rather, the business analyst and requirements engineer work as often as they would in a waterfall model, and no common understanding about what should be delivered to the software developers exists. The agile HERMES scenario doesn't provide a single process that describes in detail how to prepare the groundwork and how to execute the requirements engineering in the case of later agile software development. HERMES 5, at its core, is a result-oriented approach, which contradicts agile approaches by way of its cumbersome documentation.

The main purpose of this master's thesis is to develop a requirement engineering process that meets both the requirements of the HERMES 5 project management method and takes advantage of the SCRUM agile software development method. The process could perhaps enable the Swiss Federal Administration to define their requirements in new software development projects more efficiently in terms of time, resources and costs, and at the same time as ensuring the necessary level of documentation.

1.2 Problem Statement

For its IT projects, the Swiss Federal Administration primarily uses the project management method HERMES 5, while for the software development part it increasingly uses the agile development framework SCRUM. The idea and concept of HERMES 5 and SCRUM are different, and hence there are challenges involved in combining them.

The agile HERMES scenario "customized IT application (agile)" performs this combination on a high theoretical level (see Figure 1). The agile software development component (Agile Development) was inserted at the right position in the procedure, but the challenges based on the combination aren't described in detail. The following sections give a rough overview of challenges, problems and points of conflict when HERMES 5 and SCRUM are in use together.

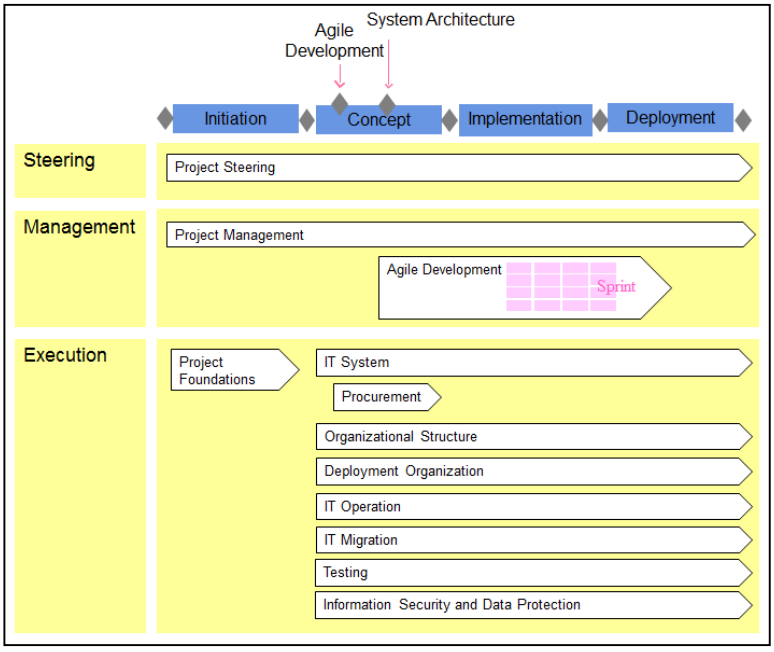


Figure 1: Agile HERMES scenario (based on (Eicher, Kruschitz & Mourgue d'Algue 2014))

The source for the problems lies in the lack of harmonization between the two methods. HERMES 5 and SCRUM have been developed autonomously and independently from one another. The underlying approaches of the two methods are different at their core. SCRUM focuses on agile software development and doesn't consider how the input for software development was selected in advance. HERMES 5, in contrast, focuses on all activities in an IT project but has its roots in the traditional waterfall approach. These circumstances make it difficult to harmonize the methods, and highlight the following main problems.

Delivery Objects/ Documentation Level: HERMES 5 is result-oriented, and the single outcomes (documents) constitute a central aspect in the execution of an IT project (Eicher, Kruschitz & Mourgue d'Algue 2014). HERMES 5 defines for the whole project lifecycle, the outcomes which have to be delivered in the particular project phases. In contrast to the other scenarios, the agile HERMES scenario provides more flexibility concerning the point of time for delivery, yet still determines which documents have to be delivered as mandatory results. SCRUM is result-oriented too, but the orientation refers to the end result of an IT project (the final working software). One value asserted by the Agile Manifesto is that a working software is more important than comprehensive documentation (Beck et al. 2001). HERMES 5 defines several mandatory documents and provides templates for their creation. SCRUM, in contrast, doesn't define any form of documentation. The delivery objects and the level of granularity is therefore a possible point of conflict when using HERMES 5 and SCRUM together. It is important to coordinate with the development to find out what they need.

Procedure: Another facet concerns procedures. As mentioned in the short introduction to this section, at its core HERMES 5 follows partially a traditional waterfall approach with four phases. The agile HERMES scenario has the four mentioned phases too, but the use of SCRUM allows it to execute the activities of the affected phases "concept", "implementation" and "deployment" at any point of time within these phases. Practical experiences have shown that the release of the phases is still handled in agile software development projects according to the traditional approach. After the finalization of a phase, the results will be approved through the project sponsor (often under the involvement of the steering committee members) and the next phase will start. In an agile approach, the phase transition should be smoother than it is handled in practice.

Involved Parties: The requirements engineering process and the software development process involve various people with different roles. For every phase, HERMES 5 defines which role has to execute the single tasks and which role has to participate in the creation of an outcome (document). SCRUM defines roles too, but only a few for the whole project. A challenge in using the agile HERMES scenario is to bring the roles of HERMES 5 into correspondence with those of SCRUM, and to

thereby secure that the responsibilities are clear. Another problem is found in the knowledge transfer between the different involved people. In IT projects in the Swiss Federal Administration, sometimes the knowledge transfer between the different people takes place only based on documentations, as not all people are involved during the whole project, which could lead back to the contractual situation with external employees. Sometimes in the phase "concept" other people are involved, because the people of the phase "initiation" have no more contract.

The above challenges, problems and points of conflict are immediately evident when using HERMES 5 and SCRUM together. In practice the points are well known, but no common solution for solving the problems exists. Every IT project invents its own solution for dealing with the problem, and synergies are not used inside of the Swiss Federal Administration. Figure 2 below gives a hypothetical illustration of the project structure of the Swiss Federal Administration. It is possible that an IT project concerns all or some Federal Departments of the Swiss Federal Administration or all or some Federal Offices inside a Federal Department. A solution respectively an extensive practical guide for how using HERMES 5 and SCRUM together would benefit all of the Swiss Federal Administration.

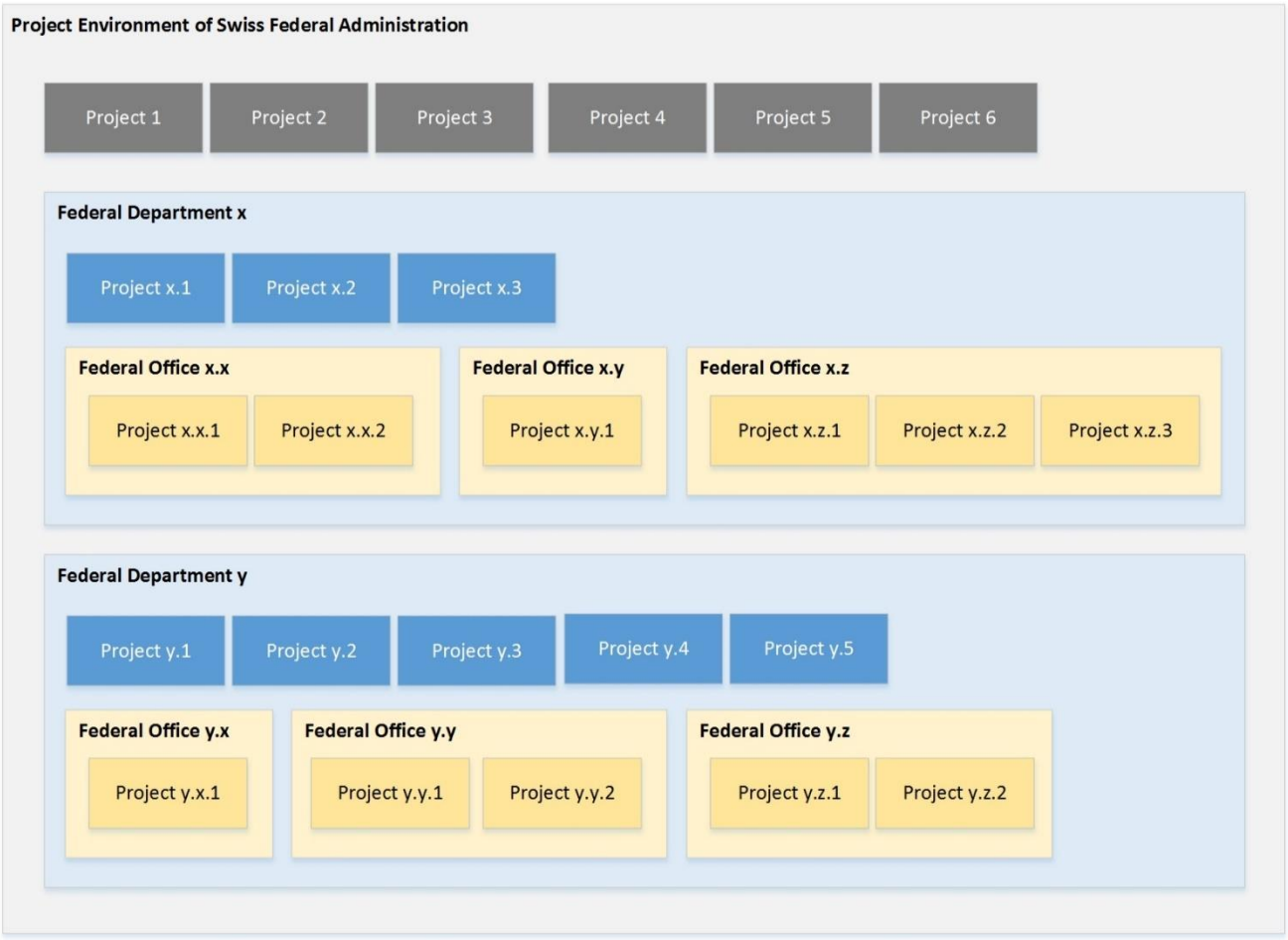


Figure 2: Project Structure Swiss Federal Administration (hypothetical illustration)

The solution on a company level does in fact bring benefits for the company, but not, in a wider context, for the whole community. Figure 3 shows the three application layers where guidelines can help to harmonize procedures. On the project layer, the level of harmonization is low, which means that every project defines its own procedure. On the middle company layer, the procedure is harmonized within one company, which means that the single departments use the same procedure for their projects. On the community level, the harmonization takes place across companies, which means that all companies in one community use the same procedure for their projects. The resolution of the problem on the framework level would bring the most benefits for the community.

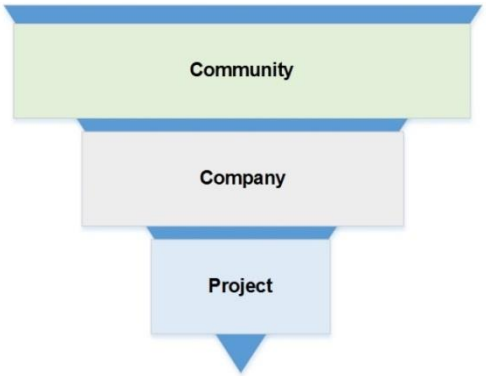


Figure 3: The thee application layer

1.3 Thesis Statement

The focus of the master's thesis is on the requirements engineering process when using HERMES 5 and SCRUM together in an IT project. The Thesis Statement (TS) is the following:

> It is possible 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.

1.4 Research Questions and Objectives

Based on the context and purpose of the research study, the primary research question (PRQ) is:

> Is it possible to develop a requirements engineering process, which both meets the requirements of the HERMES 5 project management method and takes advantage of the agile software development method SCRUM?

It is possible to split the primary research question into different sub questions, which have to be investigated in order to find an answer for the PRQ.

- Does the agile HERMES scenario meet the requirements of HERMES 5 and take advantage of SCRUM?
- How does a requirements engineering process look when using HERMES 5 and SCRUM?
- How can the roles of HERMES 5 and SCRUM be combined?
- How can the inputs and outputs of HERMES 5 and SCRUM be combined?
- How can the activities of both methods be combined?

In order to answer the PRQ and the sub questions, the following research objective has been formulated:

- To develop a requirements engineering process that meets the requirements of the HERMES 5 project management method and takes advantage of the SCRUM agile software development method.

1.5 Scope and Limitations of Scope

This chapter describes the scope of the master's thesis and explains the limitations imposed upon it. In the thesis statement below, the points that define the scope of the study are marked with a red rectangle:

> It is possible to develop a requirements engineering process, which both meets the requirements of the project management method HERMES 5 and takes advantage of the agile development method SCRUM.

In order to provide an understanding about the scope and its limitations, the next sections explain the red-marked points in the above thesis statement. Beside the red-marked points, it is worth mentioning that the focus of the master thesis is on IT projects in the Swiss Federal Administration. The study, especially the analysis part, excludes IT project from cantons, municipalities and private enterprises.

Requirements engineering process: Due to time restrictions, the research objective focuses only on the requirements engineering process in an IT project. The handling of the whole end-to-end process in depth, from the Business Analysis over Requirements Engineering to Software Development (including Testing), is too complex for one master's thesis.

HERMES 5: The Swiss Federal Administration uses HERMES 5 as a project management method in the most of its IT projects. For this reason, the study does not consider other project management methods. HERMES 5 has two different agile scenarios, "customized IT application (agile)" and "product/service (agile)". For this study, only the scenario "customized IT application (agile)" is relevant, because the study focuses on the development of individual business software.

SCRUM: A lot of agile development methods or frameworks exist (e.g. Extreme Programming, KANBAN, Crystal, Feature Driven Development, and SCRUM). The investigation of all available agile development methods or frameworks is too wide for one master's thesis. Due to the fact that the agile HERMES scenario proposes the use of SCRUM, the investigations of all the other methods would not be constructive.

The scope of the study allows it to develop a general requirements engineering process that can be used in every IT project in the Swiss Federal Administration that uses the HERMES 5 scenario "customized IT application (agile)".

1.6 Definition of Terms

This chapter explains frequently used terms and their usage in the context of the study.

Requirements Engineering: The IREB definition of Requirements Engineering (Pohl & Rupp 2011) says *"Requirements Engineering is a systematic and disciplined approach to the specification and management of requirements with the following goals:*

- *Knowing the relevant requirements, achieving a consensus among the stakeholders about these requirements, documenting them according to given standards, and managing them systematically.*
- *Understanding and documenting the stakeholder's desires and needs.*
- *Specifying and managing requirements to minimize the risk of delivering a system that does not meet the stakeholders' desires and needs."*

Business Analysis: Peter and Angela Hathaway (Hathaway & Hathaway 2014) define Business Analysis as *"The business process of assessing an organization's structure, processes, technologies and capabilities to identify and define solutions to roadblocks that impede the achievement of organizational goals. Business Analysis encompasses all activities that are necessary to study the entire organization or a specific unit thereof to identify business problems and define suitable solutions, often involving an information technology component."* In the Swiss Federal Administration the business analysis activities in an IT project are based on its own experiences, mostly involving the analysis of the single business processes to identify weaknesses and the potential to increase efficiency.

Software development: Software development is the whole process (including Business Analysis and Requirements Engineering) for developing a software product. Beside the activities of Business Analysis and Requirements Engineering, Software Development includes implementation, testing and maintenance activities. (Janssen 2014)

Agile HERMES scenario: The agile HERMES scenario "customized IT application (agile)" is not always named by name in the text. Instead of the whole name, the abbreviated "agile HERMES scenario" is used. During the creation of the master thesis, the FITSU published in June 2014 the new HERMES 5.1 release. The term "agile HERMES scenario" also corresponds to the scenario "customized IT application (agile)" in HERMES 5.1.

1.7 Assumptions

The presumption is that the Swiss Federal Administration doesn't execute IT projects for the development of new individual business software very efficiently in terms of time, resources and costs and that the level of documentation of the requirements is based on the result-oriented project management method HERMES 5, often too cumbersome for an agile software development.

1.8 Rationale of the Study

In this section, the rationale for the master's thesis is discussed from three perspectives: scientific, personal and commercial.

From a scientific perspective, the thesis takes up a previously unexplored area. One study "HERMES and Agility" (Federal IT Steering Unit 2010) does exist that investigates the combination of the old HERMES version HERMES 2003/2005 and SCRUM on a theoretical level and only for a few specific points. But the impact of using SCRUM on the whole requirements engineering process in the environment of the Swiss Federal Administration has not been explored to date.

From a personal perspective, the topic of the master's thesis represents personal interest arising from daily work. I am currently working in a big IT program in the Swiss Federal Administration on the development of individual business software for different consumption taxes. The IT program consists of different IT projects. All these IT projects have the goal to develop the individual shaping's to handle the tax collection for their specific consumption taxes. The vision of the IT program is to develop one individual business software for all consumption taxes, because nearly 80% of the core functions are the same. The IT program and the different projects use HERMES 5 as the project management method and the programming part use the agile software development method SCRUM. The challenge is that the different consumption taxes have different framework requirements (e.g. introduction date depending on a law) and the initial situation of every project is different (some have already done the specifications, other have not yet even started). The problem is precisely that the project has to provide the initial specification and doesn't begin with the requirements engineering activities. In order to stick to the introduction date, it is necessary to start development in a few months. The specifications have to be done now, and in a very short period of time, yet the development of a good working procedure for efficient requirements engineering takes time.

From a commercial perspective, the result of the master's thesis could lead to cost and time savings in future IT projects in the Swiss Federal Administration. The thesis should provide a standard requirements engineering process, which could be used in every IT project in the Swiss Federal Administration for the development of new individual business software.

1.9 Contribution of the Study

The contribution of the study is presented from two perspectives: a theoretical perspective and a practical perspective.

From a theoretical perspective, the study is worth doing, because no theory exists about how a requirements engineering process should look when using the combination of HERMES 5 and SCRUM. HERMES 5 doesn't say how to execute the requirements engineering process, it just says which outcomes have to be delivered. In practice, every IT project in the Swiss Federal Administration has to develop its own requirements engineering process. If a general requirements engineering process were to exist, it could be adapted to each and every specific project and wouldn't have to be reinvented over and over again. Due to the fact that HERMES 5 is an open standard, an additional goal is to distribute the master's thesis to the Federal IT Steering Unit as a recommendation for an extension of HEMRES 5.

From a practical perspective, the study is worth doing, because, as already mentioned in the section before and described with a practical example in chapter 1.8 "Rationale of the Study" under the section "personal perspective", there are projects in the Swiss Federal Administration that currently don't know how to execute the requirements engineering process. With the development of a requirements engineering process that both meets the requirements of HERMES 5 and takes advantage of SCRUM, it is possible to provide a general process that could be used in practice by everyone.

1.10 Outline of the Study

The master's thesis consists of six parts, wherein the fourth part corresponds to the main part of the study. The figure below shows the outline of the study. The content of the single chapters is described in connection to the figure.

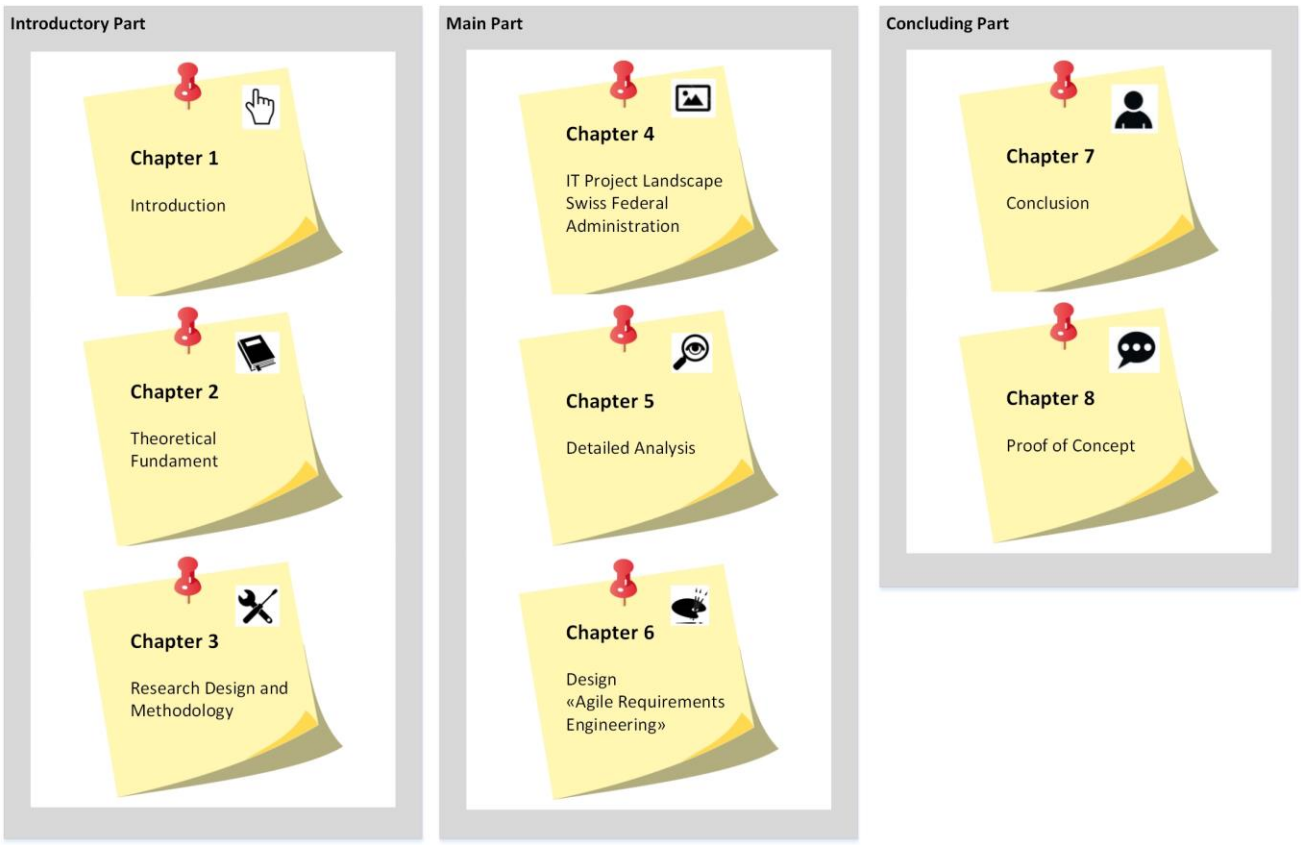


Figure 4: Outline of the study

Chapter 1: Introduction

Chapter 1 is the introduction. It contains a short introduction to the topic by giving some background information, and describes the general problem, the thesis statement and the research question and objective, providing information about the scope and its limitations.

Further, chapter 1 shows how frequently used terms are applied in the study, reveals the underlying assumptions of the study and explains its rationale and contribution.

Chapter 2: Theoretical Fundament (Literature Review)

Chapter 2 contains the theoretical foundation of the study and provides the basic information, which will be necessary later on to develop a requirements engineering process that meets both the requirements of HERMES 5 and takes advantage of SCRUM. Chapter 2 provides basic information on the topics "Agility", "Requirements Engineering", "SCRUM", "HERMES 5" and the agile HERMES scenario "customized IT application (agile)".

Chapter 3: Research Design and Methodology

Chapter 3 describes the research design and methodology. The chapter explains how to proceed in order to find an answer to the research questions.

Chapter 4: IT Project Landscape

Chapter 4 marks the beginning of the main part of the master's thesis. The chapter provides some information about the current IT project landscape of the Swiss Federal Administration.

Chapter 5: Detailed Analysis

Chapter 5 includes the detailed analysis, which consists of three parts. The first part investigates the whole agile HERMES scenario "customized IT application (agile)", and points out its general weaknesses. This first part also focuses on topics other than requirements engineering. The second part of the detailed analysis investigates how the requirements engineering is handled in the agile HERMES scenario "customized IT application (agile)". And the third part of the analysis investigates how the agile software development is solved in the agile HERMES scenario "customized IT application (agile)", as well as whether or not it corresponds to the official SCRUM process. Aside from the formal analysis sections, chapter 5 contains information on how requirements are handled today in the departments of the Swiss Federal Administration and what the expectations are regarding requirements engineering form the point of view of development.

Chapter 6: Design Agile Requirements Engineering

Chapter 6 develops an agile requirements engineering process that takes into account the findings of the detailed analysis. In addition to the description of the requirements engineering process, the chapter contains information on how the process interacts with the already existing module "Agile Development".

Chapter 7: Conclusion

Chapter 7 contains the conclusion in which the primary research question is answered and the thesis statement is justified.

Chapter 8: Proof of Concept

Chapter 8 introduces the manner in which the validity of the concept can be verified (set up for the proof of concept), gives information about the single inspectors and summarizes the inspection result. The goal of chapter 8 is to make a statement about whether the developed requirements engineering process could be used in practice as part of the agile HERMES scenario.

2. Literature Review

2.1 Introduction

The purpose of this chapter is to provide an introduction to the main concepts that are relevant for the master's thesis. The literature review should answer the following basic questions:

- What are the requirements of the HERMES 5 project management method?
- How does the requirements engineering process look today with?
- What are the advantages of the SCRUM agile software development method?
- What are the key activities in a requirements engineering process?
- Does the agile HERMES scenario meet the requirements of HERMES 5 and take advantage of SCRUM?

The literature review describes the information available concerning the topics "Agility", "Requirements Engineering", "HERMES 5", "SCRUM" and the agile HERMES scenario "customized IT application (agile)", and explains the relations and points of conflict between the topics. In additional to the points mentioned, the literature review gives an overview about existing studies in the research field to show what has and has not already been investigated.

The section on "Agility" explains the term "agility" as well as its general importance for an enterprise and how it changes the manner of software development from a sequential to an incremental and iterative method.

The section on "Requirements Engineering" explains the requirements engineering process and its main activities. For this part, the main source of information is published by the international requirements engineering board (IREB). The section furthermore shows how agility influences requirements engineering activities.

The section on "SCRUM" gives an introduction to the software development method SCRUM. It describes the idea and philosophy of SCRUM and outlines the advantages of executing a software development project with SCRUM.

The section on "HERMES 5" gives a short introduction to the project management method HERMES 5 and the requirements that have to be fulfilled in order to execute a project with this method. For the master's thesis the agile HERMES scenario has a high relevance. Therefore, one sub chapter is devoted to analyzing the agile HERMES scenario according to its concrete weaknesses and points of conflict.

2.2 Agility

2.2.1 Definition of the term "Agility"

According the Oxford English Dictionary (2014) agility is *the ability to move quickly and easily*. Alberts and Takai (2011) define agility as *the ability to successfully cope with changes in circumstances*. The combination of both definitions declares that agility is the ability to react quickly to changes that influence the current situation in order to deal with them successfully.

In the context of software development the Oxford English Dictionary (2014) defines agility as *characterized by splitting tasks into short pieces of work that have to be reassessed frequently and which lead to a permanent adaption of the plan to the current situation*. The agile alliance (2014) defines agile software development *as a method in which communication has a high significance and is delivered to the customer early and often*. Deriving from these different definitions, the term "agility" is defined as follows for the study.

> Agility is the ability to react fast and without excessive effort to changing requirements by splitting tasks into short pieces of work that allow the project team to reassess and reschedule the situation frequently.

2.2.2 Agility in Software Development

As mentioned in the background information (chapter 1.1), the concept of agile software development dates back to the early 20th century, and includes iterative and incremental methods (Perring 2010) that have been used for projects in the military and aerospace area (Basili, Victor & Larman 2003). In the scholarly literature, a few studies exist, which when taken together, give a sound and thorough overview of the history of the iterative and incremental approaches with respect to the later named agile approach ((Basili, Victor & Larman 2003), (Günal 2012), (Abbas et al. 2008)). The table below illustrates a short compilation of software development methods that have been used for iterative and incremental software development, sorted according to their founding year.

Founding year	**Method/Framework**
1980	Rapid Iterative Production Prototype
1990	Crystal
1992	Adaptive Software Development
1993	SCRUM
1994	Dynamic System Development Methods
1996	eXtreme Programming
1997	Feature Driven Development

Table 1: Overview agile software development methods/frameworks (based on (Günal 2012), (Voigt 2004), (Goyal 2007), (Case Maker Inc. & Aspects 2000))

Table 1 shows that agile methods/frameworks for software development already existed in 1980. According Abbas et al. (2008), the agile approach is not new, but has taken time to become recognized as an effective method.

In 2001 representatives from the majority of the agile methods found in Table 1, as well other stakeholders who believed in the agile approach, came together to discuss agility in the context of software development. All the participants were in agreement about the idea and philosophy of agility, and the meeting resulted in an Agile Manifesto for software development, which defines four values and twelve principles for agile software development (Beck et al. 2001). The Agile Manifesto defines the values and principles for all individual agile software development methods. According to Jeff Sutherland, agile software development is an umbrella term and not a software development method itself (Sutherland n.d.).

2.2.3 The Agile Manifesto

The list below shows the four values for agile software development contained in the Agile Manifesto. Every agile software development method has to foster these values in its own way (Sutherland n.d.).

Agile Values

The authors (Beck et al. 2001) of the Agile Manifesto write…

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

The agile principles that were defined in the agile manifesto are the following.

Agile Principles

The authors (Beck et al. 2001) of the Agile Manifesto write…

- *"Our highest priority is to satisfy the customer through early and continuous delivery of valuable software."*
- *"Welcome changing requirements, even late in development. Agile processes harness hange for the customer's competitive advantage."*
- *"Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale."*
- *"Business people and developers must work together daily throughout the project."*
- *"Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done."*
- *"The most efficient and effective method of conveying information to and within a development team is face-to-face conversation."*
- *"Working software is the primary measure of progress."*
- *"Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely."*
- *"Continuous attention to technical excellence and good design enhances agility."*
- *"Simplicity - the art of maximizing the amount of work not done - is essential."*
- *"The best architectures, requirements, and designs emerge from self-organizing teams."*
- *"At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly."*

2.3 Requirements Engineering

The goal of this chapter is to answer the question, "What are the key activities in a requirements engineering process?" The chapter gives a definition of the term "Requirements Engineering", and describes the single key activities of a requirements engineering process under in terms of both a traditional waterfall approach as well as an agile approach in order to highlight the differences. The main sources for this chapter are taken from the publications of the International Requirements Engineering Board (IREB).

Definition Requirements Engineering

The IREB definition of Requirements Engineering (Pohl & Rupp 2011) says *"Requirements Engineering is a systematic and disciplined approach to the specification and management of requirements with the following goals:*

- *Knowing the relevant requirements, achieving a consensus among the stakeholders about these requirements, documenting them according to given standards, and managing them systematically.*
- *Understanding and documenting the stakeholder's desires and needs.*
- *Specify and managing requirements to minimize the risk of delivering a system that does not meet the stakeholders' desires and needs."*

Requirements engineering is an important part of every software development project regardless of which method is used. In a traditional waterfall method, as well in an agile methods, it is always necessary to identify and document the stakeholders' needs (Grau et al. 2014). False requirements lead to wrong results and a wrong result makes the stakeholders unhappy. It is therefore very important to verify the requirements in collaboration with the stakeholders. The later a false requirement is recognized in the requirements engineering process, the higher the costs (Pohl & Rupp 2011).

For the development of a requirements engineering process that fulfills the requirements of HERMES 5 and that takes the advantages of SCRUM, it is important to understand the core activities of requirements engineering and how they differ according to the approach used (waterfall or agile). IREB published the article "Requirements Engineering and Agile Development (Grau et al. 2014)", which investigates how the discipline of requirements engineering can be made to fit an agile approach. The findings regarding the key activities make up part of the following sections. Figure 5 shows the general requirements engineering process and its core activities as they are defined by the international requirements engineering board.

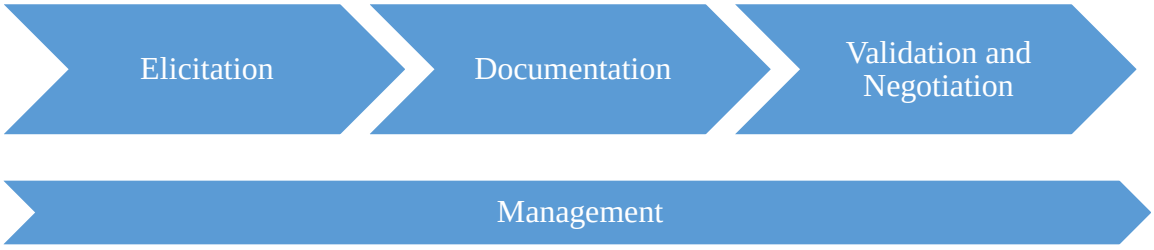


Figure 5: Requirements engineering process (based on (Pohl & Rupp 2011))

Elicitation: The elicitation phase contains the identification and elicitation of requirements. In waterfall models like HERMES 5 as it is partially, the elicitation of requirements takes place within a single project phase. In agile methods like SCRUM, the elicitation takes place continuously. The techniques are in both methods the same (Grau et al. 2014). According to the IREB the elicitation techniques are the following:

Technique	Example
Survey technique	Interviews, questionnaires
Creativity technique	Brainstorming, brainstorming paradox, change of perspective, analogy technique
Document-centric technique	System archaeology, perspective-based reading, requirements reuse
Observation techniques	Field observation, apprenticing
Support techniques	Mind mapping, workshops, CRC cards, audio and video recording, use case modelling, prototypes

Table 2: Requirements Engineering - Elicitation Techniques (based on (Pohl & Rupp 2011), (Frühauf et al. 2011))

Documentation: The documentation phase contains the activities for documenting the identified requirements using textual or conceptual models. The documentation has to be done in waterfall as well as in agile methods. The techniques and criteria for documentation are independent of the model (Pohl & Rupp 2011) (Frühauf et al. 2011) (Grau et al. 2014). The main difference in the documentation phase is the level of the documentation. The requirements in waterfall models are specified into the deep, whereas agile projects start on a higher level and go into the deep continuously. The requirements document in agile project is a dynamic document and can change from time to time (Pohl & Rupp 2011) (Frühauf et al. 2011) (Grau et al. 2014).

Validation and Negotiation: The validation and negotiation phase contains the activities for validating and negotiating the requirements together with the stakeholders. In waterfall projects the validation and negotiation takes place during or at the end of the "concept" phase. Independent of the moment, the validation and negotiation is based on every occasion on the available documentation, which contains the requirements (Pohl & Rupp 2011) (Frühauf et al. 2011) (Grau et al. 2014). In agile projects the validation and negotiation is not based only on available documents. Agile methods define a set of means (e.g. fast feedback by short iterations, definitions of done, acceptance criteria etc.), which guarantee the validation and negotiation throughout the entire lifecycle (Grau et al. 2014).

Management: The management phase is necessary during all phases. The management of requirements contains the structuring of the requirements in a repository and the maintaining of the requirements over the course of the whole project. In both waterfall and agile methods, the management of requirements is necessary, but in the waterfall method the Requirements Engineer is normally solely responsible for the management. In agile projects the management of the requirements is split among the different team members (Pohl & Rupp 2011) (Frühauf et al. 2011) (Grau et al. 2014).

In a waterfall approach the activities of the requirements engineering process are the same as in an agile approach. The difference lies in the order in which the activities are executed, on the amount of different roles that are involved in requirements engineering and in the focus on cooperation with the stakeholder. The documentation format is the same in both approaches (waterfall and agile), with the difference taking place at the abstraction level that has to be defined at the beginning of the requirements engineering process (Grau et al. 2014).

The next chapters about SCRUM (chapter 2.4) and HERMES 5 (chapter 2.5) take a deeper look into the method and the method-specific constraints about requirements engineering.

2.4 SCRUM

The goal of this chapter is to answer the question "What are the advantages of the SCRUM agile development method?" The chapter gives a short introduction to SCRUM. The first part (chapter 2.4.1) gives an overview about the history of SCRUM. The second part (chapter 2.4.2) introduces the most important elements of SCRUM. After that the advantages and stumbling blocks of introducing and using SCRUM are pointed out (chapter 2.4.3 and 2.4.4), as these should be considered for the development of the requirements engineering process. The last section (chapter 2.4.5) points out specific information concerning requirements engineering activities in SCRUM.

2.4.1 History

SCRUM was invented by Ken Schwaber and Jeff Sutherland. Both worked independently of each other on a new method for a faster way to develop software. The procedure models on which they worked were, however, very similar, which is why they later aligned with each other and collaborated on the topic. In 1999, together with other authors, they published the article "SCRUM: An extension pattern language for hyper productive software development (Beedle et al. 1999)". Sutherland and Schwaber, as well other authors, subsequently continued to publish works about SCRUM, and today SCRUM is used in fields beyond that of software development (Dräther et al. 2013).

The term "SCRUM" comes from rugby. A scrum in rugby is used to restart the game after a small breach of rules. For a layman the scrum looks very chaotic (see Figure 6). In fact, however, each position is occupied by a player with a specific role. The team members have to work together in a scrum so that the defense reaches the rugby ball that was dropped in the middle of the scrum. As soon as the defense reaches the rugby ball, they can execute a move (iSport 2014). In terms of the SCRUM agile development, this means that the whole SCRUM team works together to reach the goals.

Figure 6: rugby scrum (ZETWAL SARL n.d.)

2.4.2 Introduction to SCRUM

SCRUM is an agile framework used in software development projects as well as other projects. In the following sub chapters the most important elements of SCRUM are described in detail.

2.4.2.1 Values and Principles

SCRUM is based on the values and principles of the Agile Manifesto, which have to be applied for every agile method (see chapter 2.2.2). In addition to the values of the Agile Manifesto, SCRUM has its own foundation of five values that are described below:

SCRUM values

The SCRUM Alliance (2012) definition of the value says…

Commitment / Forecast

- *"Because we have great control over our own destiny, we become more committed to success."*

Focus

- *"Because we focus on only a few things at a time, we work well together and produce excellent work. We deliver valuable items sooner."*

Openness

- *"As we work together, we practice expressing how we're doing, and what's in our way. We learn that it is good to express concerns, so that they can be addressed."*

Respect

- *"As we work together, sharing successes and failures, we come to respect each other, and to help each other become worthy of respect."*

Courage

- *"Because we are not alone, we feel supported and have more resources at our disposal. This gives us the courage to undertake greater challenges."*

In contrast to its values, SCRUM doesn't explicitly define its own principles. The following SCRUM principles, therefore, are collected from literature.

SCRUM Principle: (Dräther et al. 2013), (Wirdemann 2011)

- Self-organization
- Self-organized team that acts on their own responsibility to realize the requirements, which were defined by the product owner.
- Empirical Process Control (inspect and adapt)
- Consecutively checks during the process (e.g. on results or how the team works together) that gives the team the possibility change or correct their techniques, methods or behavior in a way that has a positive influence on the further project work.
- Transparency
- Provide transparency to recognize occurring problems and barriers.
- Value-based Prioritization

- Realize first the functions that are most important for the customer, so that the customer has as soon as possible a visible result, which generates a high value for him.
- Time-boxing
- Realize with small periods because small pieces are always easier to guess regarding the content and scope. Secure that the periods have the same length during the whole project (max. one month).
- Fast delivery
- Develop with an iterative and incremental procedure to allow fast delivery.

2.4.2.2 Process

The chapter gives an overview and short explanation about the SCRUM process.

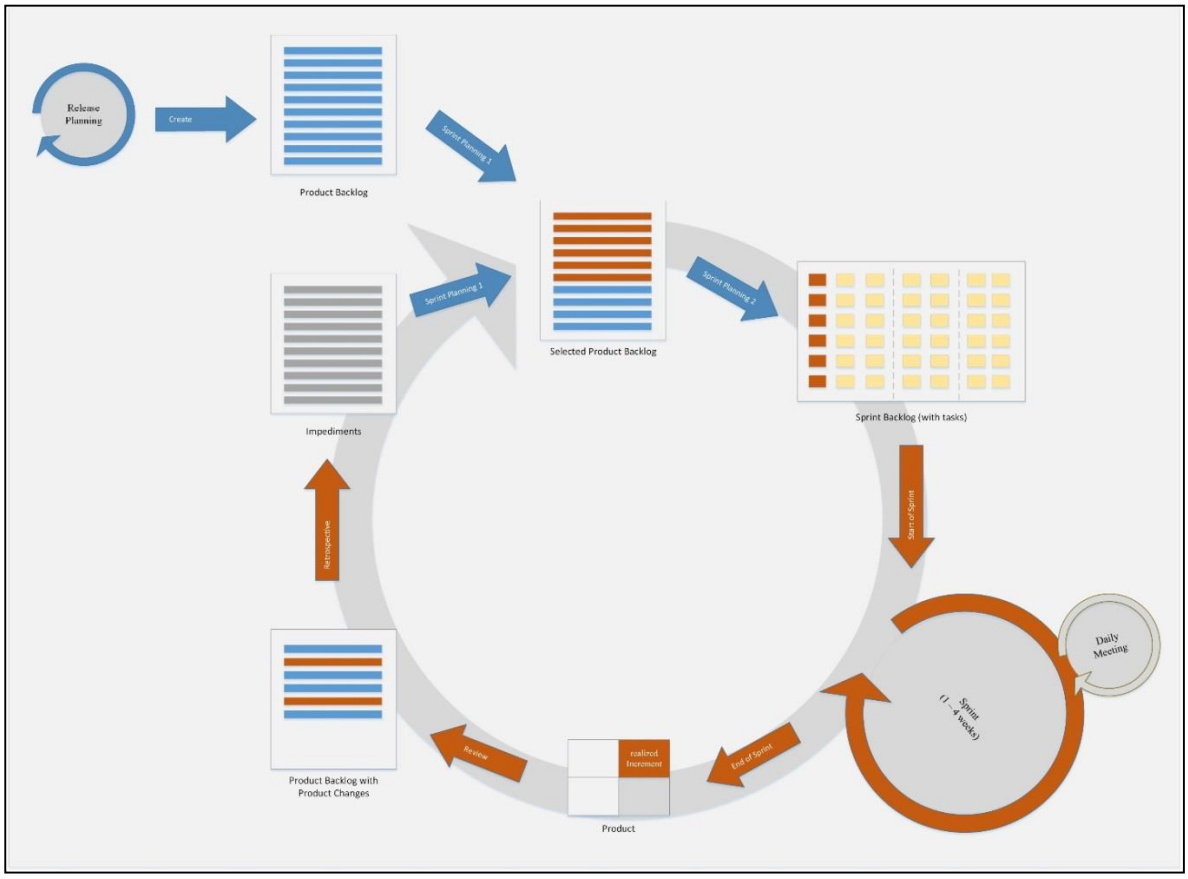


Figure 7: The SCRUM Process (based on (DAS SCRUM Team AG 2012))

Figure 7 shows the complete SCRUM Process. The precondition for initiating the SCRUM process is conceptual product development based on a vision. In contrast to a traditional waterfall approach, it is not necessary for all the requirements to be identified and documented by the time of the planning release. In SCRUM the requirements engineering activities take place iteratively and incrementally and are executed parallel to the development process (Dräther et al. 2013). The sections below describe the most important activities and results of the SCRUM Process, all of which are executed in the form of meetings in which the whole SCRUM Team participates (see chapter 2.4.2.3 for the different roles in a SCRUM team).

Release Plan (artifact)

The Product Owner is responsible for the release plan. The release plan contains the different releases and the single sprints of each release. One release can have one or more sprints. The release plan defines the start and end date of each release.

Product Backlog (artifact)

The Product Backlog contains all the requirements in the form of backlog items that are necessary for the development of the product. During the run-time of the project it is possible that backlog items change based on internal and external influences. As long as the product is under development, such a long time the backlog items can change. The Product Backlog is under the responsibility of the Product Owner. Other team members can add new backlog items but only the Product Owner is authorized to prioritize the backlog items according to importance. The Product Backlog has no structure and grows continuously. At the beginning of a project only the backlog items that are next to be realized must have a high granularity. The Product Backlog has to be accessible by all team members and allows everyone to get an insight into the scope of the project. The Product Backlog builds the precondition for the sprint planning 1. (Dräther et al. 2013), (DAS SCRUM Team AG 2012), (Wirdemann 2011)

Sprint Planning 1 (meeting)

Before the start of the first sprint, the SCRUM team plans the next sprint together. Important inputs for the sprint planning 1 are the Product Backlog, the previous velocity of the team and the current state of the product. The Product Owner presents a set of proposed backlog items for the next sprint. The team asks questions and the Product Owner is responsible for answering these questions. The goal of the sprint planning 1 is that the team understands the presented functions and can give a firm assessment about what it can deliver in the next sprint. The concrete output of the sprint planning 1 is a common understanding of the requirements, a selected Product Backlog and a formulated sprint goal (Dräther et al. 2013), (DAS SCRUM Team AG 2012), (Wirdemann 2011).

Selected Product Backlog (artifact)

The selected Product Backlog is the output of the sprint planning 1. It is nothing other than the Product Backlog with the selected backlog items for the next sprint. (DAS SCRUM Team AG 2012)

Sprint Planning 2 (meeting)

The second part of the sprint planning has the goal of defining the concrete tasks necessary to reach the sprint goal. The Product Owner cleans the task board (also called SCRUM board) of the last sprint and places the new items (e.g. user stories) for the next sprint on the task board according to their priority. In a second step, the team defines the necessary tasks. If the developers recognize that, based on the identified tasks, it is unrealistic to reach the sprint goal, they discuss it with the Product Owner and adapt the backlog items. (Dräther et al. 2013)

Sprint Backlog (artifact)

The Sprint Backlog is the output of the sprint planning 2 and contains all the backlog items and corresponding tasks, which have to be realized within the planned sprint. Normally the Sprint Backlog is represented visually as a task board. (Dräther et al. 2013), (DAS SCRUM Team AG 2012)

Sprint (activity)
The Sprint starts after the Sprint Backlog is prepared. The whole SCRUM team is responsible for realizing the tasks on the task board. One sprint has a maximum length of one month. The communication and synchronization within the team is assured by the daily meeting. (DAS SCRUM Team AG 2012)

Daily Meeting (meeting)
The SCRUM team comes together every day to present the current state of their work. Every developer explains what he did in the last days, what he is working on now and what he plans to do next. The daily meeting provides the platform to address problems and barriers that allow the SCRUM team to react immediately and to solve an occurring problem. A daily meeting should take a maximum of 15 minutes. (DAS SCRUM Team AG 2012)

Increment of the Product (artifact)
The increment is the result of a sprint. All the developed increments taken together make up the end product at the end of the project.

Review (meeting)
The review meeting contains a demonstration of the newly realized feature. The focus of the review is on the delivery content. The team members and maybe other invited persons (e.g. customer) have a look at what they did and may recognize things that have to be changed. Besides the content review, the meeting provides the team members the possibility to give feedback, which could be used as input for the next sprint planning. (DAS SCRUM Team AG 2012)

Product Backlog with Product Changes (artifact)
After the review meeting the Product Backlog is updated with product changes.

Retrospective (meeting)
The retrospective focuses on the procedure and how the team worked together in the last sprint. Based on the experience the team defines actions for the next sprint to improve the procedure and team-work. The retrospective is only for the SCRUM team, others are not allowed to participate (except invited persons). (DAS SCRUM Team AG 2012)

Impediment Backlog (artifact)
The impediment backlog contains the impediments encountered during the project. The Impediment Backlog should be discussed and up-dated during the retrospective. The Impediment Backlog should not only be a list for managing the impediments. The idea of the Impediment Backlog is to track and trace the impediments with the goal of finding a solution for each impediment (Dräther et al. 2013).

2.4.2.3 Roles

SCRUM defines only three roles, which are described below:

SCRUM Master: The SCRUM Master is responsible for the productivity of the team and the success of SCRUM. The task of the SCRUM Master is to ensure that all the team members abide

	by the rules and principles. He is responsible for making sure that single team members fulfill the responsibilities of their position. (Wirdemann 2011), (Dräther et al. 2013)
Product Owner	The Product Owner is responsible for the quality of the product. One important task of the Product Owner is to represent the needs of the customer. The Product Owner must be able to make decisions in the interest of the customer. For that he must be granted complete authority of the customer. The Product Owner is responsible for the prioritization of the product backlog and provides professional information to the team if necessary. The Product Owner decides when the increment is ready for delivery. (Wirdemann 2011), (Dräther et al. 2013)
The Team	The team consists of developers, architects, testers, usability experts etc., who are responsible for the implementation of the product and the success of every sprint. The team has the full right to undertake any measures necessary to reach the sprint goal (e.g. call an expert). (Wirdemann 2011), (Dräther et al. 2013)

2.4.3 Advantages and Disadvantages of SCRUM

This section points out the advantages of using the agile development framework SCRUM. The official literature about SCRUM mentions the following advantages (Schwaber & Sutherland 2012)

- Reduced development time
- Fast value delivery to the customer
- High quality
- Low development risk
- Higher customer satisfaction
- Higher employee satisfaction

In theory the advantages are straightforward and self-explanatory, but for the later parts of this master's thesis it is important to understand the advantages and disadvantages from a practical point of view. For this purpose I conducted a small informal survey with persons that were already involved in IT projects using SCRUM. The table below summarizes the results of the survey.

Advantages (+)	**Disadvantages (-)**
+ High motivation of the team + Common understanding based on tight collaboration between the Product Owner and the Development Team + Fast harvest of "long hanging fruits" (short-term wins), which leads to customer confidence + No long and demotivating phase for creating documents that allow space for interpretation + Early recognition of problems and faults + Trustable planning	- No dynamic and long implementation times (if SCRUM is used incorrectly) - The statement "you have not to know exactly what to do" is false. The developers have to know the system requirements for the development. - High responsibility to maintain a comprehensive overview - The writing of user stories requires both business know-how and technical know-how. These skills are rarely combined in one person.

Advantages (+)	Disadvantages (-)
+ Short implementation times (if SCRUM is used right)	- First sprints are needed to gain experience and find out the team's work pace.

Table 3: Advantages and Disadvantages of SCRUM

The findings above show that the advantages are congruent with the theory. The disadvantages show that SCRUM is in some areas perhaps too open-ended. In practice it is important to define such open-ended issues before the work begins.

2.4.4 Stumbling blocks of SCRUM

In practice, some stumbling blocks can occur in the process of introducing and using SCRUM. An open space discussion in the context of the SCRUM Product Owner certification in July 2014 lead to the following results, which could be considered as the main stumbling blocks:

Maturity level of the enterprise
The enterprise must have the necessary maturity level to introduce SCRUM, which means that all people that are involved in the project must commit to SCRUM. They must know what SCRUM is, understand the values and principles of SCRUM and know how SCRUM works. Only if these conditions are fulfilled, is it possible to successfully introduce SCRUM. Individual members of the SCRUM team that don't understand and/or don't believe in SCRUM can disrupt the process.

No correct use of SCRUM
If an enterprise decides to introduce and use SCRUM it is important that SCRUM is used correctly. It is only possible to have success if the values and principles of SCRUM are adhered to.

Insecurity of the customer
The insecurity of the customer is a big stumbling block. Especially in big projects, customers want to have security in terms of reliable statements about time, price, scope and quality. In agile software development it is not possible to give such reliable statements at the beginning of the development process because the level of granularity depends on the project status.

Ken Schwaber and Jeff Sutherland also write about enterprise barriers, which hamper the introduction of SCRUM. They identified four kinds of barriers that can occur (SCRUM Process, human factors, development practice and organizational aspects). According Ken Schwaber and Jeff Sutherland, the barriers cannot be identified at the beginning. Rather, they occur from time to time during process. A lot of the barriers can only be removed with a reorganization of the enterprise (Schwaber & Sutherland 2012).

An informal survey conducted among persons that have already worked in a SCRUM team revealed different stumbling blocks, which could be assigned to one of the main stumbling blocks. Table 4 shows that nearly all of the stumbling blocks that were observed in practice hamper the successful execution of the SCRUM Process. Human factors slow down the introduction of SCRUM.

P = SCRUM Process, HF = human factor, DP = development practice, OA = organizational aspect

General stumbling blocks	Stumbling blocks out of practical experience	Kind of barrier			
		P	HF	DP	OA
Maturity level of the enterprise	All stakeholders want to use SCRUM but they don't know how it works	x			
	The SCRUM team isn't trained at the beginning of the project (different understanding about the mechanism)	x			
	It takes time to set up a good working SCRUM team	x	x		
	It takes time to change the culture (result orientation instead of assignment of guilty)	x	x		
No correct use of SCRUM	False occupation of the SCRUM Master and the Product Owner (unclear roles and responsibilities)	x			
	Procedure and tools are not defined clearly at the beginning of the project	x			
	Product Owner is not available	x			x
	Product Owner is not able to make fast decisions	x			x
	User stories are not as precise as necessary	x		x	
	No strict leading of the Sprint Planning and Daily Meetings	x			
Insecurity of the customer	Customer insists on the approval of the requirements	x			x

Table 4: Stumbling blocks

2.4.5 Requirements Engineering in SCRUM

SCRUM doesn't define how the requirements engineering activities are executed. The Product Backlog is the central instrument for managing the backlog items, whereby a backlog item corresponds to a requirement, a function, a feature, enhancements etc. (Sutherland & Schwaber 2011). SCRUM doesn't define the form of a single backlog item.

A lot of studies on the topic recommend the use of epics and user stories. Bergsmann (2014) refers to different agile requirements engineering methods (behavior driven development, specification by example and test driven development) in order to achieve the fast and easy specification of requirements. In all these methods the necessary artifacts are drivers for software development, and can be seen as requirements.

2.5 HERMES 5

This chapter aims to answer the questions, "what are the requirements of the project management method HERMES 5?", and "how does the requirements engineering process look today with HERMES 5?" The first section (chapter 2.5.1) provides a short overview of the history of HERMES. The second section (chapter 2.5.2)

explains the most important elements of HERMES 5. And the last section (chapter 2.5.3) highlights the requirements that have to be fulfilled when using HERMES 5.

2.5.1 History

In 1970 the Swiss federal administration started with the development of its own project management method for IT projects. Five years later, in 1975, the first HERMES version was published. The name HERMES stands for "**H**andbuch der **E**lektronischen **R**echenzentren des Bundes, eine **M**ethode für die **E**ntwicklung von **S**ystemen (Eng.: manual for electronic computer centers of the swiss federal administration, a method for the development of systems)" (Federal IT Steering Unit 2013a).

The second version, HERMES 1986, was published in 1986 after a revision. Almost ten year 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 today 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. The concept of flexible adaption (tailoring) is still a main feature of the method (Federal IT Steering Unit 2013a).

2.5.2 Introduction HERMES 5

HERMES 5 is a project management method for various kinds of projects. HERMES 5 follows a modular and extensible concept, which allows it to be used in every project. The following subchapters explain the main elements of the method HERMES 5 (Eicher, Kruschitz & Mourgue d'Algue 2014).

2.5.2.1 Scenarios

A scenario provides a set of necessary methodological elements for a project with concrete characteristics. HERMES 5 provides the following standard scenarios (Eicher, Kruschitz & Mourgue d'Algue 2014) :

- Standard IT application
- Customized IT application
- Customized IT application (agile)
- IT application upgrade
- IT Infrastructure
- Service/product
- Service/product (agile)
- Organizational adjustments

HERMES 5 is an open standard and in addition to the above standard scenarios provides the possibility to create one's own individual scenario. For this study, only the scenario "customized IT application (agile)" is relevant.

2.5.2.2 Phases and Milestones

Every project consists of different phases. HERMES 5 is partially a waterfall method and provides the following four phases. It is notable that it is according to the reference handbook HERMES 5 possible to execute the phases "implementation" and "deployment" iterative.

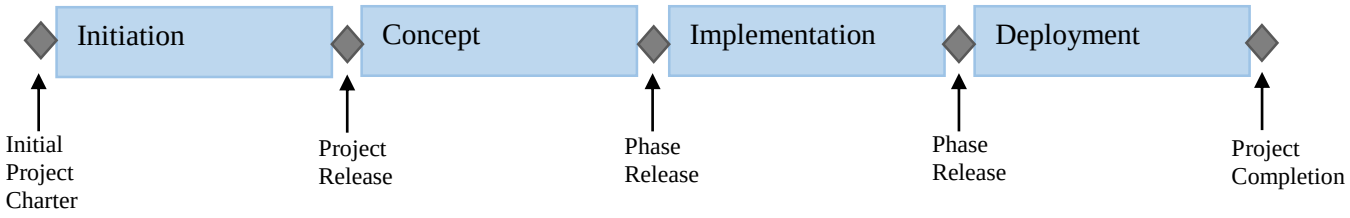


Figure 8: Project phases and milestones

Initiation: This phase investigates the initial situation of a project and provides the basic project information and the project charter for the decision concerning whether or not the project is going to be approved. The decision about the project release takes places at the end of the initiation phase. A central result in the initiation phase is the "study", which analyzes the possible solutions and gives a concrete recommendation for the preferred solution. The project charter refers to the recommended solution (Eicher, Kruschitz & Mourgue d'Algue 2014).

Concept: The concept phase substantiates the chosen solution. The results of the concept phase are used for the procurement or development of the future solution. In HERMES 5 the requirements engineering activities (as described in chapter 2.3) normally take place in the concept phase. At the end of the concept phase a decision is reached regarding whether or not the next phase is going to be approved (Eicher, Kruschitz & Mourgue d'Algue 2014).

Implementation: The implementation phase covers the development and testing of the future solution based on the results of the concept phase. The phase already contains the preparation for the deployment. At the end of the implementation phase a decision is reached whether or not the next phase is going to be approved (Eicher, Kruschitz & Mourgue d'Algue 2014).

Deployment: The deployment phase transfers the solution into the productive environment. At the end of the phase the project has to be approved. The project has to be completed with the formulation of a project end report (Eicher, Kruschitz & Mourgue d'Algue 2014).

2.5.2.3 Modules

Every scenario consists of different modules, and each module bundles the tasks and results that belong together. HERMES 5 provides the following 13 standard modules (Eicher, Kruschitz & Mourgue d'Algue 2014).

- IT System
- Procurement
- Deployment Organization

- Agile Development
- Organizational Structure
- IT Operation
- IT Migration
- Testing
- IT Security and Data Protection
- Product
- Project Management
- Project Foundation
- Project Steering

A detailed description of every module is available in the HERMES 5 reference handbook on pages 27 to 33 (Eicher, Kruschitz & Mourgue d'Algue 2014). HERMES 5 provides both the standard modules and the possibility to define individual modules.

2.5.2.4 Roles

HERMES 5 has a model that defines standard roles for the core organization and the project organization. The core organization represents the organization of the customer, whereas the project organization defines the temporary organization during the run-time of the project (Eicher, Kruschitz & Mourgue d'Algue 2014).

The core organization contains roles that can be divided into three different groups or categories. The first group is the "executive board", and it is responsible for leading the project portfolio, the project's prioritization and the allocation of the resources. The second group is the "project management competence center", and it is responsible for the allocation of methods, coaching and other project management directives. The third group is "control and compliance", and it is responsible for the definition of guidelines (Eicher, Kruschitz & Mourgue d'Algue 2014).

The project organization, as already mentioned, is a temporary organization. The project organization consists of different roles that define the tasks, competences and responsibilities of each project team member. Figure 9 shows the line and project organization, wherein the project organization displays the most important roles of a project (Eicher, Kruschitz & Mourgue d'Algue 2014). A detailed description of the different roles is available in the HERMES 5 reference handbook on pages 38 to 74 (Eicher, Kruschitz & Mourgue d'Algue 2014).

