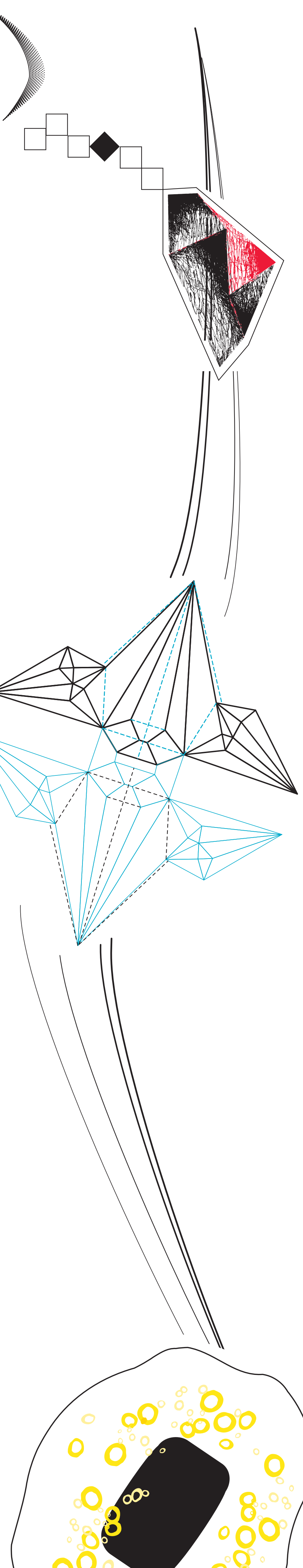


November 29, 2017

MASTER THESIS

DESIGNING A REQUIREMENTS ENGINEERING PROCESS FOR PERFORMANCE MONITORING

Evenynke Terpstra

UNIVERSITY OF TWENTE
FACULTY OF ELECTRICAL ENGINEERING,
MATHEMATICS AND COMPUTER SCIENCE

GRADUATION COMMITTEE
dr. M. Daneva (University of Twente)
dr. N. Sikkel (University of Twente)
drs. N.A. Kienhuis (The Backbone B.V.)
P.B. Klijndijk (The Backbone B.V.)

thebackbone
UNIVERSITY OF TWENTE.

SUMMARY

PURPOSE Requirements engineering processes (REPs) are critical to the success of software projects. Despite the increasing body of knowledge in REPs, for many organizations deficient requirements are one of the top 5 reasons for software project failure. This thesis reports on a project aiming to design a REP for a specific software product: End-User-Performance (EUP), of a small company in The Netherlands, The Backbone. The history of the product is relatively short at the time of writing this thesis. Similarly to many other small companies, The Backbone did not have a structured REP, nor documented the software requirements in a repeatable way across projects. Therefore, The Backbone believes that using a structured REP would help them focus their staff's energy on the most important requirements activities in order to achieve requirements of better quality and a more time-efficient REP.

RESEARCH PROCESS For the purpose of designing the new REP for the EUP monitoring service of The Backbone, this project employed the Design Science Research Methodology (DSRM) of Peffers et al. Three iterations of the DSRM by designing and evaluation of a REP led to the proposal of a new REP for the EUP monitoring service of The Backbone. This REP is based on objectives from The Backbone, empirical findings in literature, findings in the current REP practice of The Backbone, and evaluations of possible new REPs. Each iteration of the DSRM concluded with different validity evaluations consisting of interviews with the end-user of the REP, an observation of the REP in use, and interviews with professionals in the field of requirements engineering that are employed by other organisations next to The Backbone.

RESULTS The REP proposed to The Backbone consists of five phases: (1) sales, (2) requirements composition, (3) implementing, (4) performance monitoring, and (5) evaluation. In this REP, the results of the previous phase are the input for the next phase. The REP is iterative. It is designed in a way that allows to go back to a previous phase if it appears that more information from that previous phase is needed to continue the REP. The proposed REP yields a new artefact, the design document, which leads the REP-practising consultant through the process phases and encourages him or her to document the requirements in a structured way. The applicability and the suitability of our proposed REP is evaluated in a real-world case. In this evaluation, The Backbone found that the proposed REP is an improve-

ment in respect to their current REP in terms of usability, usefulness, time-efficiency, quality of written requirements, implementation quality of requirements, and professional image.

RECOMMENDATIONS The REP proposed in this thesis is directly applicable for the EUP product of The Backbone. The recommendations for the future of the REP that crystallised during this master project, concern: (1) the use of technology in support of the process, and (2) the use of artefacts. We came up with three recommendations for leveraging technological support for requirements engineering:

- Build a modular software tool around the REP, make use of clickpaths to search in big EUP projects.
- Link the software tool around the REP with a database.
- Use artificial intelligence in the software tool to compare requirements and evaluations automatically to each other.

Next, we formulated two recommendations regarding artefacts:

- Treat the design document of the REP as a living document.
- Use a general form of the REP for other custom software projects inside The Backbone.

PREFACE

With this thesis, I'll end my adventure of being a student of the University of Twente. This adventure started in the summer of 2011 with the bachelor Business & IT. It was in my second year during the course Requirements Engineering that I realised that requirements engineering was the direction I wanted to go with my studies. After my bachelor thesis on negotiation methods for requirements engineering I was completely sure about this direction. Unfortunately, there was no master in requirements engineering specific, so I decided to go for the master Computer Science with the specialisation Information and Software Engineering, in which I had a lot of space to find my own way. Even though, finding my own way in this was not always easy, I'm satisfied with the courses I took and the knowledge I gathered during my master's.

During my adventure at the University of Twente, I gathered a lot of knowledge about Business & IT and Computer Science. But above all, I met lovely people and had a lovely time!

In the process of my master thesis, I've been supervised by Maya and Klaas for which I really want to thank them. I want to thank them for their critical notes, their trust in me, for the nice talks during our meetings, and of course I want to thank them for all the times we worked together before.

The research described in this thesis is conducted at a company called The Backbone. I want to thank Nico and Peter for our brainstorm sessions especially at the start of the process during my time there. I also want to thank Serge and Rob for their willingness to cooperate to this research and their trust in taking me with them to customers.

Furthermore, I want to thank Inge and Stefan for making the time for me to interview them.

I also want to thank Regie, who made time for reading parts of my thesis and comment on them.

Last, but certainly not least, I want to thank Jarmo, my family, and my friends for their support during my studies.

Evenynke Terpstra
Deventer, November 2017

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

CRM	Customer Relationship Management
COTS	Commerical Off The Shelf
DSRM	Design Science Research Methodology
ERP	Enterprise Resource Planning
EUP	End-User Performance
GBA	Goal Based Approaches
JAD	Joint Application Development
OMS	Operations Management Suite
RE	Requirements Engineering
REP	Requirements Engineering Process
RQ	Research Question
SBC	Service Based Computing
SCOM	Systems Center Operations Manager
SLA	Service Level Agreement
SME	Small and Medium Enterprises

1 INTRODUCTION

Requirements engineering processes (REPs) are very important for software projects to become successful [5], [20], [21]. Deficient requirements are the number one reason for software projects to fail [21]. It is preferred to know whether requirements are deficient as early as possible in the software project, since the amount of rework, and therefore also costs, grow exponentially during the project [10].

This thesis describes the research that has led to the design of a new REP for an organisation called The Backbone. The Backbone is a small to medium enterprise (SME) in The Netherlands focusing on Business, IT, and End-User Performance monitoring. The Backbone offers, among others, a monitoring service that gives insights in IT application performances, identity and access performances, and IT service level agreement (SLA) performances. The software product that is used for this monitoring service is called the End-User Performance (EUP). The existence of the EUP is very short at the time of writing. The REP designed in this thesis is designed especially for the EUP.

In SMEs it is not common to have a structured REP, nor is it common to document the software requirements [23]. Despite this, The Backbone believes that using a structured REP leads to less deficient requirements and a more efficient REP. The current REP practice at The Backbone lacks structured documentation of information and requirements. The lack of documentation would become very inconvenient in the case of illness of the responsible employee of The Backbone. Besides that, the lack of documentation can also lead to an incomplete implementation of the requirements since the responsible employee of The Backbone can forget some requirements.

There is a lot of scientific literature available about REPs. However, by implementing this literature in an organisation, one has to take things like the organisation, its customers, and its employees into account.

The design science research methodology (DSRM) of Peffers et al. [31] is used to design the new REP for the EUP monitoring service of The Backbone. Three iterations of the DSRM by designing and evaluating a REP led to the proposal of a new REP for the EUP monitoring service of The Backbone. This REP is based on objectives from The Backbone, literature, findings in the current REP practice of The Backbone, and evaluations of possible new REPs.

1.1 RESEARCH QUESTIONS

1. **RQ1: What would be a good requirements engineering process for the End-User Performance monitoring service of The Backbone based on literature and the current requirements engineering process of The Backbone?** To answer this question, first the requirements that The Backbone has for such a process must be clear. Second, the findings of the research topics for changes in the REP also belong here. Then, the first design of the new REP (REP v.0) will be introduced and elaborated.

2. **RQ2: How do employees of The Backbone experience the proposed requirements engineering process, does it improve the work?** To answer this question, we will evaluate the usefulness and utility of the proposed REP. The evaluation will consist of an interview with the end-user of the proposed REP: the performance consultant of The Backbone. During the interview, the performance consultant gives feedback on the REP as proposed in RQ 1.

3. **RQ3: What would be a good requirements engineering process for the End-User Performance monitoring service of The Backbone based on the evaluation with employees of The Backbone?** Based on the feedback of the performance consultant, REP v.0 will be updated to REP v.1.

4. **RQ4: How do employees and customers of The Backbone experience the updated version of the proposed requirements engineering process, does it improve the work?** REP v.1 will be evaluated by the performance consultant of The Backbone and a customer who just bought the EUP product. This evaluation consists of observations and interviews. The observations will be done at the moments the REP v.1 is used by the performance consultant and/or the customer. After the observations, the performance consultant as well as the customer will be interviewed. During these interviews, the interviewees are asked for feedback on the REP v.1.

5. **RQ5: What would be a good requirements engineering process for the End-User Performance monitoring service of The Backbone based on the evaluation with employees and customers of The Backbone?** The final version of the proposed REP (REP v.2) will be designed by updating REP v.1 with the feedback given by the performance consultant and the customer.

6. **RQ6: What do professionals in the field of requirements engineering think of the proposed requirements engineering process, does it work?** Finally, REP v.2 will be evaluated by the professionals in the field of requirements engineering (RE)

working in other companies than The Backbone. This evaluation consists of an interview with each of the professionals. Professionals in RE outside The Backbone are used to gather other opinions on REP v.2. During these interviews, the interviewees are asked for feedback on REP v.2.

2 CONTEXT

This chapter provides information on the context in which this research took place.

This research is carried out at a small IT company with 22 employees in The Netherlands called The Backbone and proposes a new REP for their End-User Performance (EUP) monitoring service as already mentioned in chapter 1. Therefore, this section gives information about The Backbone and their EUP product.

2.1 THE BACKBONE

The Backbone is a specialist in the area of hybrid monitoring solutions based on two Microsoft products: Systems Center Operations Manager (SCOM) and Operations Management Suite (OMS). The Backbone sees IT as 'the enabler' for the main processes of every organisation. Therefore, the IT of organisations needs to be protected. This is the reason for The Backbone to support organisations by protecting their business continuity. The Backbone protects business continuity by the use of chain monitoring and end-user monitoring of the main applications of the customer. The Backbone is operating in this field for more than ten years already.

Through time, the IT landscape of an organisation is getting more complex by IT applications getting linked and integrated with each other. This complexity results in the need for monitoring solutions. With the products of The Backbone and their expertise in embedding applications in organisations, multiple stakeholders can have their own dashboard that gives them insights on the availability, performance, and security of the main applications, portals, and IT infrastructure. Monitoring solutions can exist of chain monitoring, end-user monitoring, dashboards, and reports [38].

The customers database of The Backbone contains, among others, (academic) hospitals and other health care organisations. Also municipalities, other (semi-)governmental institutions, higher educational institutions, and organisations in the industrial and services sectors belong to the customers database of The Backbone [37].

Until now, the customers only contain organisations located in The Netherlands. However, The Backbone has a partner abroad that is also going to sell the products of The Backbone. Therefore, it is quite possible for The Backbone to have non-Dutch customers as well in their customer database in the near future.

2.2 END-USER PERFORMANCE MONITORING

The product of The Backbone this research focusses on is the EUP, which measures and monitors the performance of applications from the end-user's perspective. The EUP does this by running scripts which contain several steps to simulate end-users. For example, the script for an email application contains the following steps: starting the application, create a new email, writing text in this email, filling in the receiving email address, sending the email, receiving the email, opening the email, checking the text in the email, deleting the email, and finally closing the application. Scripts can start again directly after successfully finishing the previous attempt. Performance is measured by measuring the time needed for each individual step and for the simulation as a whole. The data gets processed and finally, gets visualised [36].

This product can be used in three different scopes:

1. First, the EUP can help with troubleshooting. It measures if the applications used in the scripts are working and if they do not exceed the set time thresholds.

2. Next, if the EUP focusses on end-user performance on authentication. The identity and access performance of the applications in the scripts are tested to find whether authorised accounts get access to the applications and non-authorised do not get access to the applications. The EUP also measures the time needed for these tests.

3. Finally, the EUP can also be used to compare the measurements of the applications in the scripts with the Service Level Agreements (SLAs) of the applications.

This research focusses on the REP of the EUP product as a whole and is therefore not focussed on a specific scope.

3 BACKGROUND

In this chapter, background information is provided on REPs and on the current REP practice for the EUP monitoring service at The Backbone. This information is gathered during a previous research, it is placed in this thesis as background information. A literature review is done on REPs in general, whilst the current REP practice is found via a triangulation research. The triangulation research consisted of two interviews, three observations, and one documentary study.

3.1 REQUIREMENT ENGINEERING PROCESSES

A lot is written in literature about REPs. However, the literature is not consistent in which components should be in a REP. Therefore, some literature is compared to each other and the common components together form a REP as is used in this research.

In this research the following components are adopted in REPs:

1. **Elicitation** [2], [11], [21], [26], [28], [40], sometimes also called discovery [5], or gathering [4].
2. **Specification** [2], [11], [26], [28], [40], sometimes also called documenting [4], [5], or modelling [21].
3. **Validation and verification** [2], [4], [5], [11], [21], [26], [28], [40].
4. **Managing** [4], [11], [26], [28], sometimes also called maintaining [5].

Below, these four components are described in detail and are summarised with their relationships in Figure 1. The arrows in Figure 1 represent relationships between phases. These relationships are described in more detail below as well.

3.1.1 *Elicitation*

In the *elicitation* phase, requirements and domain specific knowledge are gathered from users and other stakeholders [2], [4], [26], [28].

The reliability of requirements gathered in the *elicitation* phase, depends on the communication skills of the requirements engineer [4], [11]. An elicitation technique is a method for mediating in the communication between the requirement engineer and the stakeholders [14].

There are different techniques for requirement elicitation in literature [4], [26]. Not every elicitation technique is useful in every situ-

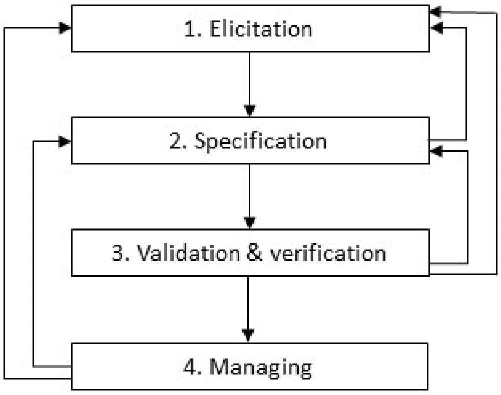


Figure 1: Requirements engineering process

ation and one technique does not cover all the requirements that need to be elicited [4], [14], [26]. Examples of elicitation techniques are:

- **Stakeholder analysis** [26]: During stakeholder analysis, the goal is to find out who the stakeholders are and which goals they have for the system. Especially since some of the stakeholders finance the project, or have to contribute with effort. This can be done in a meeting with all the stakeholders, in meetings with smaller groups of stakeholders, or individual interviews.

- **Interviews** [4], [21], [26], [28]: Interviews are a useful technique to find out the day-to-day work and problems. An important thing is to ask the interviewee about critical tasks. Since interviewees can feel intimidated by the question why they do a certain task the way they do it, it helps to ask them when they do that certain task. Interviews can be structured, unstructured, or semi-structured.

- **Questionnaires** [4], [26]: Questionnaires are a technique to get information on the day-to-day work and problems from many people at once in an time-efficient and quick way. If you want to get statistics, you need to ask closed questions. If you want to get opinions and suggestion, open questions are useful. In a questionnaire, only a limited depth of knowledge is possible and you cannot check if the participants understood your questions correctly and with open questions you may misunderstand their answers.

- **Observation** [4], [26], [28]: Observations lead to a more truthful and complete answer to what users do with the current system and how they do that, than during interviews or questionnaires. It is important to analyse the users without interfering them. Observations are a very time consuming technique.

- **Brainstorming** [4], [26]: For brainstorm sessions, it is of high importance to create a stimulating and focused atmosphere in a group of stakeholders. The participants can come up with ideas on the future system and an analyst writes every idea down. These ideas may never be criticized. If people do not find the creativity needed for generating new ideas, the analyst can help by raising issues that are noticed during former elicitation meetings.

- **Prototyping** [4], [26], [28]: Prototypes are simplified version, or parts, of the system. By using prototypes you get feedback from the stakeholders and it can be checked if the system is realistic feasible. New requirements can also be elicited from the prototype, for example details of the user interface.

- **Scenarios** [4], [28]: A scenario is a set of steps that simulate a task that the system should be able to perform. Requirements can be found in the steps that the system needs to be able to perform.

- **Task demonstration** [26]: Task demonstration is actually a form of interviewing and observation, in which you ask the users to show how they execute a certain task.

- **Joint Application Development (JAD)** [4]: JAD sessions are well structured and involve all stakeholders. In a JAD session, problems and solutions are discussed. Because all the stakeholders are in the session, decisions can be made directly. Since all stakeholders are at the JAD session, the JAD session is not about the main goals of the system. That is done, before all the stakeholders are involved.

- **Focus Groups** [21], [26]: Focus groups are more structured brainstorm sessions. The stakeholders are separated into groups. Each group comes up with problems in the current way of working. Then, the groups think of the ideal solution to this problem. At the end of the session, all the groups tell individually what their high priority issues are. It is important that every group gets solutions to some of the issues they identified with a high priority.

- **Goal Based Approaches (GBA)** [4]: In GBAs, high-level goals for the system are decomposed and elaborated into subgoals. Those subgoals are also further decomposed and refined, etc. This process ends at the level where individual requirements are elicited. This technique is only useful in situations where only high-level goals are available. Otherwise, the chance is that an error in a high-level goal makes that the requirements elicited with the GBA technique are incorrect.

- **Viewpoints** [4]: Viewpoints model the domain from different perspectives. A system can for example be explained in its features, as well as in its interfaces. Viewpoints do not make it easy to represent non-functional requirements.

3.1.2 *Specification*

After the requirements are elicited in the *elicitation* phase, they need to be described [2], [11], [40]. That is done in this *specification* phase (Figure 1). According to Aurum & Wohlin, a properly structured and controlled specification is even required to deliver high quality software on time and within budget [4]. A specification can be documented in a formal format, as well as in an informal format [28].

Two well-known templates for requirements specification used by requirement engineers are IEEE Std 830-1998 Software Requirements Specification [22] and Volere Requirements Specification Template [32].

In practice, most companies use text documents, with or without a standardized template, spreadsheets, and diagrams, as ERD and UML, to specify their requirements [19].

A well-defined requirements specification needs to be readable and understandable. It meets these criteria if [40]:

- the requirements in the specification are unambiguous,
- the specification is complete,
- the specification is consistent,
- the requirements in the specification are prioritised,
- the requirements in the specification are verifiable, and
- the requirements in the specification are traceable.

During the *specification* phase, incomplete requirements can be noticed [4]. Therefore, new requirements can get elicited, as can be seen in the relationship between the *specification* and *elicitation* phase in Figure 1.

3.1.3 *Validation & verification*

When the requirements are specified, they need to be validated and verified (Figure 1). Validation is the determination with the stakeholders that the correct requirements have been specified [11], [21], [26], [28], [40]. Verification is the determination that the specification has no internal inconsistencies [11], [21], [26], [40].

During *validation & verification*, new requirements can be found as can be seen in the relationship between the *validation & verification* phase and the *elicitation* phase in Figure 1. The relationship between the *validation & verification* phase and the *specification* phase is that existing requirements may need change, or need to be deleted (Figure 1).

The scenarios from the *elicitation* phase are useful for validating the requirements [4], as are prototypes and design screen pictures [26], [28]. For verification, acceptance tests are a useful tool [26].

3.1.4 *Managing*

During the project, the requirements can change for various reasons, for example because the stakeholders' understanding of the problem changes, or requirements that can be more precisely defined in a later phase of the project than in an early phase [28]. The specification therefore needs to be updated during the project [4], [5], [11], [26]. This happens in the *managing* phase, after the *validation & verification* is done (Figure 1).

The relationship between the *managing* phase and the *elicitation* phase in Figure 1 shows that new requirements can get elicited in the *managing* phase. Existing requirements can need to be changed or deleted in the *managing* phase. This is shown by the relationship between the *managing* phase and the *specification* phase in Figure 1.

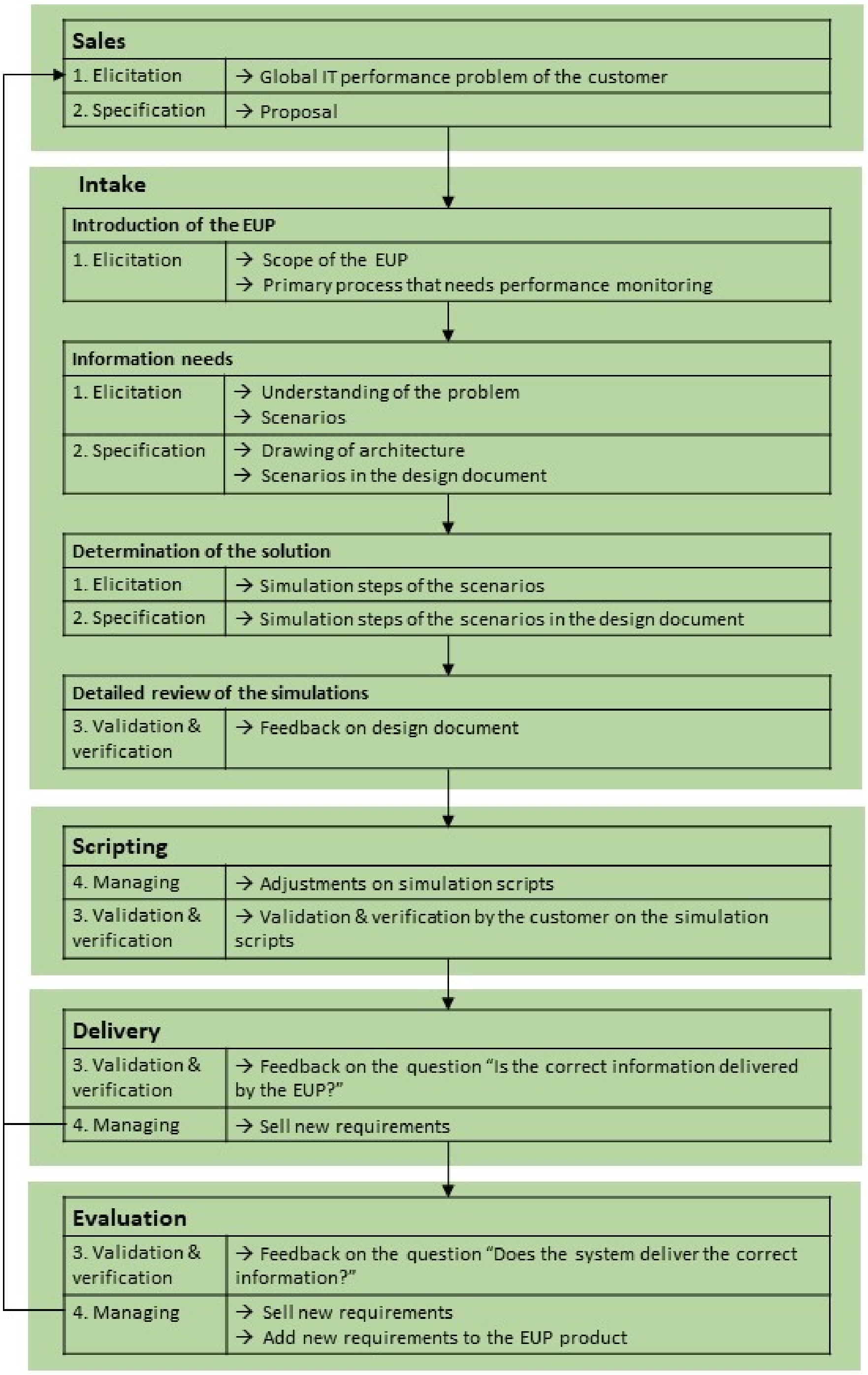
3.2 CURRENT REQUIREMENT ENGINEERING PROCESS PRACTICE AT THE BACKBONE

The current REP practice of The Backbone for their EUP monitoring solution is found by the author of this thesis using a triangulation research consisting of interviews, observations, and documentary research. Users validated the current REP practice as found by the author. In Figure 2, the current REP practice of The Backbone is summarised.

As can be seen, the current REP practice is divided into five phases: *sales*, *intake*, *scripting*, *delivery*, and *evaluation*. Each phase has its own purpose. The *intake* phase is divided into four different subphases. In Figure 2, the arrows indicate that the results of the previous (sub)phase are the input for the next (sub)phase. It is also possible to go back to a previous (sub)phase if it appears that more information from that previous (sub)phase is needed to continue the REP. The five phases of the REP of The Backbone are described in detail in the subsections below.

3.2.1 *Sales*

The first phase of the REP is *sales*. The purpose of this phase is to sell the EUP to a potential customer. The REP of The Backbone starts therefore at the account manager. The account manager calls organisations with the aim to inform them about the possibilities and be-



Sales	
1. Elicitation	→ Global IT performance problem of the customer
2. Specification	→ Proposal

Intake

Introduction of the EUP	
1. Elicitation	→ Scope of the EUP → Primary process that needs performance monitoring

Information needs	
1. Elicitation	→ Understanding of the problem → Scenarios
2. Specification	→ Drawing of architecture → Scenarios in the design document

Determination of the solution	
1. Elicitation	→ Simulation steps of the scenarios
2. Specification	→ Simulation steps of the scenarios in the design document

Detailed review of the simulations	
3. Validation & verification	→ Feedback on design document

Scripting	
4. Managing	→ Adjustments on simulation scripts
3. Validation & verification	→ Validation & verification by the customer on the simulation scripts

Delivery	
3. Validation & verification	→ Feedback on the question "Is the correct information delivered by the EUP?"
4. Managing	→ Sell new requirements

Evaluation	
3. Validation & verification	→ Feedback on the question "Does the system deliver the correct information?"
4. Managing	→ Sell new requirements → Add new requirements to the EUP product

Figure 2: Requirements engineering process of The Backbone

nefits of the EUP. The account manager asks if the organisation he calls, experiences performance issues or is interested in measuring performances of applications. If this is the case, the organisation is a potential customer, and the account manager informs the potential customer on how the EUP could help in their particular situation.

The account manager tries to schedule a meeting with the potential customer. If this succeeds, the account manager gives a presentation to show the potential customer what the EUP is, where the EUP can be used for, and what its possible applications are. The account manager also asks for the needs of the potential customer. If the potential customer is satisfied with the possibilities of the EUP, the account manager sends a proposal with a task description and a price. If the potential customer agrees with the proposal, he signs it and the EUP is sold to The Backbone's new customer.

3.2.2 *Intake*

When the EUP is sold by an account manager in the *sales* phase, the performance consultant starts the *intake* phase. During this phase, most of the customers' requirements are mapped. The *intake* phase is divided into four subphases: *introduction of the EUP*, *information needs*, *determination of the solution*, and *detailed review of the simulations*.

3.2.2.1 *Introduction of the EUP*

The performance consultant starts the *intake* phase with contacting the contact person of the organisation to make an appointment for a meeting. This meeting most often takes place at the customer's office. The first question the performance consultant always asks, is: "Do you know what the EUP is?" This question is important since the contact person is not always the person who bought the EUP, or it may be that any of the attendants of the meeting does not have enough knowledge on the EUP to start the *intake* process with. If this is the case, the performance consultant explains what the product does (measuring the end-user experience based on performance) and what the possibilities of the product are (monitoring, doing trend analysis, checking if the organisations meets their SLAs with their users, and checking if the software suppliers of the organisation meets their SLAs).

The next question the performance consultant asks, is for the organisation to think about the applications they want to monitor and why. The performance consultant gives suggestions to the contact person if the contact person cannot think of something himself. Suggestions are done by asking about the primary processes of the organisation. For example if an organisation sells products, the performance consultant asks about their customer relationship management process. The performance consultant asks about the applications used for this process and the relationships of these applications and gives suggestions for monitoring on these applications and relationships. The customer needs some time to think about the applications and relationships he wants to monitor. Therefore, a new meeting is scheduled for at least one week later to do the actual intake. If needed, the performance consultant explains the purpose and possibilities of the EUP again.

When a customer already exactly knows what he wants to monitor, this process is a lot faster.

3.2.2.2 *Information needs*

At this moment of the *intake* meeting, the customer has thought about the applications and relationships between applications that need monitoring. Now, the performance consultant asks a lot of open questions like: what is the problem of the customer, what do they experience with that problem, which components are involved, how are these components connected to each other, which functionalities of the applications do they use, how often do they use those functionalities, when do the end-users experience problems with the applications? Depending of the scope the customer bought the EUP for (troubleshooting, authentication, or SLA measuring) the above questions are translated to the right scope. If the scope is SLA measuring, the performance consultant also asks for copies of the SLAs that need monitoring. Another very important question the performance consultant asks in this meeting is from which locations the applications are used. This information is needed to decide where watcher nodes need to be located.

During the *intake* meeting, the performance consultant draws the architecture of the components and their relationships, to gather insight on how the processes run through the IT infrastructure and how the internal information flow looks like. The performance consultant always checks his architecture drawing with the customer.

Based on the architecture drawing, the performance consultant and customer invent simulations for the EUP. These simulations do not need to be in detail, like: push this button now, but the steps that need to be taken need to be clear, like: start the email application, write an email, filling in the receiving email address, send the email, receive the email, open the email, and finally close the email application. All the components talked about earlier in the process, need to come back in the simulations. The scenarios are based on the answers the customer gave to the open questions asked earlier in the process. The performance consultant asks deeper into the open questions. This phase of the meeting is like a brainstorm session about the design and simulations with the customer and the performance consultant. The result of this meeting is the scenarios that need to be made, their priority, and the consistency between the scenarios. This result needs to be documented in a so-called design document. The structure of this design document can be found in Appendix C.

In this phase of the meeting, the performance consultant also asks about which notifications need to be sent in the case that a simulation does not run successfully. Important here is to set a threshold in milliseconds for 'warning' and 'critical' notifications. The performance consultant also needs to know to whom the notifications needs to be sent and the corresponding email address or phone number.

The dashboard is also discussed in this phase of the meeting. The performance consultant wants to know if the customer is going to use the dashboard for trend analysis or real-time information. This information is needed to choose a dashboard application. Furthermore, the performance consultants asks about the information that needs to be visible on the dashboard.

3.2.2.3 *Determination of the solution*

When the information needed is clear to the performance consultant, the performance consultant and customer start with the *determination of the solution*.

An important step in the *determination of the solution* is to create accounts for the EUP with the rights to carry out the actions of the simulations. This account also needs the rights to get the data needed for the simulations. This can be hard to achieve. For example, if a DigiD (Dutch Digital Identification) is needed, because test DigiD accounts do not exist.

Now, the performance consultant walks through the simulations step-by-step. This is a very time consuming process since the performance consultant needs to focus on every possible little detail. An example of a small detail is: does the simulation needs to press the Enter key on the keyboard or click the OK button on the screen? All these steps are documented in a table with three columns: the step, the action (click on, fill in a user name, close the window), and the check (very detailed description of the outcome when the action is carried out in the right way, for example: a certain window or piece of text appears).

The design document (Appendix C), including the table for the simulations, is detailed in such a level that another performance consultant is able to make the scripts without needing more information.

3.2.2.4 *Detailed review of the simulations*

Another benefit of this detailed design document (Appendix C) is that employees of the customer that have not been in contact with The Backbone yet, can also check the requirements and simulations. This is also why the performance consultant of The Backbone asks the customer for feedback and confirmation: is the design document correct and complete?

3.2.3 *Scripting*

After the *intake* phase is finished and feedback on the document is processed and the customer confirmed the document, the actual scripting takes place. This happens in the *scripting* phase.

Simple scripts are sometimes already scripted during the *intake* meeting. The scripting is done at the location of the customer. Preferably, an employee of the customer is around to answer questions. It happens often that during the scripting process, the performance consultant finds out that it is more convenient to adjust simulations a little bit, even with a very detailed simulation table. This can be because of new insights, or because the script technology can handle the simulation better if some parts are adjusted. During this scripting process, verification of the customer is also important. This gives the customer also the opportunity to change the requirements if needed. If simulations are adjusted, or requirements change, the design document (Appendix C) needs to be updated. The scripting process of one simulation takes one up to four working days, depending on the complexity and uniqueness of the simulation. A simulation for email is already scripted multiple times, therefore this simulation takes less time than a simulation that is uncommon.

3.2.4 *Delivery*

When all the simulations are scripted, the EUP gets delivered to the customer in the *delivery* phase.

The performance consultant trains the customer in how the information should be interpreted. The performance consultant also checks with the customer whether the correct information is delivered by the system and if the customer wants more information. This can lead to selling new simulations or functionalities to the customer. Reports, for example, is one of the functionalities customers often want when the simulations are already running. For reports it is important to know which information should be in the report and where the report must be stored or sent to.

3.2.5 *Evaluation*

A few months after the EUP is delivered to the customer, the *evaluation* phase begins. In this phase, the EUP as a product, the implementation of the EUP at the customer is evaluated, as well as how the customer works with the EUP and the results the EUP produced in the first months.

An evaluation meeting is planned to evaluate the performance of the EUP and to discuss the results the scripts of the EUP gave about the performance of monitored applications. Present at this meeting are the account manager and the performance consultant of The Backbone, both responsible for this particular customer, and representatives of the customer. This meeting takes place at the customer's location.

During this meeting, new requirements for the EUP implementation at the customer, as well as for the EUP in general, can come up. The account manager tries to sell more scripts, trainings, dashboards, and other services The Backbone offers. If this succeeds, the REP starts all over again.

4 RESEARCH DESIGN

This chapter describes the research methodology and research process used in this thesis to find answers on the research questions (RQs). The research methodology describes how to design a new artefact, in this thesis a new REP. The research process describes the structure of this research.

4.1 RESEARCH METHODOLOGY

The goal of this research is to design a artefact in the form of a new REP for The Backbone which is more time-efficient and leads to requirements of a better quality than their current REP practice. Therefore, this research makes use of design science, as design science is used to create an explicitly applicable solution to a problem [31].

The design science research methodology (DSRM) of Peffers et al. [31] is used for designing and evaluating the new REP for the EUP monitoring solution at The Backbone. This DSRM is used because this methodology is developed especially for design science research in the field of information systems. In addition, the DSRM takes the practical context of this study into account. Therefore it fits well with the goal of this research.

The DSRM is an iterative process as can be seen in Figure 3. During this study, the new REP is first designed and then twice improved, resulting in REP v.0, REP v.1, and REP v.2. The DSRM includes the following six steps [31]:

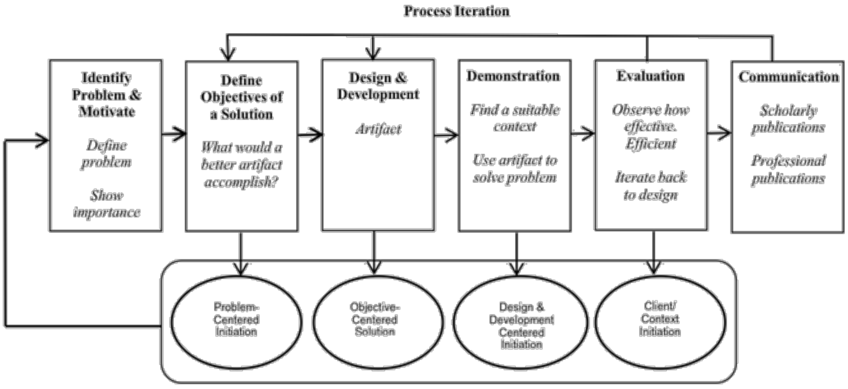

Process Iteration
Identify Problem & Motivate
Define problem
Show importance
Define Objectives of a Solution
What would a better artifact accomplish?
Design & Development
Artifact
Demonstration
Find a suitable context
Use artifact to solve problem
Evaluation
Observe how effective. Efficient
Iterate back to design
Communication
Scholarly publications
Professional publications
Problem-Centered Initiation
Objective-Centered Solution
Design & Development Centered Initiation
Client/ Context Initiation

Figure 3: Design science research methodology (DSRM) as described by Peffers et al. [31]

1. **Identify problem & motivate:** in this first step, the specific research problem is defined and the importance of a solution is made clear. The problem identification and motivation for this research can be found in chapter 1.

2. **Define objective of a solution:** after the problem identification and motivation, the next step is to define objectives of a solution. The objectives of the solution, in this thesis a better REP for the EUP monitoring solution of The Backbone, are defined by answering the question "What would a better REP accomplish?" The objectives for the three versions of the new REP are described in respectively section 5.1, section 6.1, and section 7.1.

3. **Design and development:** when the objectives of the new REP are clear, the REP gets designed in this step. All elements of the new REP design are elaborated in detail in this step. For the three versions of the REP, the designs are elaborated in respectively section 5.2, Figure 6.2, and section 7.2.

4. **Demonstration:** a demonstration of the new REP is conducted in this demonstration step which takes place after the design and development step. In this demonstration, the new REP is used in a case study which is created based on two real cases. For the evaluation of REP v.0, REP v.1 and REP v.2, the same case study is used for comparability reasons. This case study can be found in Appendix A. The demonstrations of the three versions of the new REP are elaborated in respectively section 5.3, section 6.3, and section 7.3.

5. **Evaluation:** during the evaluation step, observations are done and interviews are held to check the new REP on its effectiveness and efficiency. The evaluations of REP v.0, REP v.1, and REP v.2 can be found in respectively section 5.4, section 6.4, and section 7.4.

6. **Communication:** the last step of DSRM is to communicate the findings of this research. This is done by writing this thesis.

4.2 RESEARCH PROCESS

The research process used in this thesis, is visually summarised in Figure 4 in the style of Verschuren & Doorewaard [39].

Background knowledge on REP structures as well as the current REP practice at The Backbone is necessary before a better REP for the EUP monitoring service at The Backbone can be designed. A REP is better than the current practice of The Backbone if the REP leads to requirements of a better quality in less time. This background knowledge is the result of an earlier study, and can be found in chapter 3 for the purpose of completeness.

