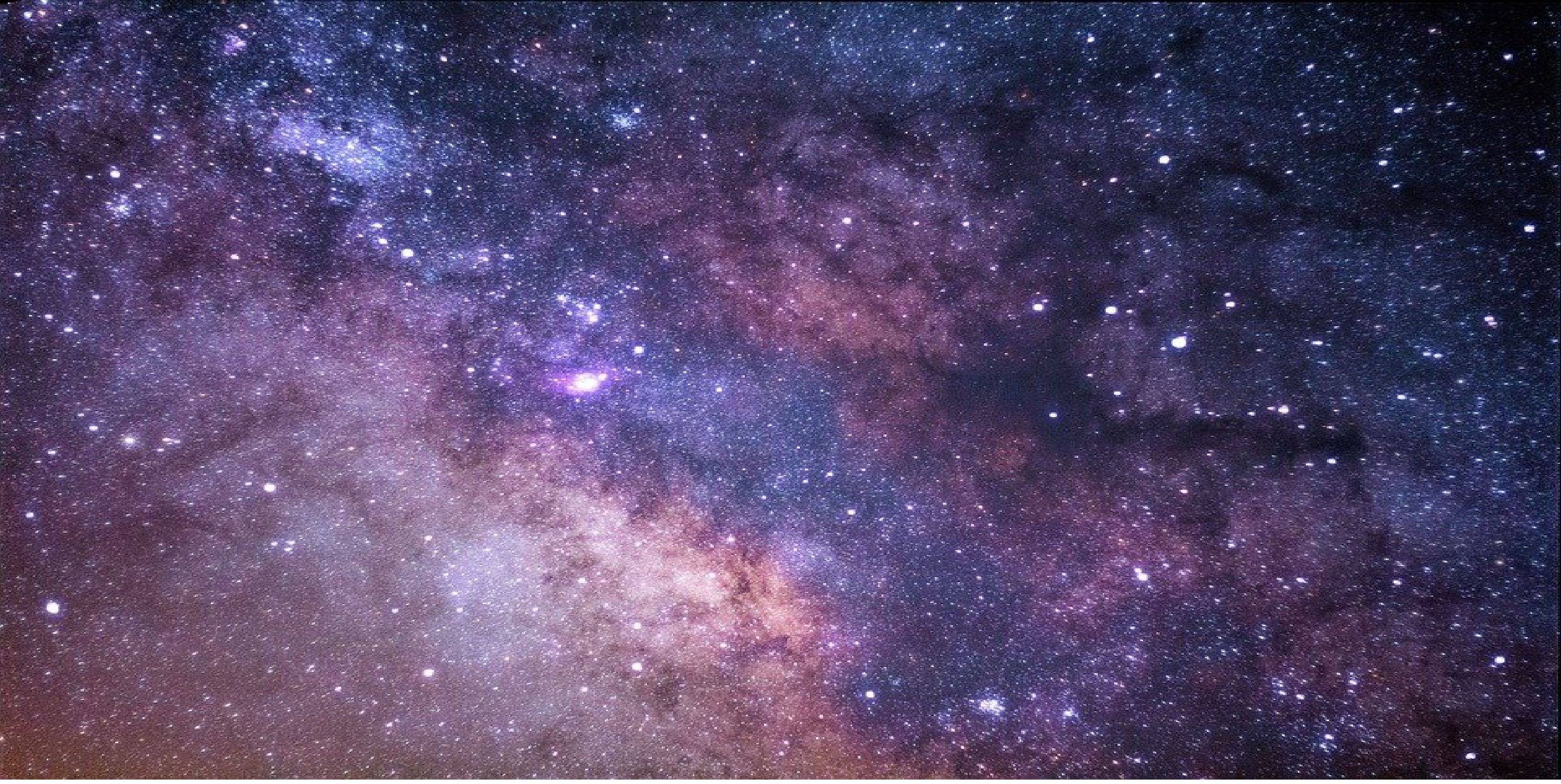


Evaluating the practical utility of ESSENCE as a reflective process

Rikke Holm Jessen

Software, sd103f21, 2021-06

Master's Project

Evaluating the practical utility of ESSENCE as a reflective process

Rikke Holm Jessen

Software, 2021-06

Master's Project

STUDENT REPORT

AALBORG UNIVERSITY

AALBORG UNIVERSITY
STUDENT REPORT

Computer Science
Aalborg University
http://www.aau.dk

Title:
Evaluating the practical utility of ESSENCE as a reflective process

Theme:
Systems Development

Project Period:
Spring Semester 2021

Project Group:
sd103f21

Participant(s):
Rikke Holm Jessen

Supervisor(s):
Ivan Aaen

Copies: 1

Page Numbers: 119

Date of Completion:
June 7, 2021

Abstract:

The ESSENCE methodology is designed to improve the reflective process of development projects, enabling development teams to understand multiple aspects of the project and move in an appropriate direction. In this work, I document the experience of using ESSENCE in a development project, in order to understand the benefits and drawbacks of ESSENCE and to identify possible elements that could be modified in order to improve the utility of ESSENCE.

0.1 Sammendrag

I denne rapport undersøger jeg ESSENCE metodologien. Formålet med ESSENCE er at hjælpe udviklings teams med at reflektere over udviklingsprojekter. Jeg undersøger ESSENCE ved at arbejde på et udviklingsprojekt hvor jeg bruger ESSENCE og løbende rapportere min oplevelse af ESSENCE. Mine resultater bør tages med forbehold, da de er meget subjektive og udelukkende baseret på mine erfaringer. Det er derfor sandsynligt at andre udviklere vil få andre resultater hvis de gentager eksperimentet. Selvom det er meget subjektivt, bidrager dette eksperiment med rige data og reflektion af ESSENCE.

Rapporten dokumenterer mit arbejde mens jeg arbejder på et udviklingsprojekt hvor jeg udvikler et værktøj til at sænke læringskurven for ESSENCE. I løbet af udviklingsprojektet dokumenterer jeg mine oplevelser af ESSENCE før og efter hver ESSENCE aktivitet. Rapportens struktur skal ses som en rejse hvor min forståelse af ESSENCE forbedres drastisk.

Rapporten begynder med et overblik over ESSENCE i kapitel 2. Dette overblik reflekterer min forståelse af ESSENCE da jeg begynder projektet. I senere kapitler arbejder jeg med ESSENCE og min forståelse udvikler sig. For at forstå mine overvejelser og valg gennem projektet, er det derfor vigtigt at forstå mit udgangspunkt.

I kapitel 3 beskriver jeg min fremgangsmåde og begrundelsen for den. Dette inkluderer både en præsentation af design som en refleksiv proces, og en beskrivelse af eksperimentet i sig selv.

Eksperimentet i sig selv er dokumenteret i kapitel 4, 5 og 6. Kapitel 4 beskriver hvordan jeg udførte pre-projekt aktiviteterne, de indledende aktiviteter i et ESSENCE udviklings projekt. I dette kapitel er min forståelse af ESSENCE mest proceduremæssig, og jeg forsøger at følge ESSENCE retningslinjer strikt.

Kapitel 5 beskriver de første tre sprint af projektet. Disse er mest fokuseret på at udvikle produktet og om ESSENCE aktiviteterne hjalp mig med at opdage ændringer i min forståelse.