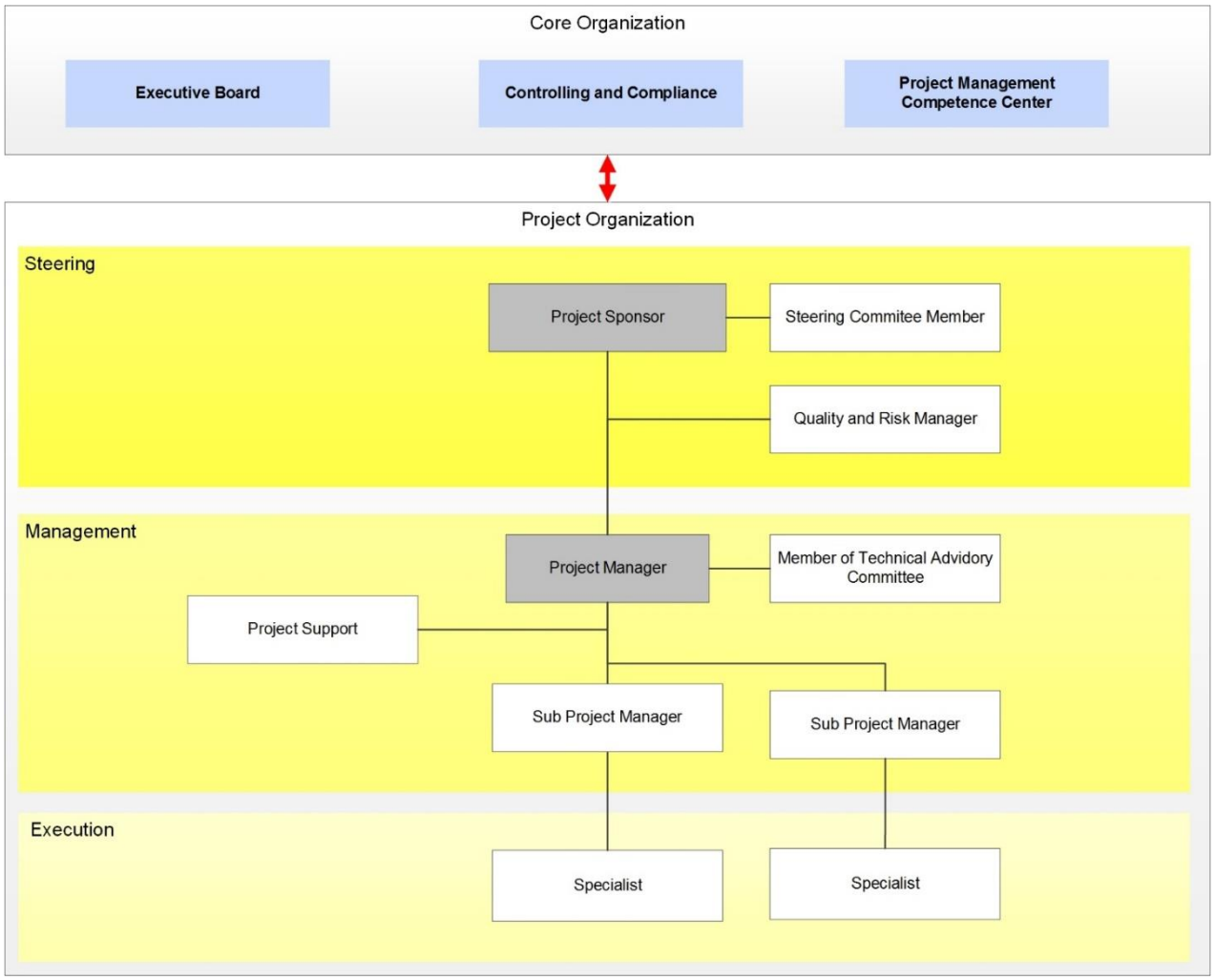


Figure 9: Core and Project Organization (based on (Eicher, Kruschitz & Mourgue d'Algue 2014))

2.5.2.5 Tasks

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 leads to a specific result. HERMES 5 defines the standard tasks that are necessary within a project (Eicher, Kruschitz & Mourgue d'Algue 2014). A detailed description of the different tasks is available in the HERMES 5 reference handbook on pages 75 to 123 (Eicher, Kruschitz & Mourgue d'Algue 2014).

2.5.2.6 Outcomes

Outcomes are the central elements of HERMES 5. HERMES 5 provides a document template for every outcome that can be used for the outcome creation. Beside a document an outcome can also be a state (e.g. IT system activated) (Eicher, Kruschitz & Mourgue d'Algue 2014).

A detailed description of the different outcomes is available in the HERMES 5 reference handbook on pages 124 to 143. Outcomes that have to be created to fulfill the requirements of the governance are marked with a * (Eicher, Kruschitz & Mourgue d'Algue 2014).

2.5.3 Requirements of HERMES 5

The Federal IT Steering Unit says that it is not possible to define the minimal requirements for a project, and that therefore every organization has to define the minimal requirements for their own projects. On the HERMES 5 website, the Federal IT Steering Unit declares that guidelines as well specific methods (e.g. Requirements Engineering) of the core organization have to be integrated in HERMES 5. The core organization can adapt the HERMES 5 elements according to their needs. It is allowed to extend or reduce elements and the outcomes can be combined or split if the content doesn't thereby change.

The HERMES 5 reference book defines some elements that cannot be left out. Table 5 points out these mandatory requirements derived from the HERMES 5 reference book.

ID	Requirement	Mandatory
1	Every project must have at a minimum the defined phase initiation, concept, implementation and deployment (creation of additional phases is allowed)	x
2	Every project must use a scenario (standard or individual scenario)	x
3	Every end of a project phase must have a milestone, which stands for the decision about the further procedure.	x
4	Every scenario has to consist of different modules (standard or individual module).	x
5	Every project has to occupy the roles project sponsor (customer), project manager and specialist.	x
6	Every project must deliver the mandatory result for the modules used (see page 125 to 127)	x

Table 5: Requirements HERMES 5 (based on (Eicher, Kruschitz & Mourgue d'Algue 2014))

The requirements in Table 5 show that HERMES 5 defines only a few requirements for the rough frame. The most restrictive requirement is the last one (ID 6), which defines the mandatory outcomes for each module.

2.5.4 Requirements Engineering in HERMES 5

A big part of the requirements engineering activities in HERMES 5 takes place in the "concept" phase (not only). The mandatory outcome "system requirements" is the output of the requirements engineering activities, whereas HERMES 5 also allows for the documentation of the system requirements in an appropriate tool. HERMES 5 defines the outcome, but it neither defines how to execute the requirements engineering activities nor how to document (textual or model-based) the requirements within the result, although HERMES 5 makes suggestions in the templates about how to document the requirements. For the outcome "system requirements" it suggests to create use cases or user stories.

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 itself, because the core organization provides no guidelines.

2.6 HERMES 5 and SCRUM

This chapter should answer the question "does the agile HERMES scenario meet the requirements of HERMES 5 and take advantages of SCRUM?" The goal of this chapter is first to give an overview of existing studies about the combination of HERMES 5 and SCRUM (chapter 2.6.1) that build the base for the agile HERMES scenario. And second (chapter 2.6.2), to analyze the agile HERMES scenario "customized IT application (agile)" in order to point out weaknesses and points of conflict.

2.6.1 Existing literature

The study "HERMES and Agility" (Federal IT Steering Unit 2010) from the Federal IT Steering Unit investigates how weaknesses that arise during the agreement between the software development and the project management can be addressed by using the techniques and methods of SCRUM. The study focuses on the following points:

- Transparent Planning
- Early fault recognition
- Distribution of the requirements
- Planning the collaboration
- Clear roles in the software development

The study shows that SCRUM can support HERMES in each of these areas. From the point of view of HERMES, it is no problem to use the methods and techniques of SCRUM. From the point of view of SCRUM, a combination between both methods isn't desirable, because the success of SCRUM is influenced by the interaction of the single SCRUM elements (Federal IT Steering Unit 2010). The study focuses only on the points here listed, and the investigations were made on a highly theoretical level.

The study "HERMES and Agility" (Federal IT Steering Unit 2010), furthermore, is not up to date according the Federal IT Steering Unit, because the study refers the old version HERMES 2003/2005 . The newest findings regarding the use of HERMES 5 and SCRUM are documented in the HERMES reference handbook (Eicher, Kruschitz & Mourgue d'Algue 2014). On the website the Federal IT Steering Unit compares HERMES 5 and SCRUM in the following points.

Influence of SCRUM on the different phases
SCRUM only influences the HERMES 5 "concept", "implementation" and "deployment" phases, whereas the whole SCRUM part is self-contained in the module "Agile Development" (Eicher, Kruschitz & Morgue d'Algue 2014).

Roles
A comparison of the SCRUM and HERMES 5 roles shows the following (Eicher, Kruschitz & Morgue d'Algue 2014):

- Candidates for the SCRUM role "Product Owner" are the HERMES 5 roles "Business Analyst", "Project Manager (customer side)", "Business Process Owner", "Product Owner" or "IT Architect"

- Candidates for the SCRUM role "Team" are the HERMES 5 roles "Developer", "Business Analyst", "Test Manager" and "Tester"
- Candidates for the "SCRUM Master" are the HERMES 5 roles "Developer" or "Business Analyst".

Task

HERMES 5 has the following tasks in the module "agile development" (Eicher, Kruschitz & Morgue d'Algue 2014):

- Decide on agile development using SCRUM
- Introduce SCRUM
- Keep a product backlog
- Design release plan
- Work in sprints

The above comparison of the HERMES 5 activities with the SCRUM Guide (Sutherland & Schwaber 2011) shows that only the tasks "keep a product backlog" and "work in sprints" are part of the SCRUM Guide (Eicher, Kruschitz & Morgue d'Algue 2014).

Method elements

Both HERMES 5 and SCRUM have defined roles. The activities in HERMES 5 correspond to the events in SCRUM and the results correspond to the artifacts (Eicher, Kruschitz & Morgue d'Algue 2014).

The comparison on the HERMES 5 website is, like the study about "HERMES and Agility", highly theoretical. The influence of SCRUM on the different HERMES 5 phases is logical, but the comparison between the roles is not reproducible. It is, for example, not explained why the SCRUM "Product Owner" does not correspond to the HERMES 5 "Product Owner", it is not clear why the HERMES 5 "Business Analyst" could be a part of the SCRUM "Team" or why the HERMES 5 "Developer" could be a "SCRUM Master". The comparison of the activities likewise takes place on a highly theoretical level, and it is unfortunate to make the comparison with the SCRUM Guide, which is a short guide of practice that defines the rules of SCRUM. Although it is true that the activities "decide on agile development using SCRUM", "introduce SCRUM" and "design release plan" are not part of the SCRUM Guide, this does not necessarily mean that using SCRUM entails ignoring these activities. Different literatures about SCRUM cover the most important points when considering whether to introduce SCRUM. And the activity "design release plan" is in fact a really important activity in SCRUM that is executed by the Product Owner. Without a release plan, no sprints and without sprints no agile development.

The points above show that the Federal IT Steering Unit has provided various advisory guides concerning the combined use of HERMES 5 and SCRUM. It is, however, initially through conversations with persons that are using HERMES 5 and SCRUM in practice that indicate that the problems often lie in the details and practical applications. In the next sub chapter the whole agile HERMES scenario "customized IT application (agile)" for the agile development of individual business software is analyzed in detail in order to find out the weaknesses and critical points of combination.

2.6.2 Analysis of the agile HERMES scenario

This chapter briefly introduces the agile HERMES scenario "customized IT application (agile)". The following figure shows the agile HERMES scenario with all its phases, modules and most important milestones. The subsequent sections analyze the elements of the agile HERMES scenario in detail.

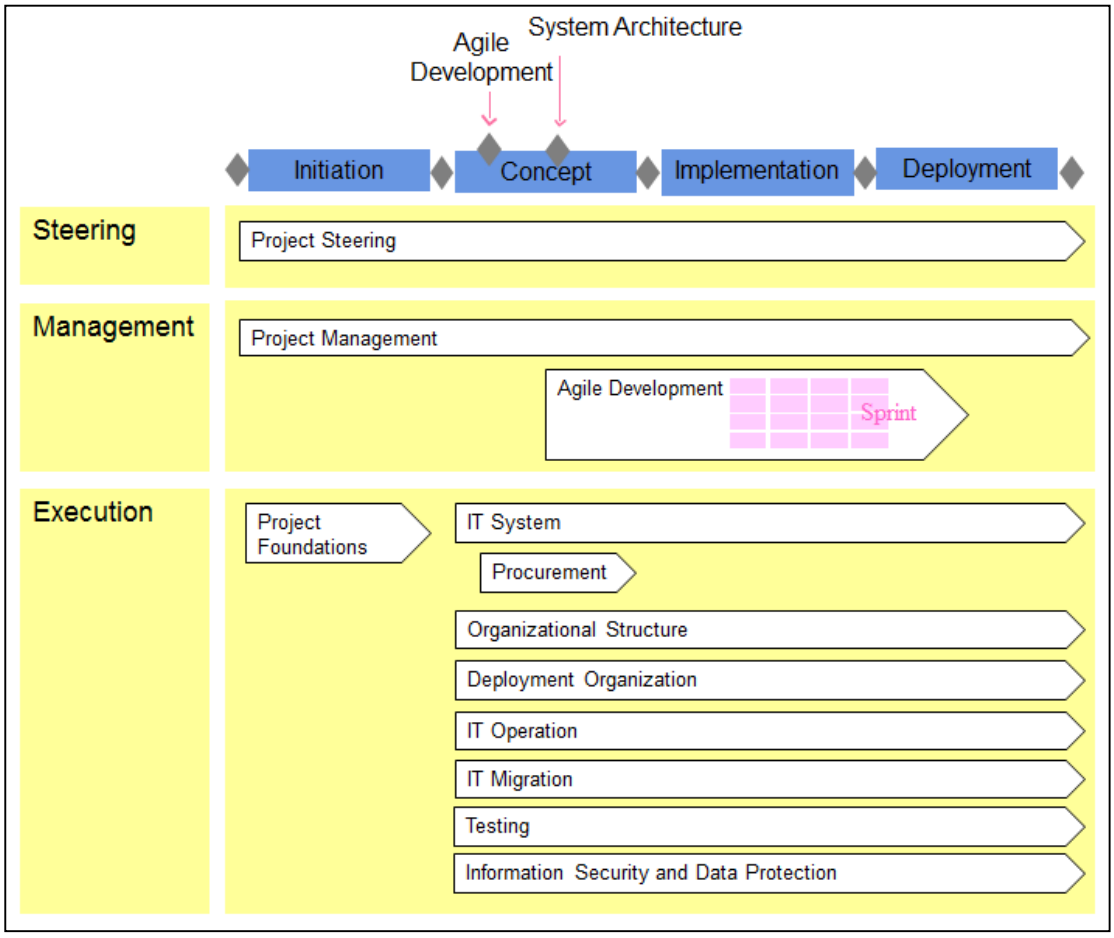


Figure 10: Agile HERMES scenario (based on (Eicher, Kruschitz & Morgue d'Algue 2014))

At the beginning of the "concept" phase the project manager makes the decision to develop according to SCRUM. This automatically leads to the insertion of the module "agile development", which merges the "concept", "implementation" and "deployment" phases into one development phase. The activities and results of these three phases can be executed/created at any time within these phases (Eicher, Kruschitz & Morgue d'Algue 2014).

The combination on this abstract level looks simple and logical, but informal interviews with people that already worked with SCRUM and HERMES 5 indicate that the combination is not as simple as this makes it seem. Based on this finding, the next chapter analyzes the agile HERMES scenario in depth.

2.6.2.1 Structural analysis

This chapter contains the results of a structural analysis of the agile HERMES scenario. The analysis takes place on the level of the tasks and outcomes. Appendix 1 in 14.1 shows a breakdown of the complete work structure

of the agile HERMES scenario. Please note that a lot of the results (e.g. project management plan) are working documents, which grow over different phases. The mandatory outcomes of HERMES 5 are marked with a star (*). The red marked points represent weaknesses or points of conflict, which are explained in the next sections.

Project charter 1

The project charter in HERMES 5 is worked out during the "initiation" phase. It is the central decision document for the project release and contains important information, which define the frame for the whole project (e.g. project goals, description of the future solution, project plan and organization, financial and non-financial instruments). The decision for the agile development is made later, at the beginning of the "concept" phase. In order to make this decision the project manager has to again work out a basis for decision-making (e.g. goals and expectations for the agile way of working, plan, role concept, risks). The time span between the project release and the decision regarding agile development is in practice mostly brief. One stumbling block is that the project sponsor (customer) can't commit to SCRUM, because he doesn't know what it is and how it works. Yet for using SCRUM successfully every project member on all hierarchical levels has to believe in the concept of SCRUM.

The early design of the project charter in the absence of information about the later agile development is a weakness. Why wait to make the decision about agile development if the project manager could make it earlier in the "initiation" phase having already worked out the project charter with the intention of using an agile development? The advantage of making the decision earlier would be that the project sponsor (customer) has to commit to the agile development with the signing of the project charter. This gives the project manager the possibility to sensitize the customer to the topic and the consequences for further project work at an early stage of the project.

Change Management 2

The management of the change management is part of the module "Project Steering", which means that it is necessary in every phase ("concept", "implementation" and "deployment"). The results of the task "manage change management" are the "amendment request" and the "change status list". Every change has to be described as an "amendment request". After that, each amendment request has to pass through the change management process, which is defined in the project management plan. Regarding the Eicher et al. (2014)SCRUM doesn't make change management unnecessary, but only fundamental changes in the scope of services lead to an extensive change management.

Change management is another point of conflict that arises from using HERMES 5 and SCRUM together. It is absolutely true that changes have to be assessed, but an assessment of every change through the customer or the responsible controlling and guidance position is not really in accordance with the nature of SCRUM. It is difficult in practice in the Swiss Federal Administration to press for a fast decision. The change advisory board holds at most one meeting per week, because the customer isn't high available for the project. With a two-week sprint using SCRUM, the agility gets lost. For the agile HERMES scenario it is important to carefully work out a change management process that allows high agility and at same time fulfills the HERMES 5 requirements in accordance with the necessary results.

Prepare release of the phase implementation 3

The release of the "implementation" phase is not a big deal according to the task description. In HERMES 5 (scenario "customized IT application") the approval of the system requirements is a very important step at the end of the "concept" phase. The development based on the approved requirements and changes in the later procedure are often very expensive. The mandatory result "system requirements" has to be worked out independently in the "concept" phase if the agile or traditional scenario is used.

The iterative and incremental development with SCRUM starts in the agile HERMES 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 on which level of granularity in order to start on the development, nor why one should work out the mandatory result "system requirements" when the product backlog later comprises the central instrument for managing the requirements. Why not start from the beginning of the "concept" phase with the product backlog?

SCRUM itself doesn't define exactly how the requirements engineering part should look before and during the development. By combining HERMES 5 and SCRUM it is necessary to develop a concrete 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 (customer) first wants to approve the requirements before the development starts. When using the agile HERMES scenario, only a set of specific requirements is specified in detail at the end of the "concept" phase. The project sponsor (customer) has to know and understand why all the requirements are not specified in detail. Only if he understands this he will begin the next phase.

Procurement → Tender documentation 4

The procurement of the supplier for development takes place at the end of the "concept" phase. If the costs for the creation and operation (for four years after deployment) are higher than CHF 230'000.00, the customer has to make a public WTO tender.

The tender documentation has to be detailed in a manner that makes it possible for the potential suppliers to make a realistic estimation. But when using SCRUM at the end of the "concept" phase only the requirements for the first releases are clear in their details. The rest of the requirements have to be worked out parallel to the development process and have a low granularity in a first step. It again comes down to the question of how detailed the documents have to be for a procurement.

Release Plan 5

In the breakdown of the work structure of the agile HERMES 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 therefore is not a necessary result. From the point of view of SCRUM, the release plan is absolutely necessary for planning the single releases and the sprints for each release.

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 of time.

2.6.2.2 Practical Challenges

A survey of people who already worked in an IT project that used HERMES and SCRUM together leads to the following practical challenges. Most of the recognized practical challenges are congruent with the previous findings in this chapter.

Practical Challenge	Description
Project Organization	The merge of the HERMES project organization and the SCRUM organization is difficult to conduct. E.g., who makes which decisions (the Project Manager or the Product Owner)?
Importance of Product Owner	The Product Owner has to represent the customer interests. The project sponsor (customer) often doesn't understand the importance of the Product Owner and is not willing to provide one of his/her own employees for the role of the Product Owner. A missing Product Owner on the business side leads to professional gaps.
Loss of Control	The Project Manager and the project sponsor (customer) have the feeling that they lost control over the project when using SCRUM.
Results	The level of granularity is mostly not clear, and the HERMES results of the concept phase are, in particular, not composed in the sense of SCRUM and have to be interpreted in another way (e.g. result "system requirements").
Task Sharing	HERMES defines clearly which activities have to be conducted by whom. In SCRUM nearly all activities are carried out as team work.

Table 6: Practical Challenges

2.7 Conclusion

The literature review has shown that only one study exists, which investigates the topic "HERMES and Agility". Unfortunately, this study focuses on only a few specific points that were investigated on a highly theoretical level and refers to the old version HERMES 2003/2005. Additional to the study already mentioned, the Federal IT Steering Unit compares HERMES 5 and SCRUM on their website. The findings are not explained in detail and are, as a consequence, not reproducible for an outsider.

The analysis of the agile HERMES scenario "customized IT application (agile)" has shown some weaknesses and critical points that arise when combining HERMES 5 and SCRUM. The following list shortly summarizes the findings:

- The project charter at the end of the "initiation" phase doesn't contain information about later agile development. But the decision for agile development has big influence on the whole project setting, which is normally defined in the project charter.
- The HERMES 5 change management can contradict the idea of SCRUM. It is important to define in agile IT project the change management under taken into account the values and principles of SCRUM.

- The delivery objects for the development are not clear. The agile HERMES scenario defines the mandatory result “system requirements”, but only makes suggestions how the content has to look.
- The level of granularity for the single delivery objects is not clear.
- The release plan is optional in the agile HERMES scenario, but for SCRUM it is absolutely necessary.
- The release of the “implementation” phase is difficult to enforce in practice when using the agile HERMES scenario. The project sponsor (customer) wants to approve the requirements, which are not all specified in detail. From the point of view of the project sponsor the requirements are incomplete.
- It is difficult to bring the project organization of HERMES 5 together with SCRUM.
- HERMES 5 defines clear responsibilities for single activities, whereas SCRUM executes as many activities as possible together in a team.

The agile HERMES scenario as well as the single methods HERMES 5 and SCRUM doesn’t define how the requirements engineering process has to look in order to support the agile development in the best way possible. HERMES 5 recommends integrating the requirements engineering process of the core organization, but unfortunately the core organization mostly doesn’t have such a standardized process. According IREB the main activities of requirement engineering are the same regardless of the approach used (agile or waterfall). The main difference is the moment when the activities are executed. The investigations confirm that it is necessary to develop a requirements engineering process that fulfills both the requirements of HERMES 5 and takes advantage of SCRUM. Based on the information acquired, it is possible to recombine it in order to develop a process that can be easily used in every IT project of the Swiss Federal Administration. The new requirements engineering process for the agile HERMES scenario must clarify the weaknesses and points of conflict that have been found for the requirements engineering part.

3. Research Design

This chapter describes the research design and methodology that will be applied for the master's thesis. The first section (chapter 3.1) describes the research philosophy, followed by the research approach (chapter 3.2) and the research strategy (chapter 3.3). The fourth section (chapter 3.4) gives an overview about the data collection methods, and the last section (chapter3.5) presents the time plan for the creation of the master's thesis.

3.1 Research Philosophy

The research philosophy describes how to arrive at the research findings. Here one faces a choice between qualitative and quantitative research methods.

Quantitative: The quantitative research philosophy aims to investigate a phenomena by collecting numerical data about this phenomena and using mathematical calculation to analyze the data and generate new findings (Balnaves & Caputi 2001).

Qualitative: The qualitative research philosophy aims to describe and understand a phenomenon by gathering information from people that are familiar with the phenomenon. In qualitative research the information has to be analyzed, compared and interpreted to generate the findings (Flick 2009).

For this study I will use qualitative research methods. The phenomenon that I want to describe and understand is the combination of the project management method HERMES 5 and the software development method SCRUM for the requirements engineering component in an IT project. In order to carry out the research in this area, it is necessary to collect information about practical experiences. This information is then analyzed and used later on for the development of a concrete solution.

3.2 Research Approach

The interpretive approach for qualitative research methodology is normally inductive, which means that the study ends with a general theory (Siegel 2002).

Figure 11 shows the inductive research approach. In the next sections, I shortly describe the concrete research approach for the master's thesis based on the general inductive research approach:

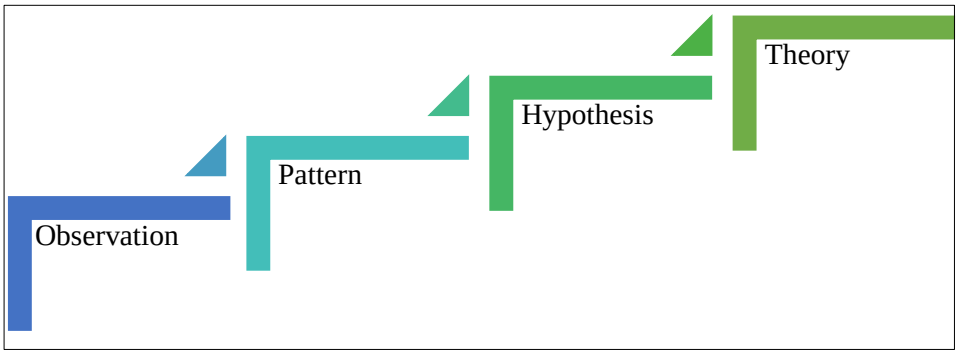


Figure 11: Inductive Research Approach (based on (Brown 2014))

Observation: In a first step I will search for more precise information about the topic. The findings in the literature review build the base, but the investigations could go more in depth on selected points. The step "Observations" contains a further literature review that goes into more depth if necessary. Additional to the selection of information, it is necessary to gain more facts about IT projects in the Swiss Federal Administration (for example to find out how many IT projects are currently running in the Swiss Federal Administration, how many of these IT projects are using the agile HERMES scenario and the reasons for project failures). Depending on the concrete point of investigation, it may be necessary to conduct more interviews to get information about the practical application.

Pattern: In the second step I will identify and analyze patterns that emerge. The patterns, for example, result from finding the same practical experience among different people that worked on different IT projects. A lot of the problems using HERMES 5 and SCRUM together have already been identified in the literature review (chapter 2).

Hypothesis: The hypothesis can already be derived from the thesis statement. The thesis statement says that it is possible to develop a requirements engineering process that meets the requirements of the project management method HERMES 5 and takes advantage of the agile software development method SCRUM. The process should enable the Swiss Federal Administration to define their requirements in new software development projects more efficiently in terms of time, resources and costs, and at the same time, insure the necessary level of documentation.

Theory: In the third step I will create the general requirements engineering process based on the information obtained. Due to the fact that it is not possible to actually carry out an IT project for proving the hypothesis, I will instead conduct a few interviews with selected professionals.

It should be mentioned that the steps "Observation", "Pattern" and "Hypothesis" have already been partially covered in the previous chapters. I will summarize and concretize the findings in the main body of the master thesis.

3.3 Research Strategy

The research strategy describes how to do the research. The following list shows possible research strategies for a qualitative research (based on (Van Der Merwe 2014))

- Observation
- Survey
- Case Study
- Focus Groups
- Action Research
- Ethnography
- Archival Research
- Design Research

The research strategy for this study is "design research". The result of the master's thesis is a requirements engineering process. The research objectives determine that the process has to meet the requirements of HERMES 5 and take advantage of SCRUM. To design a requirements engineering process according the research objectives it is necessary to consider the theoretical and practical knowledge base:

- Theoretical knowledge base:
  - Information about HERMES 5
  - Information about SCRUM
  - Information about Requirements Engineering
- Practical knowledge base:
  - Information about IT projects in the different Federal Departments
  - Used Requirements Engineering Process in the different Federal Departments
  - Practical experiences from the point of view of various project members
  - Information about already received suggestions for improvement
  - Expectation from development for requirements engineering in agile software development projects

Both the theoretical and practical knowledge base must be developed first. The development of the practical knowledge base needs especially good preparation. The following groups are intended to bring in the practical knowledge (the intended data collection methods are described in chapter 3.4).

Information provider	**Information about…**
Federal Departments: • Federal Department of Foreign Affairs (FDFA) • Federal Department of Home Affairs (FDHA) • Federal Department of Justice and Police (FDJP) • Federal Department of Defence, Civil Protection and Sport (DDPS) • Federal Department of Finance (FDF) • Federal Department of Economic Affairs, Education and Research (EAR) • Federal Department of the Environment, Transport, Energy and Communications (DETEC)	• current projects • used requirements engineering process
• Federal Office of Information Technology, Systems and Telecommunication (FOITT) • Portfolio Management of the different departments • Project Management of the Project "ICT Portfolio Management"	• current projects • used requirements engineering process
• Federal IT Steering Unit	• received suggestions for improvement
• Project Members (Project Managers, Requirements Engineers, Business Analysts, Developers, IT Architects etc.)	• Practical experience

Table 7: Intended information provider

The practical knowledge base provides important information for the future requirements engineering process, whereby, as a first step, the information is compared and analyzed with the goal of identifying common patterns. The theoretical knowledge base provides information for several principles.

As described in the problem description, the requirements engineering process for the agile software development process is handled differently in the single Federal Departments and Federal Offices (statement based on firsthand information). The goal is to develop a process for the whole Swiss Federal Administration, but in order to benefit the whole community the process should be constructed into a framework, which allows other companies to reuse the results of this study.

After the development of the requirements engineering process it is necessary to verify the result. For the purposes of verification the following options are possible, although only option 2 or 3 can be conducted during the timeframe of the master's thesis.

Option Nr. and Name	Description	Time effort for validation
1 Practical Application	Practical application in a real agile software development project in a Swiss Federal Department	Several Months or Years depending on the project size
2 Expert Review	Review of the results through experts of HERMES 5 and SCRUM. The challenge would be to find one expert, which knows both areas.	Ca. one month (incl. preparation)
3 Discussion of the topic in an expert group	Discussion of the topic in a group of experts. The experts should be an established group that is very familiar with the topic and able to assess the results professionally.	Ca. one months (incl. preparation)

Table 8: Validation options

The proof of concept has to assess the result and make a statement as to whether the findings are traceable and plausible.

3.4 Data Collection Method

For the data collection I intend to use the following methods:

Source	Method	Description	Use intention
Secondary and primary sources	Inspection of existing literature	Inspection of existing books, studies, documentations, reports, templates etc.	Information about HERMES 5 from the Federal IT Steering Unit (primary sources). Information about SCRUM from literature of the founders Jeff Sutherland and Ken Schwaber (primary sources). Information about SCRUM from secondary sources.
Primary source	E-Mail Survey	Structured questionnaire with open and closed questions, which would be sent by e-mail to the federal departments. Number of receivers: 8	Information about current projects (facts), used requirements engineering process or other given standards in relation to the topic.

Source	Method	Description	Use intention
	Online survey	Structured questionnaire with open questions. For the online survey I will use an online survey tool, which allows a comfortable answering of the questions and a simple interpretation of the result. Number of receivers: ca. 100 Goal: Experience from ca. 50 projects	Information about personal perception in relation to the topic
	Interview	Formal or informal interviews with project members.	Information about personal perception in relation to the topic
	Content Analysis	Analysis of the different requirements engineering processes, which are used in the Federal Departments (if they are available)	Reorganization of similarities between the different used requirements engineering processes
	Focus Group / Expert Inspection	Discussion of the result in a focus group or inspection of the results through experts. The goal is to receive a statement about the primary research question. Possible focus group: eCH-Standard, Section HERMES	Proof of concept of the result

Table 9: Intended data collection method

3.5 Time Plan

The following table gives an overview about the time plan for the creation of the master thesis:

Point of time	Description of milestone or working phase
31th of July 2014	Submission master thesis research proposal
August to September 2014	Detailed data collection in the specific field of study
October to December 2014	Development of the requirements engineering process based on collected information
End of December 2014	First draft of the master thesis and first proofreading.
January 2015	Final adjustment and second proofreading. Proof of concept of the results.
31th of January 2015	Submission master thesis

Table 10: Time plan

The study has a cross-sectional time horizon. This means that the information is collected during a single period of time during the study. If you have a look at the plan above you see that the detailed data collection takes place in August and September 2014.

4. IT Project Landscape of the Swiss Federal Administration

The scope of the master's thesis encompasses projects in the Swiss Federal Administration. The requirements engineering process, which has to be developed in this study, should be used in future IT projects of the Swiss Federal Administration. For this reason this chapter should give a brief impression of the IT project landscape of the Swiss Federal Administration.

According the Federal IT Steering Unit, the Swiss Federal Administration spends 1 billion (CHF 1'105 million according the state bill 2013) for the federal IT, of which one third (ca. 350 million) is invested in IT projects (Morgue d'Algue 2014) (Federal Department of Finance (FDF) 2013).

It would be interesting to see how many IT projects are currently running in the Swiss Federal Administration and the single Federal Departments and how exactly the 350 million is allocated. With the realization of the ICT Strategy of the Swiss Federal Administration 2012 -2015 this information is now since the 1 January 2015 available (Federal IT Steering Unit 2011). Under point 01 of the ICT Strategy "Controlling and Management of the ICT", the federal council stipulates that federal departments, the federal chancellery and the FITSU have to build up a federal ICT portfolio (Federal IT Steering Unit 2011). The new federal ICT portfolio is entered into effect on 1 January 2015. This means that the departments have to enter their data periodically (at least four times in a year) into the ICT portfolio management system or transmit it over an interface. (Federal IT Steering Unit 2015) (Federal IT Steering Unit 2013b). But it should be noted that the federal ICT portfolio doesn't keep data on all IT projects. Only IT projects, studies or applications with a financial investment of more than CHF 250'000.- in one financial plan year or IT projects with a total effort of more than CHF 400'000.- have to be entered in the federal ICT management system. (Federal IT Steering Unit 2013b). For this study it was unfortunately not possible to get more information about the new federal ICT portfolio.

To give an insight into the project landscape of the Swiss Federal Administration, I conducted an informal interview with the project portfolio management of the FOITT, an internal ICT provider of the Swiss Federal Administration. The FOITT is responsible for the operation of data centers and business applications (for 3 departments), the management of workstations (for four departments) and the operation of data networks and telecommunications infrastructure for the whole federal administration (BIT 2014). The FOITT has currently 63 internal projects and 93 external customer projects running (state September 2014), in which it is at the moment not possible to say how many of these projects are only software development projects that use an agile procedure (Bolley 2014).

5. Detailed Analysis

The detailed analysis aims to analyze different areas that provide a valuable input for the later design of the requirements engineering process meeting the requirements of the project management method HERMES 5 and takes advantage of the agile software development method SCRUM.

The detailed analysis investigates the following areas in detail:

- Agile HERMES scenario (chapter 5.1)
- Requirements engineering in practice (chapter 5.2)
- Expectations of the development in requirements engineering (chapter 5.3)

5.1 Analysis of the agile HERMES scenario

This chapter contains a detailed analysis of the project management method HERMES 5. The following figure shows the agile HERMES scenario. The agile module lies over the “concept”, “implementation” and “deployment” phases. Therefore, it is necessary to analyze in detail all of the modules that are part of these three phases.

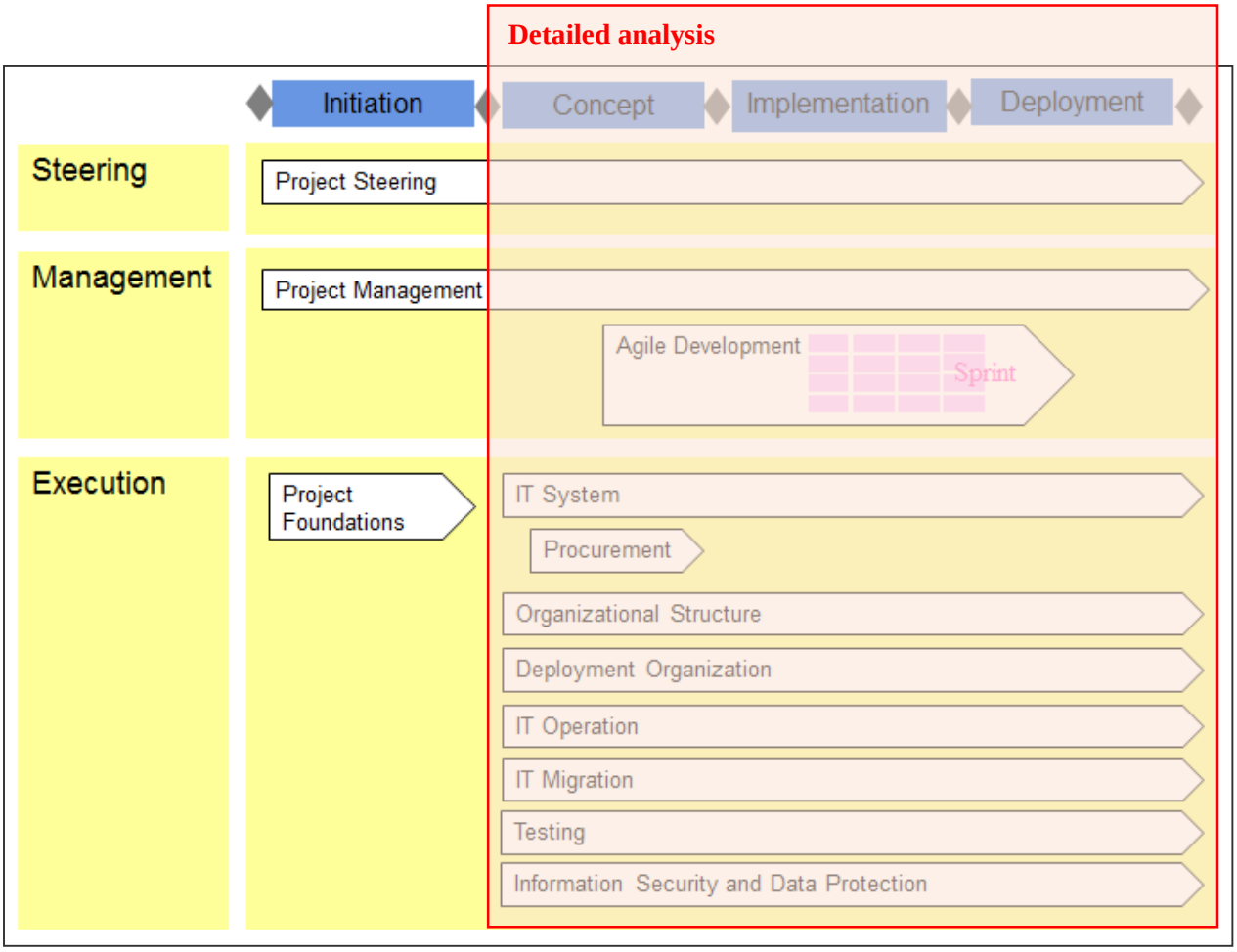


Figure 12: Investigation area HERMES5

In preparation for the detailed analysis it was necessary to take a closer look inside every module. HERMES 5 doesn't provide process models for the modules, which made it necessary to model every process based on the available information on the internet and in the reference book (Eicher, Kruschitz & Mourgue d'Algue 2014).

One module consists of different tasks. Together, these tasks build the main process for a module, wherein the single tasks represent sub processes with more than one activity. The sub processes were modeled according to the task description, which consists of a main description, key considerations of HERMES and a list of the single activities, as well as a relationship table (see Figure 13). The sequence in which the different tasks inside a module and the single activities inside a task are to be carried out are not described by HERMES 5. For this reason the sequence of a process and the sub processes were modeled based on personal project experience.

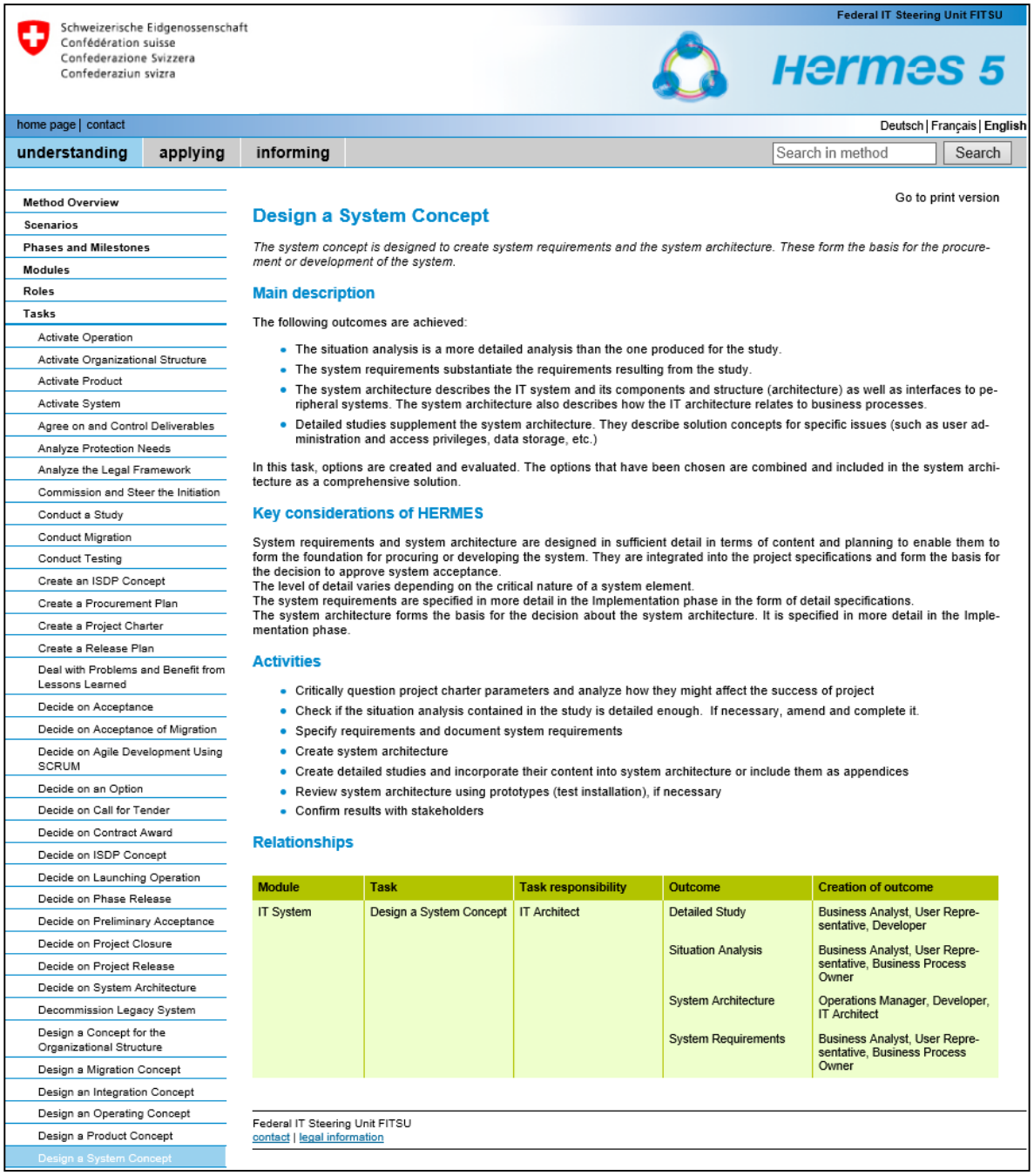


Figure 13: Task description

The detailed analysis of HERMES 5 consists of three parts. Figure 14 illustrates the initial situation.

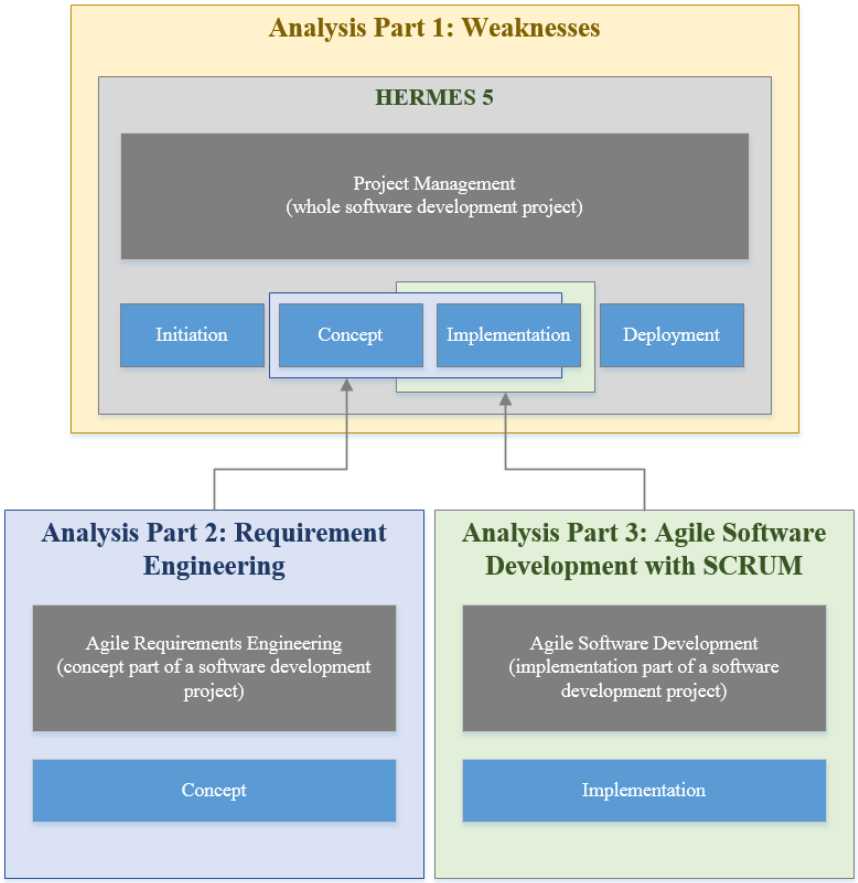


Figure 14: Areas of the detailed analysis – HERMES 5

- The first part of the analysis "Weaknesses" **(yellow marked area)** analyzes the Hermes 5 in general and points out the weaknesses and inconsistencies of the method. It should be noted that part one of the analysis could be used as a proposal for improvement independently of the results of this master's thesis.
- The second part of the analysis "Requirements Engineering" **(blue marked area)** analyzes which modules and tasks are affected by requirements engineering and investigates how the requirements engineering is executed in detail in the agile HERMES scenario.
- The third part of the analysis "Agile Software Development with SCRUM" **(green marked area)** analyzes how the module "Agile Development" handles the implementation component and if SCRUM is applied correctly in the agile HERMES scenario.

The appendix 2 in chapter 14.2 shows what tasks are part of what part of the analysis. The assignment of the tasks to the individual analysis parts has been taken based on the gained knowledge from the process modeling of each task.

5.1.1 Analysis Part 1: Weaknesses

The analysis part 1 aims to reveal weaknesses and inconsistencies of the agile HERMES scenario (investigation areas concept, implementation and deployment). The process models of the agile HERMES scenario provided the basis for the analysis. All weaknesses and inconsistencies were visually marked in the process models (see

appendix 3 in chapter 5.1.1) with an appropriate reference to the ID in the table below. **It is necessary to have a view on the process models in the appendix 3 to understand the weaknesses unequivocally.**

ID	Affect	Name	Description
1	General	Responsibility creation of outcome (result)	HERMES 5 doesn't define the responsibility for the creation of an outcome clearly. For one outcome different roles are declared for the creation, whereas in practice one role has the overall lead for the creation of the outcome. **Example 1:** Situation analysis: Business Analyst, User Representative, Business Process Responsible **Example 2:** Product Backlog: Business Analyst, Developer **Improvement proposal:** From a practical viewpoint it would be helpful if HERMES 5 defines beside the roles that are involved in the creation of the outcome also the role, which has the responsibility over the whole creation process (as suggestion).
2	General	Responsibility activity	HERMES 5 doesn't define the responsibility of the different activities of a task. This carries the danger that it is not clear from the reader's point of view who has to execute the activity within one task. **Example:** Task: implement prototype Activities: implement prototype, evaluate prototype etc. **Improvement proposal:** Define the responsible role, which has to execute the single activities of a task.
3	General	Obvious activities	In some tasks HERMES 5 mentions activities whose execution is in the nature of the thing. The explicit listing of such activities may give the impression that they are only required in this task, which is wrong. **Example:** Conduct meetings, produce minutes **Improvement proposal:** Don't mention absolutely obvious activities.

4	Project Management	Impossible multiple execution of an activity	Some tasks in the module project management contain activities, which have to be conducted in the phase initiation in the task "manage and control initiation". These are mainly activities for the definition of the process or the set up for a supporting area (e.g. change management, risk management) **Example 1:** Modul: Project Management → Task: Lead Change Management (see chapter 14.3.1.7) Activities: Decide on change process, describe and declare change process **Example 2:** Module: Project Management → Task: Manage risk Activities: Collect information about project and its environment, incorporate risk management process and risk assessment in project management plan, establish risk management process and risk assessment measures **Improvement proposal:** Delete these activities in the task description in the tasks of the phase concept, implementation and deployment.
5	General	Unnecessary listing of an activity	In the task description there is an activity, which is already implicitly covered through other activities. **Improvement proposal:** Delete the activity from the task description.
6	General	No process reuse	Another module or task already describes for the sequence of activities, or a single activity, a similar task with similar activities. **Improvement proposal:** Reuse of the activity/activities.
7	General	Outcome reference	The activity refers to an outcome, which is not declared for the corresponding task. Or: in the task description an outcome is declared, which is not assignable to an activity. Or: another outcome handles the same content. **Improvement proposal:** declare the outcome, which is mentioned in the activity for the task.

			For the second case, delete the outcome in the task description or create a corresponding activity
8	General	Stakeholder	The activity “confirm with stakeholder” gives a broad range for the activity. **Improvement proposal:** Name the concrete role (e.g. steering committee, project sponsor) if possible
9	General	Decision-makers	It is not clear, who the decision-makers are. In some cases the decision-maker is the project manager and in other cases the project sponsor or the steering committee. **Improvement proposal:** Name the concrete role (e.g. steering committee, project sponsor) if possible.
10	General	Outcome vs. milestone	Some of the outcomes are only milestones that have to be reached during the execution. ln my opinion these milestones should be declared with its own milestone element instead of the task element. Example: Prototype implemented, organization activated **Improvement proposal:** Declare these outcomes as milestones.

Table 11: Results as-is analysis

5.1.2 Analysis Part 2: Requirements Engineering

The second part of the analysis investigates the central point of departure for the study. The goal of the study is to find out if it is possible 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.

In order to be able to carry out further work, it is first necessary to find out how the requirements engineering is executed in the agile HERMES scenario. Appendix 2 in chapter 14.2 points out that the following three tasks are affected by requirements engineering:

- Design a System Concept
- Perform Quality Assurance
- Lead Change Management

The following tables analyze the above tasks in detail with a focus on requirements engineering. It is worth mentioning that the today's practical application/meaning bases on my own experience in agile software development projects in the Swiss Federal Administration.

<table>
<tr><td>Module</td><td colspan="2">IT System</td></tr>
<tr><td>Task</td><td colspan="2">Design a System Concept (see appendix 4, chapter 14.4.2.1)</td></tr>
<tr><td>Affected</td><td colspan="2">Yes</td></tr>
<tr><td>Responsible Role (execution task)</td><td colspan="2">IT Architect</td></tr>
<tr><td rowspan="2">Affected Activities</td><td>Work out system requirements</td><td rowspan="2">Today's practical application/meaning:
The business analyst is the main responsible for working out the outcomes "system requirements" and "detailed study".
HERMES 5:
Doesn't define the requirements engineering process (also no proposal available).
Define the "system requirements" as mandatory outcome.</td></tr>
<tr><td>Work out detailed study</td></tr>
<tr><td rowspan="2">Affected Outcomes</td><td>System requirements*
(Business Analyst, User Representative, Business Process Owner)</td><td rowspan="2">Today's practical application/meaning:
Often a word document that contains all system requirements. A document is often the preferred form for the acceptance.</td></tr>
<tr><td>Detailed study
(Business Analyst, User Representative, Developer)</td></tr>
<tr><td>Questions regarding execution in an agile software development project</td><td colspan="2">How would the requirements engineering process look in order to fit as best as possible to an agile development with SCRUM?
What are the appropriate delivery objects derived from the requirements engineering process in an agile software development project?
How to document the system requirements?</td></tr>
</table>

Table 12: Analysis task "Design a System Concept"

<table>
<tr><td>Module</td><td colspan="2">Project Management</td></tr>
<tr><td>Task</td><td colspan="2">Perform Quality Assurance (see appendix 4, chapter 14.4.1.1)</td></tr>
<tr><td>Affected</td><td colspan="2">Partly</td></tr>
<tr><td>Responsible Role (execution task)</td><td colspan="2">Project Manager</td></tr>
<tr><td rowspan="4">Affected Activities</td><td>Set quality objectives for the phase and the project itself</td><td rowspan="4">Today’s practical application/meaning:
In practice the main outcome of requirements engineering (system requirements) has undergo often a formal and substantive inspection.
Both kinds of inspections are often documented in writing.
The documented inspection results represent often a binding document for the project sponsor (a kind of contract, which defines what has to be delivered).
The quality manager is responsible for the set-up of the quality assurance and the formal inspection.
HERMES 5:
Doesn’t define the inspection objects and inspection procedure.
Doesn’t define that the results have to be recorded in the inspection protocol (not mandatory)
Doesn’t define the quality assurance process (no proposal available)</td></tr>
<tr><td>Establish inspection procedure for outcomes and processes/tasks</td></tr>
<tr><td>Carry out inspection</td></tr>
<tr><td>Record results of inspection in inspection report</td></tr>
<tr><td rowspan="2">Affected Outcomes</td><td>Project management plan*
(Project Manager)</td><td>Today’s practical application/meaning:
Quality manager defines objectives and procedures in project management plan.</td></tr>
<tr><td>Inspection protocol
(Project Manager)</td><td>Today’s practical application/meaning:
Quality manager writes inspection protocol for formal inspection.
Mostly more than one role writes inspection protocol for substantive inspection (depending on inspection object).</td></tr>
<tr><td>Questions regarding execution in an agile software development project</td><td colspan="2">Which outcomes of requirements engineering have to undergo an inspection?
Which inspection procedure is appropriate in an agile software development project?
How can the results of the inspection be documented, and is it necessary to document the results of an inspection in an agile software development project?</td></tr>
</table>

Table 13: Analysis task “Perform Quality Assurance”

<table>
<tr><td>Module</td><td colspan="2">Project Management</td></tr>
<tr><td>Task</td><td colspan="2">Lead Change Management (see appendix 4, chapter 14.4.1.2)</td></tr>
<tr><td>Affected</td><td colspan="2">Partly</td></tr>
<tr><td>Responsible Role (execution task)</td><td colspan="2">Project Manager</td></tr>
<tr><td rowspan="6">Affected Activities</td><td>Decide on the change process</td><td rowspan="6">Today’s practical application/meaning:
The quality manager is responsible for the set-up of the change management process and for checking if new requests are available.
Defined roles approve the amendment request depending on the classification and topic area of the request.
In the area of requirements engineering the business analyst approves, plans, implements and checks the change, often in collaboration with the quality manager and the user representative.
HERMES 5:
Doesn’t define who has to approve an amendment request.
Defines the amendment request and change status list as mandatory documents.
Doesn’t define the change management process (a proposal available).</td></tr>
<tr><td>Describe an declare change process</td></tr>
<tr><td>Approve amendment request</td></tr>
<tr><td>Plan change</td></tr>
<tr><td>Implement change</td></tr>
<tr><td>Check change</td></tr>
<tr><td rowspan="3">Affected Outcomes</td><td>Project management plan*
(Project Manager)</td><td>Today’s practical application/meaning:
Quality manager describes change process in project management plan.</td></tr>
<tr><td>Amendment request*
(Business Analyst, User Representative)</td><td>-</td></tr>
<tr><td>Change status list*
(Project Manager)</td><td>Today’s practical application/meaning:
Quality manager checks and updates the change status list.</td></tr>
<tr><td>Questions regarding execution in an agile software development project</td><td colspan="2">How should the change management process look for changing requirements in an agile software development project?
Is it really necessary to create an amendment request for each change and to update the change status list?
Who has to approve an amendment request?</td></tr>
</table>

Table 14: Analysis task “Lead Change Management”

The analysis part 2 shows that the requirements engineering activities in HERMES 5 aren't described in detail. Out of the available 54 tasks the tasks "design a system concept" from the module "IT System" is affected most by requirements engineering activities, which are represented in the task by a single activity. The agile HERMES scenario thus doesn't define a requirements engineering process that supports the later agile development with SCRUM in the best manner possible.

5.1.3 Analysis Part 3: Agile Software Development with SCRUM

The third part of the analysis aims to find out how the module "Agile Development" handles the agile software development with SCRUM (implementation part) and if SCRUM is applied as it is recommended in the SCRUM guide (Sutherland & Schwaber 2011) and other literature. For this part of the analysis the tasks of the module "Agile Development" have to be compared with the SCRUM guide. According to the appendix 2 in chapter 14.2 the following tasks are part of the analysis part 3:

- Decide on Agile Development using SCRUM
- Introduce SCRUM
- Design a Release Plan
- Keep a Product Backlog
- Work in Sprints

Module	Agile Development	
Task	Decide on Agile Development using SCRUM (see appendix 5, chapter 14.5.2.1)	
Comparison	HERMES 5	SCRUM
Responsible Role	Project Manager	Not defined
Other involved role/roles	Quality and Risk Manager	Not defined
Activities	• Clarify objectives and expectation • Determine role assignment, planning, tools • Assess effects on projects and possible risks • Update decision-making checklist with further criteria • Conduct with user, developer and operator • Make decision • Communicate decision	• doesn't define a process for reaching a decision on agile development using SCRUM (not part of the SCRUM process) • sees the decision as a precondition
Outcomes	• Checklist • Project decision management/execution	Not defined
Improvement Proposal	Precondition for the start of the module "Agile Development"	

Table 15: Analysis task "Decide on Agile Development using SCRUM"

Module	Agile Development	
Task	Introduce SCRUM (see appendix 5, chapter 14.5.2.2)	
Comparison	HERMES 5	SCRUM
Responsible Role	Project Manager	Not defined
Other involved role/roles	No other roles involved	Not defined
Activities	• Agree on details for the application • Introduce instruments and tools • Establish method of estimation • Estimate outlay • Conduct first sprints • Share understandings gained and implement improvements • Evaluation after SCRUM introduction	• doesn't define a process for introducing SCRUM (not part of the SCRUM process) • sees the introduction of SCRUM as a precondition that has to be fulfilled before the start of the project • conduct the first development sprints after the complete introduction of SCRUM (is similar to "work in sprints")
Outcomes	Project management plan	Not defined
Improvement Proposal	• Precondition for the start of the module "Agile Development" • Delete the activity "conduct first sprints". This should be part of the task "work in sprints".	

Table 16: Analysis task "Introduce SCRUM"

Module	Agile Development	
Task	Design a Release Plan (see appendix 5, chapter 14.5.2.3)	
Comparison	HERMES 5	SCRUM
Responsible role	Developer	Product Owner
Other involved role/roles	• Operation Manager • Business Analyst	• SCRUM Team
Activities	• Create Release plan • Confirm release plan with stakeholders	• The release planning is not part of the SCRUM process
Outcomes	Release plan	Release plan
Improvement Proposal	• Use the SCRUM roles.	

Table 17: Analysis task "Design a Release Plan"

Module	Agile Development	
Task	Keep a Product Backlog (see appendix 5, chapter14.5.2.4)	
Comparison	HERMES 5	SCRUM
Responsible role	Project Manager	Product Owner
Other involved role/roles	• Developer • Business Analyst	No other roles involved
Activities	• Create/update product backlog • Prioritize requirements • Analyze effects of prioritization on reaching project objectives • Monitor project scope and content • Steer new and obsolete requirements	• Part of the SCRUM process, but SCRUM doesn't define a comprehensive process for the creation of the backlog
Outcomes	Product Backlog	Product Backlog
Improvement Proposal	• Use the SCRUM roles.	

Table 18: Analysis task "Keep a Product Backlog"

Module	Agile Development	
Task	Work in Sprints (see appendix 5, chapter 14.5.2.5)	
Comparison	HERMES 5	SCRUM
Responsible role	Developer	SCRUM Team
Other involved role/roles	• Project Manager • Business Analyst	No other roles involved (all part of SCRUM Team)
Activities	• Plan sprint and document details in a sprint backlog • Develop increment • Conduct daily SCRUM meeting • Conduct sprint review • Conduct retrospective	• Comparable to the SCRUM process (see appendix 14.5.1) with the following activities: o Sprint planning 1 o Sprint planning 2 o Development o Daily SCRUM o Sprint Review o Update product backlog o Sprint retrospective
Outcomes	• Sprint Backlog • Increment • Minute	• Selected product backlog • Sprint backlog • Increment • Product backlog with changes • Impediment backlog
Improvement Proposal	• Use the SCRUM roles. • Rename the outcome "minute" in "impediment backlog"	

Table 19: Analysis task "Work in Sprints"

The analysis part 3 shows that **the agile HERMES scenario uses SCRUM correctly**, but why the SCRUM process is divided up into the tasks "keep a product backlog" and "work in sprints" is questionable. Furthermore, the agile HERMES scenario should try to **use the SCRUM roles** and try to **name the artifacts as they are defined in the SCRUM guide**. An artificial adjustment of the SCRUM process creates more ambiguities than clarity. In the terms of SCRUM, the tasks "decide on agile development with SCRUM" and "introduce SCRUM" are preconditions that have to be fulfilled before the start of the SCRUM process.