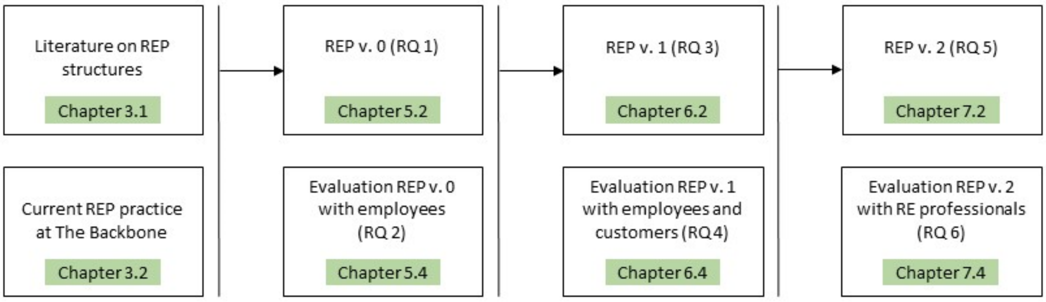

Literature on REP structures
Chapter 3.1
Current REP practice at The Backbone
Chapter 3.2
REP v. 0 (RQ 1)
Chapter 5.2
Evaluation REP v. 0 with employees (RQ 2)
Chapter 5.4
REP v. 1 (RQ 3)
Chapter 6.2
Evaluation REP v. 1 with employees and customers (RQ 4)
Chapter 6.4
REP v. 2 (RQ 5)
Chapter 7.2
Evaluation REP v. 2 with RE professionals (RQ 6)
Chapter 7.4

Figure 4: Research process

With this background knowledge, objectives for the first version of the new REP (REP v.0) are described, after which REP v.0 is designed, demonstrated, and evaluated in chapter 5. The DSRM of Peffers et al. [31] is used for this purposes. Since REP v.0 is the first version of a new design, its evaluation consists of an interview with the performance consultant to find the biggest teething problems before using the precious time of customers.

The evaluation of REP v.0 has led to improvements which are implemented in REP v.1 (chapter 6). This version of the REP is evaluated by observations where REP v.1 is used at a customer and afterwards interviews are held with the performance consultant and the customer.

New insights have been gathered from the evaluation of REP v.1 and these insights are used to improve the REP design. This led to the design of REP v.2, which can be found in chapter 7. This final version of the REP is evaluated by interviews with professionals in RE from other software suppliers as input for a discussion on the validity and generalisability of this REP. The discussion can be found in chapter 8.

5 REP V.0

This chapter describes the work done to answer RQ1 and RQ2 by designing and evaluating REP v.0. Before REP v.0 gets designed, its objectives are described first. Next, all elements in the design of REP v.0 are elaborated into detail. After REP v.0 is designed, a demonstration of REP v.0 is conducted via a case study. Finally, the REP gets evaluated and learnings from the demonstration and evaluation for REP v.1 are described.

5.1 OBJECTIVES OF REP V.0

As already mentioned in section 4.1, the objectives for the new REP for the EUP monitoring solution of The Backbone are the answers to the question "What would a better REP accomplish?"

First, there are objectives defined by The Backbone. These objectives apply to all three versions of the new REP. The objectives defined by The Backbone are:

- O-1: The new REP should be more time-efficient than the current REP practice at The Backbone.
- O-2: The new REP should lead to less deficient requirements than the current REP practice at The Backbone.
- O-3: The new REP should lead to a better implementation of the EUP than the current REP practice at The Backbone.
- O-4: The new REP should lead to a more professional image with customers than the current REP practice at The Backbone.
- O-5: The new REP should be visible for customers.
- O-6: The new REP should leave room for the performance consultants to use their own knowledge.
- O-7: The new REP should make it possible for another performance consultant to take over the project from the initial performance consultant in case of emergency.

Second, there is one other objective that applies to all three versions of the new REP. This objective came from an earlier literature study and is summarised in section 3.1 for completeness purposes. The objective found in literature is:

- O-8: The new REP should contain all four phases of which a REP consists of according to literature (section 3.1): *elicitation*, *specification*, *validation & verification*, and *managing*.

Third, there are objectives specific to REP v.0. These objectives consist of solutions to time inefficiencies and documentation inefficiencies found by the author of this thesis in the current REP practice for the EUP monitoring solution at The Backbone. These solutions help at the same time to achieve objectives O-1 to O-8. The current REP practice at The Backbone is elaborated in section 3.2.

The first time-inefficiency in the current REP of The Backbone is that sometimes a new meeting needs to be scheduled because the customer has not thought about applications and simulations before the first meeting. This costs the performance consultant a lot of time which is unnecessary, since he has to visit the company twice with the same goal. A possible solution for this inefficiency could be to somehow enforce the customer to think about applications and simulations for performance monitoring beforehand. This solution would lead to more time-efficiency since the performance consultant does not have to visit the customer twice with the same goal(O-1), a more professional image towards the customer since this would show the customer that The Backbone has experience with the implementation of the EUP and therefore knows what information they need (O-4), a REP that is visible for the customer since the customer has to participate in it by thinking about applications and simulations(O-5), and a REP that clearly contains an *elicitation* phase since a lot of information is asked in the beginning of the project (O-8).

Another time-inefficiency found in the current REP is that if the EUP is bought for SLA measuring, the performance consultant asks for copies of the SLAs that need monitoring. This only happens in the *information needs* subphase of the *intake* phase. It would be more time-efficient if the customer sends the SLAs that need monitoring as soon as possible to the performance consultant, or at least has copies of the SLAs ready for inspection at the meeting. This solution leads to more time-efficiency since the performance consultant does not have to wait during the meeting for the customer to find the needed SLAs or, worse, has to come back (O-1), a more professional image towards the customer by asking the customer to prepare himself for the meeting (O-4), and a REP that is visible for the customer since the customer has to prepare himself for the meeting (O-5).

Time-inefficiency also happens due the fact that the performance consultant draws the architecture of the applications, IT infrastructure, and their relationships during the *intake* meeting in the *information needs* subphase. This costs time and is also more error sensitive than if the customer would have prepared this information in advance of the *intake* meeting. This solution leads to more time-efficiency since the performance consultant does not have to wait during the meeting for the customer to prepare the IT architecture or, worse, has to come back (O-1), a better implementation of the EUP since the customer has time to check and think about their IT architecture beforehand (O-3), a more professional image towards the customer by asking the customer to prepare himself for the meeting (O-4), and a REP that is visible for the customer since the customer has to prepare himself for the meeting (O-5).

The current REP is also inefficient on the topic of documentation. In section 3.2, it is described that all components discussed earlier in the process do need to come back in the simulations. To be sure of this, all components talked about need to be documented. However, this documentation is not present in their current REP. A checklist would help to document those components and check whether all components are processed in the simulations. This solution would lead to less deficient requirements since it is checked whether all components are present in the simulations (O-2), a better implementation of the EUP since all components that are discussed during the meeting are taken into account (O-3), the possibility for the performance consultant to use his own knowledge since it is his task to fill in the components this solution only offers a structured way of documenting the components (O-5), the possibility for another performance consultant to take over the project in case of emergency since the components talked about are documented in a structured way (O-7), and the REP having a clear *specification* phase since components are documented in a structured way (O-8).

The last documentation-inefficiency is about the *evaluation* phase. During the evaluation meeting, the customer can propose improvements for the EUP product and agreements are made between The Backbone and the customer. During the meeting, field notes are made by the performance consultant and account manager of The Backbone. However, these field notes are not properly elaborated into a readable document and sent to the customer to sign, even when agreements have been made. As a result, some agreements are never gotten back to, which can lead to confidence and reputation loss, but also to less sales. This solution leads to a better implementation of the EUP since feedback is documented and this can be used to improve the EUP product (O-3), a more professional image with customers since they will get the feeling that the evaluation meeting is taken seriously (O-4), and a REP with clear validation & verification and managing phases since it is asked during the evaluation meeting whether the EUP delivers the correct information and The Backbone has the opportunity to sell new requirements and find new requirements for the EUP product that gets documented as well (O-8).

Solutions to the above presented inefficiencies form the objectives specific to REP v.0. These objectives are:

- O-9: The new REP should encourage the customer to think about applications and simulations that need performance monitoring before the first meeting with the performance consultant.

- O-10: The new REP should encourage the customer to collect copies of the SLAs that need performance monitoring before the first meeting with the performance consultant.

- O-11: The new REP should encourage the customer to prepare the architecture needed for the implementation of the EUP before the first meeting with the performance consultant.

- O-12: The new REP should provide a checklist for the performance consultant to check whether all components talked about in the intake meeting are processed in the simulations.

- O-13: The new REP should encourage the performance consultant to elaborate his field notes of the evaluation meeting.

5.2 DESIGN OF REP V.0

Based on the objectives described in section 5.1, the current REP practice at The Backbone for their EUP monitoring solution is adjusted. These adjustments to the current REP resulted in REP v.0. For REP v.0, changes have been made in Appendix C, these changes are processed in Appendix Ftogether with the changes made in the design document in chapter 6 and chapter 7. The current REP practice at The Backbone is elaborated in section 3.2 and summarised in Figure 1.

The current REP is adjusted on the following elements:

- (Sub)phases are renamed into terms that cover the content of the (sub)phase better.

- The content of the (sub)phases is based on the four phases a REP should contain according to literature as described in section 3.1 (O-8).

- A questionnaire is added to the REP. This questionnaire encourages the customer to think of applications and simulations that need performance measuring before the performance consultant visits them (O-9). The questionnaire also encourages the customer to think of their IT architecture (O-11) and to collect copies needed SLAs for the EUP (O-10) before the performance consultant visits them.

- Together with the questionnaire, the website link of the EUP product is also sent to the customer to give them the information they need before filling in the questionnaire.

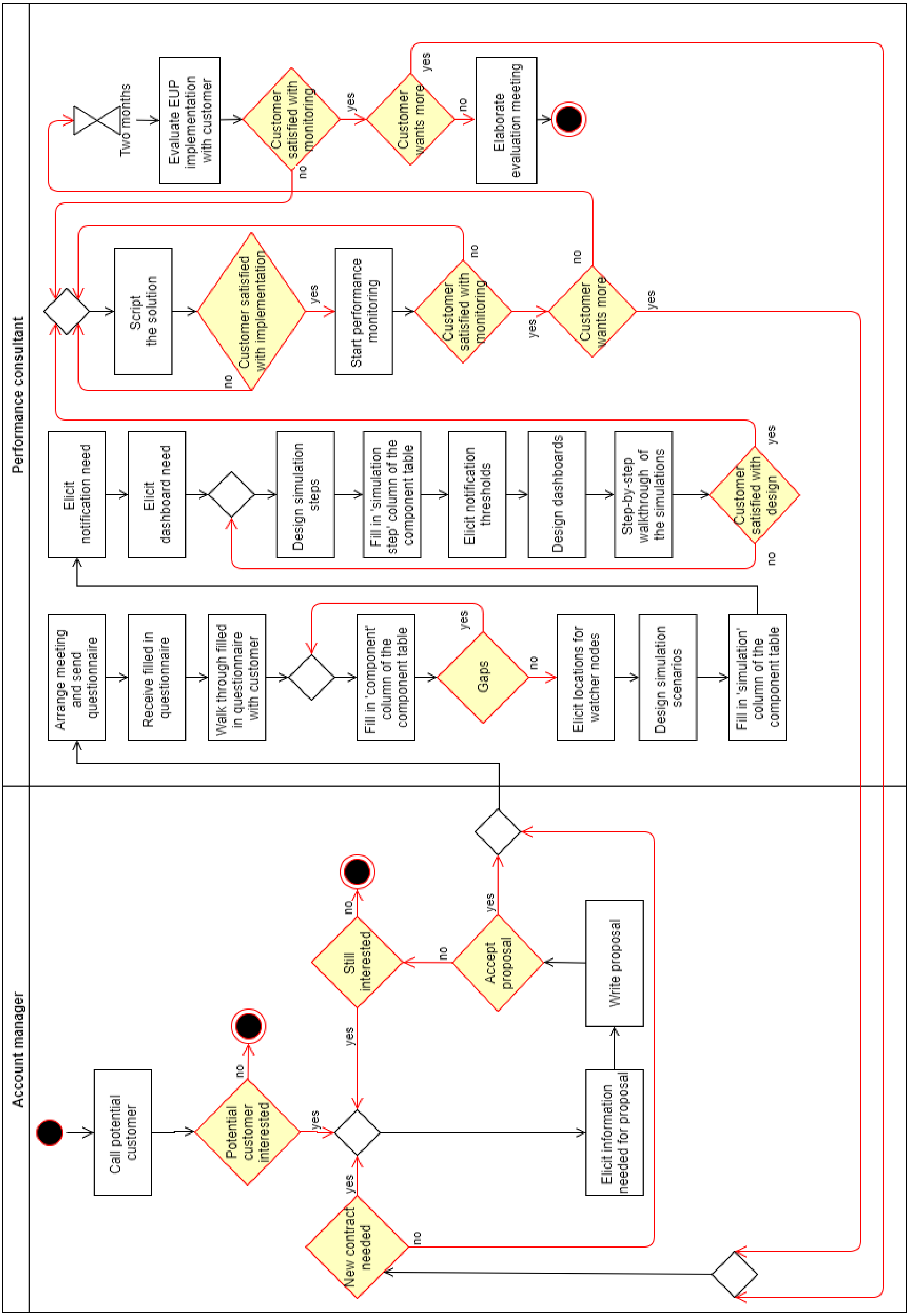
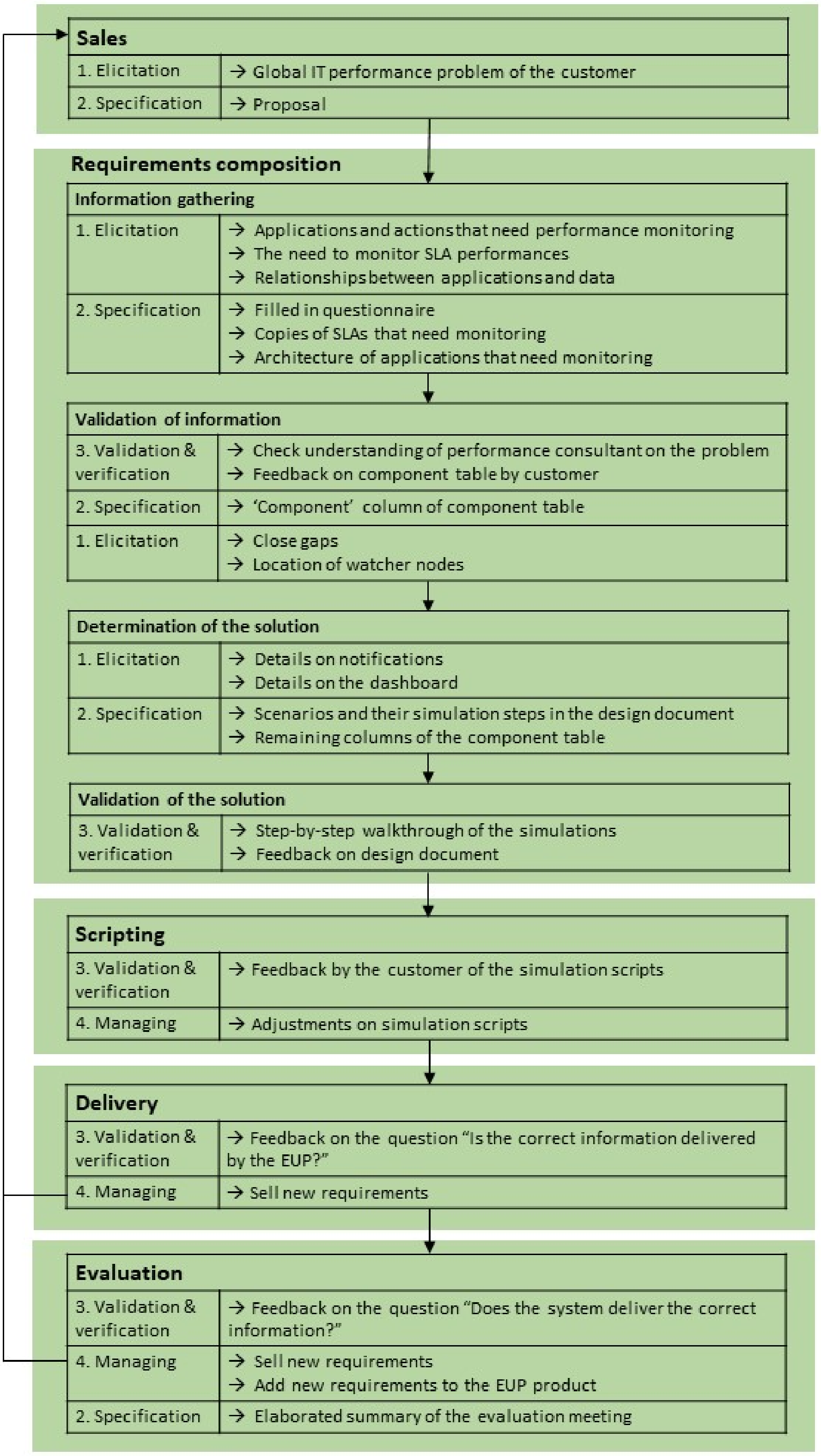

Account manager
Performance consultant
Call potential customer
Potential customer interested
yes
no
New contract needed
yes
no
Elicit information needed for proposal
Write proposal
Accept proposal
yes
no
Still interested
yes
no
Arrange meeting and send questionnaire
Receive filled in questionnaire
Walk through filled in questionnaire with customer
Fill in 'component' column of the component table
Gaps
yes
no
Elicit locations for watcher nodes
Design simulation scenarios
Fill in 'simulation' column of the component table
Elicit notification need
Elicit dashboard need
Design simulation steps
Fill in 'simulation step' column of the component table
Elicit notification thresholds
Design dashboards
Step-by-step walkthrough of the simulations
Customer satisfied with design
yes
no
Script the solution
Customer satisfied with implementation
yes
no
Start performance monitoring
Customer satisfied with monitoring
yes
no
Customer wants more
yes
no
Two months
Evaluate EUP implementation with customer
Customer satisfied with monitoring
yes
no
Customer wants more
yes
no
Elaborate evaluation meeting

Figure 5: Activity diagram REP v.0



Sales

1. Elicitation	→ Global IT performance problem of the customer
2. Specification	→ Proposal

Requirements composition

Information gathering

1. Elicitation	→ Applications and actions that need performance monitoring → The need to monitor SLA performances → Relationships between applications and data
2. Specification	→ Filled in questionnaire → Copies of SLAs that need monitoring → Architecture of applications that need monitoring

Validation of information

3. Validation & verification	→ Check understanding of performance consultant on the problem → Feedback on component table by customer
2. Specification	→ 'Component' column of component table
1. Elicitation	→ Close gaps → Location of watcher nodes

Determination of the solution

1. Elicitation	→ Details on notifications → Details on the dashboard
2. Specification	→ Scenarios and their simulation steps in the design document → Remaining columns of the component table

Validation of the solution

3. Validation & verification	→ Step-by-step walkthrough of the simulations → Feedback on design document

Scripting

3. Validation & verification	→ Feedback by the customer of the simulation scripts
4. Managing	→ Adjustments on simulation scripts

Delivery

3. Validation & verification	→ Feedback on the question "Is the correct information delivered by the EUP?"
4. Managing	→ Sell new requirements

Evaluation

3. Validation & verification	→ Feedback on the question "Does the system deliver the correct information?"
4. Managing	→ Sell new requirements → Add new requirements to the EUP product
2. Specification	→ Elaborated summary of the evaluation meeting

Figure 6: Summary REP v.0

- A table is added to the so-called design document (Appendix F) which functions as a checklist to check whether all components talked about during the process are in the simulations (O-12).
- In REP v.0, the performance consultant and account manager of The Backbone need to elaborate their field notes of the evaluation meeting (O-13).

The adjusted REP forms REP v.0, and is described into detail below. Figure 5 shows the activity diagram of REP v.0 in which the activities are grouped per actor (the account manager and the performance consultant). REP v.0 is divided into five phases based on the phases of the current REP practice at The Backbone: *sales*, *requirements composition*, *scripting*, *delivery*, and *evaluation*. Each phase has its own purpose. The *requirements composition* phase is divided into four different subphases. REP v.0 is summarised in Figure 6. In this figure, the arrows indicate that the results of the previous (sub)phase are the input for the next (sub)phase. The five phases of REP v.0 are described in detail in the subsections below.

5.2.1 *Sales*

The first phase of the REP is *sales*. The purpose of this phase is to sell the EUP.

REP v.0 starts therefore by the account manager. The account manager tries to sell the EUP to potential customers by asking if they experience IT performance problems or if they are interested in SLA monitoring or monitoring for incident management. In such a case, the account manager sketches them a solution with the EUP product of The Backbone. If the potential customer is interested in buying the EUP, the account manager needs to know the IT performance problem of the customer. It is preferred to gather this information during a sales meeting since a meeting leads to more information than a phone call does. This information is needed for the account manager to make the proposal as accurate as possible. Gathering information on the IT performance problem of the customer belongs to the *elicitation* phase of Figure 1. The account manager does not have to know the IT performance problem in very much detail, but he must have gathered enough information to make a decent proposal. If the customer accepts the proposal, the EUP is sold and the next phase starts. The proposal belongs to the *specification* phase of Figure 1, since the global IT performance problem is documented in the proposal.

Since not every sales meeting leads to a sale, the performance consultant is normally not involved in this process. However, if it is already clear that the potential customer is really interested or the potential customer indicates that technical staff is joining the meeting, the account manager can bring the performance consultant to the sales meeting for bringing in more technical knowledge to this sales meeting.

5.2.2 *Requirements composition*

After the EUP is sold by an account manager, the requirements for the EUP simulations need to be clear before the design and implementation of the EUP can start. This takes place in the *requirements composition* phase. To have the requirements for the EUP simulations clear, the performance consultant needs consecutive to: gather information on the IT performance problem of the customer, validate his understanding of the IT performance problem with the customer, think of a solution for this IT performance problem, and validate this solution with the customer. Based on this, the next subphases need to be carried out by the performance consultant: *information gathering*, *validation of the information*, *determination of the solution*, *validation of the solution*.

5.2.2.1 *Information gathering*

The first thing the performance consultant has to do is to gather a lot of information on the IT performance problem the customer experiences. To let the customer already think of applications and simulations they want to have measured before the performance consultant goes to meet the customer, a questionnaire is sent. This questionnaire can be found in Appendix B. Along with the questionnaire, an information package is also sent to the customer. This information package prevents that the performance consultant goes to the customer just to tell them again what the EUP is. The questionnaire helps the customer to already think of the applications and simulations they want to monitor the performance of and therefore prevents that the performance consultant has come to the customer for no reason. In this questionnaire, the customer is also asked to send copies of SLAs if they want to monitor the performances of SLAs. Finally, the customer is also asked to draw an architecture of the applications, functions of the applications, and needed information for those applications. In this way, the performance consultant can prepare himself optimally before visiting the customer and the customer has thought about their problem as well. All information gathered in this *information gathering* subphase, belongs to the requirements *elicitation* phase in Figure 1, but the documents that conduct all this information belong to the *specification* phase as represented in Figure 1.

5.2.2.2 *Validation of requirements information*

After the performance consultant received all the information and has prepared himself, there is a solution meeting between the per-

Table 1: Component table REP v.0

#	COMPONENT	SIMULATION	SIMULATION STEP
1	Login on website	Website	Step 11
2			
3			
4			
5			

formance consultant and the customer. The goal of this meeting is to validate the performance consultant's understanding of the IT performance problem of the customer completely, to design the solution, and to validate this solution.

The validation of the performance consultant's understanding of the IT performance problem of the customer is done by walking through all the information given to the performance consultant together with the customer. The validation of the performance consultant's understanding belongs to the *validation & verification* phase of Figure 1. All components coming up in the meeting are put into the component table (Table 1).

At this point of time, only the 'component' column needs to be filled in. Later, every component gets a number for reference reasons which can become useful later in the process and for communication purposes. Filling in the 'component' column of Table 1 belongs to the *specification* phase of the REP of Figure 1. At the end of the meeting, the component table gets validated by the customer to be sure nothing is missing. This belongs to the *validation & verification* phase of Figure 1. It is also possible that the performance consultant observes gabs in the IT performance problem, or the information handed to him. These gaps can be filled during the meeting and therefore new requirements can also get elicited.

One question the performance consultant should never forget to ask is from which locations the applications that get their performance measured are used. This information is needed to discuss and decide where watcher nodes need to be located. Eliciting requirements to fill gaps and the locations of which applications are used belong to the *elicitation* phase of Figure 1.

5.2.2.3 *Determination of the solution*

At this point in the process, the performance consultant has a complete understanding of the IT performance problem of the customer and therefore can start working on the solution. The performance consultant invents simulations for the EUP together with the customer.

This is one meeting with the *validation of requirements information* subphase.

First, the simulation scenarios need to be invented. This means that there is no need to invent the simulations directly into much detail, like: push this button. However, the steps that need to be taken need to be clear, like: start the email application, write an email, filling in the receiving email address, send the email, receive the email, open the email, and finally close the email application. The simulations get documented in the design document (Appendix F). All the components talked about earlier in the process and written down in the component table (Table 1) need to come back in the simulations. For referring, the 'simulation' column of Table 1 needs to be filled at this moment. The scenarios, their priority, and the consistency between the scenarios must be clear before designing the scenarios in detail.

In this phase of the meeting, the performance consultant also asks about which notifications need to be sent in the case that a simulation does not run successfully. Important here is to set a threshold in milliseconds for 'warning' and 'critical' notifications. The performance consultant also needs to know to whom the notifications needs to be sent and the corresponding email address or phone number.

The dashboard is also discussed in this phase of the meeting. The performance consultant wants to know if the customer is going to use the dashboard for trend analysis or real-time information. This information is needed to choose a dashboard application. Furthermore, the performance consultants asks about the information that needs to be visible on the dashboard.

When the simulations are roughly designed, the notifications are clear, and the functionalities of the dashboard are clear, the simulations can be further designed into detail. All simulation steps need to be described very clearly into the document as shown in Appendix C. Accordingly, the component table (Table 1) needs to be updated by filling the 'simulation step' column.

With the design of the detailed simulations, it must be considered that some simulations need accounts that give the EUP the rights to carry out the actions of the simulations. This can be hard to achieve. For example, if a DigiD (Dutch Digital Identification) is needed, because test DigiD accounts do not exist.

Documenting the simulations in the design document (Appendix F) and to fill in the remaining columns of the component table (Table 1) belong to the *specification* phase of Figure 1. Gathering the requirements concerning notifications and dashboards belong to the *elicitation* phase of Figure 1.

5.2.2.4 *Validation of the solution*

Before the scripting can start, the designed solution for the customer needs validated. The performance consultant walks through the simulations step-by-step. This is a very time consuming process since the performance consultant needs to focus on every possible little detail. An example of a small detail is: does the simulation needs to press the Enter key on the keyboard or click the OK button on the screen? All these steps are documented in a table with three columns: the step, the action (click on, fill in a user name, close the window), and the check (very detailed description of the outcome when the action is carried out in the right way, for example: a certain window or piece of text appears). This check belongs to the *validation & verification* phase of Figure 1. Since a lot of questions can arise to the performance consultant during this check, it is preferred for this subphase to occur during the same meeting as the *validation of requirements information* and *determination of the solution* subphase.

The design document (Appendix F), including the table for the simulations, is detailed in such a level that another performance consultant is able to make the scripts without needing more information.

Another benefit of this detailed design document is that employees of the customer that have not been in contact with The Backbone yet, can also check the requirements and simulations. This is also why the performance consultant of The Backbone asks the customer for feedback and confirmation: is the design document correct and complete? This belongs to the *validation & verification* phase of Figure 1.

5.2.3 *Scripting*

After feedback on the document is processed and the customer confirmed the document, the actual scripting takes place. This happens in the *scripting* phase.

Simple scripts are sometimes already scripted during the *requirements composition* meeting. The scripting is done at the location of the customer. Preferable, an employee of the customer is around to answer questions. During this scripting process, the need to adjust or to change requirements can arise, mainly due to technical challenges. If simulations are adjusted, or requirements change, the design document (Appendix F) needs to be updated. This belongs to the *management* phase of Figure 1. The verification step, of the *validation & verification* phase of Figure 1, of the customer is also important during this phase.

5.2.4 *Delivery*

When all the simulations are scripted, the EUP gets delivered to the customer in the *delivery* phase.

The performance consultant trains the customer in how the data that the EUP generates should be interpreted. The performance consultant also checks with the customer if the correct information is delivered by the system and if the customer wants more information, which belongs to the *validation & verification* phase in Figure 1. This can lead to selling new simulations or functionalities to the customer, this belongs to the *managing* phase of Figure 1.

5.2.5 *Evaluation*

A few months after the EUP is delivered to the customer, the *evaluation* phase begins. In this phase, the EUP as a product, the implementation of the EUP at the customer is evaluated, as well as how the customer works with the EUP and the results the EUP produced in the first months.

An evaluation meeting is planned to evaluate the performance of the EUP and to discuss the results the scripts of the EUP gave about the performance of monitored applications. This belongs to the *validation & verification* phase of Figure 1. Present at this meeting are the account manager and the performance consultant of The Backbone, both responsible for this particular customer, and representatives of the customer. This meeting takes place at the customer's location.

During this meeting, new requirements for the EUP implementation at the customer, as well as for the EUP in general, can come up for discussion. The account manager tries to sell more scripts, trainings, dashboards, and other services The Backbone offers. If this succeed, the REP starts all over again. Selling more products or services that The Backbone offers belongs to the *managing* phase of Figure 1.

The account manager and performance consultant take field notes during the evaluation meeting. Afterwards, they elaborate them into a structured summary. This summary is sent to the customer to sign since agreements can be made during the evaluation meeting. This part of the *evaluation* phase belongs to the *specification* phase of Figure 1.

5.3 DEMONSTRATION OF REP V.0

REP v.0 is demonstrated below by the use of a case study in the company. The organisation name and the applications used in the case study are anonymised to guarantee the privacy of the real organisations. The case study can be found in Appendix A. This case study is a combination of two real cases The Backbone had in the past.

The first case concerns a municipality in The Netherlands. The account manager of The Backbone tried to sell the EUP to the municipality while they did not experience performance issues and no budget was left for non-urgent IT projects. However, a few weeks later end-users were complaining about two applications that were performing too slow. Both applications run on a Service Based Computing (SBC) system. The municipality had no insight into the extent of the problem and therefore wanted to gather insights on which end-users were experiencing the performance problems, which times the applications were slow, and the root cause of the performance problem for example a certain action. Therefore, they contacted The Backbone by themselves and asked for a proposal. Due to the problems experienced at the municipality, the performance consultant of The Backbone implemented the EUP three working days after the municipality agreed on the proposal of The Backbone. There were two meetings needed to implement three scripts. The municipality had an architectural drawing ready to be used during the second meeting. With the use of the EUP, the municipality was able to find the root cause of the performance problem of both applications. With this information, the municipality fixed the performance problem of both applications.

The second case is a company in the industry sector for whom the performance of the Customer Relationship Management (CRM) application is crucial. The CRM application is linked with their Order Management System and their Document Management System. The CRM application is used by hundreds of end-users at the same time every day. Most end-users use a SBC system to have access to the CRM application. End-users are complaining that the CRM application is being slow. However, since the CRM application is linked to multiple systems, the IT department of the company wants to know whether the CRM application or one of the linked systems is being slow. The account manager of The Backbone sold the EUP to the company. This company was already a customer of The Backbone for two years and already had purchased multiple projects and products of The Backbone. To implement the EUP, the performance consultant needed to visit the company twice. The first time for tuning the EUP to the wishes of the company, the second time to review the solution made by the performance consultant. The solution consisted of the definition of five scripts, of which two are actually implemented. The company was prepared for the meetings with the performance consultant by already thinking about the steps needed in the scripts. In the end, the company could not solve their performance problem with the use of the EUP. However, the application manager of the company received other information as well that helped him to solve the problem anyway.

For the case study, information from both these cases is used. From the first case, the urgency is used as well as being a new customer for The Backbone. This information is used, since it becomes more likely to attract new customers for The Backbone since their current customers already know of the EUP.

From the second case, the CRM application is used. This part is used since it is a very important application in most companies. Besides that, a CRM application is mostly linked to a couple of other applications and is accessed by end-users via a SBC. This access via a SBC happens a lot as can also be seen in the first case.

5.3.1 *Sales*

ABC calls an account manager of The Backbone because ABC experiences IT performance issues now and wants to know whether the EUP can help them tracing the problem or not. The account manager explains ABC how the EUP could help tracing the performance problem. Since the ABC wants the performance issue to be solved as soon as possible, they do not feel the need for the account manager to come over for a meeting to give a presentation about the EUP. Instead, ABC wants to receive a proposal as soon as possible, also to compare the costs of the EUP to those of other solutions.

The proposal gets accepted by ABC and therefore the EUP is sold.

5.3.2 *Requirements composition*

Directly after the proposal is accepted by ABC, the performance consultant of The Backbone contacts ABC to arrange a meeting with them. The performance consultant directly starts the information gathering subphase at this moment at time.

5.3.2.1 *Information gathering*

To start this subphase, the performance consultant of The Backbone sends the questionnaire and the information package of the EUP to ABC.

The next day, The Backbone receives the ABC representative's answers to the questionnaire (Appendix B):

1. **Why did you purchase the EUP?**

   ABC: We purchased the EUP because our end-users are complaining about our CRM application being slow. This CRM system is just implemented in our organisation, but this application is used hundreds times each day and is very important for the continuity of our business. We want to know whether the CRM application is slow or one of the linked applications does not perform well.

2. **Which applications would you like to monitor? (Examples are: Outlook, Hix, Website: http://www.thebackbone.nl)**

   ABC: The CRM application. But also our Order Management System (OMS) and our Enterprise Resource Planning (ERP) application since those are linked to the CRM application.

3. **Of which actions in these applications do you want to monitor the performance?**

   ABC: From the CRM application we want to monitor the performance of starting the CRM application, the logging in performance, the performance of adding new data to the CRM application, and the performance of searching through the CRM application. From the OMS and ERP applications, we want to monitor an action in CRM that needs to get its data from the OMS and ERP applications.

4. **Do you want to monitor SLAs? If so, please add them as an attachment.**

   ABC: Yes, we want to monitor the SLA of the CRM application.

5. **How are the applications that need to be monitored linked? (What is the architecture like, where does the required data come from, etc.?)**

   The answer to this question is skipped because it has no added value for this research.

5.3.2.2 *Validation of requirements information*

Since the IT performance issues ABC experiences at the moment do have a lot of impact on the organisation, the performance consult of The Backbone meets ABC three days after ABC answered the questionnaire.

Together with ABC, the performance consultant walks through all the information ABC has given him so far. The performance consultant asks a lot of questions to the ABC representative to get a complete understanding. During this meeting, the performance consultant fills in the 'component' column of Table 2. At the end of the meeting, the ABC representative is asked to check whether all components talked about are represented in this table.

The performance consultant asks the ABC representative from which locations end-users approach the CRM, OMS, and ERP applications. The representative answers that end-users use those applications from desktops in the ABC office as well as from their business laptops. The performance consultant asks if those laptops are also used outside the office of ABC, the representative tells him that this is often the case. Therefore the watcher nodes are going to be on a laptop located in the locked cabinet on the working floor with a good Wi-Fi signal and on a desktop with a LAN internet connection.

5.3.2.3 *Determination of the solution*

Now, the simulations are determined by filling in the 'simulation' column of Table 2.

Table 2: Component table demonstration REP v.0

#	Component	Simulation	Simulation step
1	Start CRM application	CRM intern data	CRM_start
2	Login CRM application	CRM intern data	CRM_login
3	Add data to CRM application	CRM intern data	CRM_add_data
4	Search in CRM application	CRM intern data	CRM_search_data
5	CRM needs data from OMS	CRM OMS	
6	CRM needs data from ERP	CRM ERP	
7	Close CRM application	CRM intern data	CRM_logoff
			CRM_close

Next, the performance consultant asks for notification requirements. For every ‘warning’ an email must be sent to the ABC representative, for every ‘critical’ an email and a SMS must be sent to the ABC representative.

The ABC representative and the performance consultant now discuss the dashboards. They agree on one dashboard to start with, which is going to be used for trend analysis. This decision is based on the fact that for every warning or critical an email gets sent to the ABC representative.

The following step is that Table 3 of Appendix C is filled in for the simulation *CRM intern data*. The tables for the simulations on the components *CRM OMS* and *CRM ERP* are not made for the purpose of this demonstration.

After the simulation steps have been designed, the ‘simulation step’ column of Table 2 gets updated and in Table 4 from the design document (Appendix F) the thresholds for ‘critical’ and ‘warning’ notifications are set.

After the ABC representative creates a CRM account for the EUP, this phase is finished.

5.3.2.4 *Validation of the solution*

The performance consultant and the ABC representative both check if the simulation steps are formulated well and if nothing is missing. Everything is good, therefore the *scripting* phase can start.

Table 3: Simulation steps REP v.0

Step ID	Action	Check
CRM_start	Open the CRM application	Text "ABC User Dashboard"
CRM_login	Fill in username and password	Text "Welcome, Marie."
CRM_add_data	Click on: "Add customer"	Text "Customer name is:"
	Give customer the following name: "12345678"	Text "12345678"
	Click on: "Add"	Text "Customer 12345678 is added to the database"
CRM_search_data	Search for customer with name: "12345678". Press Enter	Text "12345678"
CRM_delete_data	Select "Customer name: 12345678". Press Delete	Text "12345678" cannot be found
CRM_logoff	Click on: "Log off"	Text "Are you sure you want to log off?"
	Click "Yes"	Text "You're logged off"
CRM_close	Click "X" in upper right corner	Text "Recycle bin" on desktop

Table 4: Notification thresholds REP v.0

Step ID	Warning	Critical
CRM_start	20000	25000
CRM_login	18000	23000
CRM_add_data	7000	10000
CRM_search_data	10000	12000
CRM_delete_data	15000	18000
CRM_logoff	4000	6000
CRM_close	4000	6000
CRM_intern_data_total	80000	100000

5.3.3 *Scripting*

The *CRM_intern_data* script is not very complicated and therefore gets scripted directly after the simulation steps are checked. The ABC representative stays close to the performance consultant to answer possible questions. The dashboard is later built in the office of The Backbone.

5.3.4 *Delivery*

After the scripts and the dashboards are finished, the performance consultant goes the ABC office for another meeting with the ABC representative. The performance consultant installs the scripts on the watcher nodes and gives the ABC representative access to the dashboard.

Accordingly, the performance consultant explains to the ABC representative how the dashboard should be read and how the results can be analysed using the dashboard.

Also the first results are explained by the performance consultant to the ABC representative.

5.3.5 *Evaluation*

Two months after the *delivery* phase, the performance consultant has another meeting with the ABC representative, but now the account manager also joins them.

The EUP works well and the problem is found, but the ABC representative wants to have a weekly report on the notifications of last week. Agreements on this are made during the meeting.

Afterwards, the performance consultant and the account manager of The Backbone create a structured summary and send it to ABC to sign it.

5.4 EVALUATION OF REP V.0

REP v.0 was evaluated by a walk-through of the REP and an interview with the performance consultant of The Backbone. This evaluation led to learnings that are used in chapter 6 to design REP v.1. The evaluation was carried out in Dutch, since this is the native language of both the author of this thesis and the performance consultant. The evaluation was voice recorded. Field notes have also been made and were elaborated directly after the evaluation. Accordingly, feedback was given on the elaborated version of the field notes by the performance consultant. It was made clear to the performance consultant that the goal of this evaluation is to receive as much feedback as possible. Therefore, the performance consultant was encouraged to ask questions during the evaluation and to not save them until the end of the evaluation. The evaluation was divided into two parts. In the first part, the author of this thesis and the performance consultant walked through REP v.0 together. In the second part, the author of this thesis conducted an interview with the performance consultant. The interview questions in Dutch and in English can be found in Appendix D.

5.4.1 *Walk-through REP v.0*

During the walk-through, the performance consultant and the author of this thesis discussed REP v.0 part-by-part. The performance consultant gave his opinion about all (sub)phases. Below, the (sub)phases are presented, along with improvements that are found during the walk-through part of the evaluation:

5.4.1.1 *Sales*

The EUP does not necessarily have to be used in case of an 'IT performance problem'. The EUP is all about the 'performance need of IT services from the end-user perspective'.