I kapitel 6 overvejer jeg hvordan ESSENCE kan repræsenteres for at hjælpe studerende med at forstå koncepterne. I denne fase modnes min forståelse af ESSENCE og mit fokus er ændret. Hvor jeg tidligere mest fokuserede på om ESSENCE hjalp mig med at blive opmærksom på ændringerne i min forståelse, fokuserer jeg her mere på selve reflekteringen. I dette sprint introducerer jeg også konceptet *update matrix* og *fill out matrix*. Jeg ser disse matricer som den mest brugbare ændring af ESSENCE jeg har identificeret i projektet. Jeg forventer at introduktion af disse ville forbedre brugbarheden af ESSENCE meget.

Kapitel 7 afslutter rapporten med en dybdegående diskussion af min oplevelse af ESSENCE og en konklusion om fordele og ulemper af ESSENCE og forslåede ændringer. Jeg oplister også forslag til fremtidig research som jeg forventer kunne forbedre ESSENCE.

Contents

0.1 Sammendrag iii

Preface **vii**

1 Introduction **1**

1.1 Background 1

1.2 Project purpose 2

1.3 Report overview 2

2 The ESSENCE methodology **4**

2.1 Preface 4

2.2 Introduction 4

2.3 ESSENCE Overview 5

2.4 Challenge 6

2.5 Views 6

2.6 Roles 7

2.7 Ecology 8

2.8 Leverage 11

2.9 Pre-project 14

2.10 Configuration 15

2.10.1 Example of configuration 18

2.11 Levels 18

2.12 RST Review 19

2.13 Prospect Scenarios 20

2.14 Pivot or persevere 21

3 Approach **26**

3.1 Designing as reflective conversation 26

3.1.1 Seeing-moving-seeing 26

3.1.2 Methodology design 27

3.2 Project design 27

4 Case initiation **29**
4.1 Project Initiation 29
4.2 Preface 29
4.2.1 Challenge 29
4.3 Pre-project 29
4.3.1 Pre-project Expectations 29
4.3.2 Take stock of the challenge 30
4.3.3 Defining Relevant Ecology Objects 31
4.3.4 Defining Strategic Leverage Points 32
4.3.5 Initial Problem 33
4.3.6 Pre-project Evaluation 34
4.4 Prospect Scenarios 35
4.4.1 Prospect Scenario Expectations 35
4.4.2 Scenarios 35
4.4.3 Prospect conclusion 38
4.4.4 Prospect Scenario Evaluation 38

5 Initial development **40**
5.1 Sprint 1 40
5.1.1 Sprint 1 scope 40
5.1.2 Leverage choices 41
5.1.3 Sprint 1 feature expansion 46
5.1.4 Product Build 48
5.1.5 Product evaluation 48
5.1.6 Reflections about ESSENCE 48
5.1.7 RST Review expectations 50
5.1.8 RST Review 50
5.1.9 RST Review Evaluation 51
5.2 Sprint 2 54
5.2.1 Feature expansion 54
5.2.2 Scope 54
5.2.3 Product Build 54
5.2.4 RST Review expectations 54
5.2.5 RST Review 57
5.2.6 RST Review Evaluation 57
5.3 Sprint 3 59
5.3.1 Scope 59
5.3.2 Product Build 59
5.3.3 RST Review expectations 60
5.3.4 RST Review 60
5.3.5 RST Review Evaluation 62

6 Late development **65**

6.1 Sprint 4 . . . 65

6.1.1 Sprint scope . . . 65

6.1.2 Design of part 1 . . . 65

6.1.3 Design research . . . 67

6.1.4 Design choices . . . 69

6.1.5 Thoughts about ESSENCE . . . 80

6.1.6 Sprint 4 conclusion . . . 86

6.1.7 RST Review expectations . . . 86

6.1.8 RST Review . . . 86

6.1.9 RST Review Evaluation . . . 87

6.2 Sprint 5 . . . 88

6.2.1 Scope . . . 88

6.2.2 Final product . . . 88

7 Findings **93**

7.1 Discussion . . . 93

7.1.1 How did my understanding of ESSENCE evolve during the project? . 93

7.1.2 Benefits, drawbacks, and modifications of ESSENCE concepts . . . 94

7.1.3 Did I succeed in reducing the learning curve of ESSENCE? . . . 95

7.2 Conclusions . . . 96

7.3 Future work . . . 97

7.4 Acknowledgements . . . 98

Bibliography **99**

A Appendix **100**

A.1 Appendix . . . 100

A.2 SWOT analysis for ecology objects . . . 100

A.3 Leverage PCRT analysis . . . 107

A.4 Extracted features for sprint 1 . . . 109

A.5 Sprint 1 evaluation comments . . . 110

A.6 Sprint 2 RST Review . . . 111

A.7 Sprint 3 feature extraction . . . 112

A.8 Sprint 3 RST Review . . . 113

A.9 Updated configuration table . . . 115

A.10 Configuration table update Sprint 4 . . . 116

A.11 Further work on the product . . . 118

Preface

This report is written by a Software student at the Computer Science department at Aalborg University, Denmark. It is assumed that readers have a basic understanding of agile software development.

Aalborg University, June 7, 2021

Rikke Holm Jessen
<Rhja16@Student.aau.dk>

Chapter 1

Introduction

1.1 Background

There is no finite set of instructions applicable in all development projects. The appropriate approach depends on the nature of the development project as well as the state of the project. The philosopher Donald A. Schön describes the activity of designing as *seeing-moving-seeing* [6], where the designer reacts to the current design by seeing and interpreting the qualities of the current design in relation to the task, moving - updating the design based on the interpretation, and then seeing the updated design. In its nature, a development project is very much a design situation. The development team sees the current product in relation to the problem, interprets the quality of the solution, and updates the product according to their interpretation.

Agile approaches for development projects are becoming popular, in particular Scrum[8], which focuses on product development in small increments, and early delivery to customers. By getting feedback from customers early, the team can move based on the end-users interpretation and not just their own assumptions about how the user will interpret the product.

While Scrum contains structures for learning about the quality of the product and improving the process (by retrospectives), it does not contain any structures for reflection on the project as a whole. However, there is much more to a project than just the product, and paying attention to other domains, like whether or not the problem is worth solving, or understanding how the solution should bring value to the customer, can be important for success of a project. If the project is solving a problem no one cares about, even if it solves it well, the project will be likely to fail.

The ESSENCE methodology is an attempt to incorporate reflection on more aspects of development projects. However, it is unclear whether or not it actually works as intended, as ESSENCE is largely a theoretical methodology, and have never been tested in a real development project. Therefore, there is a need to examine the utility of ESSENCE.

1.2 Project purpose

Without proper examination, it is difficult to tell whether ESSENCE is useful for a development team in real development projects. Additionally, like any other type of design, a methodology can be designed based on a *seeing-moving-seeing* approach. Therefore, testing the qualities of ESSENCE is not only useful for understanding the current utility of ESSENCE, but also for improving the design. An experiment can therefore have the following two advantages:

1. An understanding of how using ESSENCE affects development projects.
2. Identifying possible modifications of ESSENCE that could improve the utility.

In this project, I carry out such an experiment. Specifically, I aim to answer the following questions:

- **Q1**: Which benefits and drawbacks can ESSENCE have in a development project?
- **Q2**: How could ESSENCE be modified in order to improve the utility?

1.3 Report overview

As a software developer myself, I have experience with development projects. In this project, I utilize this experience in order to answer the project questions, through the course of a development project. The results of this project are highly subjective as they are based entirely on my experience and interpretation. There should therefore be certain reservations with regards to my findings, as the experiment would likely provide other results if repeated by other developers, their conclusion would likely differ. Ideally, this study should be carried out by many different developers in order to gain a general insight. However, this is a costly study, so an experiment with many participants does not appear realistic at this point. Also, even though it is highly subjective, it still provides rich data and reflection about ESSENCE.