5.1.4 Conclusion

This chapter contains a short summary of the detailed analysis. The most important findings are summarized in the table below.

Analysis Part	Findings
Analysis part 1 "Weaknesses"	The agile HERMES scenario in particular, as well as HERMES 5 in general has some weaknesses that leave the reader room for interpretation (e.g. responsibilities of activities and outcomes, no clear reference of the roles). This can lead back to the fact that HERMES has to define the elements in a way that they fit for every kind of project. The identified weaknesses are seen from the perspective of the development of an individual business software and it is a proposition to overtake these changes for the other kind of projects.
Analysis part 2 "Requirements Engineering"	The main task that is affected by requirements engineering is the task "design a system concept". The agile HERMES 5 doesn't define the requirements engineering process that supports the later agile development with SCRUM in the best manner possible.
Analysis part 3 "Agile Software Development with SCRUM"	The agile HERMES scenario applies the SCRUM process in the core correctly, but is questionable if it is really necessary to force the SCRUM process and its components into HERMS 5 tasks.

Table 20: Findings detailed analysis

The findings of the analysis part 2 "Requirements Engineering" is a relevant starting point for further investigation. The agile HERMES scenario doesn't define the requirements engineering process. The single requirement from HERMES 5 is the mandatory outcome "system requirements". How this outcome has to be created or if it is really necessary for a later agile development with SCRUM has to be investigated.

5.2 Requirements Engineering in Practice

As the master's thesis follows an inductive research approach, it is necessary to investigate the practical implementation of the requirements engineering in the different departments of the Swiss Federal Administration. To achieve this, I first tried to get the information via general inquiries of the different general secretariats, with no result. In a second attempt, I asked various personal contacts who gave me the following information (no guarantee on correctness).

Department	Regulation	Details
Federal Department of Foreign Affairs (FDFA)	no	No department specific regulation available.
Federal Department of Home Affairs (FDHA)	no	No department specific regulation available.
Federal Department of Justice and Police (FDJP)	-	No information received.
Federal Department of Defence, Civil Protection and Sport (DDPS)	Partly	In the area "defense" the department has developed and integrated requirements engineering in its management processes. The bases are IREB, the requirements abstract model, HERMES, V-Model and the architecture modeling method NAF.
Federal Department of Finance (FDF)	no	No department specific regulation available.
Federal Department of Economic Affairs, Education and Research (EAR)	partly	• Requirements **must** be documented electronically (free choice of the tool) • The testing **must** consider the requirements • The whole chain from requirements engineering to testing to bug tracking **should** be without any media break
Federal Department of the Environment, Transport, Energy and Communications (DETEC)	no	No department specific regulation available.

Table 21: Results Requirements Engineering in Practice

5.3 Expectations of Development in Requirements Engineering

Besides the information about how the requirements engineering is handled today, an important element for the later design phase is input on how the development sees the role of requirements engineering in agile software development projects. To investigate this circumstance, I conducted a small non-representative online survey. The main questions of the online survey were open questions to get comprehensive explanations, and the intention was to get valuable input for the design phase and not to make a comparative analysis.

The recipients of the online survey are employees that work in the software development area. Nearly the half of the participants have the position "Developer", more than one-fourth "Leader of Development" and the rest have various functions (Test Manager, Enterprise Architect, Project Manager). The overwhelming majority (85%) of the participants work in the public sector. Taken all together, the participants have practical experience in around **46 – 62 agile software development projects**. In response to the question how they perceived the collaboration between requirements engineering and development in agile software development projects, more than half of the participates answered that the collaboration was satisfactory, nearly one-third found the collaboration good and only one-fourth found the collaboration bad.

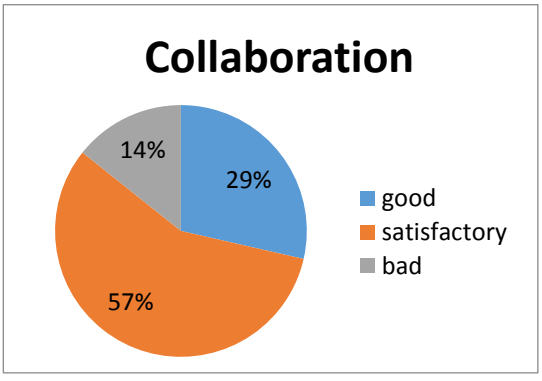


Figure 15: Collaboration Requirements Engineering and Software Development

The following list shows the reasons mentioned for the collaboration between requirements engineering and software development, sorted into positive (+) and negative (-) reasons.

+ The requirements engineers and developers knew each other for a long time, which led to less misunderstandings
+ Joint discussions led to an understanding of the other's needs
+ The Requirements Engineering clarified the feasibility continuously with the development
+ Use of epics, themes and user stories for the requirements
+ Professional formulation of the requirements using user stories
+ The development was allowed to communicate directly with the customer if necessary
- Only partly possible to divide the requirements on single backlog items
- Requirements try to force a solution
- Requirements engineering tried to capture the whole problem and tended often to comprehensive information procurement and preparation
- The role of the Product Owner was not properly perceived by the customer
- The requirements engineering tried without success to change its approach to a complete agile approach (user stories, product backlog, etc.). Classical use cases still work for the agile development
- Insufficient detail of the requirements. Requirements Engineering was neglected
- Late creation of the specification (during development sprints)
- Not all persons of the project team knew SCRUM
- Late creation and delivery of the requirements
- Time-consuming correction of requirements engineering errors that were found during the deployment

In response to the questions, "how can the collaboration between requirements engineering and development be improved" and "what is expected from requirements engineering in an agile software development project", the participants gave the following answers. The answers for the both questions were very similar, which is why the results are presented as one section.

Collaboration:

- Close collaboration between requirements engineering, development and testing (use common approach and tools). Set up as one team.
- Central management of the requirements (same base for requirements engineering, development, testing and the customer)
- Sensitize the Product Owner for the agile discipline
- Involve the requirements engineering team till the completion of the project

Method:

- Reasonable requirements engineering before the start of the implementation of SCRUM
- Introduce an agile Change Management
- Use a top down approach for the requirements (from epics to user stories)
- Clear regulations
- Continuous integration of the requirements

Delivery Objects:

- Early technical verification of the requirements through the development and the customer.
- On time completion of the requirements (at least before the sprint)
- Clear formulation of the requirements (e.g. don't use abbreviations, avoidance of ambiguity)
- Only describe the "what" in requirements engineering (not the "how"). Professional formulated requirements (business focus, no technical focus)
- Transfer the requirements into the product backlog after the verification
- Use epics, user stories and screen mocks (inclusion of an usability expert)
- Create in team sketches during discussions
- Use visual artifacts alongside textual requirements
- Describe the normal path, alternative path and errors

6. Design "Agile Requirements Engineering"

This chapter corresponds to the main part of the master's thesis in which I develop a requirements engineering process that meets the requirements of the project management method HERMES 5 and takes advantage of the agile software development method SCRUM.

6.1 Initial Situation

This chapter summarizes the findings of the previous chapters. These findings contain the input for the design of the requirements engineering process.

Summary of the findings
The requirements of HERMES 5 (details see chapter 2.5.3) are nearly all defined at a high level and don't affect the design work.
The weakness analysis (for details see chapter 5.1.1) points out the following main weaknesses that should be avoided in the creation of the requirements engineering process: • No clear definition of the responsibilities for the creation of the outcomes and the execution of the single activities. • No use of free interpretable terms • No use of unnecessary or obvious activities, that are not essential to the success of the process
The agile HERMES scenario doesn't define the requirements engineering process (especially not for a later agile development)
SCRUM is generally applied correctly in the agile HERMES scenario, but there are still some small differences.
No regulations exist for the topic requirements engineering in the different Federal Departments.
Different expectations from the development side on the requirements engineering in agile software development projects. These expectations constitute a valuable input for the design of the requirements engineering process.

Table 22: Summary analysis result

6.2 Embedding

The requirements engineering process should meet the requirements of the project management method HERMES 5 and take advantages of the agile software development method SCRUM. The target group that should use the requirements engineering process is the Swiss Federal Administration. For this reason, the requirements engineering process should be embedded into the agile HERMES scenario (customized IT Application (Agile). Figure 16 below shows how the requirements engineering process is embedded in the agile HERMES scenario. Instead of the module "IT System", there is the new module "Agile Requirements Engineering", which interacts continuously with the module "Agile Development".

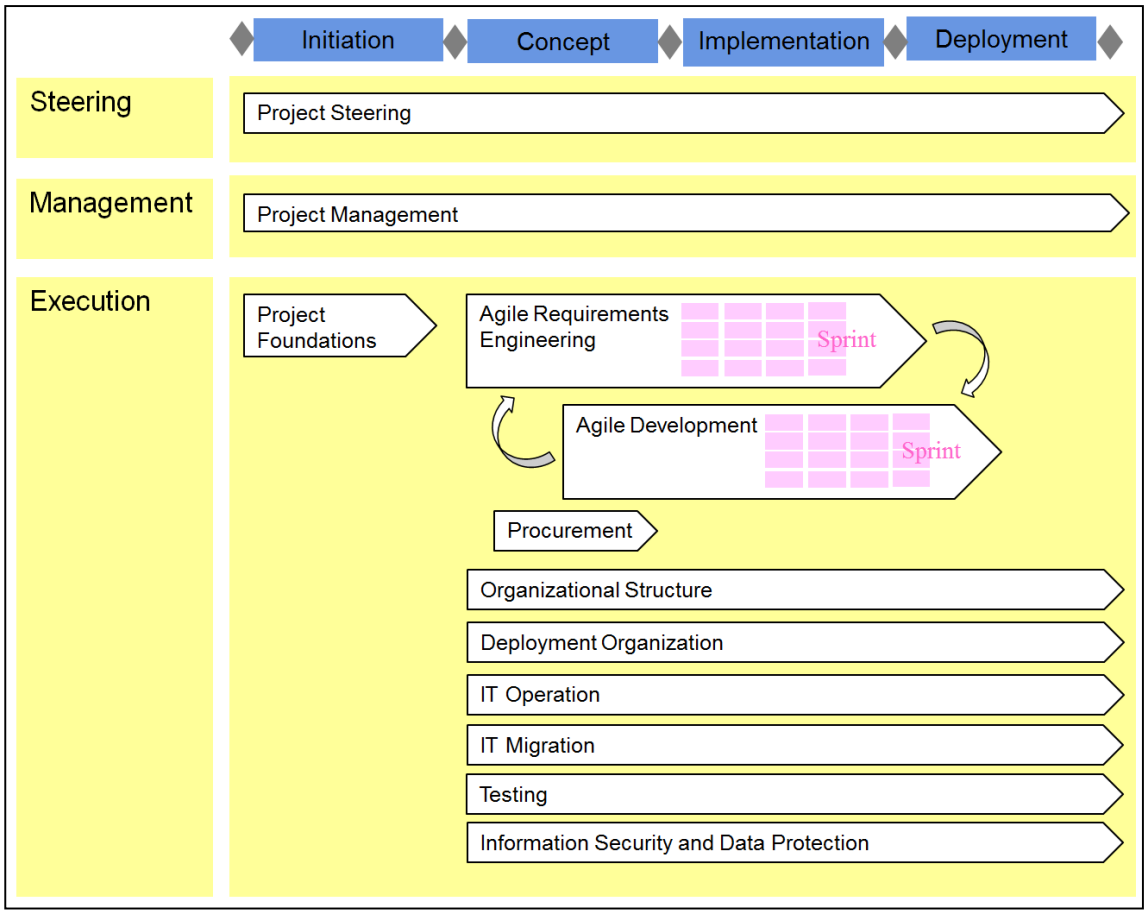


Figure 16: Embedding Requirements Engineering in the agile HERMES scenario

In the following chapters the changes of the adjusted “agile HERMES scenario” are described in detail. These changes affect the following modules:

- Project Management (changes)
- Agile Requirements Engineering (new)
- Agile Development (changes)

The other scenarios were not adjusted, because they are not directly influenced by agile requirements engineering. In the context of the agile scenario there are further possibilities to make the scenario more agile (e.g. agile testing), but this is beyond the scope of this work and would be a diversion from the focus on the main topic, the requirements engineering with HERMES 5 and SCRUM.

6.3 Changes on the “agile HERMES scenario”

Figure 17 gives an overview of the modules of the adjusted agile HERMES scenario. You see on the first view that three tasks that were originally in the module “Agile Development” are now in the module “Project Management”. The module “Agile Development” now only consists of two tasks corresponding to the official SCRUM process. The module “Agile Requirements Engineering” has not been affected by task movements, but there are some tasks, which could be deleted. The task “design a system concept” contains the main requirements engineering activities, and its execution is described in detail in chapter 6.3.3.

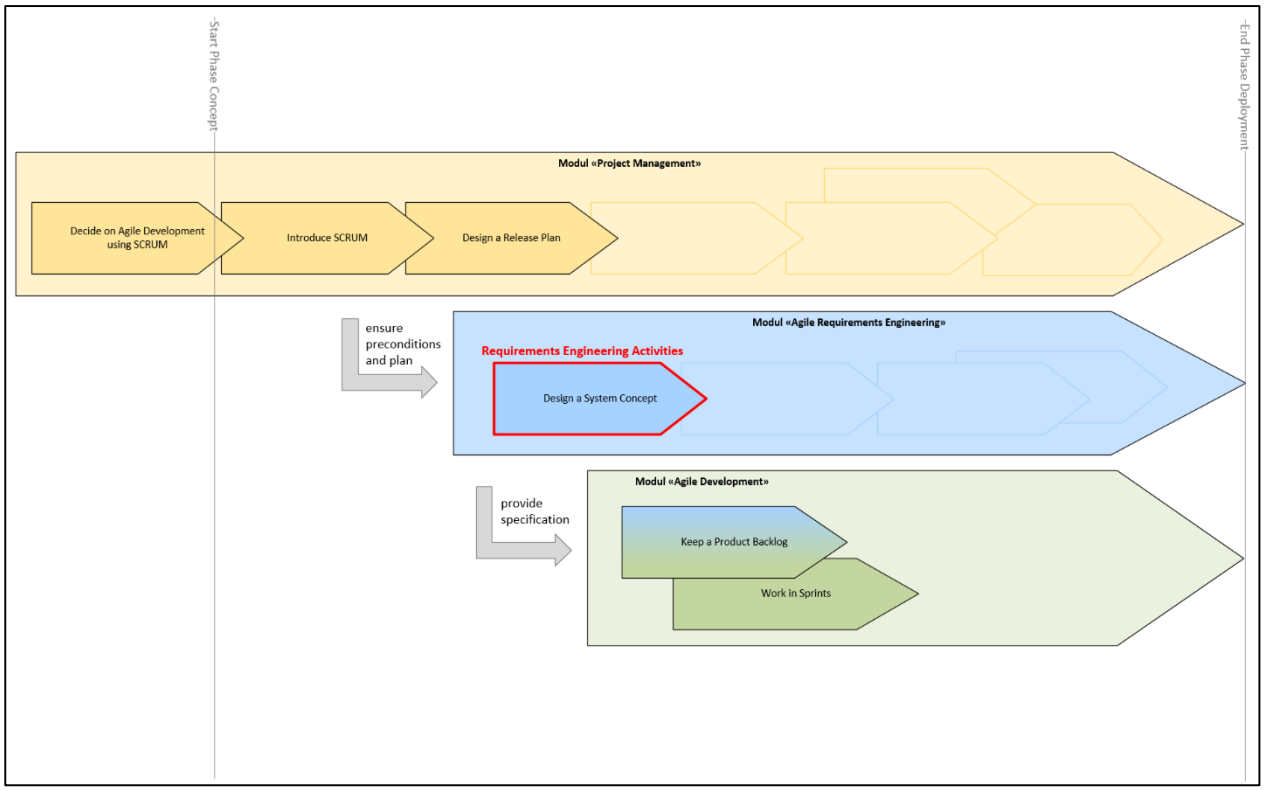


Figure 17: Overview affected modules and tasks

6.3.1 Changes in Module "Project Management"

The detailed analysis showed that the tasks "decide on agile development using SCRUM", and "introduce SCRUM", are not part of the SCRUM process. They are preconditions that have to be fulfilled before the start of the agile development.

Initiation	Concept	Implementation	Deployment
Manage and Control Initiation	**Introduce SCRUM**		
	Manage and Control a Project	Manage and Control a Project	Manage and Control a Project
Decide on an Option	Agree on and Control Deliverables	Agree on and Control Deliverables	Agree on and Control Deliverables
Create a Project Charter	Deal with Problems and Benefit from Lessons Learned	Deal with Problems and Benefit from Lessons Learned	Deal with Problems and Benefit from Lessons Learned
Decide on Agile Development using SCRUM	Manage Stakeholders and Communication	Manage Stakeholders and Communication	Manage Stakeholders and Communication
	Perform Quality Assurance	Perform Quality Assurance	Perform Quality Assurance
	Manage Risks	Manage Risks	Manage Risks
	Lead Change Management	Lead Change Management	Lead Change Management
	Prepare Phase Release	Prepare Phase Release	Prepare Project Closure
	Create a Release Plan	**Create a Release Plan**	

Figure 18: Task Overview "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 how the implementation is done, but also on the requirements engineering activities. For this reason the task

"decide on agile development using SCRUM" must be taken up at an early stage in the module "Project Management". As visualized in Figure 17 the task should be moved additionally from the beginning of the "concept" phase to the end of the "initiation" phase. The structural analysis of the whole agile HERMES scenario (see chapter 2.6.2.1) shows that the time span between the project release based on the "Project Charter" and the decision for agile development is often short. It is to be recommended that, if possible, the decision for agile development with SCRUM is made early enough that the "Project Charter" can consider this circumstance. A big advantage is also that the Project Sponsor (customer) has to commit to agile development with the sign of the "Project Charter".

The task **"introduce SCRUM"** was originally executed in the module "Agile Development" as well. But after a decision for agile development using SCRUM has been made, to define the tools/instruments, procedure etc. is not the only important task for the development side. It is equally important to do the same for the requirements engineering side and to define the collaboration between requirements engineering and development. This includes common definitions 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, direct in the product backlog etc.). A very important point is that the training of the project team members (including customer side) is missing in the task "introduce SCRUM". It is important that all involved parties know what SCRUM is and how it works. At the very least I recommend deleting the activity "conduct first sprints" in the task, because it corresponds to the task "work in sprints". Accordingly, 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 moved is the task **"design a release plan"**. This task was originally part of the module "Agile Development". Practical experience from the development side showed that the requirements engineering activities are often executed too late during the sprint. The release plan should not only build the basic plan for the deployment phase, but also for the concept phase as well. 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 to 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, which is adjusted in every sprint of the requirements engineering and in the sprints of the developments. The creation of the release plan is the responsibility of the Project Manager, but he has to involve the SCRUM Product Owner, the Business Analyst, The SCRUM Master and the Developers.

6.3.2 Changes in the Module "Agile Development"

After the above mentioned changes, the module "Agile Development" consists only of the tasks **"keep a product backlog"** and **"work in sprints"**, which correspond to the SCRUM theory taking into account the few proposals for improvement in chapter 5.1.3.

Initiation	Concept	Implementation	Deployment
	* ~~Decide on Agile Development Using SCRUM~~ * ~~Introduce SCRUM~~ Keep a Product Backlog * ~~Create a Release Plan~~ Work in Sprints	Keep a Product Backlog * ~~Create a Release Plan~~ Work in Sprints	Keep a Product Backlog * ~~Create a Release Plan~~ Work in Sprints

***move to module "Project Management"**

Figure 19: Task Overview "Agile Development"

It should be noted that the task "keep a product backlog" belongs to the module "Agile Development", but also supports the module "Agile Requirements Engineering". For this reason, the task is against the improvement proposal in chapter 5.1.3 not integrated into the task "work in sprints".

6.3.3 New Module "Agile Requirements Engineering"

The module "Agile Requirements Engineering" consists of nearly the same tasks as the original module "IT System".

Initiation	Concept	Implementation	Deployment
	Design a System Concept ~~Implement Prototype~~ Design an Integration Concept Decide on System Architecture	~~Implement Prototype~~ Implement System Prepare System Integration	Activate System

Figure 20: Task Overview "Agile Requirements Engineering"

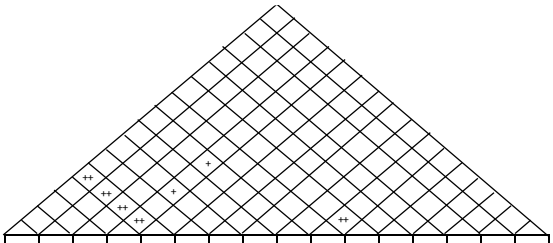
The task **"implement prototype"** covers the specification, implementation and evaluation/documentation of the prototype. In an agile software development project, the creation of a prototype should also use the agile procedure. For this reason the part of the specification the prototype can be handled as well as over the tasks in the concept phase. 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 absolutely necessary to list this separate task.

The activities of the task **"implement system"** are, like the activities of the task "implement prototype", partly covered by other existing tasks. The activity "work out the detailed specification", in particular, takes part in agile development already present in the task "design a system concept", or later during the sprints.

The last mentioned point 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 "design a system concept". The goal of this master's thesis is to make concrete suggestions of how requirements engineering could look. The next sub chapters treat the task "design a system concept" in detail. For the development of an agile requirements engineering process

it is necessary to identify on one side the customers' needs, and on the other side, the capabilities of the agile requirements engineering framework. To achieve this I created a house of quality, which shows the relation between the customer needs and the capabilities in the context of the agile requirements engineering.

The inputs for the customers' needs come out of the detailed analysis in chapter 5. The design requirements come out of an analysis of the different functions of the two methods with respect to their frameworks.



Direction of Improvement				▲			▲	▲	▲
Customer Requirements (What) / Design Requirements (How)		Importance (1-5)	Importance (%)	application hints	search function	release information	transparent change management (community)	standard reports	expert help
easy handling	fast finding of the neccessary information	5	7,46	1	9	1	1	3	1
	clear navigation structure	4	5,97	1	1	1	1	1	1
operational readiness	low effort for training	2	2,99	3	1	1	1	1	9
	short introduction time	3	4,48	3	1	1	1	1	1
customization	easy customization	5	7,46	9	3	1	1	1	3
multilinguism	multi language support	2	2,99	1	1	1	1	1	1
intelligibility	clear description	5	7,46	1	3	1	1	1	1
	clear declaration of responsibilities	5	7,46	1	1	1	1	1	1
results	fast visibility of a result	5	7,46	1	1	1	1	1	1
	understandability of the result	5	7,46	1	1	1	1	1	1
	early recognition of specification faults	4	5,97	1	1	1	1	1	3
resources	save costs	3	4,48	1	1	1	1	1	1
	save time	3	4,48	1	1	1	1	1	1
	save personal resources	3	4,48	1	1	1	1	1	1
standard documentation	reuse of the results	4	5,97	3	1	1	1	3	1
	comparability of the results	4	5,97	3	1	1	1	3	1
organizational structure	lean organization	5	7,46	1	1	1	1	1	1
Assessment		**67**	**100**	108	102	62	62	88	96
Rank				8	9	15	15	12	10

Direction of Improvement																				
Customer Requirements (What) / Design Requirements (How)		Importance (1-5)	Importance (%)	comprehensive overview	overview section "tasks"	overview section "roles"	overview section "outcomes"	cross-section interaction	availability of the documentation	in the context embedded templates	transparent language switch	download assistance for templates	assistant for customization
easy handling	fast finding of the neccessary information	5	7,46	9	9	9	9	9	3	9	1	1	1
	clear navigation structure	4	5,97	9	9	9	9	9	3	3	3	3	3
operational readiness	low effort for training	2	2,99	3	9	9	9	3	3	3	1	1	1
	short introduction time	3	4,48	3	3	3	3	3	3	3	1	1	3
customization	easy customization	5	7,46	1	1	1	1	1	1	1	1	1	9
multilinguism	multi language support	2	2,99	1	1	1	1	1	1	1	9	1	1
intelligibility	clear description	5	7,46	3	3	3	3	1	1	3	1	1	1
	clear declaration of responsibilities	5	7,46	1	9	9	9	3	1	1	1	1	1
results	fast visibility of a result	5	7,46	1	1	1	1	1	1	1	1	1	1
	understandability of the result	5	7,46	1	1	1	1	1	1	1	1	1	1
	early recognition of specification faults	4	5,97	1	1	1	1	1	1	1	1	1	1
resources	save costs	3	4,48	1	1	1	1	1	1	1	1	1	1
	save time	3	4,48	1	1	1	1	1	3	1	1	1	1
	save personal resources	3	4,48	1	1	1	1	1	1	1	1	1	1
standard documentation	reuse of the results	4	5,97	1	1	1	1	1	1	9	1	1	1
	comparability of the results	4	5,97	1	1	1	1	1	1	9	1	1	1
organizational structure	lean organization	5	7,46	1	1	3	1	1	1	1	1	1	1
Assessment		**67**	**100**	154	206	216	206	154	96	194	86	70	116
Rank				5	2	1	2	5	10	4	13	14	7

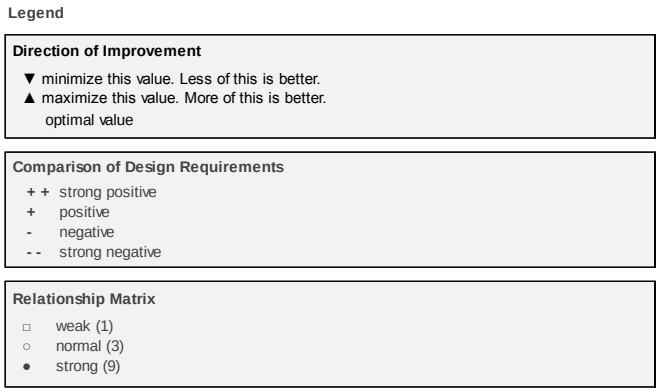


Figure 21: House of Quality – Agile Requirements Engineering

The house of quality shows that the content-related design requirements are the most important ones. For this reason, the primary focus for the development of the agile requirements engineering is on these content-related points. It is important to consider the results of the detailed analysis in order to eventually develop a requirements engineering process that fulfills the requirements of HERMES 5 and takes advantage of SCRUM.

6.3.3.1 Agile Requirements Engineering Process

As shown in Figure 17 (see page 70), the requirements engineering activities take place in the task "design a system concept". The detailed analysis of the requirements engineering component (see chapter 5.1.2) has shown that HERMES 5 doesn't define how to execute the requirements engineering process; it only includes the activity "work out system requirements" with the outcome "system requirements", and the activity "work out detailed study" with the outcome "detailed study". The goal of this master's thesis is to develop a requirements engineering process, which meets the requirements of the project management method HERMES 5 and takes advantage of the agile software development method SCRUM. Figure 22 below shows the agile requirements engineering process (see also appendix in chapter 14.6.1) with all its activities, inputs/outputs and responsible roles.

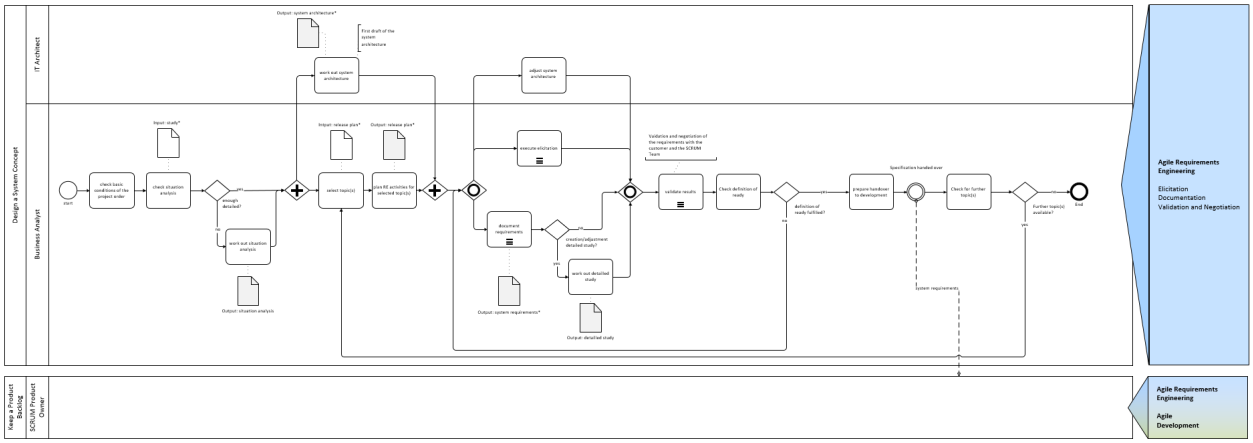
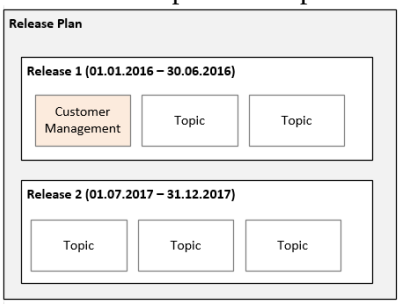
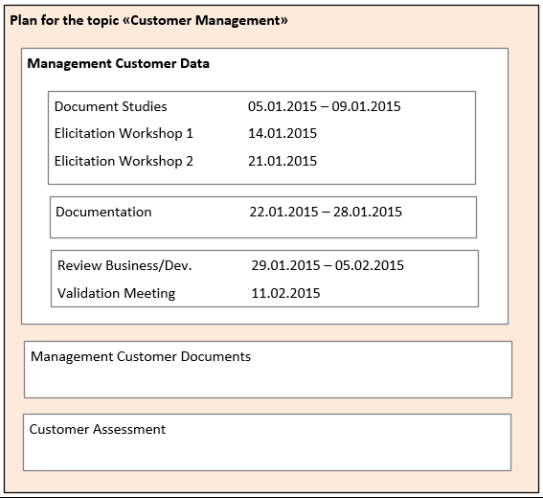


Figure 22: Requirements Engineering Process → Design a System Concept

The single activities are described in the tables below.

Activity	**Check basic conditions of the project order**
Description	As described in the HERMES 5 reference handbook (Eicher, Kruschitz & Mourgue d'Algue 2014) *"critically question project charter parameters and analyze how they might affect the success of project".* This activity is carried out at the beginning of the "concept" phase. The project charter parameters are, for example, the delivery times according to the project plan, and the available resources.
Input	Project Order
Output	-
Responsible Role	Business Analyst
Contributory Role	Project Manager
Example	-
Activity	**Check situation analysis**
Description	As described in the HERMES 5 reference handbook (Eicher, Kruschitz & Mourgue d'Algue 2014) *"check if the situation analysis contained in the study is detailed enough"*
Input	Study
Output	-
Responsible Role	Business Analyst
Contributory Role	Project Manager
Example	-
Activity	**Work out situation analysis**
Description	If the situation analysis in the study is not detailed enough, than the Business Analyst has to complete the situation analysis (Eicher, Kruschitz & Mourgue d'Algue 2014)
Input	Study
Output	Situation Analysis
Responsible Role	Business Analyst
Contributory Role	User Representative, Business Process Owner
Example	-
Activity	**Work out system architecture**
Description	Work out a first draft of the system architecture based on the information of the study and the situation analysis. The system architecture is subsequently continuously adjusted based on the new information obtained.
Input	Study, Situation Analysis
Output	System Architecture
Responsible Role	IT Architect
Contributory Role	SCRUM Developer, Operations Manager, Business Analyst
Example	-
Activity	**Select topic(s)**
Description	Select one or more topic that is due next according to the release plan.
Input	Release Plan
Output	-
Responsible Role	Business Analyst
Contributory Role	SCRUM Product Owner

	For the selection of a topic it is important to already involve the SCRUM Product Owner to find out which topic brings the most business value and with what degree of priority the topics should be handled.
Example	Example of a simple release plan. The topic "Customer Management" is selected for the further process steps. 
Activity	**Plan RE activities for selected topic(s)**
Description	One topic includes a larger investigation area. Therefore, it is necessary to plan requirements engineering activities for the selected topic(s) in detail. For the detailed planning of one or more topic it is useful to identify any subordinated topic areas if they exist.
Input	-
Output	Release Plan (adjusted release plan)
Responsible Role	Business Analyst
Contributory Role	SCRUM Product Owner, Project Manager
Example	Detail plan for the requirements engineering activities of the selected topic "Customer Management". The topic "Customer Management" contains the subordinated topic areas "Management Customer Data", "Management Customer Documents" and "Customer Assessment". In the example below only the topic area "Management Customer Data" is planned in detail. 
Activity	**Execute elicitation**
Description	Execute the elicitation according to the detailed plan for the topic. As a first step, I recommend document-centric and observation techniques (see chapter 2.3) to gain an insight into the current process handling. Based on this information it may already be possible to document some requirements, which could later be discussed with the customer. In a second step, use survey, support or creativity techniques for the elicitation of the system requirements (see chapter 2.3).

	The elicitation could be executed more than one time. It is possible to make one elicitation round with all user representatives or with every user representative one round.
Input	-
Output	-
Responsible Role	Business Analyst The Business Analyst has the leading role for the execution of the elicitation. But it is important if possible to involve the Development, the SCRUM Product Owner and the superiors of the user representatives in the elicitation from the very beginning.
Contributory Role	User Representative, Business Process Owner, SCRUM Product Owner, SCRUM Developer
Example	-
Activity	**Document requirements**
Description	Documentation of the requirements. It must be clear how the requirements have to be documented and stored. This is part of the task "introduce SCRUM" in the module "Project Management" The online survey showed that it is important that the requirements are documented and saved at a central place. This could be requirements engineering tools such as the Enterprise Architect, or other tools that allow a central storage of requirements (e.g. Microsoft Sharepoint when using Word and Excel). The chapter 6.3.3.3 gives an overview of the various documentation variants. The online survey showed that the development side mostly expects the following delivery objects from requirements engineering in the case of agile development. • Epics • User Stories • Screen Mocks • Use Cases The above delivery objects could be expanded with other UML models if the development side wishes (e.g. state transition diagram, activity diagram). The chapter 6.3.3.3 deepens the topic. **Hint: It is necessary to define in every project together with the development the delivery objects.**
Input	-
Output	System Requirements
Responsible Role	Business Analyst
Contributory Role	User Representative, SCRUM Product Owner, SCRUM Developer
Example	
Activity	**Work out detailed study**
Description	Work out detailed study about a specific topic or subordinated topic area. It is important that the customer or the development ask for the detailed study. Only create it when it's necessary.
Input	-
Output	Detailed Study
Responsible Role	Business Analyst
Contributory Role	User Representative, SCRUM Developer
Example	-
Activity	**Adjust system architecture**
Description	Adjustment of the system architecture based on new gained knowledge.

Input	System Architecture, Detailed Study
Output	System Architecture
Responsible Role	IT Architect
Contributory Role	SCRUM Developer, Operations Manager, Business Analyst
Example	-
Activity	**Validate results**
Description	Validation of the results. Discuss the documented requirements together with the customer and the development. The validation corresponds to an inspection. It is recommended to make informal inspections (e.g. walkthroughs), which do not have mandatory documentation in writing. 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 check the "definition of ready", which defines when the system requirements are ready for the handover to the development.
Input	System Requirements
Output	-
Responsible Role	Business Analyst
Contributory Role	User Representative, Superiors of the User Representatives, SCRUM Product Owner, SCRUM Developer
Example	-
Activity	**Approve topic(s) for use**
Description	If the validation was successful and the system architecture was adjusted, then the user representative and the superiors of the user representatives have to approve the topic for use. "For use" means that development can use the requirements in the module "Agile Development", but that it is still possible that the requirements can change based on discussions or decisions of the SCRUM Product Owner.
Input	System Requirements
Output	-
Responsible Role	Business Analyst
Contributory Role	User Representative, Superiors of the User Representatives, SCRUM Product Owner
Example	-
Activity	**Prepare handover to development**
Description	Prepare the handover to the development. This means that the Business Analyst coordinates with the SCRUM Product Owner when and how to transfer the system requirements into the product backlog.
Input	-
Output	-
Responsible Role	Business Analyst
Contributory Role	SCRUM Product Owner
Example	-
Activity	**Check for further topic(s)**
Description	The Business Analyst checks if further topics are available for which he has to raise, document and validate a first set of requirements.
Input	Release Plan
Output	-

Responsible Role	Business Analyst
Contributory Role	-
Example	-

Table 23: Description of the activities

Besides the execution of the activities described above it is recommended to conduct a **weekly meeting** during the requirements engineering activities (elicitation, documentation and validation). 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. It can be assumed that one requirements engineering iteration takes more than 4 weeks. Experience has shown that, especially in complex projects with a lot of variables, the clarification of open points requires time.

6.3.3.2 Roles

The agile requirements engineering process in chapter 6.3.3.1 already shows that only two roles are responsible for the execution of the tasks: the Business Analyst and IT Architect. Beside these two leading roles, other roles participate in the activities. 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 makes a statement about which roles in "Agile Requirements Engineering" are still involved later on in "Agile Development".

	Agile Requirements Engineering								
Agile Development	HERMES 5 / SCRUM	Business Analyst	IT Architect	User Repre-sentative	Business Process Owner	SCRUM Developer	SCRUM Product Owner	Operations Manager	Project Manager
	SCRUM Master								
	SCRUM Product Owner						x		
	SCRUM Team	x	x	(x)	(x)	x		x	

Table 24: Role matrix

Explanation of the matrix:

- The Business Analyst and IT Architect (the leading role in "Agile Requirements Engineering") are subsequently in "Agile Development" part of the SCRUM Team if the SCRUM Developers need their help.
- The User Representative and Business Process Owner have an important role in "Agile Requirements Engineering". They are the source for the elicitation of the requirements. In the later "Agile Development", the interests of the User Representative and Business Process Owner will be represented by the SCRUM Product Owner. If necessary, the SCRUM Product Owner can consult with the User Representative and Business Process Owner.
- The SCRUM Developer is the heart of the SCRUM Team and already involved in "Agile Requirements Engineering for the validation".

- The SCRUM Product Owner is involved in "Agile Requirements Engineering" and "Agile Development". In both modules he is responsible for the management of the requirements, the prioritization of the requirements and representing the User Representatives interests across the development. The SCRUM Product Owner has a fulltime job, and that he takes over a double role should be avoided (e.g. User Representative and SCRUM Product Owner).
- The Operations Manager could be part of the SCRUM Team later in "Agile Development" if the SCRUM Developers need to consult with him. This is less often the case.
- The Project Manager is not involved operatively in the module "Agile Development", but he has still the responsibility over the whole project. The SCRUM Master has to report to the Project Manager.

HERMES 5 makes the assumption that one person can have a HERMES 5 role in addition to a SCRUM role. For the SCRUM Team this assumption is absolutely correct, but the SCRUM Product Owner and SCRUM Master have to concentrate on one single thing. In the SCRUM Team the roles other than the Developers have more of an advisory function, which is why it is no problem.

HERMES 5 proposition for role accumulation	**Reasons**
SCRUM Product Owner ➔ HERMES 5: Business Analyst, Project Manager (customer side), Business Process Owner, Product Owner or IT Architect	**No** The SCRUM Product Owner should represent the customer side. He must have the ability to make fast decisions, must be reachable and must know the whole bandwidth of the business. From my point of view, the SCRUM Product Owner is a User Representative that knows all the multifaceted aspects of the business. The Business Analyst and Product Owner are not deeply enough involved in the business activities (they have another perspective). The Business Analyst, Project Manager (customer side) and IT Architect do not know the business enough to overtake this role.
SCRUM Team ➔ HERMES 5: Developer, Business Analyst, Test Manager and Tester	**Yes** In "Agile Development" the Developer, Business Analyst, Test Manager and Tester and various other roles can be part of the SCRUM Team.
SCRUM Master ➔ HERMES 5: Developer or Business Analyst	**No** The Developer should be a fully incorporated part of the SCRUM Team. It is not useful if one Developer of the SCRUM Team additionally has the role of SCRUM Master. But it is desirable if the SCRUM Master knows the development side. I propose for the role SCRUM Master a person with leadership ability who knows development (e.g. leader of a development team).

Table 25: Check of the role accumulation proposition

6.3.3.3 Outputs

The outputs of the developed agile requirements engineering process and the original task "design a system concept" are almost the same. The "detailed study" and the "system architecture" are the same documents according the HERMES 5 templates.

The "release plan" is an output, which is created in the task "create release plan" of the module "Project Management". The Project Manager is responsible for making the release plan at an early stage of the project in collaboration with the responsible persons for requirements engineering, development and the Product Owner. It is important to include the Product Owner in the creation of the release plan for the identification of the business value of the single epics. The business value indicates which topics have to be handled first. In the task "create a system concept", the release plan is continuously refined for every selected topic.

The most important output of the task is the document "system requirements", which is described in detail in the following chapter.

System Requirements

For the "system requirements" it is recommended to make it so that it can be handed over without making changes to development. To this end, it is vital to discuss the topic as early as possible with development.

The following list gives an overview of the content according the currently HERMES 5 template.

- Supported Business Processes
- Roles and Access Privileges
- Business Objects and Attributes
- Data Related Requirements
- Functional Requirements
- Non Functional Requirements
- Security Requirements

The document template defines a chapter as a placeholder for each of the above categories. The chapters are not mandatory and every author is free to adapt the document's content. The following figure gives an overview of the artifacts in traditional requirements engineering (left side), and the artifacts that are used in agile development (right side). As the online survey showed, requirements engineering often has the same thinking in agile software development projects as it does in traditional projects, and follows this approach to capture the whole problem and to create a comprehensive documentation. The agile development with SCRUM works with a few artifacts.

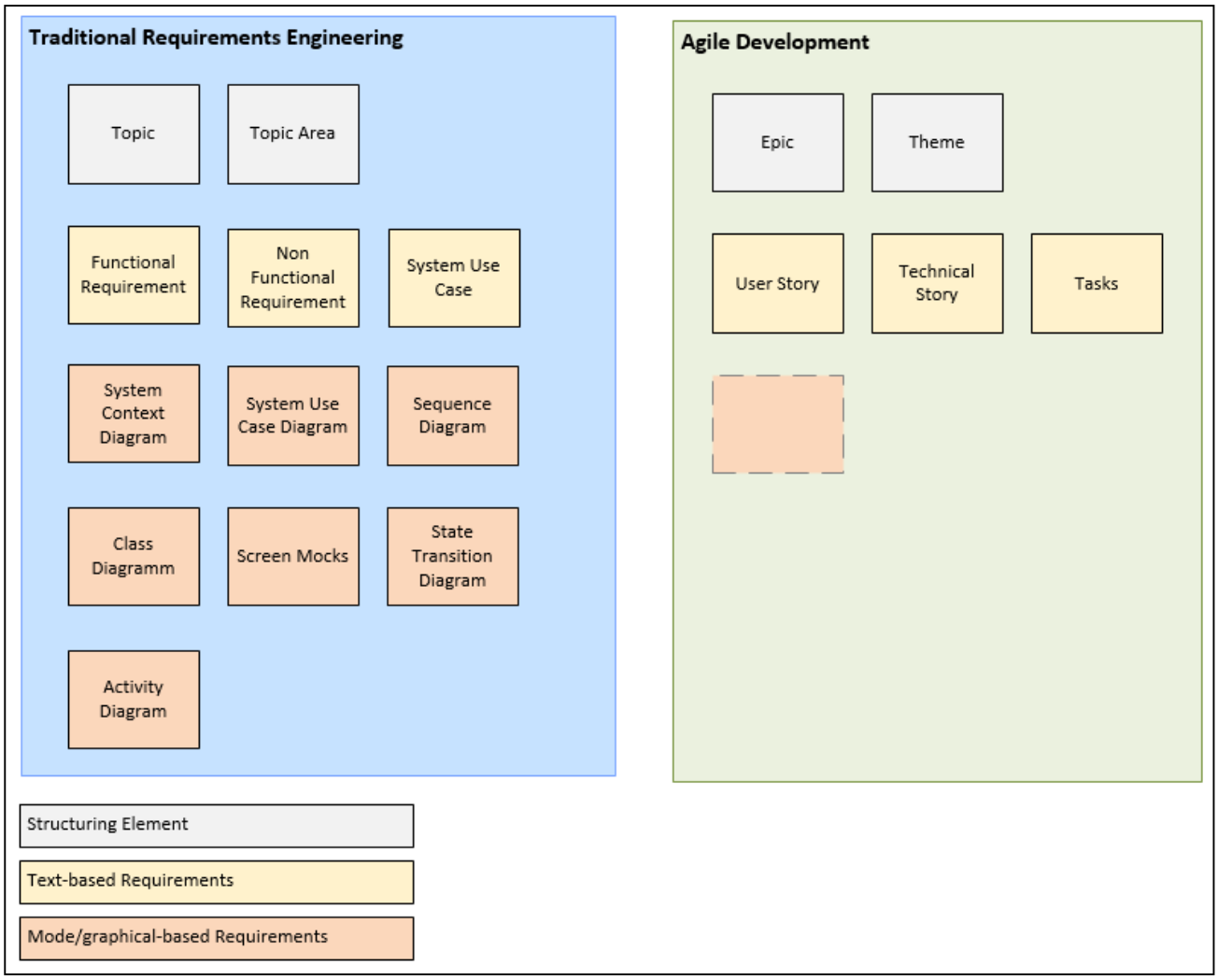


Figure 23: Artefacts. Traditional Requirements Engineering vs. Agile Development (based on (Rupp & SOPHISTen 2014) (Pohl & Rupp 2011) (Bergsmann Johannes 2014))

For requirements engineering it is desirable that the artifacts can be handed over to development without changes. The online survey has shown that, in practice, development uses epics, themes and user stories, which later on are broken down in sprint planning 2 to tasks. In the online survey more than one person mentioned that the System Use Cases are a good documentation technique in agile software development projects. Furthermore, System Use Cases can later be used in testing for the creation of the test cases.

The following figure makes a proposition concerning which artifacts to create during the agile requirements engineering processes. The figure shows that, if necessary, in development the SCRUM Team creates some technical user stories and the tasks for the realization of the stories. The other artifacts are created during the requirements engineering process and handed over to the development. At least all of the artifacts below build the base for the implementation.

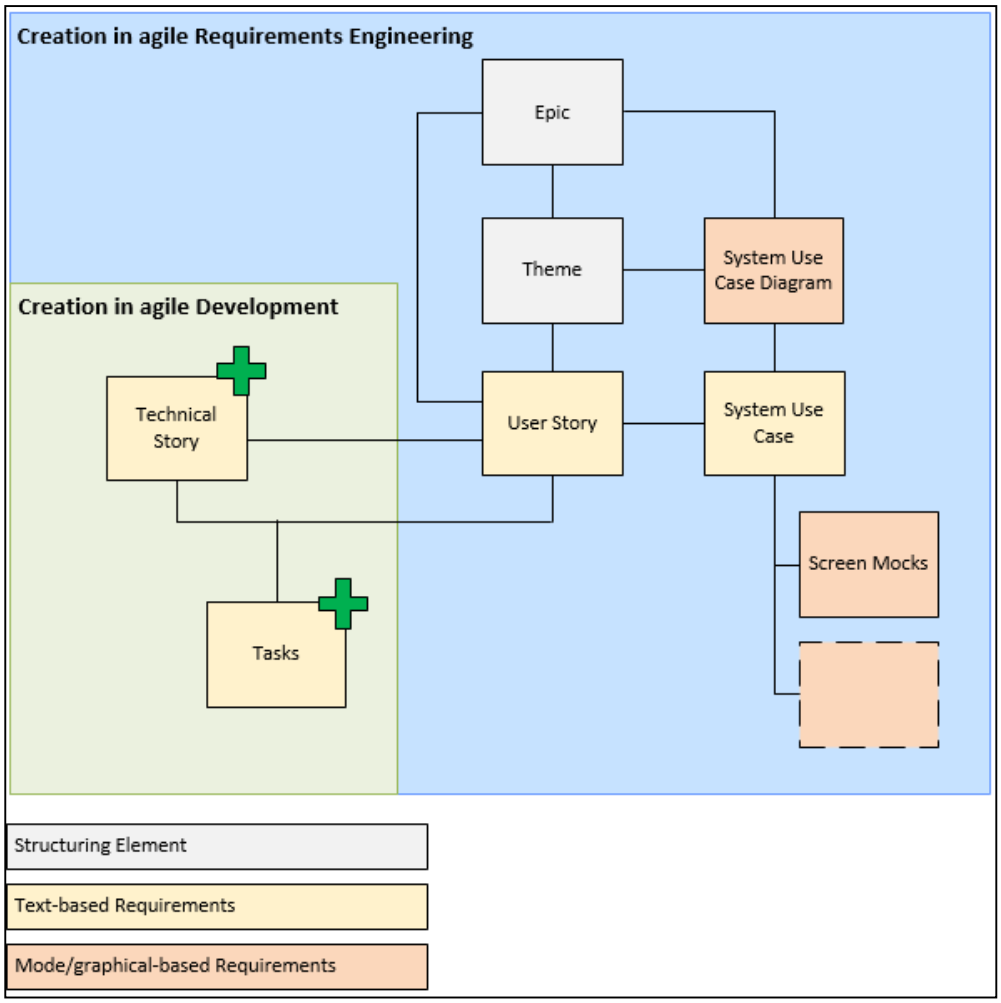


Figure 24: Artefacts. Agile Requirements Engineering vs. Agile Development

The following table gives a short explanation of Figure 24. The creation of the single artifacts takes place in the activity "document system requirements". It has not been attempted here to describe the single artifacts in detail, because there is enough literature on the topic.

Delivery Object	Description	Creation in
Epic	An epic is a high level requirement, which is often represented by a single keyword (e.g. user management) An epic is further divided into themes or user story diagrams.	Agile Requirements Engineering
Theme	A theme divides an epic into different subordinated functional topics, which contain a group of user stories.	Agile Requirements Engineering
User Story	A user story describes a functional requirement, which has a business value. User Stories have to be independent, negotiable, valuable, estimable, small and testable (INVEST quality criteria). During requirements engineering it is in addition important to define the acceptance criteria for the user stories. Acceptance criteria: Every user story should have acceptance criteria, which have to be fulfilled after implementation. It should be noted that non-functional requirements can be acceptance criteria.	Agile Requirements Engineering

Technical Story	A technical story (also known as developer story) is a story for a technical requirement. A technical story has no business value.	Agile Development
Task	Divides a user story or technical story into the concrete development tasks.	Agile Development
System Use Case Diagram	A system use case diagram summarizes several system use cases for a specific functional part (theme).	Agile Requirements Engineering
System Use Case	A system use case describes an interaction between the system and the user for a specific scenario. In the context of an agile requirements engineering system, use cases can be used for the detailed description of a user story.	Agile Requirements Engineering
Screen Mock	A screen mock visualizes what the system could look like. A screen mock can be a simple sketch or a clickable html.	Agile Requirements Engineering

Table 26: Description of the artefacts (based on (Bergsmann Johannes 2014) (Wirdemann 2011) (Rupp & SOPHISTen 2014))

Documentation Variants

The sections above have made a proposition of the content of the document system requirements. Another question is how to document the system requirements. There are several variants to document the system requirements. Which variant for the documentation of the requirements is appropriate depends on the project. The kind of documentation is a topic that has to be discussed in the task "introduce SCRUM" together with development.

Variant 1

In variant 1 the documentation of the text- and model-based requirements takes place directly in the product backlog. In order to do this the product backlog has to support the modeling activities. The Business Analyst documents the requirements directly in the Product Backlog during the requirements engineering process. Later on, the SCRUM Product Owner has the possibility to manage the requirements during the SCRUM process.
A big advantage of this variant is that the SCRUM Product Owner has to make almost no effort for the preparation of the Product Backlog for development. He just has to check and prioritize the requirements.

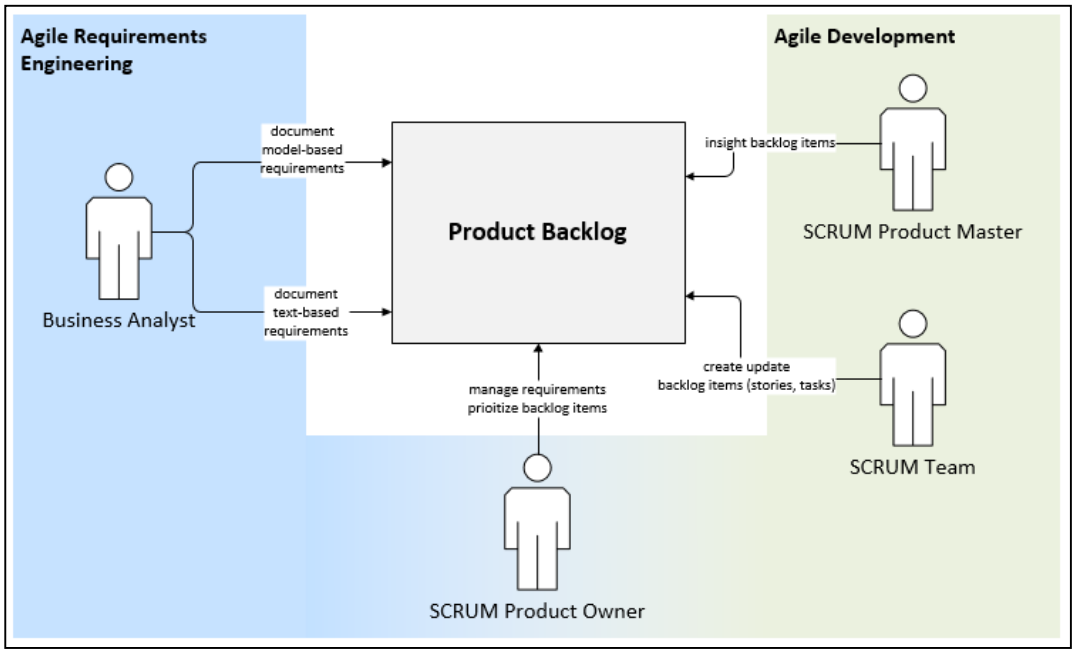


Figure 25: Full Requirement Documentation in Product Backlog

Variant 2

In variant 2 the documentation of the requirements takes place partially in the product backlog. The text-based requirements are in the Product Backlog. Additional model-based requirements are managed in a separate requirements engineering tool. In this case, the Product Owner can manage the text-based requirements in the product backlog on their own at a later stage. Changes on models have to be done through the Business Analyst.

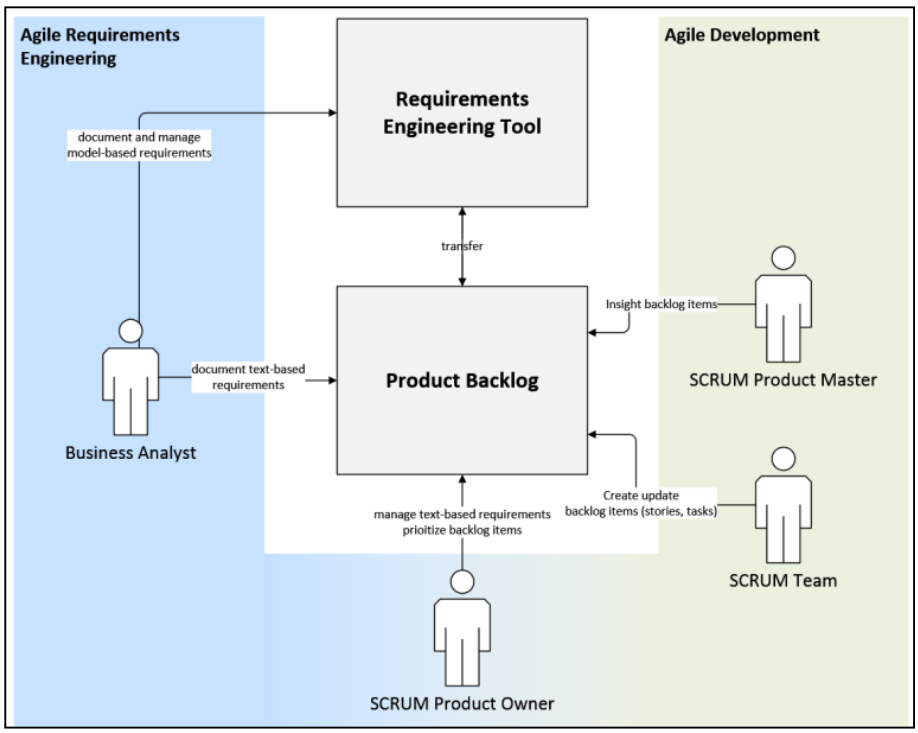


Figure 26: Partially Requirement Documentation in Product Backlog

Variant 3

In variant 3 the documentation of the text- and model-based requirements takes place in a specific requirements engineering tool (e.g. enterprise architect). After approval for use (when the "definition of ready" are fulfilled), the requirements are transferred into the Product Backlog.

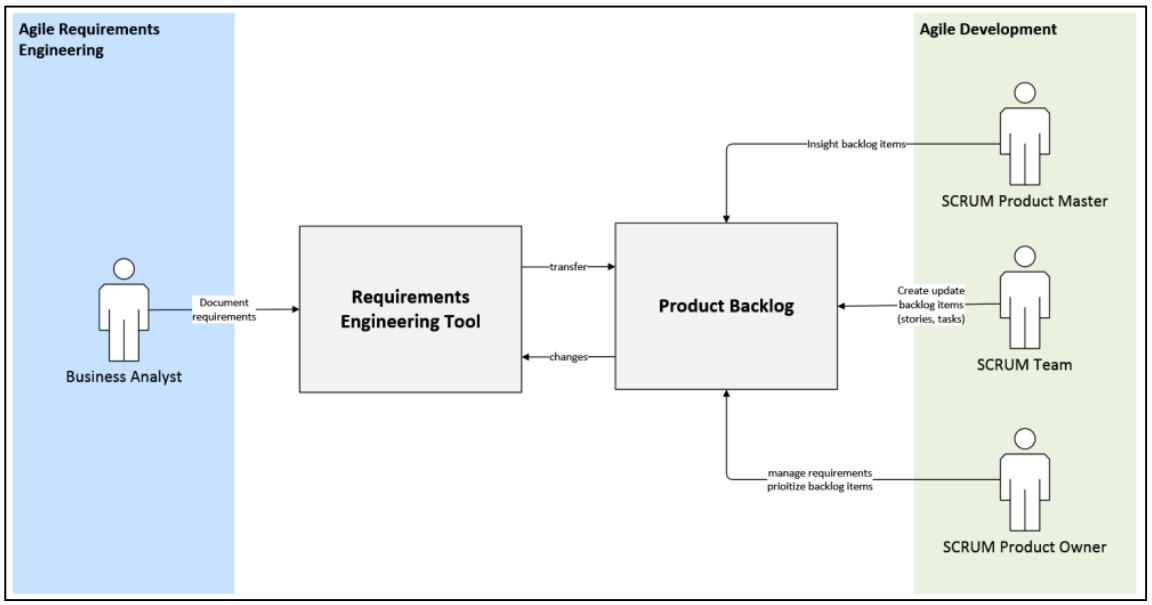


Figure 27: Requirement Documentation in separate Requirements Engineering Tool

Variant 4

The documentation of the requirements takes place in a document (e.g. word or excel).

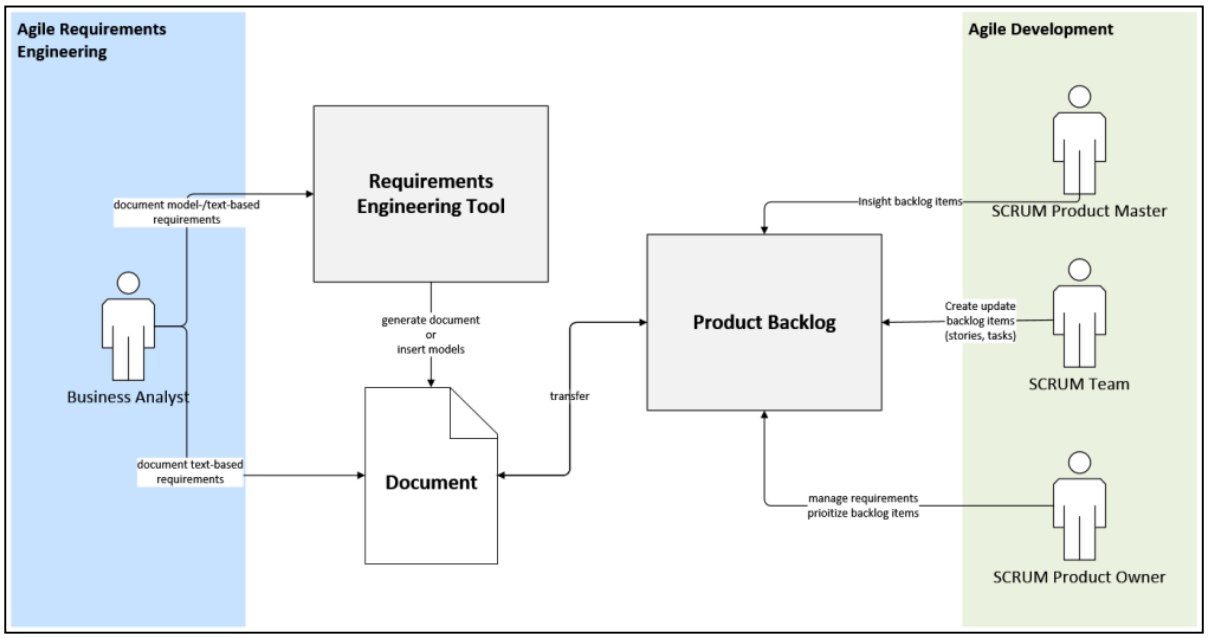


Figure 28: Requirement Documentation in Document

As already mentioned, the chosen variant depends on the project. Projects with a high complexity tend more towards variant 1, whereas projects with a low complexity can work well with variant 4.