5.4.1.2 *Requirements composition*

Requirements composition is a logical name for this phase.

INFORMATION GATHERING Instead of an information package in the form of a document, the link to a website can be sent to the customer. This saves sending documents to the customer and a website is also more up to date than extra documentation. 'IT services' is a better term to use in this case than 'applications', because for the implementation of some services, multiple applications are needed.

VALIDATION OF REQUIREMENTS INFORMATION The component table (Table 1) is useful to check whether all components are actual used in the simulations.

DETERMINATION OF THE SOLUTION The elicitation of the simulations and the simulation steps are missing in this subphase. However, those are two different components in the *elicitation* phase. First the simulations are elicited, then the notification and dashboard needs, and only then the steps of the simulations. A good addition to the process would be to add a table in which the notification and dashboard needs can be documented and can be checked when the needs are implemented in the notifications or dashboards. *Solution design* is a better term for the functionality of this subphase than *'determination of the solution'* since the solution gets designed in this subphase.

Questionnaire: Asking multiple persons of the customer to fill in the questionnaire, leads to a completer image of the different needs. Filling in the questionnaire should not take longer than 10 minutes for one person. To prevent getting a lot of documents for each project, this questionnaire could be an attachment in the design document (Appendix F).

Question 1: This question needs to be reformulated. It is already described in the contract why the customer bought the EUP. However, this question is used for the customer to tell us his needs in his own words. Different persons have different needs and this information is needed for a complete view of the performance needs of the customer. The question needs probably a short introduction. Therefore an option of this question is: "Your goal of the EUP is mentioned in the contract. However, could you please tell us your needs for the EUP using your own words?"

Question 2: "Applications and/or services" is a better term for "applications" since not only applications need monitoring, but services do need monitoring as well. Other examples of applications and/or services are: Citrix, ERP, and CRM applications.

Question 3: The following question is clearer:

- "Of which of the commonly used activities by end-users do you want to monitor its performance? and/or
- Of which activities and/or actions where you experience performance issues, do you want to monitor their performance? and/or
- Of which of the activities that are appointed in a Service Level Agreement (SLA) do you want to monitor the performance?"

Question 4: Not every organisation wants to send their SLAs. A part of this question is also answered in the new formulation for question 3. Therefore, this question can be rephrased as follows: "If you are going to use the End-User Performance Monitor for monitoring SLAs, please sent the relevant SLAs as an attachment or have them ready before the meeting."

Question 5: The technical components related to the service, like hardware and storage, so also belong to the architecture and therefore should be named in the question.

VALIDATION OF THE SOLUTION This phase seems to be completely logical.

5.4.1.3 *Scripting*

Implementing is a better term for this phase than *scripting* is, since this phase is more than just scripting. Making dashboards is, for example, also a component of this phase.

5.4.1.4 *Delivery*

The term *delivery* is misplaced for this phase. *Delivery* sounds if the process is finished after this phase, while in practice the process just started. First data gets gathered, after which it needs to be analysed. Mostly these analyses lead to new needs, after which the whole REP starts over. Therefore the term *start performance monitoring* is a better name for this phase.

5.4.1.5 *Evaluation*

The actual time between the end of the *delivery* phase and the start of the *evaluation* phase with the evaluation meeting is nowadays too long. Therefore, this time may be reduced to one month. Not every organisation wants to sign a document like the one generated after the evaluation meeting. On the other side, some organisations find it more professional to sign such a document instead of giving their confirmation by email. Therefore it is better to leave the decision whether to ask an organisation to sign it or to get a confirmation by email to the performance consultant.

5.4.1.6 *Iterations*

Whenever there are new requirements sold in the *delivery* or *evaluation* phase, REP v.0 always returns to the *sales* phase. This is mainly to check the contract if the customer already paid for extra requirements like the new ones, or if a new contract is needed. In the case that the customer already paid for the extra requirements, the *sales* phase takes way less time than normal.

5.4.2 *Interview*

This interview took place directly after the walk-through. The author of this thesis interviewed the performance consultant of The Backbone about his opinion about REP v.0. Like the walk-through, the interview took place in Dutch since this is the mother language of both the author of this thesis and the performance consultant. The questions that are asked during this interview can be found in Appendix D.

The performance consultant thinks that REP v.0 is better than the current REP practice of The Backbone in all aspects, like: usability, usefulness, efficiency, quality of written requirements, implementation quality of requirements, and professional image (Appendix D). However, the additions done in the process like the questionnaire (Appendix B) must be integrated into the design document (Appendix F) before it can really be used. Also a list of actions may lead to a better usability of REP v.0. The reasons why REP v.0 are an improvement of the current REP practice at The Backbone are:

- REP v.0 gives structure and is therefore also less ad hoc. This structure gives the customer the feeling of dealing with a professional and provides also more realistic expectations towards the customer. Besides, structure in a process indicates experience and therefore leads to more confidence by the customer. Structure increases the portability when another performance consultant needs to take over the project and also gives guidance to do all steps needed.

- By using REP v.0, the information needed for the project to succeed is complete in the beginning of the project and also gets verified by the customer. This creates a base to work with.

- REP v.0 leads to better requirements, since the requirements are written down and also verified by the customer before implementing them.

- REP v.0 probably leads to more requirements than there is time to implement those requirements at the first contract. This may lead to new contracts and therefore more profit for The Backbone.

5.5 CONCLUSIONS OF REP V.0

This chapter's most important conclusion is that REP v.0 is an improvement in respect to the current REP practice at The Backbone. REP v.0 is considered by the performance consultant to perform better on aspects like: usability, usefulness, efficiency, quality of written requirements, implementation quality of requirements, and professional image than the current REP practice at The Backbone. Besides that, all objectives O-1 until O-13 are processed into the design of REP v.0.

However, before the REP can be used by The Backbone, some aspects need more improvement:

- The terminology should be changed on some points to make the REP clearer for the user. Sometimes, the terminology changes are small, like: rephrasing "application" into "IT services". Sometimes, these terminology changes have a bigger impact, like rephrasing all the questions of the questionnaire.

- In the *determination of the solution* subphase, the elicitation of the simulations and the simulations steps should be added.

- New checklists for the notification and dashboard needs need to be added to the design document (Appendix F).

- The evaluation phase should start after one month after the EUP is implemented instead of two months.

These improvements are the input for the objectives in chapter 6.

6 REP V.1

This chapter describes the work done to answer RQ3 and RQ4 by designing and evaluating REP v.1. As in chapter 5, the objectives of REP v.1 are described before REP v.1 gets designed. The design is elaborated into detail, after which a demonstration of REP v.1 is conducted via a case study. Lastly, the REP gets evaluated and learnings from the demonstration and evaluation for REP v.2 are described.

6.1 OBJECTIVES OF REP V.1

For REP v.1, there are two types of objectives: objectives that apply to all three versions of the new REP and objectives specific to REP v.1.

The objectives that apply to all three versions of the new REP are objectives O-1 up to and including O-8 which are already described in section 5.1.

The objectives specific to REP v.1 are improvements on REP v.0. Those improvements have been found during the evaluation of REP v.0 (section 5.4) and are summarised below.

Before the REP can be used by The Backbone, some aspects need more improvement.

First of all, the terminology needs improvement. Changes to the terminology make the REP clearer for the user. Sometimes, the terminology changes are small, like: rephrasing "application" into "IT services". Sometimes, these terminology changes have a bigger impact, like rephrasing all the questions of the questionnaire.

Second, to complete the REP, the elicitation of the simulations and the simulations steps during the *determination of the solution* subphase should be added to the REP.

Third, the design document (Appendix C) needs an update. New checklists for the notification and dashboard needs need to be added to the design document.

Lastly, the evaluation phase should start after one month instead of two months after the EUP is implemented.

The improvements are summarised below:

- O-14: The new REP should use terminology that is clear for the user.
- O-15: The new REP should be complete, no steps missing in the process.
- O-16: The new REP should contain an updated version of the design document as showed in Appendix F.

- O-17: The new REP should have an evaluation phase that starts one month after the EUP is implemented.

6.2 DESIGN OF REP V.1

REP v.1 is an improved version of REP v.0 (chapter 5), therefore parts of REP v.1 are the same as in REP v.0. For REP v.1, changes have been made in Appendix C, these changes are processed in Appendix Ftogether with the changes made in the design document in chapter 5 and chapter 7. The improvements consist of the objectives given in section 6.1.

Figure 7 shows the activity diagram of REP v.1 in which the activities are grouped per actor (the account manager and the performance consultant). REP v.1 is divided into five phases: *sales*, *requirements composition*, *implementing*, *start performance monitoring*, and *evaluation*. Each of these phases has its own purpose. Four subphases form together the *requirements composition* phase. Figure 8 summarises REP v.1. The arrows in this figure indicate that the results of the previous (sub)phase are the input for the next (sub)phase. The five phases of REP v.1 are described in detail in the subsections below.

6.2.1 *Sales*

The first phase of the REP is *sales*. The purpose of this phase is to sell the EUP. REP v.1 starts therefore by the account manager. The account manager tries to sell the EUP to potential customers by asking if they experience IT performance problems or if they have a performance need of IT services from the end-user perspective, for example SLA monitoring or monitoring for incident management. In such a case, the account manager sketches them a solution with the EUP product of The Backbone. If the potential customer is interested in buying the EUP, the account manager needs to know the performance need of IT services from the end-user perspective of the customer. It is preferred to gather this information during a sales meeting since a meeting leads to more information than a phone call does. This information is needed for the account manager to make the proposal as accurate as possible. Gathering information on the performance need of IT services from the end-user perspective of the customer belongs to the *elicitation* phase of Figure 1. The account manager does not have to know the IT performance need in very much detail, but he must have gathered enough information to make a decent proposal. If the customer accepts the proposal, the EUP is sold and the next phase starts. The proposal belongs to the *specification* phase of Figure 1, since the global IT performance need is documented in the proposal.

Since not every sales meeting leads to a sale, the performance consultant is normally not involved in this process. However, if it is

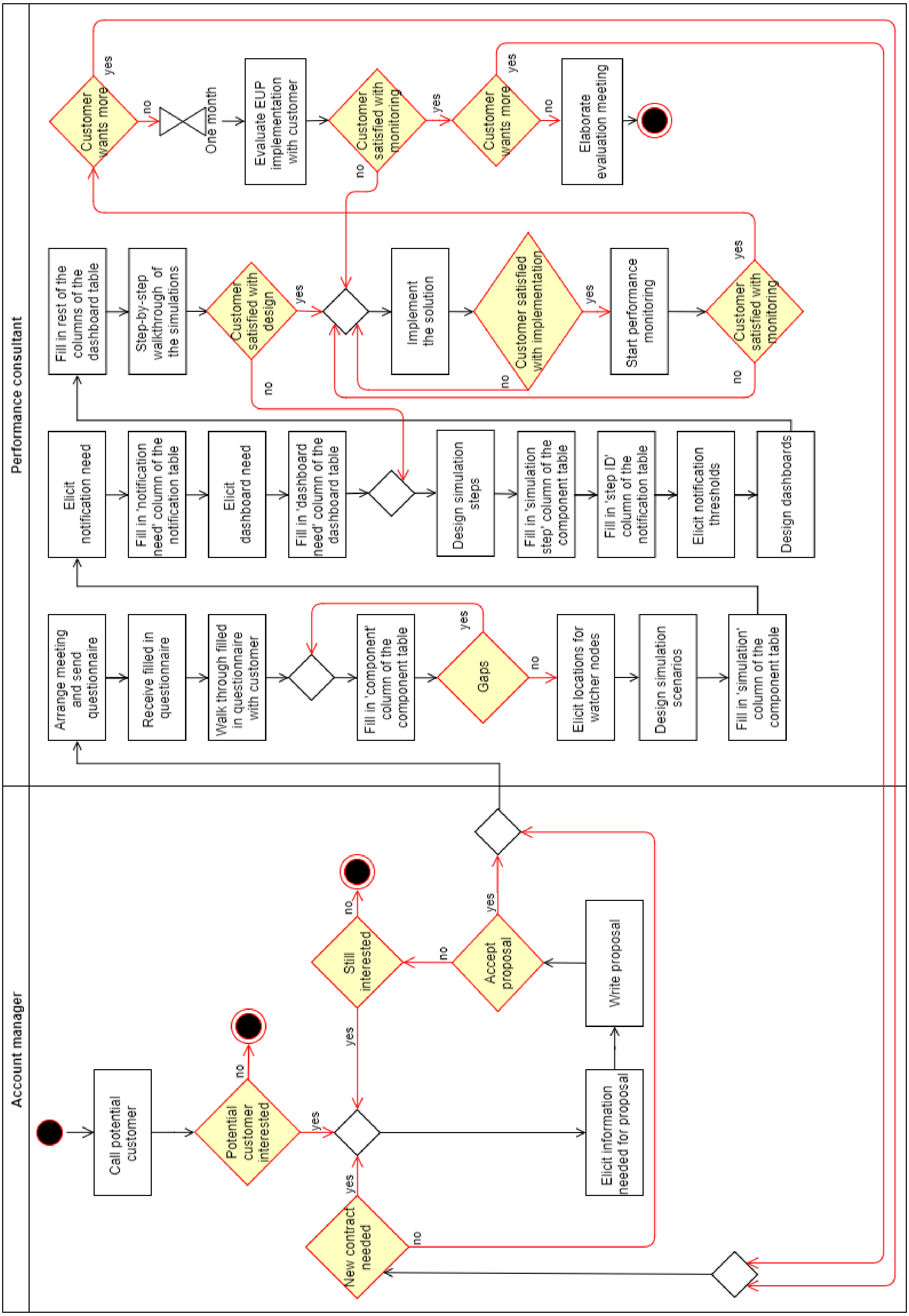

Account manager
Performance consultant
Call potential customer
Potential customer interested
no
yes
New contract needed
yes
no
Elicit information needed for proposal
Write proposal
Accept proposal
yes
no
Still interested
yes
no
Arrange meeting and send questionnaire
Receive filled in questionnaire
Walk through filled in questionnaire with customer
Fill in 'component' column of the component table
Gaps
yes
no
Elicit locations for watcher nodes
Design simulation scenarios
Fill in 'simulation' column of the component table
Elicit notification need
Fill in 'notification need' column of the notification table
Elicit dashboard need
Fill in 'dashboard need' column of the dashboard table
Design simulation steps
Fill in 'simulation step' column of the component table
Fill in 'step ID' column of the notification table
Elicit notification thresholds
Design dashboards
Fill in rest of the columns of the dashboard table
Step-by-step walkthrough of the simulations
Customer satisfied with design
yes
no
Implement the solution
Customer satisfied with implementation
yes
no
Start performance monitoring
Customer satisfied with monitoring
yes
no
Customer wants more
yes
no
One month
Evaluate EUP implementation with customer
Customer satisfied with monitoring
yes
no
Customer wants more
yes
no
Elaborate evaluation meeting

Figure 7: Activity diagram REP v.1

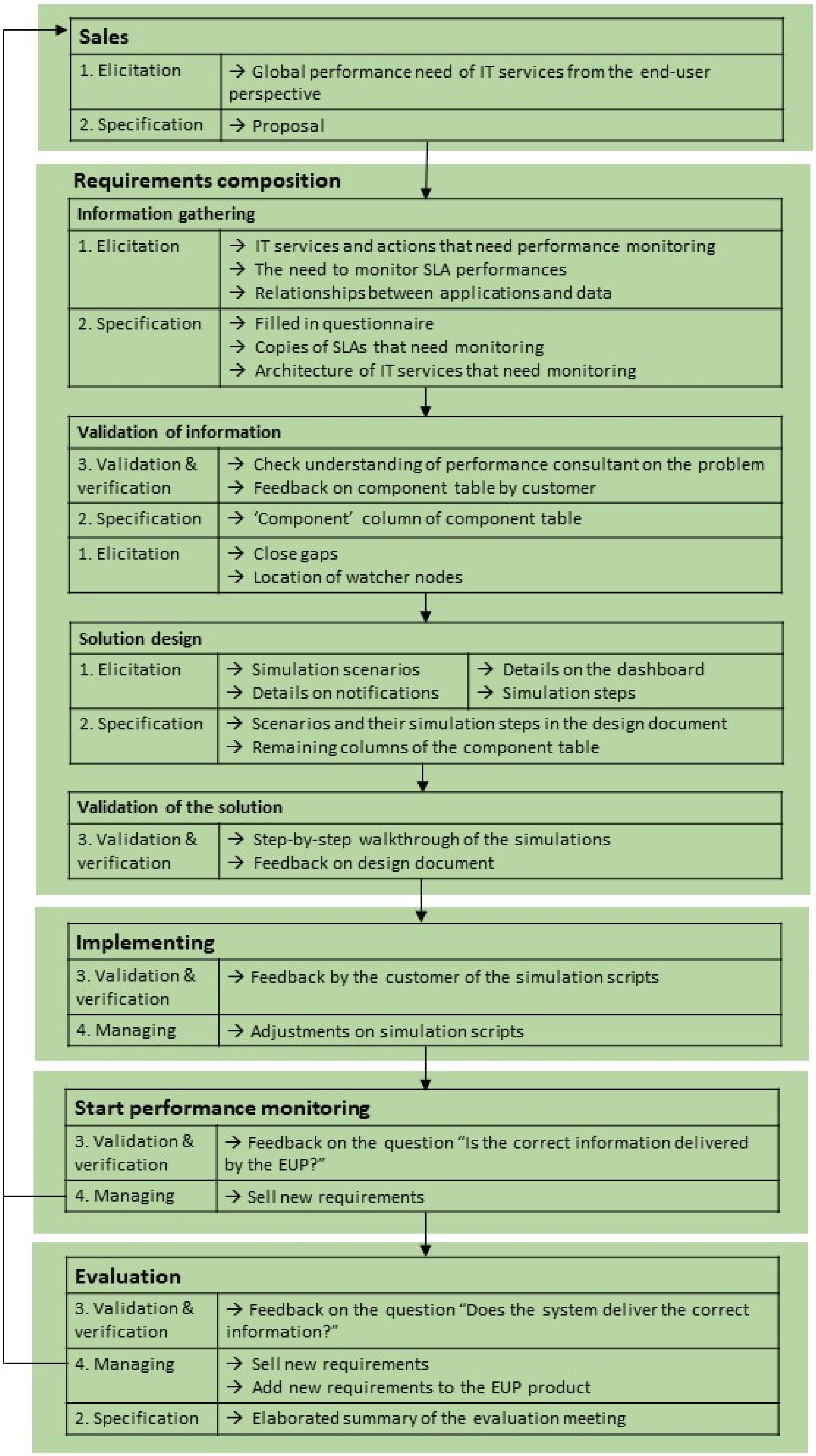

Sales
1. Elicitation
→ Global performance need of IT services from the end-user perspective
2. Specification
→ Proposal
Requirements composition
Information gathering
1. Elicitation
→ IT services and actions that need performance monitoring
→ The need to monitor SLA performances
→ Relationships between applications and data
2. Specification
→ Filled in questionnaire
→ Copies of SLAs that need monitoring
→ Architecture of IT services that need monitoring
Validation of information
3. Validation & verification
→ Check understanding of performance consultant on the problem
→ Feedback on component table by customer
2. Specification
→ 'Component' column of component table
1. Elicitation
→ Close gaps
→ Location of watcher nodes
Solution design
1. Elicitation
→ Simulation scenarios
→ Details on notifications
→ Details on the dashboard
→ Simulation steps
2. Specification
→ Scenarios and their simulation steps in the design document
→ Remaining columns of the component table
Validation of the solution
3. Validation & verification
→ Step-by-step walkthrough of the simulations
→ Feedback on design document
Implementing
3. Validation & verification
→ Feedback by the customer of the simulation scripts
4. Managing
→ Adjustments on simulation scripts
Start performance monitoring
3. Validation & verification
→ Feedback on the question "Is the correct information delivered by the EUP?"
4. Managing
→ Sell new requirements
Evaluation
3. Validation & verification
→ Feedback on the question "Does the system deliver the correct information?"
4. Managing
→ Sell new requirements
→ Add new requirements to the EUP product
2. Specification
→ Elaborated summary of the evaluation meeting

Figure 8: Summary REP v.1

already clear that the potential customer is really interested or the potential customer indicates that technical staff is joining the meeting, the account manager can bring the performance consultant to the sales meeting for bringing in more technical knowledge to this sales meeting.

Whenever new requirements are sold in the *start performance monitoring* or *evaluation* phase, the *sales* phase can takes less time than it takes for a new EUP project. The *sales* phase is in that case needed to check whether the new requirements fit in the current contract or if a new contract is needed. If no new contract is needed, the *sales* phase is finished and the *requirements composition* phase starts. If a new contract is needed, the *sales* phase is also faster since the information needed for an accurate proposal is already gathered in the *start performance monitoring* or *evaluation* phase.

6.2.2 *Requirements composition*

After the EUP is sold by an account manager or when new requirements are sold to an existing customer, the requirements for the EUP simulations need to be clear before the design and implementation of the EUP can start. This takes place in the *requirements composition* phase. To have the requirements for the EUP simulations clear, the performance consultant needs consecutively to: gather information on the performance need of IT services from the end-user perspective of the customer, validate his understanding of this IT performance need with the customer, think of a solution for this IT performance problem, and validate this solution with the customer. Based on this, the next subphases need to be carried out by the performance consultant: *information gathering*, *validation of the information*, *solution design*, *validation of the solution*.

6.2.2.1 *Information gathering*

The first thing the performance consultant has to do is to gather a lot of information on the performance need of IT services from the end-user perspective of the customer. To let the customer already think of IT services and simulations they want to have measured before the performance consultant goes to meet the customer, a questionnaire is sent. This questionnaire is an attachment in the design document (Appendix F) but can also be found in Appendix E. Multiple employees of the customer who have to work with the EUP are asked to fill this questionnaire. Along with the questionnaire, a website link to the EUP product website is also sent to the customer. This website contains all information a customer needs for the EUP. This website link prevents that the performance consultant goes to the customer just to tell them again what the EUP is. The questionnaire helps the customer to already think of the applications and simulations they

Table 5: Component table REP v.1

#	Component	Simulation	Simulation step
1	Login on website	Website	Step 11
2			
3			
4			
5			

want to monitor the performance of and therefore prevents that the performance consultant has come to the customer for no reason. In this questionnaire, the customer is also asked to send copies of SLAs if they want to monitor the performances of SLAs. Finally, the customer is also asked to draw an architecture of the applications, functions of the IT services, and needed information for those IT services. In this way, the performance consultant can prepare himself optimally before visiting the customer and the customer has thought about their problem as well. All information gathered in this *information gathering* subphase, belongs to the requirements *elicitation* phase in Figure 1, but the documents in which all this information is gathered, belong to the *specification* phase as represented in Figure 1.

6.2.2.2 *Validation of requirements information*

After the performance consultant received all the information and has prepared himself, there is a solution meeting between the performance consultant and the customer. The goal of this meeting is to validate the performance consultant's understanding of the IT performance need of the customer completely, to design the solution, and to validate this solution.

The validation of the performance consultant's understanding of the IT performance need of the customer is done by walking through all the information given to the performance consultant together with the customer. The validation of the performance consultant's understanding belongs to the *validation & verification* phase of Figure 1. All components coming up in the meeting are put into the component table by the performance consultant. The component table is integrated in the design document (Appendix F) and can also be found in Table 5. At this point of time, only the 'component' column needs to be filled in. Later, every component gets a number for reference reasons which can become useful later in the process and for communication purposes. Filling in the 'component' column of the component table belongs to the *specification* phase of the REP of Figure 1. At the end of the meeting, the component table gets validated by the customer to be sure nothing is missing. This belongs to the *validation &*

verification phase of Figure 1. It is also possible that the performance consultant observes gaps in the IT performance need, or the information handed to him. These gaps can be filled during the meeting and therefore new requirements can also get elicited.

One question the performance consultant should never forget to ask is from which locations the applications that get their performance measured are used. This information is needed to discuss and decide where watcher nodes need to be located. Eliciting requirements to fill gaps and the locations of which applications are used belong to the *elicitation* phase of Figure 1.

6.2.2.3 *Solution design*

At this point in the process, the performance consultant has a complete understanding of the IT performance need of the customer and therefore can start working on the solution. The performance consultant invents simulations for the EUP together with the customer. This is one meeting with the *validation of requirements information* subphase.

First, the simulation scenarios need to be invented. This means that there is no need to invent the simulations directly into much detail, like: push this button. However, the steps that need to be taken need to be clear, like: start the email application, write an email, filling in the receiving email address, send the email, receive the email, open the email, and finally close the email application. The simulations get documented in the design document of Appendix C. All the components talked about earlier in the process and written down in the component table (Table 5) need to come back in the simulations. For referring, the 'simulation' column of the component table (Table 5) needs to be filled at this moment. The scenarios, their priority, and the consistency between the scenarios must be clear before designing the scenarios in detail.

In this phase of the meeting, the performance consultant also asks about which notifications need to be sent in the case that a simulation does not run successfully. The notification needs of the customer get described by the performance consultant in the 'information need' column of Table 6. The performance consultant also needs to know to whom the notifications needs have to be sent and the corresponding email address or phone number.

The dashboard is also discussed in this phase of the meeting. The performance consultant wants to know if the customer is going to use the dashboard for trend analysis or real-time information. This information is needed to choose a dashboard application. Furthermore, the performance consultants asks about the information that needs to be visible on the dashboard. This information is gathered in the first row and the 'dashboard need' column of Table 7.

When the simulations are roughly designed, the notification need is clear as are the functionalities of the dashboard, the simulations

Table 6: Notification thresholds REP v.1

Notification need	Step ID	Warning	Critical
Starting CRM	CRM_start	20000	25000
Logging in in CRM	CRM_login	18000	23000
Adding data in CRM	CRM_add_data	7000	10000
Searching in the data in CRM	CRM_search_data	10000	12000
Deleting data from CRM	CRM_delete_data	15000	18000
Logging off in CRM	CRM_logoff	4000	6000
Closing CRM	CRM_close	4000	6000
Total CRM application	CRM_intern_data_total	80000	100000

Table 7: Dashboard needs REP v.1

Goal: Trend analysis / Real-time information		
Dashboard need	screen	table / chart
Trend line for the time that the CRM scenario needs to complete with a variable time frame	Main screen	Chart 1
Percentages of time each step had a 'warning' / 'critical' notification in the last 24 hours	Main screen	Table 1

can be further designed into detail. All simulation steps need to be described very clearly into the document as shown in Appendix C. Accordingly, the component table (Table 5) needs to be updated by filling the 'simulation step' column. Accordingly, the 'step ID' column of Table 6 can be filled by the performance consultant to check whether all notification needs have been integrated in the scenarios. Also, the thresholds in milliseconds for 'warning' and 'critical' notifications needs to be set. This is done in a cooperation with the customer and the performance consultant. Table 7 is also filled in further by the performance consultant while designing the dashboard. By filling in this table, the performance consultant checks whether all desired functionalities are in the dashboard design.

With the design of the detailed simulations, it must be considered that some simulations need accounts that give the EUP the rights to carry out the actions of the simulations. This can be hard to achieve. For example, if a DigiD (Dutch Digital Identification) is needed, because test DigiD accounts do not exist.

Documenting the simulations in the design document and to fill in the remaining columns of the component table belong to the *specification* phase of Figure 1. Gathering the requirements concerning notifications and dashboards belong to the *elicitation* phase of Figure 1.

6.2.2.4 *Validation of the solution*

Before the scripting can start, the designed solution for the customer needs to be validated. The performance consultant walks through the simulations step-by-step. This is a very time consuming process since the performance consultant needs to focus on every possible little detail. An example of a small detail is: does the simulation need to press the Enter key on the keyboard or click the OK button on the screen? All these steps are documented in a table with three columns: the step, the action (click on, fill in a user name, close the window), and the check (very detailed description of the outcome when the action is carried out in the right way, for example: a certain window or piece of text appears). This check belongs to the *validation & verification* phase of Figure 1. Since a lot of questions can arise to the performance consultant during this check, it is preferred for this subphase to occur during the same meeting as the *validation of requirements information* and *solution design* subphase.

The design document, including the table for the simulations, is detailed in such a level that another performance consultant is able to make the scripts without needing more information.

Another benefit of this detailed design document is that employees of the customer that have not been in contact with The Backbone yet, can also check the requirements and simulations. This is also why the performance consultant of The Backbone asks the customer for feedback and confirmation: is the design document correct and complete? This belongs to the *validation & verification* phase of Figure 1.

6.2.3 *Implementing*

After feedback on the document is processed and the customer confirmed the document, the actual scripting takes place. This happens in the *scripting* phase.

Simple scripts are sometimes already scripted during the *requirements composition* meeting. The scripting is done at the location of the customer. Preferable, an employee of the customer is around to answer questions. During this scripting process, the need to adjust or to change requirements can arise, mainly due to technical challenges. If simulations are adjusted, or requirements change, the design document needs to be updated. This belongs to the *management* phase of Figure 1. The verification by the customer, which is a step of the *validation & verification* phase of Figure 1, is also important during this phase.

6.2.4 *Start performance monitoring*

When all the simulations are scripted, the EUP starts with the performance monitoring.

In this *start performance monitoring* phase, the performance consultant also trains the customer in how the data that the EUP generates should be interpreted. The performance consultant also checks with the customer if the correct information is delivered by the system and if the customer wants more information, which belongs to the *validation & verification* phase in Figure 1. This can lead to selling new simulations or functionalities to the customer, which belongs to the *managing* phase of Figure 1.

6.2.5 *Evaluation*

One month after the performance monitoring of the EUP has been started, the *evaluation* phase begins. In this phase, the EUP as a product, the implementation of the EUP at the customer is evaluated, as well as how the customer works with the EUP and the results the EUP produced in the first months.

An evaluation meeting is planned to evaluate the performance of the EUP and to discuss the results the scripts of the EUP gave about the performance of monitored applications. This belongs to the *validation & verification* phase of Figure 1. Present at this meeting are the account manager and the performance consultant of The Backbone, both responsible for this particular customer, and representatives of the customer. This meeting takes place at the customer's location.

During this meeting, new requirements for the EUP implementation at the customer, as well as for the EUP in general, can come up for discussion. The account manager tries to sell more scripts, trainings, dashboards, and other services The Backbone offers. If this succeed, the REP starts all over again at the *sales* phase. Selling more products or services that The Backbone offers belongs to the *managing* phase of Figure 1.

The account manager and performance consultant take field notes during the evaluation meeting. Afterwards, they elaborate them into a structured summary. This summary is sent to the customer for confirmation since agreements can be made during the evaluation meeting. This confirmation can be the signature of the responsible person at the customer, or a confirmation by email. The decision whether to ask the customer to sign the document that is made after the evaluation meeting or to confirm the document by email should be left to the performance consultant. This part of the *evaluation* phase belongs to the *specification* phase of Figure 1.

6.3 DEMONSTRATION OF REP V.1

Below, REP v.1 is demonstrated by the use of a case study. The case study that is used in section 5.3 is reused in this chapter and can be found in Appendix A. The case study is a combination of two real cases The Backbone had in the past and are described in section 5.3. section 5.3 also described how this case study has been established. The organisation name and the applications are anonymised to guarantee the privacy of the real organisations.

6.3.1 *Sales*

ABC calls an account manager of The Backbone because ABC experiences IT performance issues now and wants to know whether the EUP can help them tracing the problem or not. The account manager explains ABC how the EUP could help tracing the performance problem. Since the ABC wants the performance issue to be solved as soon as possible, they do not feel the need for the account manager to come over for a meeting to give a presentation about the EUP. Instead, ABC wants to receive a proposal as soon as possible, also to compare the costs of the EUP to those of other solutions.

The proposal gets accepted by ABC and therefore the EUP is sold.

6.3.2 *Requirements composition*

Directly after the proposal is accepted by ABC, the performance consultant of The Backbone contacts ABC to arrange a meeting with them. The performance consultant directly starts the information gathering subphase at this moment at time.

6.3.2.1 *Information gathering*

To start this subphase, the performance consultant of The Backbone sends the questionnaire and the link to the EUP product website to ABC.

The next day, The Backbone receives answers from one ABC employee to the questionnaire:

1. **Your goal of the EUP is mentioned in the contract. However, could you please tell us your needs for the EUP using your own words?**

   ABC: Our end-users are complaining about our CRM application being slow. This CRM system is just implemented in our organisation, but this application is used hundreds times each day and is very important for the continuity of our business. We want to know whether the CRM application is slow or one of the linked applications does not perform well.

2. **Which applications and/or services would you like to monitor? (Examples are: Outlook, Hix, Citrix, ERP, HRM, Website: http://www.thebackbone.nl)**

   ABC: The CRM application. But also our Order Management System (OMS) and our Enterprise Resource Planning (ERP) application since those are linked to the CRM application.

3. - **Of which of the commonly used activities by end-users do you want to monitor their performance?**

     ABC: From the CRM application we want to monitor the performance of starting the CRM application, the logging in performance, the performance of adding new data to the CRM application, and the performance of searching through the CRM application. From the OMS and ERP applications, we want to monitor an action in CRM that needs to get its data from the OMS and ERP applications.

     and/or

   - **Of which activities and/or actions where you experience performance issues, do you want to monitor their performance?**

     ABC: See the answer on the above question.

     and/or

   - **Of which of the activities that are appointed in a Service Level Agreement (SLA) do you want to monitor the performance?**

ABC: We want to monitor the SLA for our CRM application.

4. **If you are going to use the EUP for monitoring SLAs, please sent the relevant SLAs as an attachment or have them ready before the meeting.**

   ABC: We will have the SLA ready before the meeting.

5. **How are the applications that need to be monitored linked with each other and their hardware and storage? (What is the architecture like, where does the required data come from, etc.?)**

   The answer to this question is skipped because it has no added value for this research.

6.3.2.2 *Validation of requirements information*

Since the IT performance issues ABC experiences at the moment do have a lot of impact on the organisation, the performance consult of The Backbone meets ABC three days after ABC answered the questionnaire.

Together with ABC, the performance consultant walks through all the information ABC has given him so far. The performance consultant asks a lot of questions to the ABC representative to get a complete understanding. During this meeting, the performance consultant fills in the 'component' column of Table 8. At the end of the meeting, the ABC representative is asked to check whether all components talked about are represented in this table.

The performance consultant asks the representative of ABC from which locations end-users approach the CRM, OMS, and ERP applications. The representative answers that end-users use those applications from desktops in the ABC office as well as from their business laptops. The performance consultant asks if those laptops are also used outside the office of ABC, the representative tells him that this is often the case. Therefore the watcher nodes are going to be on a laptop located in the locked cabinet on the working floor with a good Wi-Fi signal and on a desktop with a LAN internet connection.

6.3.2.3 *Solution design*

Now, the simulations are determined by filling in the 'simulation' column of Table 8.

Next, the performance consultant asks for notification requirements. For every 'warning' an email must be sent to the ABC representative, for every 'critical' an email and a SMS must be sent to the ABC representative. The notification needs are described in the 'notification need' column of Table 9.

Table 8: Component table demonstration REP v.1

#	Component	Simulation	Simulation step
1	Start CRM application	CRM intern data	CRM_start
2	Login CRM application	CRM intern data	CRM_login
3	Add data to CRM application	CRM intern data	CRM_add_data
4	Search in CRM application	CRM intern data	CRM_search_data
5	CRM needs data from OMS	CRM OMS	
6	CRM needs data from ERP	CRM ERP	
7	Close CRM application	CRM intern data	CRM_logoff
			CRM_close

Table 9: Notification thresholds demonstration REP v.1

Notification need	Step ID	Warning	Critical
Starting CRM	CRM_start	20000	25000
Logging in in CRM	CRM_login	18000	23000
Adding data in CRM	CRM_add_data	7000	10000
Searching in the data in CRM	CRM_search_data	10000	12000
Deleting data from CRM	CRM_delete_data	15000	18000
Logging off in CRM	CRM_logoff	4000	6000
Closing CRM	CRM_close	4000	6000
Total CRM application	CRM_intern_data_total	80000	100000

Table 10: Dashboard needs demonstration REP v.1

Goal: Trend analysis / Real-time information		
Dashboard		
need	screen	table / chart
Trend line for the time that the CRM scenario needs to complete with a variable time frame	Main screen	Chart 1
Percentages of time each step had a 'warning' / 'critical' notification in the last 24 hours	Main screen	Table 1

The ABC representative and the performance consultant now discuss the dashboards. They agree on one dashboard to start with, which is going to be used for trend analysis. This decision is based on the fact that for every warning or critical an email gets sent to the ABC representative. The 'dashboard need' column of Table 10 gets filled by the performance consultant and the ABC representative.

The following step is that Table 11 of Appendix C is filled in for the simulation *CRM intern data*. The tables for the simulations on the components *CRM OMS* and *CRM ERP* are not made for the purpose of this demonstration.

After the simulation steps have been designed, the 'simulation step' column of Table 8 gets updated and in Table 9 the step IDs of the simulation as well as the thresholds for 'critical' and 'warning' notifications are set.

The dashboard gets designed and the 'dashboard screen' and 'dashboard table / chart' columns of Table 10 are filled by the performance consultant to check whether all dashboard needs are fulfilled by the dashboard design.

After the ABC representative creates a CRM account for the EUP, this phase is finished.

6.3.2.4 *Validation of the solution*

The performance consultant and the ABC representative both check if the simulation steps are formulated well and if nothing is missing. Everything is good, therefore the *implementing* phase can start.

6.3.3 *Implementing*

The *CRM_intern_data* script is not very complicated and therefore gets scripted directly after the simulation steps are checked. The ABC representative stays close to the performance consultant to answer pos-

Table 11: Simulation steps REP v.1

Step ID	Action	Check
CRM_start	Open the CRM application	Text "ABC User Dashboard"
CRM_login	Fill in username and password	Text "Welcome, Marie."
CRM_add_data	Click on: "Add customer"	Text "Customer name is:"
	Give customer the following name: "12345678"	Text "12345678"
	Click on: "Add"	Text "Customer 12345678 is added to the database"
CRM_search_data	Search for customer with name: "12345678". Press Enter	Text "12345678"
CRM_delete_data	Select "Customer name: 12345678". Press Delete	Text "12345678" cannot be found
CRM_logoff	Click on: "Log off"	Text "Are you sure you want to log off?"
	Click "Yes"	Text "You're logged off"
CRM_close	Click "X" in upper right corner	Text "Recycle bin" on desktop

sible questions. The dashboard is later built in the office of The Backbone.

6.3.4 *Start performance monitoring*

After the scripts and the dashboards are finished, the performance consultant goes the ABC office for another meeting with the ABC representative. The performance consultant installs the scripts on the watcher nodes and gives the ABC representative access to the dashboard.

Accordingly, the performance consultant explains to the ABC representative how the dashboard should be read and how the results can be analysed using the dashboard.

Also the first results are explained by the performance consultant to the ABC representative.

6.3.5 *Evaluation*

One month after the *delivery* phase, the performance consultant has another meeting with the ABC representative, but now the account manager also joins them. And in between the performance consultant contacts and/or visits the customer on weekly basis to analyse the data and fine-tune the monitoring.

The EUP works well and the problem is found, but the ABC representative wants to have a weekly report on the notifications of last week. Agreements on this are made during the meeting.

Afterwards, the performance consultant and the account manager of The Backbone create a structured summary and send it to ABC to sign it.

6.4 EVALUATION OF REP V.1

REP v.1 is evaluated by an observation of the performance consultant using REP v.1 for the *requirements composition* phase at a new customer for the EUP. After the observation, the author of this thesis interviewed the performance consultant about his experience with REP v.1. The questions of this interview are the same questions as the interview in section 5.4 and can be found in Appendix D.