The report documents my work on a development project where I develop a tool for lowering the ESSENCE learning curve. During the development project, I perform the appropriate ESSENCE activities, and before and after each activity I document my thoughts about ESSENCE. The structure of the report should therefore be viewed as a journey from my initial understanding and thoughts about ESSENCE, to my current understanding of ESSENCE. Throughout the report, my understanding of ESSENCE changes dramatically.

The report begins with an overview of ESSENCE in chapter 2. This overview reflects my understanding of ESSENCE at the beginning of the project. In later chapters, I work with ESSENCE, and my understanding of ESSENCE continually develops. In order to understand my decisions and reflections throughout the project, it is therefore important for the reader to understand my starting point.

In Chapter 3 I describe my approach and the reasoning behind it. This both includes a presentation of the design as a reflective conversation and a description of how the experiment in itself is structured.

The experiment in itself is documented in Chapter 4, 5 and 6. Chapter 4 describes how I performed the pre-project activities that are the initial activities in an ESSENCE development project. In this chapter, my understanding of ESSENCE is mostly procedural, and I attempt to strictly follow the guidelines in ESSENCE.

Chapter 5 describes the first three sprints of the project. These are mainly focused on developing the product and whether or not the ESSENCE activities helped me become aware of changes in my understanding.

In Chapter 6 I consider how ESSENCE could be represented in order to help students understand it. During this phase, my understanding of ESSENCE is matured. My understanding of the usefulness of ESSENCE has shifted, from mostly related to *becoming aware that my understanding of the project has changed*. In this part if understand the main utility of ESSENCE to be reflection in itself. In this sprint, I also introduce the idea of an *update matrix* and a *fill out matrix*. In my opinion, these matrices are the most useful modification to ESSENCE I have identified. I expect that introducing these into the ESSENCE methodology would greatly improve ESSENCE.

Chapter 7 concludes the report with an in-depth discussion about my experience working on this project, a conclusion about the benefits, drawbacks, and suggested modifications, as well as a listing of future work that I find, would be useful for ESSENCE. This project is to my knowledge the first work to evaluate ESSENCE. My findings, therefore, provide valuable insights into how ESSENCE can be used and what could be done to improve it. However, as I am only one person using ESSENCE in a specific development project, I have barely scratched the surface of ESSENCE. Therefore, I have proposed many different directions for future work that I find would be important contributions.

Chapter 2

The ESSENCE methodology

2.1 Preface

In this chapter, I describe the ESSENCE methodology as I understood it initially in the project. Although all information in the chapter is based on a draft provided by the ESSENCE author, Ivan Aaen2, and the description was approved by Aaen as factually correct, the description is far from how I would describe ESSENCE at the end of the project. However, the general structure of the entire report is designed to reflect how my understanding of ESSENCE evolves. Therefore, updating this chapter to reflect my understanding at the end of the project, which would likely provide the reader with a higher quality overview of ESSENCE, would make many of my thoughts and choices later in the project confusing. Therefore I leave this chapter in its original form.

2.2 Introduction

When development projects are initiated, the understanding of the project is usually very limited. During the development project, we go through the stages of understanding the problem, identifying a proper solution, and developing the solution.

Getting from the beginning of the project to a good solution is usually not a linear process, and the understanding of both the problem and what a good solution constitutes generally evolves during the development of that solution. Also, the world is becoming increasingly hypercomplex, so is it becomes impossible to select the optimal configuration for the project at initiation[1, p. 6]. Sometimes, new information about the project can inspire the development team to change course. However, changing course can often mean that work already done becomes obsolete. In this case, the development team has to decide whether to change course and waste already finished work or to continue on the current path even though they have identified a better one.

Identifying an appropriate path as early as possible in the development project can therefore help limit the resources needed in the project and/or get to a better solution.

One tool for this is the ESSENCE methodology.

This section describes ESSENCE and explains relevant concepts. All information about ESSENCE is based on an ESSENCE draft I received directly from the author, Ivan Aaen[1].

2.3 ESSENCE Overview

ESSENCE provides a collection of activities and concepts that can be used to support a development project. It is important to remember that ESSENCE is not a method or framework, but a collection of concepts and activities that should help guide the development of a project rather than add an extra layer of complexity to an already complex project. The activities in an ESSENCE project are designed to help the development team understand:

1. What they are actually trying to solve.
2. What is possible to develop.
3. What the plan for the project is at any given time.
4. What the context of the project is

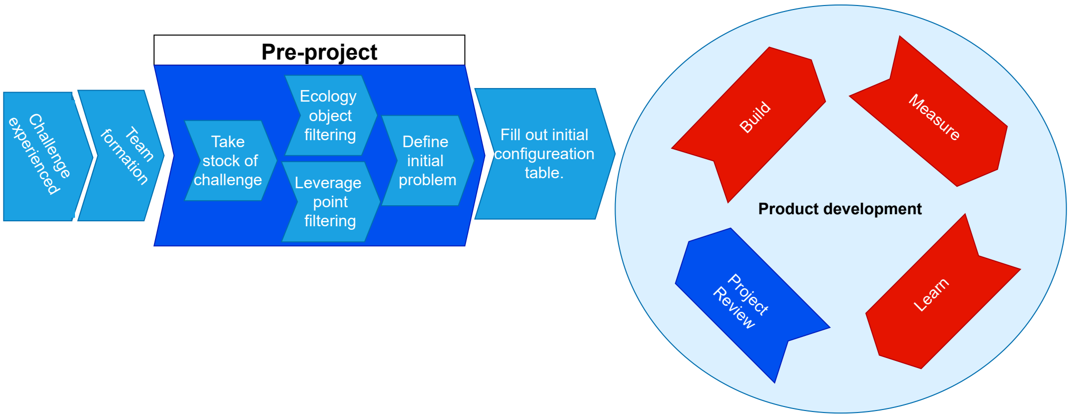


Figure 2.1: Overview of ESSENCE development project.

Figure 2.1 shows an overview of an ESSENCE project. A project does not necessarily have to follow this exact structure, but ESSENCE projects generally consist of two main phases, the pre-project, and product development.

Development projects, ESSENCE or not, usually begin with someone experiencing a **challenge**. The overall goal of the development project is then to create a solution to the challenge. At this point, even though there might be a vague idea about some solution to the challenge, the only element that is really defined is the challenge. As it is usually not possible to completely solve the challenge during the project an ESSENCE project begins by choosing a solvable sub-problem within the challenge that the project can use as a starting point. To select a proper problem, ESSENCE introduces the pre-project. In the pre-project, the team gets a quick overview of the possibilities, and based on this, they chose an initial problem for the project.

After the pre-project, the team should have a general idea about what they plan to develop. They can now fill out an initial configuration table for the project. The configuration table represents the current understanding of the project as a whole; what is being developed, how it should be developed, and why we should develop it. This leads us to the next step, product development. ESSENCE does not provide any specific practices or approaches for product development but uses configuration tables to guide the progress of the project, helping the team understand the project as well as communicate and question the current plan. For each iteration or major activity in the ESSENCE project, the configuration table should be re-evaluated and updated as the understanding of the project evolves.