The following table compares all variants depending on their advantages and disadvantages. Variant 1 has the most advantages. The variant forces the collaboration between requirements engineering and development. The SCRUM Product Owner is part of the requirements engineering process, because he is responsible for the management of the text-based requirements in the product backlog. The handover to development is based on the same simple database as in the other variants and there is no risk of loss or falsification of the requirements.

	Variant 1	Variant 2	Variant 3	Variant 4
Supporting the Collaboration between Requirements Engineering and Development	close (2)	close (2)	distant (0)	distant (0)
Effort for handover to the Development	low (2)	low (2)	normal (1)	high (0)
Risk for information lost/falsification during transfer into Product Backlog	low (2)	low (2)	normal (1)	normal (1)
Effort to carry out changes to text-based requirements	low (2)	low (2)	normal (1)	high (0)
Effort to carry out changes to model-based requirements	low (2)	normal (1)	normal (1)	high (0)
Central Management of the requirements	central (2)	not central (0)	central (2)	not central (0)
Total	**12**	**9**	**6**	**1**

Table 27: Evaluation of the Variants

6.3.3.4 Interaction with Agile Development

This chapter shows how the task “design a system concept” interacts with the tasks of the module “Agile Development”. The task “design a system concept” handles the main requirements engineering activities “elicitation”, “documentation” and “validation and negotiation”. Upon completion of the requirements engineering process, the Business Analyst hands over the “system requirements” to development.

Assuming that the documentation of the requirements takes place in the product backlog, the handover occurs in a meeting in which the Business Analyst discusses the requirements with the SCRUM Product Owner. Later on, the SCRUM Product Owner is responsible for the 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”.

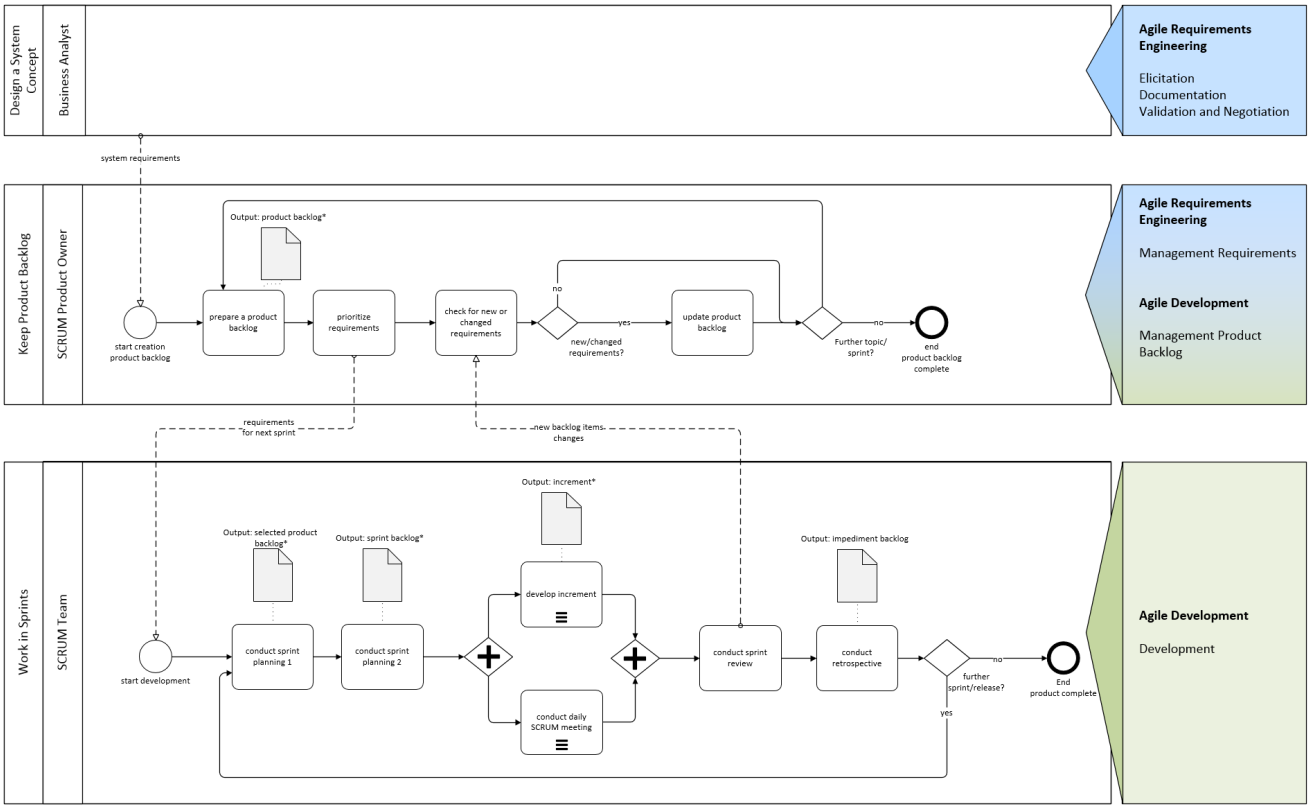


Figure 29: Interaction with Agile Development

7. Conclusion

The previous chapters show which parts of the agile HERMES scenario are affected by an agile requirements engineering and makes a proposition for changes to arrive at an agile HERMES scenario whose requirements engineering component meets the requirements of HERMES 5 and makes use of the advantages of SCRUM. This chapter aims to summarize and evaluate the result of the design phase.

The new adjusted HERMES 5 scenario has a new module called “Agile Requirements Engineering”, which works hand in hand with the module “Agile Development”. The tasks of the module “Agile Requirements Engineering” are still the same as in the previously used module “IT System”. The biggest changes have occurred in the task “design a system concept”. These tasks contain the requirements engineering activities “elicitation”, “document” and “validate and negotiate”. The procedure is comparable with the official SCRUM process.

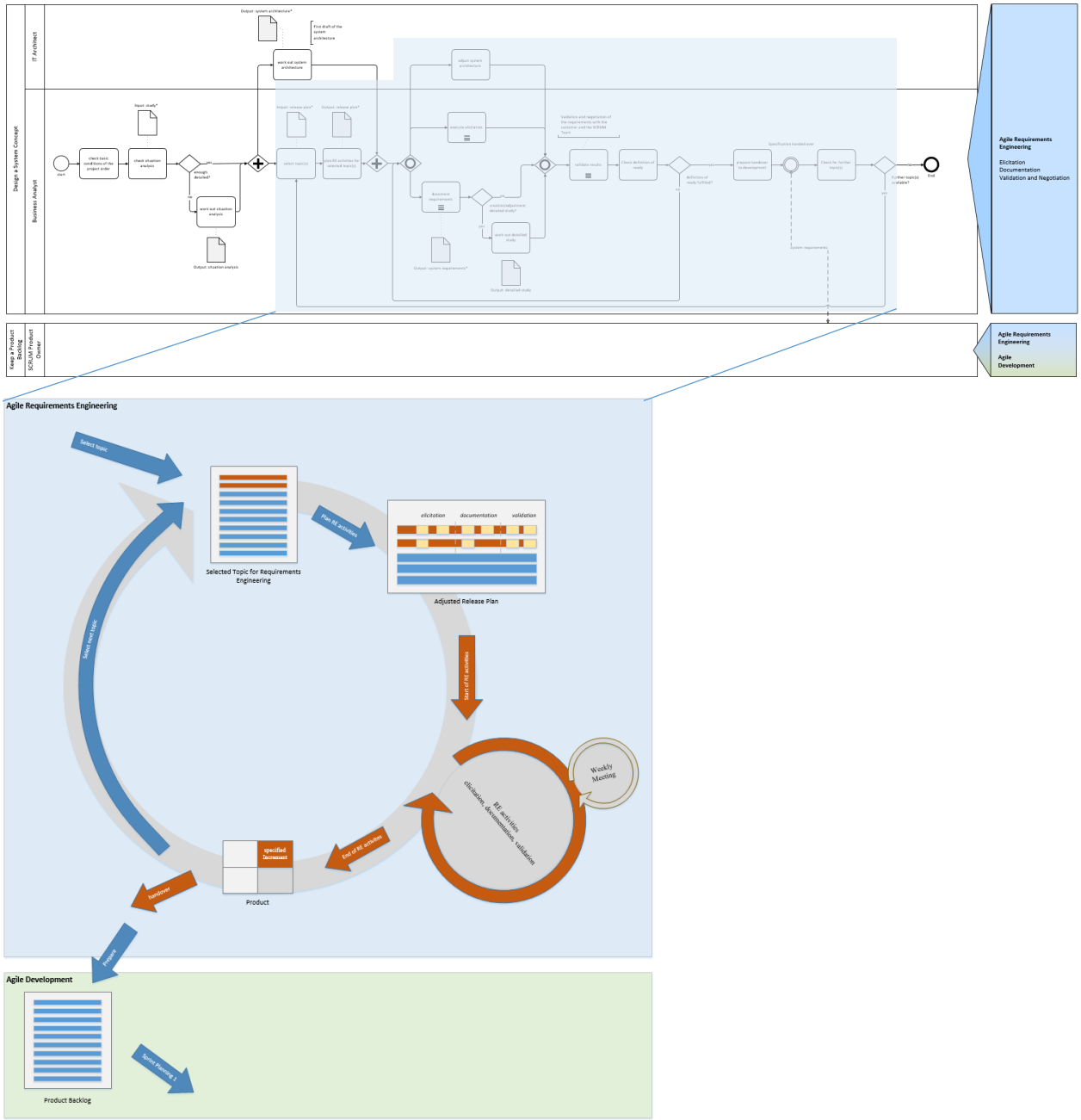


Figure 30: Agile Requirements Engineering

Based on a Release Plan, the Business Analyst knows which business topic he has for doing first the requirements engineering. The Business Analyst selects a topic for the first requirements engineering iteration. The Business Analyst plans the single RE activities and then starts the execution. This could happen in multiple loops, depending on the topic. A big and complex topic needs more loops than simple topics do. After the requirements engineering activities are finished, the system specification for the selected topic is handed over to development or more concrete to the Product Owner, which prepares the Product Backlog for the development. The documentation of the requirements takes place during requirement engineering directly in the Product Backlog; therefore, the Product Owner just has to check and prioritize the requirements for development. After that, the official SCRUM process starts and the business Analyst begins the next requirement engineering iteration for the next business topic. The new adjusted HERMES 5 scenario handles the initial requirements engineering. The requirements engineering activity "management" takes place in the module "Agile Development". The Product Owner is responsible for clarifying upcoming questions as quickly as possible during the development sprints.

The developed module "Agile Requirements Engineering" meets the requirements of the project management method HERMES 5. As has been mentioned several times, the requirements of HERMES 5 are on a highly theoretical level. On the other hand, it is possible to take advantage of the agile software development method SCRUM by adapting the procedure to an iterative and incremental (agile) approach. This doesn't mean, however, that an agile requirements engineering benefits from the same advantages as agile software development. The following diagram shows the summarized advantages of the agile software development method SCRUM, and how the advantages can be utilized depending on the method used. For requirements engineering, both approaches, traditional vs. agile requirements engineering, are included in the diagram. The single points are explained in the sections on the following pages.

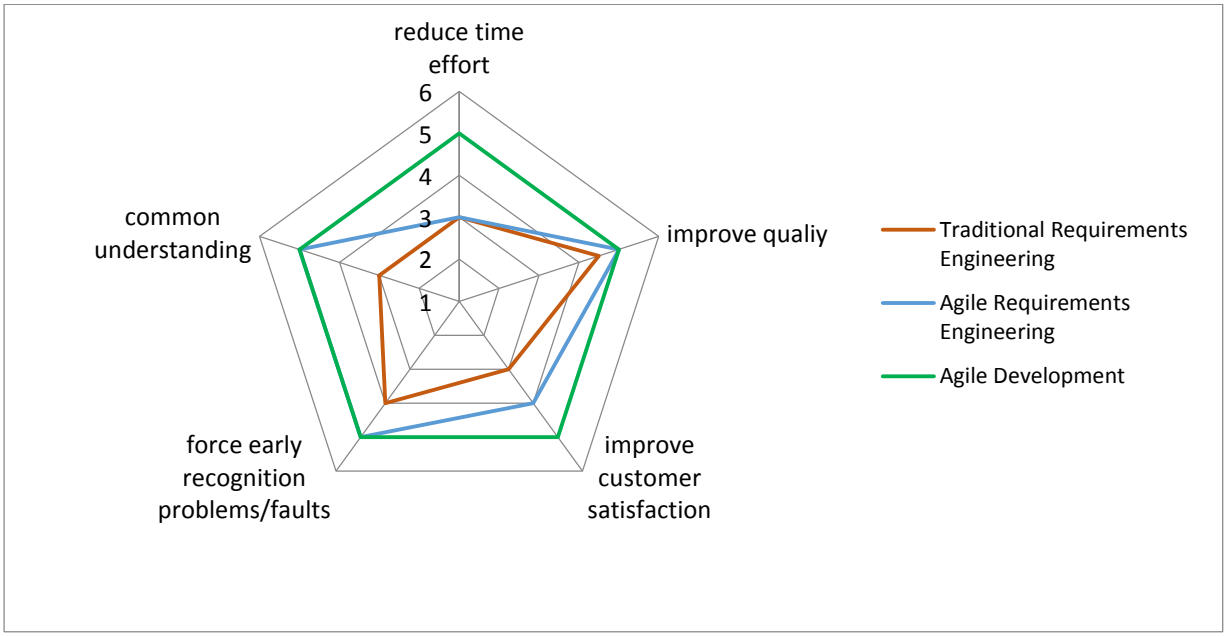


Figure 31: Net Diagram Advantages

Reduce Time Effort

In agile software development the time needed for the implementation is reduced through the use of SCRUM. In the context of agile requirements engineering, this advantage could not be used under every condition. Especially in the case of big and complex IT projects for the development of individual business software, the requirements engineering part cannot be accelerated. The complexity of an IT project is characterized by the project organization, the amount of foreign systems and their level of development, legal and political factors, the complexity of the investigated business area and dependencies from other running projects—to name just a few examples. The more complex a project is, the more ambiguities that can occur: which at the very least entails that requirements engineering cannot smoothly reach the "definition of ready". The moving of ambiguities into the SCRUM process is not purposeful, because the SCRUM Product Owner is not able to clarify them in appropriate time. This leads to the conclusion that IT projects with a high complexity have to conduct a more extensive initial requirements engineering than IT projects with a low complexity.

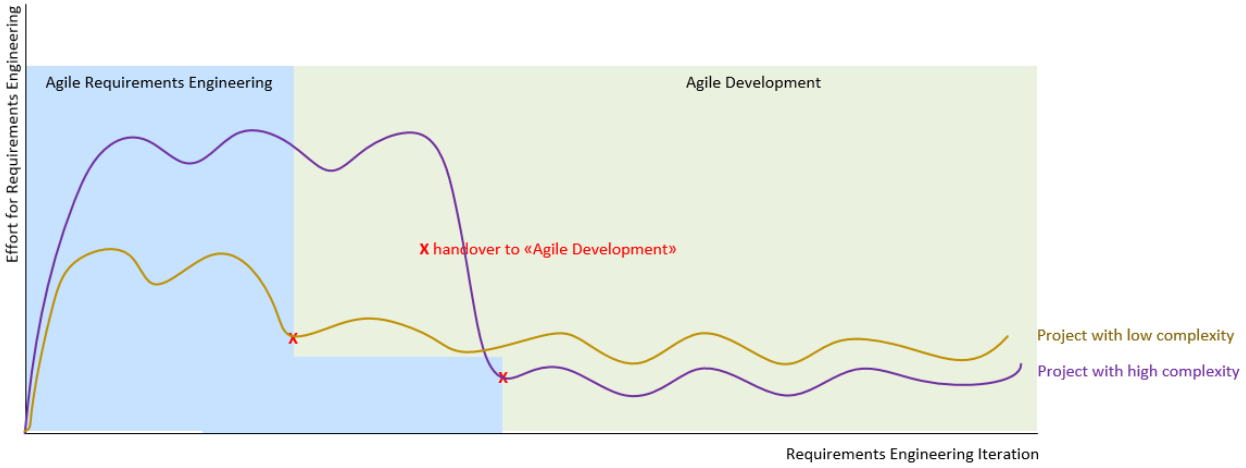


Figure 32: Requirements Engineering. Complex vs. not complex IT Projects

Overall the time for the agile requirements engineering itself is the same as in traditional requirements engineering. The main difference is that the effort is distributed differently. The assumption that the effort for the creation of the system requirements can be reduced with the agile approach is not correct.

But the consequences of the agile approach, the iterative and incremental execution of the requirements engineering activities, are that the customer sees a result out of the requirements engineering activities early on, even if that is only a part of the whole.

Improve Quality

The survey has shown that the detail of the system requirements is often insufficient, and that the system requirements try to force a solution. In the developed requirements engineering process the activity "validate results" ensures that the system requirements are validated by the customer, and, on other side, also by development. Only if the "definitions of ready" are fulfilled, which are defined at the beginning of the phase together with development, can the system requirements be handed over to development for starting the SCRUM process. The agile requirements engineering process can with its defined activities force the improvement of the quality. But it should be mentioned that traditional requirements engineering can reach the same level of quality as agile requirements engineering, if it also considers the Developers at an early stage of the project. The improvement

of quality isn't dependent on the applied method. It depends on the basis conditions that were set together with development for requirements engineering and from the execution of the review activities.

Improve Customer Satisfaction

The satisfaction of the customer is very high in agile software development projects because the customer is directly involved in the SCRUM process through the Product Owner, as well as able to see a concrete result from the development process after every iteration. In the agile requirements engineering process the situation is the same, but the result does not have such a high recognition effect. The results of the requirements engineering process are the system requirements. For the customer, it is a source of satisfaction to see that the system requirements contain his needs, but it is not a concrete result that he could use and test. It is not possible to say that the customer is in traditional requirements engineering involved more or less than in agile requirements engineering. But in agile requirements engineering the customer is through the SCRUM Product Owner, which should be a person from customer side, closer to the development and involved directly in the requirements engineering activities. He is able to actively participate in the creation of the system specification.

Force early Recognition of Problems and Faults

Through the agile approach it is possible to recognize early problems and faults. The online survey has shown that in agile software development errors are often recognized that happened in traditional requirements engineering. With the application of the agile approach in requirements engineering (specification units and releases) and the early involvement of development, problems and faults can be identified with a higher probability during the course of requirements engineering. The assumption that the agile approach forces an early recognition of problems and faults is therefore true.

Common understanding

In agile software development projects that use SCRUM, a common understanding is achieved through the close collaboration between development and the Product Owner, and through the frequent coordination meetings (daily meeting). The new respectively adjusted agile HERMES scenario prescribes a close collaboration between development and requirements engineering, which forces as a consequence a common understanding. In addition, the initial set up of the project (task "introduce SCRUM") already defines on a higher level a common understanding about the delivery objects and the manner in which the collaboration should take place during the IT project. In traditional requirements engineering a common understanding is likewise desirable, but in practice, unfortunately, requirements engineering is further away from development than from agile requirements engineering (that presupposes the close collaboration). The assumption that the agile approach encourages a common understanding is correct.

The sections above have shown that the developed requirements engineering process takes advantage of SCRUM through the adoption of the same approach; and, vice versa, that the development with SCRUM takes advantage of the adjusted agile HERMES scenario.

In terms of average agility it is possible to make the statement, that complex IT projects are less agile than IT projects with a low complexity (see Figure 33).

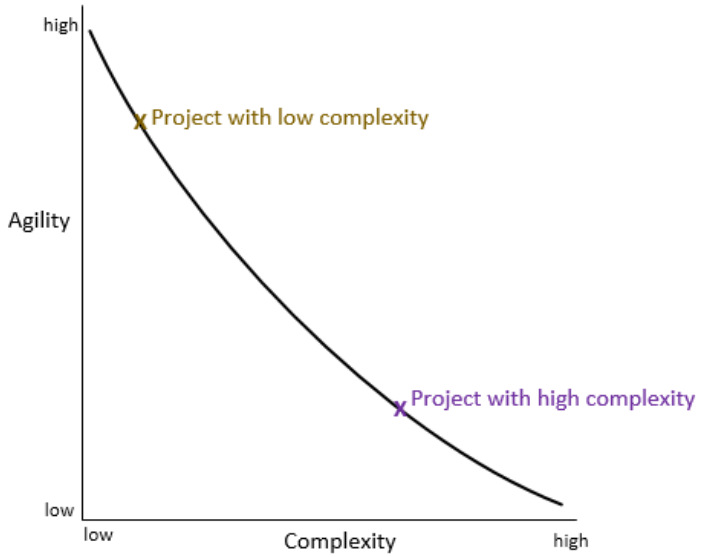


Figure 33: Complexity vs. Agility

The sections above show that the requirements engineering process developed within the module "Agile Requirements Engineering" meets the requirements of HERMES 5 and takes advantage of SCRUM. In the single analysis part in this master's thesis I identified several points, which could respectively be improved with the developed requirements engineering process. The structural analysis of the originally agile HERMES scenario pointed out general weaknesses (e.g. missing responsibilities, unclear roles). During the development of the module "Agile Requirements Engineering", I tried to consider the critical points that are identified in a first structural analysis of the originally agile HERMES scenario. The following table reflects these points again for the adjusted agile HERMES scenario.

Identified weakness/criticality	**Realization with the new adjusted agile HERMES scenario**
Project Charter doesn't take into account the decision for agile development with SCRUM	The task "take decision for agile development using SCRUM" is taken up anew in the phase "initiation". Based on this circumstance it is possible to consider the decision already for the creation of the "Project Charter".
Change Management is not agile	The topic is based on the scope, not explicitly and comprehensively handled in the new adjusted agile HERMES scenario. During the agile requirements engineering process changes on requirements can be carried out without a formal change management process. After the requirements are handed over to development changes that are identified during the sprint review lead to new backlog items. Each backlog item has a status. In the sense of an agile Change Management, the status should be "change" which allows the Project Manager to use the Product Backlog as "Change Status List".
Level of documentation for creating tender documentation	Due to the focus of this master's thesis on agile requirements engineering, it was not possible to handle this topic. The topic is a candidate for a further study.

Prepare release of the phase implementation		This point is still sensitive after the adjustment of the agile HERMES scenario. But through training the project team members in the task "introduce SCRUM", all members should understand from the beginning of the concept phase the idea of agility.
Release plan		The outcome "Release Plan" is a mandatory document that has to be created in the module "Project Management". The "Release Plan" is updated continuously during agile requirements engineering and agile development.

Table 28: Reflection findings structural analysis in chapter 2.6.2.1

The detailed analysis part of this master thesis (see chapter 5) has pointed out…

- The general weaknesses of the agile HERMES scenario (analysis part 1)
- Important points that have to be considered for the design of the agile requirements engineering process (analysis part 2)
- The differences between the module "Agile Development" and the official SCRUM process (analysis part 3)

The following table illustrates how the identified weaknesses/criticalities were solved in the new adjusted HERMES 5 scenario.

Reference Detailed Analysis	**Identified weaknesses/criticalities/questions**	**Realization with the new adjusted agile HERMES scenario**
Analysis Part 1	The general weaknesses of the agile HERMES scenario	 The weaknesses for the task "design a system concept" and the tasks of the module "Agile Development" are adjusted
Analysis Part 2	Adjustment of the requirements engineering for a later agile software development with SCRUM	 Proposition of an agile requirements engineering process, which works out the artifacts in an iterative manner.
Analysis Part 2	Artefacts out of the agile requirements engineering process	 Proposition of a set of artifacts, which have to be created during the requirements engineering process.
Analysis Part 2	Necessity of the document "system requirements"	 The creation of the document is not necessary. In an optimal case, the requirements are documented directly through requirements engineering into the Product Backlog.
Analysis Part 2	Inspection	 The topic is due to the scope of this study not explicitly and comprehensively handled in the new adjusted agile HERMES scenario. The activity "validation" corresponds to the inspection point. In agile procedure it is recommended to use simple inspection techniques (e.g. walkthroughs)
Analysis Part 2	Change Management	 See comment in Table 28

Analysis Part 3	differences between the module “Agile Development” and the official SCRUM process		The differences are adjusted. The module “Agile Development” corresponds to the official SCRUM process.

Table 29: Reflection findings detailed analysis

The reflection shows that the majority of the identified points have been supported within the agile requirements engineering process. The result of this master’s thesis shows that it is possible to develop a requirements engineering process that meets the requirements of HERMES 5 and takes advantage of SCRUM. Depending on the complexity of the IT project and the basic conditions in an organization, however, it is not always possible to force agility. For the scope of this study (big and complex IT projects for the development of individual business software) the developed requirements engineering process doesn’t force agility, but helps software development receive what they expect. Last but not least, I draw the conclusion that requirements engineering in complex IT projects cannot be agile in such a way like software development. The results of the master thesis shows a solution of how the two methods HERMES 5 and SCRUM could move closer together, but it doesn’t provides a solution how to reach agility.

Lastly, there are still points that are unclear for agile software development projects that have to be investigated as part of a separate study. In the context of requirements engineering it is not clear how to handle the topic Change Management, Quality Assurance or Procurement from a practical viewpoint within an agile IT project. Furthermore, it would be interesting to have also a deeper look inside other modules of the agile HERMES 5 scenario in context of agility. As an example the module “Testing”, which I assume could be handled also by an iterative and incremental approach. It would be interesting to see how testing would work and interact together with “Agile Requirements Engineering” and “Agile Development”. Based on the limited scope of this study it was unfortunately not possible to go deeper into the detail for these mentioned topics.

The next chapter summarizes the results of the proof of concept and provides a final statement regarding the benefit of the master thesis.

8. Proof of Concept

This chapter contains the proof of concept for the content in the previous chapter. The proof of concept is an assessment of the master thesis through internal and external experts of the Federal IT Steering Unit that are partially members of the eCH Standard Group, Section HERMES.

The goal of the proof of concept is to find out if the results of the master thesis could later on be used as input for the next HERMES 5 release and bring therefore a benefit for the whole HERMES community. The results of the assessments are concretely proposed amendments and general statements about the plausibility of the results.

It is planned that the master thesis is after the submission at the University of Applied Sciences Northwestern Switzerland handed-in official to the eCH Standard Group Section HERMES for the acceptance to add the master thesis as a best practice guide to the eCH Standard 0054.

8.1 Experts

The assessment of the previous chapters was executed through the following experts:

Prename and last name	**Employer**	**Function**	**Member eCH Standard Group, Section HERMES**
Guido Eicher	Federal IT Steering Unit	Procurement Coordinator, Project Manager	Yes, chairmanship Author HERMES 5
Jérôme Galeuchet	Swiss Post	Leader Enterprise Architecture and Quality Assurance	Yes, member
Bernhard Kruschitz	BKI AG	Owner and Chief Executive Officer of BKI AG	Yes, member Co-Author HERMES 5
Joscha Jenni	Mimacom AG	Head of Projects	No (external consulter of the FITSU)

Table 30: Overview Inspectors

It is worth mentioning that two of the experts are involved in the operative creation of the HERMES 5 reference handbook. Guido Eicher (Federal IT Steering Unit) is the author and Bernhard Kruschitz the co-author of the reference handbook.

8.2 Findings

The detailed evidences of the proof of concept can be find in appendix 7 in chapter 14.7. The individual evidences are commented there regarding their current state of consideration.

The majority of the evidence refers to a small error that occurred through a false interpretation of the HERMES 5 reference handbook's content. All this errors were directly corrected based on the experts proposed amend-

ment. The analysis part of the master thesis focuses on the phases “concept”, “implementation” and “deployment”. This scope was chosen as it is at the beginning of the analysis part, because the module “agile development” touches only these phases. From the point of view of Joscha Jenni (Mimacom AG) it would be interesting to analyze also the phase “initiation” from perspective of the requirements engineering process and to question if it is really correct to start at the phase “concept” with agile development. Furthermore, he added that maybe some more tasks of HERMES 5 have to be take into account for requirements engineering. The last point was introduced too by Bernhard Kurschitz (BKI AG). Both (Bernhard Kurschitz and Joscha Jenni) mentioned that the requirements engineering activities take place also in the phase “initiation”, “implementation” and “deployment”. I am absolutely agree with this evidence. The study refers only on one task in the module “IT System” (and two supporting tasks in the module “Project Management”), because the largest proportion of the requirements activities are handled from my point of view in this task.

Joscha Jenni (Mimacom AG) supports the idea to integrate requirements engineering in HERMES 5 (5.1) and sees this as an added value for HERMES. In the implementation and improvement suggestions he sees still potential (e.g. cooperation of requirements engineering and testing), but this is also from his point of view beyond the scope of the master thesis. Joscha Jenni verify that I can confirm the selected hypothesis.

According to Bernhard Kruschitz (BKI AG) the eCH Standard Group has already discussed to introduce a module “Requirements Engineering”. But they decided against it, because the requirements engineering activities take place in several modules. This would breached the principle of the thematic structure of the modules of HERMES 5. The question about the cross-module issues, would probably be regarded as more in-depth.

According Guido Eicher (FITSU) the master thesis can deliver some valuable inputs for the next release of HERMES 5. The master thesis will handed-in official to the eCH Standard Group Section HERMES, where the master thesis in general and the already made evidences are discussed again. The members of the eCH Standard Group take finally the decision if the master thesis can be added as best practice guide to the eCH Standard 0054 and if the project management method HERMES 5 have to be adjusted in some points.

9. Bibliography

Abbas, N., Gravell, A.M. & Wills, G.B., 2008. Historical Roots of Agile Methods : Where did " Agile Thinking " Come from ?

Agile Alliance, 2014. Agile Alliance. Available at: http://www.agilealliance.org/ [Accessed May 21, 2014].

Alberts, D.S. & Takai, T.M., 2011. *The Agility Advantage: A Survival Guide for Complex Enterprises and Endeavors*,

Balnaves, M. & Caputi, P., 2001. Introduction to quantitative research. In *Introduction to Quantitative Research*. SAGE Puclications, pp. 1–10.

Basili, Victor, R. (University of M. & Larman, C. (Valtech), 2003. Iterative and Incremental Development: A brief history. , (June), p.10.

Beck, K. et al., 2001. Manifesto for Agile Software Development. Available at: http://agilemanifesto.org/.

Beedle, M. et al., 1999. *SCRUM : An extension pattern language for hyperproductive software development*,

Bergsmann Johannes, 2014. *Requirements Engineering für die agile Softwareentwicklung* 1st ed., dpunkt.verlag GmbH.

BIT, 2014. Das BIT. Available at: http://www.bit.admin.ch/org/index.html?lang=de [Accessed November 16, 2014].

Bolley, E., 2014. Discussion on portfolio management FOITT. (Personal communication, 11 november 2014).

Brown, R., 2014. Fueling Innovation Deductive vs Inductive Thinking - Rebel Brown.

Casali, D., 2012. A brief history of agile methods. Available at: http://intenseminimalism.com/2012/a-brief-history-of-agile-methods/.

Case Maker Inc. & Aspects, E., 2000. What is Rapid Application Development?

Dräther, R., Holger, K. & Carsten, S., 2013. *SCRUM - kurz & gut* 1th Editio., Köln: O'Reilly Verlag.

Eicher, G., Kruschitz, B. & Morgue d'Algue, H., 2014. Agiles Projektmanagement mit HERMES und SCRUM. , 44(6).

Eicher, G., Kruschitz, B. & Mourgue d'Algue, H., 2014. *HERMES 5 - Referenzhandbuch*, Federal IT Steering Unit.

Federal Department of Finance (FDF), 2013. *Staatsrechnung Band 3*,

Federal IT Steering Unit, 2013a. *Die Geschichte der Projektführung mit HERMES*, Available at: http://www.isb.admin.ch/themen/methoden/01661/01666/index.html?lang=de.

Federal IT Steering Unit, 2013b. Faktenblatt Nr. 5: Bundesinformatik 2012-2015. , (5), pp.1–8.

Federal IT Steering Unit, 2015. Faktenblatt Nr. 6: Bundesinformatik 2012-2015. , (6), pp.1–8.

Federal IT Steering Unit, 2011. *IKT-Strategie des Bundes 2012–2015*,

Federal IT Steering Unit, 2010. *Studie HERMES und Agilität*,

Flick, U., 2009. Introduction to Qualitative Research. In *An Introduction to Qualitative Research*. SAGE Puclications, pp. 1–18.

Frühauf, K. et al., 2011. IREB Certified Professional for Requirements Engineering - Foundation Level - Syllabus. , 2011(March), p.29.

Goyal, S. (Technical U.M., 2007. Major Seminar on Feature Driven Development.

Grau, B.R. et al., 2014. Requirements Engineering and Agile Development - collaborative , just enough , just in time , sustainable - Content. , pp.1–18.

Günal, V., 2012. Agile Software Development Approaches and Their History.

Hathaway, T. & Hathaway, A., 2014. What is Business Analysis? In *Business Analysis Definded: Lean what BA is, when to do it in Agile and Waterfall SDM, and what techniques are in use.* BA-EXPERTS, p. 81. Available at: http://books.google.ch/books?id=e14mBAAAQBAJ&printsec=frontcover&dq=Definition+Business+Analysis&hl=de&sa=X&ei=--EbVJfrD8TnyQOi4ILQDQ&ved=0CEMQ6AEwAg#v=onepage&q=Definition Business Analysis&f=false.

iSport, 2014. The Rules of the Scrum. Available at: http://rugby.isport.com/rugby-guides/the-rules-of-the-scrum [Accessed July 6, 2014].

Janssen, C., 2014. Techopedia: Software Development Process. Available at: http://www.techopedia.com/definition/13295/software-development-process [Accessed September 19, 2014].

Van Der Merwe, A., 2014. *The Research Onion*,

Morgue d'Algue, H., 2014. Information about HERMES 5 and SCRUM. [e-mail] (Personal communication, 22 july 2014).

Oxford University Press, 2014. agile definition of agile in Oxford dictionary (American English). Available at: http://www.oxforddictionaries.com/definition/american_english/agile#agile__12 [Accessed April 28, 2012].

Perring, R.J., 2010. Lean Meets Agile Software Development & Project Management Lean Meets Agile Software Development and Project Management. , p.12. Available at: http://www.aspe-sdlc.com/offers/pickups_0386572/lean_meets_agile.pdf.

Pohl, K. & Rupp, C., 2011. *Basiswissen Requirements Engineering*, dpunkt.verlag GmbH.

Rupp, C. & SOPHISTen, 2014. *Requirements Engineering und Management* 6th Editio., Carl Hanser Verlag München Wien.

Schwaber, K. & Sutherland, J., 2012. *Software in 30 Tagen* 1st ed., dpunkt.verlag GmbH.

SCRUM Alliance, 2012. Core Scrum. Available at: http://www.scrumalliance.org/why-scrum/core-scrum-values-roles [Accessed July 11, 2014].

DAS SCRUM Team AG, 2012. *SCRUM Reader*,

Siegel, D., 2002. Qualitative versus Quantitative. Available at: http://www.gifted.uconn.edu/siegle/research/qualitative/qualquan.htm [Accessed July 19, 2014].

Sutherland, J., Agile Principles and Values. Available at: http://msdn.microsoft.com/en-us/library/dd997578.aspx [Accessed May 21, 2014].

Sutherland, J. & Schwaber, K., 2011. The Scrum Guide. , (October).

Voigt, B.J.J., 2004. Dynamic System Development Method. , (January).

Wang, X. & Conboy, K., 2009. UNDERSTANDING AGILITY IN SOFTWARE DEVELOPMENT.

Wirdemann, R., 2011. *SCRUM mit User Stories* 2nd ed., Carl Hanser Verlag München Wien.

ZETWAL SARL, Coloriages Gratuites.

10. Statement of Authenticity

I the undersigned declare that all material presented in this paper is my own work or fully and specifically acknowledged wherever adapted from other sources.

I declare that all statements and information contained herein are true, correct and accurate to the best of my knowledge and belief.

Grenchen, 31th of January 2015

Birgit Schär

11. List of Figures

12. List of Tables

13. List of Abbreviations

DDPS	Department of Defence, Civil Protection and Sport
DETEC	Department of the Environment, Transport, Energy and Communications
dt.	In deutscher Sprache
EAR	Department of Economic Affairs, Education and Research
eng.	In english language
etc.	Et cetera
e.g.	For example
FDF	Federal Department of Finance
FDFA	Federal Department of Foreign Affairs
FDHA	Federal Department of Home Affairs
FDJP	Federal Department of Justice and Police
FITSU	Federal IT Steering Unit
FOITT	Federal Office for Information Technology and Telecommunication
HERMES	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)
ICT	Information and Communication Technology
IT	Information Technology
IREB	International Requirements Engineering Board
n.d.	No date
PDSA	Plan, Do, Study, Act
PRQ	Primary Research Question
RE	Requirements Engineering
TS	Thesis Statement

14. Appendix

14.1 Appendix 1: Agile HERMES Scenario (Work Break Down)

Phase	Module	Task	Result
Initiation	Project Steering	commission and steer initiation	project initiation order*
		decide on project release	checklist
			project charter*
			project decision steering*
	Project Management	manage and control initiation	work order
			project status report*
			minutes
			stakeholder list*
		decide on option	checklist
			project decision management / execution*
		design a project charter	project management plan*
			project charter*
	Project Foundations	conduct a study	study*
		analyze the legal framework	analysis of legal framework*
		analyze protection needs	protection needs analysis*
Concept	Project Steering	steer a project	QA- and risk report
			project decision steering
		decide on phase release	checklist
			QA- and risk report
			project decision steering*
	Project Management	manage and control a project	project management plan*
			work order
			project status report*
			minutes
		agree on and control deliverables	enquiry
			evaluation report*
			agreement
		deal with problems and benefit from lessons learned	lessons learned
		manage stakeholders and communication	project management plan*
			stakeholder list*

1

1

		perform quality assurance	project management plan*
			inspection report
		manage risks	project management plan*
			project status report*
		lead change management*	project management plan*
			amendment request*
			change status list*
		prepare phase release	project management plan*
			project status report*
			phase report*
	Procurement	create a procurement plan	project management plan*
		prepare a call for tenders	tender documentation*
		issue a call for tenders	tender documentation*
			offer*
		evaluate offers	evaluation report*
			minutes
		decide on call for tender	checklist
			project decision steering*
		Decide on contract award	checklist
			Project decision steering*
			publication*
		Draw up an agreement	agreement
	Organizational Structure	design a concept for the organization structure	concept for organizational structure*
	Deployment Organization	design deployment concept	deployment concept*
	IT System	design a system concept	situation analysis
			system requirements*
			detailed study
			system architecture*
		implement prototype	prototype developed
			prototype documentation
		design an integration concept	integration concept
		decide on system architecture	checklist
			project decision management / execution*
	IT Operation	design an operating concept	operating concept*
			agreement
	IT Migration	design an migration concept	migration concept*

	Testing	design a test concept	test concept*
	Information Security and Data Protection	create an ISDP concept	ISDP concept*
		decide on ISDP concept	checklist
			project decision management / execution*
	Agile Development	decide on agile development using SCRUM	checklist
			project decision management / execution*
		introduce SCRUM	project management plan*
		keep a product backlog	product backlog*
		create a release plan	release plan
		work in sprints	sprint backlog*
			increment*
			minutes
Implementation	Project Steering	steer a project	QA- and risk report
			project decision steering
		decide on phase release	checklist
			QA- and risk report
			project decision steering*
	Project Management	manage and control a project	project management plan*
			work order
			project status report*
			minutes
		agree on and control deliverables	offer query
			evaluation report*
			agreement
		deal with problems and benefit from lessons learned	lessons learned
		manage stakeholders and communication	project management plan*
			stakeholder list*
		perform quality assurance	project management plan*
			inspection report
		manage risks	project management plan*
			project status report*
		lead change management	project management plan*
			amendment request*
			change status list*
		prepare phase release	project management plan*
			project status report*

5

2

			phase report*
	Organizational Structure	Implement organizational structure	process description*
			description of organization*
			organization realized
	Deployment Organization	Prepare deployment	deployment measures and organization implemented
		decide on preliminary acceptance	acceptance report*
			checklist
			project decision management / execution*
	IT System	prototype completed	prototype completed
			prototype documentation
		system developed / parameterized	detailed specification
			system developed and parameterized
			system architecture*
			user manual
		prepare system integration	interface realized
			system architecture*
			integration and installation manual
			detailed specification
	IT Operation	operation completed	operation infrastructure implemented
			operating manual*
			operational structure implemented
		integrate a system within operation	operating manual
			system integrated
	IT Migration	implement migration procedure	detailed specification
			migration procedure
	Testing	implement test infrastructure	test system
			test data
		conduct testing	test report*
			test concept*
	Information Security and Data Protection	implement ISDP concept	ISDP measures*
			ISDP concept*
	Agile Development	keep a product backlog	product backlog*
		design a release plan	release plan

5

		work in sprints	sprint backlog*
			increment*
			minutes
Deployment	Project Steering	steer a project	QA- and risk report
			project decision steering
		take decision for project closure	QA- and risk report
			checklist
			project decision steering*
	Project Management	manage and control a project	project management plan*
			work order
			project status report*
			minutes
		agree on and control deliverables	enquiry
			evaluation report*
			agreement
		deal with problems and benefit from lessons learned	lessons learned
		manage stakeholders and communication	project management plan*
			stakeholder list*
		perform quality assurance	project management plan*
			inspection report
		manage risks	project management plan*
			project status report*
		manage change management	project management plan*
			amendment request*
			change status list*
		prepare project closure	lessons learned
			final project evaluation*
	Organizational Structure	activate organizational structure	organizational structure activated
	Deployment Organization	execute deployment	deployment measures executed
		decide on launching operation	checklist
			project decision steering*
		decide on acceptance	acceptance report*
			checklist
			project decision management / execution*
	IT System	activate system	system activated

2

	IT Operation	activate operation	operating manual*
			operation activated
	IT Migration	conduct migration	migration executed
		decide on acceptance of migration	acceptance report*
			checklist
			project decision management / execution
		decommission legacy system	legacy system decommissioned
	Testing	conduct tests	test report*
			test concept*
		transfer test concept and infrastructure	minutes
	Agile Development	keep a product backlog	product backlog*
		design a release plan	release plan
		work in sprints	sprint backlog*
			increment*
			minutes
	Information Security and Data Protection	transfer ISDP concept	ISDS concept*
			checklist

Table 31: Work break down structure (based on (Eicher, Kruschitz & Mourgue d'Algue 2014))

14.2 Appendix 2: Analyzed Modules and Tasks

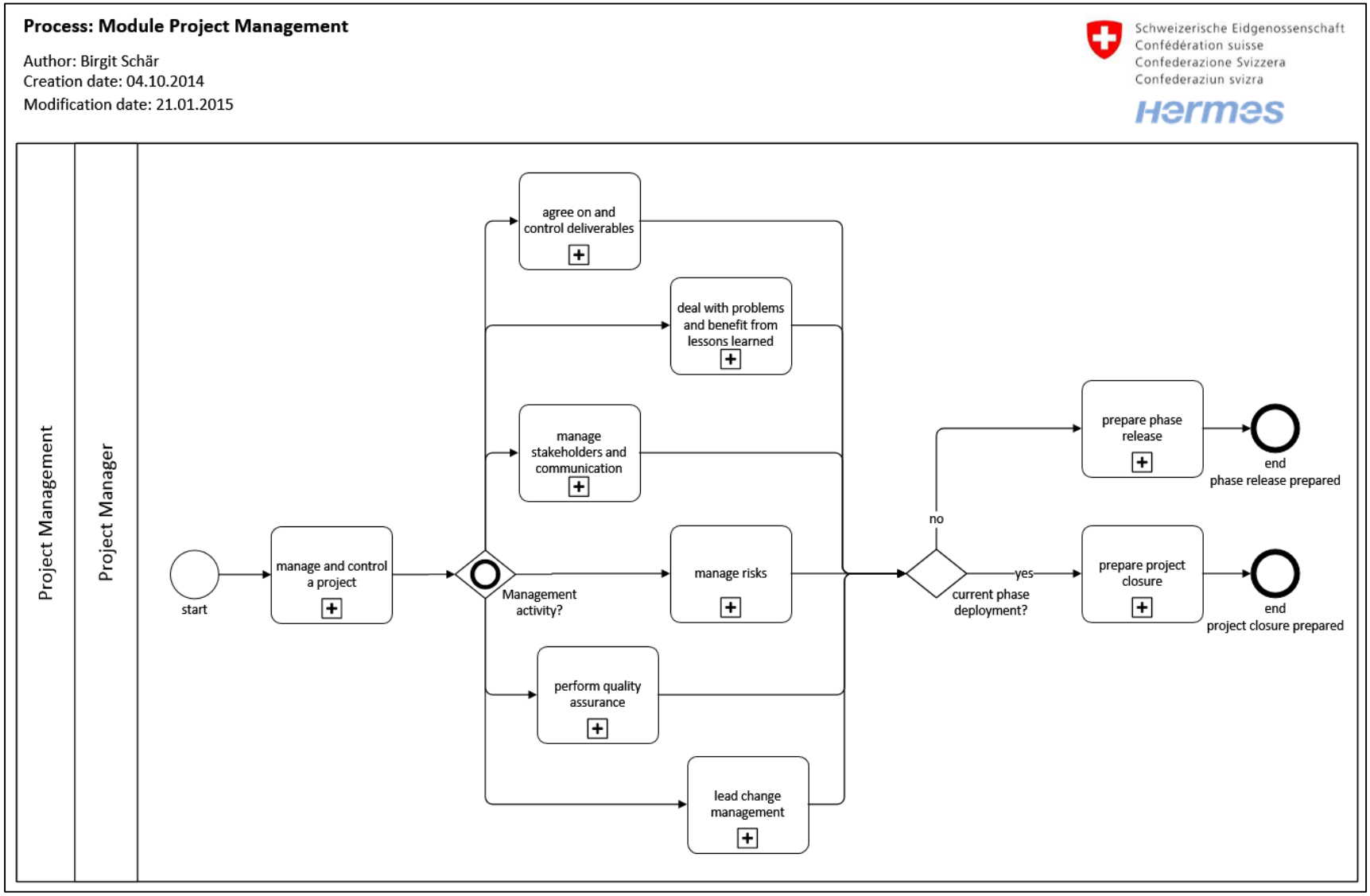
Module (M) / Task (T)	Analysis Part 1	Analysis Part 2	Analysis Part 3
M: Project Management			
T: Manage and control a project	x		
T: Agree and control deliverables	x		
T: Deal with problems and lessons learned	x		
T: Manage stakeholders and communication	x		
T: Perform quality assurance	x	x	
T: Manage risks	x		
T: Lead change management	x	x	
T: Prepare phase release	x		
T: Prepare project closure	x		
M: Project Steering			
T: Steer a project	x		
T: Decide on phase release	x		
T: Decide on project closure	x		
M: Agile Development			
T: Decide on agile development using SCRUM	x		x
T: Introduce SCRUM	x		x
T: Keep a product backlog	x		x
T: Design a release plan	x		x
T: Work in sprints	x		x
M: IT System			
T: Design a system concept	x	x	
T: Design an integration concept	x		
T: Implement prototype	x		
T: Decide on system architecture	x		
M: Procurement			
T: Create procurement plan	x		
T: Prepare a call for tender	x		
T: Decide on call for tender	x		
T: Issue a call for tender	x		
T: Evaluate offers	x		
T: Decide on contract award	x		
T: Draw up an agreement	x		
M: Organizational Structure			
T: Design a concept for the organizational structure	x		
T: Implement organizational structure	x		
T: Activate organizational structure	x		
M: Deployment Organization			
T: Design deployment concept	x		
T: Prepare deployment	x		
T: Decide on preliminary acceptance	x		
T: Execute deployment	x		
T: Decide on launching operation	x		
T: Decide on acceptance	x		
M: IT Operation			
T: Design an operating concept	x		
T: Implement operation	x		
T: Integrate system into operation environment	x		
T: Activate operation	x		
M: IT Migration			

T: Design a migration concept	x		
T: Implement migration procedure	x		
T: Conduct migration	x		
T: Decide on acceptance of migration	x		
T: Decommission legacy systems	x		
M: Testing			
T: Design a test concept	x		
T: Set up a test infrastructure	x		
T: Conduct testing	x		
T: Transfer test concept and test infrastructure	x		
M: Information Security and Data Protection			
T: Create an ISDP concept	x		
T: Decide on ISDP concept	x		
T: Realize ISDP concept	x		
T: Transfer ISDP concept	x		

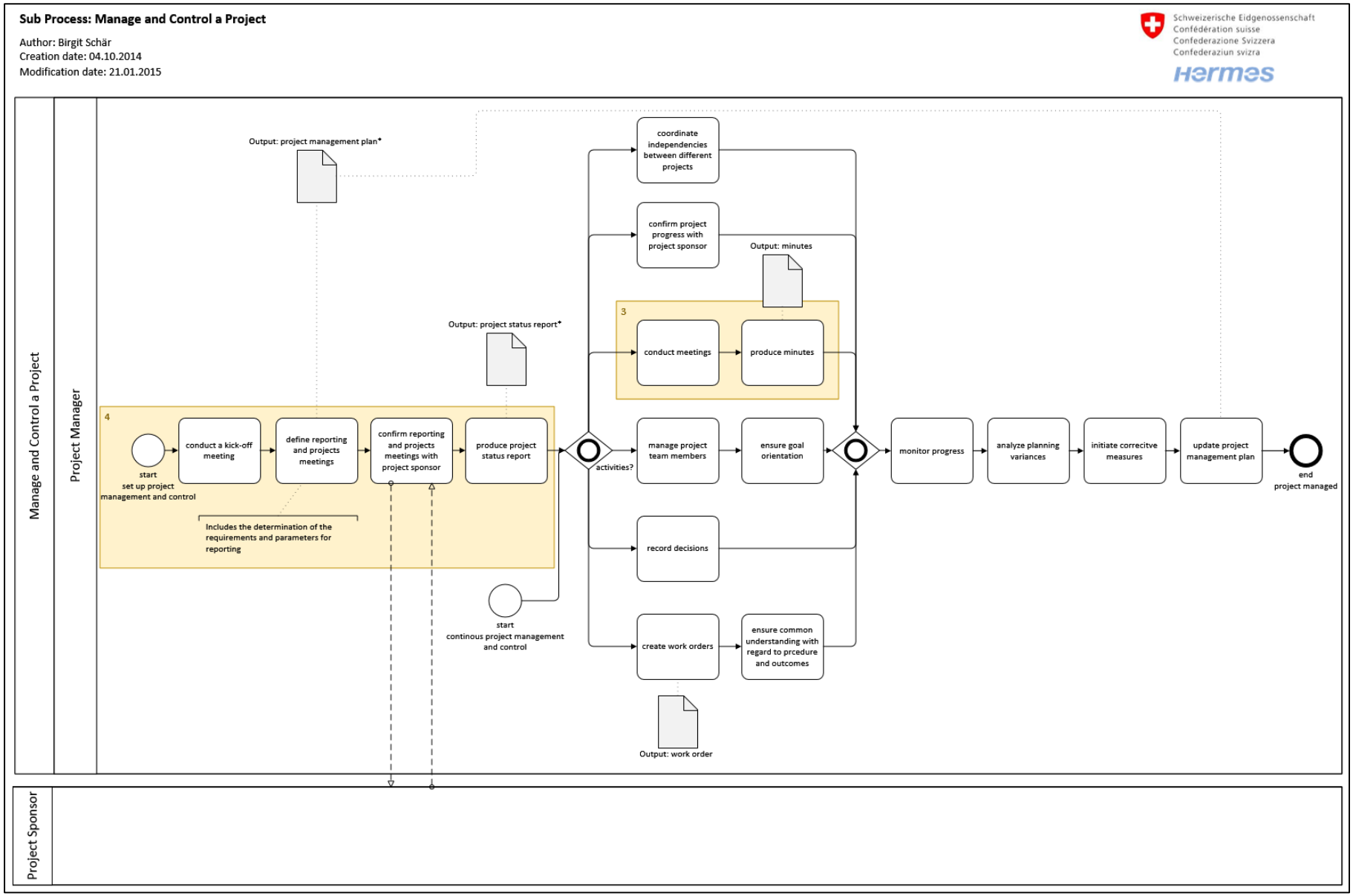
Table 32: Content of the detailed analysis

14.3 Appendix 3: Process Models for Analysis Part 1

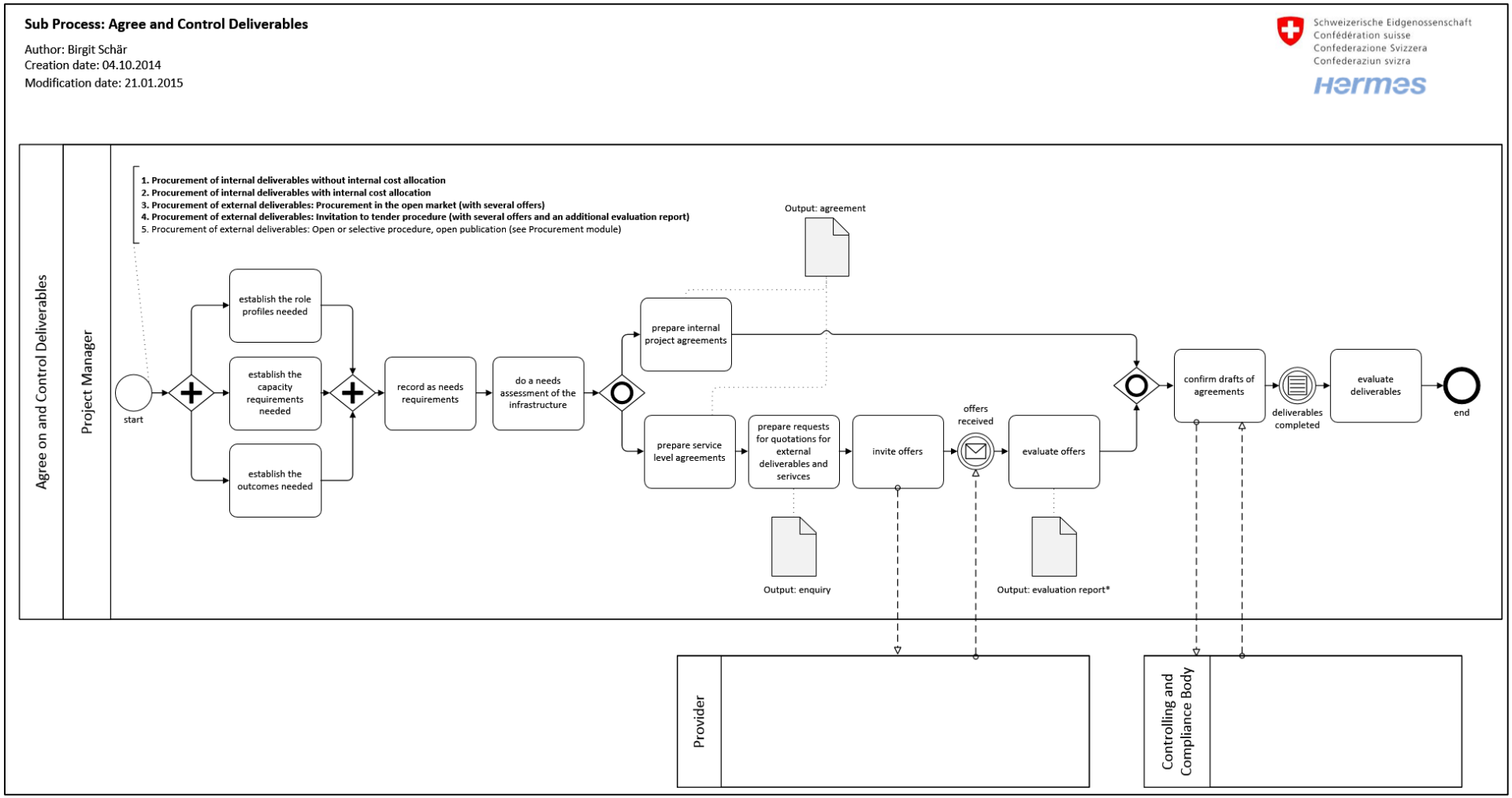
14.3.1 Module “Project Management”