6.4.1 *Observation*

Before the observation could take place, the author of this thesis first briefed the performance consultant about REP v.1 and what was expected from the performance consultant while using REP v.1. It was made clear towards the performance consultant that it is very important to send the questionnaire to the customer on time. Because it was already clear which application needed performance monitoring, the performance consultant changed the questionnaire a little, mostly by mentioning the name of the application in the questions.

The observation took place during a meeting between the performance consultant of The Backbone and the customer at the location of the customer. The customer in this case was a large educational institution in The Netherlands. In this case, the implementation of the EUP product is part of a larger project. The representative of the company was an application manager of the application that needed performance monitoring. The end-user performance of the application which the application manager manages is of high impact for the students as well as the teachers of the educational institution. This observation took place in English since the customer's representative does not speak Dutch. Both, the author of this thesis and the performance consultant had no problem with working in English instead of Dutch. Due to time limits for this thesis, the meeting with the customer was scheduled two days before the meeting took place. Therefore, the questionnaire was not send directly after the proposal was signed by the customer. The questionnaire was filled in and send back by the customer's representative two hours before the meeting started. This gave the author of this thesis and the performance consultant time to scan the answers on the questions quickly. Due to time limits on the performance consultant's side, it was not possible to test the complete REP. However, the parts of the REP in which most changes are made compared to the current REP practice at The Backbone are evaluated. The evaluated parts of REP v.1 are the subphases *information gathering*, *validation of information*, and *solution design* of the requirements composition. It was possible to test these three subphases in two hours' time, mainly due to the monitoring background of the application manager of the customer. Having a customer representative with monitoring experience is an exception with a small chance to appear.

The performance consultant started the meeting with the question whether the application manager had questions about the EUP. Since the performance consultant gave a demonstration of the EUP the week before, this was not the case. Therefore the performance consultant started to discuss the answers that were given in the questionnaire. It took the application manager 15 minutes time to fill in the questionnaire. Since the answers to the questionnaire were send by mail to the performance consultant, the performance consultant opened the file on his laptop and typed his field notes on the right places in the document. The monitoring experience background of the application manager caused more precise answers to the questions in the questionnaire than expected. Therefore, the task of the performance consultant changed a bit. Instead of asking deeper, the performance consultant had to ask back to the motives of the customer for this answers and from that point the performance consultant was able to ask deeper. The application manager, for example, already designed the scenarios in the questionnaire. After the answers to the questionnaires were discussed, the components were filled in the component table (Table 5) by the performance consultant and the application manager. It was noticed by the author of this thesis that the performance consultant was able to ask deeper during the whole meeting. This resulted in more components than The Backbone is able to implement in the EUP with the current contract. The author of this thesis noticed that it would help the process if the components can get prioritised. After the components were discussed, the dashboard needs followed by the notification needs were discussed. It was noticed by the author of this thesis that the meeting was more structured than meetings with the current REP practice of The Backbone. Sometimes, the performance consultant needed Figure 8 to realise what his next step should be. The next part of the discussion was the simulations and simulation steps. During the meeting, two simulations are roughly designed. However, there was no time left for the performance consultant to design those simulations in detail. In the end of the meeting, the watcher nodes were discussed. This was more practical than discussing them before the simulations were roughly designed. The author of this thesis noticed that the use of REP v.1 gave a professional and structured image towards the customer.

6.4.2 *Interview*

Directly after the observation, the author of this thesis conducted an interview with the performance consultant about his opinion on REP v.1. The interview questions in Dutch and in English can be found in Appendix D. This interview took is conducted in Dutch since this is the mother language of both the author of this thesis and the performance consultant.

The performance consultant believes that REP v.1 is better than the current REP practice of The Backbone in all aspects, like: usability, usefulness, efficiency, quality of written requirements, implementation quality of requirements, and professional image (Appendix D).

REP v.1 gives the performance consultant structure and directions during the meeting. Since this process is new to the performance consultant, it helps him to have Figure 8 ready because it is a summary of REP v.1 and therefore the performance consultant can check easily what the next step he has to conduct is. The performance consultant mentioned that the questionnaire definitely helped him to ask as much as possible. The questionnaire also forced to customer to think about a solution before the performance consultant arrives for the meeting. Because of these reasons, the performance consultant intends to use REP v.1 more often.

The only feedback that the performance consultant had was to mention in the questionnaire that all links must be visible in the architecture.

6.5 CONCLUSIONS OF REP V.1

This chapter's most important conclusion is that REP v.1 is an improvement in respect to the current REP practice at The Backbone as well as to REP v.0. The performance consultant believes that REP v.1 performs better on aspects like: usability, usefulness, efficiency, quality of written requirements, implementation quality of requirements, and professional image than the current REP practice at The Backbone. Furthermore, all objectives formulated in section 6.1 are processed into the design of REP v.1.

The use of REP v.1 caused a structured meeting between the performance consultant and the customer. This gave a more professional image towards the customer. The structured way of working with REP v.1 led to the performance consultant asking deeper questions to the customer and more components that should come back in the simulations. Through the use of REP v.1, all those components were directly documented by the performance consultant.

There are some improvements for the REP:

- The questionnaire should mention that all links between components in the architecture must be visible.
- A possibility to prioritise the components in the component table (Table 5) should be added.

These improvements are the input for the objectives in chapter 7.

7 REP V.2

This chapter describes the work done to answer RQ5 and RQ6 by designing and evaluating REP v.2. First the objectives for REP v.2 are described, before REP v.2 is designed and described into detail. After the design, a demonstration of REP v.2 follows. Finally, REP v.2 is evaluated and conclusions about REP v.2 are given.

7.1 OBJECTIVES OF REP V.2

For REP v.2, there are two types of objectives: objectives that apply to all three versions of the new REP and objectives specific to REP v.2.

The objectives that apply to all three versions of the new REP are objectives O-1 up to and including O-8 which are already described in section 5.1.

The objectives specific to REP v.2 consists of improvements on REP v.1. Those improvements have been found during the evaluation of REP v.1 (section 6.4) and are summarised below.

The Backbone can already use REP v.1, however there are some improvements to fine-tune the new REP.

The first improvement is for the questionnaire to mention that all links between components in the architecture must be visible. The performance consultant needs this information and customers may forget drawing these links in the architecture otherwise.

The second improvement is to add a possibility to prioritise the components in the component table. This improvement is needed because with the use of the new REP it is easier to get more components that the customer wants to see in the simulations than that there is time to implement those components. If those components can get prioritised, the performance consultant can create a list with the customer of the components that are 'must haves' and 'nice to haves'. The improvements are summarised into objectives below:

- O-18: The new REP should describe in the questionnaire exactly what information is needed from the customer.
- O-19: The new REP should have the possibility to prioritise the components that need to come back in the simulations.

7.2 DESIGN OF REP V.2

REP v.2 is an improved version of REP v.1 (chapter 6). Since REP v.2 is improved on only two parts with respect to REP v.1, most parts of

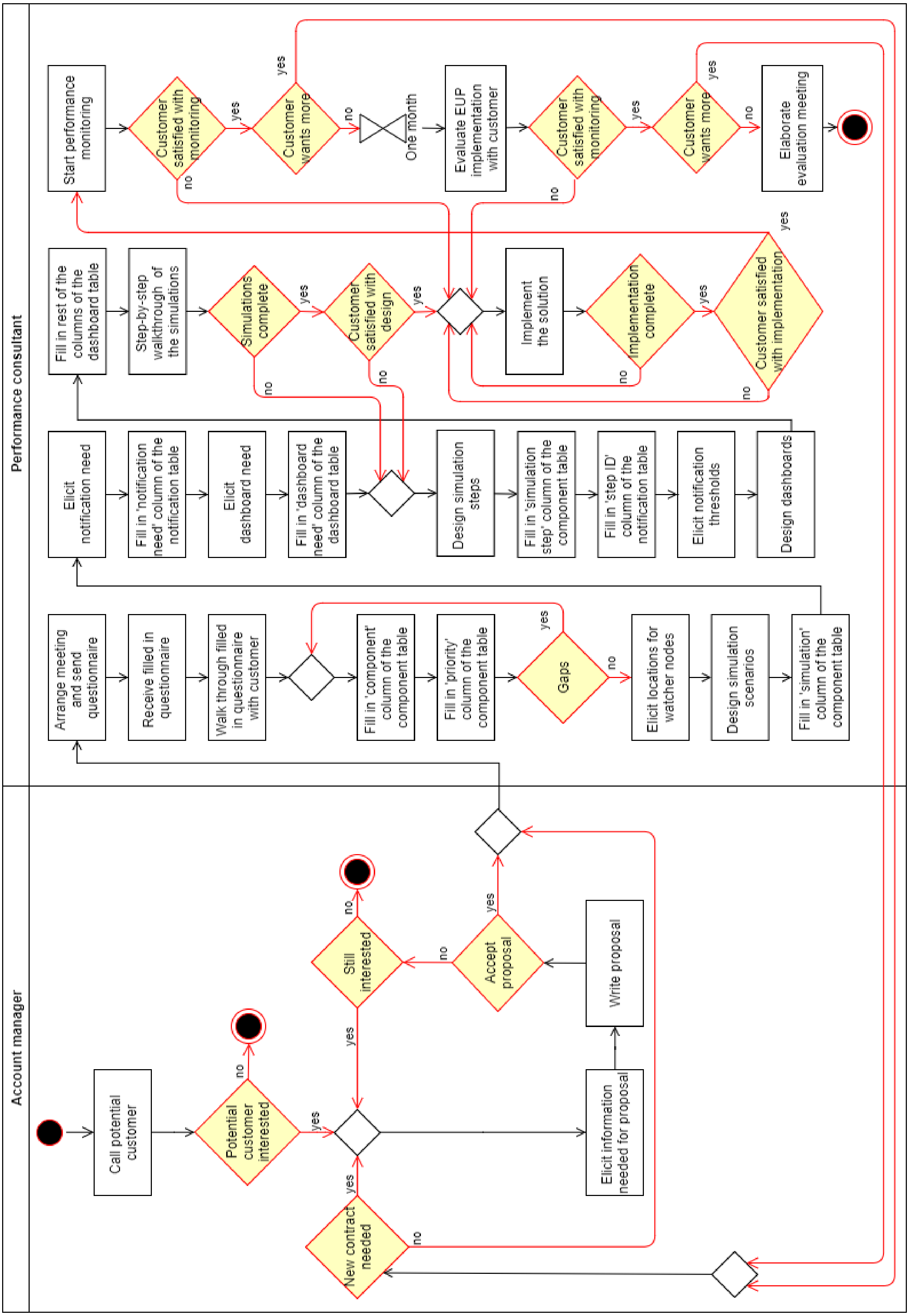
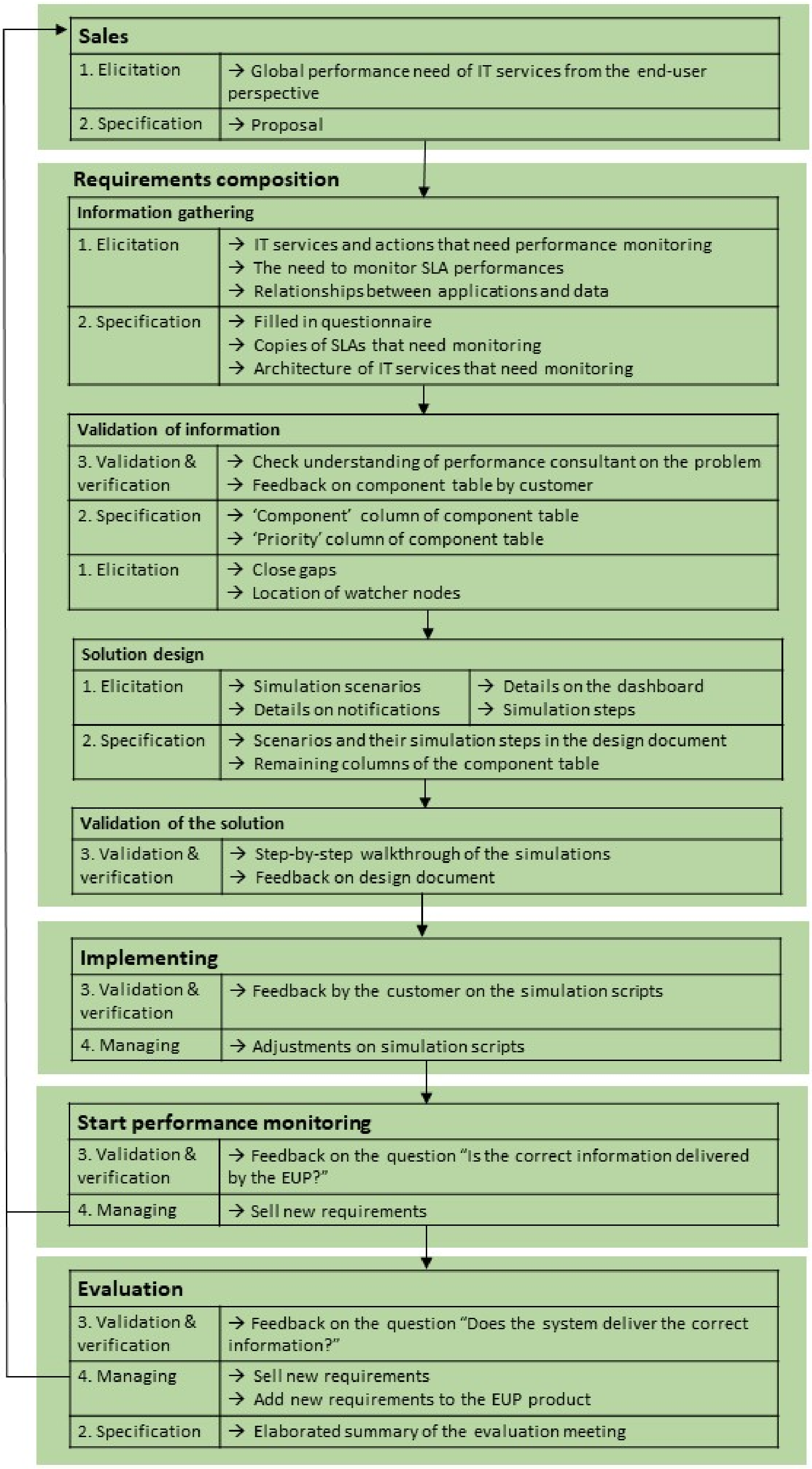

Account manager
Performance consultant
Call potential customer
Potential customer interested
yes
no
Elicit information needed for proposal
Write proposal
Accept proposal
yes
no
Still interested
yes
no
New contract needed
yes
no
Arrange meeting and send questionnaire
Receive filled in questionnaire
Walk through filled in questionnaire with customer
Fill in 'component' column of the component table
Fill in 'priority' column of the component table
Gaps
yes
no
Elicit locations for watcher nodes
Design simulation scenarios
Fill in 'simulation' column of the component table
Elicit notification need
Fill in 'notification need' column of the notification table
Elicit dashboard need
Fill in 'dashboard need' column of the dashboard table
Design simulation steps
Fill in 'simulation step' column of the component table
Fill in 'step ID' column of the notification table
Elicit notification thresholds
Design dashboards
Fill in rest of the columns of the dashboard table
Step-by-step walkthrough of the simulations
Simulations complete
yes
no
Customer satisfied with design
yes
no
Implement the solution
Implementation complete
yes
no
Customer satisfied with implementation
yes
no
Start performance monitoring
Customer satisfied with monitoring
yes
no
Customer wants more
yes
no
One month
Evaluate EUP implementation with customer
Customer satisfied with monitoring
yes
no
Customer wants more
yes
no
Elaborate evaluation meeting

Figure 9: Activity diagram REP v.2



Sales

1. Elicitation	→ Global performance need of IT services from the end-user perspective
2. Specification	→ Proposal

Requirements composition

Information gathering	
1. Elicitation	→ IT services and actions that need performance monitoring → The need to monitor SLA performances → Relationships between applications and data
2. Specification	→ Filled in questionnaire → Copies of SLAs that need monitoring → Architecture of IT services that need monitoring

Validation of information	
3. Validation & verification	→ Check understanding of performance consultant on the problem → Feedback on component table by customer
2. Specification	→ 'Component' column of component table → 'Priority' column of component table
1. Elicitation	→ Close gaps → Location of watcher nodes

Solution design		
1. Elicitation	→ Simulation scenarios → Details on notifications	→ Details on the dashboard → Simulation steps
2. Specification	→ Scenarios and their simulation steps in the design document → Remaining columns of the component table	

Validation of the solution	
3. Validation & verification	→ Step-by-step walkthrough of the simulations → Feedback on design document

Implementing

3. Validation & verification	→ Feedback by the customer on the simulation scripts
4. Managing	→ Adjustments on simulation scripts

Start performance monitoring

3. Validation & verification	→ Feedback on the question "Is the correct information delivered by the EUP?"
4. Managing	→ Sell new requirements

Evaluation

3. Validation & verification	→ Feedback on the question "Does the system deliver the correct information?"
4. Managing	→ Sell new requirements → Add new requirements to the EUP product
2. Specification	→ Elaborated summary of the evaluation meeting

Figure 10: Summary REP v.2

REP v.2 are the same as in REP v.1. For REP v.2, changes have been made in Appendix C, these changes are processed in Appendix F together with the changes made in the design document in chapter 5 and chapter 6. The improvements consist of the objectives given in section 7.1.

Figure 9 shows the activity diagram of REP v.2 in which the activities are grouped per actor (the account manager and the performance consultant). REP v.2 is divided into five phases: *sales*, *requirements composition*, *implementing*, *start performance monitoring*, and *evaluation*. Each of these phases has its own purpose. Four subphases form together the *requirements composition* phase. Figure 10 summarises REP v.2. The arrows in this figure indicate that the results of the previous (sub)phase are the input for the next (sub)phase. The five phases of REP v.2 are described in detail in the subsections below.

7.2.1 *Sales*

The first phase of the REP is *sales*. The purpose of this phase is to sell the EUP. REP v.2 starts therefore by the account manager. The account manager tries to sell the EUP to potential customers by asking if they experience IT performance problems or if they have a performance need of IT services from the end-user perspective, for example SLA monitoring or monitoring for incident management. In such a case, the account manager sketches them a solution with the EUP product of The Backbone. If the potential customer is interested in buying the EUP, the account manager needs to know the performance need of IT services from the end-user perspective of the customer. It is preferred to gather this information during a sales meeting since a meeting leads to more information than a phone call does. This information is needed for the account manager to make the proposal as accurate as possible. Gathering information on the performance need of IT services from the end-user perspective of the customer belongs to the *elicitation* phase of Figure 1. The account manager does not have to know the IT performance need in very much detail, but he must have gathered enough information to make a decent proposal. If the customer accepts the proposal, the EUP is sold and the next phase starts. The proposal belongs to the *specification* phase of Figure 1, since the global IT performance need is documented in the proposal.

Since not every sales meeting leads to a sale, the performance consultant is normally not involved in this process. However, if it is already clear that the potential customer is really interested or the potential customer indicates that technical staff is joining the meeting, the account manager can bring the performance consultant to the sales meeting for bringing in more technical knowledge to this sales meeting.

Whenever new requirements are sold in the *start performance monitoring* or *evaluation* phase, the *sales* phase can takes less time than it takes for a new EUP project. The *sales* phase is in that case needed to check whether the new requirements fit in the current contract or if a new contract is needed. If no new contract is needed, the *sales* phase is finished and the *requirements composition* phase starts. If a new contract is needed, the *sales* phase is also faster since the information needed for an accurate proposal is already gathered in the *start performance monitoring* or *evaluation* phase.

7.2.2 *Requirements composition*

After the EUP is sold by an account manager or when new requirements are sold to an existing customer, the requirements for the EUP simulations need to be clear before the design and implementation of the EUP can start. This takes place in the *requirements composition* phase. To have the requirements for the EUP simulations clear, the performance consultant needs consecutively to: gather information on the performance need of IT services from the end-user perspective of the customer, validate his understanding of this IT performance need with the customer, think of a solution for this IT performance problem, and validate this solution with the customer. Based on this, the next subphases need to be carried out by the performance consultant: *information gathering*, *validation of the information*, *solution design*, *validation of the solution*.

7.2.2.1 *Information gathering*

The first thing the performance consultant has to do is to gather a lot of information on the performance need of IT services from the end-user perspective of the customer. To let the customer already think of IT services and simulations they want to have measured before the performance consultant goes to meet the customer, a questionnaire is sent. This questionnaire is an attachment in the design document (Appendix F) but can also be found in Appendix G. Multiple employees of the customer who have to work with the EUP are asked to fill this questionnaire. Along with the questionnaire, a website link to the EUP product website is also sent to the customer. This website contains all information a customer needs for the EUP. This website link prevents that the performance consultant goes to the customer just to tell them again what the EUP is. The questionnaire helps the customer to already think of the applications and simulations they want to monitor the performance of and therefore prevents that the performance consultant has come to the customer for no reason. In this questionnaire, the customer is also asked to send copies of SLAs if they want to monitor the performances of SLAs. Finally, the customer is also asked to draw an architecture of the applications, functions of

Table 12: Component table REP v.2

#	Priority	Component	Simulation	Simulation step
1	1	Login on website	Website	Step 11
2				
3				
4				
5				

the IT services, needed information for these IT sevices, and the links between all these elements. In this way, the performance consultant can prepare himself optimally before visiting the customer and the customer has thought about their problem as well. All information gathered in this *information gathering* subphase, belongs to the requirements *elicitation* phase in Figure 1, but the documents in which all this information is gathered, belong to the *specification* phase as represented in Figure 1.

7.2.2.2 *Validation of requirements information*

After the performance consultant received all the information and has prepared himself, there is a solution meeting between the performance consultant and the customer. The goal of this meeting is to validate the performance consultant's understanding of the IT performance need of the customer completely, to design the solution, and to validate this solution.

The validation of the performance consultant's understanding of the IT performance need of the customer is done by walking through all the information given to the performance consultant together with the customer. The validation of the performance consultant's understanding belongs to the *validation & verification* phase of Figure 1. All components coming up in the meeting are put into the component table by the performance consultant. The component table is integrated in the design document (Appendix F) and can also be found in Table 12. At this point of time, only the 'component' column needs to be filled in.

In the case that there are more components formulated than there is time available in the contract for The Backbone to implement all those components, the performance consultant has to prioritise the components. The MoSCoW rules are a straight forward way to classify all components into four different groups of importance [41]. Therefore, the MoSCoW rules are a classification of importance of the components. These MoSCoW rules are [41]:

1. **Must have**: all components with a 'must have' classification must be implemented in the system, otherwise the system would not work
2. **Should have**: components with a 'should have' classification are important to the system and contribute a significant value. However, these components can be omitted due to time constraints.
3. **Could have**: components with a 'could have' classification improve the system with functional items, but can be easily implemented at another time.
4. **Want to have**: components with a 'want to have' classification serve only a limited group of users and are of little added value and can also be implemented another time.

The performance consultant can easily put the priority number (1 up to 4) next to the component in the 'priority' column of the component table.

Every component also gets a number for reference reasons which can become useful later in the process and for communication purposes. Filling in the 'component' column of the component table (Table 12) belongs to the *specification* phase of the REP of Figure 1. At the end of the meeting, the component table gets validated by the customer to be sure nothing is missing. This belongs to the *validation & verification* phase of Figure 1. It is also possible that the performance consultant observes gaps in the IT performance need, or the information handed to him. These gaps can be filled during the meeting and therefore new requirements can also get elicited.

One question the performance consultant should never forget to ask is from which locations the applications that get their performance measured are used. This information is needed to discuss and decide where watcher nodes need to be located. Eliciting requirements to fill gaps and the locations of which applications are used belong to the *elicitation* phase of Figure 1.

7.2.2.3 *Solution design*

At this point in the process, the performance consultant has a complete understanding of the IT performance need of the customer and therefore can start working on the solution. The performance consultant invents simulations for the EUP together with the customer. This is one meeting with the *validation of requirements information* subphase.

First, the simulation scenarios need to be invented. This means that there is no need to invent the simulations directly into much detail, like: push this button. However, the steps that need to be taken need to be clear, like: start the email application, write an email, filling in the receiving email address, send the email, receive the email, open the email, and finally close the email application. The simulations

Table 13: Notification thresholds REP v.2

Notification need	Step ID	Warning	Critical
Starting CRM	CRM_start	20000	25000
Logging in in CRM	CRM_login	18000	23000
Adding data in CRM	CRM_add_data	7000	10000
Searching in the data in CRM	CRM_search_data	10000	12000
Deleting data from CRM	CRM_delete_data	15000	18000
Logging off in CRM	CRM_logoff	4000	6000
Closing CRM	CRM_close	4000	6000
Total CRM application	CRM_intern_data_total	80000	100000

get documented in the design document of Appendix F. All the components talked about earlier in the process and written down in the component table (Table 12) need to come back in the simulations. For referring, the 'simulation' column of the component table (Table 12) needs to be filled at this moment. The scenarios, their priority, and the consistency between the scenarios must be clear before designing the scenarios in detail.

In this phase of the meeting, the performance consultant also asks about which notifications need to be sent in the case that a simulation does not run successfully. The notification needs of the customer get described by the performance consultant in the 'information need' column of Table 13. The performance consultant also needs to know to whom the notifications needs have to be sent and the corresponding email address or phone number.

The dashboard is also discussed in this phase of the meeting. The performance consultant wants to know if the customer is going to use the dashboard for trend analysis or real-time information. This information is needed to choose a dashboard application. Furthermore, the performance consultants asks about the information that needs to be visible on the dashboard. This information is gathered in the first row and the 'dashboard need' column of Table 14.

When the simulations are roughly designed, the notification need is clear as are the functionalities of the dashboard, the simulations can be further designed into detail. All simulation steps need to be described very clearly into the document as shown in Appendix F.

Table 14: Dashboard needs REP v.2

GOAL: TREND ANALYSIS / REAL-TIME INFORMATION		
DASHBOARD		
NEED	SCREEN	TABLE / CHART
Trend line for the time that the CRM scenario needs to complete with a variable time frame	Main screen	Chart 1
Percentages of time each step had a 'warning' / 'critical' notification in the last 24 hours	Main screen	Table 1

Accordingly, the component table (Table 12) needs to be updated by filling the 'simulation step' column. Accordingly, the 'Step ID' column of Table 13 can be filled by the performance consultant to check whether all notification needs are integrated in the scenarios. Also the threshold in milliseconds for 'warning' and 'critical' notifications needs to be set. This is done in a cooperation with the customer and the performance consultant. Table 14 is also filled in further by the performance consultant while designing the dashboard. By filling in this table, the performance consultant checks whether all desired functionalities are in the dashboard design.

With the design of the detailed simulations, it must be considered that some simulations need accounts that give the EUP the rights to carry out the actions of the simulations. This can be hard to achieve. For example, if a DigiD (Dutch Digital Identification) is needed, because test DigiD accounts do not exist.

Documenting the simulations in the design document and to fill in the remaining columns of the component table belong to the *specification* phase of Figure 1. Gathering the requirements concerning notifications and dashboards belong to the *elicitation* phase of Figure 1.

7.2.2.4 *Validation of the solution*

Before the scripting can start, the designed solution for the customer needs to be validated. The performance consultant walks through the simulations step-by-step. This is a very time consuming process since the performance consultant needs to focus on every possible little detail. An example of a small detail is: does the simulation need to press the Enter key on the keyboard or click the OK button on the screen? All these steps are documented in a table with three columns: the step, the action (click on, fill in a user name, close the window), and the check (very detailed description of the outcome when the action is carried out in the right way, for example: a certain window or piece of text appears). This check belongs to the *validation & verification* phase of Figure 1. Since a lot of questions can arise to the performance consultant during this check, it is preferred for this subphase to occur during the same meeting as the *validation of requirements information* and *solution design* subphase.

The design document (Appendix F), including the table for the simulations, is detailed in such a level that another performance consultant is able to make the scripts without needing more information.

Another benefit of this detailed design document is that employees of the customer that have not been in contact with The Backbone yet, can also check the requirements and simulations. This is also why the performance consultant of The Backbone asks the customer for feedback and confirmation: is the design document correct and complete? This belongs to the *validation & verification* phase of Figure 1.

7.2.3 *Implementing*

After feedback on the document is processed and the customer confirmed the document, the actual scripting takes place. This happens in the *scripting* phase.

Simple scripts are sometimes already scripted during the *requirements composition* meeting. The scripting is done at the location of the customer. Preferable, an employee of the customer is around to answer questions. During this scripting process, the need to adjust or to change requirements can arise, mainly due to technical challenges. If simulations are adjusted, or requirements change, the design document (Appendix F) needs to be updated. This belongs to the *management* phase of Figure 1. The verification by the customer, which is a step of the *validation & verification* phase of Figure 1, is also important during this phase.

7.2.4 *Start performance monitoring*

When all the simulations are scripted, the EUP starts with the performance monitoring.

In this *start performance monitoring* phase, the performance consultant also trains the customer in how the data that the EUP generates should be interpreted. The performance consultant also checks with the customer if the correct information is delivered by the system and if the customer wants more information, which belongs to the *validation & verification* phase in Figure 1. This can lead to selling new simulations or functionalities to the customer, which belongs to the *managing* phase of Figure 1.

7.2.5 *Evaluation*

One month after the performance monitoring of the EUP has been started, the *evaluation* phase begins. In this phase, the EUP as a product, the implementation of the EUP at the customer is evaluated, as well as how the customer works with the EUP and the results the EUP produced in the first months.

An evaluation meeting is planned to evaluate the performance of the EUP and to discuss the results the scripts of the EUP gave about the performance of monitored applications. This belongs to the *validation & verification* phase of Figure 1. Present at this meeting are the account manager and the performance consultant of The Backbone, both responsible for this particular customer, and representatives of the customer. This meeting takes place at the customer's location.

During this meeting, new requirements for the EUP implementation at the customer, as well as for the EUP in general, can come up for discussion. The account manager tries to sell more scripts, trainings, dashboards, and other services The Backbone offers. If this succeed, the REP starts all over again at the *sales* phase. Selling more products or services that The Backbone offers belongs to the *managing* phase of Figure 1.

The account manager and performance consultant take field notes during the evaluation meeting. Afterwards, they elaborate them into a structured summary. This summary is sent to the customer for confirmation since agreements can be made during the evaluation meeting. This confirmation can be the signature of the responsible person at the customer, or a confirmation by email. The decision whether to ask the customer to sign the document that is made after the evaluation meeting or to confirm the document by email should be left to the performance consultant. This part of the *evaluation* phase belongs to the *specification* phase of Figure 1.

7.3 DEMONSTRATION OF REP V.2

REP v.2 is demonstrated below by the use of a case study. The case study that is used in section 5.3 and in section 6.3 is reused in this chapter and can be found in Appendix A. The case study is a combination of two real cases The Backbone had in the past and are described in section 5.3. section 5.3 also described how this case study has been established. The organisation name and the applications are anonymised to guarantee the privacy of the real organisations. Since REP v.2 improves REP v.1 on two points (section 7.1), this demonstration is equal to the demonstration of REP v.1 in section 6.3 on most parts.

7.3.1 *Sales*

ABC calls an account manager of The Backbone because ABC experiences IT performance issues now and wants to know whether the EUP can help them tracing the problem or not. The account manager explains ABC how the EUP could help tracing the performance problem. Since the ABC wants the performance issue to be solved as soon as possible, they do not feel the need for the account manager to come over for a meeting to give a presentation about the EUP. Instead, ABC wants to receive a proposal as soon as possible, also to compare the costs of the EUP to those of other solutions.

The proposal gets accepted by ABC and therefore the EUP is sold.

7.3.2 *Requirements composition*

Directly after the proposal is accepted by ABC, the performance consultant of The Backbone contacts ABC to arrange a meeting with them. The performance consultant directly starts the information gathering subphase at this moment at time.

7.3.2.1 *Information gathering*

To start this subphase, the performance consultant of The Backbone sends the questionnaire and the link to the EUP product website to ABC.

The next day, The Backbone receives answers from one ABC employee to the questionnaire:

1. **Your goal of the EUP is mentioned in the contract. However, could you please tell us your needs for the EUP using your own words?**

   ABC: Our end-users are complaining about our CRM application being slow. This CRM system is just implemented in our organisation, but this application is used hundreds times each day and is very important for the continuity of our business. We want to know whether the CRM application is slow or one of the linked applications does not perform well.

2. **Which applications and/or services would you like to monitor? (Examples are: Outlook, Hix, Citrix, ERP, HRM, Website: http://www.thebackbone.nl)**

   ABC: The CRM application. But also our Order Management System (OMS) and our Enterprise Resource Planning (ERP) application since those are linked to the CRM application.

3. - **Of which of the commonly used activities by end-users do you want to monitor their performance?**

ABC: From the CRM application we want to monitor the performance of starting the CRM application, the logging in performance, the performance of adding new data to the CRM application, and the performance of searching through the CRM application. From the OMS and ERP applications, we want to monitor an action in CRM that needs to get its data from the OMS and ERP applications.

and/or

- **Of which activities and/or actions where you experience performance issues, do you want to monitor their performance?**

  ABC: See the answer on the above question.

  and/or

- **Of which of the activities that are appointed in a Service Level Agreement (SLA) do you want to monitor the performance?**

  ABC: We want to monitor the SLA for our CRM application.

4. **If you are going to use the EUP for monitoring SLAs, please sent the relevant SLAs as an attachment or have them ready before the meeting.**

   ABC: We will have the SLA ready before the meeting.

5. **What are the applications that need to be monitored and how are they linked with each other and their hardware and storage? (What is the architecture like, where does the required data come from, what are the links between the applications, etc.?)**

   The answer to this question is skipped because it has no added value for this research.

7.3.2.2 *Validation of requirements information*

Since the IT performance issues ABC experiences at the moment do have a lot of impact on the organisation, the performance consult of The Backbone meets ABC three days after ABC answered the questionnaire.

Together with ABC, the performance consultant walks through all the information ABC has given him so far. The performance consultant asks a lot of questions to the ABC representative to get a complete understanding. During this meeting, the performance consultant fills in the 'component' column of Table 8. After the 'component' column is filled in, the performance consultant and the ABC representative discuss the priorities of the components. They agree on the classification for most components as 'must haves', however CRM needing

Table 15: Component table demonstration REP v.2

#	Priority	Component	Simulation	Simulation step
1	1	Start CRM application	CRM intern data	CRM_start
2	1	Login CRM application	CRM intern data	CRM_login
3	1	Add data to CRM application	CRM intern data	CRM_add_data
4	1	Search in CRM application	CRM intern data	CRM_search_data
5	2	CRM needs data from OMS	CRM OMS	
6	2	CRM needs data from ERP	CRM ERP	
7	1	Close CRM application	CRM intern data	CRM_logoff
				CRM_close

data from OMS and ERP get the classification of 'should have'. At the end of the meeting, the ABC representative is asked to check whether all components talked about are represented in this table.

The performance consultant asks the representative of ABC from which locations end-users approach the CRM, OMS, and ERP applications. The representative answers that end-users use those applications from desktops in the ABC office as well as from their business laptops. The performance consultant asks if those laptops are also used outside the office of ABC, the representative tells him that this is often the case. Therefore the watcher nodes are going to be on a laptop located in the locked cabinet on the working floor with a good Wi-Fi signal and on a desktop with a LAN internet connection.

7.3.2.3 *Solution design*

Now, the simulations are determined by filling in the 'simulation' column of Table 15.

Next, the performance consultant asks for notification requirements. For every 'warning' an email must be sent to the ABC representative, for every 'critical' an email and a SMS must be sent to the ABC representative. The notification needs are described in the 'notification need' column of Table 16.

The ABC representative and the performance consultant now discuss the dashboards. They agree on one dashboard to start with,

Table 16: Notification thresholds demonstration REP v.2

Notification need	Step ID	Warning	Critical
Starting CRM	CRM_start	20000	25000
Logging in in CRM	CRM_login	18000	23000
Adding data in CRM	CRM_add_data	7000	10000
Searching in the data in CRM	CRM_search_data	10000	12000
Deleting data from CRM	CRM_delete_data	15000	18000
Logging off in CRM	CRM_logoff	4000	6000
Closing CRM	CRM_close	4000	6000
Total CRM application	CRM_intern_data_total	80000	100000

which is going to be used for trend analysis. This decision is based on the fact that for every warning or critical an email gets sent to the ABC representative. The 'dashboard need' column of Table 17 gets filled by the performance consultant and the ABC representative.

The following step is that Table 18 of Appendix F is filled in for the simulation *CRM intern data*. The tables for the simulations on the components *CRM OMS* and *CRM ERP* are not made for the purpose of this demonstration.

After the simulation steps have been designed, the 'simulation step' column of Table 15 gets updated and in Table 16 the step IDs of

Table 17: Dashboard needs demonstration REP v.2

Goal: Trend analysis / Real-time information		
Dashboard need	**screen**	**table / chart**
Trend line for the time that the CRM scenario needs to complete with a variable time frame	Main screen	Chart 1
Percentages of time each step had a 'warning' / 'critical' notification in the last 24 hours	Main screen	Table 1

Table 18: Simulation steps REP v.2

Step ID	Action	Check
CRM_start	Open the CRM application	Text "ABC User Dashboard"
CRM_login	Fill in username and password	Text "Welcome, Marie."
CRM_add_data	Click on: "Add customer"	Text "Customer name is:"
	Give customer the following name: "12345678"	Text "12345678"
	Click on: "Add"	Text "Customer 12345678 is added to the database"
CRM_search_data	Search for customer with name: "12345678". Press Enter	Text "12345678"
CRM_delete_data	Select "Customer name: 12345678". Press Delete	Text "12345678" cannot be found
CRM_logoff	Click on: "Log off"	Text "Are you sure you want to log off?"
	Click "Yes"	Text "You're logged off"
CRM_close	Click "X" in upper right corner	Text "Recycle bin" on desktop

the simulation as well as the thresholds for 'critical' and 'warning' notifications are set.

The dashboard gets designed and the 'dashboard screen' and 'dashboard table / chart' columns of Table 17 are filled by the performance consultant to check whether all dashboard needs are fulfilled by the dashboard design.

After the ABC representative creates a CRM account for the EUP, this phase is finished.

7.3.2.4 *Validation of the solution*

The performance consultant and the ABC representative both check if the simulation steps are formulated well and if nothing is missing. Everything is good, therefore the *implementing* phase can start.

7.3.3 *Implementing*

The *CRM_intern_data* script is not very complicated and therefore gets scripted directly after the simulation steps are checked. The ABC representative stays close to the performance consultant to answer possible questions. The dashboard is later built in the office of The Backbone.

7.3.4 *Start performance monitoring*

After the scripts and the dashboards are finished, the performance consultant goes the ABC office for another meeting with the ABC representative. The performance consultant installs the scripts on the watcher nodes and gives the ABC representative access to the dashboard.

Accordingly, the performance consultant explains to the ABC representative how the dashboard should be read and how the results can be analysed using the dashboard.

Also the first results are explained by the performance consultant to the ABC representative.

7.3.5 *Evaluation*

One month after the *delivery* phase, the performance consultant has another meeting with the ABC representative, but now the account manager also joins them. And in between the performance consultant contacts and/or visits the customer on weekly basis to analyse the data and fine-tune the monitoring.

The EUP works well and the problem is found, but the ABC representative wants to have a weekly report on the notifications of last week. Agreements on this are made during the meeting.

Afterwards, the performance consultant and the account manager of The Backbone create a structured summary and send it to ABC to sign it.

7.4 EVALUATION OF REP V.2

REP v.2 is evaluated by interviewing two professionals in the field of requirements engineering from other organisations than The Backbone for which REP v.2 is designed. The professionals are interviewed separately. Both interviews took place in Dutch, since this is the native language of the author of this thesis and both the interviewees.

The interviewees are asked to participate in this research for three reasons. The first reason is that both interviewees are professionals on the field of requirements engineering for software. The second reason is that the interviewees both work at organisations that are more mature in software developing than The Backbone is. The Backbone does not have a lot of experience with developing software and they also do not have a process defined for this. The third reason is that the organisations were the interviewees work, develop software for large organisations which depend on their software for their business continuity. Therefore, they are precise with their requirements.