2.4 Challenge

Every software development project should originate in the desire to solve an overall problem someone is facing. The project *challenge* is then to solve the whole or part of the problem. We illustrate this concept using an example of a common problem many people are facing, namely that it can be difficult to figure out whether you should go to a doctor for a given symptom. The overall *challenge* of a software project that aim to mitigate this problem would then be to *help people decide whether to arrange a doctor's appointment for a given symptom.*

2.5 Views

A central concept, in ESSENCE, is that of views. The view represents the angle of view that somebody takes when looking at the project. We observe different elements of the project depending on which view we take. Understanding views can both be helpful when people with different views are to communicate with each other and to change perspective and generate ideas.

ESSENCE defines four different views:

- Paradigm view
- Product view
- Project view
- Process view

The *paradigm* view is used to understand the problem and the context the solution is going to fit into; the problem domain. When we look at the project from a paradigm view, we ignore the solution and try to understand what the solution is trying to solve, and what the context should be for any solution.

The *project* view is related to the solution of the problem, that is, knowing the problem and context of the problem, what would be a good solution to the problem, what is the scope of that solution, and what should be a priority.

The *product* view is development-focused and looks at the problem from a technical point of view. This view is about understanding *how* the solution to the problem should be developed.

The *process* view is focused on the development process, and determining whether the project is moving in the right direction for solving the problem.

2.6 Roles

In every software project, there is a *developmentTeam* responsible for creating a solution. Each developmentTeam consists of a group of people, typically software developers. There are different roles in a development team, and each ESSENCE role is related to a specific view.

The *challenger* represents the customer and is related to the *project* view. The challenger is not a technical member of the development team, but is responsible for providing information on the problem domain and make decisions about the project scope and feature priority. In ESSENCE, we name these team members *challenger*, as they understand the project challenge.

The second ESSENCE role is the *responder*. The responder is equivalent to a developer, and it is the responsibility of the responders to apply their technical knowledge to develop a solution. The responder represents the project view, and most team members have the responder role. In ESSENCE, we name these team members *responder*, as they respond to the project challenge.

To keep the process going smoothly, one member of the development team is assigned the role of *anchor*. This is usually one of the responders. It is the responsibility of the anchor to keep the project running smoothly. The anchor represents the process view.

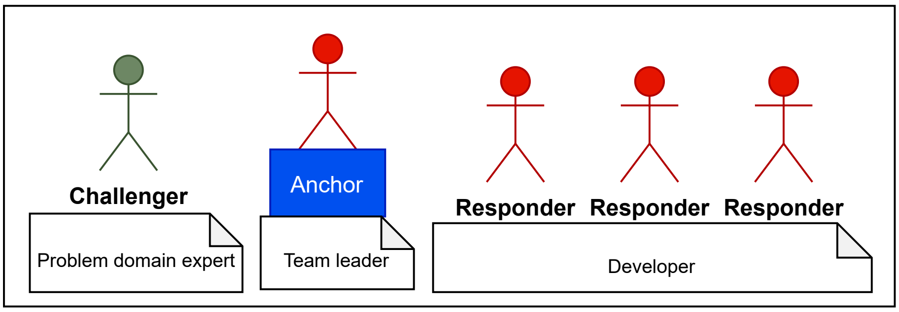


Figure 2.2: ESSENCE roles.

Figure 2.2 shows the three permanent roles in essence.

There is one additional role, the *child* role. This role is fleeting and can be assumed by any team member or any person outside the team. The child role is used to question the current understanding of the problem or desired solution, much like a real child questions many things that grown-ups have just accepted to be true. The goal of this role is to challenge assumptions and generate ideas that the team members would not otherwise think of. This role is related to the paradigm view.

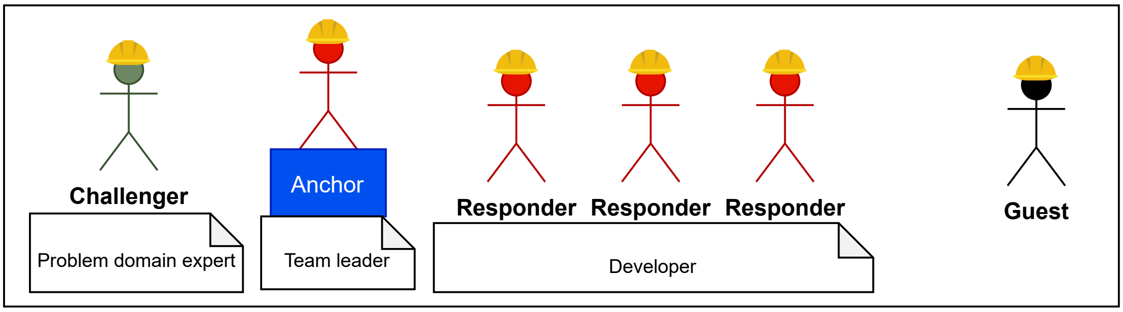


Figure 2.3: ESSENCE roles with child included.

Figure 2.3 illustrates how any team member or outside actor (guest) can assume the child role temporarily.

2.7 Ecology

Today, many systems interface with external resources to provide functionality to the user. This enables the developer to focus on a subset of the modules he would else have to develop to get a similar system.

The possible external resources that the system can interact with to provide functionality is called the *ecology* of the system. Each resource in the ecology is then called an *ecologyObject*.

There are different types of ecology objects. This include:

- Services
- Artifacts
- Repositories
- People

A service in the ecology is an external program that the system can interact with to solve part of the problem. This can e.g., be the Google Maps API, or any other system the system can communicate with without any modification.

Another type of ecology object is artifacts. These are devices outside the system that is already present in the problem domain and that the system can take advantage of. This can e.g., be smartphones or smartwatches.

The third type of ecology object is repositories. These contain already gathered information that the system can take advantage of. One example of this is a weather update app, that collects the weather information from an established source, rather than gathering the information itself.

The last type of ecology object, people, are people that the system could interface with to provide functionality. In a health system, this could be health professionals that the user can chat with and ask questions.

Ecology Object Filtering

To identify possible objects and choose the appropriate objects to interface within the solution, ESSENCE provides the *ecology object filtering* activity.

This consists of three steps:

1. Create a list of potential ecology objects in the problem domain.
2. Evaluate each object.
3. Select the most useful objects for the project.

As our understanding of a project is limited at all times, objects are always evaluated based on the current understanding of the project, not a universal truth.

The ecology filtering activity is done based on a known challenge. We illustrate this activity with a project based on the challenge described in 2.4;

> *Help people decide whether to arrange a doctor's appointment for a given symptom*

We call this project the *Health System*.

Step 1: For the first step, we identify possible ecology objects in the problem domain.

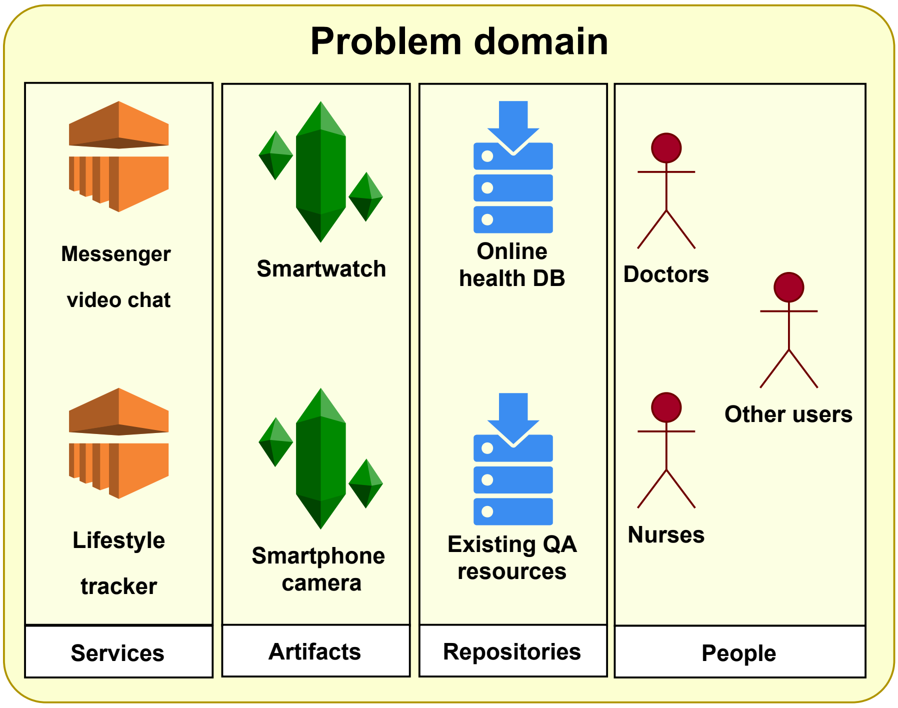


Figure 2.4: Some ecology objects for the health system.