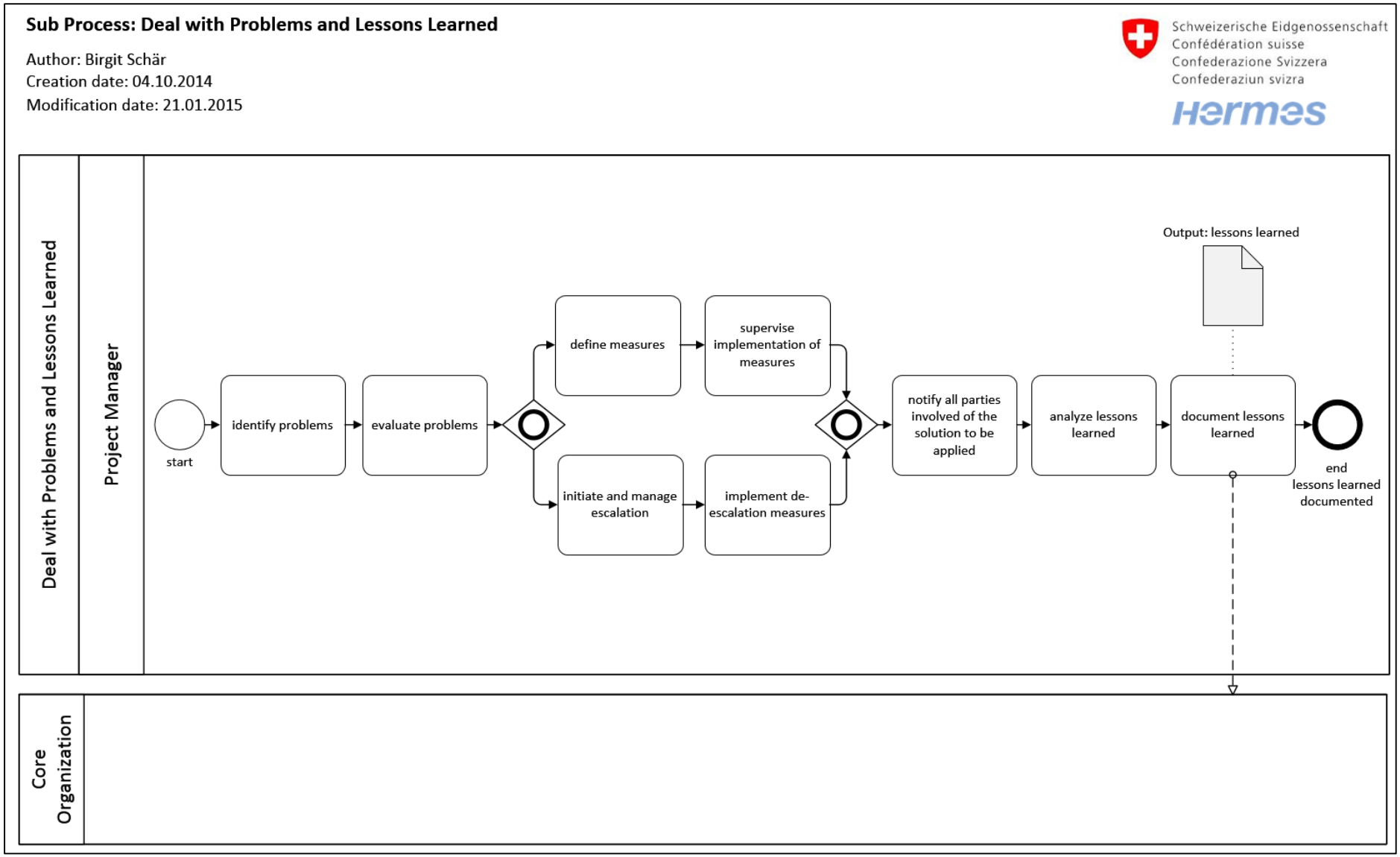
14.3.1.1 Manage and Control a Project



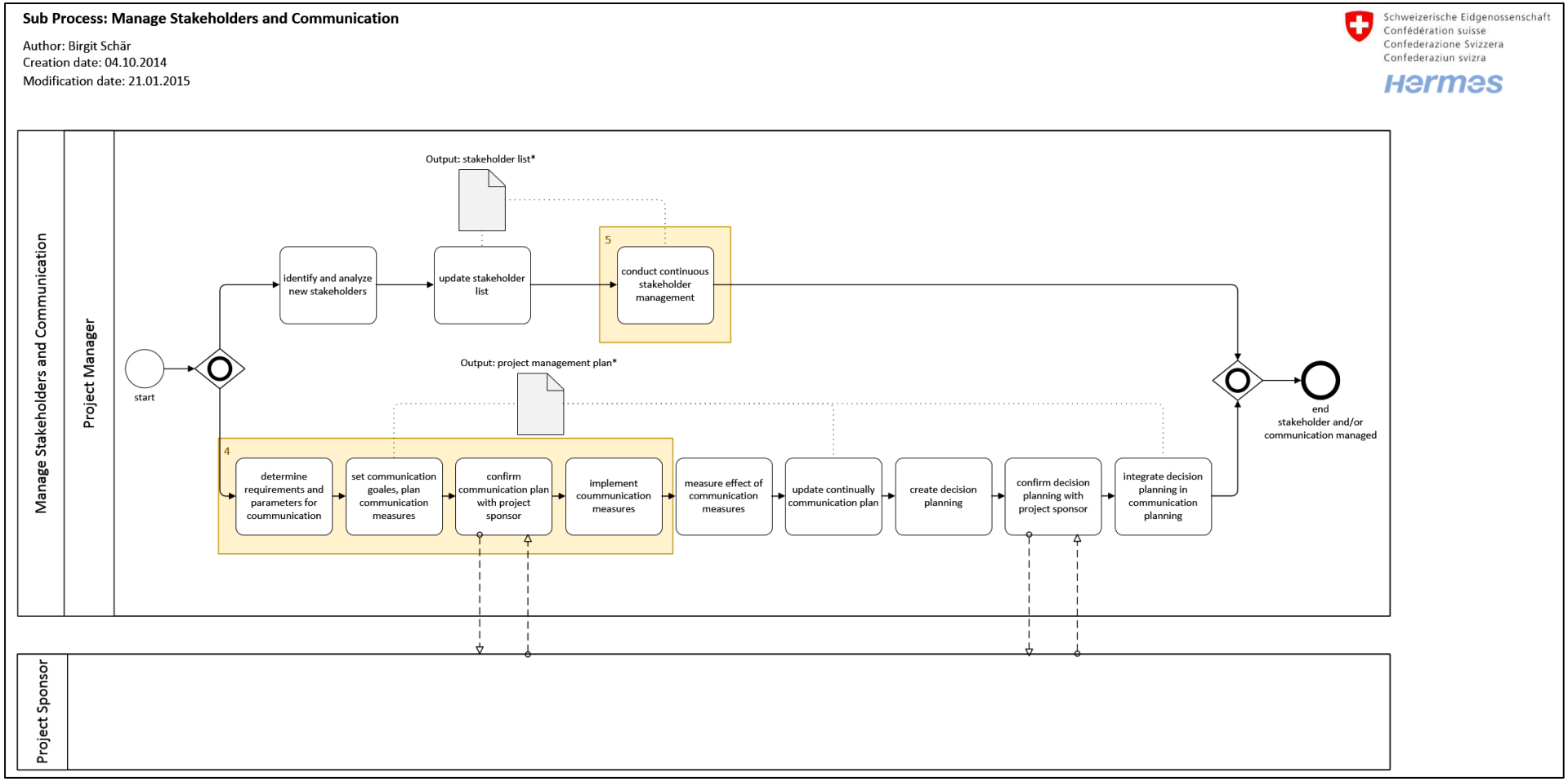
14.3.1.2 Agree and Control Deliverables



14.3.1.3 Deal with Problems and Lessons Learned



14.3.1.4 Manage Stakeholders and Communication

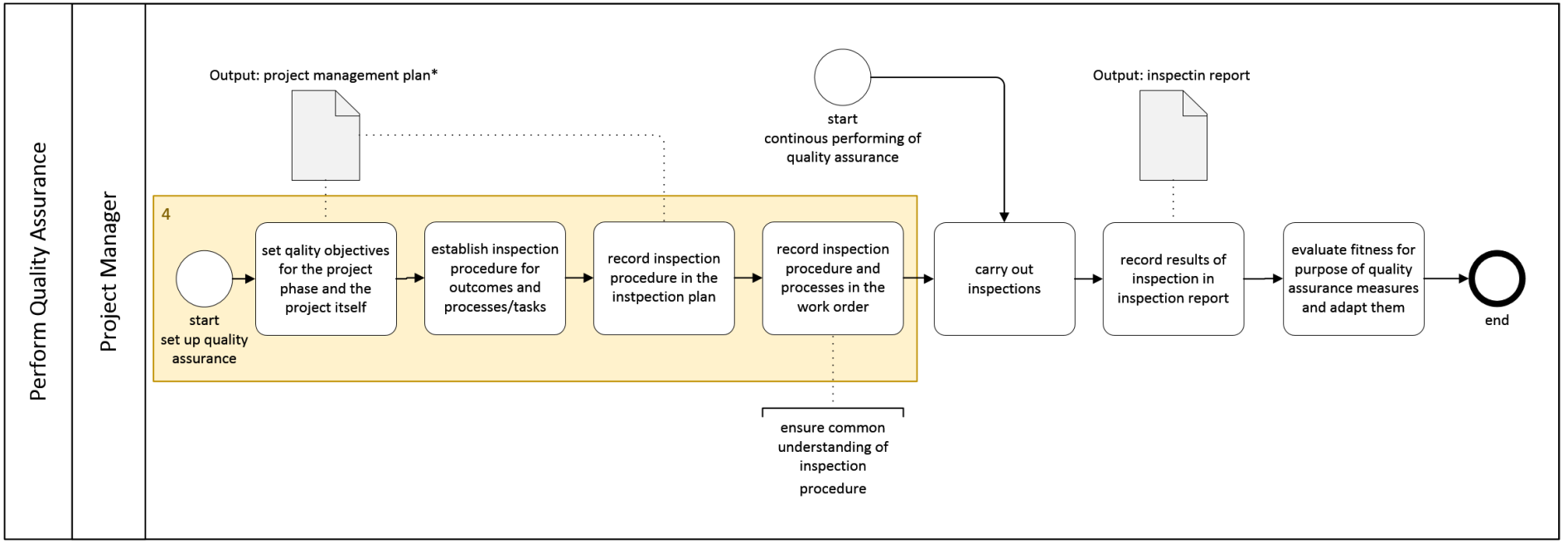


14.3.1.5 Perform Quality Assurance

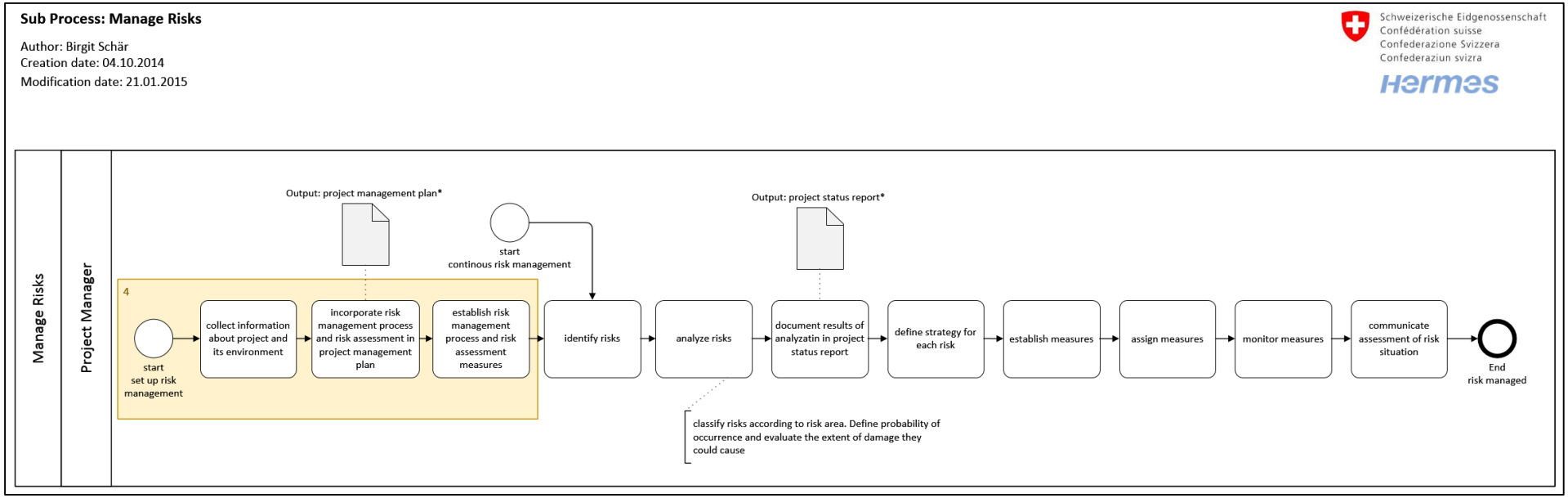
Sub Process: Perform Quality Assurance

Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

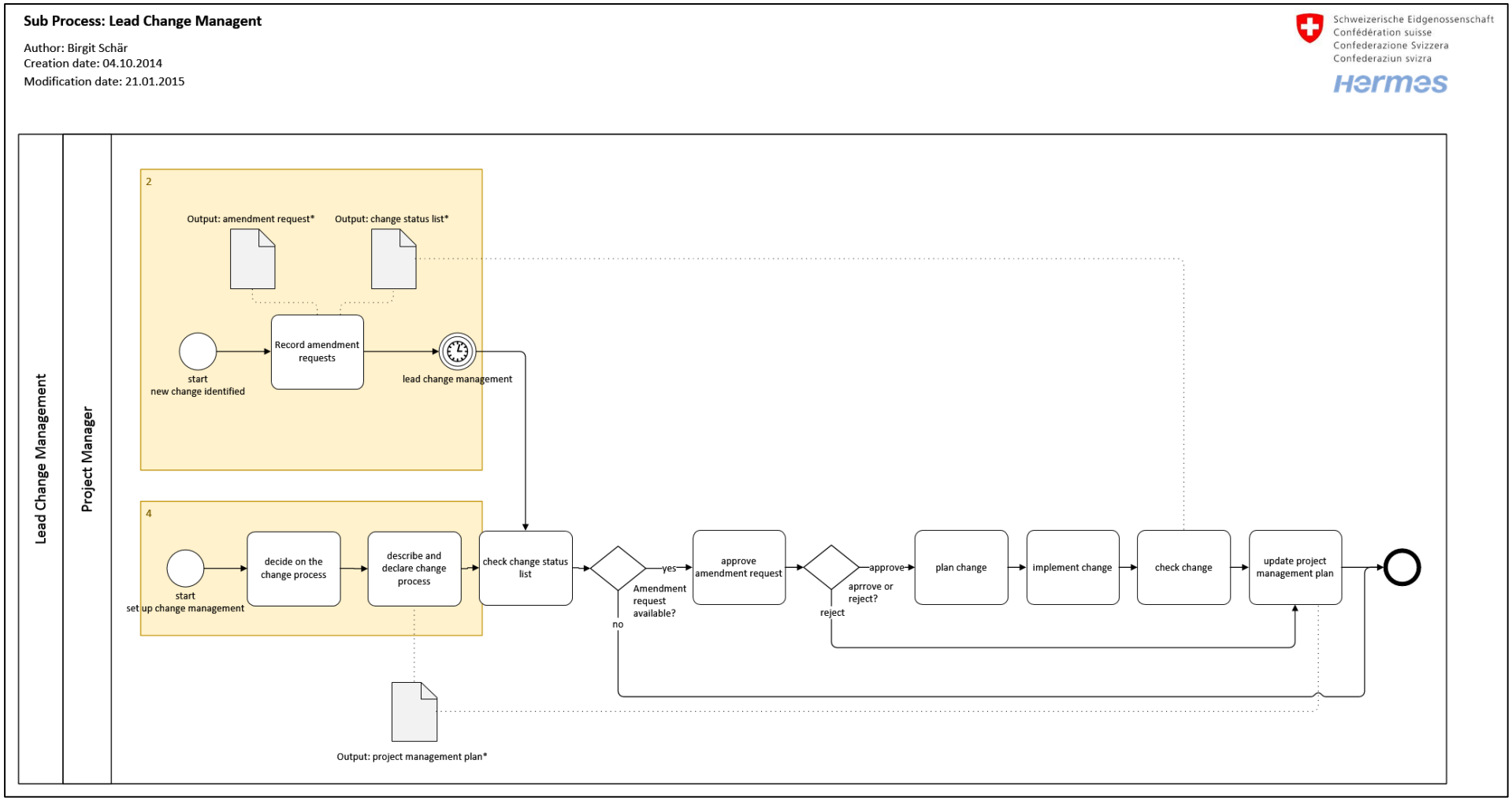
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Hermes



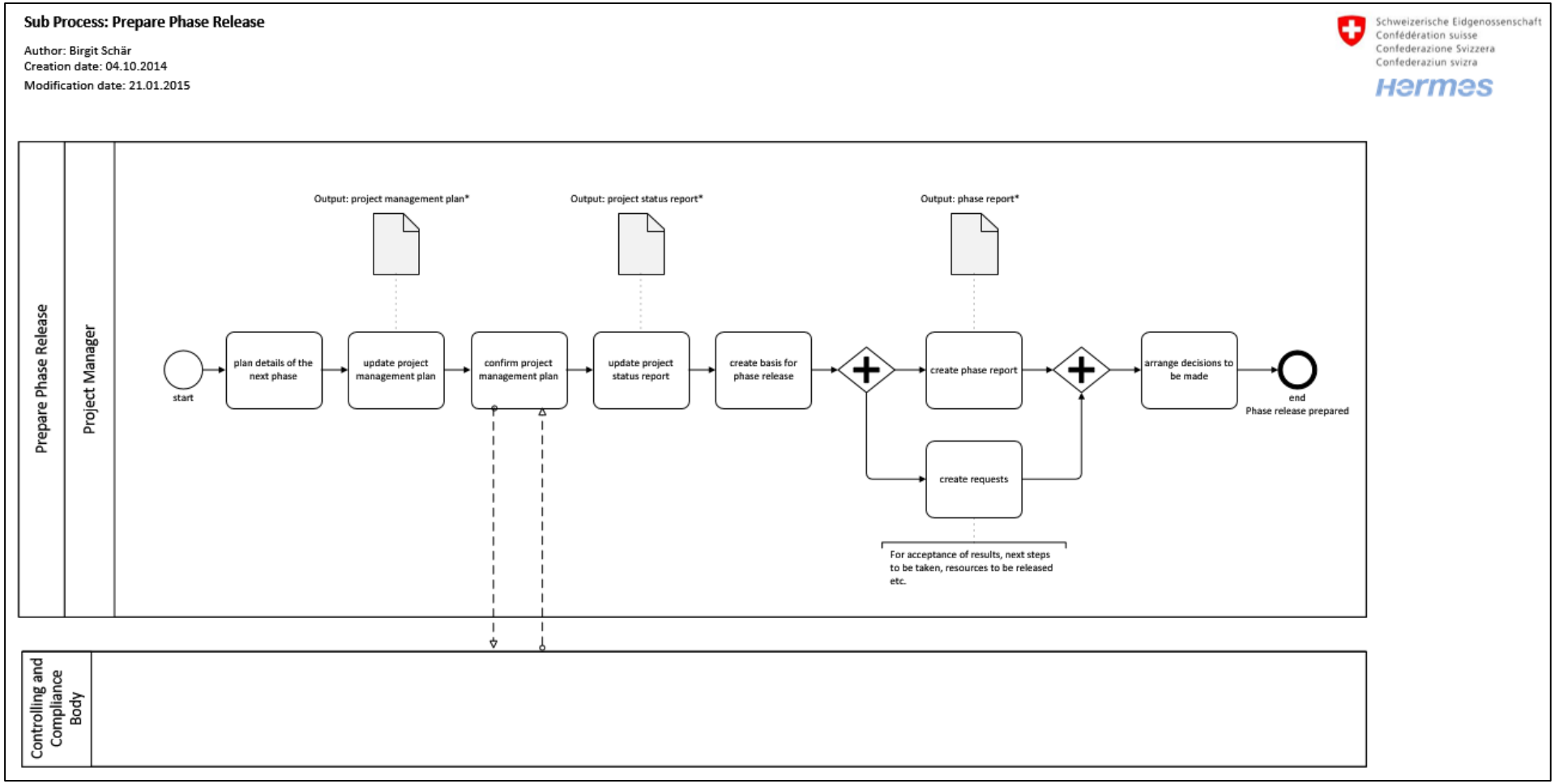
14.3.1.6 Manage Risks



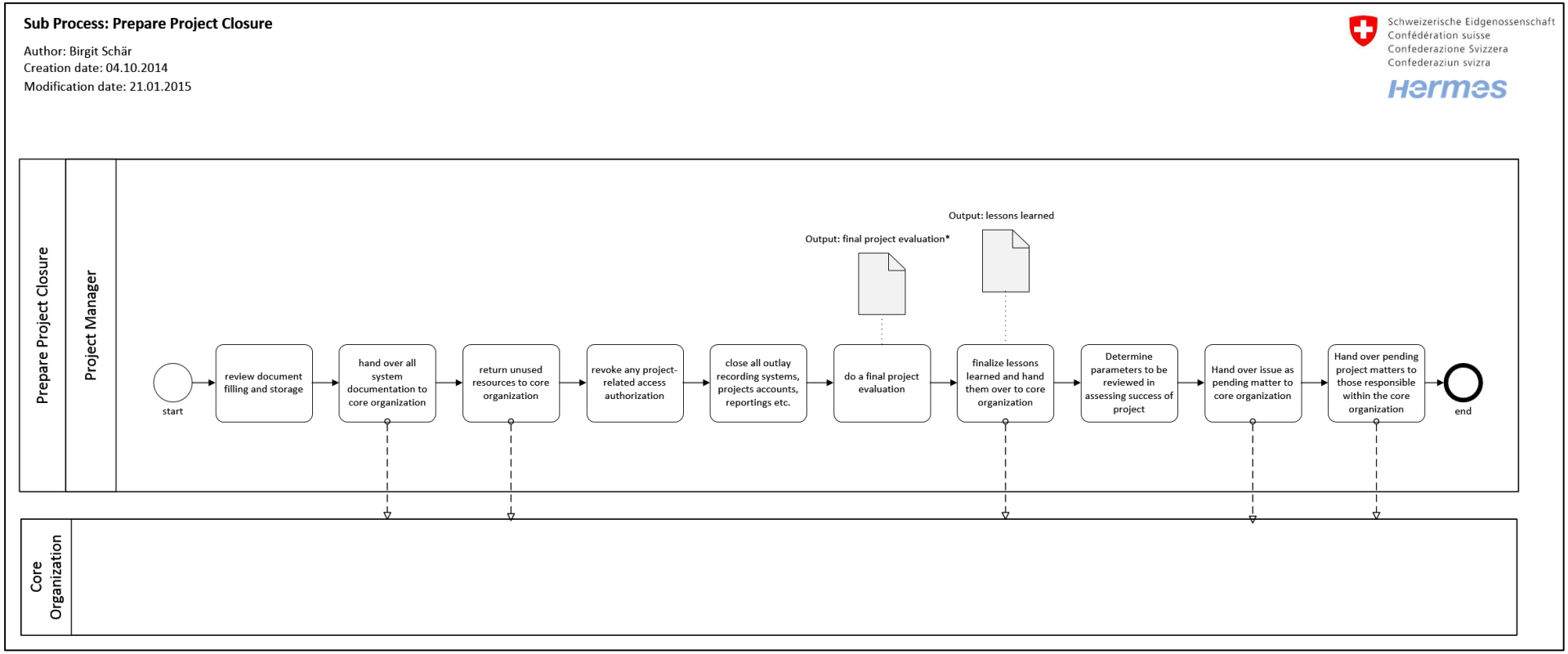
14.3.1.7 Lead Change Management



14.3.1.8 Prepare Phase Release



14.3.1.9 Prepare Project Closure



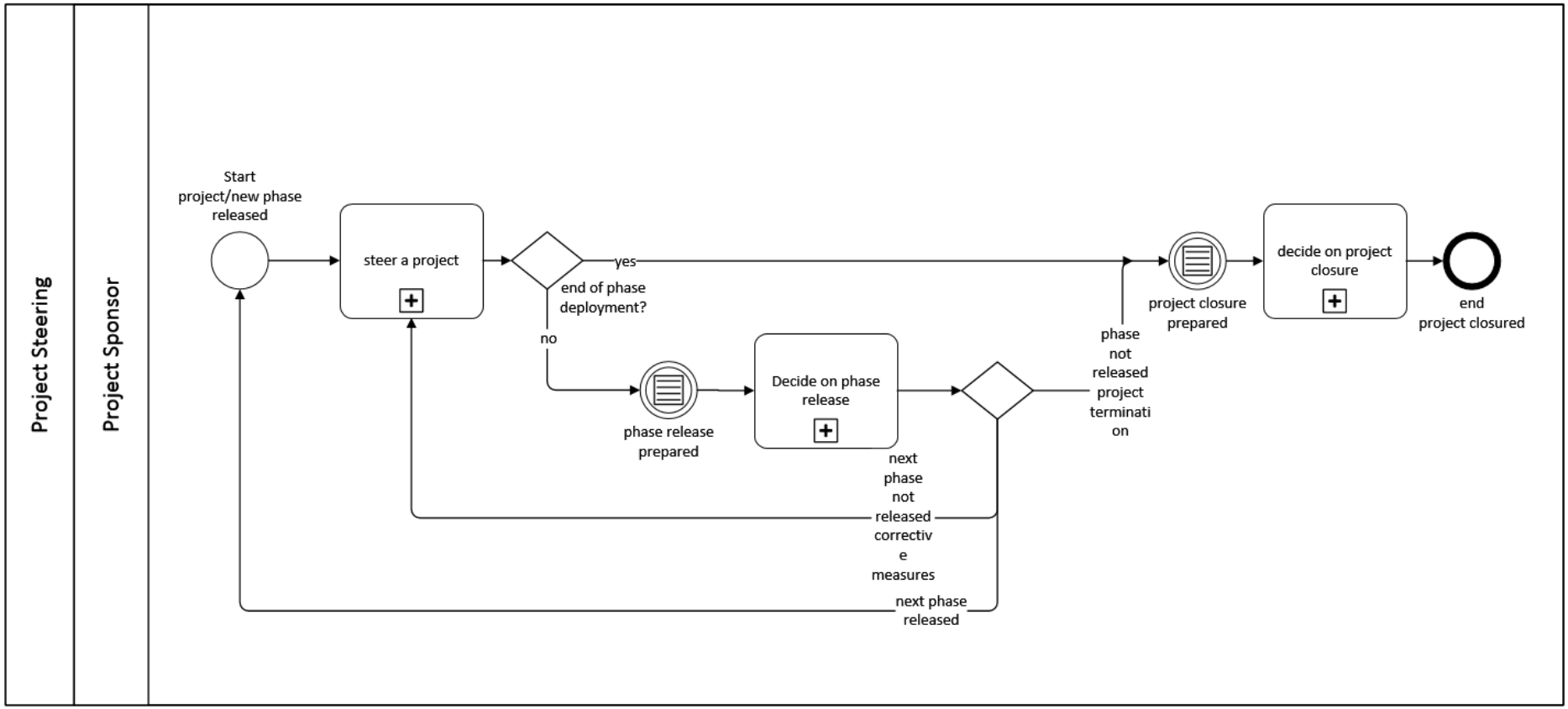
14.3.2 Module “Project Steering”

Process: Module Project Steering

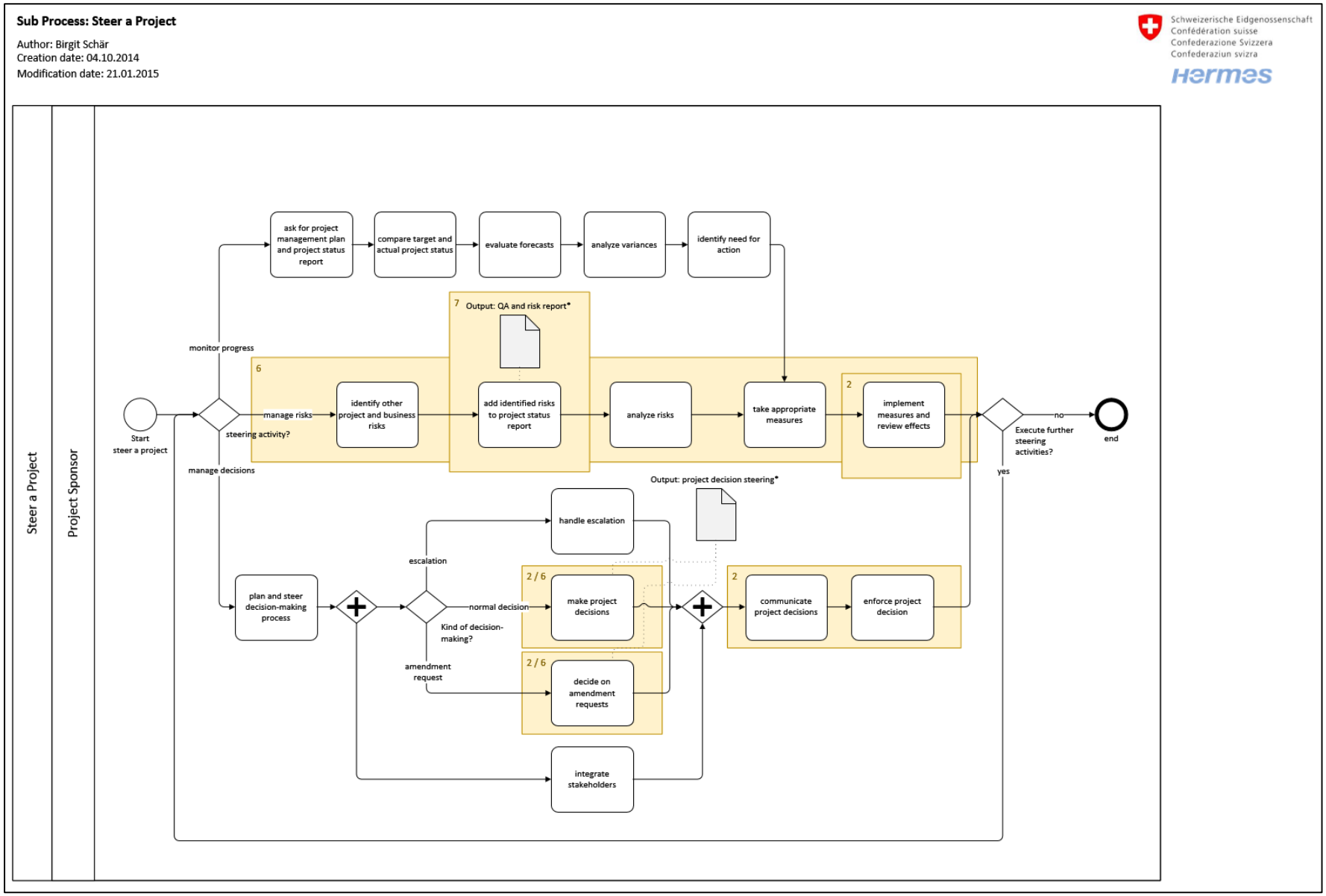
Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

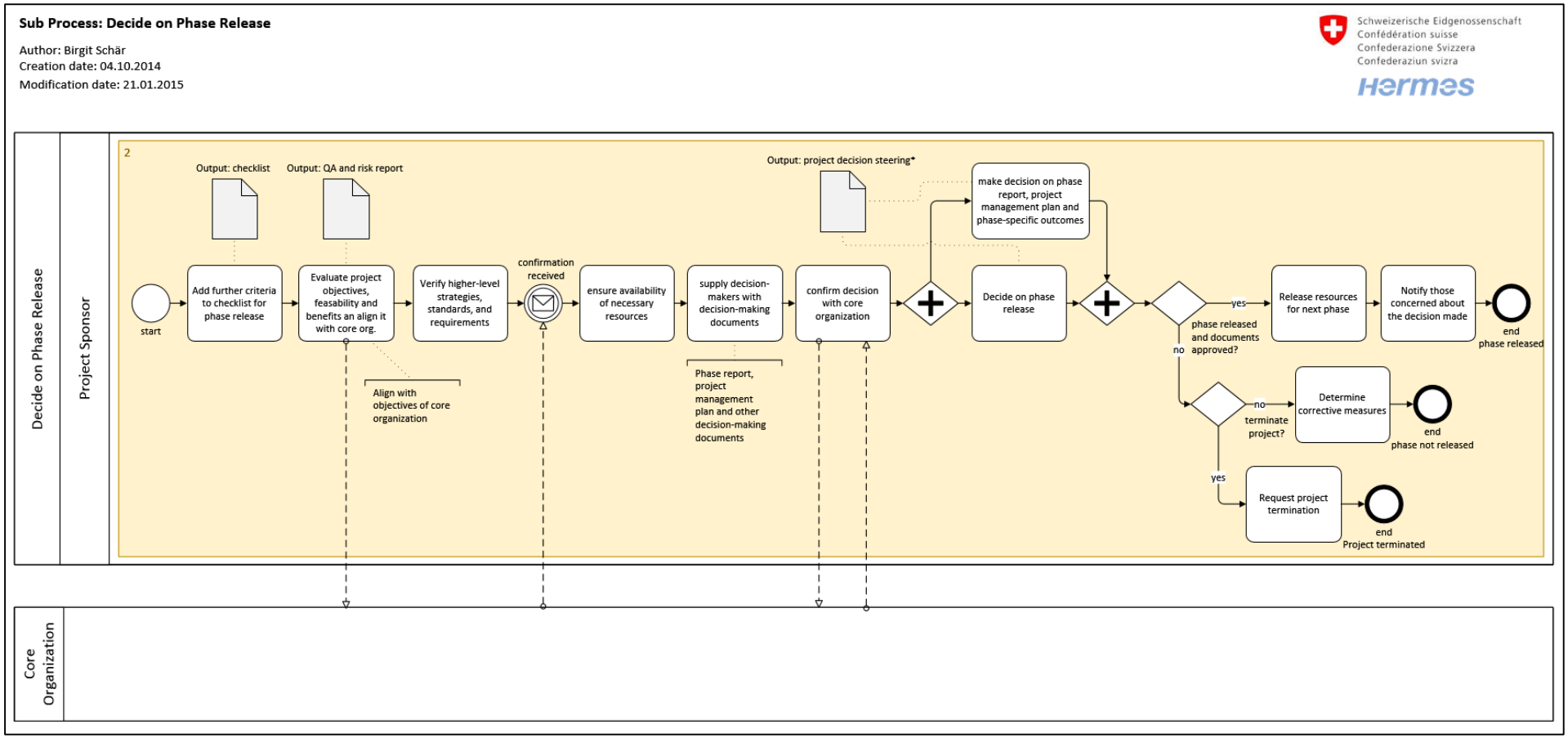
Hermes



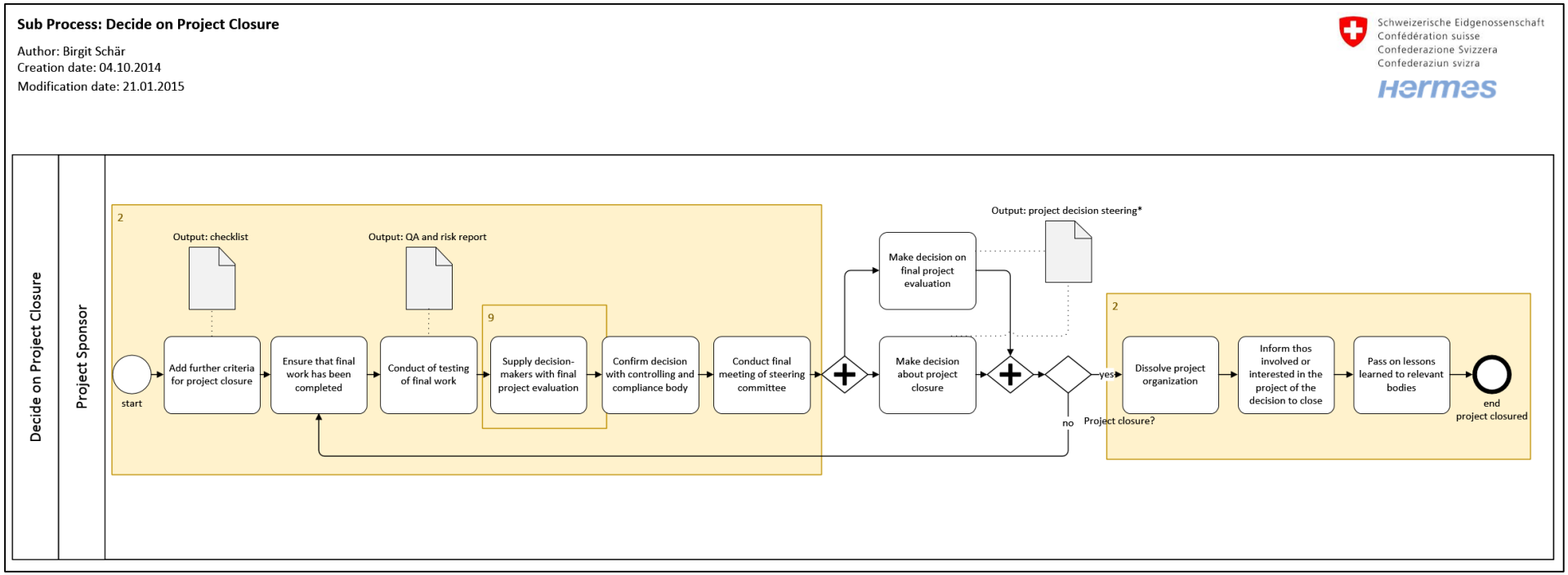
14.3.2.1 Steer a Project



14.3.2.2 Decide on Phase Release



14.3.2.3 Decide on Project Closure

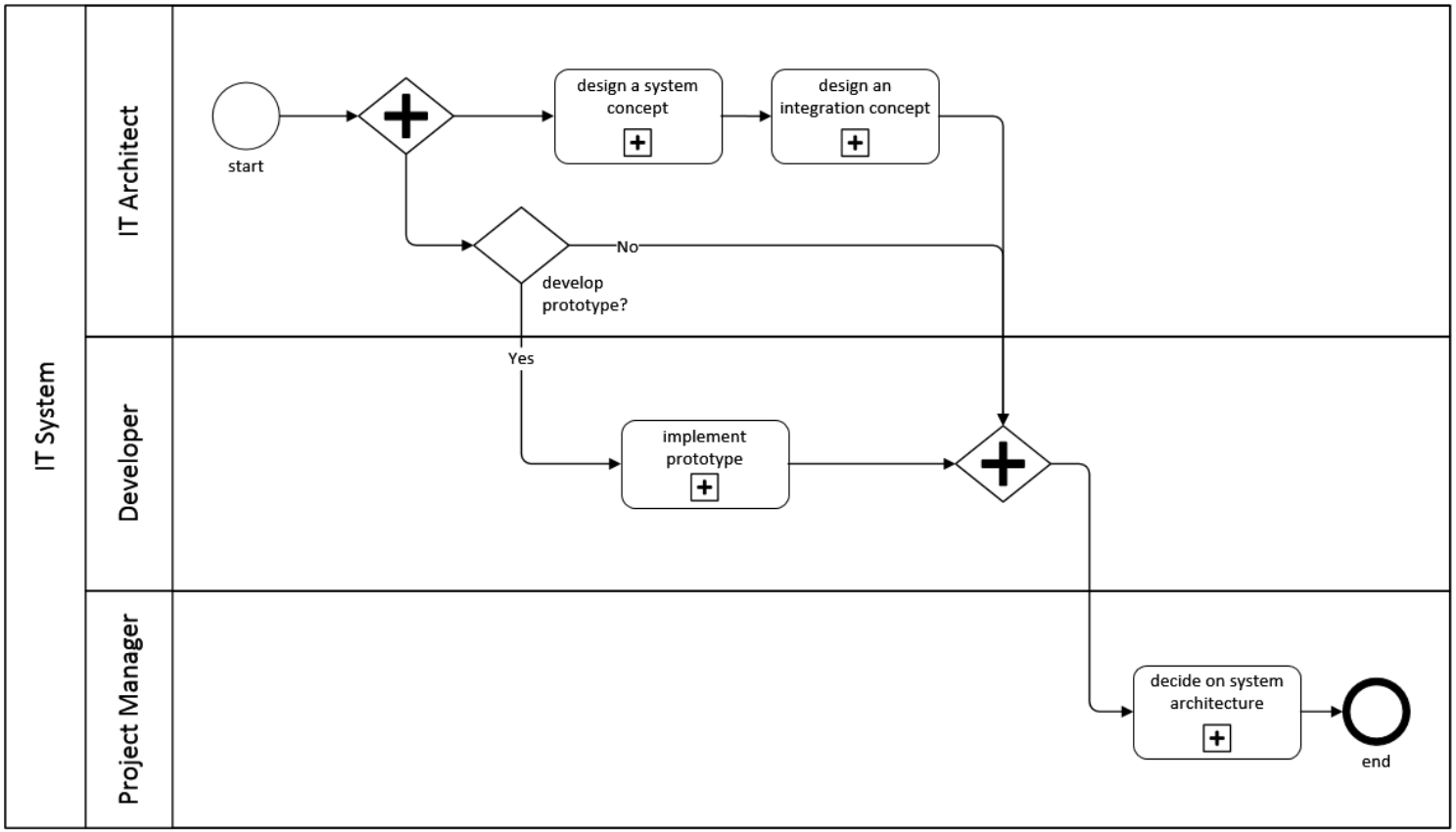


14.3.3 Module “IT System”

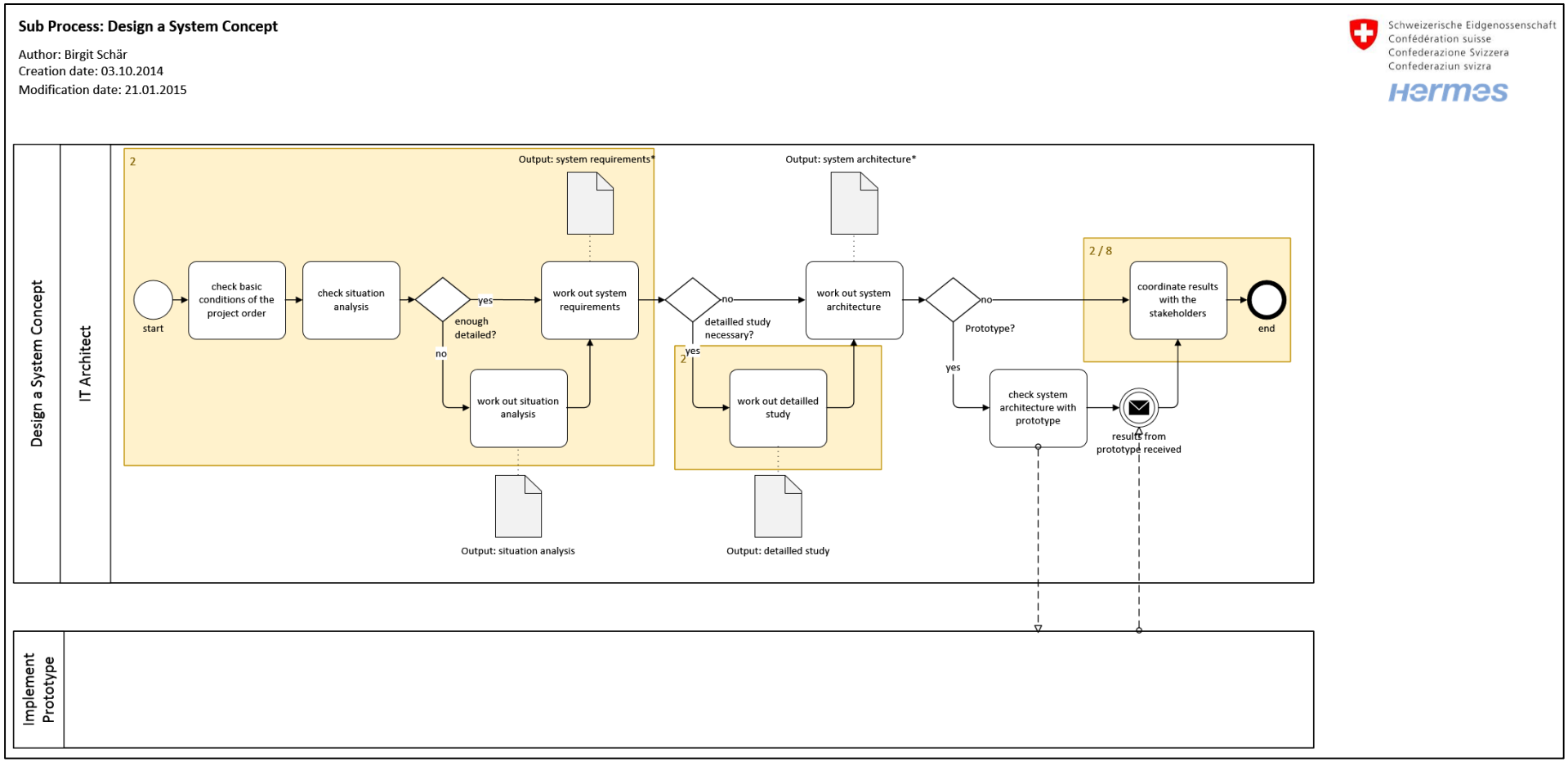
Process: Module IT System

Author: Birgit Schär
Creation date: 03.10.2014
Modification date: 21.01.2015

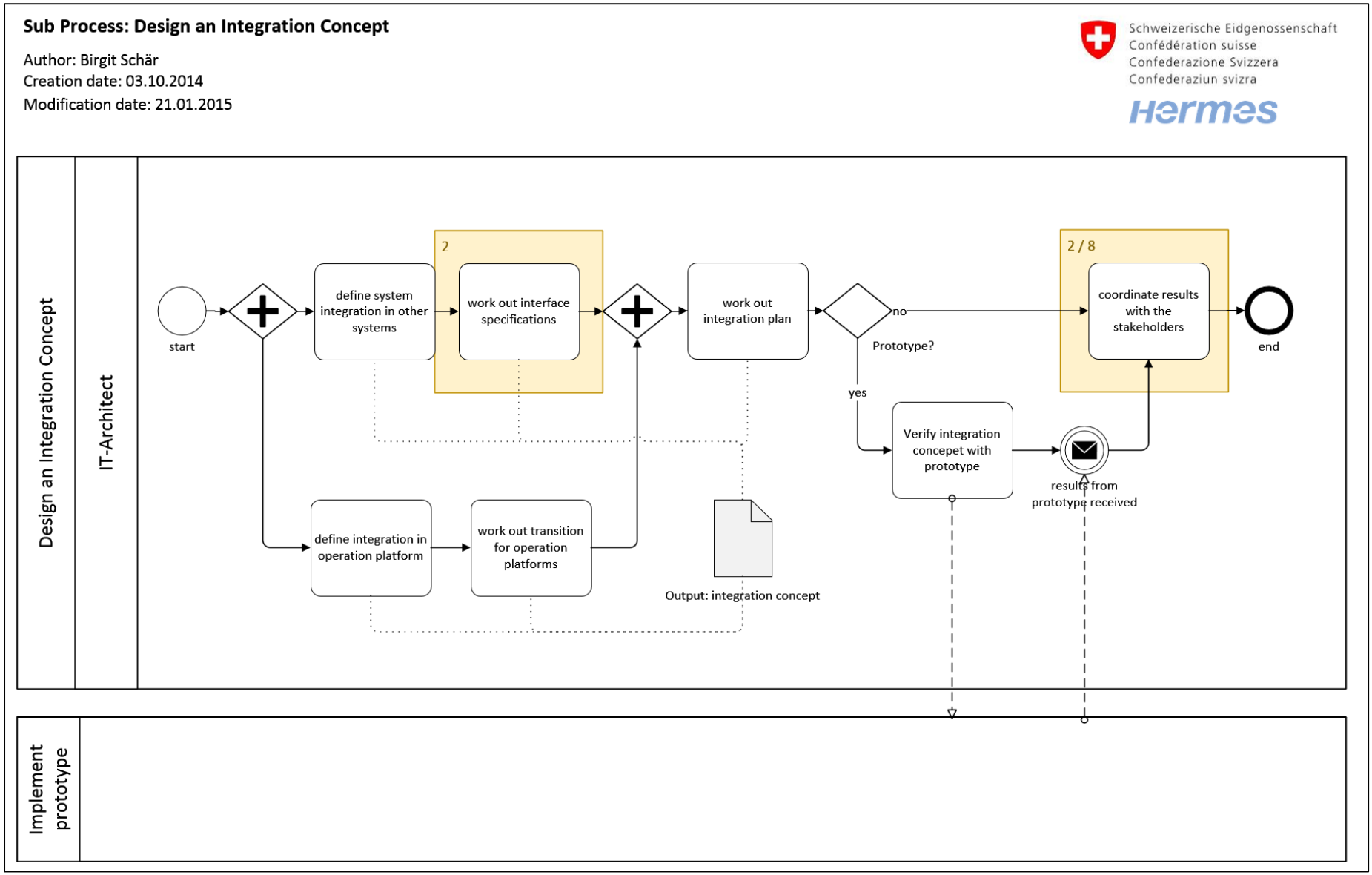
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Hermes



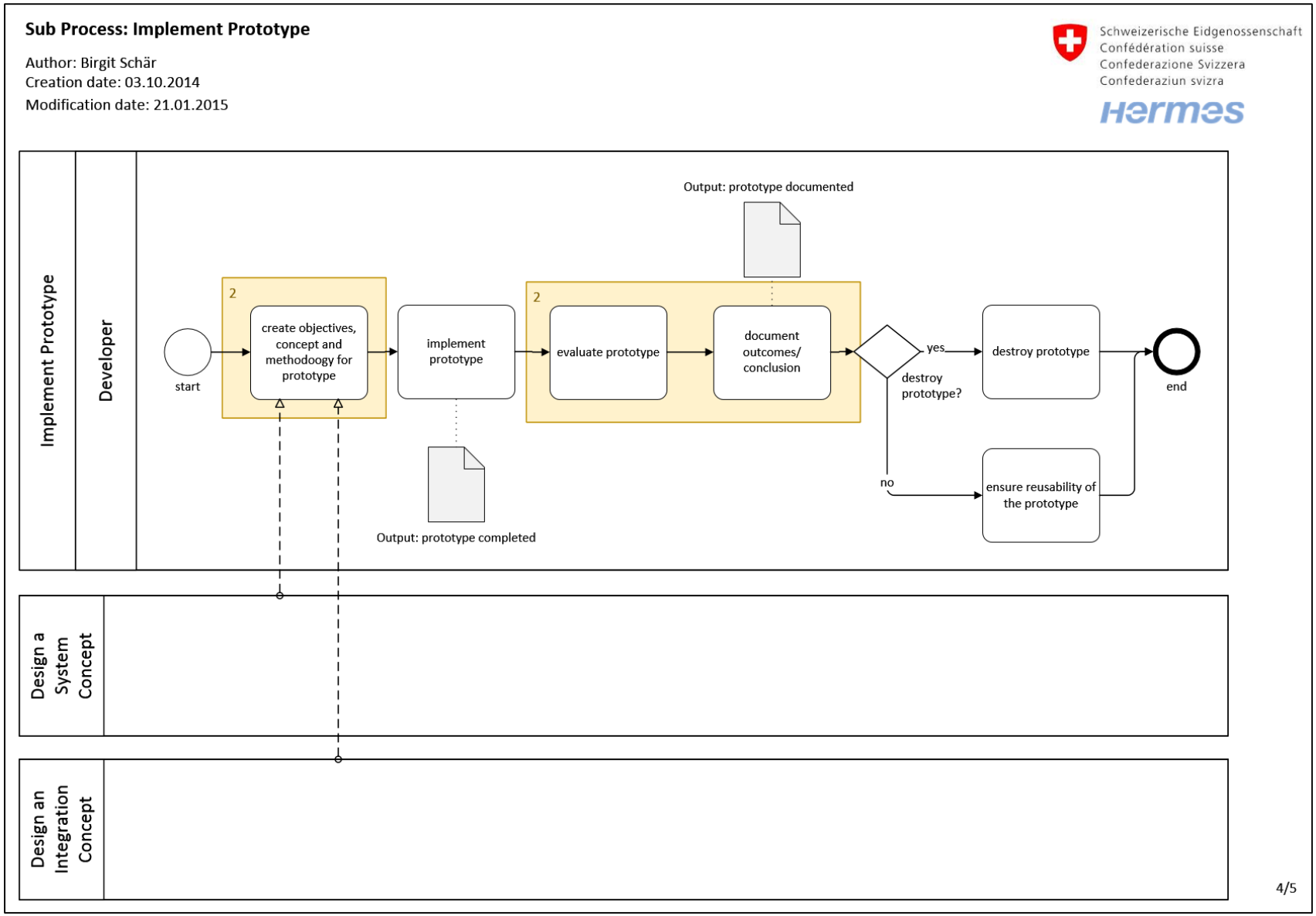
14.3.3.1 Design a System Concept



14.3.3.2 Design an Integration Concept



14.3.3.3 Implement Prototype



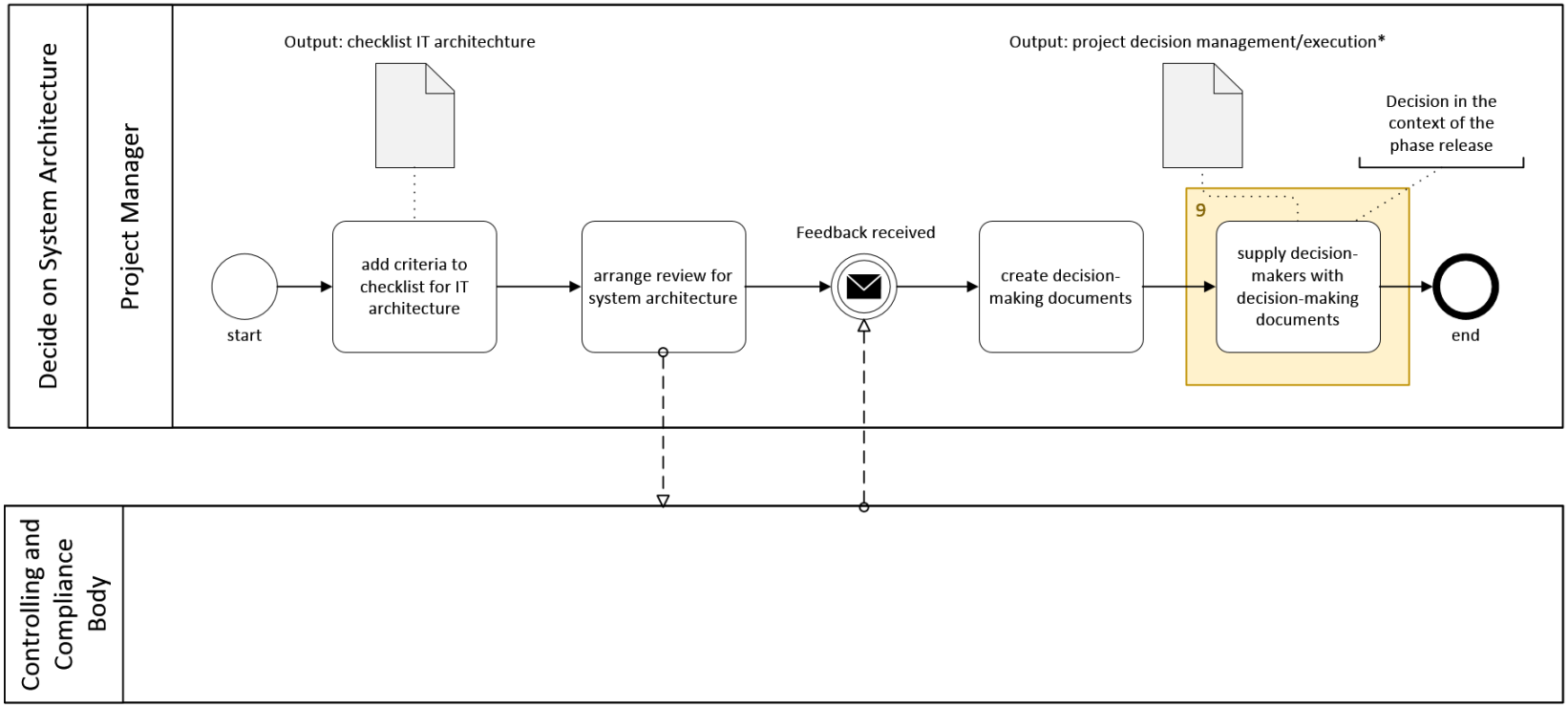
14.3.3.4 Decide on System Architecture

Sub Process: Decide on System Architecture

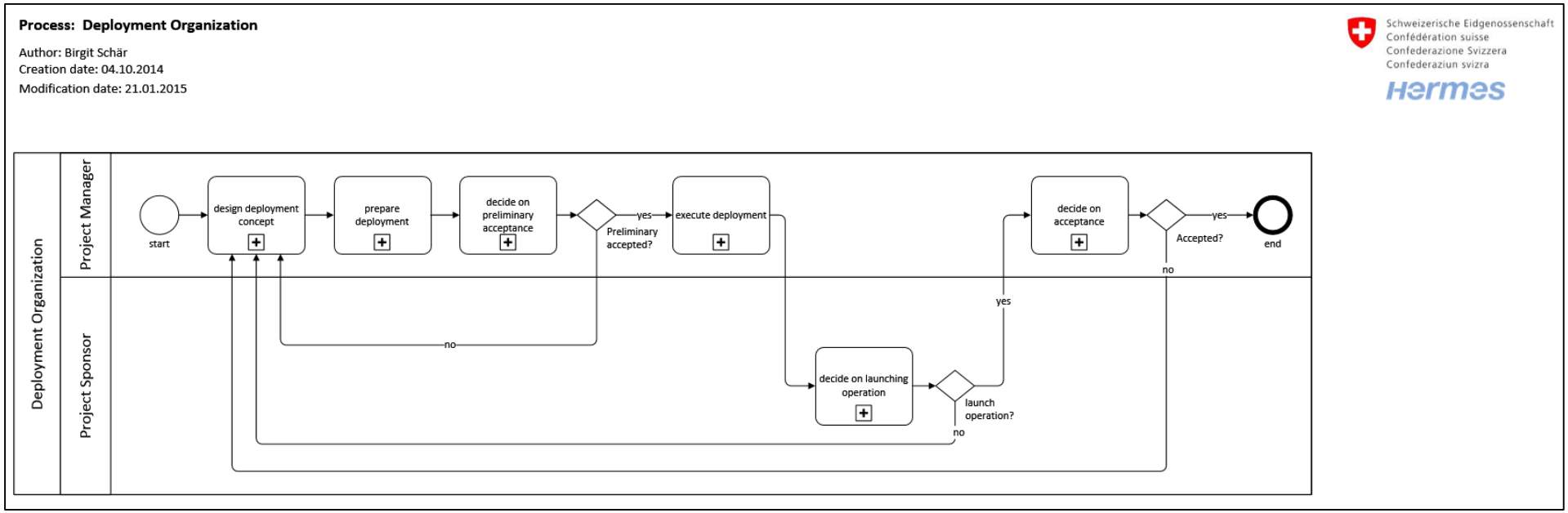
Author: Birgit Schär
Creation date: 03.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Hermes



14.3.4 Module “Deployment Organization”



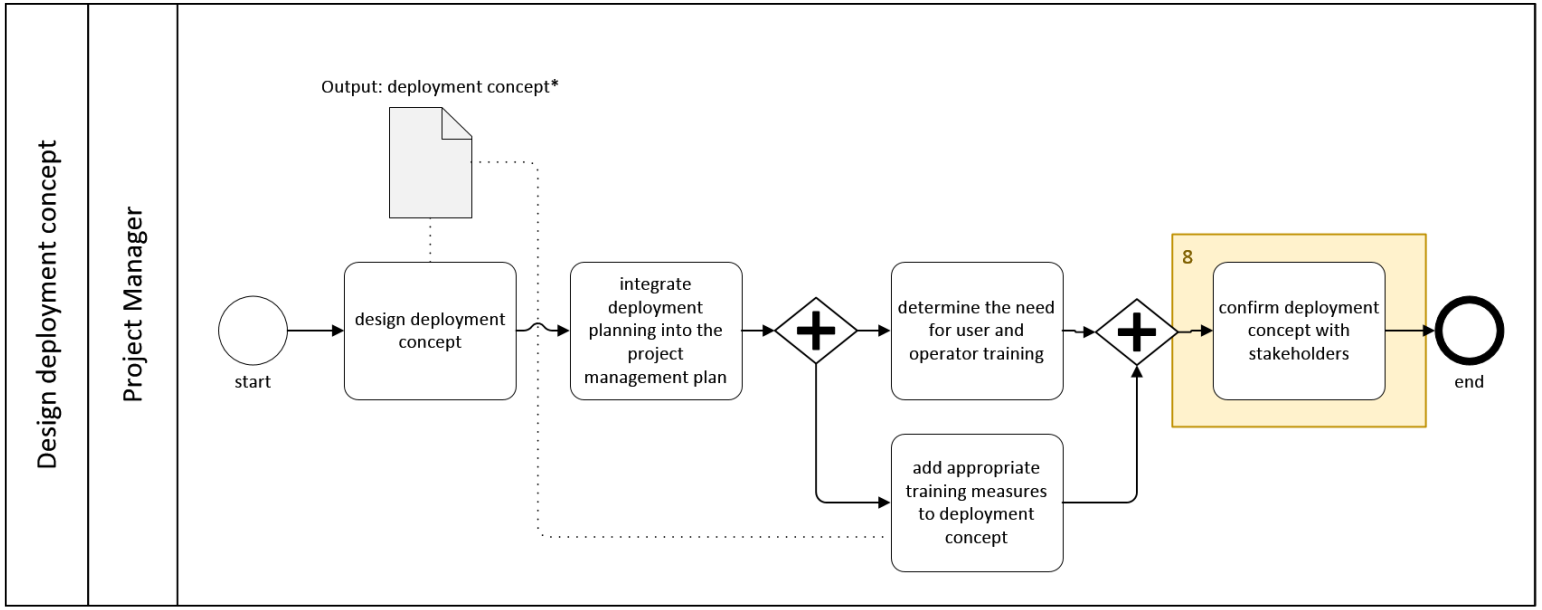
14.3.4.1 Design Deployment Concept

Sub Process: Design Deployment Concept

Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Hermes



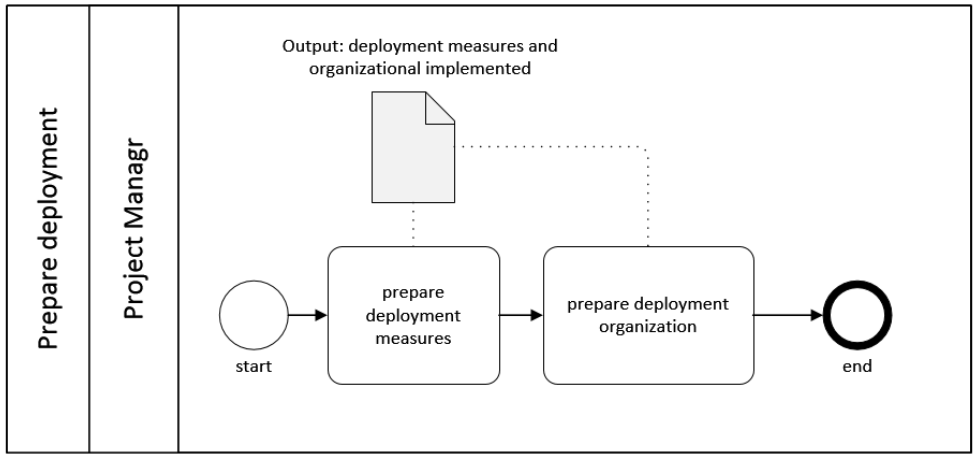
14.3.4.2 Prepare Deployment

Sub Process: Prepare Deployment

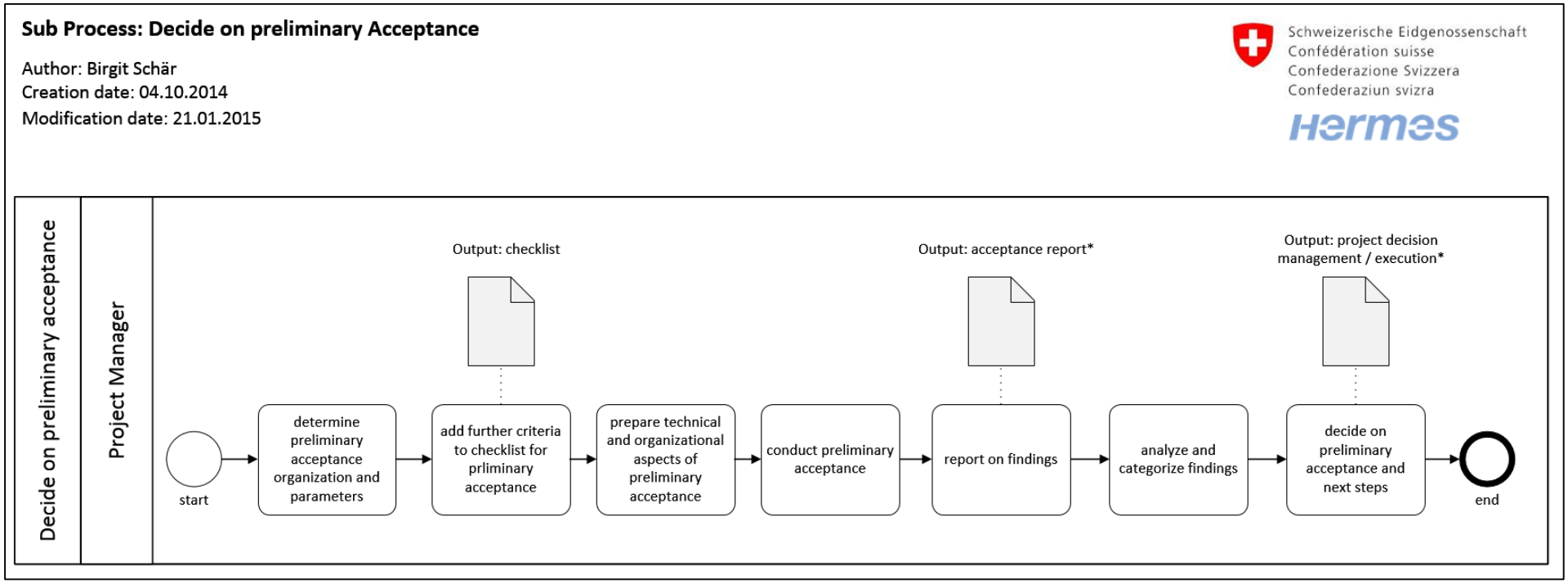
Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Hermes