The first interviewee is an information analyst and has one year experience with requirements engineering. The projects in which the interviewee is involved, concern a software product for mortgages consisting of a standard package which is the same for every customer and add-ons. The interviewee uses requirements engineering for these add-ons that are custom-made.

The second interviewee is also an information analyst. The interviewee has 20 years experience in software engineering of which 12 years in requirements engineering. The projects the interviewee is involved in, concern mostly custom-made projects for one of the public transporters in The Netherlands. The team of the interviewee is developing software for only this customer. Therefore, the interviewee knows the customer well.

The interviews were voice recorded and field notes were made during the interviews. These field notes were elaborated directly after the interviews. It was made clear to both interviewees that the goal of the interview was to receive as much feedback as possible. Therefore, the interviewees were encouraged to ask questions during the evaluation and to not save them until the end of the evaluation. The interviews were divided into two parts. First, the author of this thesis gave explained REP v.2 by a walkthrough with the use of Figure 10. In this part, the interviewees gave already their opinion on the (sub)phases and the components of REP v.2. Second, the author of this thesis interviewed the professionals on their opinion on certain aspects of REP v.2. The questions asked during the interview are the same questions that are used in the evaluations of REP v.0 (section 5.4) and REP v.1 (section 6.4). The questions can be found in Appendix D.

7.4.1 *Walkthrough*

During the walkthrough, the author of this thesis explained every (sub)phase of REP v.2, and the interviewees gave their opinion and their way of working about that phase directly afterwards. The comments of the interviewees described per (sub)phase below.

7.4.1.1 *Sales*

Both interviewees mentioned that due to the size of their software projects they need more requirement elicitation than just the global problem before they can make an accurate proposal. Since the software projects of The Backbone are smaller in size, the interviewees believe that the global problem should be enough information for The Backbone to make an accurate proposal. The principal is also the same: in the *sales* phase, the information needed to make an accurate proposal should be elicited.

The first interviewee mentioned that for the company the interviewee works for it could take even months before a proposal can be made, due to the complexity of the software they develop. Some components of the *requirements composition* phase (for example: close gaps and reporting which is comparable to notifications and dashboards) are already completed before the proposal can be made, since this information is needed for them to make an accurate proposal. Besides that, the *sales* phase is at this organisation more 'ongoing'. This means that sales takes place during the whole process, also in the *requirements composition* phase and the *implementing* phase.

7.4.1.2 *Requirements composition*

Both interviewees recognised the division of the *requirements composition* phase in the four subphases: *information gathering*, *validation of information*, *solution design*, *validation of the solution*.

INFORMATION GATHERING Information that The Backbone would gather in the *information gathering* subphase, is already gathered during the *sales* phase by the two organisations of the interviewees. However, the goal of The Backbone as well as the organisations of the interviewees in this subphase is to deepen the information they already received.

The first interviewee also mentioned that they gather information from external sources (the customer) as well as internal sources (experts on mortgages working for the organisation). The internal sources are used since mortgages are extremely complex on multiple aspects.

The Backbone would not benefit from having internal sources since they work with a lot of different subjects. Besides that, if they are working with a more complex subject, there are experts on that field in the organisation of the customer who can help The Backbone with questions about that subject.

VALIDATION OF INFORMATION Both interviewees recognise this subphase from their own requirement engineering process. However, the first interviewee mentions that their validation of the information is divided into an internal validation where software engineers check whether the requirements would cause performance issues or high maintenance and an external validation where they check the requirements with the customer and search for hidden requirements. This internal validation would be a good addition for The Backbone since it leads to a more realistic view on the solution for the customer.

SOLUTION DESIGN To design the solution, the organisations of both interviewees use other methods. The first interviewee uses story-mapping for the design of the solution. The second interviewee makes a functional design for this purpose. For The Backbone, these methods would lead to over-engineering the solution design since their product is smaller in size. The current method fits the solution design purpose best.

VALIDATION OF THE SOLUTION Both the organisations of the interviewees validate their solution internal as well as external. The internal validation exists of a check by the team whether there are requirements missing or if the solution design is complete. The external validation exists of the customer giving feedback on the solution design. The internal validation is a good addition to REP v.2 for The Backbone, checking the design by yourself before asking the customer for feedback leads to a better quality of the product.

7.4.1.3 *Implementing*

Both the organisations of the interviewees validate their software internal as well as external. The check whether all requirements are actually implemented is done in the internal validation. The feedback of the customer on the software is the external validation. This internal validation should also be explicit for The Backbone since this is a crucial step in the process to deliver a good product.

7.4.1.4 *Start performance monitoring*

This is a very specific term for the EUP product. Both organisations of the interviewees would recognise themselves with a term as 'start running the product'.

7.4.1.5 *Evaluation*

According to the first interviewee, the organisation the interviewee is working for could learn from this evaluation phase since the organisation the interviewee works for does not evaluate often enough. When there is an evaluation, only a sales person joins this meeting and no technical person is attending. Besides these real evaluation meetings which are being done too little, the organisation monitors the satisfaction of their customers more often with questions about how often they still use the software product for example.

The second interviewee has contact with the customer every other week. Besides this, every six months till one year, the interviewee visits them to do a proper evaluation.

Repeatable evaluating implementations of the EUP product would help The Backbone to improve the EUP product but also give them the knowledge on how their EUP product still gets used by their customers after some time.

7.4.2 *Interviews*

After the walkthroughs, the interviews took place.

First it was asked with which parts of REP v.2 the interviewee could work in the organisation of the interviewee. It appeared that all (sub)phases of REP v.2 can be used by both interviewees. The *evaluation* phase could even improve the REP of the organisation of the first interviewee.

Both the interviewees also had some suggestions to improve REP v.2 further:

- A planning in time and tasks should be added to inform the customer what sources you will need from them and what they can expect from you.
- Internal validation is missing in REP v.2. During an internal validation, employees of the organisations itself are checking whether the results so far satisfy.
- The *evaluation* phase should be repeated every six months till one year.

Next, the questions from Appendix D were asked to the interviewees. Both the interviewees believe that REP v.2 is scoring great on all aspects, like: usability, usefulness, efficiency, quality of written requirements, implementation quality of requirements, and professional image.

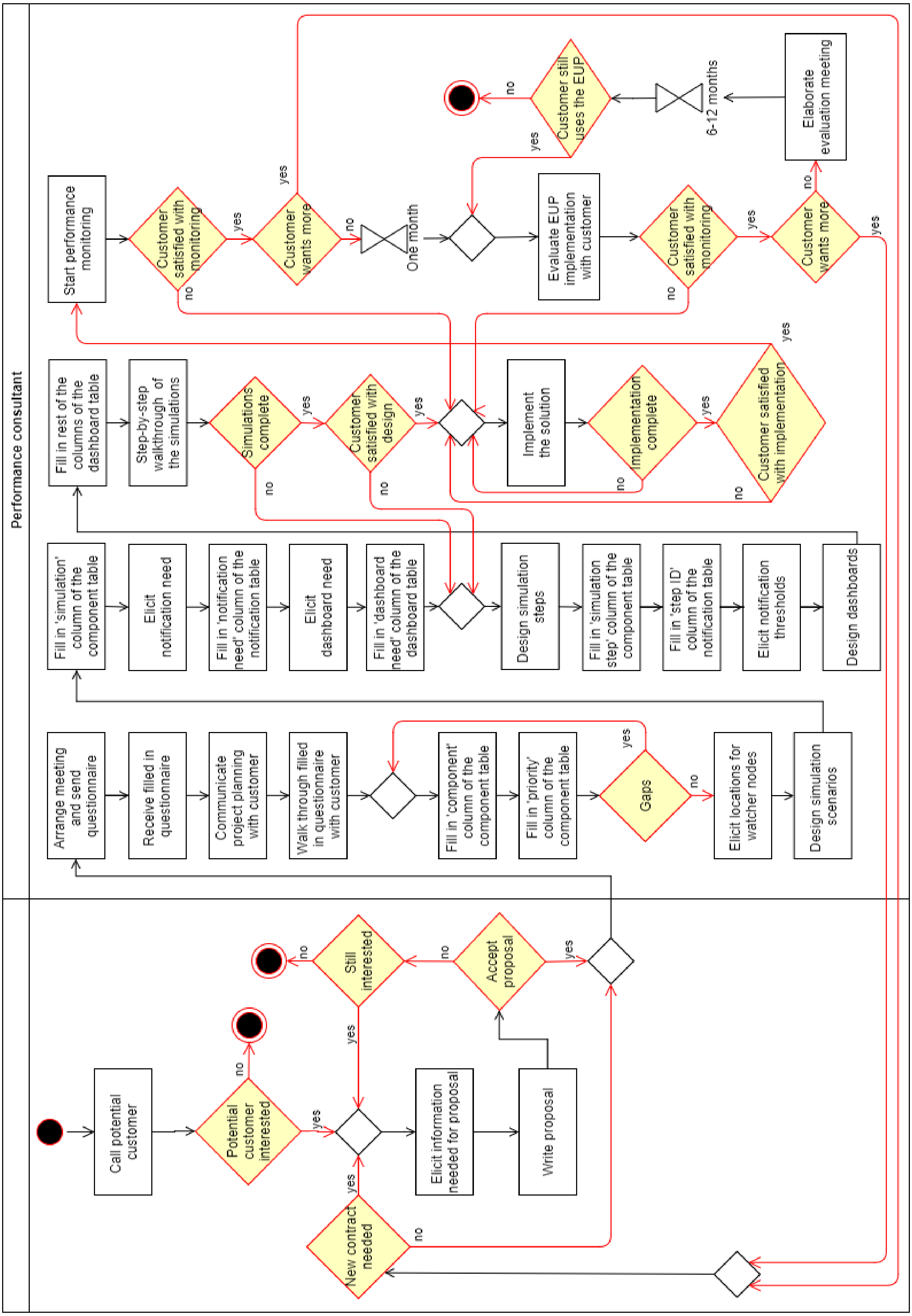
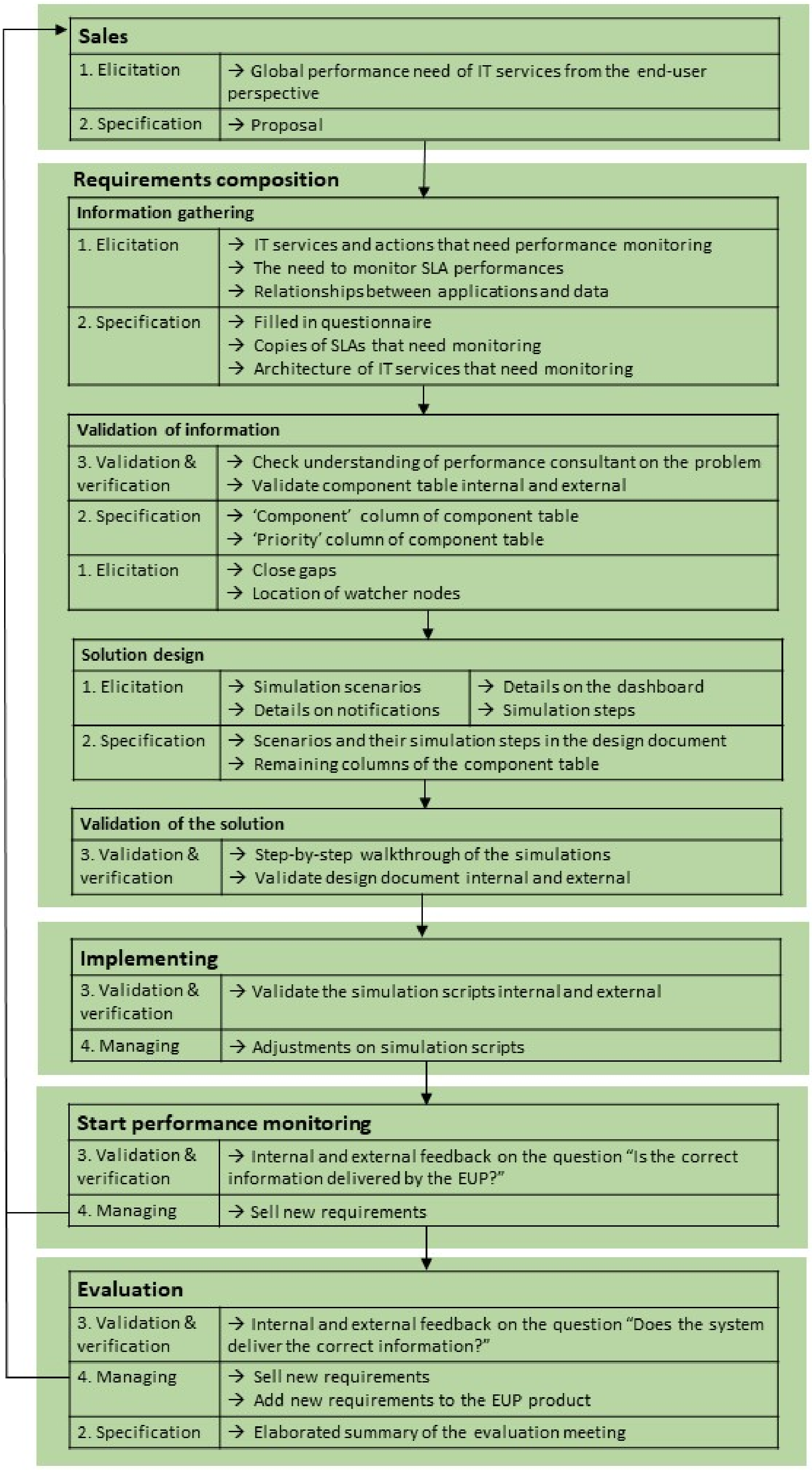

Performance consultant
Call potential customer
Potential customer interested
no
yes
Elicit information needed for proposal
Write proposal
Accept proposal
no
yes
Still interested
yes
no
New contract needed
yes
no
Arrange meeting and send questionnaire
Receive filled in questionnaire
Communicate project planning with customer
Walk through filled in questionnaire with customer
Fill in 'component' column of the component table
Fill in 'priority' column of the component table
Gaps
yes
no
Elicit locations for watcher nodes
Design simulation scenarios
Fill in 'simulation' column of the component table
Elicit notification need
Fill in 'notification need' column of the notification table
Elicit dashboard need
Fill in 'dashboard need' column of the dashboard table
Design simulation steps
Fill in 'simulation step' column of the component table
Fill in 'step ID' column of the notification table
Elicit notification thresholds
Design dashboards
Fill in rest of the columns of the dashboard table
Step-by-step walkthrough of the simulations
Simulations complete
yes
no
Customer satisfied with design
yes
no
Implement the solution
Implementation complete
yes
no
Customer satisfied with implementation
yes
no
Start performance monitoring
Customer satisfied with monitoring
yes
no
Customer wants more
yes
no
One month
Evaluate EUP implementation with customer
Customer satisfied with monitoring
yes
no
Customer wants more
yes
no
Elaborate evaluation meeting
6-12 months
Customer still uses the EUP
yes
no

Figure 11: Activity diagram REP v.2 after evaluation



Sales

1. Elicitation	→ Global performance need of IT services from the end-user perspective
2. Specification	→ Proposal

Requirements composition

Information gathering	
1. Elicitation	→ IT services and actions that need performance monitoring → The need to monitor SLA performances → Relationships between applications and data
2. Specification	→ Filled in questionnaire → Copies of SLAs that need monitoring → Architecture of IT services that need monitoring

Validation of information	
3. Validation & verification	→ Check understanding of performance consultant on the problem → Validate component table internal and external
2. Specification	→ 'Component' column of component table → 'Priority' column of component table
1. Elicitation	→ Close gaps → Location of watcher nodes

Solution design		
1. Elicitation	→ Simulation scenarios → Details on notifications	→ Details on the dashboard → Simulation steps
2. Specification	→ Scenarios and their simulation steps in the design document → Remaining columns of the component table	

Validation of the solution	
3. Validation & verification	→ Step-by-step walkthrough of the simulations → Validate design document internal and external

Implementing

3. Validation & verification	→ Validate the simulation scripts internal and external
4. Managing	→ Adjustments on simulation scripts

Start performance monitoring

3. Validation & verification	→ Internal and external feedback on the question "Is the correct information delivered by the EUP?"
4. Managing	→ Sell new requirements

Evaluation

3. Validation & verification	→ Internal and external feedback on the question "Does the system deliver the correct information?"
4. Managing	→ Sell new requirements → Add new requirements to the EUP product
2. Specification	→ Elaborated summary of the evaluation meeting

Figure 12: Summary REP v.2 after feedback

7.5 CONCLUSIONS OF REP V.2

We conclude that REP v.2 is an improvement in respect to the current REP practice at The Backbone as well as to REP v.0 and REP v.1. Two professionals in the area of requirements engineering believe that REP v.2 performs well on aspects like: usability, usefulness, efficiency, quality of written requirements, implementation quality of requirements, and professional image. All objectives of section 7.1 are processed in the design of REP v.2.

Both professionals recognised all aspects from REP v.2 from their own REPs and the *evaluation* phase of REP v.2 could even improve the REP of one of the interviewees.

There are also some improvements for REP v.2 mentioned by the interviewees:

- A planning with times and tasks should be added into the REP to inform the customer what sources you will need from them and what they can expect from you.
- Internal validation should be added into REP v.2 and to not only let the customer validate the requirements.
- The *evaluation* phase should be repeated every six months till a year.

These improvements are embedded into the design document (Appendix F) and into Figure 11. Besides that, this final proposal of REP v.2 is summarised in Figure 12.

Figure 11 can be seen as the final design proposed in this thesis.

8 DISCUSSION

This chapter discusses the results of this research in light of its usability in certain circumstances, its position in literature and its generalisability. Furthermore, the current REP practice for the EUP at The Backbone is compared to REP v.2.

8.1 POSITION OF REP V.2

The discussion below describes when REP v.2 can be used. It discusses those sizes of the software project, ways of working, the nature of the software projects, and the environmental factors of organisations in which REP v.2 is useful.

8.1.1 *REP v.2 and the size of the software project*

REP v.2 is designed especially for the EUP product of The Backbone. An implementation of the EUP product is a small software project with relatively few requirements, measured in function points [1], compared with other software projects. Function points can be determined from the requirements specification [34]. Function points counts are a more consistent measure for the software size than source lines of code are [27]. Despite the fact that REP v.2 is designed especially for a small software project as the EUP is, REP v.2 is probably usable for all sizes of software projects.

During the interviews with professionals in RE reported in section 7.4 it is found that REP v.2 can also be used in a generalised version (section 8.2) by large software projects with many requirements, measured in function points. For large software projects, more requirements elicitation is needed before a cost estimation can be done and an accurate proposal can be made by the software vendor than that is needed in small software projects. The more complex a software project becomes, the more requirements elicitation is needed [35]. This leads to more requirements elicitation in the *Sales* phase, but due to the software project being large there are still requirements to be elicited in the *Requirements composition* phase. Large software projects also have more software developers working on the software. Not every developer needs all requirements and therefore parts of REP v.2 can be done in parallel by multiple persons at the same time by focussing on the requirements of different aspects of the software project. The need for communication [18] and coordination [24] depends on the size of the software project. Besides face-to-face communication, documentation is also an important part of communication [16]. Large software projects need more documentation, communication, and coordination to keep all stakeholders and developers up to date about the progress of the software project and to keep the same goal in mind.

For very small software projects with just a handful of function points, REP v.2 can also be used in its generalised version (section 8.2). However, as The Agile Manifesto states as one of its purposes, getting the software to work is more important than comprehensive documentation [17]. Using REP v.2 for just a handful of function points, would probably lead to more documentation than needed and therefore cost unnecessary time of the software vendor.

8.1.2 *REP v.2 and the project type*

There are different types of projects, and in all of these types, requirements have a different role [26]. Lauesen distinguishes between seven different project types in his book Software Requirements. These seven project types have different characteristics.

REP v.2 exists for a great extent of communication between the supplier and customer to elicit the customer's requirements (section 7.2). It is therefore an important characteristic for a project type to have a lot of communication between the supplier and customer for the requirements elicitation.

Another great extent of REP v.2 is the specification of the elicited requirements (section 7.2), in other words: another important characteristic for a project type is that requirements are written down.

In his book, Lauesen distinguishes between the following project types [26]:

1. **In-house development:** there is communication about the requirements elicitation with the customer department and the development department of the company. Most of the time, there are no requirements specified.

2. **Product development:** the marketing department and the development department communicate for the requirements elicitation. Requirements are used and specified at many levels.

3. **Time-and-material based development:** the customer and supplier communicate to elicit requirements. Requirements are unwritten and develop over time. However, there is a version of this project type where written requirements are introduced. Written requirements help here to keep costs and expectations realistic.

4. **(Fully) customer off the shelf (COTS) purchase:** fully COTS software products have had no requirements elicited from the

buying customers. And there are therefore also no requirements from the customer written down. There do also exist COTS software products that still need a lot of configuration which have to be done by a consultant. In this case, communication between the supplier and customer is needed to elicit the requirements of the configuration. These requirements may or may not be written down.

5. **Tender:** the customer writes his own requirements and send them to potential software suppliers. The supplier therefore does not elicit the requirements from the customer.

6. **Contract development:** the supplier and customer work together for the requirements elicitation and specification.

7. **Sub-contracting:** these type of projects can be either requirements based or time-and-materials based without written requirements.

In Table 19, the characteristics of the seven project types that are important for the usability of REP v.2 are summarised. The 'communication' column shows whether there is communication between the supplier and the customer for requirements elicitation in the project type. The 'written requirements' column describes whether there are written requirements from the customer in the project type. Please note that only fully COTS purchase is summarised in this table and not the type of COTS purchasing that still needs configuration since this type is a combination between fully COTS purchasing and contract development.

Table 19 shows that the following project types meet the characteristics that are important for REP v.2:

- Product development
- Time-and-material based (with written requirements)
- Contract development
- Sub-contracting (requirements based)

REP v.2 is therefore usable in projects of the above itemised project types.

8.1.3 *REP v.2 and the nature of the software project*

In REP v.2, requirements are fixed before the implementation of these requirements starts. REP v.2 is therefore applicable with every software project with the nature to have requirements fixed before the implementation starts.

Table 19: Project types characteristics

Project type	Communication	Written requirements
In-house development	Yes	No
Product development	Yes	Yes
Time-and-material based development	Yes	Yes & No
Fully COTS purchase	No	No
Tender	No	Written by the customer himself
Contract development	Yes	Yes
Sub-contracting	Yes	Time-and-material based development: No Requirements based: Yes

When the software project is developed according to software development life cycles as the Waterfall model [33], the V model [15], and the Y model [13], having fixed requirements before implementing is already natural.

Developing software according to software development life cycles as the Spiral model [9], Agile development [7], and Staged development [8], would need to be adjusted to having fixed requirements before starting to implement the software project before REP v.2 can be used in these circumstances.

8.1.4 *REP v.2 and the environmental factors of the organisation*

Kruchten [25] claims that context is extremely important for software processes to become successful. He listed five environmental factors of the organisation that is developing software that influence the processes that could be used. Below, these environmental factors are listed, applied to The Backbone, and discussed in which circumstances REP v.2 can be used:

1. **Business domain:** The Backbone deals in multiple aspects of performance monitoring solutions, namely supply chain performance, IT SLA performance, end-user performance, application performance, identity & access performance, and IT per-

formance [38]. The proposed REP v.2 is especially focussed on end-user performance. However, with only minimal changes (mostly textual) this REP could apply to other the performance aspects of The Backbone as well. For working in completely other business domains, REP v.2 is too specific and therefore will not work. However, according to professionals in RE, the general REP (section 8.2) will work in completely other business domains, for example the business domains of the professionals (section 7.4).

2. **Number of instances:** The current REP practice of The Backbone functions for the few dozen instances of the EUP that are currently implemented. Because The Backbone wants to grow this number, a more structured REP practice is needed [30]. The proposed REP v.2 fulfils this need and will work for a larger number of instances as well. The more structured REP v.2 helps making the EUP product modular and therefore it will be easier to make new instances of the EUP product.

3. **Maturity of organisation:** During this research by The Backbone, it is concluded that REP v.2 works with small enterprises and small enterprises that are moving to become more mature in their processes. From the interviews with experts in RE, described in (section 7.4), it can be concluded that REP v.2 will work after a few changes in larger enterprises with mature processes as well. The generalised REP (section 8.2) includes these changes.

4. **Level of innovation:** The Backbone is a small and resilient organisation who has to adjust its strategy regularly on the basis of the market interests. This innovativeness of The Backbone results in the possibility of new software products in the future in the field of IT monitoring for which REP v.2 can be used. The innovativeness also leads to the probability of the EUP product changing over time. The Backbone should therefore treat the design document (Appendix F), in which REP v.2 is incorporated, as a living document.

5. **Culture:** Currently, the account managers and the performance consultant of The Backbone are very flexible in how they gather requirements from their customers. The proposed REP v.2 allows them to be still very flexible [12], but at the same time it supports them to ask all necessary questions to speed up the REP and deliver a more complete list of requirements.

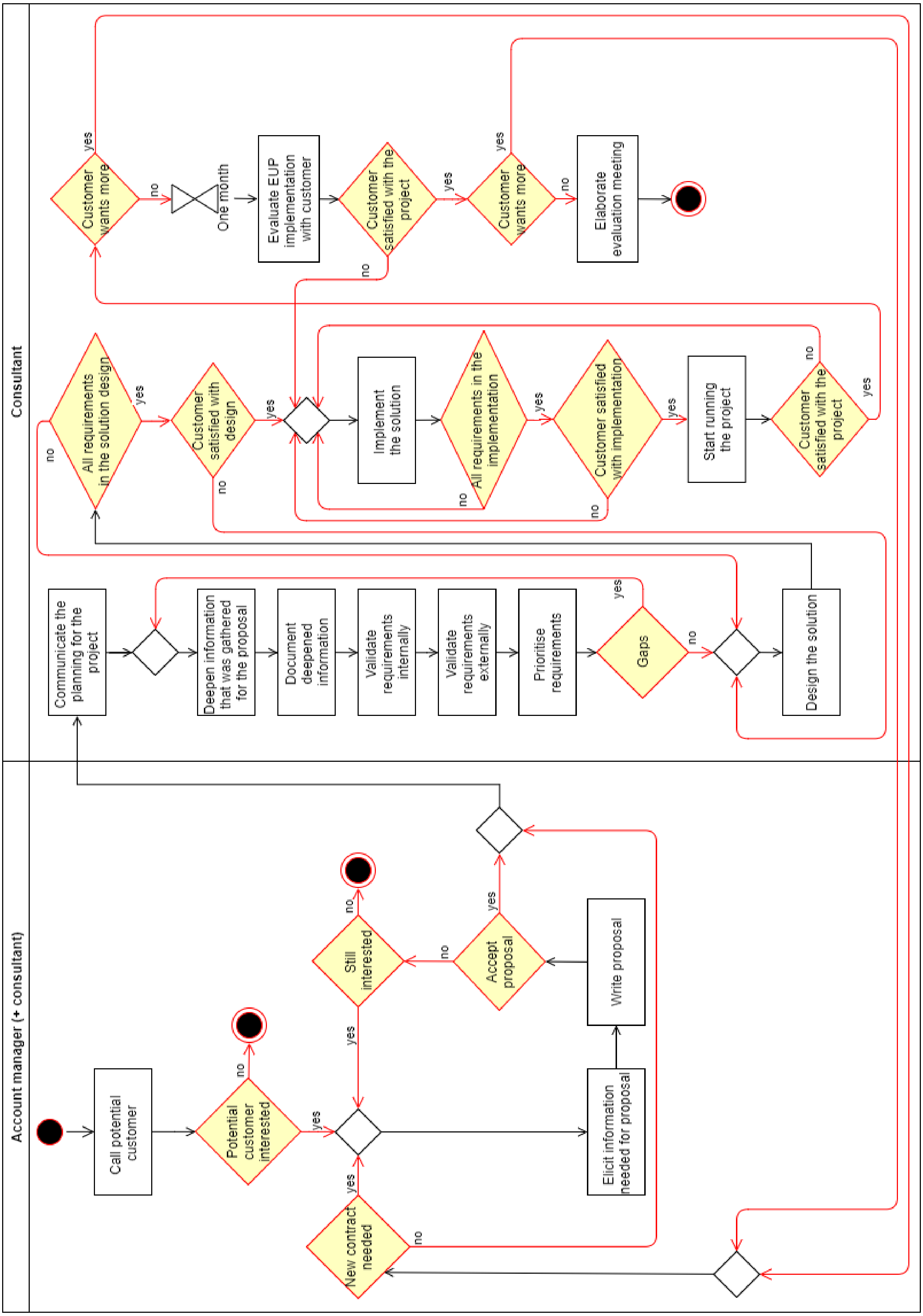
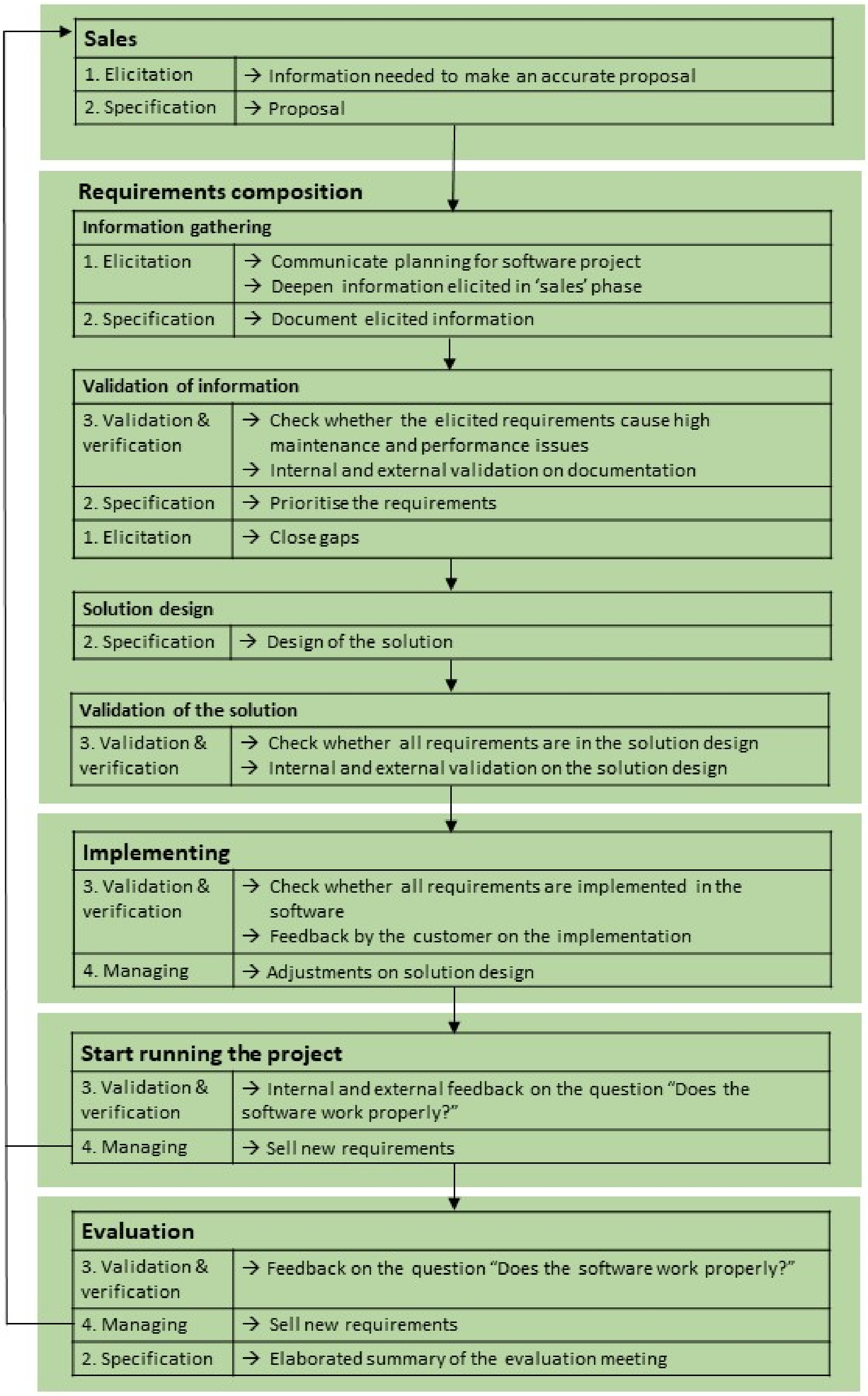

Account manager (+ consultant)
Consultant
Call potential customer
Potential customer interested
no
yes
Still interested
no
yes
Accept proposal
no
yes
Write proposal
Elicit information needed for proposal
New contract needed
yes
no
Communicate the planning for the project
Deepen information that was gathered for the proposal
Document deepened information
Validate requirements internally
Validate requirements externally
Prioritise requirements
Gaps
yes
no
Design the solution
All requirements in the solution design
no
yes
Customer satisfied with design
no
yes
Implement the solution
All requirements in the implementation
no
yes
Customer satisfied with implementation
no
yes
Start running the project
Customer satisfied with the project
no
yes
Customer wants more
yes
no
One month
Evaluate EUP implementation with customer
Customer satisfied with the project
no
yes
Customer wants more
yes
no
Elaborate evaluation meeting

Figure 13: Activity diagram general REP



Sales

1. Elicitation	→ Information needed to make an accurate proposal
2. Specification	→ Proposal

Requirements composition

Information gathering

1. Elicitation	→ Communicate planning for software project → Deepen information elicited in 'sales' phase
2. Specification	→ Document elicited information

Validation of information

3. Validation & verification	→ Check whether the elicited requirements cause high maintenance and performance issues → Internal and external validation on documentation
2. Specification	→ Prioritise the requirements
1. Elicitation	→ Close gaps

Solution design

2. Specification	→ Design of the solution

Validation of the solution

3. Validation & verification	→ Check whether all requirements are in the solution design → Internal and external validation on the solution design

Implementing

3. Validation & verification	→ Check whether all requirements are implemented in the software → Feedback by the customer on the implementation
4. Managing	→ Adjustments on solution design

Start running the project

3. Validation & verification	→ Internal and external feedback on the question "Does the software work properly?"
4. Managing	→ Sell new requirements

Evaluation

3. Validation & verification	→ Feedback on the question "Does the software work properly?"
4. Managing	→ Sell new requirements
2. Specification	→ Elaborated summary of the evaluation meeting

Figure 14: General REP

8.2 GENERALISABILITY OF REP V.2

REP v.2 is very specific and directly applicable for the EUP product of The Backbone. The interviews reported in section 7.4 with the professionals in the field of RE had a dual purpose. The primary purpose was to use their knowledge on RE to validate REP v.2. The second purpose was to compare REP v.2 with the REP practices of the interviewees, in order to discuss its generalisability.

During the interviews reported in section 7.4 it became clear that both professionals in the field of RE could use all elements of REP v.2 after some changes in their own software projects. These changes are already described in detail in section 7.4. With the information gathered from the professionals, REP v.2 is generalised and described.

In Figure 13, the general REP is showed in an activity diagram. This general REP is not tested by interviews, nor in practice due to time constraints. Here, the first actor, the account manager, sells the software project. It can be useful for the account manager to get the help of a consultant for eliciting requirements needed to make an accurate proposal. The second actor, is the consultant who elicits requirements, designs, and implement the solution. The evaluation is best be done by both the account manager and the consultant. In large organisations, both actors can be existing of multiple people, or even teams. The general REP is summarised in Figure 14. The general REP differs on one point from REP v.2: in the *solution design* subphase, there is no elicitation part in the general REP where there is an elicitation part in REP v.2 (Figure 10) for very detailed requirements for the solution design. This elicitation part in the *solution design* subphase is not included in the general REP since this extra step of elicitation is probably not necessary for most other software projects than the EUP.

The general REP is divided into five phases: *sales*, *requirements composition*, *implementing*, *start running the project*, and *evaluation*. In the text below, the general REP is described per (sub)phase in text below:

8.2.1 *Sales*

In the *sales* phase, enough information is elicited to be able to write an accurate proposal. All information gathered gets properly documented. Accordingly, the proposal is sent to the customer. If the customer agrees on the proposal and signs it, the *requirements composition* phase starts.

8.2.2 *Requirements composition*

The requirements composition is divided into the following four subphases: *information gathering*, *validation of information*, *solution design*, and *validation of the solution*.

8.2.2.1 *Information gathering*

This subphase starts with communicating the planning for the project towards the customer. In this planning it is made clear what is expected from which party at which time. Furthermore, the information elicited during the *sales* phase is deepened and documented.

8.2.2.2 *Validation of information*

Accordingly, the information gathered so far is validated internally and externally. During the internal validation it is checked whether the requirements would cause high maintenance and performance issues. During the external validation the customer gives feedback on the requirements so far. The requirements are prioritised for which the MoSCoW-rules can apply (section 7.2). At this moment in the REP, it is also time to check whether there are gaps in the requirements and to close these.

8.2.2.3 *Solution design*

In this subphase, the solution gets designed based on the validated requirements from the *validation of information* subphase.

8.2.2.4 *Validation of solution*

The solution gets validated internally as well as externally in this subphase. During the internal validation it is checked whether all requirements are in the solution design. Next, the customer gives his feedback on the solution design during the external validation.

8.2.3 *Implementation*

In this phase, the software product is developed according to the requirements gathered before. Before running the project, the implementation is validated internally and externally. During the internal validation it is checked whether all requirements are implemented in the software. The external validation is done by the customer giving his feedback on the software. If necessary, adjustments on the solution design can be made.

8.2.4 *Start running the project*

In this phase, the software starts to run. The software is validated externally by the customer giving feedback on its working. The supplier also tries to sell more requirements to the customer.

8.2.5 *Evaluation*

The *evaluation* phase takes place one month after the software product started to run. A meeting takes place between the supplier and the customer to discuss the workings of the software that is made during this REP. During this meeting both parties discuss if the system delivers the correct information. Besides that, the supplier tries to sell new requirements to the customer and to find new requirements for their software. After the meeting, an elaborated summary of the evaluation is made. This *evaluation* phase is repeated every six till twelve months.

9 CONCLUSIONS

This thesis proposes a REP for the EUP product of The Backbone, a small enterprise in The Netherlands. A generalised version of this REP is also proposed to be applicable to other IT companies and software products. This research can be counted as a field study on RE at a small enterprise that provides a mapping between methodical improvements and the type of company they can be applied to [29].

This chapter summarises the findings done during this research, its practical implications, and reflects on the research process of this thesis.

9.1 ANSWERS TO RESEARCH QUESTIONS

In this research, the answers on six research questions are given. The answers to these research questions can be summarised as follows:

1. **RQ1: What would be a good requirements engineering process for the End-User Performance monitoring service of The Backbone based on literature and the current requirements engineering process of The Backbone?** Based on objectives from The Backbone, literature, and findings in the current REP of The Backbone by the author of this thesis, REP v.0 has been designed. REP v.0 exists of five phases based on the phases of the current REP practice at The Backbone: sales, requirements composition, scripting, delivery, and evaluation. Each phase has its own purpose. The requirements composition phase is divided into four different subphases.

   REP v.0 is described in section 5.2 and summarised in Figure 6.

2. **RQ2: How do employees of The Backbone experience the proposed requirements engineering process, does it improve the work?** The performance consultant of The Backbone considers REP v.0 to perform better on all aspects than the current REP practice at The Backbone: usability, usefulness, efficiency, quality of written requirements, implementation quality of requirements, and professional image than the current REP practice at The Backbone.

   The performance consultant suggested some improvements on REP v.0 before it can be used. These improvements consist mostly of terminology changes, adding eliciation of simulation scenarios and simulation steps in the determination of the solution subphase, and process REP v.0 in the design document.