Figure 2.4 shows some possible ecology objects in the problem domain of the health system. There are two suggestions for services, the Messenger Video Chat, which can be used to ask live questions from health professionals, and the Lifestyle Tracker, which can be used to gain additional insights into the user, possibly helping to gain a more nuanced answer. For the artifacts, we have a smartwatch, which can be used to interact with the app on the wrist, and a smartphone camera, which can be used to take pictures of e.g., rashes to gain a qualified answer. For repositories we have online health databases and existing QA resources, that could be imported into the solution to provide information. For people, we can provide contact information for doctors, nurses, and other users, that the users could contact to get answers to specific questions.

Step 2: For step 2, we evaluate each of the ecology objects. There are many different possibilities for evaluating the objects, and ESSENCE provides suggestions, although it is not required to use these. One approach is the SWOT evaluation, where the Strengths, Weaknesses, Opportunities, and Threats are listed for each object. We demonstrate this for the lifestyle tracker in figure 2.1, and the doctor in figure 2.2.

Step 3: To compare the different ecology objects to each other, and select the most appropriate, we assign a score to each object. This can be done using an adapted version of the SWOT analysis, where the strength and opportunities are assigned a value of between 0 and 5, while the weaknesses and threats are assigned a value of between 0 and -5.

Strengths(internal)	**Weaknesses(internal)**
- Possibility of more nuanced answers	- Difficult to implement - Not guaranteed to be accurate - Many people do not use lifestyle trackers.
Oppotunities(external)	**Threats(external)**
- Likely to get better with time - Possible to make health statistics	- Personal information can give problems with GDPR - Some people will not give information

Table 2.1: SWOT analysis of lifestyle tracker.

Strengths(internal)	**Weaknesses(internal)**
- Accurate answers - Simple to implement - Easy to get answers for user - Some users might prefer asking a human	- Doctors are not always available.
Opportunities(external)	**Threats(external)**
- Easy to adapt to specialized areas of medicine - Can get help from all around the world	- Quality of the help can vary, depending on the doctor

Table 2.2: SWOT analysis of Doctor.

Figure 2.3 shows values assigned to the ecology objects. We see that object that received the highest score is the *online health DB*, as even though *Doctors* has higher strength, the online health DB does not have as significant weaknesses and threats. This ranking can now be used to guide the initial ideas for the project.

2.8 Leverage

Similar to Ecology, the idea behind leverage is to identify resources that can be used in the product to reduce the tasks for the developer and improve the product, however, where ecology is about interfacing with external resources or objects, leverage is about including resources in the product.

Resources identified in the leverage category are named *leverage points*. Usual types of leverage points are:

- Technologies
- Artifacts

Ecology Object	Strengths	Weaknesses	Opportunities	Threats	Score
Messenger video chat	3	-2	3	-2	**2**
Lifestyle tracker	2	-3	3	-2	**0**
Smartwatch	2	-2	3	-1	**2**
Smartphone Camera	3	-1	3	-1	**4**
Online health DB	4	-1	4	-1	**6**
Existing QA resources	3	-2	2	-2	**1**
Doctors	5	-4	4	-4	**1**
Nurses	3	-2	2	-2	**1**
Other users	4	-2	3	-2	**3**

Table 2.3: Ecology objects ranking

- Repositories
- People

Technologies are existing scientific knowledge or machinery that the design can be based on. This can be anything like neural networks or activity analysis, which can be leveraged in the solution.

Artifacts are equipment and sensors that can be integrated into the solution.

Repositories are databases that are used for actively collecting, storing, and using information for the solution. This can be an error log or a user pattern database.

People can be specific users, authorities, networks, organizations, or others that have a particular role in building the solution

Leverage Point Filtering

The leverage point filtering activity consists of the same steps as the ecology object activity:

1. Identify resources that could be used as leverage points in the solution.
2. Evaluate each leverage point.
3. Select the most useful leverage points from the list.

Step 1: For the health system, we identify the possible leverage points shown in figure 2.5.

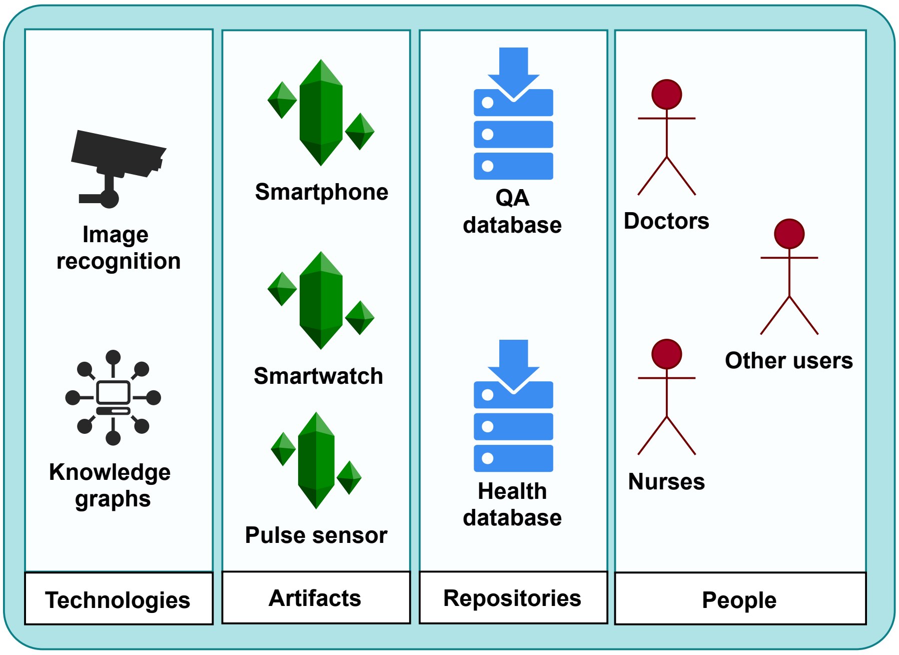


Figure 2.5: Some ecology objects for the health system.

For the *technologies*, we could use image recognition to try to identify similar rashes or knowledge graphs to search for similar conditions. For *artifacts*, we could use a smartphone for the UI, the smartwatch which can be used to track the user's vitals, to gain more insights, and a smartphone pulse sensor to gain additional information about the user. For *repositories* it would make sense to develop and maintain a QA database and a health database, containing all health-related information we can obtain. Instead of providing contact information or rely on collaboration with doctors and nurses, as would be the case if these are included in the ecology, we can hire some to be available for questions. This would give more reliable and available service, but would also be a great deal more expensive than relying on collaboration with health professionals. We could also develop a reward system for other users and integrate them into the system.