14.3.4.3 Decide on preliminary Acceptance



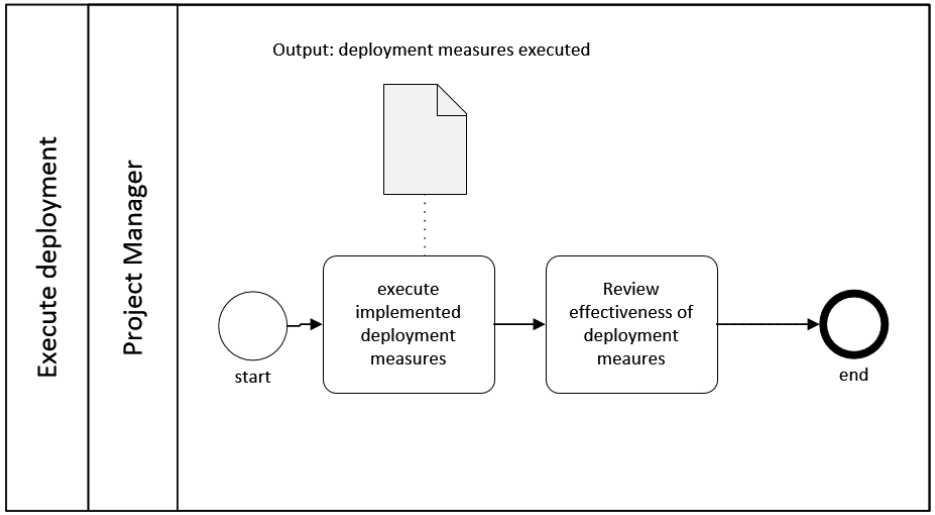
14.3.4.4 Execute Deployment

Sub Process: Execute Deployment

Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Hermes



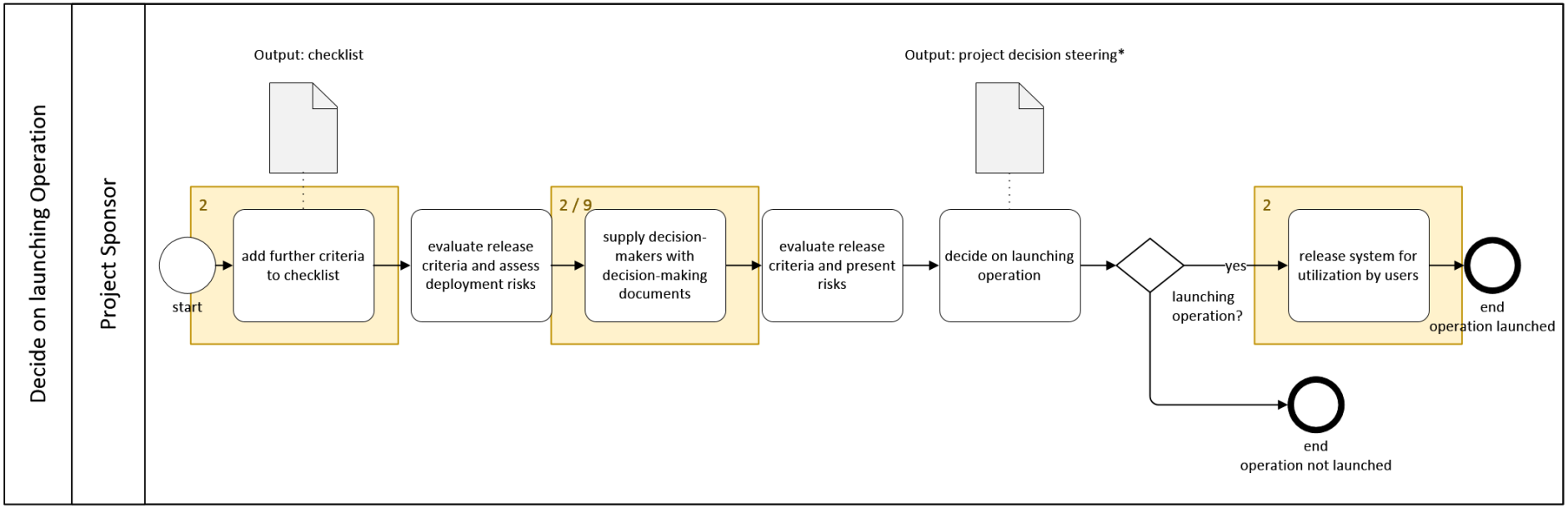
14.3.4.5 Decide on launching Operation

Sub Process: Decide on launching Operation

Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Hermes



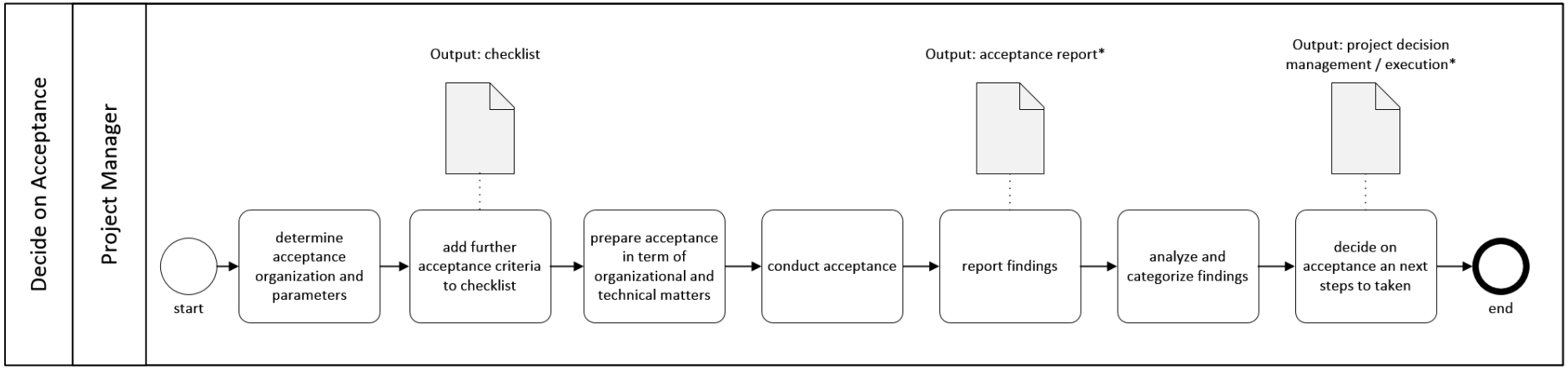
14.3.4.6 Decide on Acceptance

Sub Process: Decide on Acceptance

Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Hermes

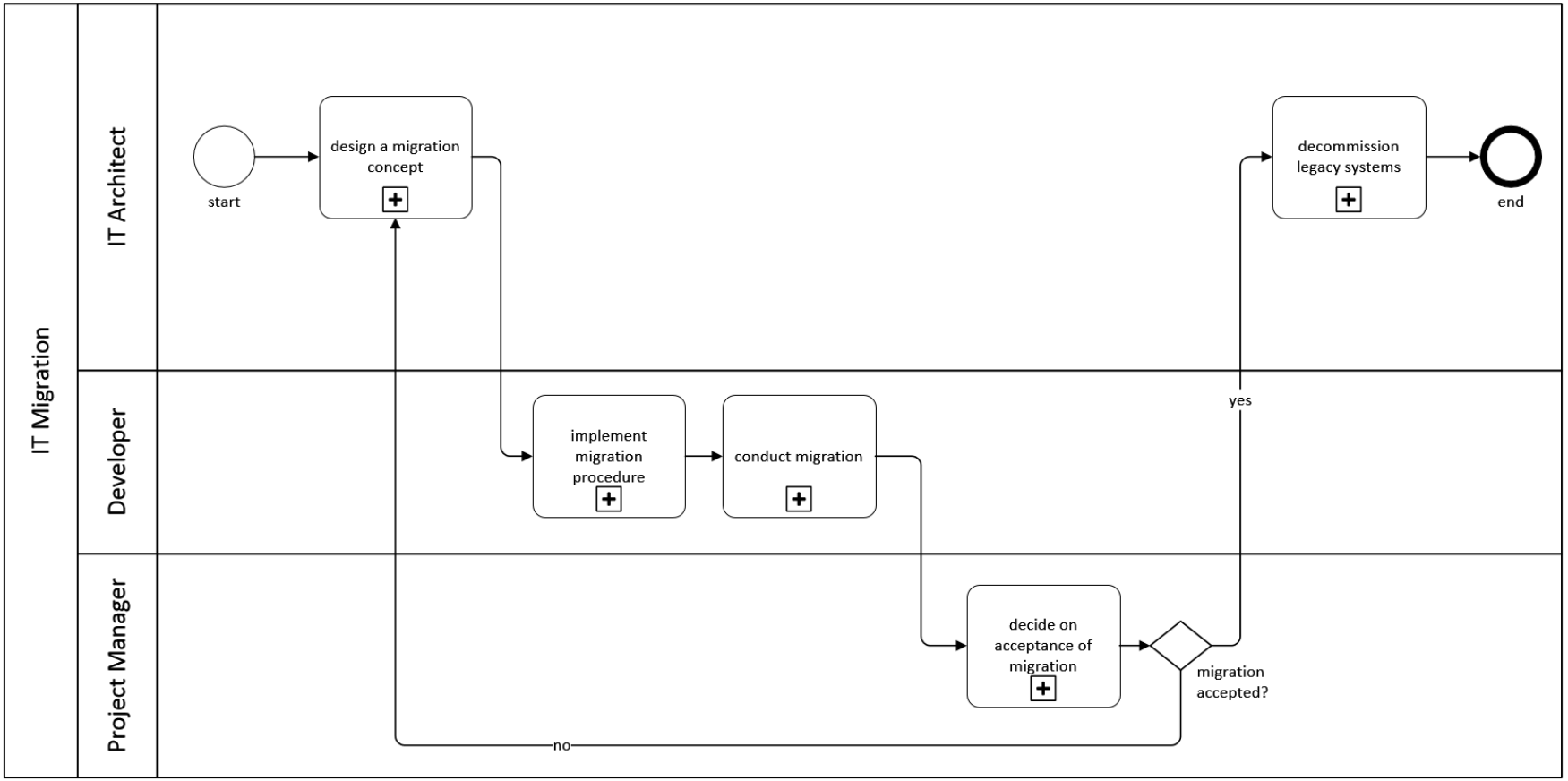


14.3.5 Module “IT Migration”

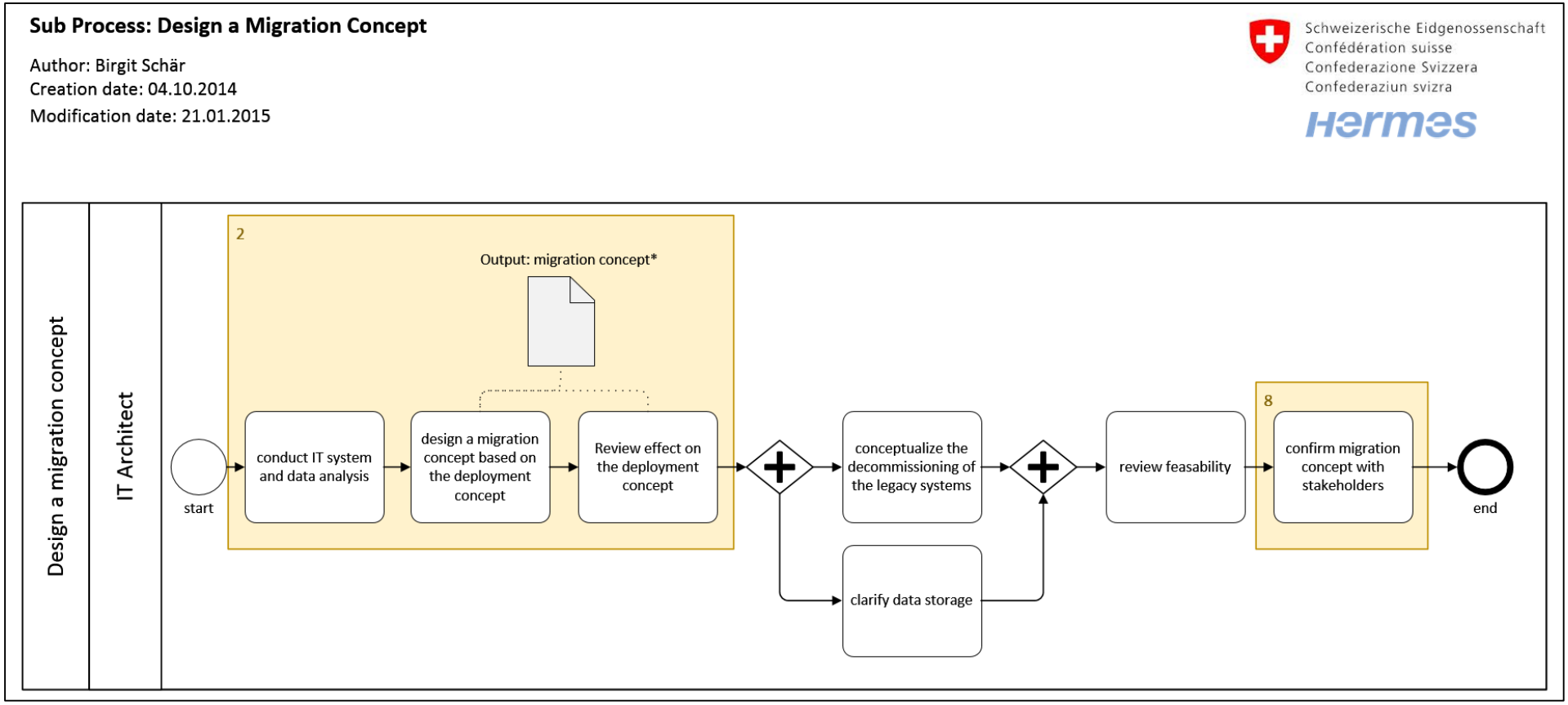
Process: Module IT Migration

Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Hermes



14.3.5.1 Design a Migration Concept



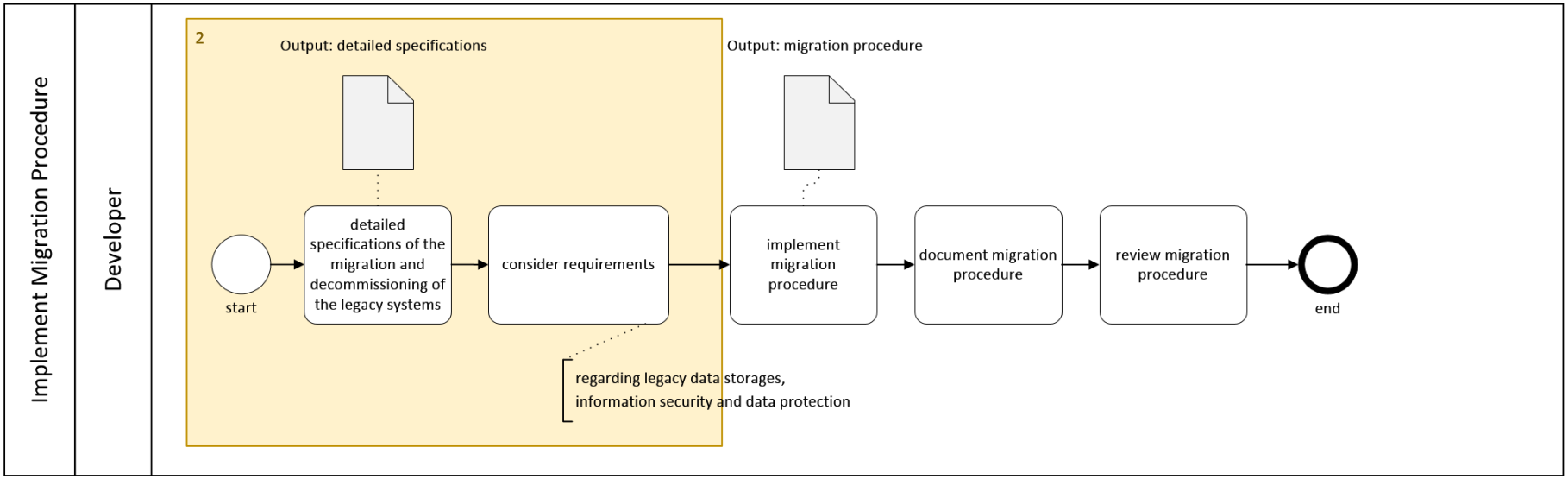
14.3.5.2 Implement Migration Procedure

Sub Process: Implement Migration Procedure

Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Hermes



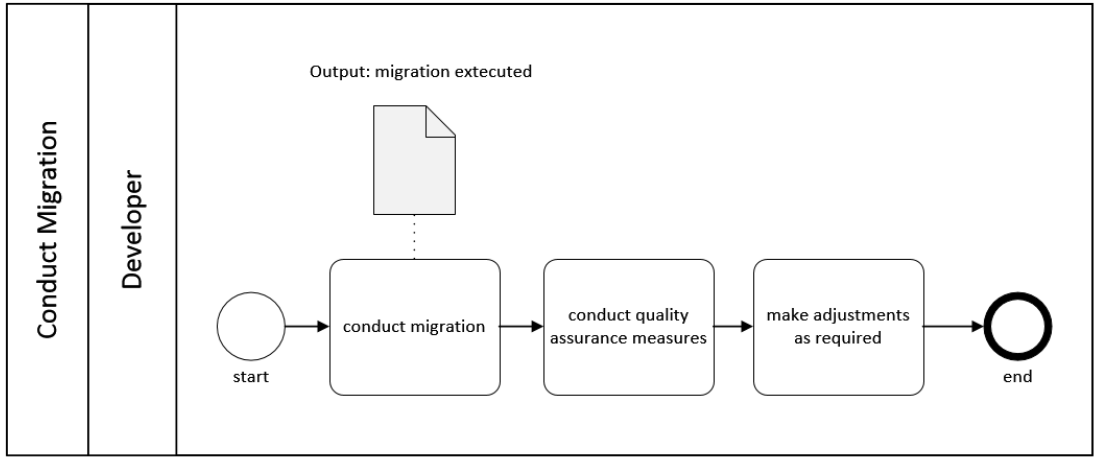
14.3.5.3 Conduct Migration

Sub Process: Conduct Migration

Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Hermes

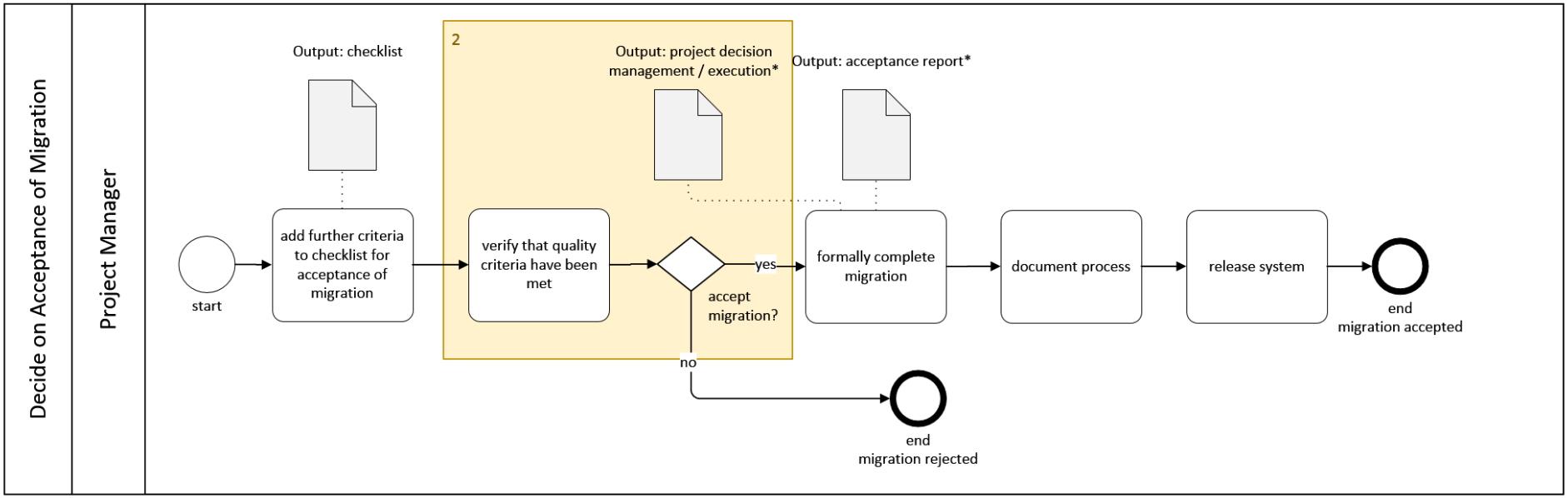


14.3.5.4 Decide on Acceptance of Migration

Sub Process: Decide on Acceptance of Migration

Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Hermes



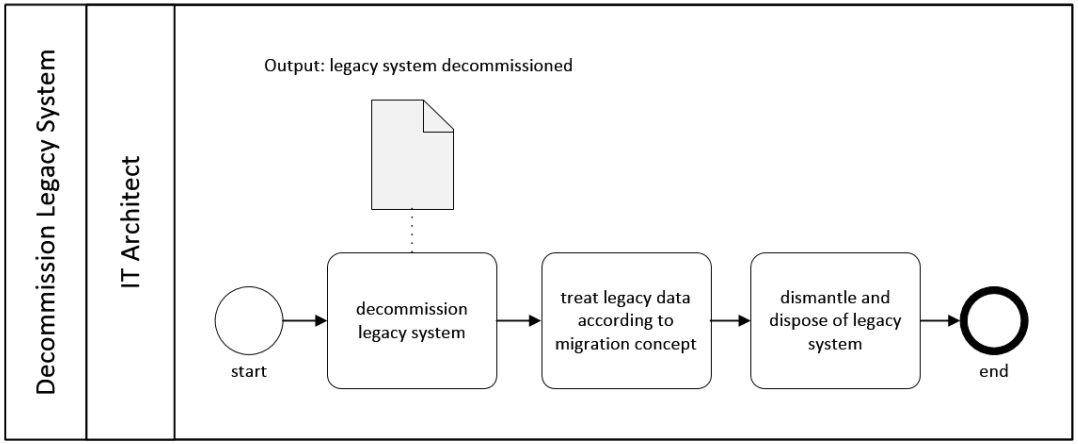
14.3.5.5 Decommission Legacy System

Sub Process: Decommission Legacy System

Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Hermes



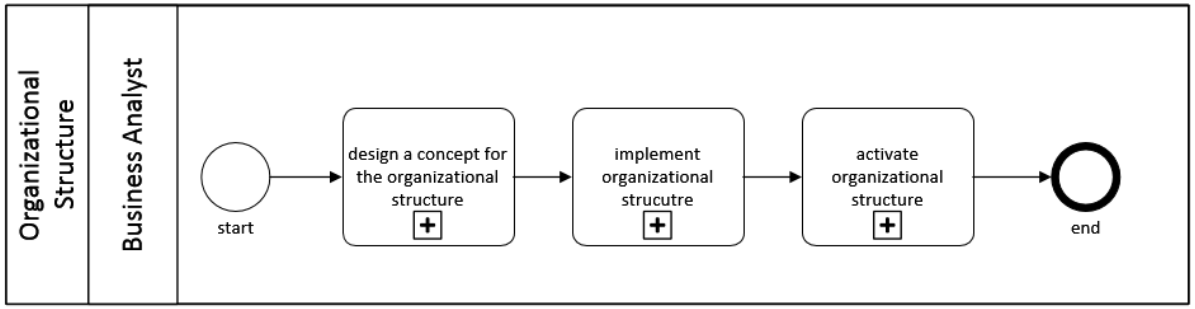
14.3.6 Module “Organizational Structure”

Process: Module Organizational Structure

Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Hermes



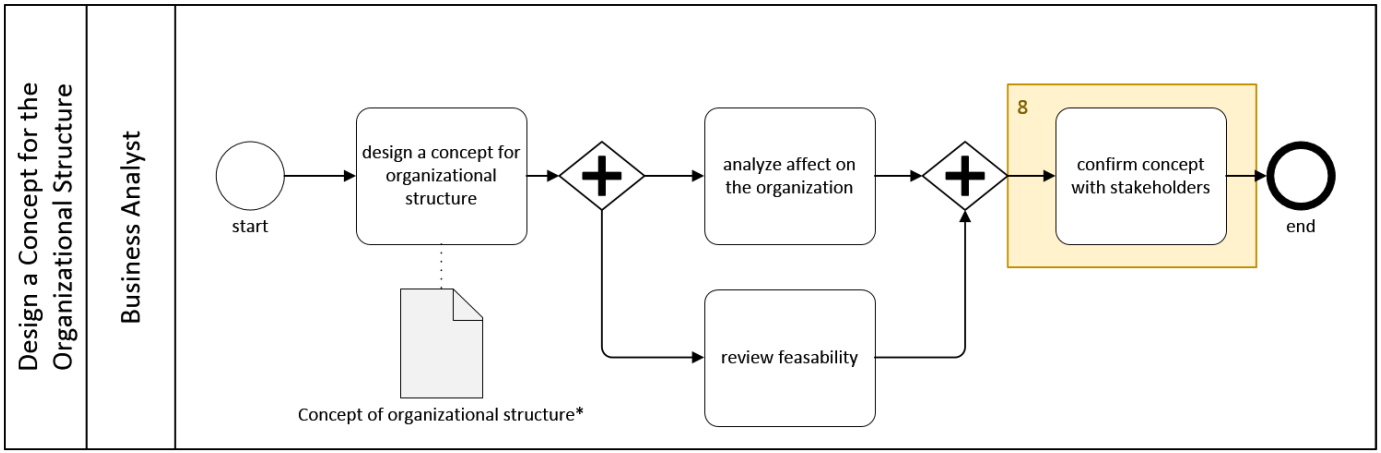
14.3.6.1 Design a Concept for the Organizational Structure

Sub Process: Design a Concept for Organizational Strucutre

Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Hermes

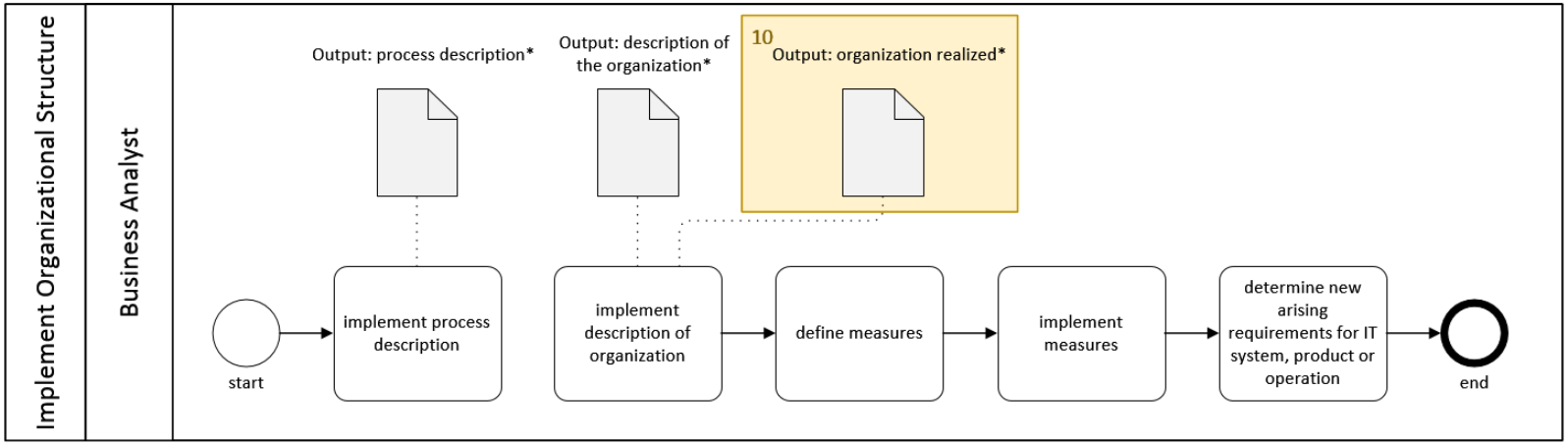


14.3.6.2 Implement Organizational Structure

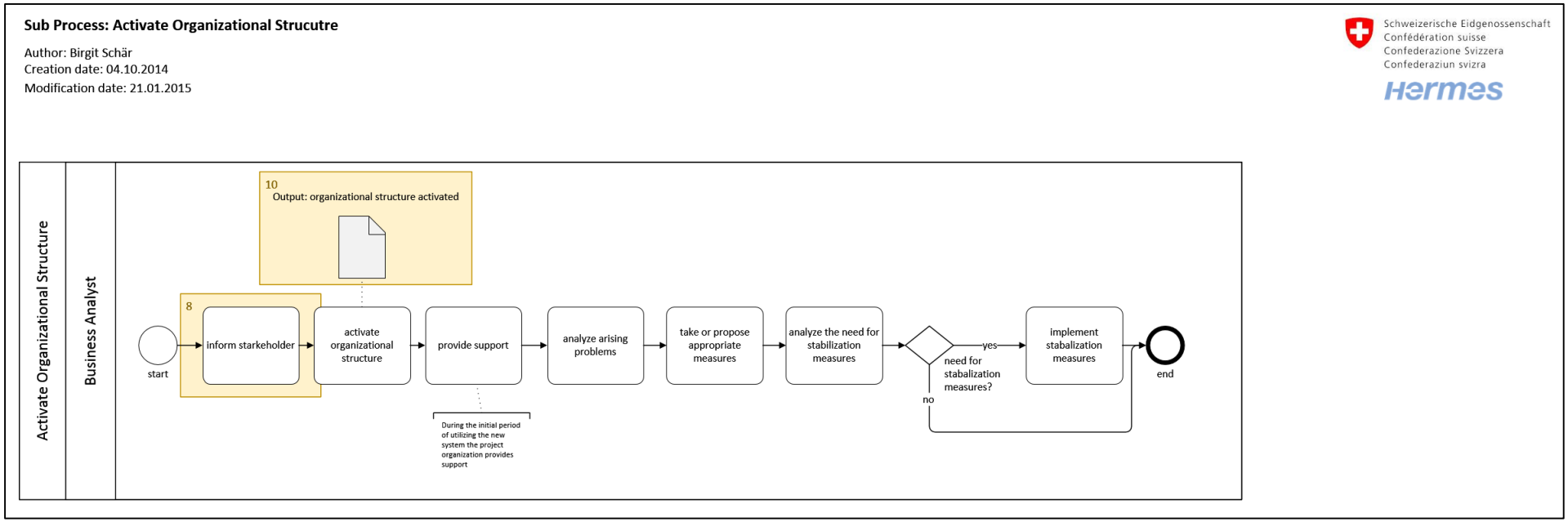
Sub Process: Implement Organizational Structure

Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

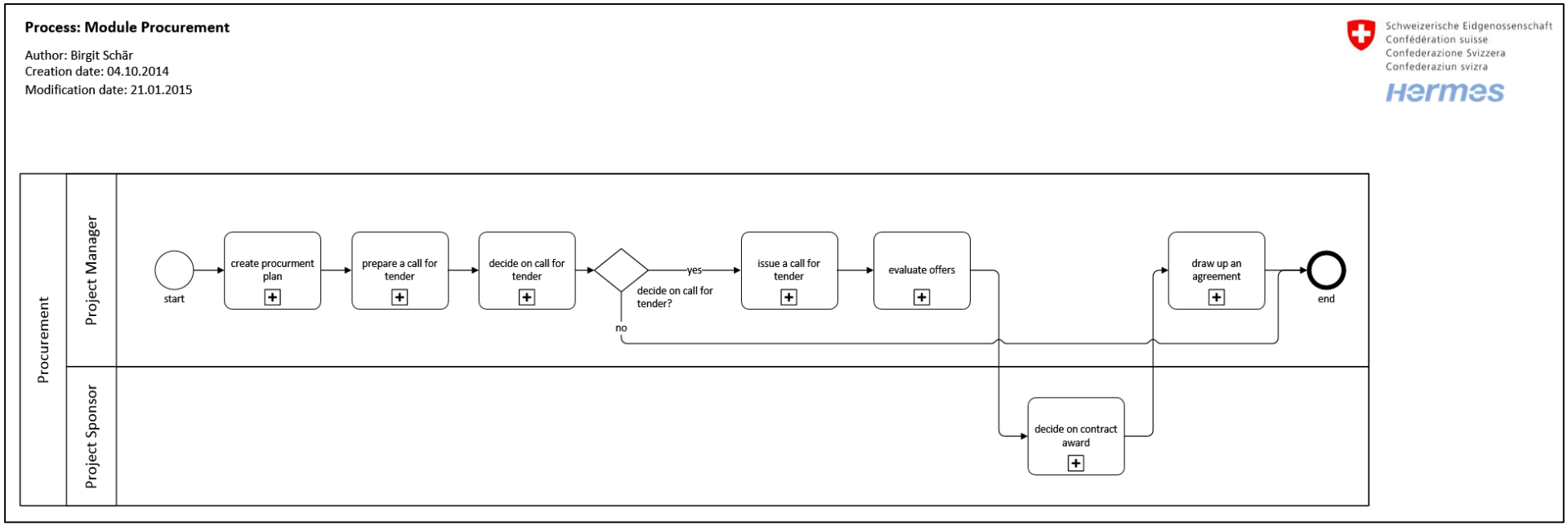
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Hermes



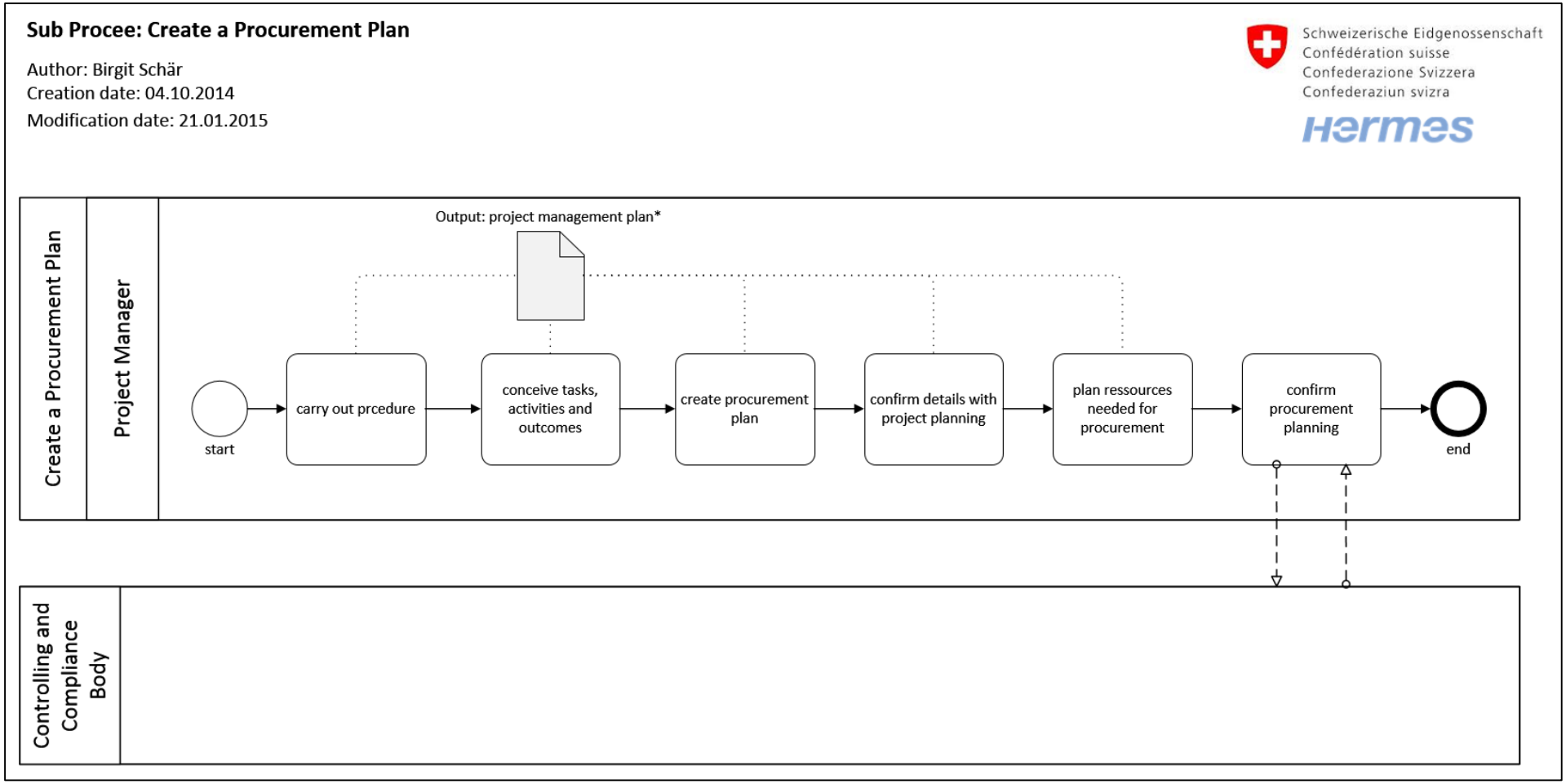
14.3.6.3 Activate Organizational Structure



14.3.7 Module “Procurement”



14.3.7.1 Create a Procurement Plan



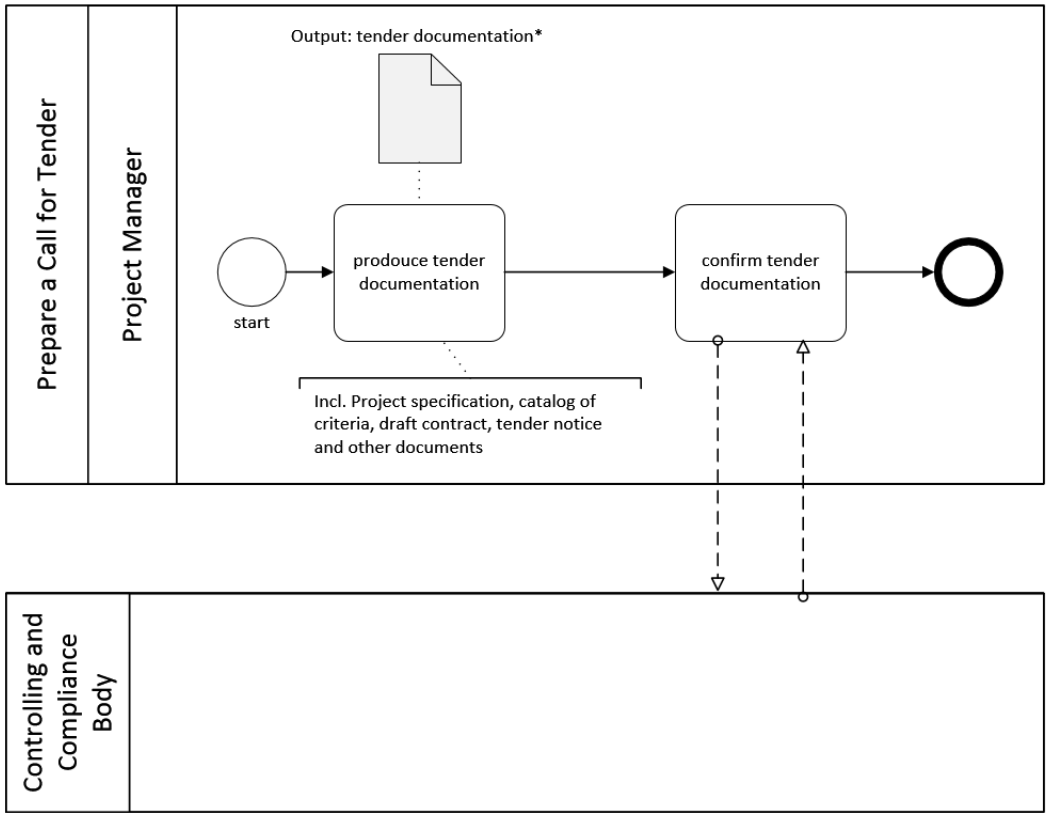
14.3.7.2 Prepare a Call for Tender

Sub Process: Prepare a Call for Tender

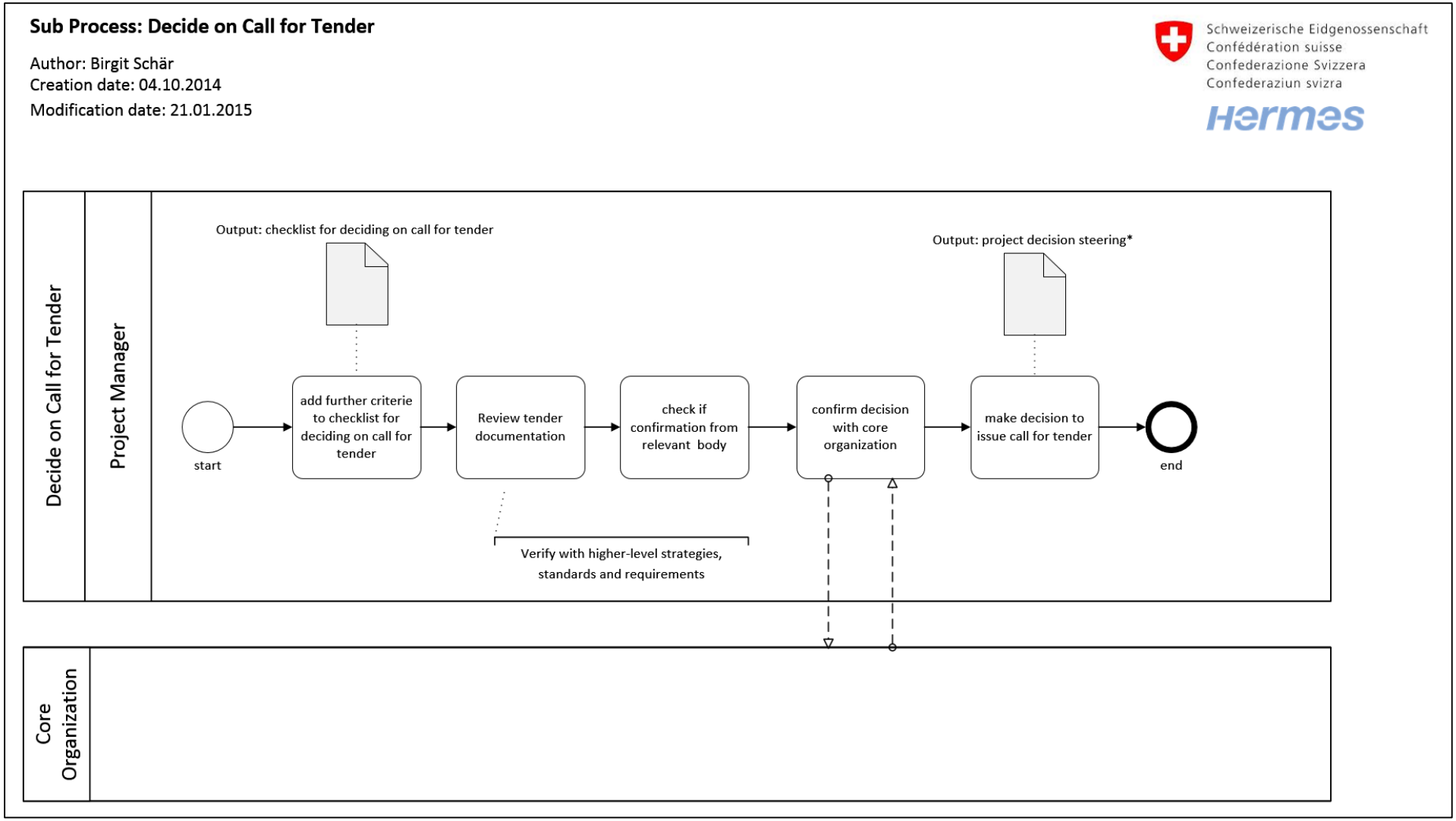
Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Hermes



14.3.7.3 Decide on Call for Tender



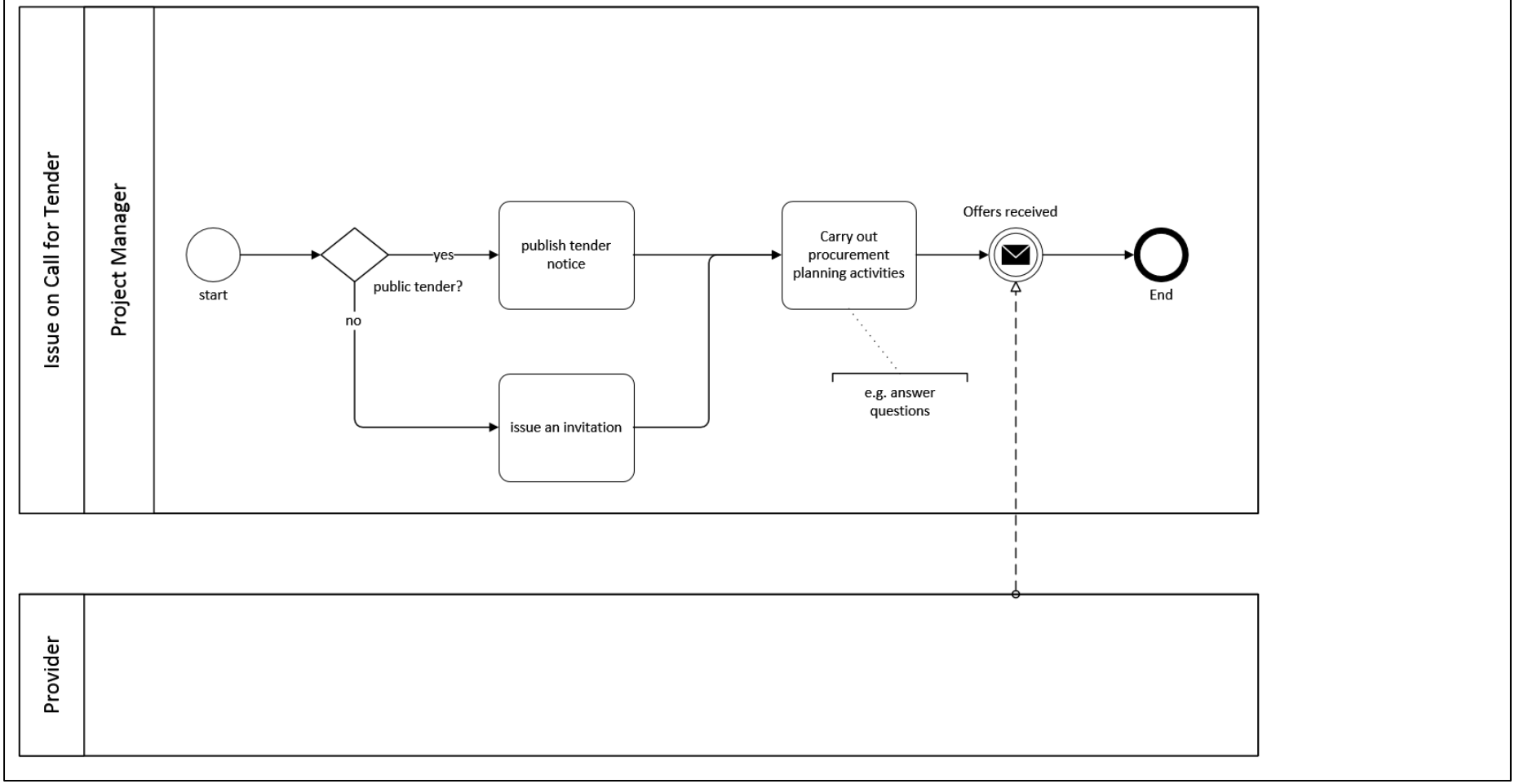
14.3.7.4 Issue a Call for Tender

Sub Process: Issue a Call for Tender

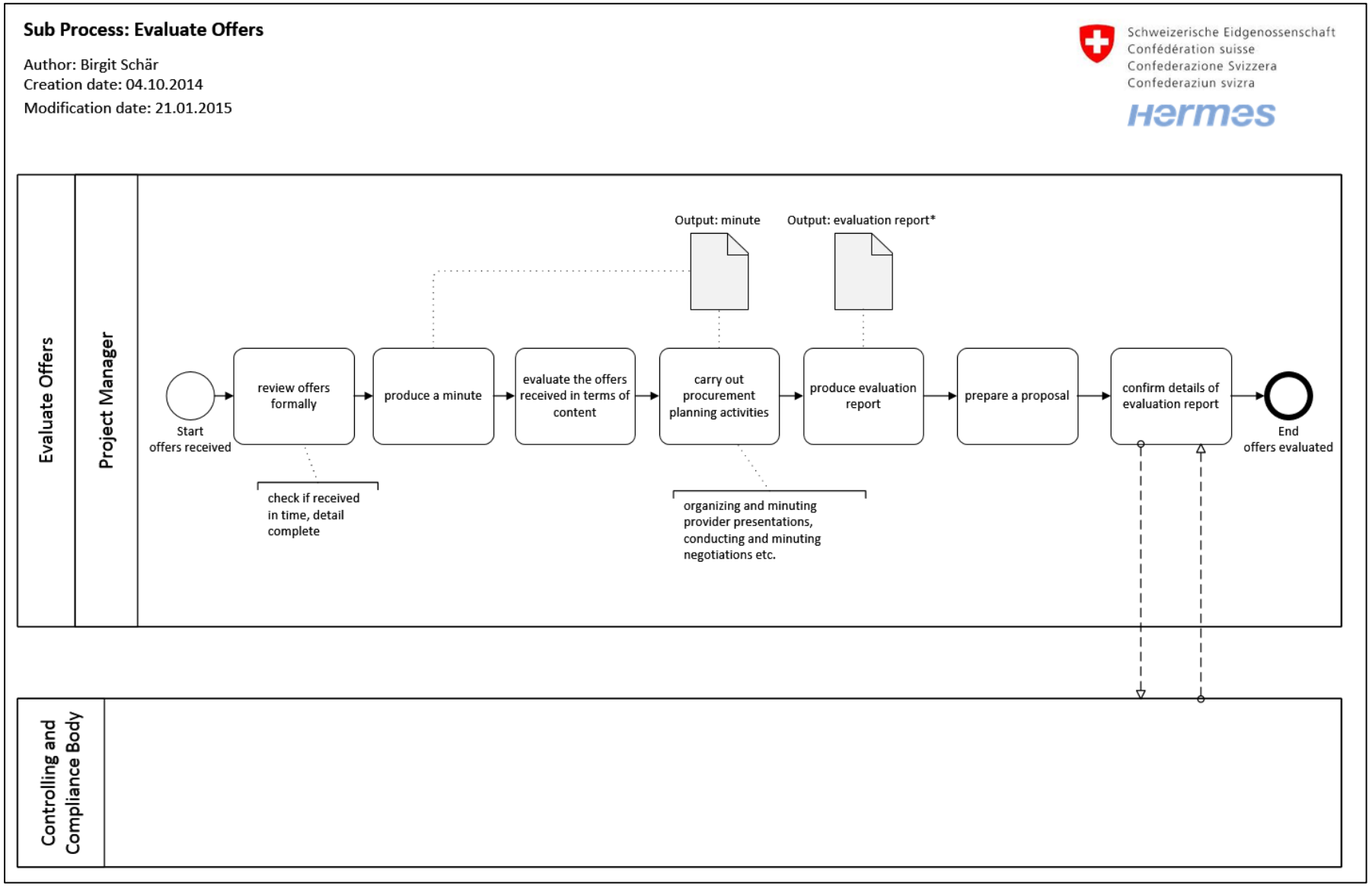
Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

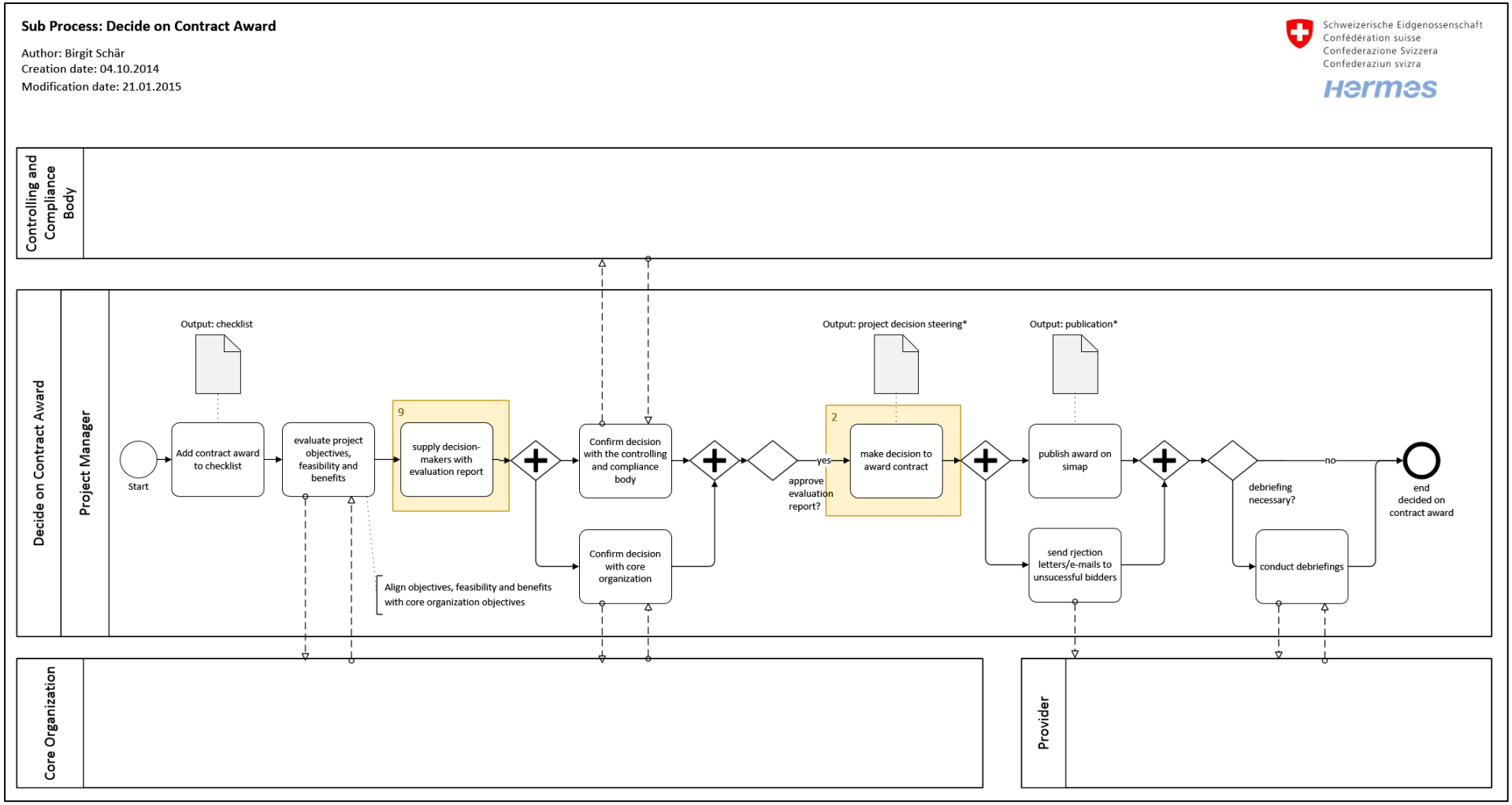
Hermes



14.3.7.5 Evaluate Offers



14.3.7.6 Decide on Contract Award

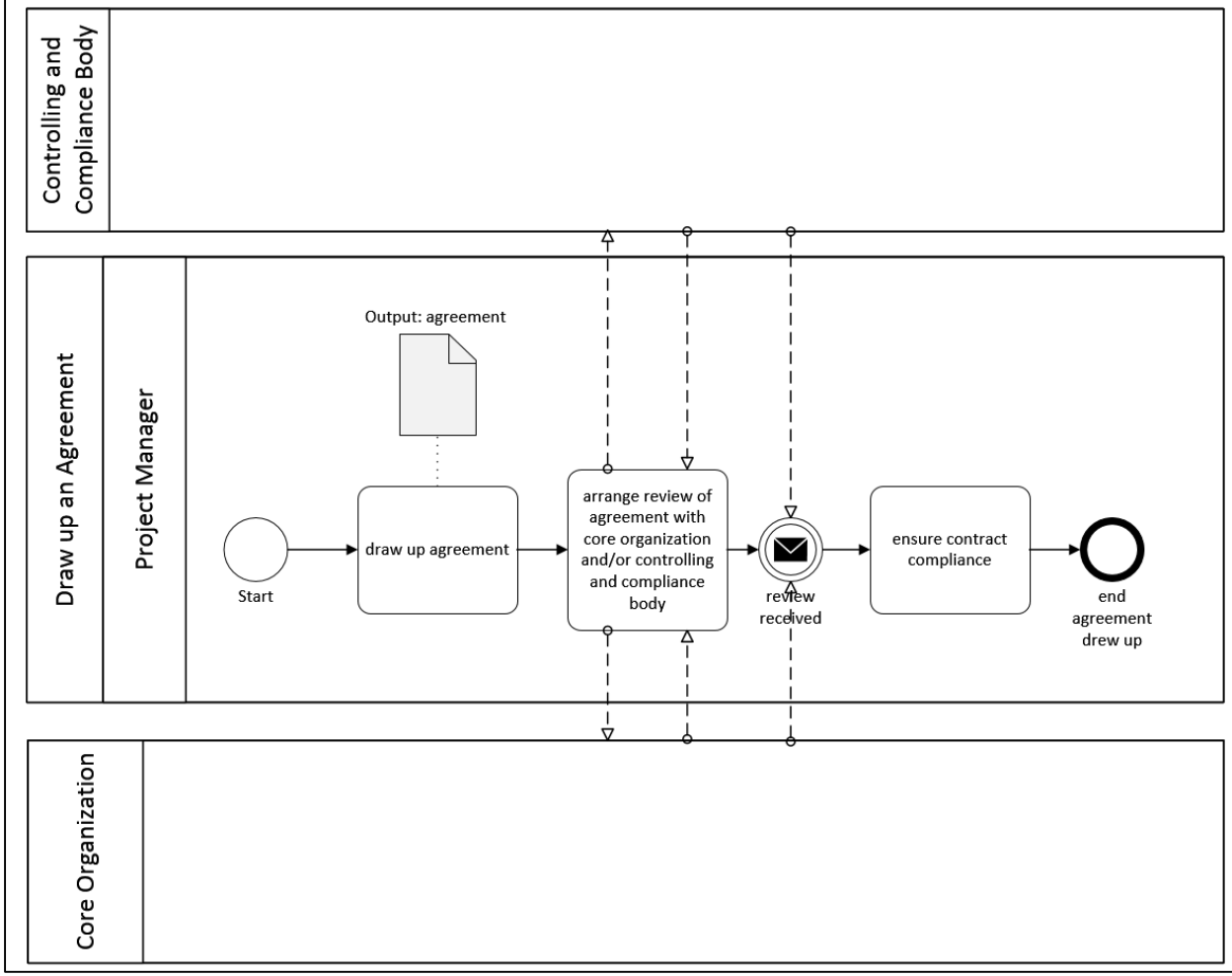


14.3.7.7 Draw up an Agreement

Sub Process: Draw up an Agreement

Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Hermes



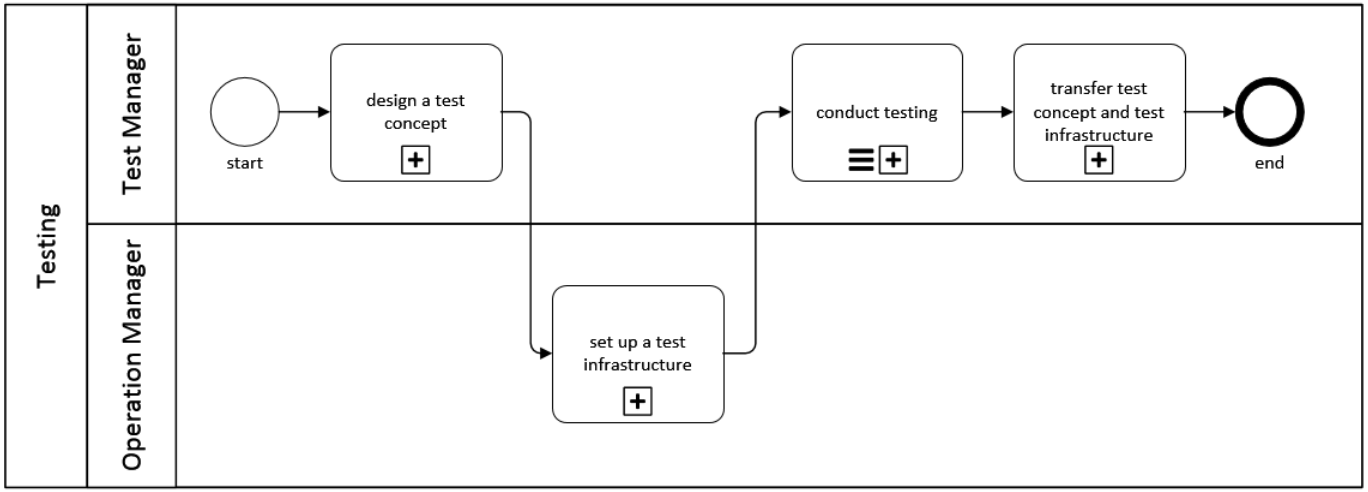
14.3.8 Module “Testing”