3. **RQ3: What would be a good requirements engineering process for the End-User Performance monitoring service of The Backbone based on the evaluation with employees of The Backbone?** Based on the objectives used to design REP v.0 and findings in REP v.0 by the performance consultant and the author of this thesis, REP v.1 is designed. REP v.1 exists of five phases: sales, requirements composition, implementing, start performance monitoring, and evaluation. Each phase has its own purpose. The requirements composition phase is divided into four different subphases.

   REP v.1 is described in Figure 6.2 and summarised in Figure 8.

4. **RQ4: How do employees and customers of The Backbone experience the updated version of the proposed requirements engineering process, does it improve the work?** The performance consultant of The Backbone thinks that REP v.1 is an improvement of the current REP practice as well as REP v.0. REP v.1 performs better on all aspects than the current REP practice: usability, usefulness, efficiency, quality of written requirements, implementation quality of requirements, and professional image than the current REP practice at The Backbone.

   Both, the performance consultant and the customer liked the structured way of working with REP v.1.

   During the use of REP v.1 with a customer, the performance consultant and the author of this thesis found two improvements on the REP. The first improvement is for the questionnaire to mention that all links between components in the architecture must be visible. The second improvement is to add a possibility to prioritise the components in the component file.

5. **RQ5: What would be a good requirements engineering process for the End-User Performance monitoring service of The Backbone based on the evaluation with employees and customers of The Backbone?** Based on the objectives used to design REP v.0, REP v.1, and findings in REP v.1 by the performance consultant and the author of this thesis, REP v.2 is designed. REP v.2 exists of five phases: sales, requirements composition, implementing, start performance monitoring, and evaluation. Each phase has its own purpose. The requirements composition phase is divided into four different subphases.

   REP v.2 is described in section 7.2 and summarised in Figure 10. Besides the summary, an activity diagram can be found in Figure 9.

6. **RQ6: What do professionals in the field of requirements engineering think of the proposed requirements engineering process, does it work?** REP v.2 is validated by interviewing two

professionals in the field of requirements engineering. Both professionals in the field of requirements engineering believe that REP v.2 performs well on all the following aspects: usability, usefulness, efficiency, quality of written requirements, implementation quality of requirements, and professional image.

The *evaluation* phase of REP v.2 could even improve the REP of one of the interviewees.

The professionals also suggested some improvements for REP v.2. The first improvement is to add a planning with time and tasks to inform the customer what sources you will need from them and what they can expect from you. The second improvement is to add internal validation into REP v.2 and to not only let the customer validate the requirements. The thirds improvement is to repeat the *evaluation* phase every six months till a year. These improvements are embedded into the design of REP v.2 as showed in Figure 11, the summary of REP v.2 as showed in Figure 12, and into the design document can be found in Appendix F.

9.2 PRACTICAL IMPLICATIONS

This research is done to help The Backbone with their REP for their EUP product. Summarised below are the improvements needed on their current REP to design REP v.2, how REP v.2 still can be improved, and some unsolved issues for future work.

9.2.1 *Improvements done on the current REP of The Backbone*

During this research, REP v.2 is designed and evaluated for the EUP product of The Backbone. Since the evaluation of REP v.2 (section 7.4), the current REP practice of The Backbone is replaced with REP v.2 due to the improvements in REP v.2 in respect to the current REP practice.

Besides parts of the current REP practice that needed to be improved, there where also parts of the current REP practice that were kept the way they were. The summaries of both REPs can be found in respectively Figure 2 and Figure 12.

Below, the similarities and improvements of the current REP practice that led to the design of REP v.2 are summarised:

9.2.1.1 *Similarities*

The similarities below include parts that are renamed but still have more or less the same goal.

- The phases

- The amount of subphases
- The *sales* phase
- The *implementation* phase
- The *start performance monitoring* phase

9.2.1.2 *Improvements*

The improvements below are the changes of the current REP practice at The Backbone that led to REP v.2.

- The goals of the subphases of the *requirements composition* phase
- The addition of the questionnaire
- The addition of the component file
- The addition of the notification table
- The addition of the dashboard table
- REP v.2 is more detailed
- Validation of information gathered in the *requirements composition* phase
- The addition of the an elaborated summary in the *evaluation* phase

9.2.2 *Improvements for REP v.2*

During evaluation interviews with professionals in the field of RE, some improvements for REP v.2 were found. These changes are already processed in the design of REP v.2 as showed in Figure 11, the summary of REP v.2 as showed in Figure 12, and in the design document (Appendix F). They are also described below:

- A planning should be added to the REP with times and tasks to inform the customer what sources you will need from them and what they can expect from you.
- Internal validations should be added to REP v.2. Feedback from the customer (external validation) is not good enough, as a supplier you should also validate the requirements by yourself.
- The *evaluation* phase should be repeated every six months till a year to keep yourself up to date about how your product performs in the market.

9.2.3 *Future work*

REP v.2 is especially designed for the EUP product of The Backbone. In section 8.2, REP v.2 is generalised. However, this general REP has not been tested yet and can therefore be improved. Below, these improvements are listed:

DESIGN A REP FOR ANOTHER SOFTWARE PROJECT AT ANOTHER SOFTWARE DEVELOPING ORGANISATIONS AND COMPARE THIS TO REP V.2 OF THE BACKBONE AND THE GENERAL REP To improve the general REP, it is helpful to first design another REP for another software project at another software developing organisation. Accordingly, this REP can be compared to REP v.2 for the EUP product of The Backbone and the general REP. The elements that are in both the REP for another software project at another organisation and in REP v.2 are important to have in the general REP as well.

ELABORATE THE GENERAL REP IN MORE DETAIL The general REP as described in section 8.2 is very general. The general REP would probably be easier to use for software development organisations if the REP has more details. Since the growth of details also leads to a more specific REP and therefore harder to implement for organisations, creating a modular REP would probably help. Designing multiple REPs for other software project in different software developing organisations would help for this purpose.

TEST THE GENERAL REP ON MULTIPLE ASPECTS As already mentioned before, the general REP is not tested in practice yet. To test whether this general REP is an improvement of the current REP of the organisation, the general REP should be tested on how this REP performs on aspects like: usability, usefulness, time efficiency, quality of written requirements, implementation quality of requirements, and professional image.

TEST THE GENERAL REP IN MULTIPLE SOFTWARE DEVELOPING ORGANISATIONS AND SOFTWARE PROJECTS REP v.2 is tested only in The Backbone since it is a very specific REP that can only be used for one product in The Backbone. This is not the case with the general REP. Therefore, the general REP should be tested in multiple software developing organisations. Preferably, in organisations and in software projects that differ in size.

TEST THE GENERAL REP IN DIFFERENT PROJECT TYPES In subsection 8.1.2, the working of REP v.2 in different project types defined by [26] is described. This is all theoretical and not tested in real cases.

The general REP would be perfect for this task, since it is usable in more software projects than REP v.2.

9.3 REFLECTION ON THE RESEARCH PROCESS

Below, some thoughts of the author are described on the research process. First of all: was the use of the DSRM of Peffers et al. [31] helpful for the author during this process? And second, some figments of the author about why the inefficiencies found during this research at the current REP practice of The Backbone for their EUP product were not solved before.

9.3.1 *Using the DSRM during this research*

The use of the DSRM of Peffers et al. (section 4.1) was really helpful for the author during this research [31]. Figure 3 of the DSRM was very intuitive to read and to apply the methodology to this research. The DSRM gave structure to the design chapters (chapter 5, chapter 6, chapter 7) from the start. During these five months, the author therefore always knew what the next step in the research process was and what to do in those steps. The practical context of this research was fitted well within the DSRM. After designing a new artefact a demonstration has to be given before evaluating the artefact. This helped the author of this thesis to test the artefact on completeness before evaluating it in the real world. The author believes that this demonstration phase of the DSRM has therefore led to a better artefact than when a methodology was used without a demonstration phase. The DSRM was easy applicable to this research due to its intuitive and practical nature. As Bayazit [6] states, "design research is a systematic search and acquisition of knowledge related to design and design activity". There were some challenges in the process of acquiring the needed knowledge and executing the design activity, that were specific to the context of small IT companies. For example, dealing with undocumented assumptions and pieces of knowledge. Therefore, the author of this thesis took extra steps in order to make the design science process work. These steps are:

- doing a triangulation research consisting of interviews, observations, and documentary research to figure out the current REP practice of The Backbone section 3.2,
- interview professionals in the field of RE from other organisations than The Backbone to evaluate the proposed REP v.2 section 7.4.

These steps were necessary because there were very limited knowledge resources. Only one employee, the performance consultant, had the knowledge the author needed for this research. Therefore, multiple ways of collecting this knowledge have been used to be independent of this employee's memory. The limited knowledge resource at The Backbone was also the reason to ask professionals in the field of RE from other organisations to evaluate the proposed REP v.2. REP v.0 was evaluated using an interview with the performance consultant of The Backbone and REP v.1 was evaluated using REP v.2 in a real case during a meeting between the performance consultant and a customer. Interviewing professionals from other organisations gave new insights to improve REP v.2 to its final form as can be found in Figure 11.

The author therefore would recommend to use the DSRM of Peffers et al. in design science researches with a high practical impact in the field of RE.

9.3.2 *Inefficiencies found in the current REP practice*

This research gives a look into the REP practices at a small IT company. It shows the kind of inefficiencies that occur when there is taken no time to analyse the REP. Examples are:

- The customer is not encouraged to think of important parts of the software before the meeting takes place and therefore has a limited idea what he wants.
- The customer still has to collect the information needed for the meeting during the meeting which leads to less time for the actual software project.
- Topics discussed during meetings are not all processed in the software design and into the software itself since checklists for this are not part of the REP.
- During the evaluation meeting, the consultant and account manager make promises towards the customer and the customer makes promises towards the consultant and account manager, but those promises are forgotten afterwards since there is no elaboration of field notes made during the evaluation meeting.

The existence of the EUP is not very long at the moment of writing this thesis. The Backbone was therefore still finding its way in the REP. This research and the design of REP v.2 helped The Backbone to speed up this process towards a REP that performs well on aspects like: usability, usefulness, efficiency, quality of written requirements, implementation quality of requirements, and professional image.

Looking back, the inefficiencies are unsurprising given the specific organisational context in which RE takes place. It seems that even if a company hires bright specialists with higher education degrees in IT, composing a team of bright individuals does not always yield a smooth instantiation of a REP model [3]. Moreover, The Backbone operates in dynamic markets and experiences times of high demand in which resources are pooled up to serve a high number of clients, alternating with times of low demand when all the organisational energy goes to acquiring new clients. The rapid transition of priorities is justified from a business perspective. However, it has the side effect that systematic processes such as RE get less attention than what the company wants to give them.

10 RECOMMENDATIONS FOR THE BACKBONE

REP v.2 is designed in this thesis especially for the EUP monitoring service of The Backbone and is therefore directly applicable for them. However, there are some recommendations for the future:

- First of all, it is very important to treat the design document of REP v.2 (Appendix F) as a living document. It might be useful to make changes in this document over time, it should therefore be updated when needed.
- Second, a software tool can be built around REP v.2 that makes it easy by the use of clickpaths to search in big EUP projects for example. This software tool should be built very modular, if the design document changes (since it is a living document) it should not be take a lot of time to process those change it in the software tool as well.
- Third, if a software tool is built for REP v.2 this software tool should be linked with a database to be able to compare requirements and evaluation to each other.
- Fourth, artificial intelligence can be used to expand the software tool even more. This artificial intelligence can be used to compare requirements and evaluations automatically to each other. This may lead to new product requirements for the EUP product.
- Finally, REP v.2 can also be used in its general form (section 8.2) for other custom software projects inside The Backbone.

BIBLIOGRAPHY

[1] A. J. Albrecht, „Measuring application development productivity", in *Proceedings of the Joint SHARE/GUIDE/IBM Application Development Symposium*, 1979, pp. 83–92.

[2] M. Aoyama, T. Nakatani, S. Saito, M. Suzuki, K. Fujita, H. Nakazaki and R. Suzuki, „A Model and Architecture of REBOK (Requirements Engineering Body of Knowledge) and Its Evaluation", in *Software Engineering Conference (APSEC), 2010 17th Asia Pacific*, IEEE, 2010, pp. 50–59, ISBN: 978-0-7695-4266-9. DOI: 10.1109/APSEC.2010.16.

[3] J. Aranda, S. Easterbrook and G. Wilson, „Requirements in the wild: How small companies do it", in *Requirements Engineering Conference, 2007. RE'07. 15th IEEE International*, IEEE, 2007, pp. 39–48, ISBN: 0-7695-2935-6. DOI: 10.1109/RE.2007.54. [Online]. Available: http://ieeexplore.ieee.org/document/4384165/.

[4] A. Aurum and C. Wohlin, *Engineering and Managing Software Requirements*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2005, ISBN: 9783540250432.

[5] I. Basharat, M. Fatima, R. Nisa, R. Hashim and A. Khanum, „Requirements Engineering Practices In Small and Medium Software Companies: An Empirical Study", in *Science and Information Conference (SAI), 2013*, IEEE, 2013, pp. 218–222.

[6] N. Bayazit, „Investigating Design: A Review of Forty Years of Design Research", *Design Issues*, vol. 20, no. 1, pp. 16–29, 2004, ISSN: 0747-9360. DOI: 10.1162/074793604772933739. [Online]. Available: http://www.mitpressjournals.org/doi/pdf/10.1162/074793604772933739.

[7] K. Beck, M. Beedle, A. van Bennekum *et al.*, *Manifesto for Agile Software Development*, 2001. [Online]. Available: http://agilemanifesto.org/ (visited on 18/11/2017).

[8] K. Bennett and V. Rajlich, „Software Maintenance and Evolution: A Roadmap", in *Proceedings of the Conference on the Future of Software Engineering*, vol. 225, ACM, 2000, pp. 73–87, ISBN: 1-58113-253-0. DOI: 10.1145/336512.336534. [Online]. Available: http://portal.acm.org/citation.cfm?doid=336512.336534.

[9] B. W. Boehm, „A Spiral Model of Software Development and Enhancement", *Computer*, vol. 21, no. 5, pp. 61–72, 1988, ISSN: 00189162. DOI: 10.1109/2.59. arXiv: arXiv:1011.1669v3.

[10] B. W. Boehm, „Understanding and Controlling Software Costs", *Journal of Parametrics*, vol. 8, no. 1, pp. 32–68, 1988. [Online]. Available: http://sunset.usc.edu/TECHRPTS/1986/usccse86-501/usccse86-501.pdf.

[11] P. Bourque and R. E. Fairley, *SWEBOK Guide V3.0*. IEEE Computer Society Press, 2014, ISBN: 0-7695-2330-7. DOI: 10.1234/12345678. arXiv: arXiv:1210.1833v2. [Online]. Available: www.swebok.org.

[12] S. Bürsner and T. Merten, „RESC 2010: 1st Workshop on Requirements Engineering in Small Companies", in *16th International Working Conference on Requirements Engineering: Foundation for Software Quality*, 2010, pp. 128–130.

[13] L. F. Capretz, „Y: A New Component-Based Software Life Cycle Model", *Journal of Computer Science*, vol. 1, no. 1, pp. 76–82, 2005, ISSN: 15493636. DOI: 10.3844/jcssp.2005.76.82. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.621.8303&rep=rep1&type=pdf.

[14] J. Coughlan and R. D. Macredie, „Effective Communication in Requirements Elicitation: A Comparison of Methodologies", *Requirements Engineering*, vol. 7, no. 2, pp. 47–60, 2002, ISSN: 0947-3602. DOI: 10.1007/s007660200004.

[15] K. Forsberg and H. Mooz, „The Relationship of System Engineering to the Project Cycle", in *The 12th INTERNET World Congress on Project Management*, 1994, p. 12, ISBN: 2334-5837. DOI: 10.1002/j.2334-5837.1991.tb01484.x. [Online]. Available: http://ife2010.wikispaces.com/file/view/SE+&Project+Cycle,+Forsberg&Mooz,+1995.pdf.

[16] A. Forward and T. C. Lethbridge, „The relevance of software documentation, tools and technologies: a survey", in *Proceedings of the 2002 ACM symposium on Document engineering*, ACM, 2002, pp. 26–33.

[17] M. Fowler and J. Highsmith, „The agile manifesto", *Software Development*, vol. 9, no. 8, pp. 28–35, 2001.

[18] A. Gopal, T. Mukhopadhyay and M. S. Krishnan, „The role of software processes and communication in offshore software development", *Communications of the ACM*, vol. 45, no. 4, pp. 193–200, 2002.

[19] L. Groves, R. Nickson, G. Reeve, S. Reeves and M. Utting, „A survey of software development practices in the New Zealand software industry", in *Software Engineering Conference, 2000. Proceedings. 2000 Australian*, IEEE, 2000, pp. 189–201, ISBN: 0769506313. DOI: 10.1109/ASWEC.2000.844576. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=844576.

[20] A. M. Hickey and A. M. Davis, „Elicitation Technique Selection: How Do Experts Do It?”, in *Requirements engineering conference, 2003. Proceedings. 11th IEEE International*, IEEE, 2003, pp. 168–178.

[21] H. F. Hofmann and F. Lehner, „Requirements Engineering as a Success Factor in Software Projects”, *IEEE Software*, vol. 18, no. 4, pp. 58–66, 2001, ISSN: 07407459. DOI: 10.1109/MS.2001.936219. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=936219.

[22] IEEE, *830-1998 - IEEE Recommended Practice for Software Requirements Specifications*, 1998. DOI: 10.1109/IEEESTD.1998.88286.

[23] E. Kamsties, K. Hörmann and M. Schlich, „Requirements Engineering in Small and Medium Enterprises”, *Requirements Engineering*, vol. 3, no. 2, pp. 84–90, 1998, ISSN: 0947-3602. DOI: 10.1007/BF02919967. [Online]. Available: http://link.springer.com/article/10.1007/BF02919967.

[24] R. E. Kraut and L. A. Streeter, „Coordination in software development”, *Communications of the ACM*, vol. 38, no. 3, pp. 69–82, 1995.

[25] P. Kruchten, „Contextualizing agile software development”, *Journal of Software: Evolution and Process*, vol. 25, no. 4, pp. 351–361, 2013, ISSN: 20477473. DOI: 10.1002/smr.572. arXiv: 1408.1293. [Online]. Available: http://search.ebscohost.com/login.aspx?direct=true&db=iih&AN=87605426&lang=fr&site=ehost-live.

[26] S. Lauesen, *Software Requirements: Styles and Techniques*. Bungay: Pearson Education, 2002.

[27] G. C. Low and D. R. Jeffrey, „Function Points in the Estimation and Evaluation of the Software Process”, *IEEE Transactions on Software Engineering*, vol. 16, no. 1, pp. 64–71, 1990.

[28] D. Mairiza, D. Zowghi and V. Gervasi, *Requirements Engineering*, D. Zowghi and Z. Jin, Eds., ser. Communications in Computer and Information Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, vol. 432, pp. 31–44, ISBN: 978-3-662-43609-7. DOI: 10.1007/978-3-662-43610-3. [Online]. Available: https://link.springer.com/chapter/10.1007%2F978-3-662-43610-3_3.

[29] T. Merten, K. Lauenroth and S. Bürsner, „Towards a New Understanding of Small and Medium Sized Enterprises in Requirements Engineering Research”, *Requirements Engineering: Foundation for Software Quality*, pp. 60–65, 2011, ISSN: 03029743 16113349. DOI: 10.1007/978-3-642-19858-8_7. [Online]. Available: http://link.springer.com/10.1007/978-3-642-19858-8_7.

[30] U. Nikula, J. Sajaniemi and H. Kälviäinen, *A State-of-the-Practice Survey on Requirements Engineering in Small-and-Medium-Sized-Enterprises*. Lappeenranta University of Technology Lappeenranta, Finland, 2000, ISBN: 951 - 764 - 431 - 0.

[31] K. Peffers, T. Tuunanen, M. Rothenberger and S. Chatterjee, „A Design Science Research Methodology for Information Systems Research", *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–78, 2007, ISSN: 0742-1222. DOI: 10.2753/MIS0742-1222240302. arXiv: z0022.

[32] J. Robertson and S. Robertson, „Volere", *Requirements Specification Templates*, 2000.

[33] W. W. Royce, „Managing the development of large software systems: concepts and techniques", in *Proceedings of the 9th international conference on Software Engineering*, IEEE Computer Society Press, 1987, pp. 328–338, ISBN: 0897912160. DOI: 10.1016/0378-4754(91)90107-E. [Online]. Available: http://www.pi.informatik.tu-darmstadt.de/fileadmin/user_upload/Group_PI/LV__SE_RE/R_01_Wasserfallmodell__Folien__Schwaiger.pdf.

[34] E. E. Rudolph and G. Simpson, „Evaluation of a fourth generation language", in *Proceedings of ACS and IFIP Joint Symposium on Information Systems*, 1984, pp. 148–165.

[35] H. Saiedian and R. Dale, „Requirements engineering: making the connection between the software developer and customer", *Information and Software Technology*, vol. 42, no. 6, pp. 419–428, 2000.

[36] The Backbone, *How does it work*, 2017. [Online]. Available: https://www.enduserperformance.com/how-does-it-work/ (visited on 25/04/2017).

[37] ——, *Referenties*, 2017. [Online]. Available: http://www.thebackbone.nl/referenties (visited on 18/05/2017).

[38] ——, *The Backbone uw monitoring specialist*, 2017. [Online]. Available: http://www.thebackbone.nl/ (visited on 18/05/2017).

[39] P. Verschuren and J. Doorewaard, *Het ontwerpen van een onderzoek*. Utrecht: Lemma, 1995, ISBN: 9051895461.

[40] H. van Vliet and S. Brinkkemper, „Requirements Engineering", *Bedrijfskunde*, vol. 74, no. 1, pp. 19–29, 2002.

[41] B. J. J. Voigt, „Dynamic System Development Method", Department of Information Technology, University of Zurich, Zurich, Tech. Rep., 2004. [Online]. Available: https://files.ifi.uzh.ch/rerg/arvo/courses/seminar_ws03/14_Voigt_DSMD_Ausarbeitung.pdf.

A DEMONSTRATION CASE

This appendix provides the context of the case study used in:

- section 5.3, page 34
- section 6.3, page 55
- section 7.3, page 75

A couple of months ago, an account manager of The Backbone called the application manager of organisation ABC to inform him about the EUP product and what the EUP could mean for ABC. At that moment, ABC did not experience any performance issues and there was also no money available for an extra IT project.

Now, a few months later, ABC has purchased a new Customer Relationship Management (CRM) application. Since this CRM application is implemented, end-users complain about the CRM application being slow. The CRM application is linked to their Order Management System (OMS) and their Enterprise Resource Planning (ERP) application. All information about the customers of ABC can therefore be found in this CRM application.

The CRM application is mostly approached via a Server Based Computing (SBC) session. Hundreds of users make use of the CRM application every day.

It is very important for organisation ABC that customer information can be created and searched through very fast. The application manager of ABC wants to know whether the CRM application is slow or if one of the linked applications does not perform well.

Since the application manager of ABC had contact with an account manager of The Backbone in the past about a product that might help ABC at this point, the application manager of ABC calls the account manager of The Backbone for further information about the EUP.

B
QUESTIONNAIRE REP V.0

Dutch:

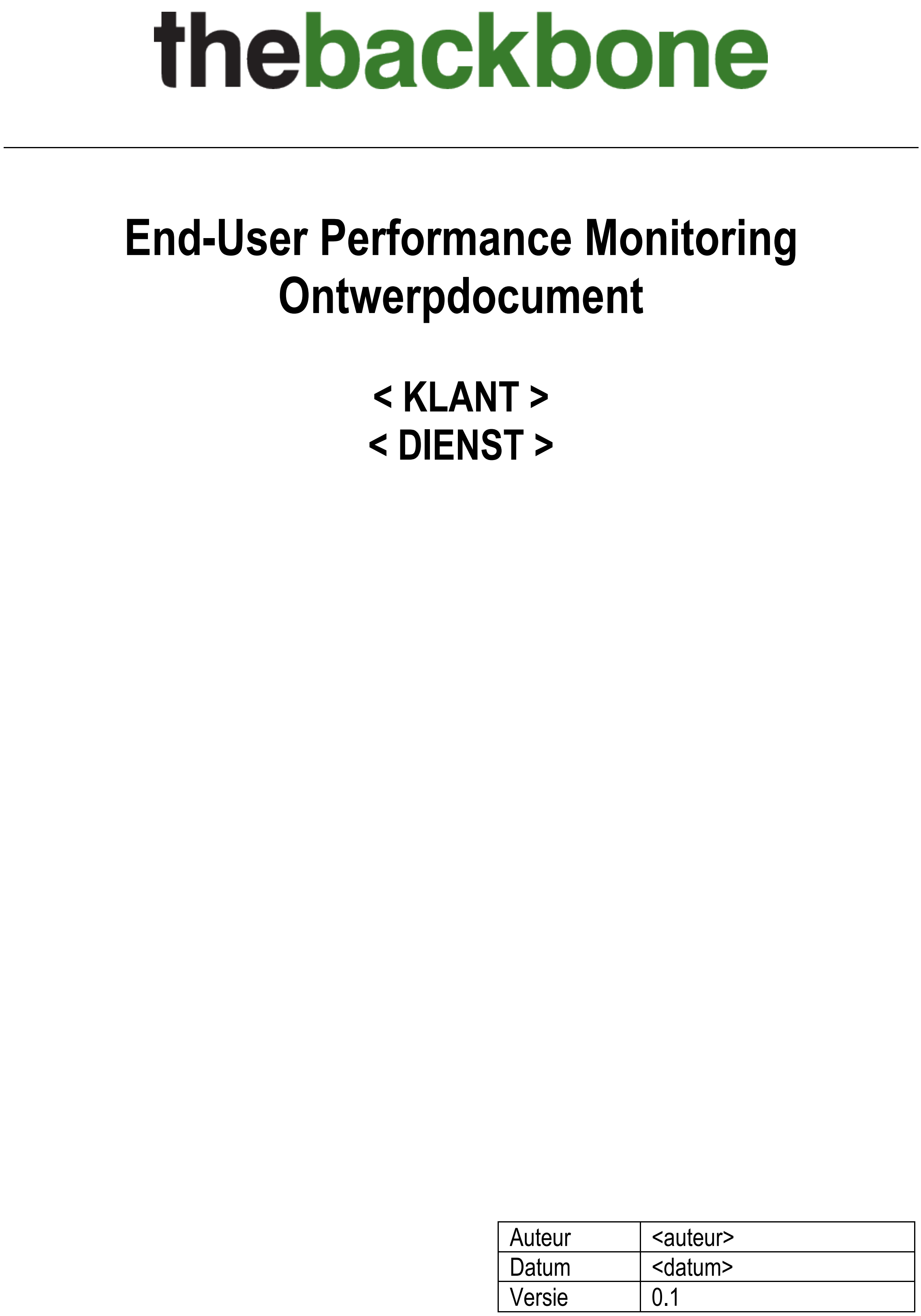
1. Waarom heeft u de EUP aangeschaft?
2. Welke applicaties zou u graag willen monitoren? (voorbeelden zijn: Outlook, Hix, website: http://www.thebackbone.nl)
3. Van welke acties in deze applicaties wilt u de performance monitoren?
4. Wilt u SLAs monitoren? Zo ja, voegt u deze dan alstublieft toe als bijlage.
5. Hoe zijn de te monitoren applicaties met elkaar gekoppeld? (Hoe ziet de architectuur eruit, waar wordt de benodigde data vandaan gehaald, etc.?)

English:

1. Why did you purchase the EUP?
2. Which applications would you like to monitor? (Examples are: Outlook, Hix, Website: http://www.thebackbone.nl)
3. Of which actions in these applications do you want to monitor the performance?
4. Do you want to monitor SLAs? If so, please add them as an attachment.
5. How are the applications that need to be monitored linked? (What is the architecture like, where does the required data come from, etc.?)

C

DESIGN DOCUMENT CURRENT REP PRACTICE

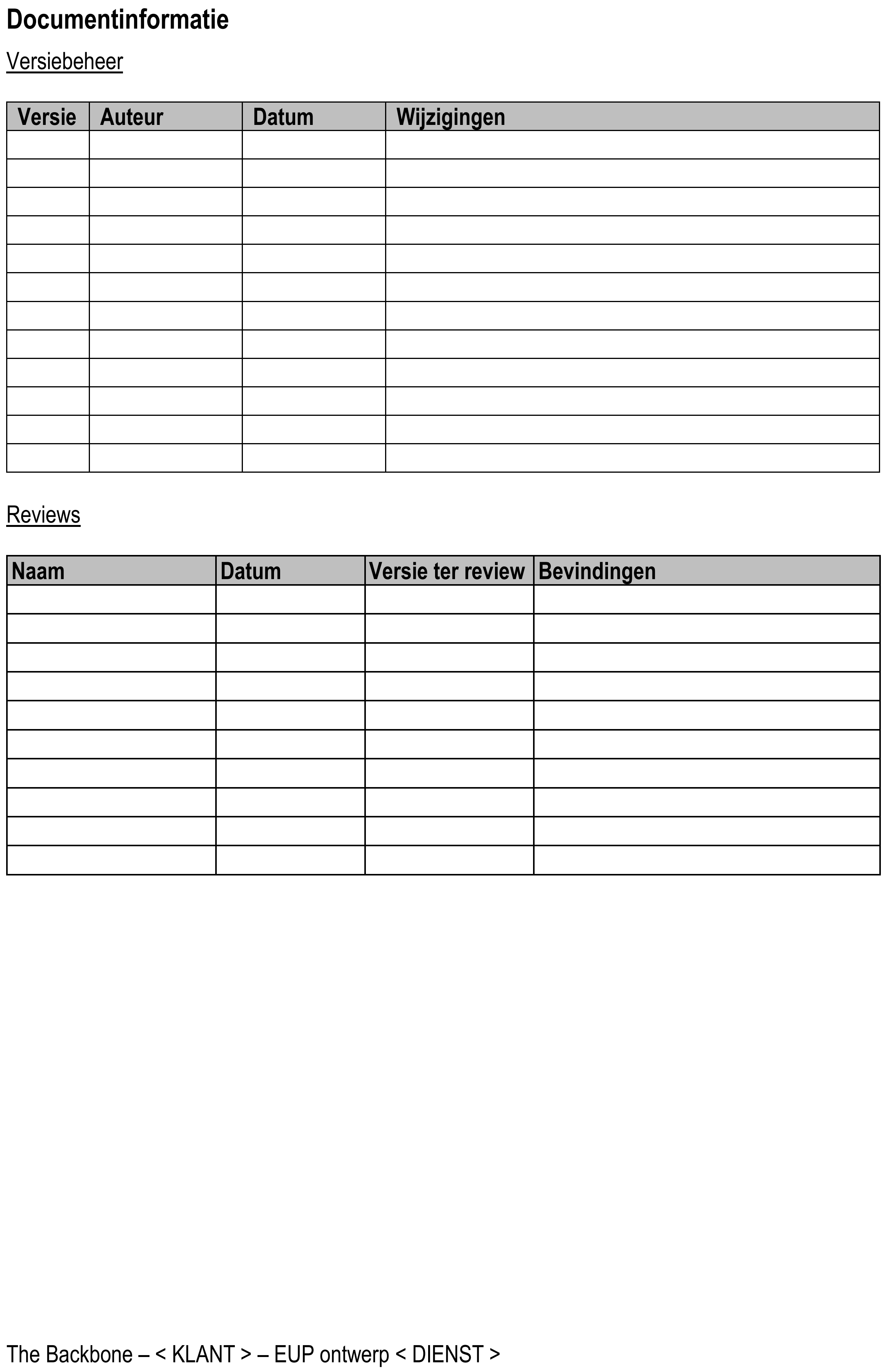


thebackbone

End-User Performance Monitoring Ontwerpdocument

< KLANT >
< DIENST >

Auteur	<auteur>
Datum	<datum>
Versie	0.1



Documentinformatie

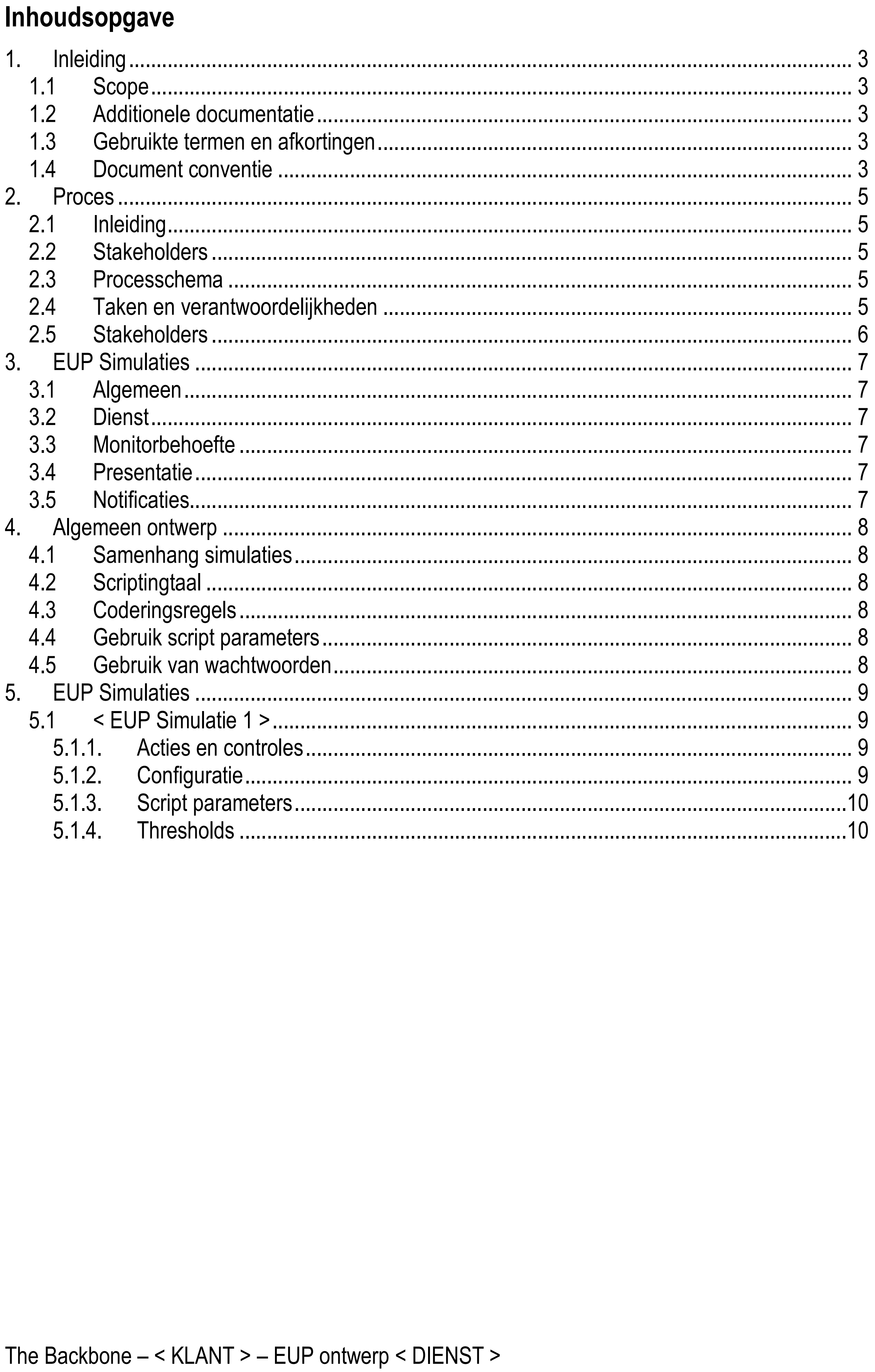
Versiebeheer

Versie	Auteur	Datum	Wijzigingen

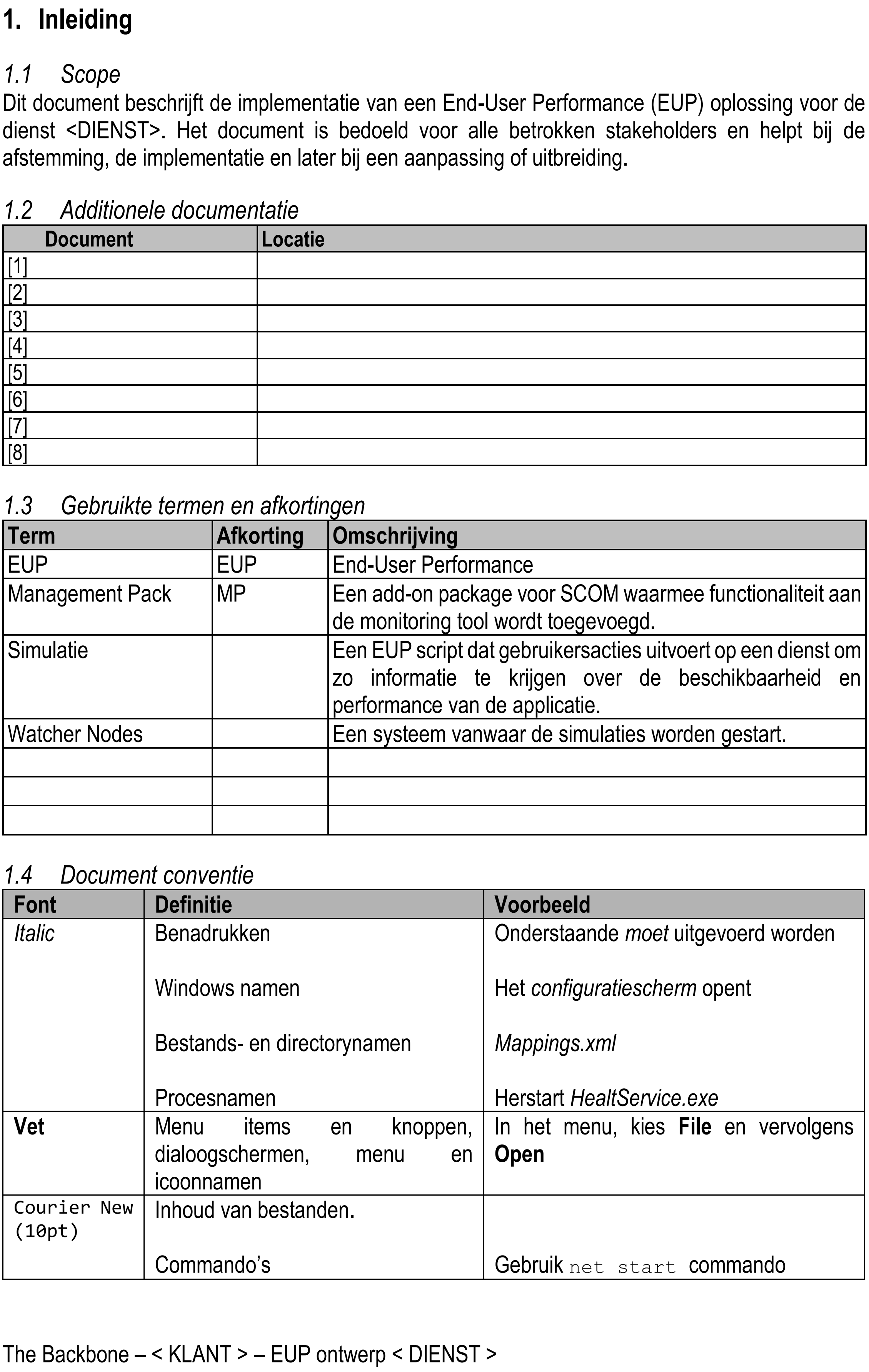
Reviews

Naam	Datum	Versie ter review	Bevindingen

The Backbone – < KLANT > – EUP ontwerp < DIENST >



Inhoudsopgave



1. Inleiding

1.1 Scope

Dit document beschrijft de implementatie van een End-User Performance (EUP) oplossing voor de dienst <DIENST>. Het document is bedoeld voor alle betrokken stakeholders en helpt bij de afstemming, de implementatie en later bij een aanpassing of uitbreiding.