Step 2: For evaluating the leverage points, we can use the same approach as for ecology objects. We will not show an example for leverage points, as this is the same approach.

Step 3: Figure 2.4 shows an overview of the rankings for the identified leverage points. We see that

We see that some leverage points, like image recognition, have a very low score, indicating that, at the current time, the downsides for including these make the inclusion not worth it, compared to other features.

Leverage point	Strengths	Weaknesses	Opportunities	Threats	Score
Image recognition	3	-4	3	-4	**-2**
Knowledge graph	3	-3	4	-2	**2**
Smartphone	5	-2	5	-1	**7**
Smartwatch	3	-3	4	-2	**2**
Pulse sensor	3	-3	4	-3	**1**
QA database	4	-1	3	-2	**4**
Health database	5	-2	4	-2	**5**
Doctors	5	-4	5	-4	**2**
Nurses	4	-3	4	-4	**1**
Other users	3	-3	3	-2	**1**

Table 2.4: Leverage points ranking

2.9 Pre-project

The purpose of the pre-project activity is to get an overview of the possibilities for the project, to have a good starting point. The pre-project begins after the project challenge is defined, and results in the initial problem.

The pre-project consists of the following sub-activities:

1. Take stock of the Challenge - understand the problem and the broader domain in which the problem resides.

2. Do ecology object filtering on the challenge.

3. Do leverage point filtering on the challenge.

4. Define the initial problem based on the filtering results.

The initial problem is a sub-problem of the main problem. The main reason for doing the filtering activities before the initial problem is decided is, that initially, it can be difficult to define a scope for the project without an overall idea of what is possible. Therefore, we use the filtering activities to gain an overview of what is possible, and then we can define the scope of the project.

Using the Health System as an example, we began with the overall challenge:

> Help people decide whether to arrange a doctor's appointment for a given symptom.

In the ecology object filtering activity, we identified several ecology objects that we estimated to have good potential for external functionality:

- Online health DB - Collect health information from existing resources online.

- Smartphone camera - Use the smartphone camera to take pictures of rashes that worry uses
- Other users - Rely on other users to answer questions

For the leverage points, we identified the following with good potential:

- Smartphone - Use smartphones as the primary UI
- Health database - Maintain a database with learned health information
- QA database - Maintain own database with questions and answers between users

The identified ecology objects and leverage points are used to identify a sub-problem for the challenge, that we believe we can solve with our current understanding of the challenge. This is the initial problem. For the health system, the initial problem could be *Many people don't know if a symptom is something that should be examined by a doctor.*

2.10 Configuration

In a software project, it is important to always be clear on *what* we are developing, *why* the project and design make sense, and *how* we are planning to develop the design.

A central concept in the ESSENCE methodology is the configuration table. We use this table to paint a clear picture of the current state of the project, and answer all questions. Figure 2.6 shows a diagram, representing the configuration table. The diagram contains 13 categories, that together form an overview of the project. Each category has a colored border. The border describes which view the category belongs to, and therefore which members should be responsible for the category.

The yellow categories are related to the *paradigm* view, and therefore to understanding the problem domain. All team members should be active when filling out these categories, adapting the *child* role.

The green categories are related to the *project* view, defining the scope and focus of the project. Here, the entire team can participate in the discussion, but it is the responsibility of the *challenger* to fill out these categories.

The red categories are related to the *product* view, the categories are related to the technological development of the solution. To participate in filling out these categories, it is necessary to have a technological understanding of how systems are developed. It is therefore the *responder*s that are responsible for these categories.

Filling out configuration table

At the beginning of each sprint in the development process, the team should update the configuration table together. This activity has the following benefits:

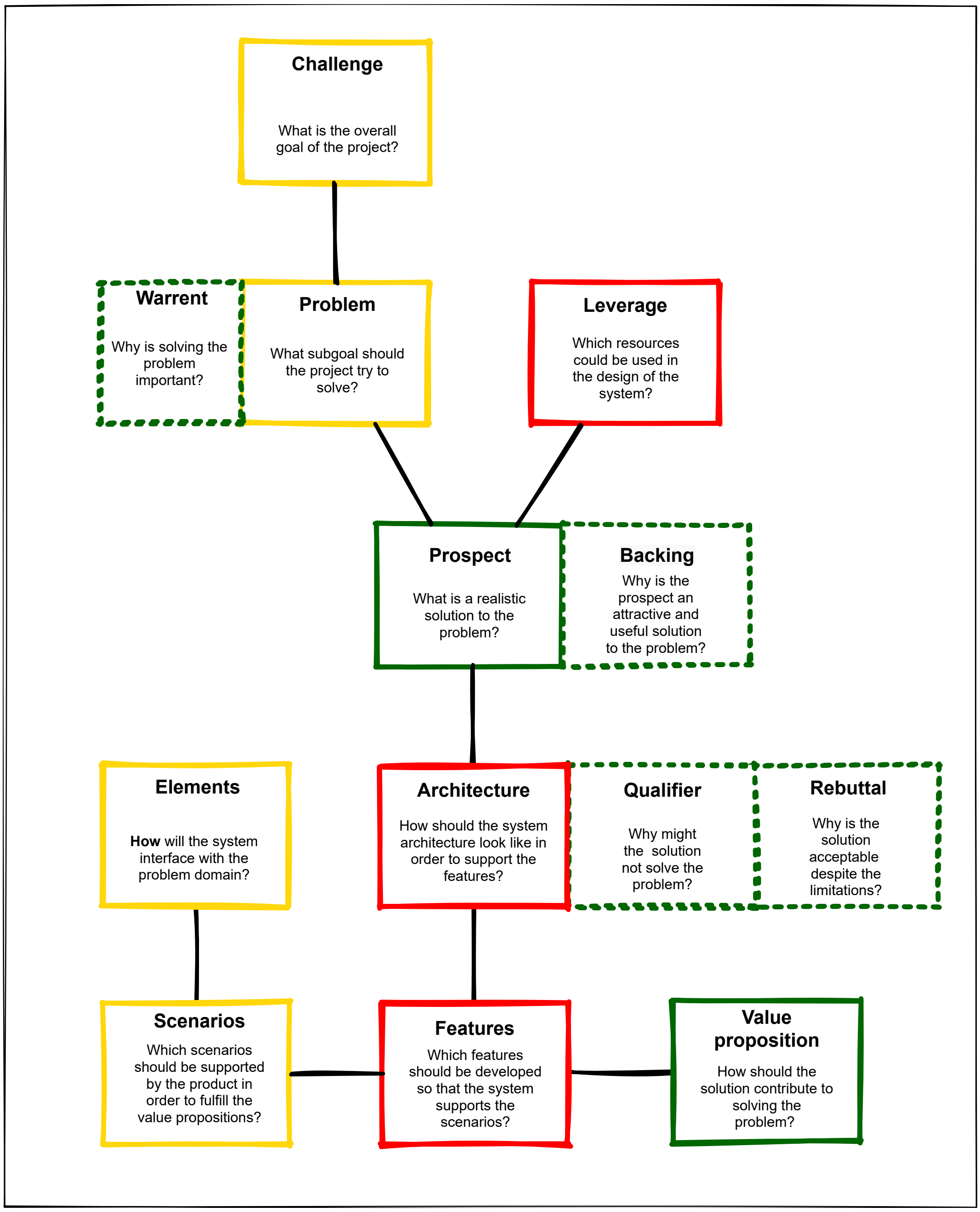

Challenge
What is the overall goal of the project?
Warrent
Why is solving the problem important?
Problem
What subgoal should the project try to solve?
Leverage
Which resources could be used in the design of the system?
Prospect
What is a realistic solution to the problem?
Backing
Why is the prospect an attractive and useful solution to the problem?
Elements
How will the system interface with the problem domain?
Architecture
How should the system architecture look like in order to support the features?
Qualifier
Why might the solution not solve the problem?
Rebuttal
Why is the solution acceptable despite the limitations?
Scenarios
Which scenarios should be supported by the product in order to fulfill the value propositions?
Features
Which features should be developed so that the system supports the scenarios?
Value proposition
How should the solution contribute to solving the problem?