Process: Module Testing

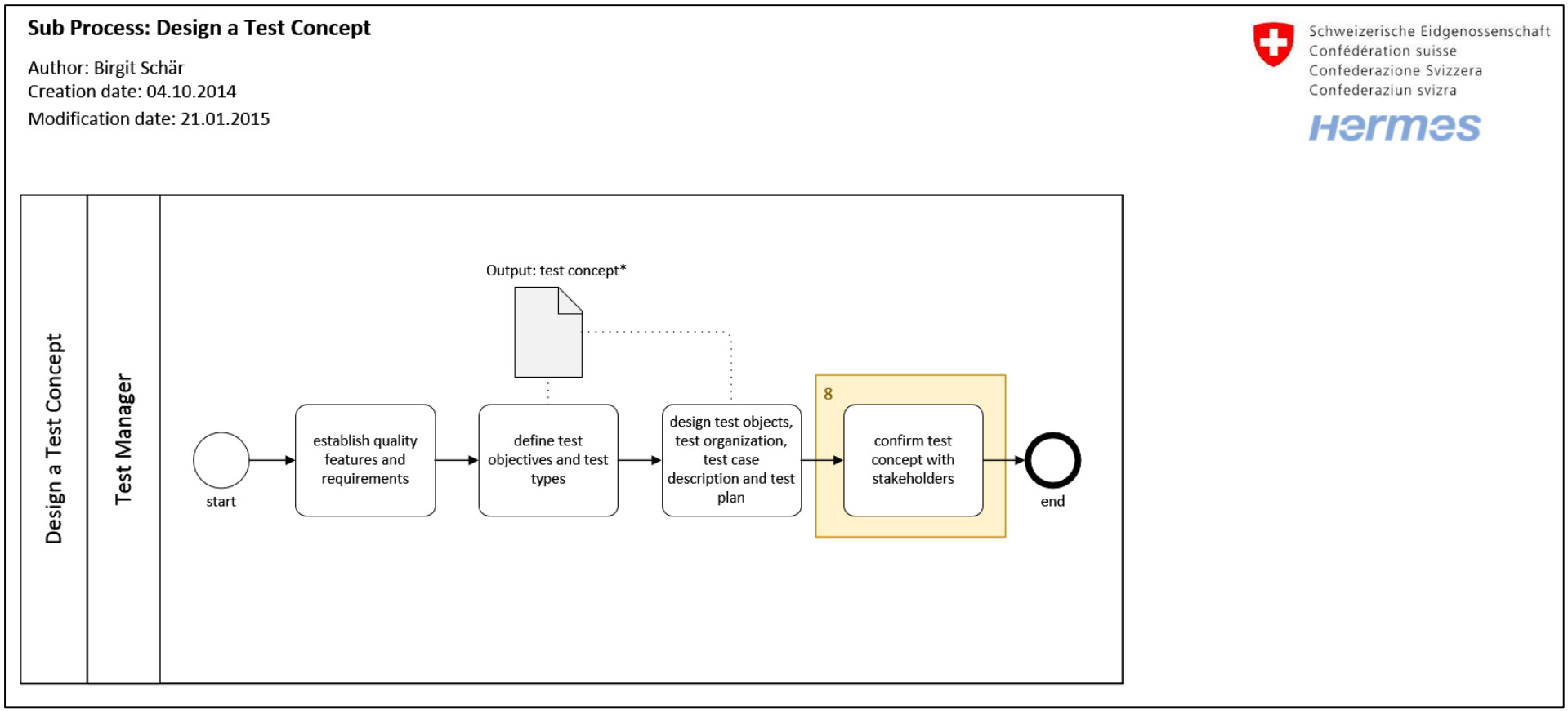
Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Hermes



14.3.8.1 Design a Test Concept

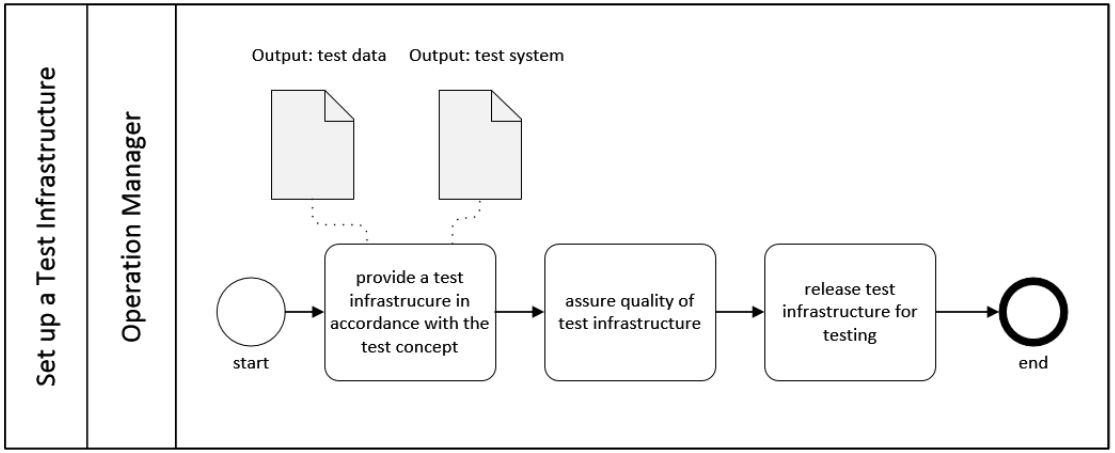


14.3.8.2 Set up a Test Infrastructure

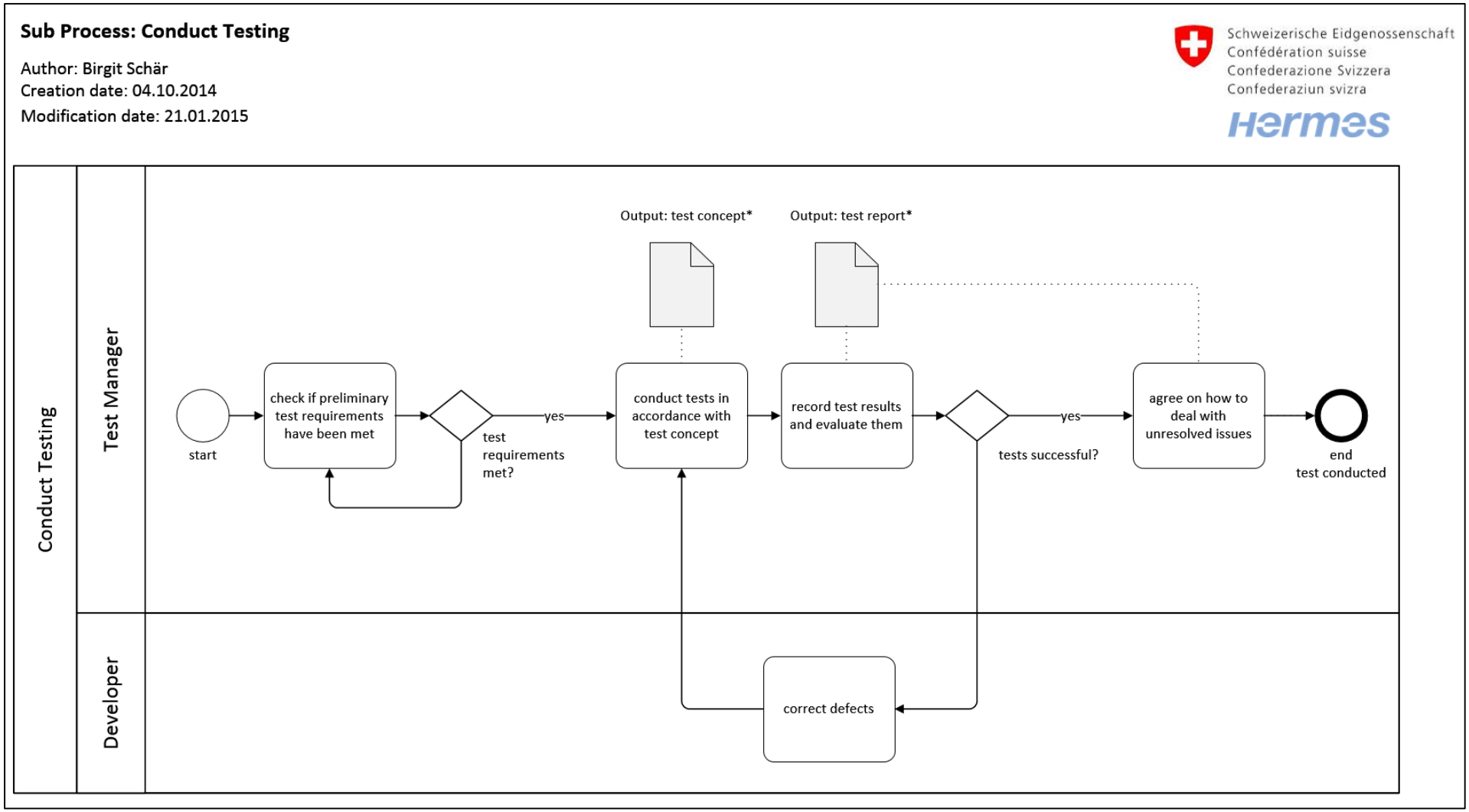
Sub Process: Set up a Test Infrastructure

Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Hermes



14.3.8.3 Conduct Testing



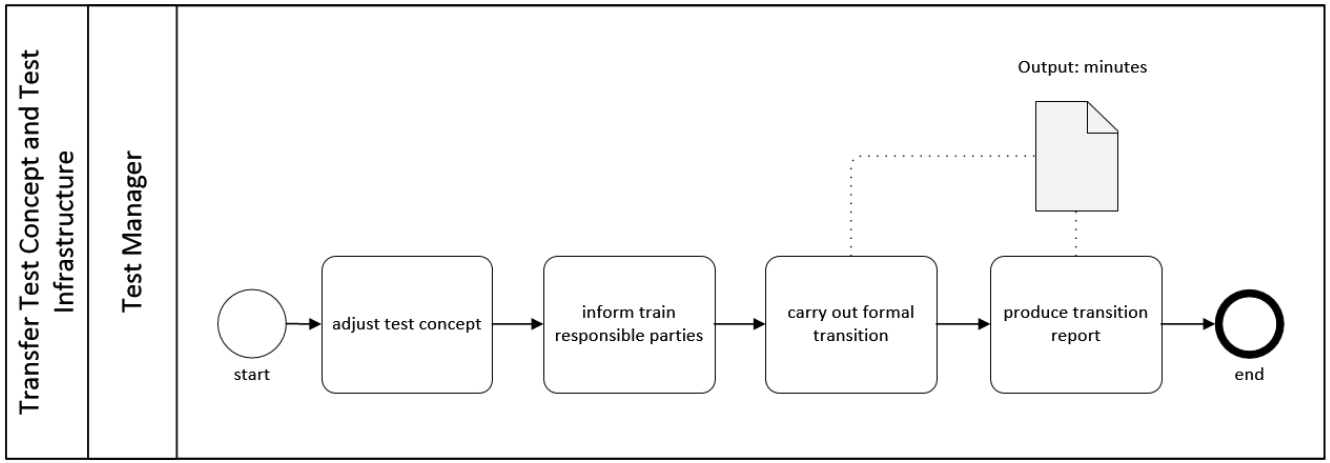
14.3.8.4 Transfer Test Concept and Test Infrastructure

Sub Process: Transfer Test Concept and Test Infrastructure

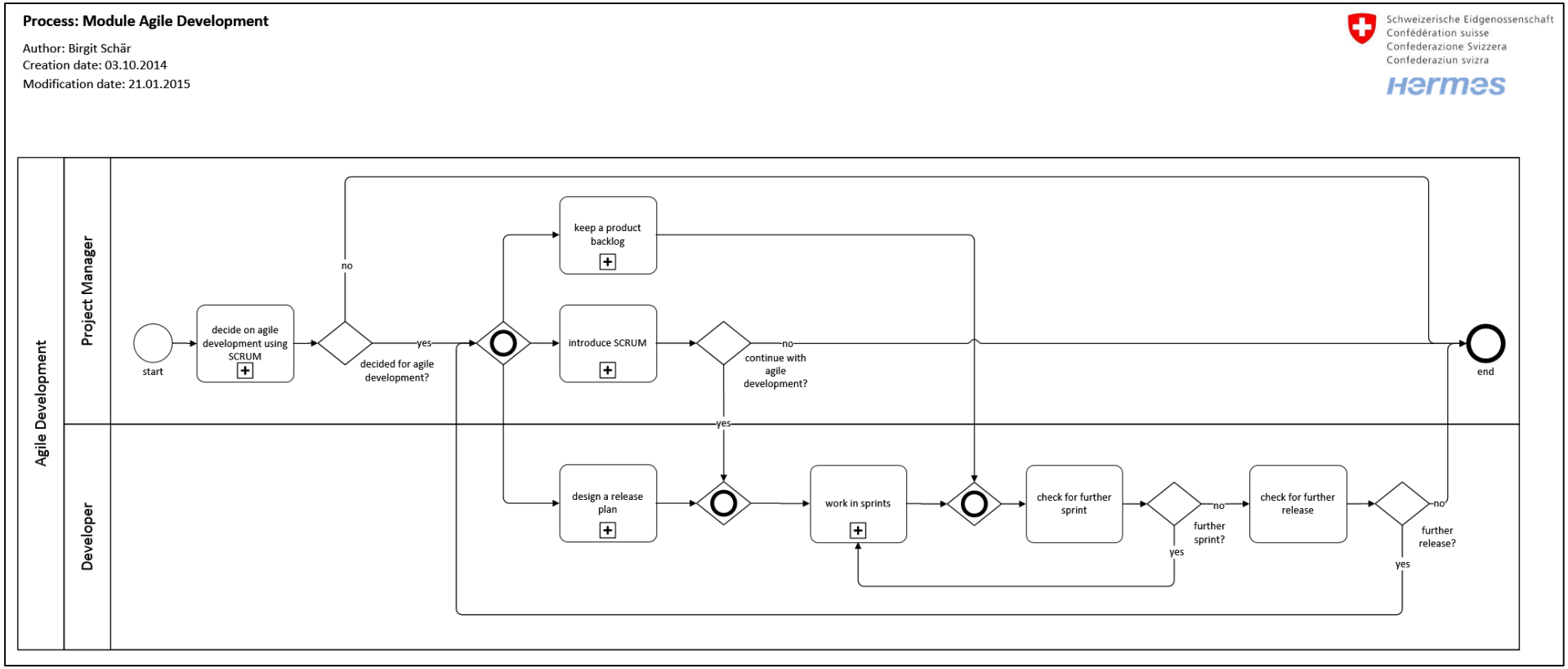
Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

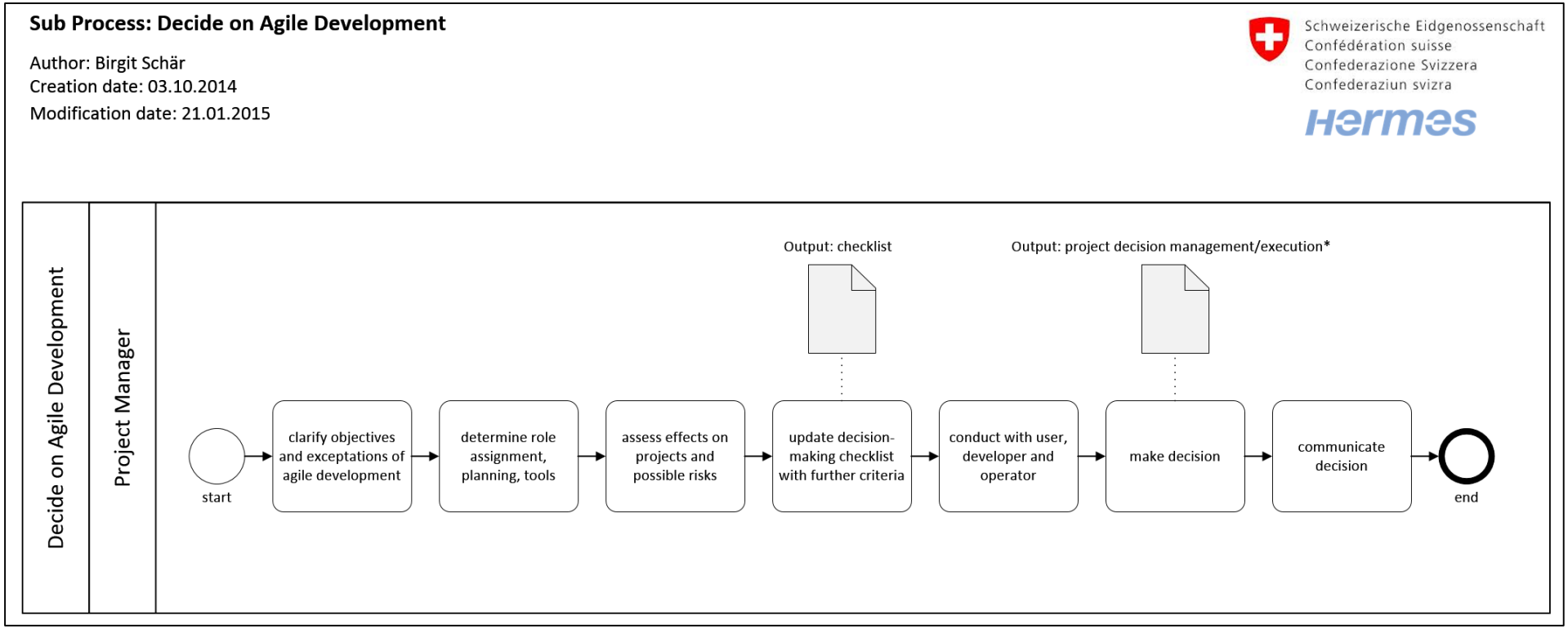
Hermes



14.3.9 Module “Agile Development”



14.3.9.1 Decide on Agile Development

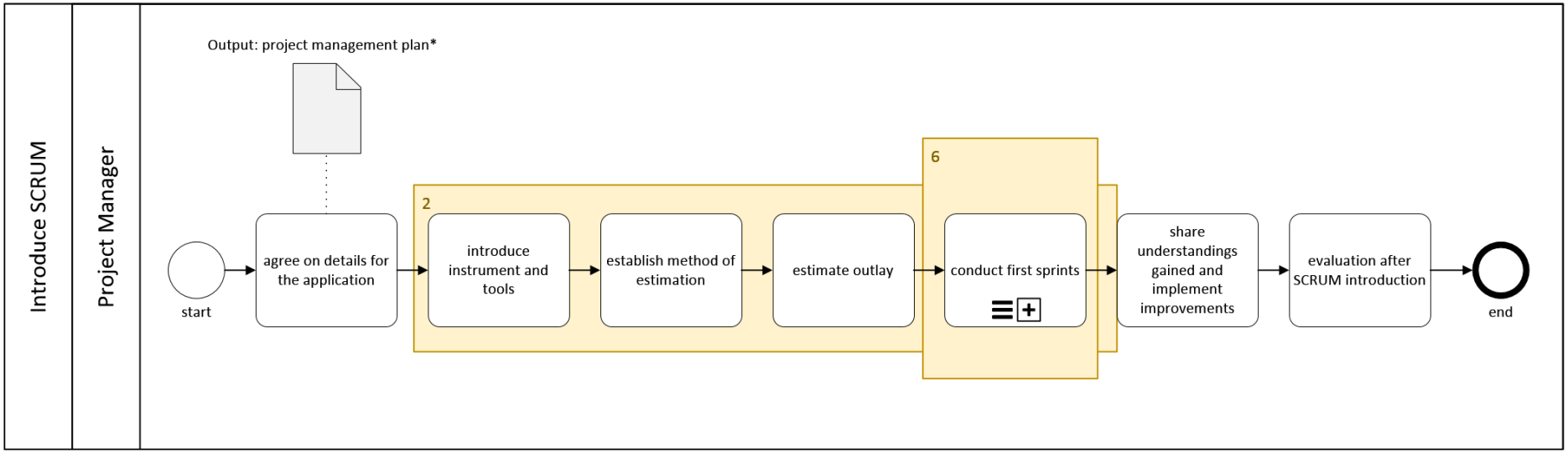


14.3.9.2 Introduce SCRUM

Sub Process: Introduce SCRUM

Author: Birgit Schär
Creation date: 03.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Hermes



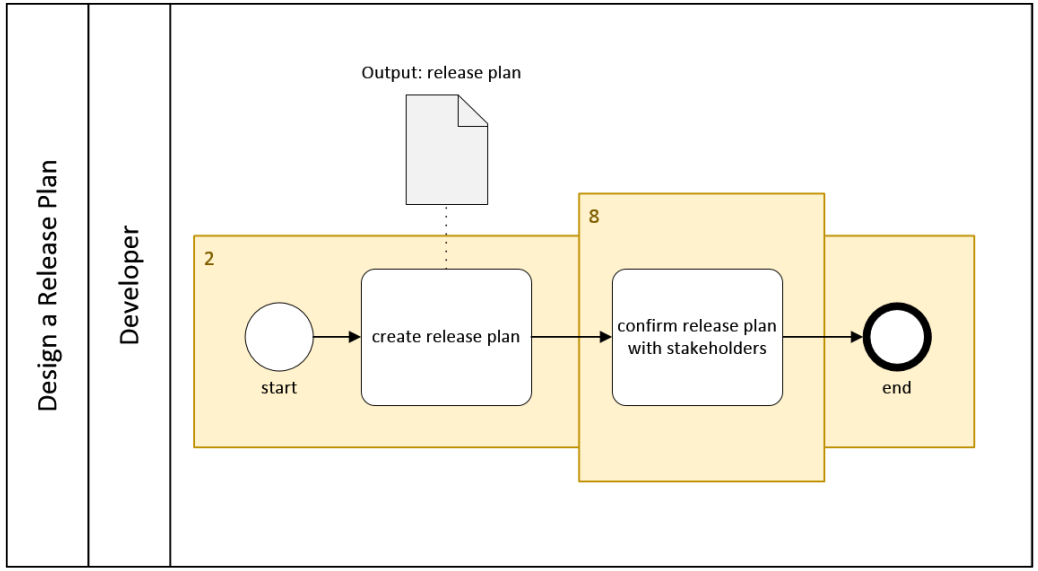
14.3.9.3 Design a Release Plan

Sub Process: Design a Release Plan

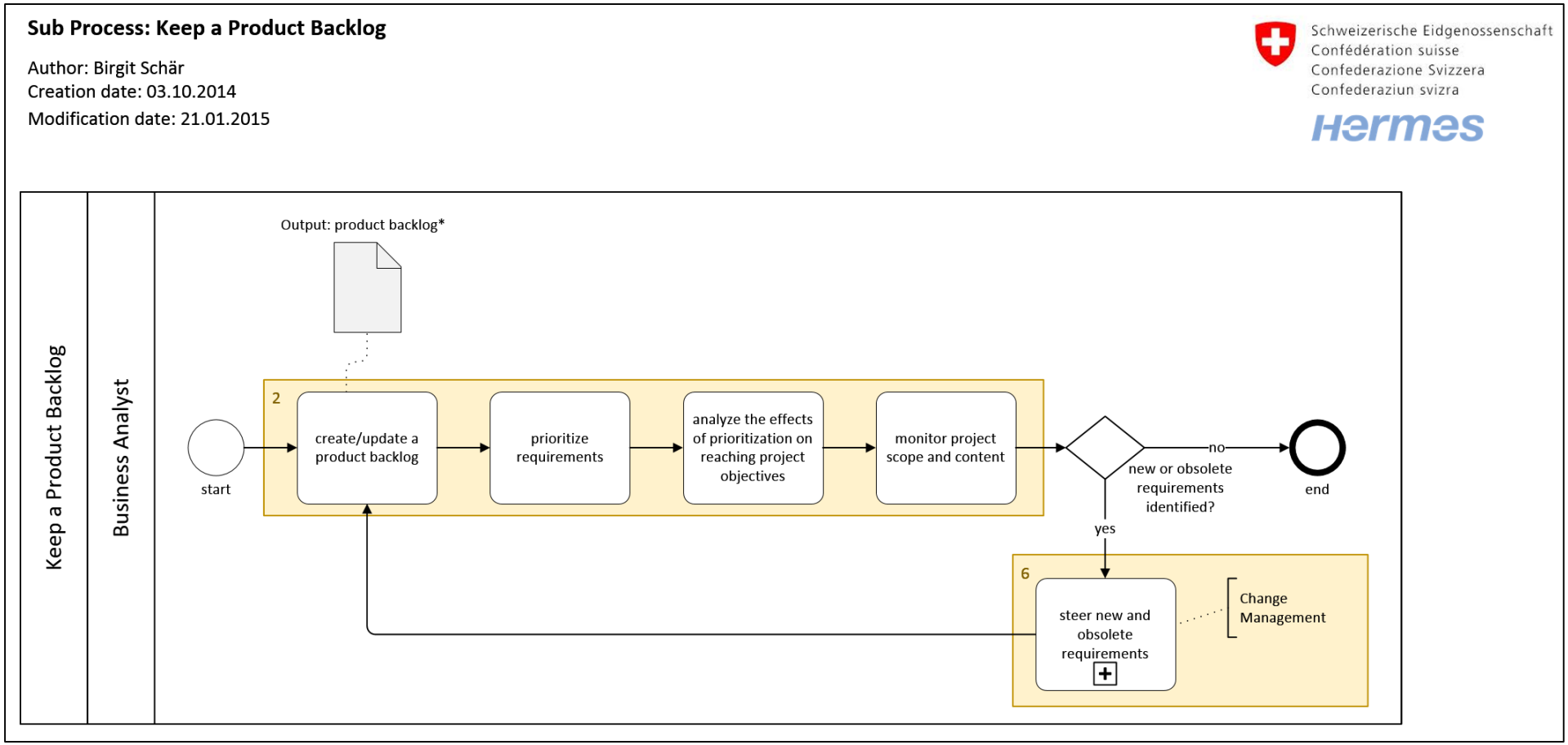
Author: Birgit Schär
Creation date: 03.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

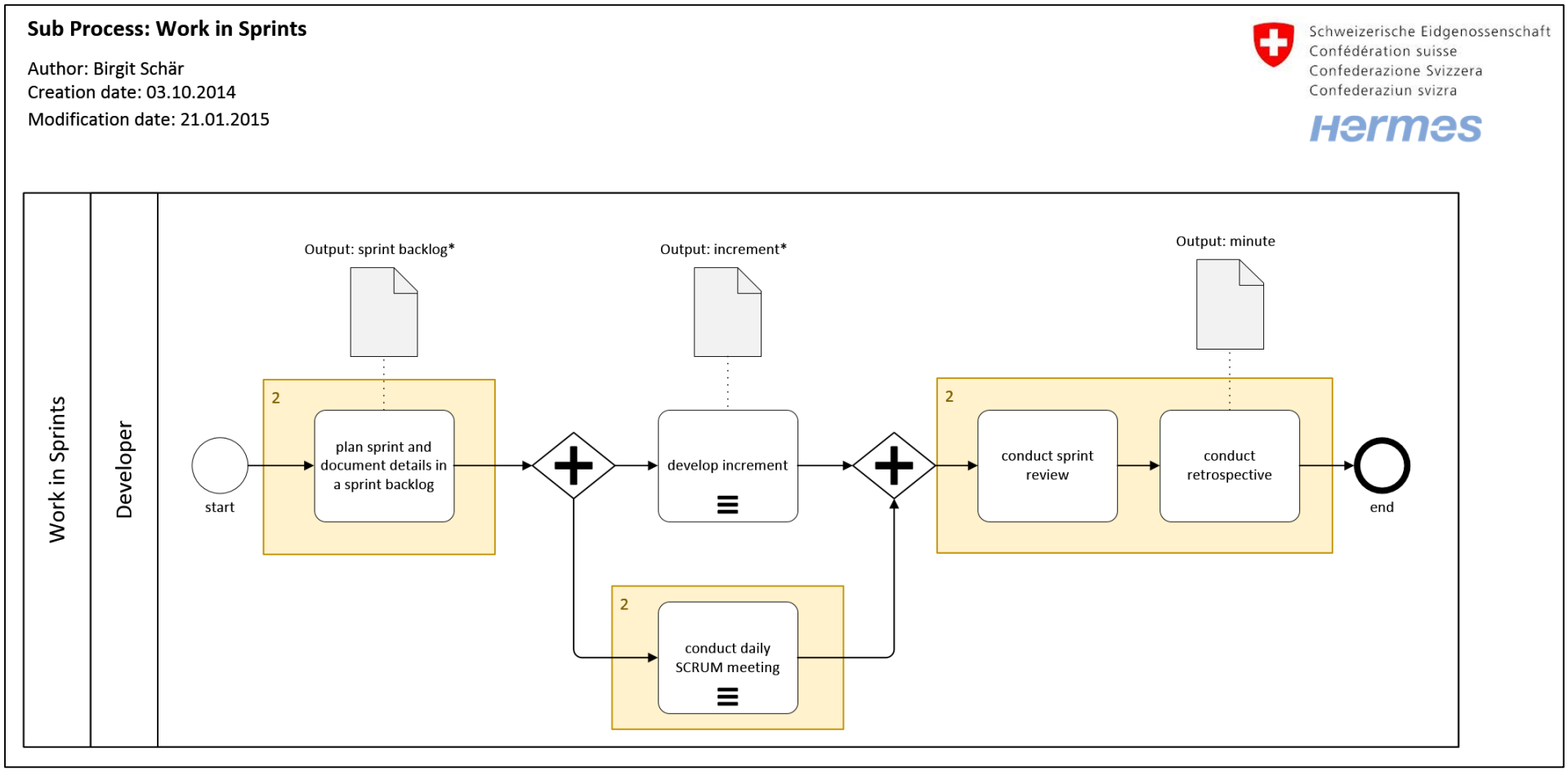
Hermes



14.3.9.4 Keep a Product Backlog



14.3.9.5 Work in Sprints



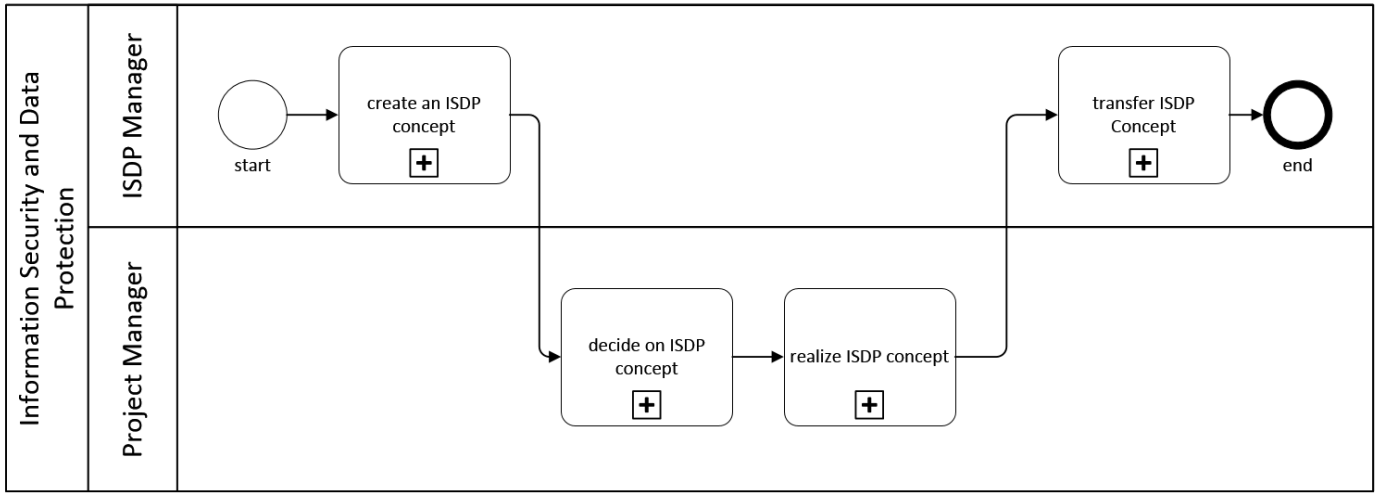
14.3.10 Module “Information Security and Data Protection”

Process: Module Information Security and Data Protection

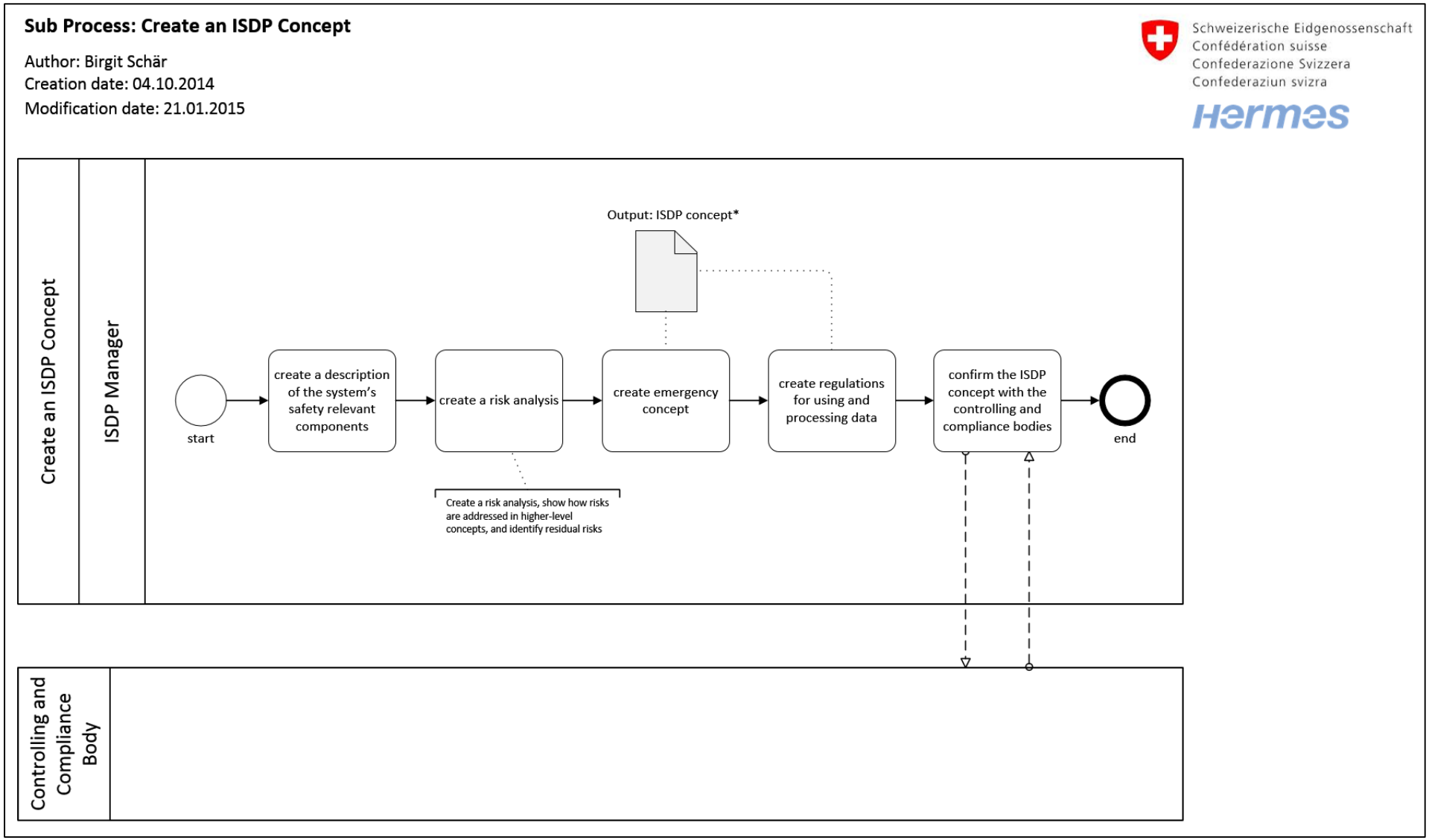
Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

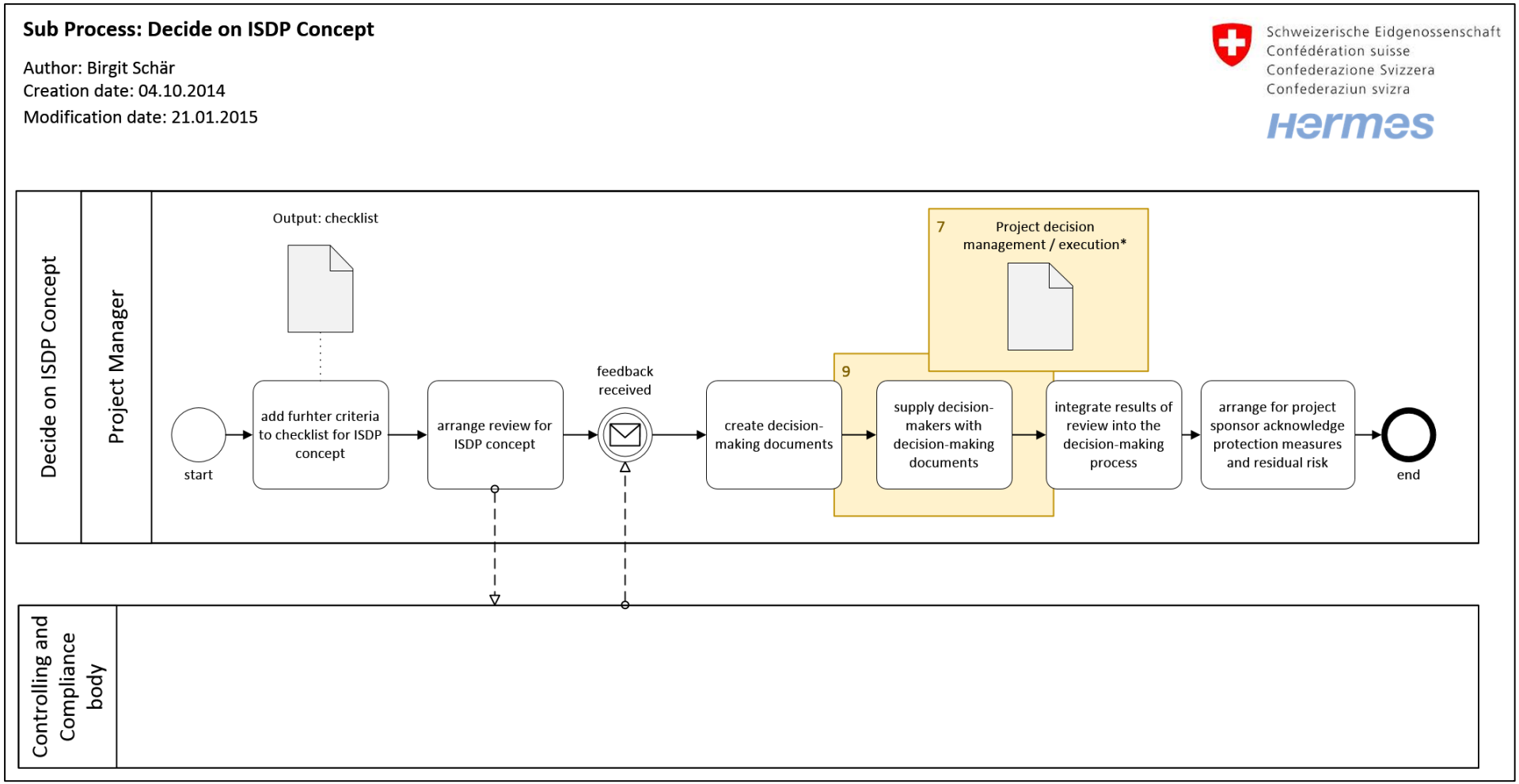
Hermes



14.3.10.1 Create an ISDP Concept



14.3.10.2 Decide on ISDP Concept



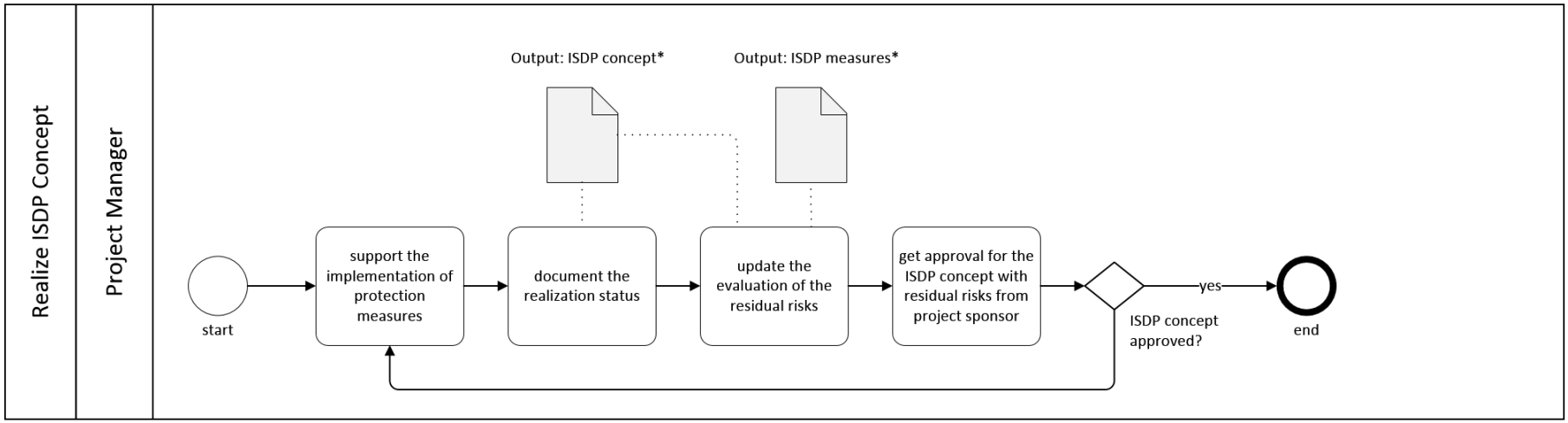
14.3.10.3 Realize ISDP Concept

Sub Process: Realize ISDP Concept

Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Hermes



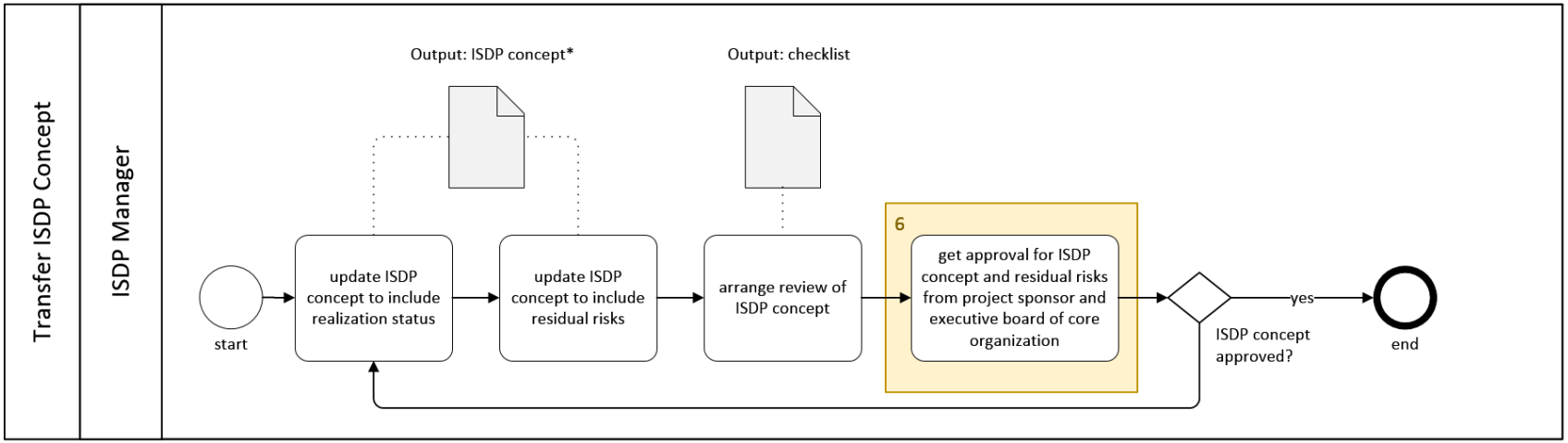
14.3.10.4 Transfer ISDP Concept

Sub Process: Transfer ISDP Concept

Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Hermes



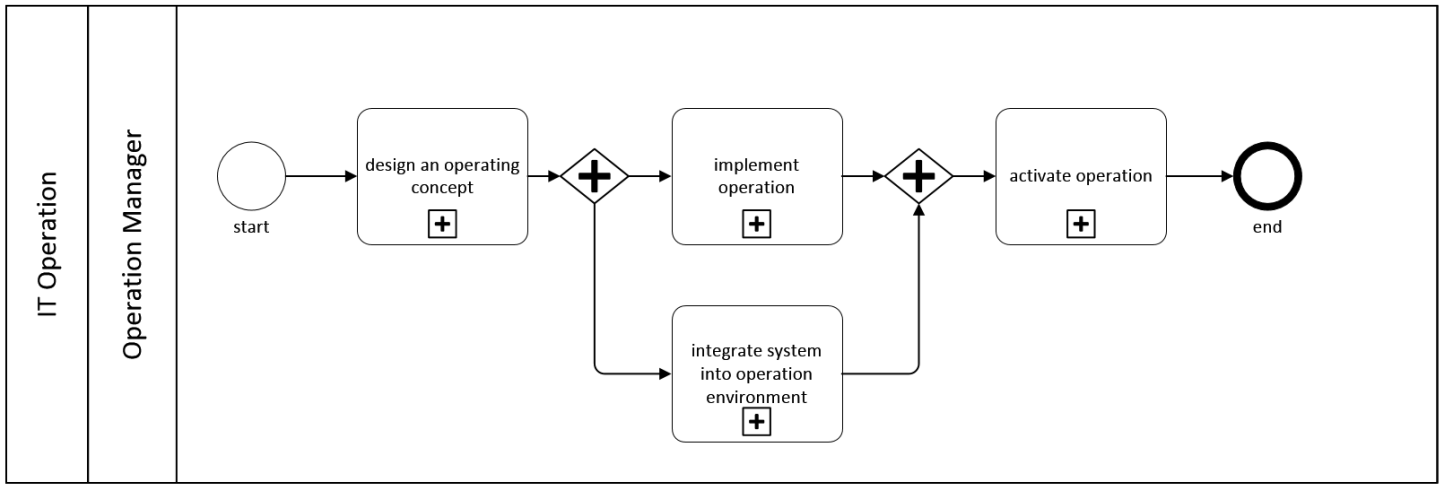
14.3.11 Module “IT Operation”

Process: Module IT Operation

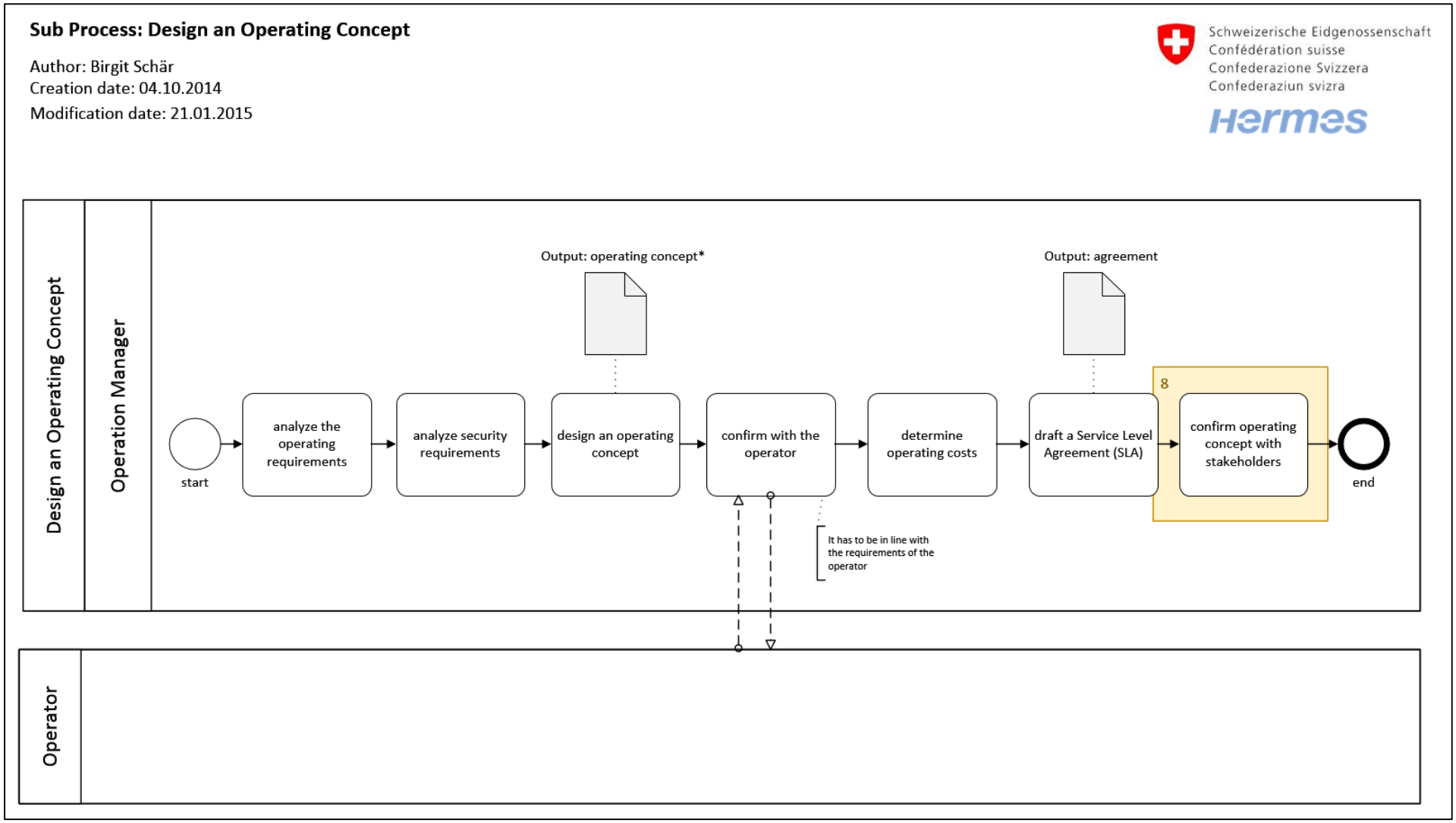
Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

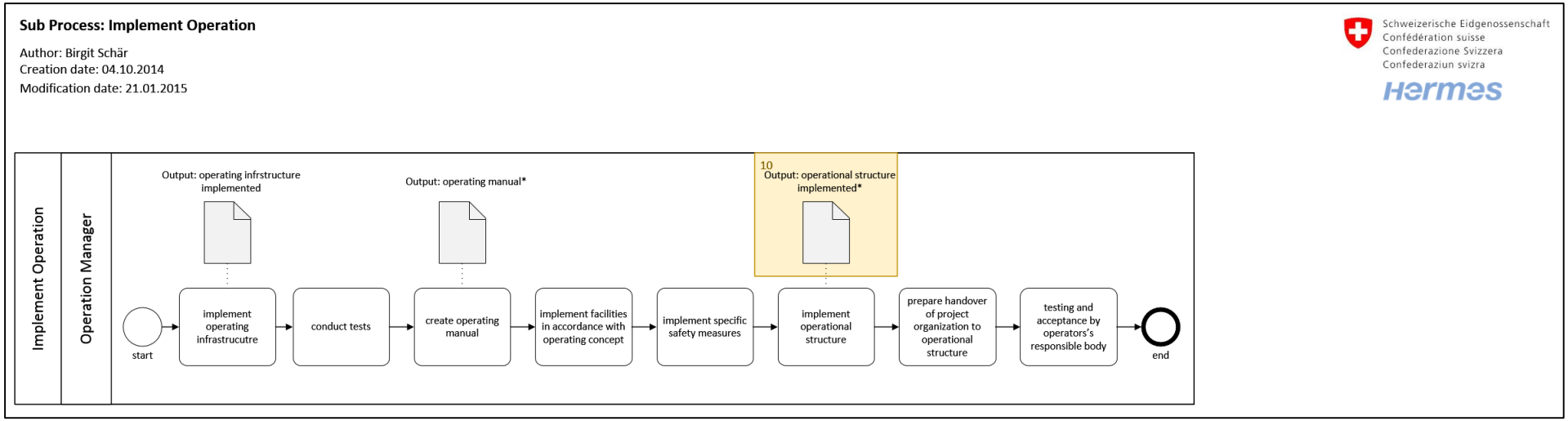
Hermes



14.3.11.1 Design an Operating Concept



14.3.11.2 Implement Operation



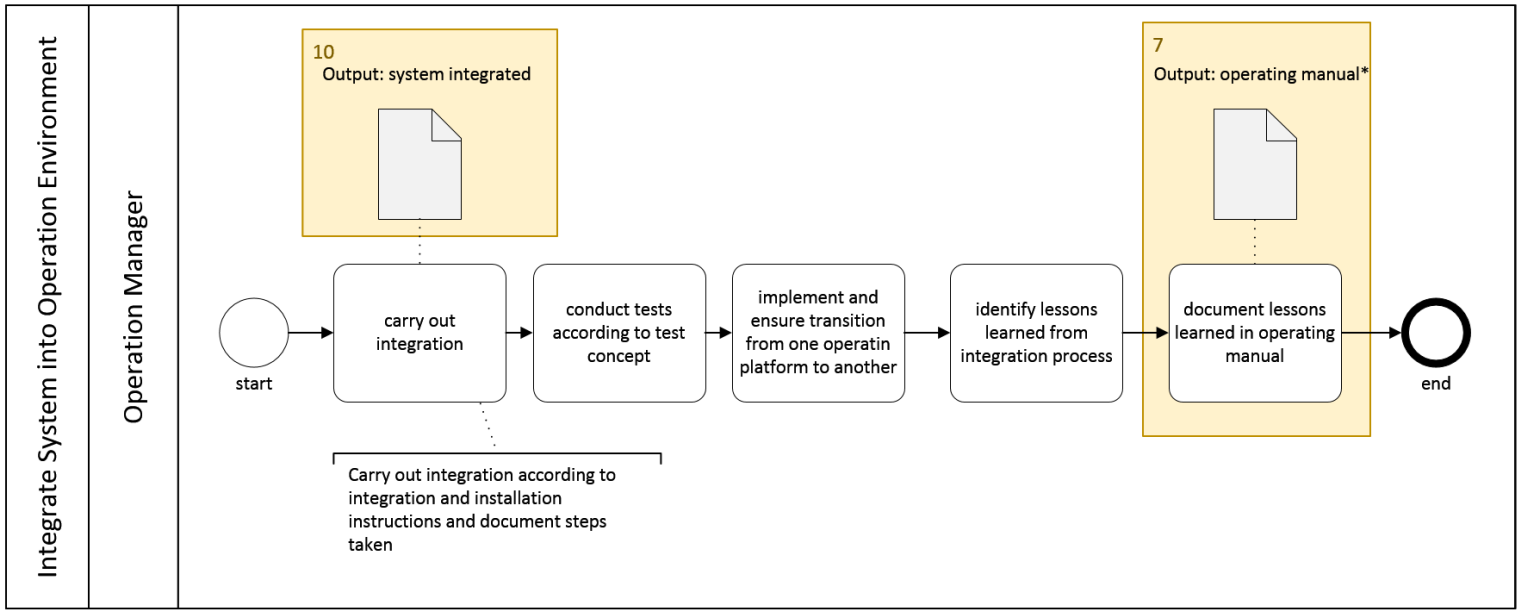
14.3.11.3 Integrate System into Operating Environment