1.2 Additionele documentatie

Document	Locatie
[1]	
[2]	
[3]	
[4]	
[5]	
[6]	
[7]	
[8]	

1.3 Gebruikte termen en afkortingen

Term	Afkorting	Omschrijving
EUP	EUP	End-User Performance
Management Pack	MP	Een add-on package voor SCOM waarmee functionaliteit aan de monitoring tool wordt toegevoegd.
Simulatie		Een EUP script dat gebruikersacties uitvoert op een dienst om zo informatie te krijgen over de beschikbaarheid en performance van de applicatie.
Watcher Nodes		Een systeem vanwaar de simulaties worden gestart.

1.4 Document conventie

Font	Definitie	Voorbeeld
Italic	Benadrukken Windows namen Bestands- en directorynamen Procesnamen	Onderstaande *moet* uitgevoerd worden Het *configuratiescherm* opent *Mappings.xml* Herstart *HealtService.exe*
Vet	Menu items en knoppen, dialoogschermen, menu en icoonnamen	In het menu, kies **File** en vervolgens **Open**
`Courier New (10pt)`	Inhoud van bestanden. Commando's	 Gebruik `net start` commando

2. Proces

2.1 Inleiding

Het ontwerpen en implementeren van een EUP oplossing wordt volgens een vast proces uitgevoerd. Dit borgt de kwaliteit van de EUP oplossing en de betrokkenheid van de stakeholders.

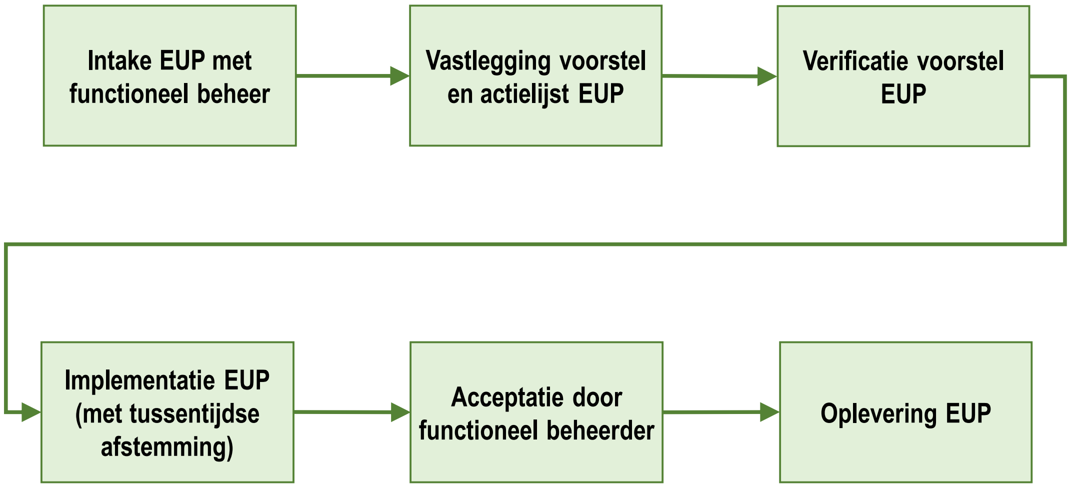
2.2 Stakeholders

Onderstaande tabel geeft een overzicht van de stakeholders en de rol in het proces:

Wie	Rol
< Klant >	Eigenaar van de te monitoren dienst. Opdrachtgever voor het implementeren van een EUP oplossing. Kennis van de dienst dat nodig is voor het ontwerpen van de EUP Simulatie. Schakel tussen alle partijen.
The Backbone	Begeleid het proces om het DAM ontwerp tot stand te laten komen, verantwoordelijk voor het uitwerken van het DAM ontwerp en de daadwerkelijke bouw en inrichting van de DAM

2.3 Processchema

Het schema hieronder toont het proces van een EUP implementatie.


Intake EUP met functioneel beheer
Vastlegging voorstel en actielijst EUP
Verificatie voorstel EUP
Implementatie EUP (met tussentijdse afstemming)
Acceptatie door functioneel beheerder
Oplevering EUP

Het is belangrijk dat pas met de volgende stap wordt gestart als de vorige is afgerond. Dit om te voorkomen dat er onnodig werk wordt verricht.

2.4 Taken en verantwoordelijkheden

Onderstaande tabel geeft de stappen weer binnen het proces met de betrokken / verantwoordelijke stakeholder(s).

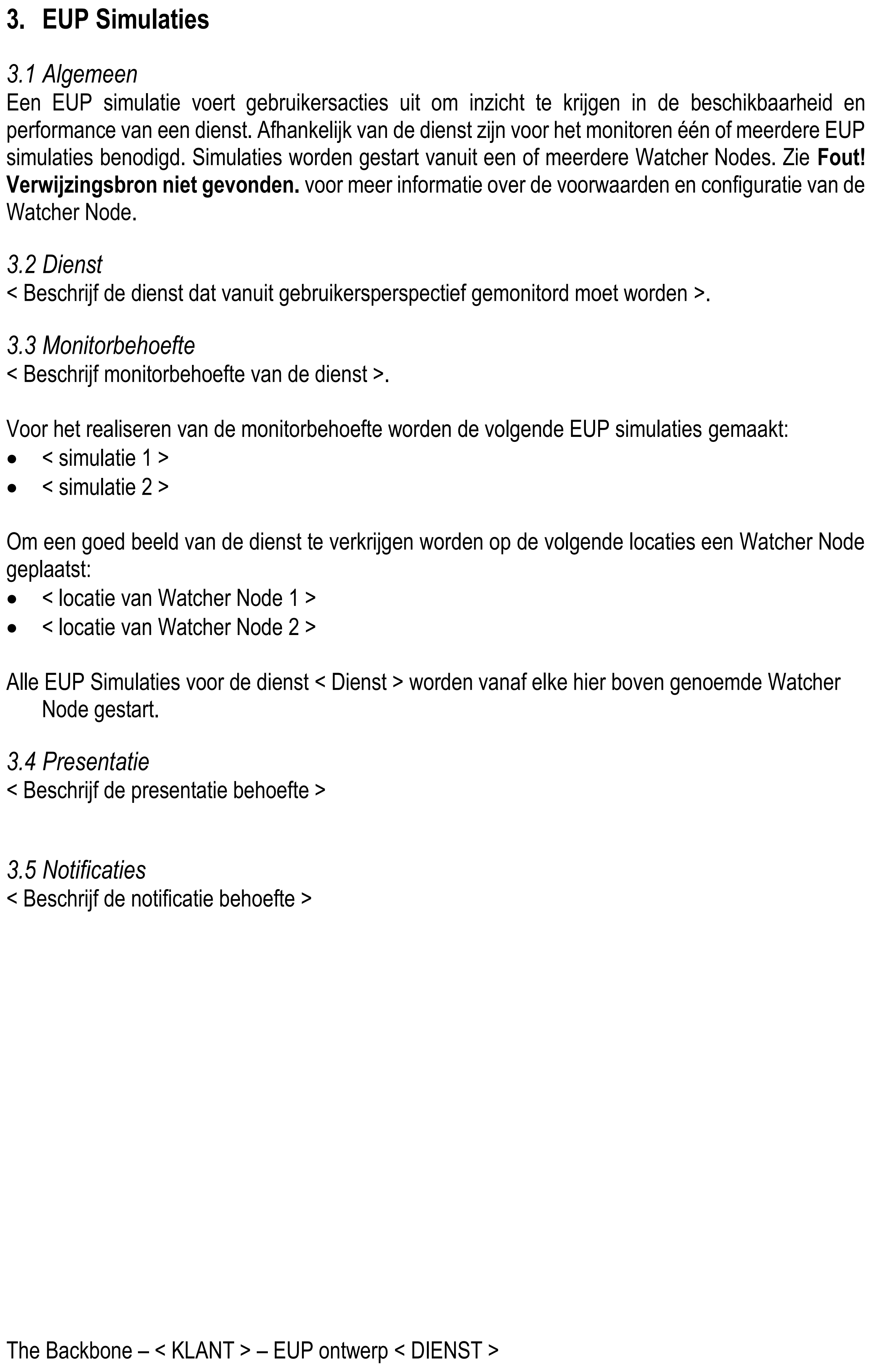
Nr.	Wat	Wie
1.	Intake EUP met functioneel beheer	The Backbone < Klant >
2.	Vastlegging voorstel en actielijst EUP	The Backbone

Nr.	Wat	Wie
3.	Verificatie voorstel EUP	< Klant >
4.	Implementatie EUP (met tussentijdse afstemming)	The Backbone (< Klant >)
5.	Acceptatie door functioneel beheer	< Klant >
6.	Oplevering EUP	The Backbone < Klant >

2.5 Stakeholders

Onderstaande personen zijn betrokken bij de ontwikkeling van het EUP monitoring:

Naam	Afdeling	Rol / functie



3. EUP Simulaties

3.1 Algemeen

Een EUP simulatie voert gebruikersacties uit om inzicht te krijgen in de beschikbaarheid en performance van een dienst. Afhankelijk van de dienst zijn voor het monitoren één of meerdere EUP simulaties benodigd. Simulaties worden gestart vanuit een of meerdere Watcher Nodes. Zie **Fout! Verwijzingsbron niet gevonden.** voor meer informatie over de voorwaarden en configuratie van de Watcher Node.

3.2 Dienst

< Beschrijf de dienst dat vanuit gebruikersperspectief gemonitord moet worden >.

3.3 Monitorbehoefte

< Beschrijf monitorbehoefte van de dienst >.

Voor het realiseren van de monitorbehoefte worden de volgende EUP simulaties gemaakt:

- < simulatie 1 >
- < simulatie 2 >

Om een goed beeld van de dienst te verkrijgen worden op de volgende locaties een Watcher Node geplaatst:

- < locatie van Watcher Node 1 >
- < locatie van Watcher Node 2 >

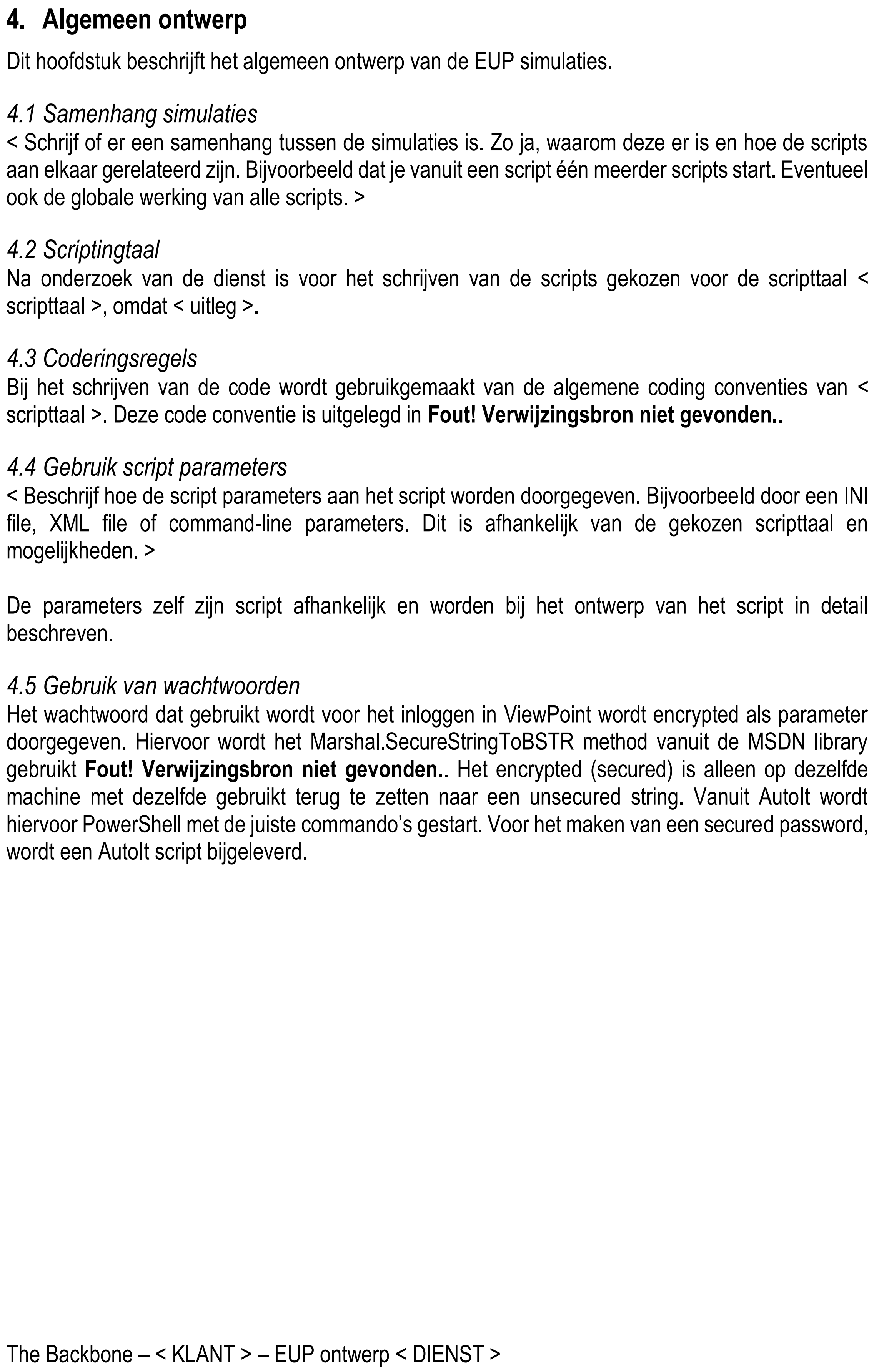
Alle EUP Simulaties voor de dienst < Dienst > worden vanaf elke hier boven genoemde Watcher Node gestart.

3.4 Presentatie

< Beschrijf de presentatie behoefte >

3.5 Notificaties

< Beschrijf de notificatie behoefte >



4. Algemeen ontwerp

Dit hoofdstuk beschrijft het algemeen ontwerp van de EUP simulaties.

4.1 Samenhang simulaties

< Schrijf of er een samenhang tussen de simulaties is. Zo ja, waarom deze er is en hoe de scripts aan elkaar gerelateerd zijn. Bijvoorbeeld dat je vanuit een script één meerder scripts start. Eventueel ook de globale werking van alle scripts. >

4.2 Scriptingtaal

Na onderzoek van de dienst is voor het schrijven van de scripts gekozen voor de scripttaal < scripttaal >, omdat < uitleg >.

4.3 Coderingsregels

Bij het schrijven van de code wordt gebruikgemaakt van de algemene coding conventies van < scripttaal >. Deze code conventie is uitgelegd in **Fout! Verwijzingsbron niet gevonden.**.

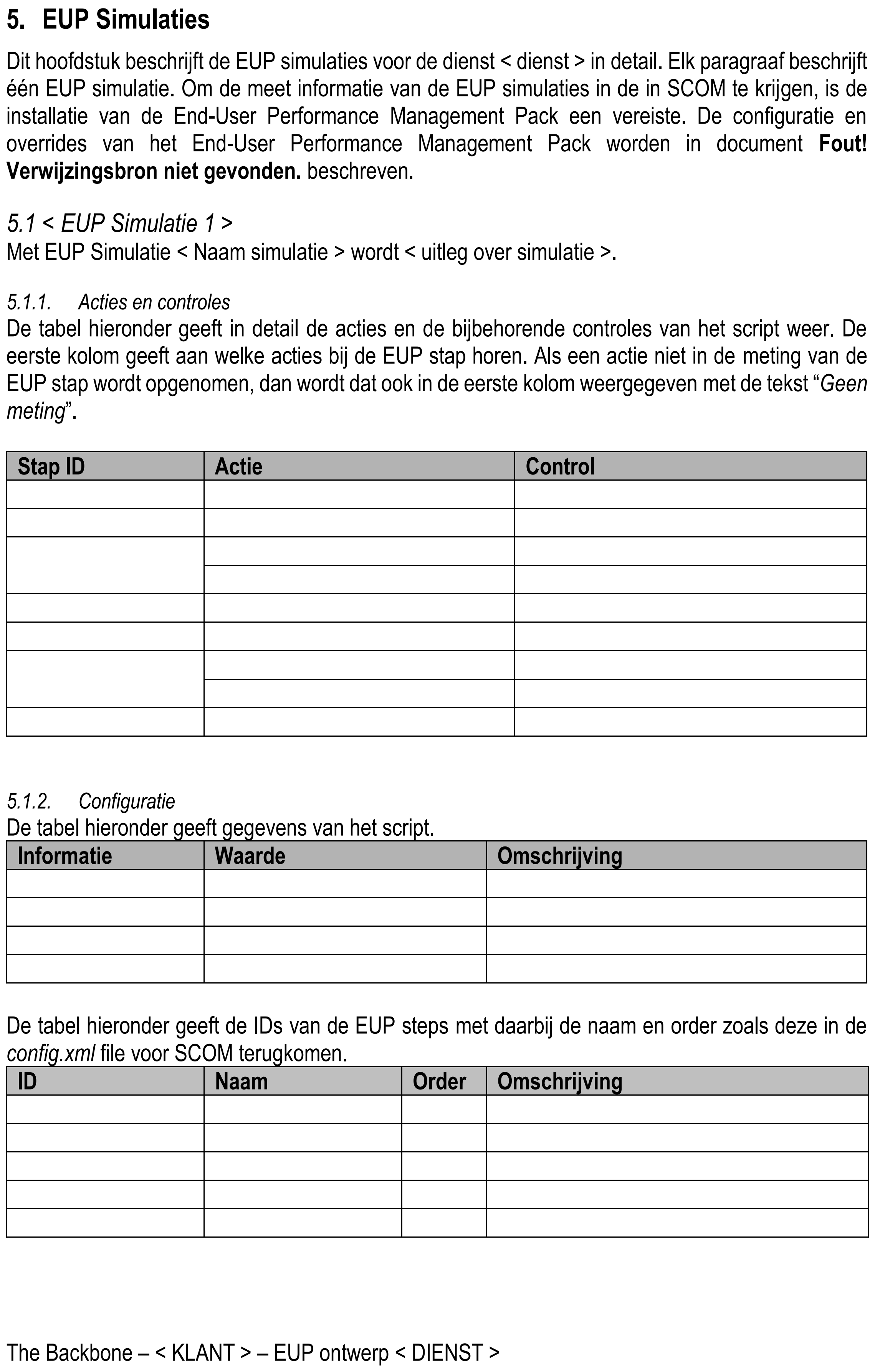
4.4 Gebruik script parameters

< Beschrijf hoe de script parameters aan het script worden doorgegeven. Bijvoorbeeld door een INI file, XML file of command-line parameters. Dit is afhankelijk van de gekozen scripttaal en mogelijkheden. >

De parameters zelf zijn script afhankelijk en worden bij het ontwerp van het script in detail beschreven.

4.5 Gebruik van wachtwoorden

Het wachtwoord dat gebruikt wordt voor het inloggen in ViewPoint wordt encrypted als parameter doorgegeven. Hiervoor wordt het Marshal.SecureStringToBSTR method vanuit de MSDN library gebruikt **Fout! Verwijzingsbron niet gevonden.**. Het encrypted (secured) is alleen op dezelfde machine met dezelfde gebruikt terug te zetten naar een unsecured string. Vanuit AutoIt wordt hiervoor PowerShell met de juiste commando's gestart. Voor het maken van een secured password, wordt een AutoIt script bijgeleverd.



5. EUP Simulaties

Dit hoofdstuk beschrijft de EUP simulaties voor de dienst < dienst > in detail. Elk paragraaf beschrijft één EUP simulatie. Om de meet informatie van de EUP simulaties in de in SCOM te krijgen, is de installatie van de End-User Performance Management Pack een vereiste. De configuratie en overrides van het End-User Performance Management Pack worden in document **Fout! Verwijzingsbron niet gevonden.** beschreven.

5.1 < EUP Simulatie 1 >

Met EUP Simulatie < Naam simulatie > wordt < uitleg over simulatie >.

5.1.1. Acties en controles

De tabel hieronder geeft in detail de acties en de bijbehorende controles van het script weer. De eerste kolom geeft aan welke acties bij de EUP stap horen. Als een actie niet in de meting van de EUP stap wordt opgenomen, dan wordt dat ook in de eerste kolom weergegeven met de tekst "*Geen meting*".

Stap ID	Actie	Control

5.1.2. Configuratie

De tabel hieronder geeft gegevens van het script.

Informatie	Waarde	Omschrijving

De tabel hieronder geeft de IDs van de EUP steps met daarbij de naam en order zoals deze in de *config.xml* file voor SCOM terugkomen.

ID	Naam	Order	Omschrijving

5.1.3. Script parameters

Het gedrag van het EUP simulatie script kan beïnvloed worden door parameters. Met deze parameters kunnen bijvoorbeeld schermnamen en zoek acties op een later moment eenvoudig aangepast worden. Een nadeel van het gebruik van parameters is dat het programmeren meer werk en complexiteit met zich meebrengt. Bij het ontwerp is zodoende een afweging gemaakt welke informatie vast in de code wordt geprogrammeerd en welke via een parameter wordt meegegeven. Een parameter kan op verschillende manieren worden meegegeven. Voor dit script worden parameter doormiddel van <command-line paramaters | XML configuratie file | INI configuratie file > aan het script gegeven. De tabel hieronder geeft de parameters en hun waarden die aan het script worden gegeven.

Parameter	**Standaard waarde**	**Verpl.**	**Omschrijving**

5.1.4. Thresholds

De tabel hieronder toont de "warning" en "critical" threshold van de EUP simulatie ID en de EUP stap IDs. Deze worden d.m.v. een override in SCOM gezet. De tekst "default waarde" wordt gebruikt als de default niet wordt aangepast[1].

ID	**Warning**	**Critical**

D EVALUATION INTERVIEW

The X in REP v.X depends on the version of the REP this interview is used for (REP v.0, REP v.1, or REP v.2).

Dutch:

1. Denk je nog ergens tegen aan te lopen als je dit proces zou gebruiken?
2. Vind je REP v.X bruikbaar, en waarom denk je dat?
3. Vind je REP v.X nuttig, en waarom denk je dat?
4. Denk je dat het huidige proces van The Backbone of REP v.X efficiënter is, en waarom denk je dat?
5. Denk je dat het huidige proces van The Backbone of REP v.X tot betere requirements leidt, en waarom denk je dat?
6. Denk je dat het huidige proces van The Backbone of REP v.X tot een betere implementatie van de EUP leidt, en waarom denk je dat?
7. Denk je dat het huidige proces van The Backbone of REP v.X tot een professioneler beeld bij klanten leidt, en waarom denk je dat?

English:

1. Do you think you will still miss something while using REP v.X?
2. Do you find REP v.X usable, and why do you think so?
3. Do you find REP v.X useful, and why do you think so?
4. Do you think the current process of The Backbone or REP v.X is more efficient, and why do you think so?
5. Do you think the current process of The Backbone or REP v.X leads to better requirements, and why do you think so?
6. Do you think that the current process of The Backbone or REP v.X leads to a better implementation of the EUP, and why do you think so?
7. Do you think the current process of The Backbone or REP v.X leads to a more professional image with customers, and why do you think so?

E

QUESTIONNAIRE REP V.1

Dutch:

1. Uw doel van de EUP staat beschreven in het contract. Wilt u hieronder uw behoeftes van de EUP in eigen bewoording opschrijven?
2. Welke applicaties en/of services zou u graag willen monitoren? (voorbeelden hiervan zijn: Outlook, Hix, Citrix, ERP, HRM, website: http://www.thebackbone.nl)
3. - Van welke veel gebruikte activiteiten door eindgebruikers wilt u de performance monitoren?

     en/of

   - Van welke activities en/of acties waar u performance problemen ervaart, wilt u de performance monitoren?

     en/of

   - Van welke van de activiteiten die beschreven staan in een Service Level Agreement (SLA) wilt u de performance monitoren?
4. Indien u de EUP gaat gebruiken voor het monitoren van SLA's, voegt u dan alstublieft de relevante SLA's toe als een bijlage of heeft u deze gereed voor de afspraak.
5. Hoe zijn de te monitoren applicaties gekoppeld met elkaar en met hun hardware en opslagruimte? (Hoe ziet de architectuur eruit, waar wordt de benodigde data vandaan gehaald, etc.?)

English:

1. Your goal of the EUP is mentioned in the contract. However, could you please tell us your needs for the EUP using your own words?
2. Which applications and/or services would you like to monitor? (Examples are: Outlook, Hix, Citrix, ERP, HRM, Website: http://www.thebackbone.nl)
3. - Of which of the commonly used activities by end-users do you want to monitor its performance?

     and/or

- Of which activities and/or actions where you experience performance issues, do you want to monitor their performance?

  and/or

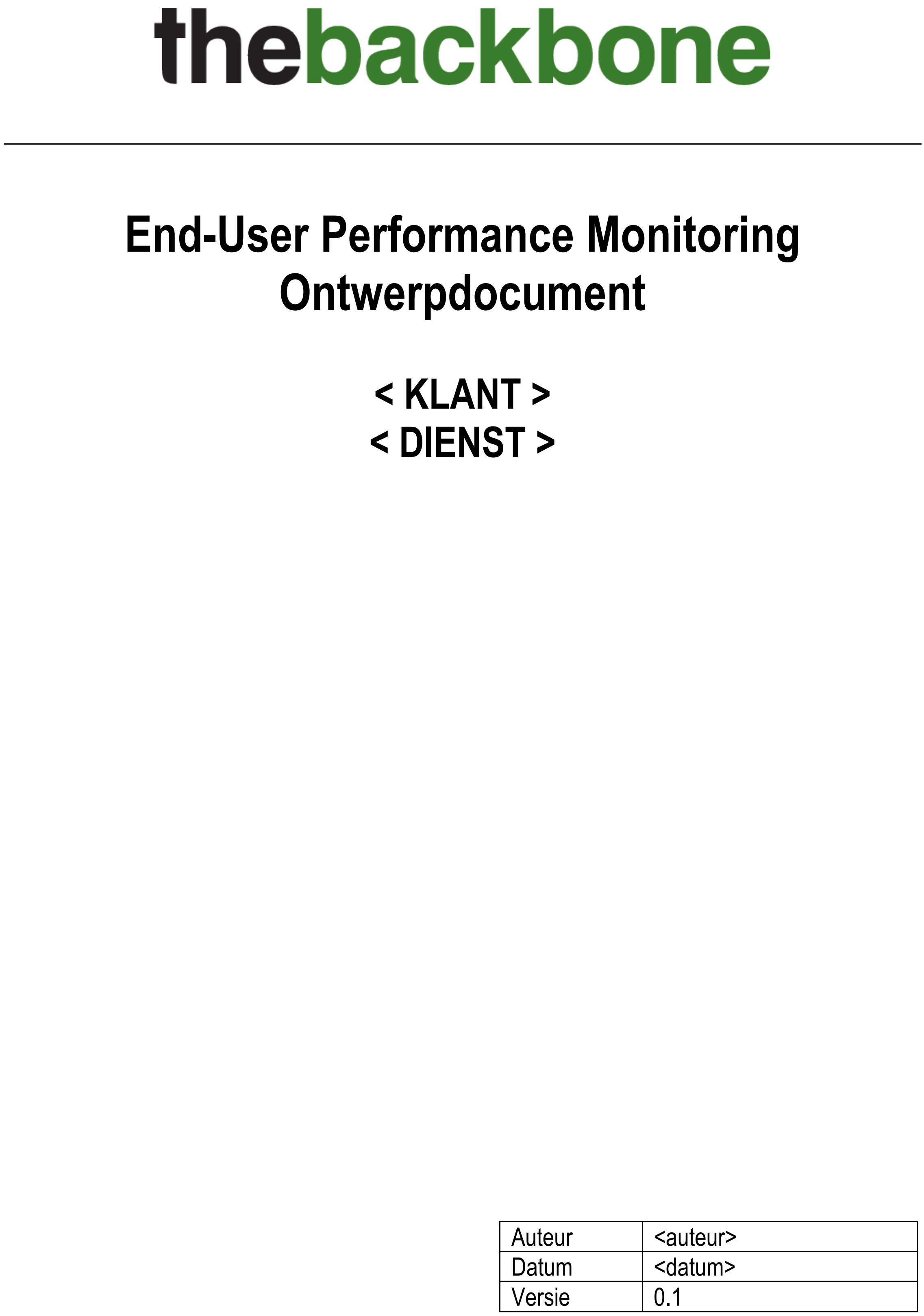
- Of which of the activities that are appointed in a Service Level Agreement (SLA) do you want to monitor the performance?

4. If you are going to use the EUP for monitoring SLAs, please sent the relevant SLAs as an attachment or have them ready before the meeting.

5. How are the applications that need to be monitored linked with each other and their hardware and storage? (What is the architecture like, where does the required data come from, etc.?)

F

DESIGN DOCUMENT REP V.2

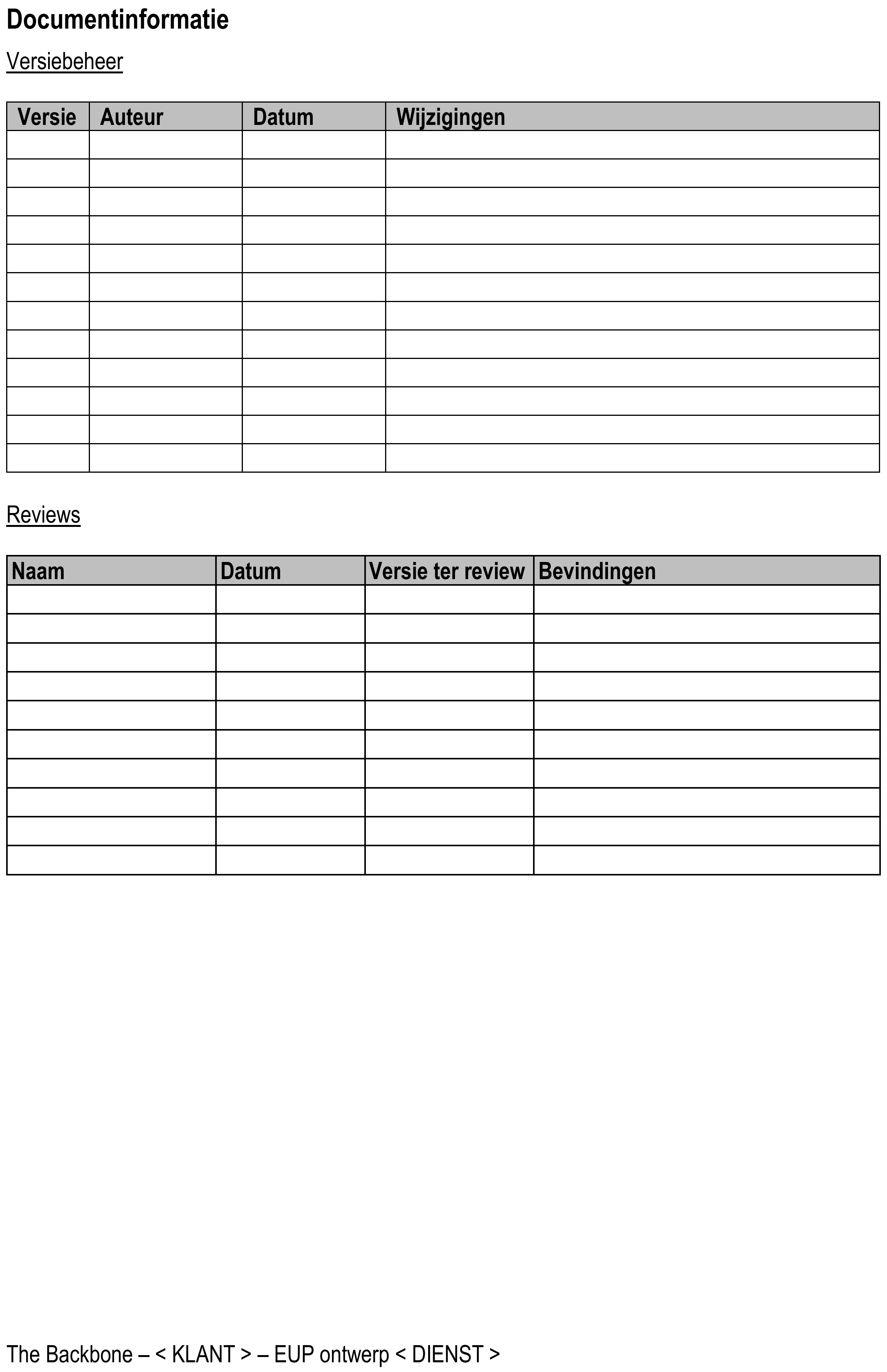


thebackbone

End-User Performance Monitoring Ontwerpdocument

< KLANT >
< DIENST >

Auteur	<auteur>
Datum	<datum>
Versie	0.1



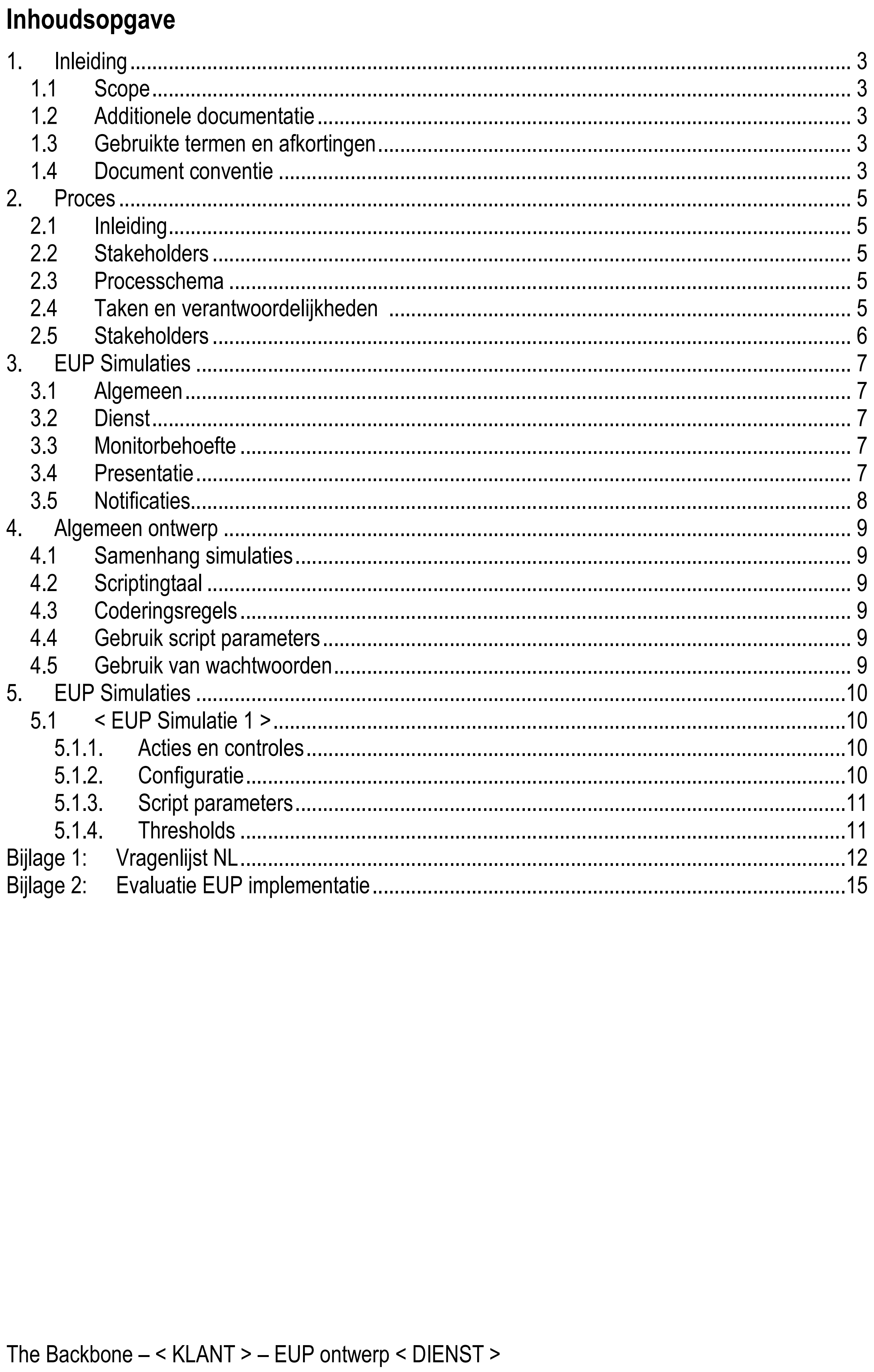
Documentinformatie

Versiebeheer

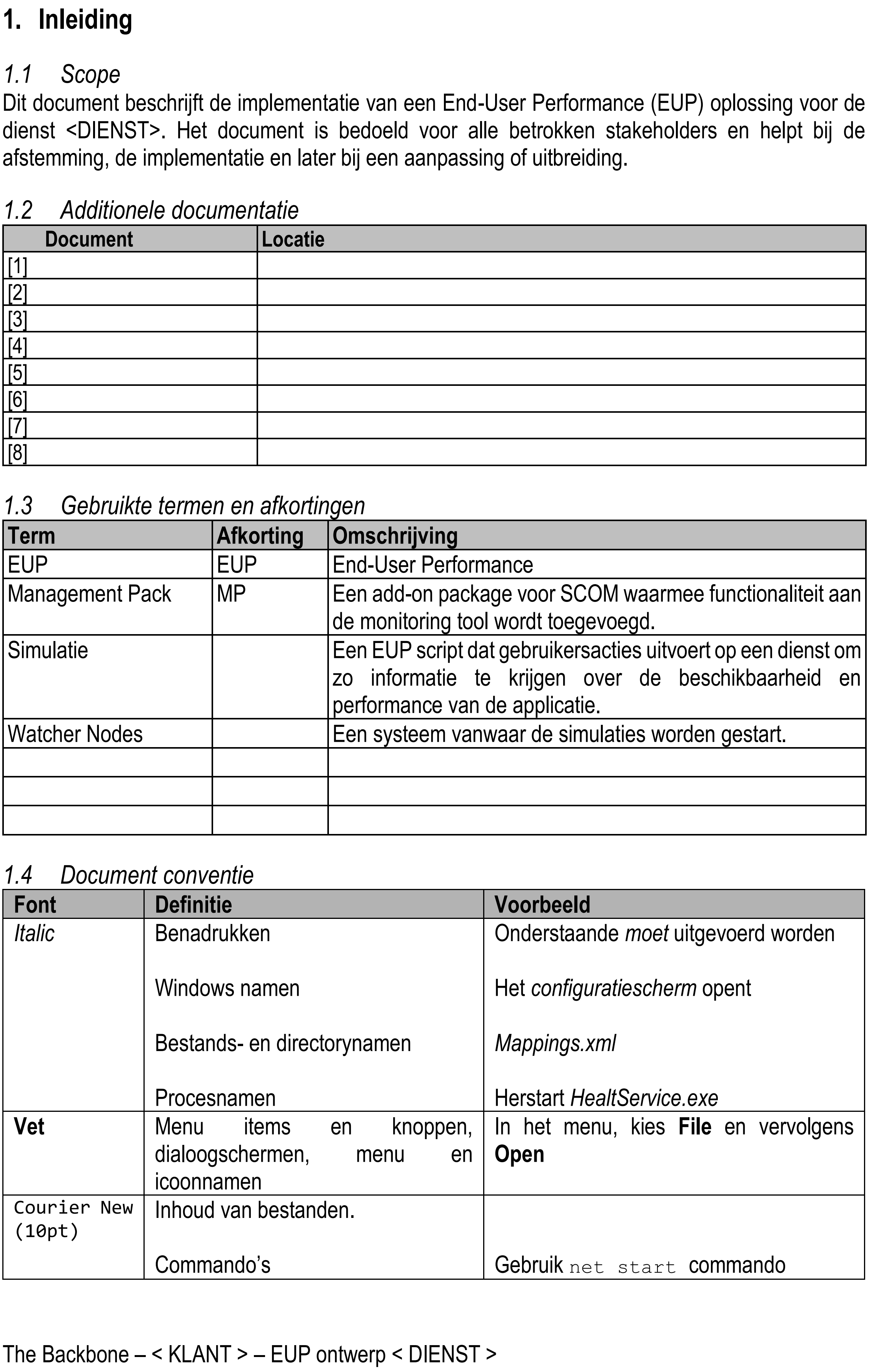
Versie	Auteur	Datum	Wijzigingen

Reviews

Naam	Datum	Versie ter review	Bevindingen



Inhoudsopgave



1. Inleiding

1.1 *Scope*

Dit document beschrijft de implementatie van een End-User Performance (EUP) oplossing voor de dienst <DIENST>. Het document is bedoeld voor alle betrokken stakeholders en helpt bij de afstemming, de implementatie en later bij een aanpassing of uitbreiding.