Figure 2.6: Configuration diagram.

- The goal and status of the projects become clear.
- Any gap in the project will become clear.
- The table helps facilitate brainstorming in the team.

We will go through each category and explain what the idea is and how it fits with other categories:

Challenge: The challenge category contains the domain of the project and explains why the project is needed in the first place.

Problem: The problem contains a specific, solvable, part of the challenge that the project aims to solve. An initial problem is defined during the pre-project, but the problem might be updated or changed during the project, if the team learns that the problem might not be solvable, or identify another more appropriate problem.

Warrant: The warrant describes why the problem is worth solving. This can be worth considering, as a product that solves a problem no one cares about, is unlikely to be successful. Some questions that could be relevant here are:

- How many people face the problem?
- How important is a solution for the people facing the problem?
- Does an appropriate solution to the problem already exist?

Answering these, as well as other relevant questions ensures, that there would actually be a market for a good solution to the problem.

Value propositions: Value propositions are statements about the specific value we want the product to offer.

Elements: The elements category is used to understand how the system will interface with the problem domain.

Scenarios: Knowing how the problem domain will interface with the system, defined in the *elements* category, and how the problem should be solved, defined in the **value propositions** category, we can define the scenarios, or use-cases, for how we will provide the value propositions given the elements in the problem domain.

Qualifier: Even if the further technical design supports the scenarios perfectly, they will often have limited coverage of the value proposition, e.g., they only support a portion of the users in the problem domain, or they might not solve the problem completely, only mitigate it. In the qualifier, we list the reasons why the solution does not necessarily solve the problem. The idea is that we should be aware of the limitations of the solution.

Rebuttal: Given the limitations listed in the **qualifier** category, the rebuttal defines why these limitations are acceptable. This might be that even a partial solution is better than no solution. This way, the team will always be aware of the limitations in the solution, as well as discuss whether it is worth developing despite the limitations.

Features: After defining the product scenarios, we can extract features that should be included in the solution, to provide the functionality needed to support the scenarios.

Leverage: In the pre-project, we found leverage points that could be used to build the solution. These served as inspiration for what is possible to develop, helping to define an initial problem. During the project, the list of leverage points can be updated, as the team learns about new knowledge and technologies they can leverage in the solution. The list of leverage points should not be exhaustive, and the leverage points do not have to be used in the solution, but the list serves as inspiration for the developers.

Architecture: Knowing which features should be included in the design, as well as which technologies we have available, we can design the overall structure of the solution. This includes:

- Which modules should be included in the design?
- How should the modules be organized?

Prospect: Given the value propositions in the project, as well as the architecture of the solution, we can define the prospect in the project. The prospect contains the essence of what we believe would solve the problem if properly adopted and used by the target users.

Backing: The last category, **backing**, contains reflections about why the prospect is an attractive and useful solution to the problem.

2.10.1 Example of configuration

To illustrate how the configuration table can be used in practice, we use the health system example.

Figure 2.7 shows the filled-out configuration table for the health system. It is important to note that this is in no way an exhaustive analysis of the project, but instead a quick overview of the thoughts at the current point in time. The purpose of the configuration table is to be a tool for discussion and thoughts.

2.11 Levels

When filling out, or updating the configuration table, we can take three different starting points, either **why** we plan to build the current solution, the rationale, **what** we are building, the strategy, and **how** we plan to build it, the tactic.

The rationale level includes the basis for the project, that is the *challenge* we are working on, and the specific problem we are trying to solve in the project. It also includes the *leverage*, as the available leverage points define the opportunities for solving the problem. Based on the problem and the available leverage points we can then define the *prospect*. The problem is supported by the *warrant*, while the prospect is supported by *backing*.

The strategic level is concerned with what we plan to build, what the system as a whole is going to look like. This involves both the system *architecture*, the overall structure of the system, and the *elements* and *ecology*. Together, this paints a picture of the solution as a whole. We reason about the solution using the qualifier and rebuttal.

The tactical level is concerned with how we are going to build the solution, this involves which *value propositions* the system should provide, which *scenarios* the system should support to support the value propositions, and which features there should be included in the solution to support the scenarios.

The categories are connected across the levels, and the configuration of one level can affect the others. E.g., the architecture and prospect are connected, and without having an idea of what the prospect is, designing the architecture will not make much sense. However, using the levels as a starting point can support updating the configuration table in a divide and concur approach.

A configuration diagram with the levels included is shown in figure 2.8.

2.12 RST Review

Every choice made in the design and implementation of the product is based on assumptions about the problem domain and the ability to actually implement the solution. During the course of the project, the team learns more about the problem and possibles for the solution. To take advantage of new knowledge, it is important to adapt the project accordingly. In ESSENCE, we do this by reviewing the project status, during the *Rationale, Strategy, and Tactics Review (RST Review)* activity, where each level of the configuration table is reviewed. The outcome of this activity is:

1. Feedback about the product(s).

2. Updated configuration table.

The RST Review should be done between every major activity in the project. This could e.g., be after each Learn step in the build, measure, learn loop.

The RST Review is facilitated by the *Anchor*, and both the *Challenger(s)* and *Responders* participate. In addition, other people, like stakeholders and customer representatives can participate when needed.

The review can be done using the following approach:

- The review should be brief, and not go into details.

- Before the review, the *Anchor* should define a list of items, products, or features that should be reviewed. There should be developed a list of criteria for each item.

- At the beginning of the review, the *Responders* should do a short presentation of each item.

- The *Challenger(s)*, guest stakeholders, and customer representatives should comment first.

- Comments should be noted but not debated. The Responders should only speak when asked.

- After all items are reviewed, the development team discusses the project status and updates the configuration for each level.

2.13 Prospect Scenarios

The prospect is the solution we at a given time think will solve the problem. The word prospect is used instead of solution, as the prospect can change during the course of the project. The prospect is based on a set of hypotheses about the problem and how to solve the problem, and these serve as the basis for the actions taken during the project. To explore problems and solutions, ESSENCE introduces the concept of prospect scenarios.

As the prospect is related to both the problem and the prospective solution, the prospect scenario consists of two sub-scenarios:

- **Problem Scenario**, used to explore the Problem Domain
- **Solution Scenario**, used to explore the Solution Domain

We use Problem Scenarios to explore different perceptions of the problem, and solution scenarios to explore different solutions to the problem.

A prospect scenario contains two axes, with each axis being labeled with two options for both the problem and solution. In this way, we can contrast different scenarios for the problem and solution. For the problem, that could be *who* has the problem, *when* is the problem experienced and *where* is it experienced. For solutions, these axes could be *why* does this solve the problem, *what* does the solution do and *how* does it solve the problem.