Sub Process: Integrate System into Operating Environment

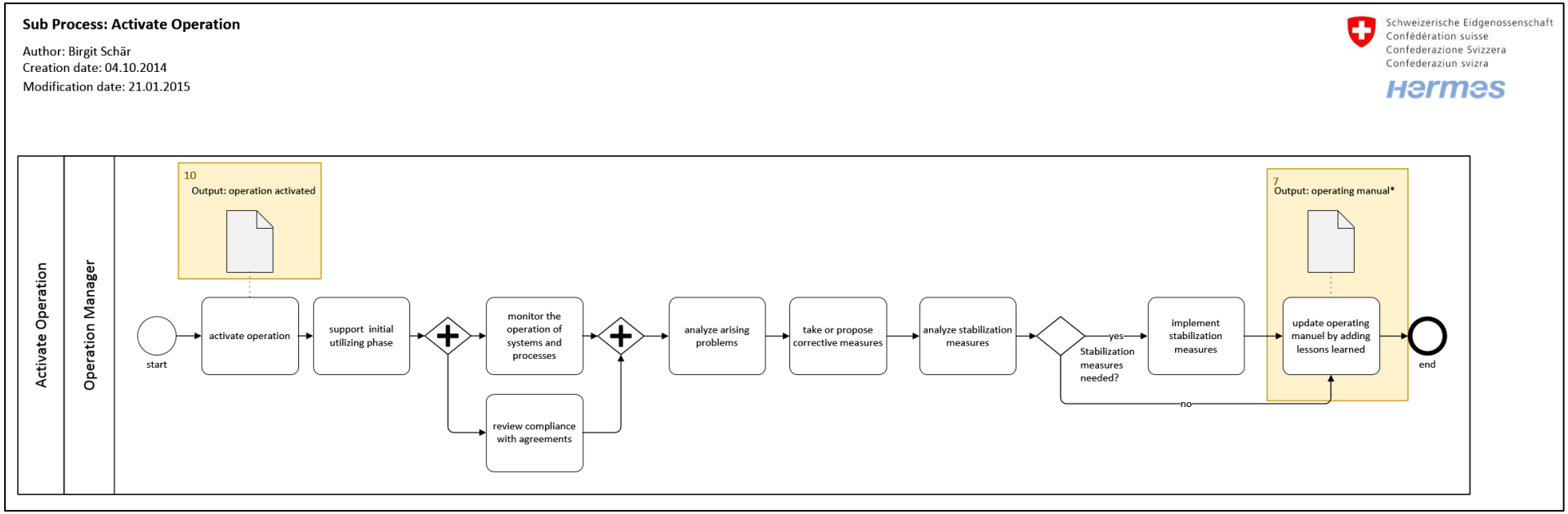
Author: Birgit Schär
Creation date: 04.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Hermes



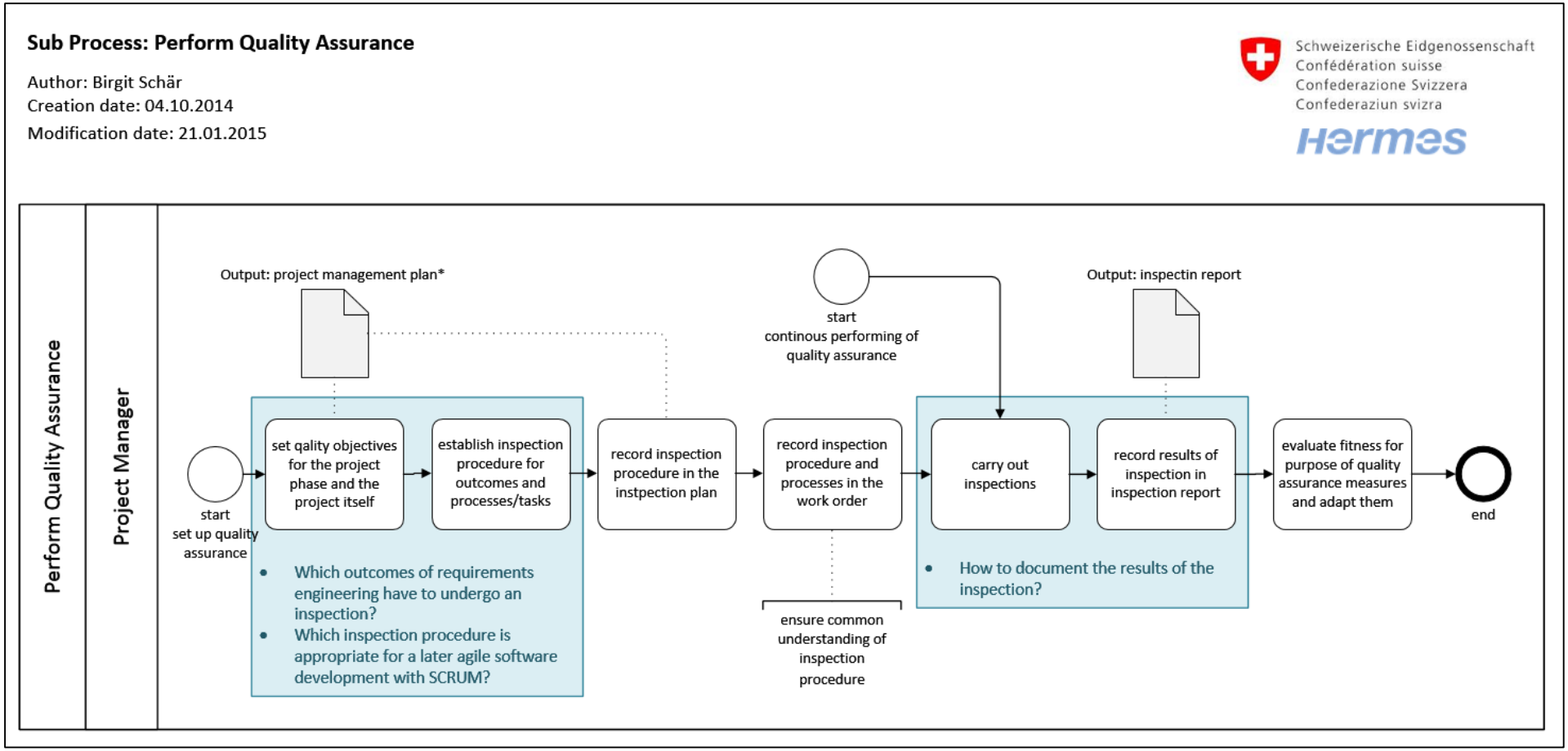
14.3.11.4 Activate Operation



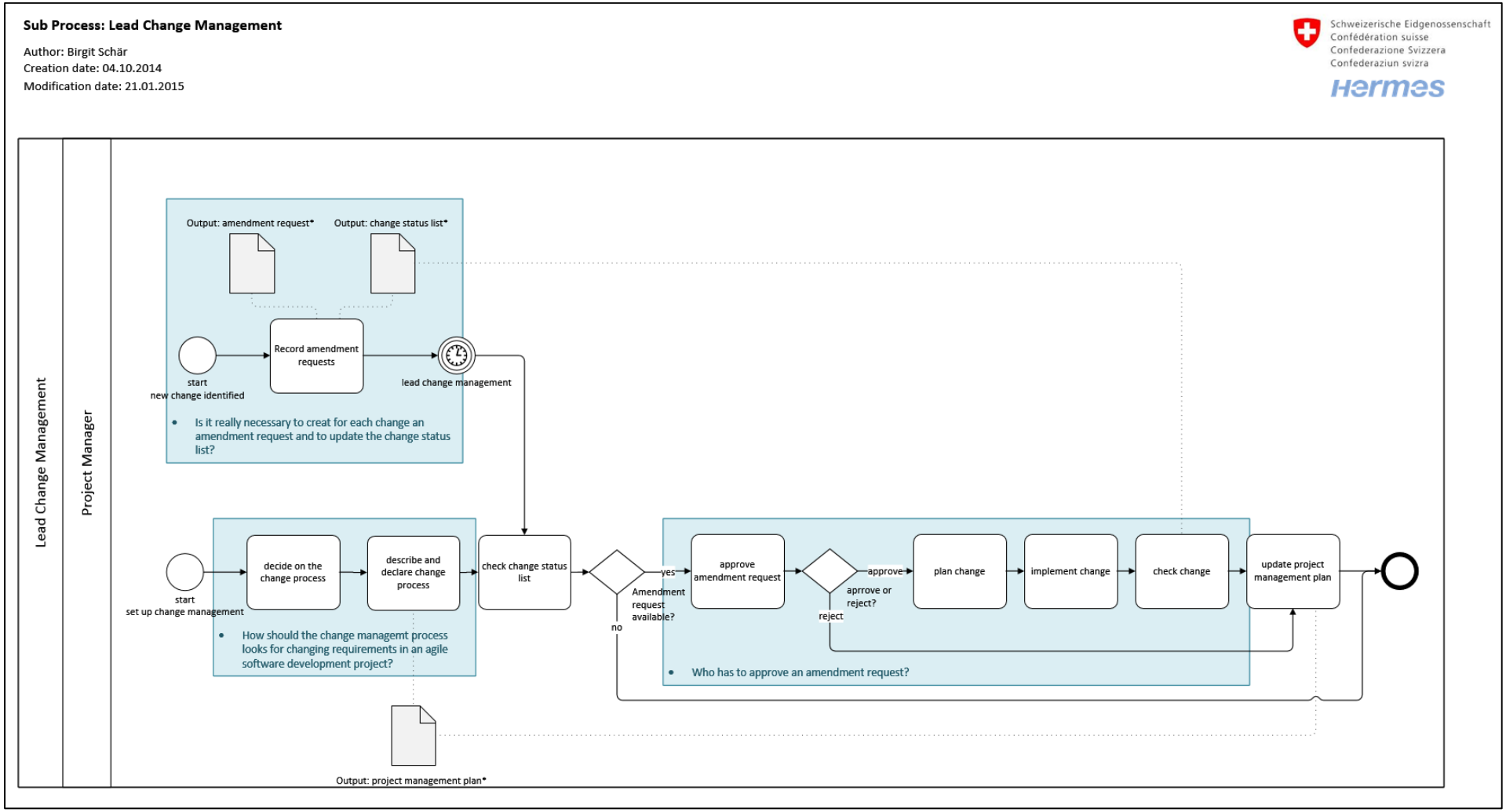
14.4 Appendix 4: Process Models for Analysis Part 2

14.4.1 Module “Project Management”

14.4.1.1 Perform Quality Assurance

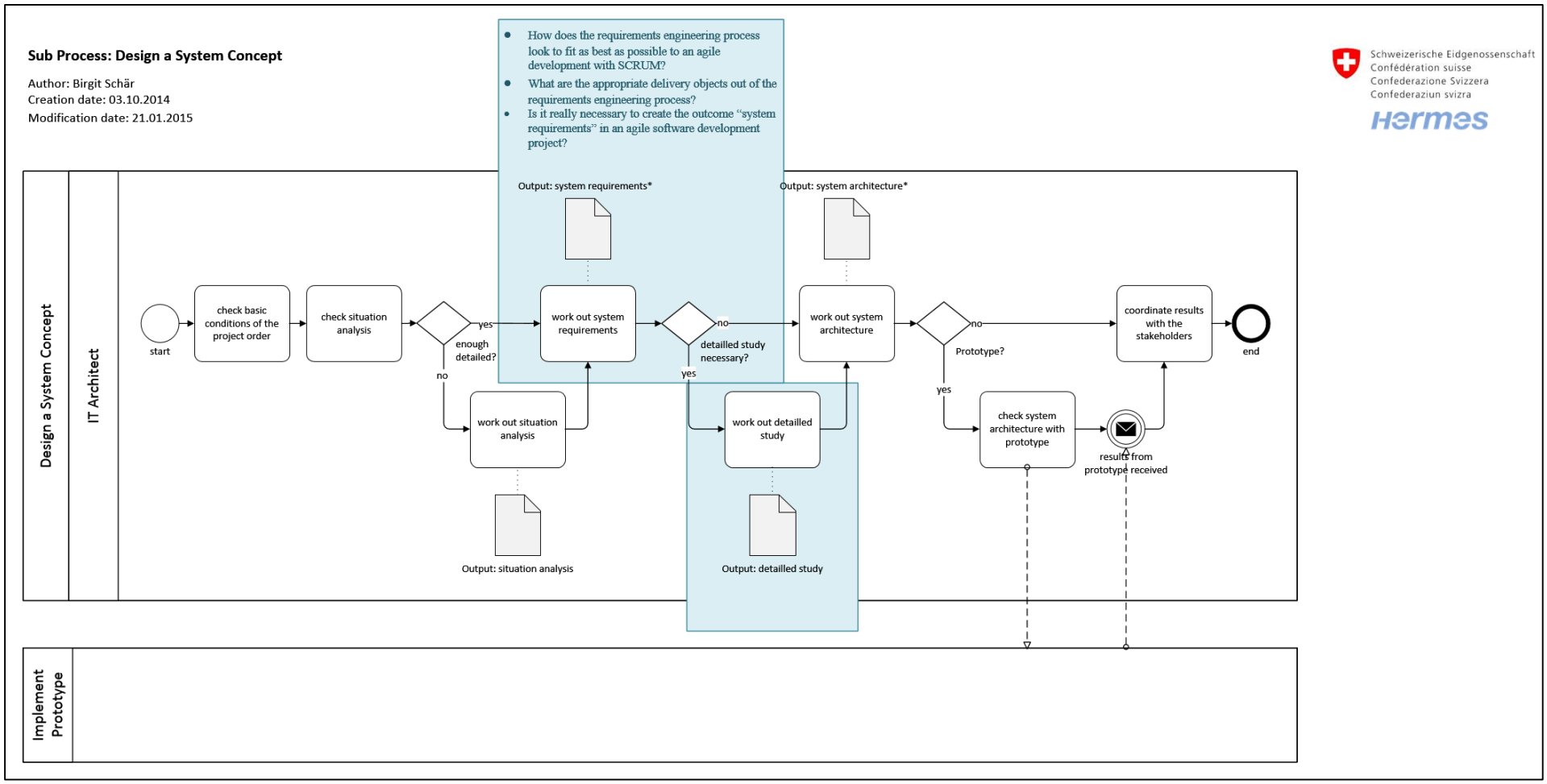


14.4.1.2 Lead Change Management



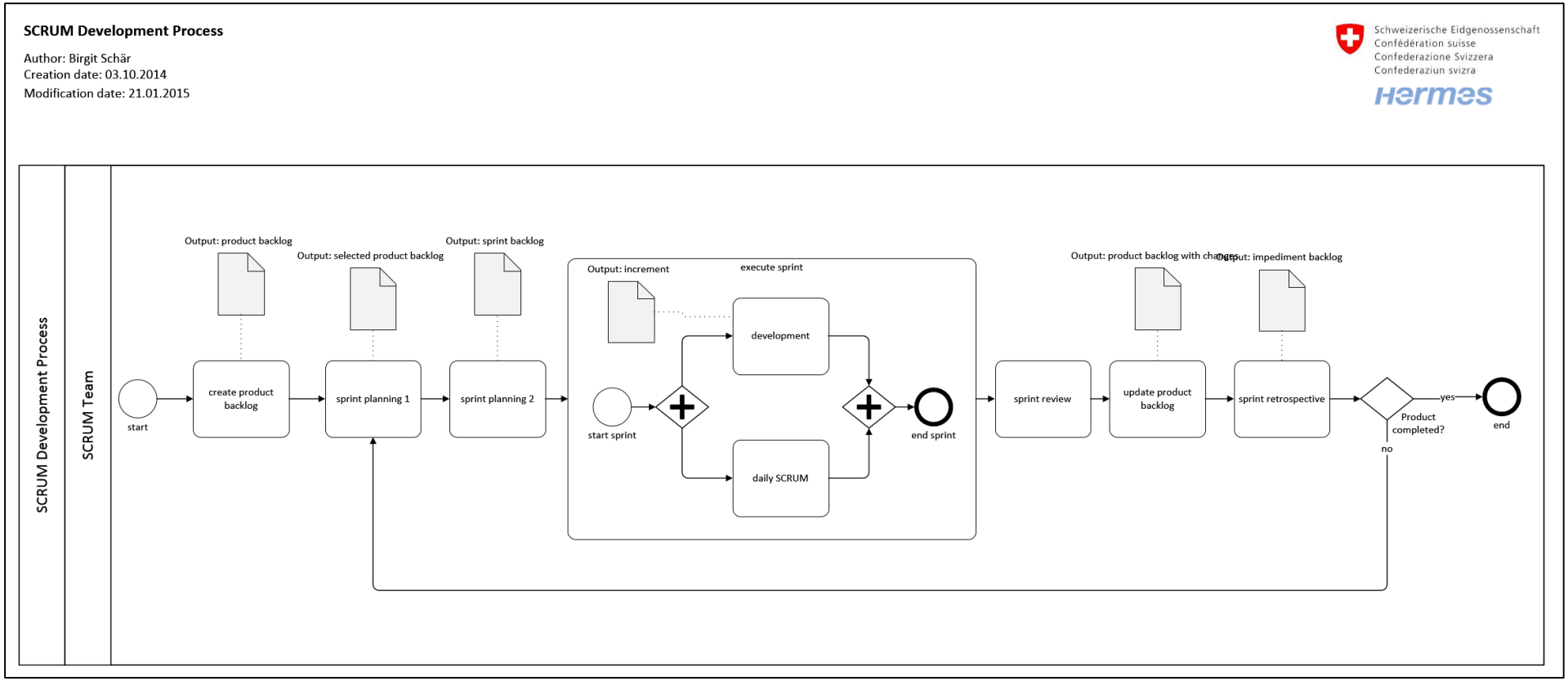
14.4.2 Module “IT System”

14.4.2.1 Design a System Concept



14.5 Appendix 5: Process Models for Analysis Part 3

14.5.1 SCRUM Development Process



14.5.2 Module “Agile Development”

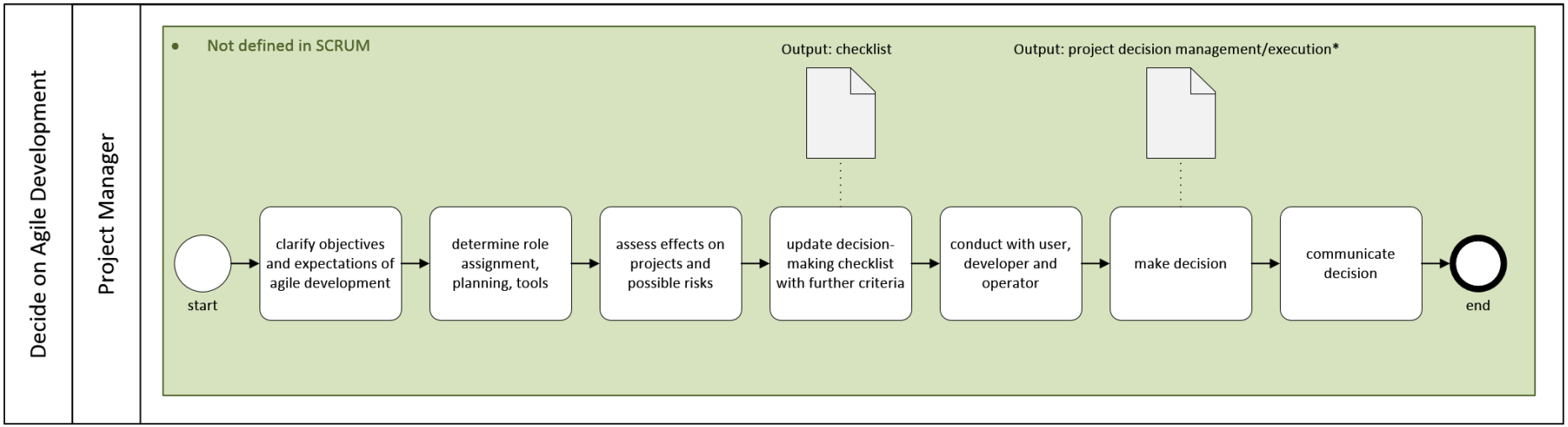
14.5.2.1 Decide on Agile Development using SCRUM

Sub Process: Decide on Agile Development

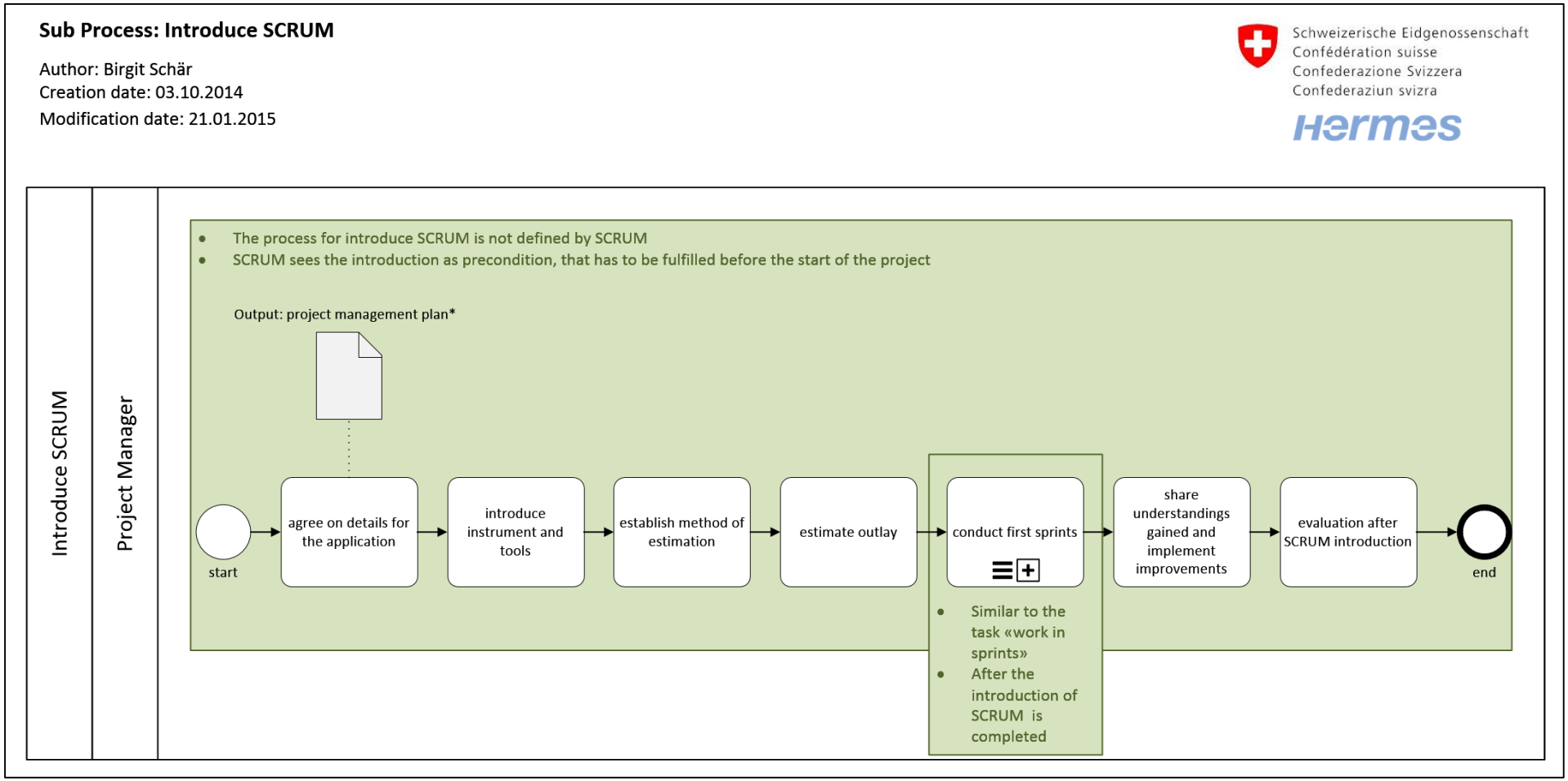
Author: Birgit Schär
Creation date: 03.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Hermes



14.5.2.2 Introduce SCRUM



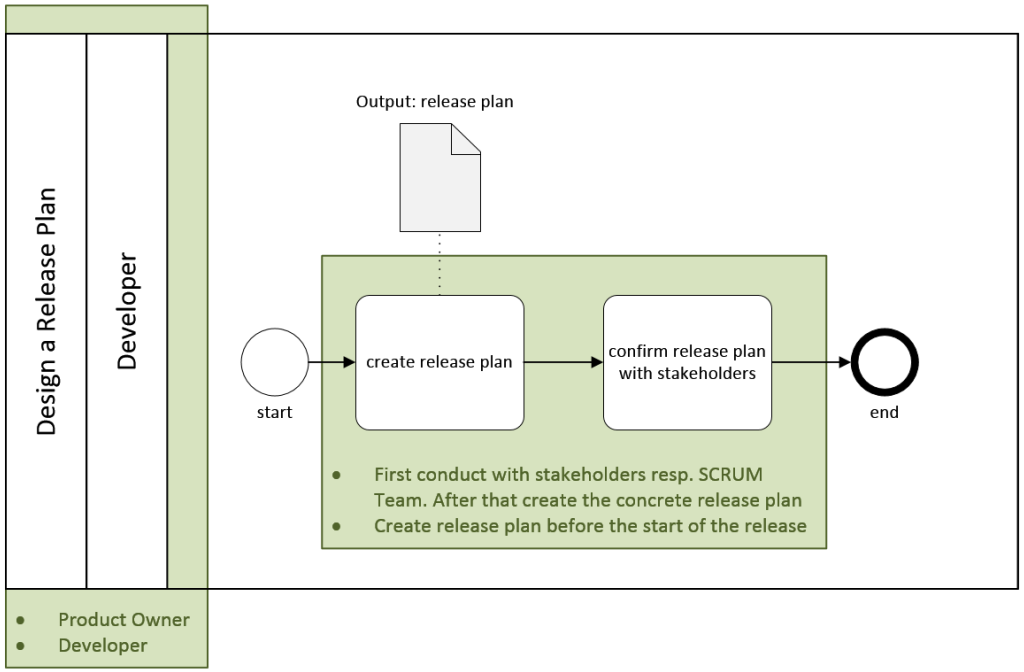
14.5.2.3 Design a Release Plan

Sub Process: Design a Release Plan

Author: Birgit Schär
Creation date: 03.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Hermes

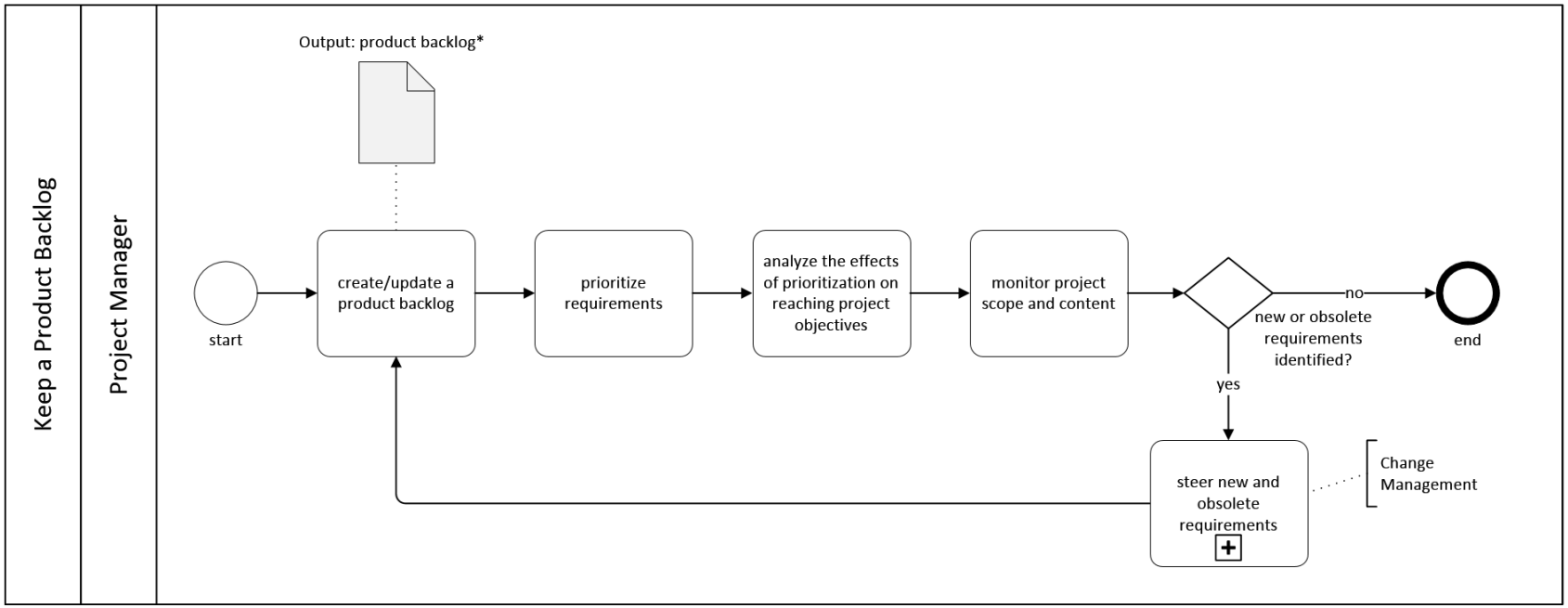


14.5.2.4 Keep a Product Backlog

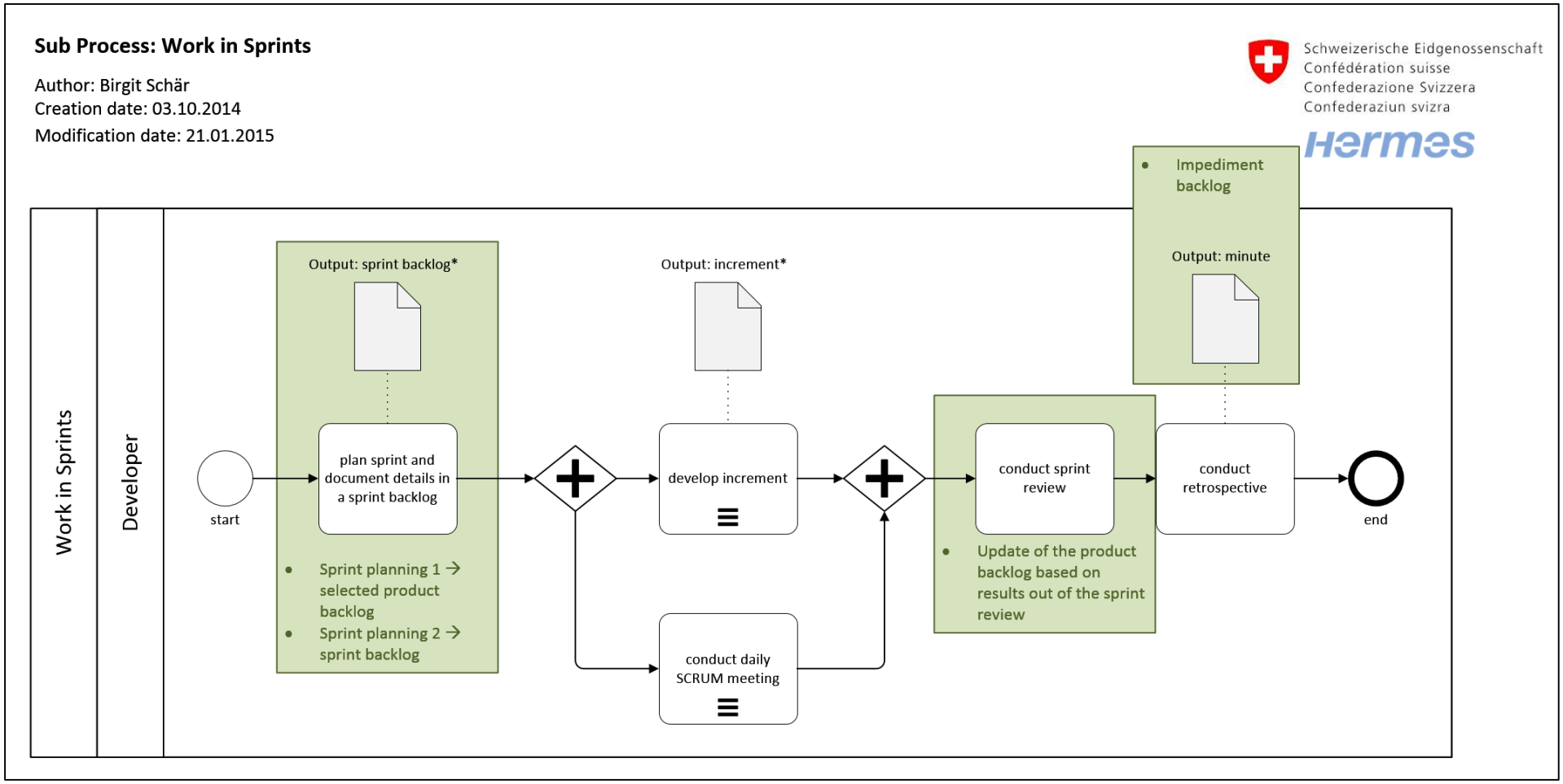
Sub Process: Keep a Product Backlog

Author: Birgit Schär
Creation date: 03.10.2014
Modification date: 21.01.2015

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Hermes

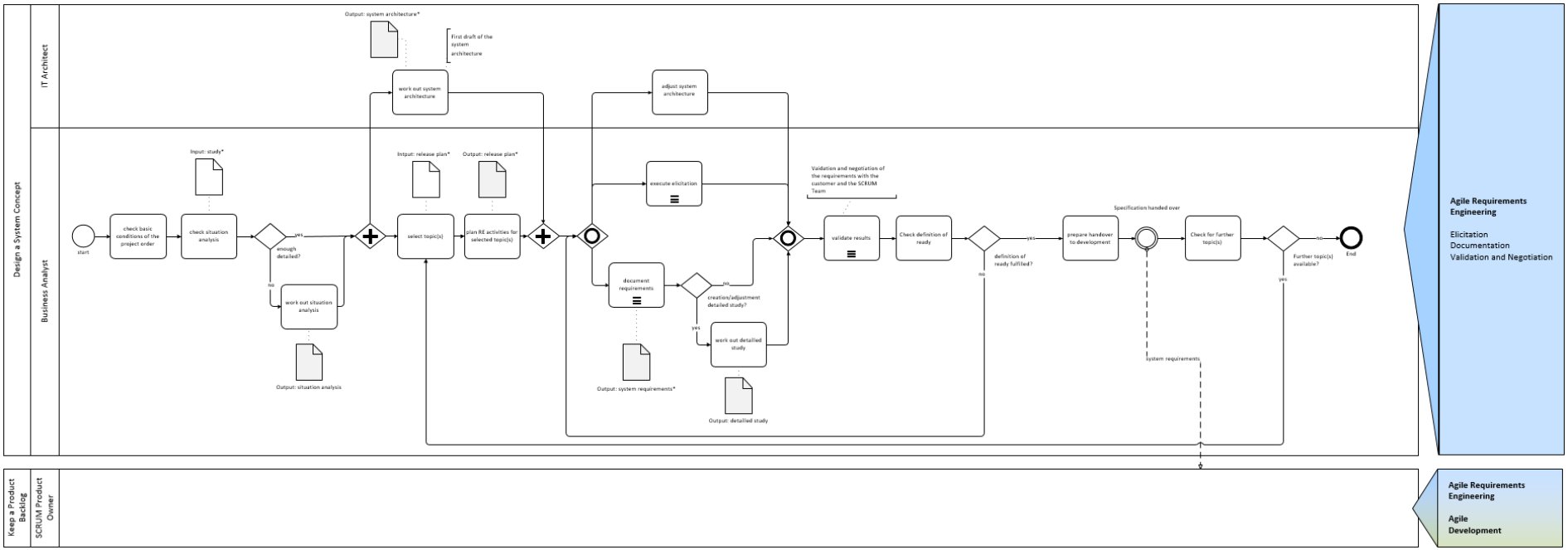


14.5.2.5 Work in Sprints

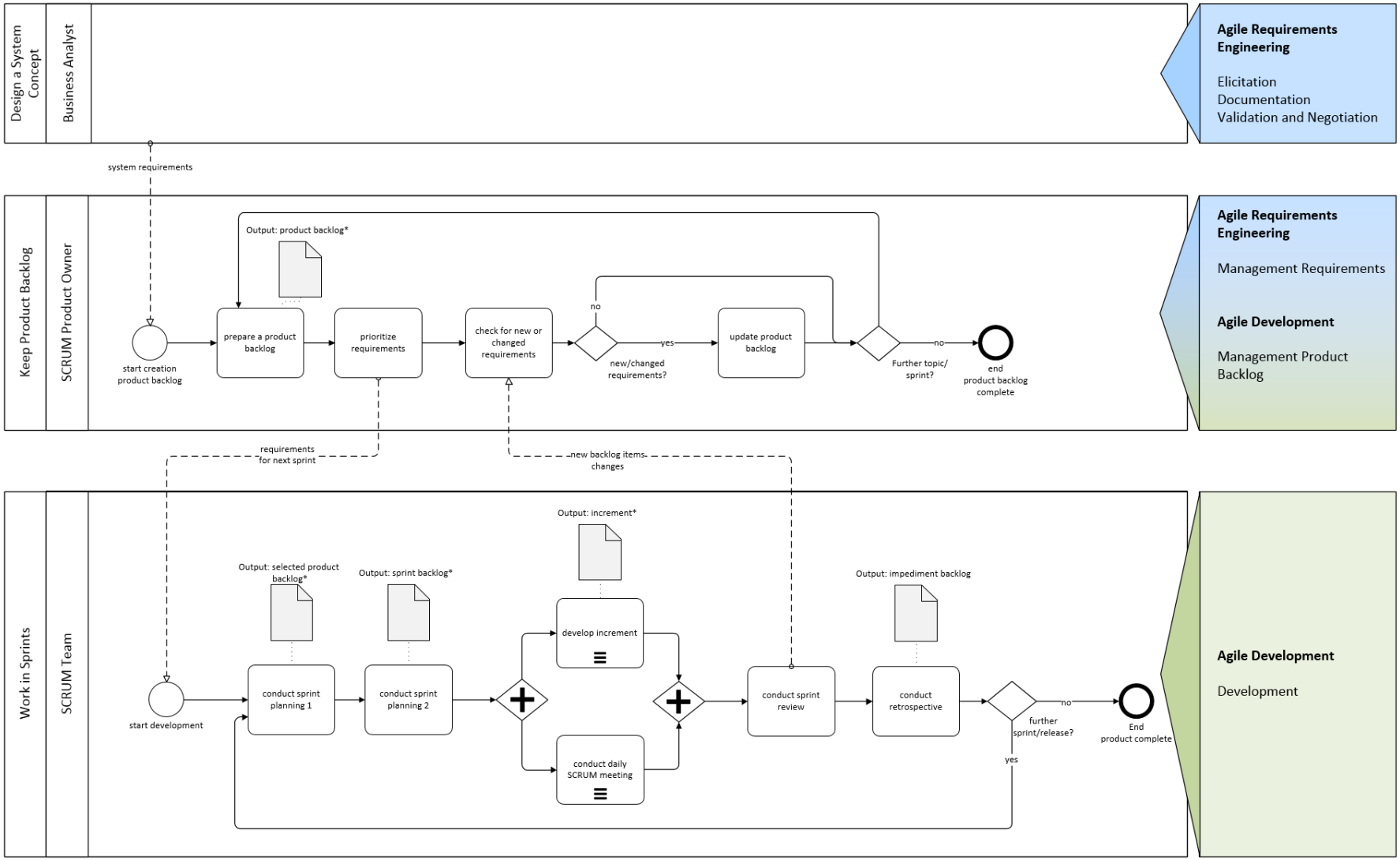


14.6 Appendix 6: Agile Requirements Engineering

14.6.1 Design a System Concept



14.6.2 Interaction Agile Requirements Engineering and Agile Development



14.7 Appendix 7: Result of the Assessment

The following table contains the results of the inspection.

Chapter Reference	Evidence / Proposed Amendment	Inspector	State of Consideration
Title	**Evidence:** The master thesis take reference on the agile development in the Swiss Federal Administration. The statements are visualized with such examples **Proposed amendment:** Extend the title in the sense of "illustrated by the development in complex programs and projects in the Swiss Federal Administration"	G. Eicher	Not taken into account Not possible to change the title
Executive Summary (section HERMES)	**Excerpt:** **"***HERMES 5 follows the traditional waterfall approach."* **Evidence:** „Traditional waterfall" is partly true. The chapter "Implementation units and Releases" in the reference handbook HERMES 5 shows how phase iterations for the continuous delivery and commissioning will be organized. In addition, the implementation can be done iteratively (specification / implementation). Important features are that HERMES covers the entire life cycle of a project from the perspective of the client and also contains those elements which do not affect the software developer. HERMES also regulates the cooperation between users, manufacturers and operators in the project and support the requirements of project governance and sustainability. Therefore, it adds SCRUM in all these points and provides the framework for the agile project management and agile development. **Proposed amendment:** Adjust the section	B. Kruschitz	Taken into account. I added a corresponding hint to the affected sections. **Comment on the evidence:** The master thesis focuses on requirements engineering. The main activities of requirements engineering happen in the phase "concept". Regarding the mentioned section in the reference book "implementation units and releases" (pages 157 to 158) only the phase "implementation" and "deployment" are considered for the iterative sequence.
Executive Summary (section HERMES)	**Excerpt:** *"…but not allowed to reduce elements…"* **Evidence:** G. Eicher: The tailoring and reduction of HERMES elements is allowed and desired (see HERMES reference handbook, page 7, point 1 and 2). J. Galeuchet: This sentence is partially false. Modules and outcomes could be removed, which leads to a new scenario.	G. Eicher, J. Galeuchet B. Kruschitz	Taken into account. Corresponding sections in the executive summary and in chapter 2.5.3 corrected.

	B. Kruschitz: It is allowed to remove or add method elements. The HERMES online tool supports the creation of the project structure plan respectively the creation of the organization specific scenario. The scalability is one essential characteristic of HERMES. **Proposed amendment:** Adjust the section		
1.2 Problem Statement (first section)	**Excerpt:** *"…, while for the software development part it increasingly uses the agile software development framework SCRUM."* **Evidence:** Is SCRUM an agile software development framework? This statement seems to me to require explanation and evokes associations that are most wrong. There should be a clear statement what SCRUM is and what not. The Scrum Guide could this be a good source. **Proposed amendment:** Adjust the section	B. Kruschitz	Taken into account. **Comment on the evidence:** According the SCRUM Guide, SCRUM is a framework for developing and sustaining complex products (Sutherland & Schwaber 2011). Corresponding sections adjusted.
1.2 Problem Statement	**Excerpt:** *"The delivery objects and the level of granularity is therefore an initial point of conflict when using HERMES 5 and SCRUM together."* **Evidence:** I not agree to this point. In an existing customer environment, such as the Swiss Federal Administration, requirements regarding IT architecture, IT security, IT operations etc. must be taken into account. One of the requirement (and framework for projects) is also the procurement law. These requirements are regardless of the used method and represent the actual point of conflict. The conflict thus not between HERMES and SCRUM but between the basic conditions in an organization and SCRUM. HERMES makes these conditions only visible. So the question is how such an environment can work agile with the existing basic condition. And that describes HERMES 5: It takes into account these basic conditions. **Proposed amendment:** Adjust the section	B. Kruschitz	Taken into account. **Comment on the evidence:** The delivery objects and level of granularity could be from the practical view still a conflict point. This mean that it is important to coordinate with the development to find out what they need at which depth for the implementation. I mitigated the sentence and explained it a bit more clearly. **Hint:** Exactly the question how such an environment can work agile with the existing basic condition is handled in the conclusion in chapter 7.
1.2 Problem Statement	**Excerpt:** *"After finalization of a phase, the results will be approved through the project board and the next phase will start."* **Evidence:** The role doesn't exist in HERMES. The project sponsor releases the next phase. **Proposed amendment:** Adjust the section	B. Kruschitz	Taken into account. **Comment on the evidence:** Absolutely correct. I used the false term. I meant that the project sponsor releases the phase (often under involvement of the steering committee members). Corresponding section corrected.

1.2 Problem Statement	**Excerpt:** *"In HERMES 5 it is normally not possible to return back to a previous phase, it is only possible to extend a phase with a good reason."* **Evidence:** It usually makes no sense to go back to an earlier phase. The exception is the implementation with implementation units (see first evidence to executive summary). The description of the problem area "Procedure" is not consistent, since the beginning of the development explicitly in the concept phase starts and runs throughout all phases up to the acceptance of the system. It is fundamentally important to understand that the developers actually not noticed the phase transition in practice. It is a concern, which affects the project sponsor. **Proposed amendment:** Adjust the section		Taken into accout. I reformulated the corresponding section, because it was not clear what I wanted to say. **Comment on the evidence:** I agree that the developer shouldn't recognize the phase transition, but I have also the opinion that the phase transition, which is a concern of the project sponsor, should be handled smoother in an agile software development project.
1.2 Problem Statement	**Excerpt:** *"In HERMES the knowledge transfer between the different roles takes place based on documentations, as not all roles are involved during the whole project."* **Evidence:** This statement is interpreted very freely. Roles are occupied by people, which could have multiple roles. The person who has the role "user representatives" has typically also the role "testers". Thus, it does not require knowledge transfer between roles. And certainly not via paper. The problem in practice is based on my experience rather the missing role differentiation of the SCRUM team in SCRUM. This is because SCRUM does not contain a development process that would allow the differentiation of roles. **Proposed amendment:** Adjust the section	B. Kruschitz	Taken into account. I reformulated the corresponding section. What I wanted to say was that it is in the Swiss Federal Administration depending on the contractual situation of external employees not always possible to involve the same people over a whole project. This is an organizational aspect and has nothing to do with the method HERMES.
1.2 Problem Statement	**Excerpt:** *"A solution respectively a guide for how using HERMES 5 and SCRUM together would bring benefit for the whole Swiss Federal Administration."* **Evidence:** This guide exists in the chapter "Instructions for use: Agile Project Management with Scrum and HERMES" in the reference handbook HERMES 5. **Proposed amendment:** Adjust the section	B. Kruschitz	Taken into account. Adjusted the corresponding section. **Comment on the evidence:** I meant extensive practical guide. The guide in the reference handbook is theoretical.
1.2 Problem Statement, figure 2	**Evidence:** I don't understand the figure 2. Important for projects in the Swiss Federal Administration would be the statement that different partners (user, creator, and operator) conduct joint projects in various offices and have to take on specific roles and responsibility for the results. HERMES regulates this interaction.	B. Kruschitz	Not taken into account. **Comment on the evidence:** I meant extensive practical guide. The guide in the reference handbook is rather theoretical.

	Proposed amendment: Adjust the section		
1.4 Research Questions and Objectives	**Excerpt:** *"Is it possible to develop a requirements engineering process, which both meets the requirements of the HERMES 5 project management method and takes advantage of the agile software development method SCRUM?"* **Evidence:** For me it is not clear what the requirements of HERMES are that have to be fulfilled at this point.	B. Kruschitz	Not taken into account respectively no changes necessary. **Comment on the evidence:** This is one of the sub questions to answer the primary research question.
1.5 Scope and Limitations of Scope	**Excerpt:** *"A lot of agile software development methods (e.g. Extreme Programming, KANBAN, Crystal, Feature Driven Development, and SCRUM) exist."* **Evidence:** According to Schwab/Sutherland SCRUM is not a software development process. It shows only the relative effectiveness of the process. **Proposed amendment:** Adjust the section	B. Kruschitz	Taken into account. **Comment on the evidence:** According the SCRUM Guide, SCRUM is a framework for developing and sustaining complex products (Sutherland & Schwaber 2011). Adjusted the corresponding section.
1.8 Rational of the Study (section 2 personal perspective)	**Excerpt:** "…One study (2010) does exist…combination of HERMES 5 and SCRUM…" is false. **Evidence:** G. Eicher: The study reference on the older version HERMES 2003. Therefore for HERMES 5 no study exists. B. Kruschitz: This statement is incorrect. The study of 2010 is based on HERMES 2003/2005 and is outdated. The HERMES 5.1 Reference Manual describes in the chapter "instructions for use" the interplay of HERMES 5 and SCRUM on a methodological level and explains the positioning of the two methods, the mapping of the methods elements (roles, results, responsibilities), and provides information on common application of the two methods. This chapter is a key chapter in understanding HERMES and SCRUM in agile project management. It shall also be taken into account and should be mentioned at most in the study that there are training courses on this topic, e.g. a one-day public course at the ZHAW Zurich University of Applied Sciences, in which the topic is explained and the challenges that will be shown on a real project can occur and how they can be mastered in a specific case. **Proposed amendment:** Adjust the section	G. Eicher B. Kruschitz	Taken into account. Corresponding sections corrected. **Comment on the evidence:** I mention only the literature in the sections and not the trainings courses, because it seems for me not appropriate to mention this in a scientific paper.

1.9 Contribution of the Study	**Excerpt:** *"…plans to launch HERMES 5.1…"* **Evidence:** The statement is old. The new release HERMES 5.1 was launched at the 3th of June 2014 (see start page of the HERMES website) **Proposed amendment:** Adjust the section	G. Eicher	Taken into account. Corresponding sections corrected.
2.5.4 Requirements Engineering in HERMES 5	**Excerpt:** *"The requirements engineering activities in HERMES 5 normally take place in the "concept" phase. The mandatory result "system requirements" is the output of the requirements engineering activities. HERMES 5 only defines the result "system requirements", but it neither defines how to execute the requirements engineering activities nor how to document (textual or model-based) the requirements within the result."* **Evidence:** The statement is false. Requirements engineering starts already in the phase "initiation". In this phase the project goals, the basic conditions and the basic requirements are worked out. Based on this information the solution variants are worked out and assessed. Requirements Engineering is also carried out after the phase "concept (e.g. "implementation", "deployment"). The result "detailed specification" is an example. HERMES 5 includes, in addition to the reference manual also aids (website, document templates, checklists etc.). The selected techniques are not documented in the reference manual, so that freedom consists in the choice of technique and these can be defined for a specific organization if necessary. HERMES 5 suggests at the level of document templates various techniques (e.g. for the target description, requirements documentation with use cases or user stories, risk assessment, the description of test cases, etc.). These can be either applied or replaced by other techniques. Please refer to the examples as templates (consider the German document templates. The English version is under construction and will be uploaded soon). HERMES does not pretend that results must be recorded in documents. The requirements can held in a dedicated tool, and are published (e.g. Enterprise Architect). This is also an indication that HERMES is well advised not to make any strict guidelines. **Proposed amendment:** Adjust the section	B. Kruschitz	Taken into account. Corresponding sections corrected. I had only considered the English templates.

2.6.1 Existing literature	**Excerpt:** *"The study HERMES and Agility (Federal IT Steering Unit 2010), furthermore, is not up to date according the Federal IT Steering Unit."* *"The study "HERMES and Agility" (Federal IT Steering Unit 2010) from the Federal IT Steering Unit investigates how weaknesses that…"* *"The newest findings regarding the use of HERMES 5 and SCRUM are documented on the HERMES 5 website (Federal IT Steering Unit 2014)."* **Evidence:** The study mentioned above is actually out of date. It refers to the old version HERMES 2003/2005. This was replaced by HERMES 5. It is questionable to mention here an old study that became obsolete with HERMES 5. **Proposed amendment:** Adjust the section	B. Kruschitz	Taken into account. Corresponding sentences adjusted.
2.6.1 Existing literature	**Excerpt:** *"The newest findings regarding the use of HERMES 5 and SCRUM are documented on the HERMES 5 website (Federal IT Steering Unit 2014)."* **Evidence:** The reference should generally not be the HERMES 5 site, but the HERMES 5 reference handbook (published on the website). The referenced section of the reference manual is called "User Information" and is part of the HERMES 5 reference handbook. It is in my opinion problematic when the Federal IT Steering Unit is the source and not the HERMES 5 reference handbook. The source is in all these cases in the document false. **Proposed amendment:** Adjust the section	B. Kruschitz	Taken into account. Corresponding sentence adjusted. Updated the sources. The authors of the reference handbook HERMES 5 are now directly named according the Harvard Reference Citations Style.
2.6.2.1 Structural Analysis (section project charter)	**Evidence:** The decision about agile development can be taken after it is clear who the development company is. Depending on the sourcing model, this may have the effect that the development partner, must be procured first. Very often, this decision can therefore be taken only at the stage concept. If you must know the developers already in the initialization phase that decision be made already there. HERMES does not prohibit this.	B. Kruschitz	No adjustment necessary.
2.6.2.1 Structural Analysis (section procurement)	**Evidence:** In the event of a WTO Government Procurement there are concrete measures such as defining and prioritizing user stories to conclude the possibility individual parts as a work contract, a set of basic functions / processes, which then form the basis for the development of other elements which optional can be ordered. If a binding	G. Eicher	Not taken into account. The section describes the finding of the analysis and don't provide a solution. Later on the topic isn't in the scope of the study.

	cost estimate exists, the item can be even-if handled by contract for work. Other elements are bonus / malus system within a cost competitive dialogue and roof.		
2.6.2.1 Structural Analysis (section change management)	**Excerpt:** *"The Federal IT Steering Unit points out on the HERMES website the importance of change management and explicitly mentions that SCRUM doesn't make change management unnecessary."* **Evidence:** The control of the services takes place by the prioritizing of the requirements over the product backlog and sprint backlogs. In performance agreements with fixed prices changes of the scope lead to a contractual adjustments. The base for this adjustment is the change status list. I interpret in this case the whole thing as following: The Product Backlog and Sprint Backlog are the basis. The change status list should be used only when changes in the scope of services lead to contractual adjustments. **Proposed amendment:** Mitigate the sentence (see the website "Agile Project Management with HERMES and Scrum")	J. Galeuchet	Already taken into account. Corresponding section corrected.
2.6.2.1 Structural Analysis (section change management)	**Evidence:** The results and the process of change management are not only related to requirements engineering but in general the whole project. Changes can be triggered even from outside the project and may not therefore be incorporated in an uncontrolled manner.	B. Kruschitz	**Comment on the evidence:** That is true. This section doesn't focus on requirements engineering alone. So no adjustments are necessary.
2.6.2.1 Structural Analysis (section release plan)	**Evidence:** The release plan can also be integrated into the project management plan and is therefore not a necessary result. He is primarily required for the coordination of the subprojects.	B. Kruschitz	Taken into account. Corresponding section adjusted.
2.7 Conclusion	**Excerpt:** *"…HERMES 5 change management contradicts the idea of SCRUM."* **Evidence:** It is not possible to say this such general. It depends on the openness of the delivery objects and on the mechanism for the extension. The Release Planning must not be identical with the Product Backlog	G. Eicher	Taken into account. Corresponding section adjusted. Statement was attenuated.
4 IT Project Landscape	**Excerpt:** *"…no federal project portfolio exists."* **Evidence:**	G. Eicher	**Comment on the evidence:** Statement of the section was already correct, but some sentence lead to a misunderstanding. Small adjustments made.

	It exists a portfolio of the ICT key projects and the SAP application "Cockpit IKT", in which all the studies, projects etc. are recorded with the corresponding status, risks etc. Beside the to-be values also the as-is values are shown.		
5.1.1 Analysis Part 1: Weaknesses, ID 1	**Excerpt:** *"…only define one responsible role."* **Evidence:** G. Eicher: How is it possible with HERMES that consider different organizational models? It is not possible to create such strict regulations for the organization. J. Galeuchet: I personally see this point not so serious, because the responsibilities are clearly defined at the level of "task". It is the responsibility of the person, which is responsible for the task, to determine depending on the project situation who is the best to create the delivery outcome	G. Eicher, J. Galeuchet	Statements taken into account. Reformulated the section respectively the improvement proposal.
5.1.1 Analysis Part 1: Weaknesses, ID 2	**Excerpt:** *"HERMES 5 doesn't define the responsibility of the different activities. This carries the danger that it is not clear from the reader's point of view who has to execute the activity."* **Evidence:** The responsibilities are clearly defined at the level of "tasks".	J. Galeuchet	Out of the statement it is clear that the sentence is not clear formulated. Reformulated the statement.
5.1.1 Analysis Part 1: Weaknesses, ID 6	**Excerpt:** Why is it not possible to list an activity more than one time?	G. Eicher	**Comment on the evidence:** This weakness comes out of the process perspective.
5.1.1 Analysis Part 1: Weaknesses, ID 4	**Excerpt:** *"The task description in the phase concept, implementation and deployment contains activities, which are already conducted in the phase initiation in the task "manage and control initiation."* **Evidence:** Yes, and that is so true. For example, in initial project charter their exposure (initialization phase) focus only on the risks that relates to the initialization phase. Upon validation of the project contract, the risks are mentioned again, but they affect the overall project, and the other phases. In addition, the risks also check regularly (e.g. recording of new risks).	J. Galeuchet	**Comment on evidence:** The statement is absolutely true. I meant the activities e.g. the definition of the change management process, which has to happen only once. I reformulated the statement that it is more clear what I meant with it.

5.1.1 Analysis Part 1: Weaknesses, ID 5	**Excerpt:** *"In the task description there is an activity, which is already implicitly covered through other activities."* **Evidence:** Which activity? (doesn't found/understood the reference)	J. Galeuchet	**Comment on evidence:** Only understandable with the process model in the appendix. Added a comment to have a view on the process models.
5.1.1 Analysis Part 1: Weaknesses, ID 8	**Excerpt:** *"The activity "confirm with stakeholder" gives a broad range for the activity".* **Evidence:** I have not found the activity "confirm with stakeholders"'. Basically, the approval process is company specific or defined by the parent organization and cannot be defined by HERMES 5 as "standard".	J. Galeuchet	**Comment on evidence:** Only understandable with the process model in the appendix. Added a comment to have a view on the process models.
5.1.1 Analysis Part 1: Weaknesses, ID 10	**Excerpt:** *"... Declare these outcomes as milestones."* **Evidence:** The logical linking between the elements bases on the SPEM-Model. How declare the reaching of a mile stone on otherwise?	G. Eicher	**Comment on evidence:** Ok, therefore it is clear why the milestones are declared as task.
5.1.4 Conclusion	**Evidence:** HERMES 5 was created for different kind of projects. The content of the reference book has to be done on a detailed level that it is general. **Proposed amendment:** If it is an improvement it has to be mentioned that this weakness is seen from the perspective of the development of an individual business software and that it is a proposition to overtake the changes for the other kind of projects	G. Eicher	Taken into account. Adjusted the section.
5.1 Detailed Analysis	**Evidence:** Seems to me clearly illustrated and logically with the visualizations and table in the appendix 2. For the "blue area marked" I would include in the table some more activities. At this point, it seems to me not very clear what criteria you have use for the assignments. From the perspective of requirements engineering process and the phase "Initiation" would have been interesting to analyze this, because in the first activities of the initialization already Requirements engineering should be started (e.g. stakeholder analysis). The question "Is it correct to start at the phase "concept" with agile development?" would have been interesting.	J. Jenni	Taken into account. **Comment on the evidence:** The assignment has been taken based on the gained knowledge from the process modeling of each task. The scope is limited to the phase "concept", "implementation" and "deployment" because they are affected by the module "Agile Development". I agree that it would be interesting to have also a look on the phase "initiation".

5.1.1 Analysis Part 1: Weaknesses	**Evidence:** I agree with you only in some potential improvements that are described in the table 13. Note: Around the responsibilities for the creation of artifacts I like to use the RACI model	J. Jenni	**Comment on the evidence:** I discussed the single points with G. Eicher. A finding was that they are formulated not always clearly, which is also shown by the evidences of J. Galeuchet. I reformulated the weaknesses and added an example if it is helpful. Furthermore I added the comment that the reader has to consider the process models when reading the weaknesses.
5.1.2 Analysis Part 2, Requirements Engineering	**Evidence:** As noted in my feedback to chapter 5.1, I would assign more activities here related to requirements engineering. Give you to the questions in the "Questions Regarding execution in to agile software development project" answers? From which references take the "Today's practical application / meaning"?	J. Jenni	**Comment on the evidence:** It is a legitimate point. Due to the time horizon it was not possible to extend the scope more. The questions in the area "Questions regarding execution in agile software development projects" are answered later in chapter 6. The paragraphs were provided with appropriate references.
5.1.3 Analysis Part 3: Agile Software Development and 5.1.4 Conclusion	**Evidence:** I believe that Scrum is not limited on the points: • Decide on agile development using SCRUM • Introduce SCRUM • Design a release plan • Keep a product backlog • Work in sprints Therefore a statement, *"... if SCRUM is applied correctly"* cannot be answered due to this basis.	J. Jenni	Taken into account. Statement reformulated.
5.3 Expectations of Development in Requirements Engineering	**Evidence:** The title does not seem to match the content of the chapter. The content is understandable. From my personal experience, I am amazed about at the results to the questions.	J. Jenni	Title not adjusted. **Comment on the evidence:** The chapter summarizes the findings of the online survey, where the participants (in majority developers and leaders of development teams) answered the question regarding their expectation in requirements engineering. It have taken into account that the participants of the online survey are principally engaged in the Swiss Federal Administration.

6 Design "Agile Requirements Engineering"	**Evidence:** I understand the embedding of agile requirements engineering as module and I see this within the constraints of HERMES 5 (5.1) as appropriate. I support your changes proposed in sections 6.3.1, 6.3.2 and 6.3.3.	J. Jenni	No changes necessary.
13 List of Abbreviations	**Evidence:** The abbreviation FOITT is missing. **Proposed amendment:** Add the abbreviation FOITT to the list of abbreviations	G. Eicher	Taken into account. Added to the list of abbreviations.