1.2 *Additionele documentatie*

Document	Locatie
[1]	
[2]	
[3]	
[4]	
[5]	
[6]	
[7]	
[8]	

1.3 *Gebruikte termen en afkortingen*

Term	Afkorting	Omschrijving
EUP	EUP	End-User Performance
Management Pack	MP	Een add-on package voor SCOM waarmee functionaliteit aan de monitoring tool wordt toegevoegd.
Simulatie		Een EUP script dat gebruikersacties uitvoert op een dienst om zo informatie te krijgen over de beschikbaarheid en performance van de applicatie.
Watcher Nodes		Een systeem vanwaar de simulaties worden gestart.

1.4 *Document conventie*

Font	Definitie	Voorbeeld
Italic	Benadrukken Windows namen Bestands- en directorynamen Procesnamen	Onderstaande *moet* uitgevoerd worden Het *configuratiescherm* opent *Mappings.xml* Herstart *HealtService.exe*
Vet	Menu items en knoppen, dialoogschermen, menu en icoonnamen	In het menu, kies **File** en vervolgens **Open**
`Courier New (10pt)`	Inhoud van bestanden. Commando's	Gebruik `net start` commando

2. Proces

2.1 *Inleiding*

Het ontwerpen en implementeren van een EUP oplossing wordt volgens een vast proces uitgevoerd. Dit borgt de kwaliteit van de EUP oplossing en de betrokkenheid van de stakeholders.

2.2 *Stakeholders*

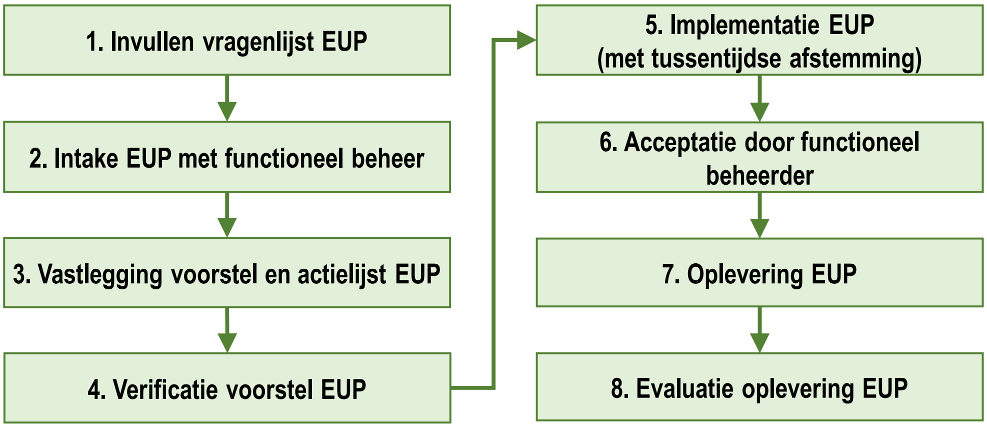
Onderstaande tabel geeft een overzicht van de stakeholders en de rol in het proces:

Wie	Rol
< Klant >	Eigenaar van de te monitoren dienst. Opdrachtgever voor het implementeren van een EUP oplossing. Kennis van de dienst dat nodig is voor het ontwerpen van de EUP Simulatie. Schakel tussen alle partijen.
The Backbone	Begeleid het proces om het DAM ontwerp tot stand te laten komen, verantwoordelijk voor het uitwerken van het DAM ontwerp en de daadwerkelijke bouw en inrichting van de DAM

2.3 *Processchema*

Het schema hieronder toont het proces van een EUP implementatie.

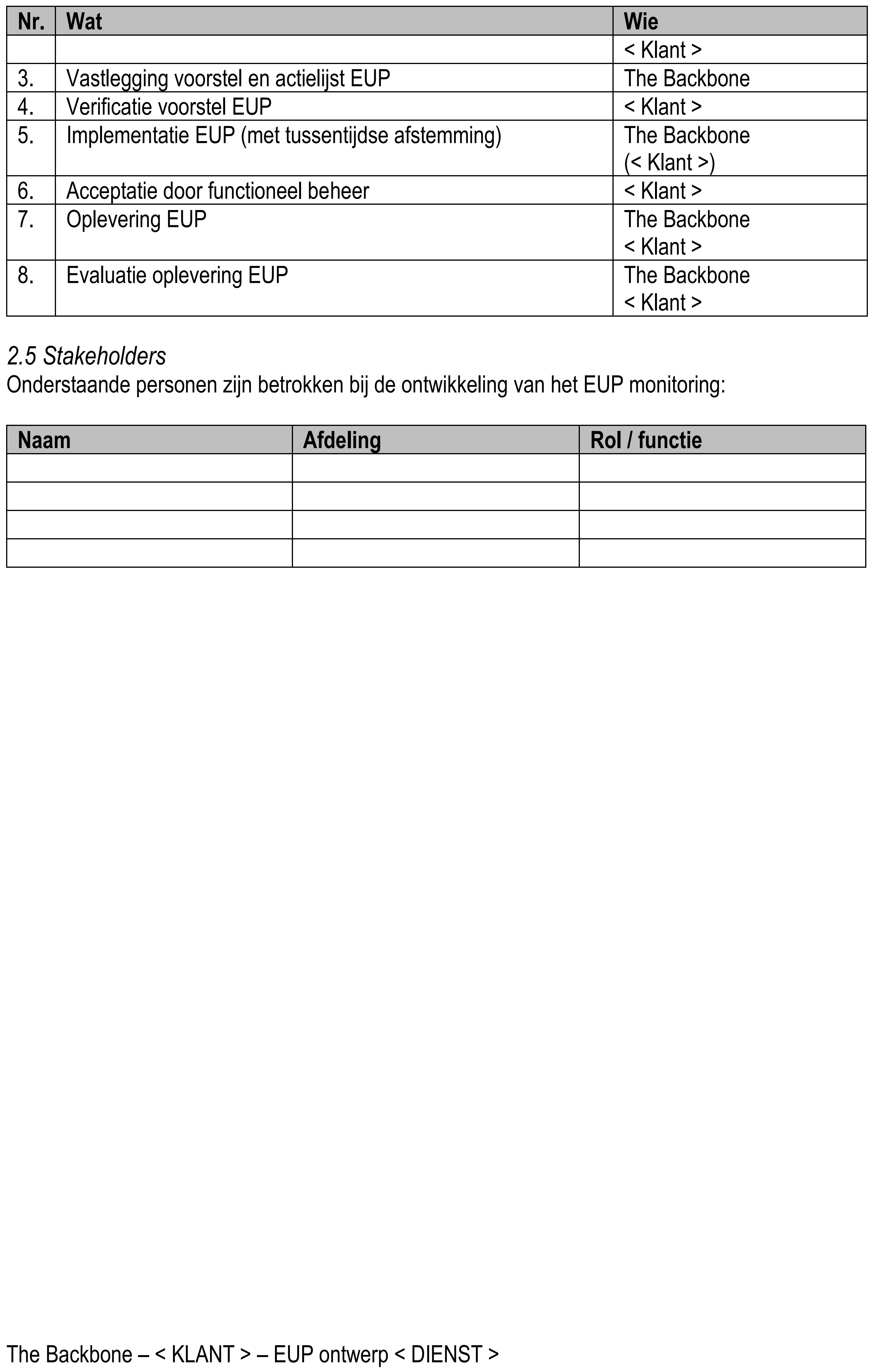
Het is belangrijk dat pas met de volgende stap wordt gestart als de vorige is afgerond. Dit om te voorkomen dat er onnodig werk wordt verricht.


1. Invullen vragenlijst EUP
2. Intake EUP met functioneel beheer
3. Vastlegging voorstel en actielijst EUP
4. Verificatie voorstel EUP
5. Implementatie EUP
(met tussentijdse afstemming)
6. Acceptatie door functioneel beheerder
7. Oplevering EUP
8. Evaluatie oplevering EUP

2.4 *Taken en verantwoordelijkheden*

Onderstaande tabel geeft de stappen weer binnen het proces met de betrokken / verantwoordelijke stakeholder(s).

Nr.	Wat	Wie
1.	Invullen vragenlijst EUP	< Klant >
2.	Intake EUP met functioneel beheer	The Backbone

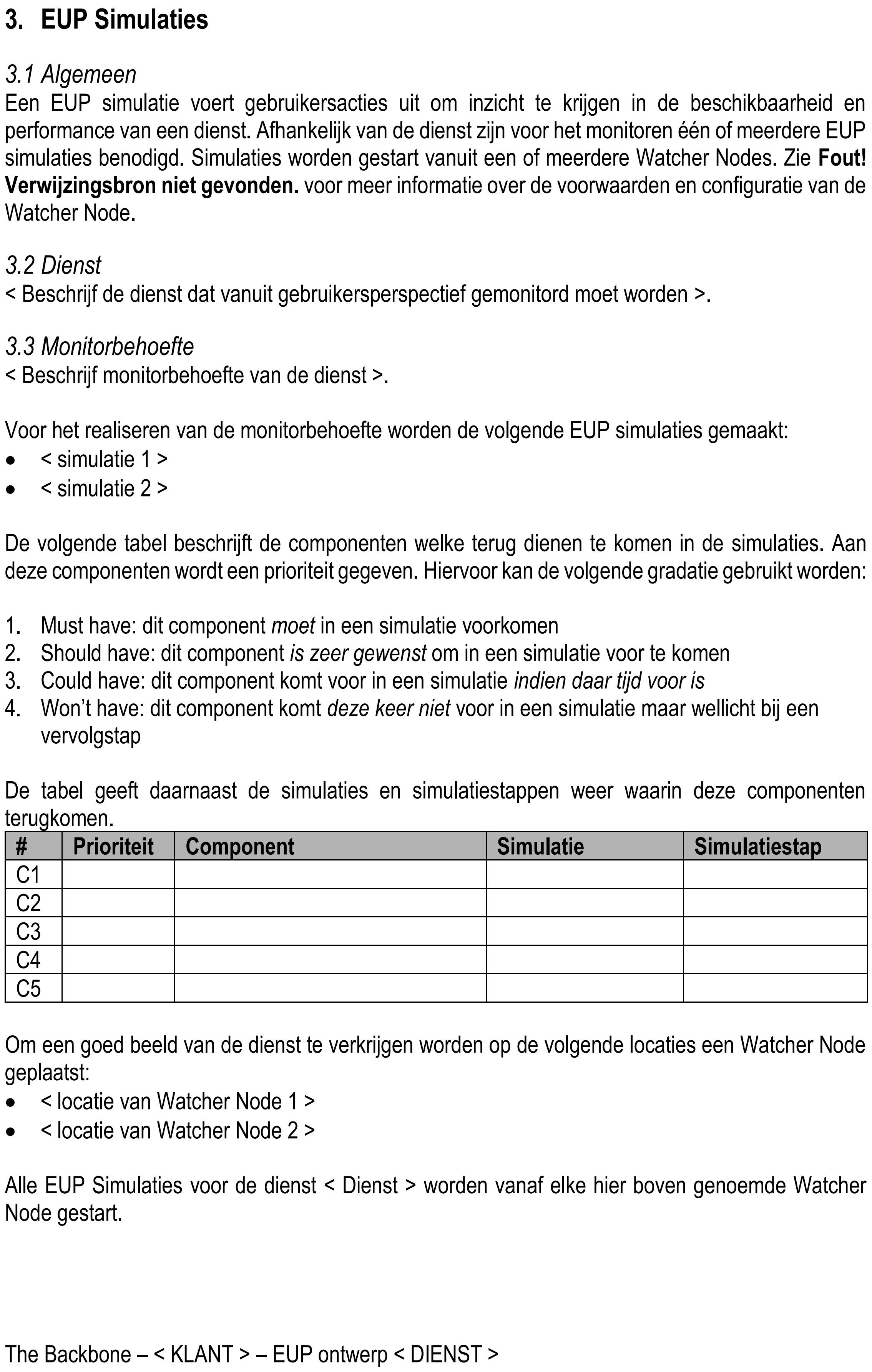


Nr.	Wat	Wie
		< Klant >
3.	Vastlegging voorstel en actielijst EUP	The Backbone
4.	Verificatie voorstel EUP	< Klant >
5.	Implementatie EUP (met tussentijdse afstemming)	The Backbone (< Klant >)
6.	Acceptatie door functioneel beheer	< Klant >
7.	Oplevering EUP	The Backbone < Klant >
8.	Evaluatie oplevering EUP	The Backbone < Klant >

2.5 Stakeholders

Onderstaande personen zijn betrokken bij de ontwikkeling van het EUP monitoring:

Naam	Afdeling	Rol / functie



3. EUP Simulaties

3.1 Algemeen

Een EUP simulatie voert gebruikersacties uit om inzicht te krijgen in de beschikbaarheid en performance van een dienst. Afhankelijk van de dienst zijn voor het monitoren één of meerdere EUP simulaties benodigd. Simulaties worden gestart vanuit een of meerdere Watcher Nodes. Zie **Fout! Verwijzingsbron niet gevonden.** voor meer informatie over de voorwaarden en configuratie van de Watcher Node.

3.2 Dienst

< Beschrijf de dienst dat vanuit gebruikersperspectief gemonitord moet worden >.

3.3 Monitorbehoefte

< Beschrijf monitorbehoefte van de dienst >.

Voor het realiseren van de monitorbehoefte worden de volgende EUP simulaties gemaakt:

- < simulatie 1 >
- < simulatie 2 >

De volgende tabel beschrijft de componenten welke terug dienen te komen in de simulaties. Aan deze componenten wordt een prioriteit gegeven. Hiervoor kan de volgende gradatie gebruikt worden:

1. Must have: dit component *moet* in een simulatie voorkomen
2. Should have: dit component *is zeer gewenst* om in een simulatie voor te komen
3. Could have: dit component komt voor in een simulatie *indien daar tijd voor is*
4. Won't have: dit component komt *deze keer niet* voor in een simulatie maar wellicht bij een vervolgstap

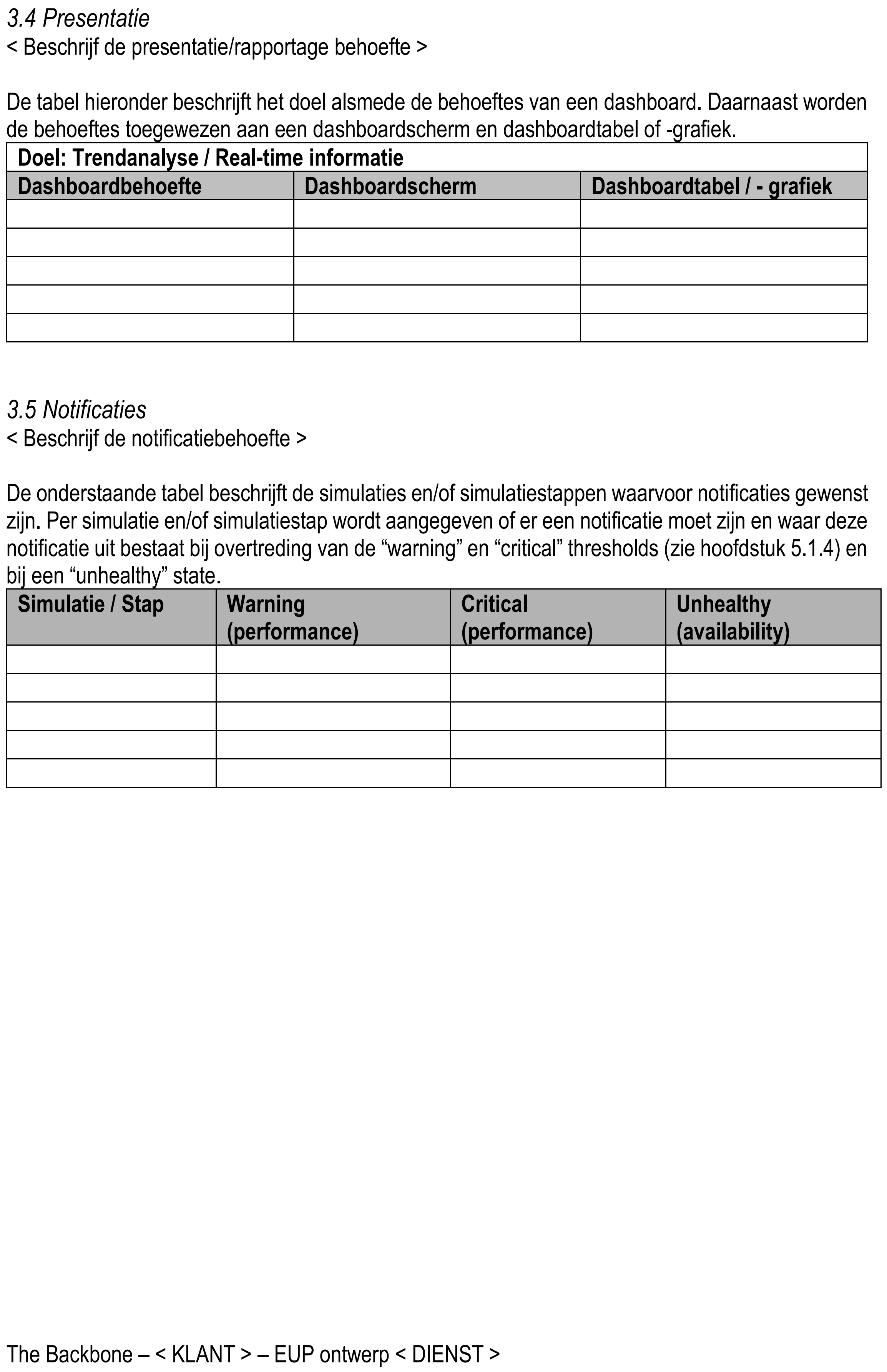
De tabel geeft daarnaast de simulaties en simulatiestappen weer waarin deze componenten terugkomen.

#	Prioriteit	Component	Simulatie	Simulatiestap
C1				
C2				
C3				
C4				
C5				

Om een goed beeld van de dienst te verkrijgen worden op de volgende locaties een Watcher Node geplaatst:

- < locatie van Watcher Node 1 >
- < locatie van Watcher Node 2 >

Alle EUP Simulaties voor de dienst < Dienst > worden vanaf elke hier boven genoemde Watcher Node gestart.



3.4 Presentatie

< Beschrijf de presentatie/rapportage behoefte >

De tabel hieronder beschrijft het doel alsmede de behoeftes van een dashboard. Daarnaast worden de behoeftes toegewezen aan een dashboardscherm en dashboardtabel of -grafiek.

Doel: Trendanalyse / Real-time informatie		
Dashboardbehoefte	**Dashboardscherm**	**Dashboardtabel / - grafiek**

3.5 Notificaties

< Beschrijf de notificatiebehoefte >

De onderstaande tabel beschrijft de simulaties en/of simulatiestappen waarvoor notificaties gewenst zijn. Per simulatie en/of simulatiestap wordt aangegeven of er een notificatie moet zijn en waar deze notificatie uit bestaat bij overtreding van de “warning” en “critical” thresholds (zie hoofdstuk 5.1.4) en bij een “unhealthy” state.

Simulatie / Stap	**Warning (performance)**	**Critical (performance)**	**Unhealthy (availability)**

4. Algemeen ontwerp

Dit hoofdstuk beschrijft het algemeen ontwerp van de EUP simulaties.

4.1 Samenhang simulaties

< Schrijf of er een samenhang tussen de simulaties is. Zo ja, waarom deze er is en hoe de scripts aan elkaar gerelateerd zijn. Bijvoorbeeld dat je vanuit een script één meerder scripts start. Eventueel ook de globale werking van alle scripts. >

4.2 Scriptingtaal

Na onderzoek van de dienst is voor het schrijven van de scripts gekozen voor de scripttaal < scripttaal >, omdat < uitleg >.

4.3 Coderingsregels

Bij het schrijven van de code wordt gebruikgemaakt van de algemene coding conventies van < scripttaal >. Deze code conventie is uitgelegd in **Fout! Verwijzingsbron niet gevonden.**.

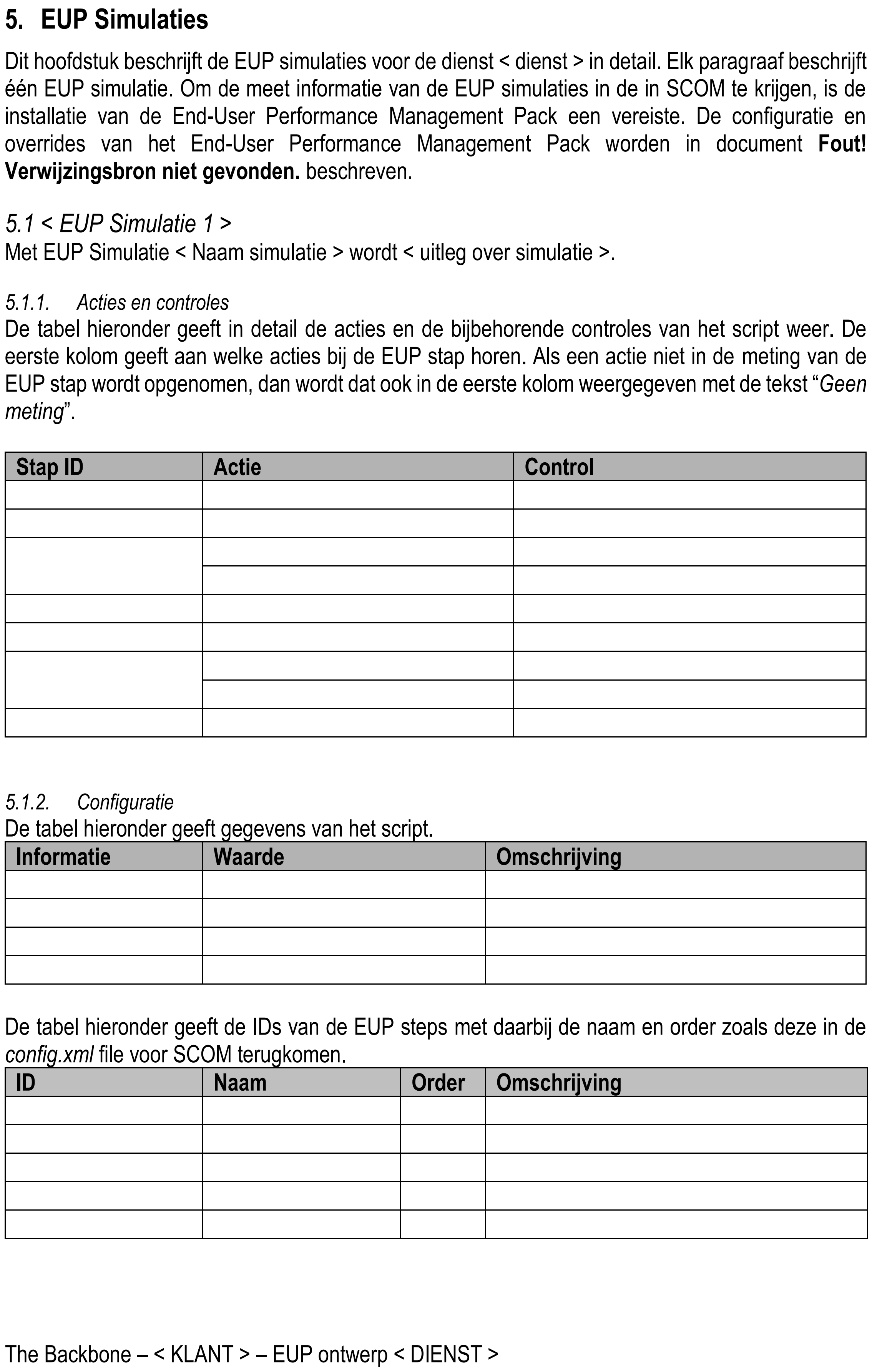
4.4 Gebruik script parameters

< Beschrijf hoe de script parameters aan het script worden doorgegeven. Bijvoorbeeld door een INI file, XML file of command-line parameters. Dit is afhankelijk van de gekozen scripttaal en mogelijkheden. >

De parameters zelf zijn script afhankelijk en worden bij het ontwerp van het script in detail beschreven.

4.5 Gebruik van wachtwoorden

Het wachtwoord dat gebruikt wordt voor het inloggen in ViewPoint wordt encrypted als parameter doorgegeven. Hiervoor wordt het Marshal.SecureStringToBSTR method vanuit de MSDN library gebruikt **Fout! Verwijzingsbron niet gevonden.**. Het encrypted (secured) is alleen op dezelfde machine met dezelfde gebruikt terug te zetten naar een unsecured string. Vanuit AutoIt wordt hiervoor PowerShell met de juiste commando's gestart. Voor het maken van een secured password, wordt een AutoIt script bijgeleverd.



5. EUP Simulaties

Dit hoofdstuk beschrijft de EUP simulaties voor de dienst < dienst > in detail. Elk paragraaf beschrijft één EUP simulatie. Om de meet informatie van de EUP simulaties in de in SCOM te krijgen, is de installatie van de End-User Performance Management Pack een vereiste. De configuratie en overrides van het End-User Performance Management Pack worden in document **Fout! Verwijzingsbron niet gevonden.** beschreven.

5.1 < EUP Simulatie 1 >

Met EUP Simulatie < Naam simulatie > wordt < uitleg over simulatie >.

5.1.1. Acties en controles

De tabel hieronder geeft in detail de acties en de bijbehorende controles van het script weer. De eerste kolom geeft aan welke acties bij de EUP stap horen. Als een actie niet in de meting van de EUP stap wordt opgenomen, dan wordt dat ook in de eerste kolom weergegeven met de tekst "*Geen meting*".

Stap ID	Actie	Control

5.1.2. Configuratie

De tabel hieronder geeft gegevens van het script.

Informatie	Waarde	Omschrijving

De tabel hieronder geeft de IDs van de EUP steps met daarbij de naam en order zoals deze in de *config.xml* file voor SCOM terugkomen.

ID	Naam	Order	Omschrijving

5.1.3. Script parameters

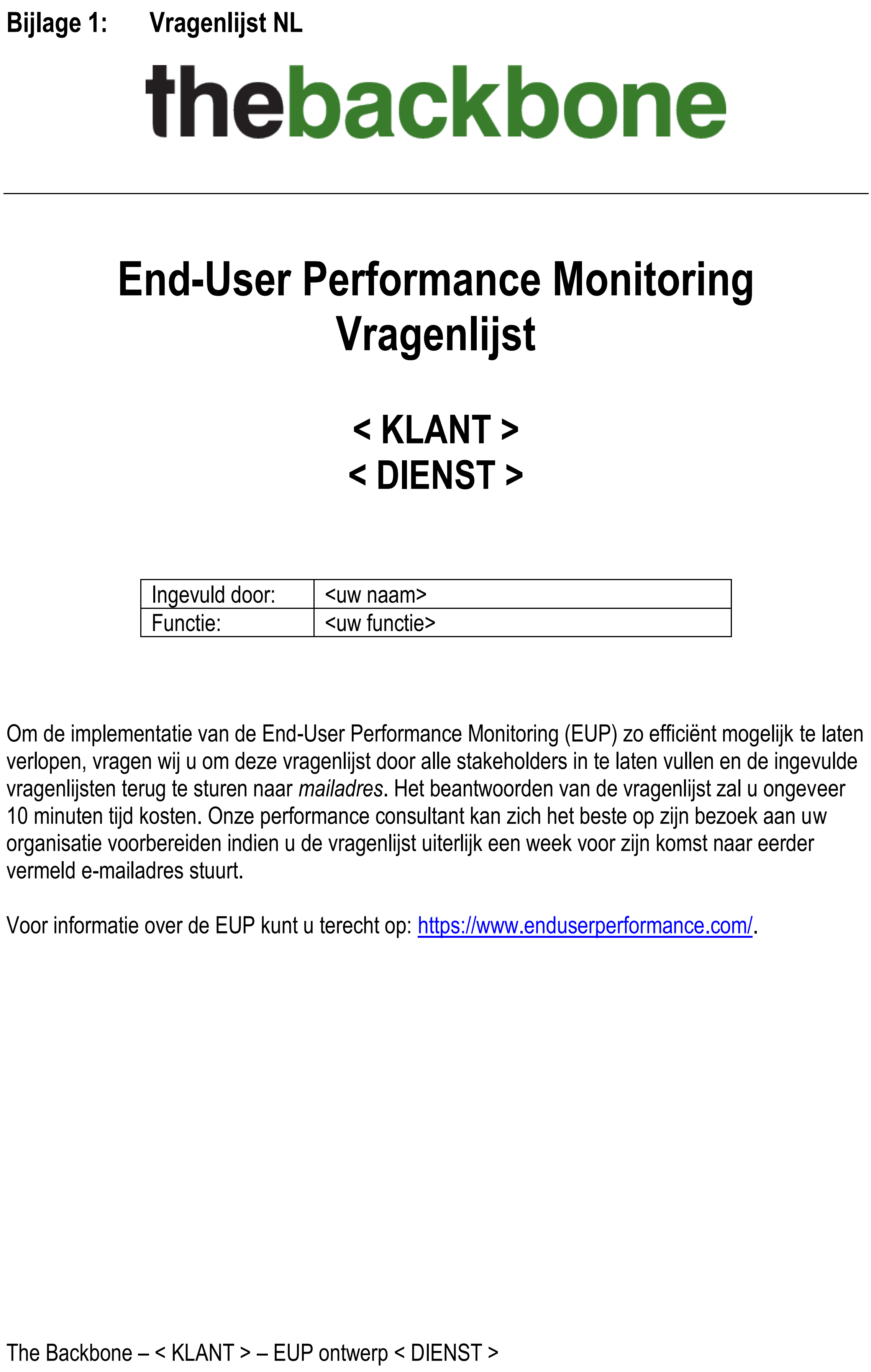
Het gedrag van het EUP simulatie script kan beïnvloed worden door parameters. Met deze parameters kunnen bijvoorbeeld schermnamen en zoek acties op een later moment eenvoudig aangepast worden. Een nadeel van het gebruik van parameters is dat het programmeren meer werk en complexiteit met zich meebrengt. Bij het ontwerp is zodoende een afweging gemaakt welke informatie vast in de code wordt geprogrammeerd en welke via een parameter wordt meegegeven. Een parameter kan op verschillende manieren worden meegegeven. Voor dit script worden parameter doormiddel van <command-line paramaters | XML configuratie file | INI configuratie file > aan het script gegeven. De tabel hieronder geeft de parameters en hun waarden die aan het script worden gegeven.

Parameter	Standaard waarde	Verpl.	Omschrijving

5.1.4. Thresholds

De tabel hieronder toont de “warning” en “critical” threshold van de EUP simulatie ID en de EUP stap IDs. Deze worden d.m.v. een override in SCOM gezet. De tekst “default waarde” wordt gebruikt als de default niet wordt aangepast[1].

ID	Warning	Critical



Bijlage 1: Vragenlijst NL

thebackbone

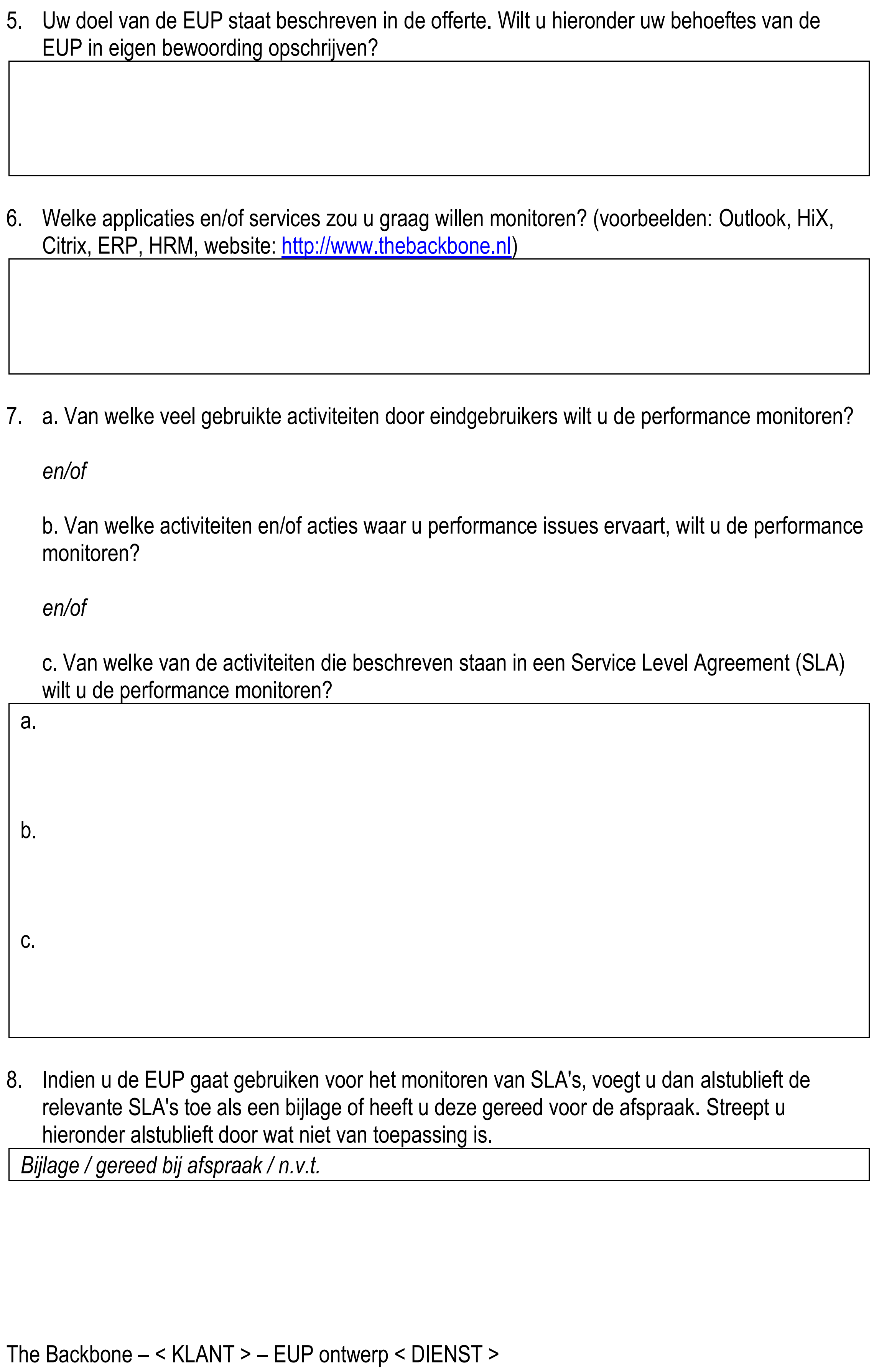
End-User Performance Monitoring Vragenlijst

< KLANT >
< DIENST >

Ingevuld door:	<uw naam>
Functie:	<uw functie>

Om de implementatie van de End-User Performance Monitoring (EUP) zo efficiënt mogelijk te laten verlopen, vragen wij u om deze vragenlijst door alle stakeholders in te laten vullen en de ingevulde vragenlijsten terug te sturen naar *mailadres*. Het beantwoorden van de vragenlijst zal u ongeveer 10 minuten tijd kosten. Onze performance consultant kan zich het beste op zijn bezoek aan uw organisatie voorbereiden indien u de vragenlijst uiterlijk een week voor zijn komst naar eerder vermeld e-mailadres stuurt.

Voor informatie over de EUP kunt u terecht op: https://www.enduserperformance.com/.



5. Uw doel van de EUP staat beschreven in de offerte. Wilt u hieronder uw behoeftes van de EUP in eigen bewoording opschrijven?

6. Welke applicaties en/of services zou u graag willen monitoren? (voorbeelden: Outlook, HiX, Citrix, ERP, HRM, website: http://www.thebackbone.nl)

7. a. Van welke veel gebruikte activiteiten door eindgebruikers wilt u de performance monitoren?

   en/of

   b. Van welke activiteiten en/of acties waar u performance issues ervaart, wilt u de performance monitoren?

   en/of

   c. Van welke van de activiteiten die beschreven staan in een Service Level Agreement (SLA) wilt u de performance monitoren?

a.

b.

c.

8. Indien u de EUP gaat gebruiken voor het monitoren van SLA's, voegt u dan alstublieft de relevante SLA's toe als een bijlage of heeft u deze gereed voor de afspraak. Streept u hieronder alstublieft door wat niet van toepassing is.

Bijlage / gereed bij afspraak / n.v.t.

9. Hoe zijn de te monitoren applicaties gekoppeld met elkaar en met hun hardware en opslagruimte? (Hoe ziet de architectuur eruit, waar wordt de benodigde data vandaan gehaald, etc.?)

Bijlage 2: Evaluatie EUP implementatie

Datum:
Aanwezigen:

Evaluatie:

G QUESTIONNAIRE REP V.2

Dutch:

1. Uw doel van de EUP staat beschreven in het contract. Wilt u hieronder uw behoeftes van de EUP in eigen bewoording opschrijven?

2. Welke applicaties en/of services zou u graag willen monitoren? (voorbeelden hiervan zijn: Outlook, Hix, Citrix, ERP, HRM, website: http://www.thebackbone.nl)

3. - Van welke veel gebruikte activiteiten door eindgebruikers wilt u de performance monitoren?

     en/of

   - Van welke activities en/of acties waar u performance problemen ervaart, wilt u de performance monitoren?

     en/of

   - Van welke van de activiteiten die beschreven staan in een Service Level Agreement (SLA) wilt u de performance monitoren?

4. Indien u de EUP gaat gebruiken voor het monitoren van SLA's, voegt u dan alstublieft de relevante SLA's toe als een bijlage of heeft u deze gereed voor de afspraak.

5. Wat zijn de te monitoren applicaties en hoe zijn deze gekoppeld met elkaar en met hun hardware en opslagruimte? (Hoe ziet de architectuur eruit, waar wordt de benodigde data vandaan gehaald, wat zijn de koppelingen tussen de applicaties, etc.?)

English:

1. Your goal of the EUP is mentioned in the contract. However, could you please tell us your needs for the EUP using your own words?

2. Which applications and/or services would you like to monitor? (Examples are: Outlook, Hix, Citrix, ERP, HRM, Website: http://www.thebackbone.nl)

3. - Of which of the commonly used activities by end-users do you want to monitor its performance?

     and/or

- Of which activities and/or actions where you experience performance issues, do you want to monitor their performance?

  and/or

- Of which of the activities that are appointed in a Service Level Agreement (SLA) do you want to monitor the performance?

4. If you are going to use the EUP for monitoring SLAs, please sent the relevant SLAs as an attachment or have them ready before the meeting.

5. What are the applications that need to be monitored and how are they linked with each other and their hardware and storage? (What is the architecture like, where does the required data come from, what are the links between the applications, etc.?)