Using the prospect scenario involves two steps:

1. Selecting axis
2. Defining prospects

Step 1: Selecting axis

For the *health system*, we can have the *who* axis for the Problem Scenario could be either the patient finding it difficult to decide whether or not to go to a doctor or the doctor wanting to help his patients make a more informed choice of whether to book an appointment.

For the other axis, we can ask *when* the problem occurs. For the health system, this could be (A) when the patient is experiencing the symptoms for the first time, or (B) as the last step before booking an appointment. If we target A, the goal of the system will be to encourage the user to seek medical attention if the symptoms suggest that something serious is happening, while for B, the goal will be to discourage booking an appointment if the symptom does not suggest that something serious is wrong.

For the solution, on the first axis, we can have that the solution is either an online website or a tool that the doctors' secretary can use when people contact her. On the other axis, we can have that the solution either encourages people to go to the doctor or dismiss symptoms that there is clearly no reason to contact a doctor for.

The axis for the problem and solution should be aligned reasonably. For each configuration of the axis, we have one quadrant. The idea is then for each quadrant to fill it out with a prospect, giving us multiple angles on the problem and solution.

Step 2: Defining prospect

Representing quadrants:

1. **Prototype**, suggest a physical design for the prospects.
2. **Metaphor**, point to strategies and principles from other domains to solve problems that share similarities with the project.
3. **Proposition**, a single statement that describes the rationale of the project and high-level expectations for the solution.
4. **Icon**, symbolic representations like images that illustrate key qualities in the solution.

Each team member should choose a distinct representation, and then alone come up with ideas for each quadrant using the chosen representation. For scenario 1 we treat the patient as the main user, and the patients should interact with the system through a website. We also primarily focus on encouraging the patient to seek medical attention when experiencing a symptom if this could indicate a serious underlying condition.

For quadrant 2, we also treat the patient as the main user, interacting through a website, but the primary focus in the solution is to make sure that the symptom that is the basis for seeking medical attention actually could be caused by an underlying condition.

In the end, we chose a quadrant and elaborate on the ideas to inspire the configuration table.

2.14 Pivot or persevere

If the development follows the build, measure, learn approach, we will test these assumptions after each iteration. This can result in new knowledge, that a given scenario does not provide the value intended. In a situation where we learn that fundamental assumptions about the solution are wrong, we must make a decision; should we persevere, and continue with the current prospect, or reconsider the solution to the problem, thereby beginning the development of a new prospect.

A pivot is a drastic decision that will often lead to developed products being more or less wasted time. It is therefore not a choice that should be taken lightly. However, a pivot can be considered in two situations:

- The team identified fundamental weaknesses or threats to the solution during the Project Review.

- The team identified major opportunities that could lead to a much better solution, during the review.

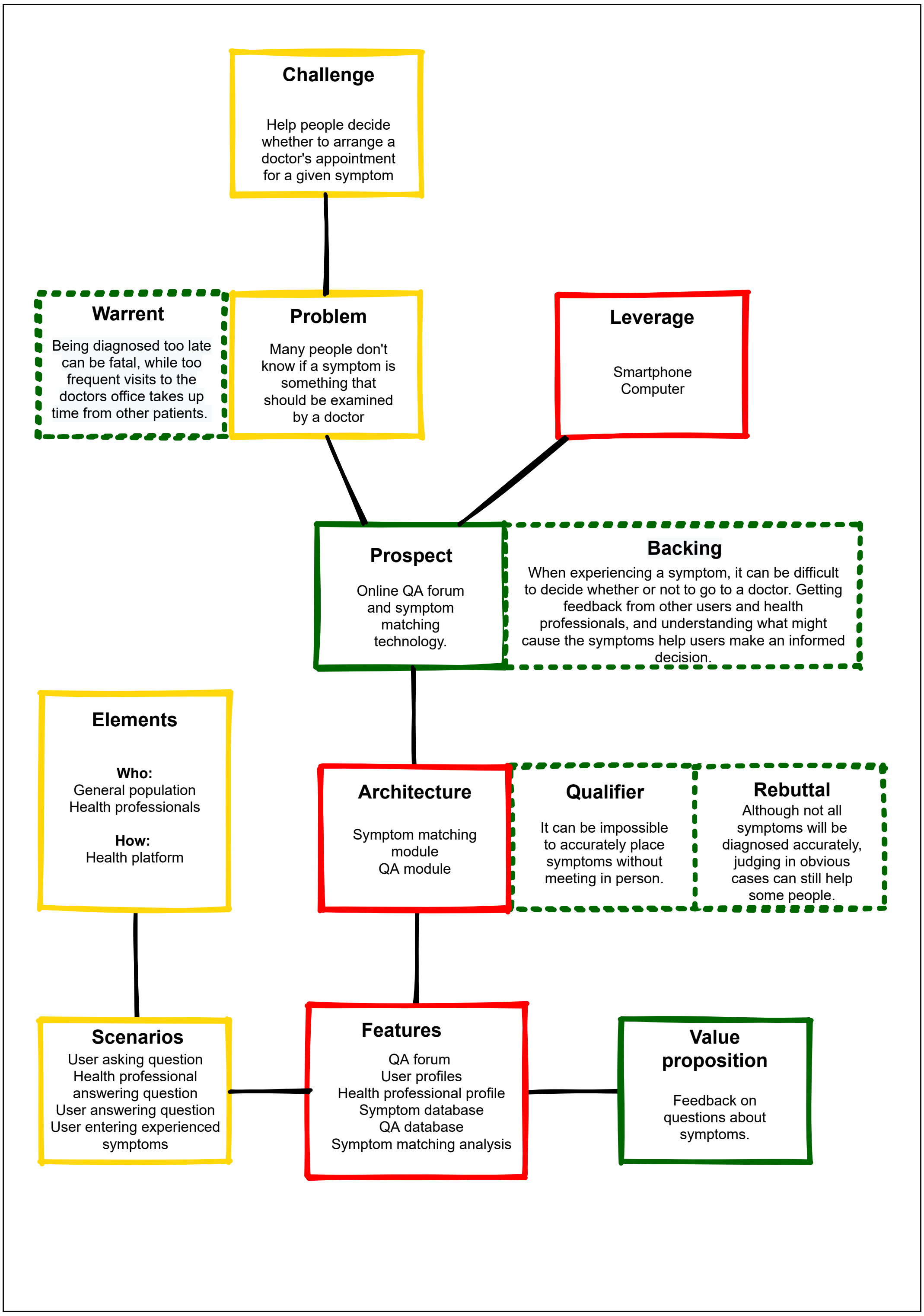

Challenge
Help people decide whether to arrange a doctor's appointment for a given symptom
Warrent
Being diagnosed too late can be fatal, while too frequent visits to the doctors office takes up time from other patients.
Problem
Many people don't know if a symptom is something that should be examined by a doctor
Leverage
Smartphone
Computer
Prospect
Online QA forum and symptom matching technology.
Backing
When experiencing a symptom, it can be difficult to decide whether or not to go to a doctor. Getting feedback from other users and health professionals, and understanding what might cause the symptoms help users make an informed decision.
Elements
Who:
General population
Health professionals
How:
Health platform
Architecture
Symptom matching module
QA module
Qualifier
It can be impossible to accurately place symptoms without meeting in person.
Rebuttal
Although not all symptoms will be diagnosed accurately, judging in obvious cases can still help some people.
Scenarios
User asking question
Health professional answering question
User answering question
User entering experienced symptoms
Features
QA forum
User profiles
Health professional profile
Symptom database
QA database
Symptom matching analysis
Value proposition
Feedback on questions about symptoms.

Figure 2.7: Initial configuration table for the health system.

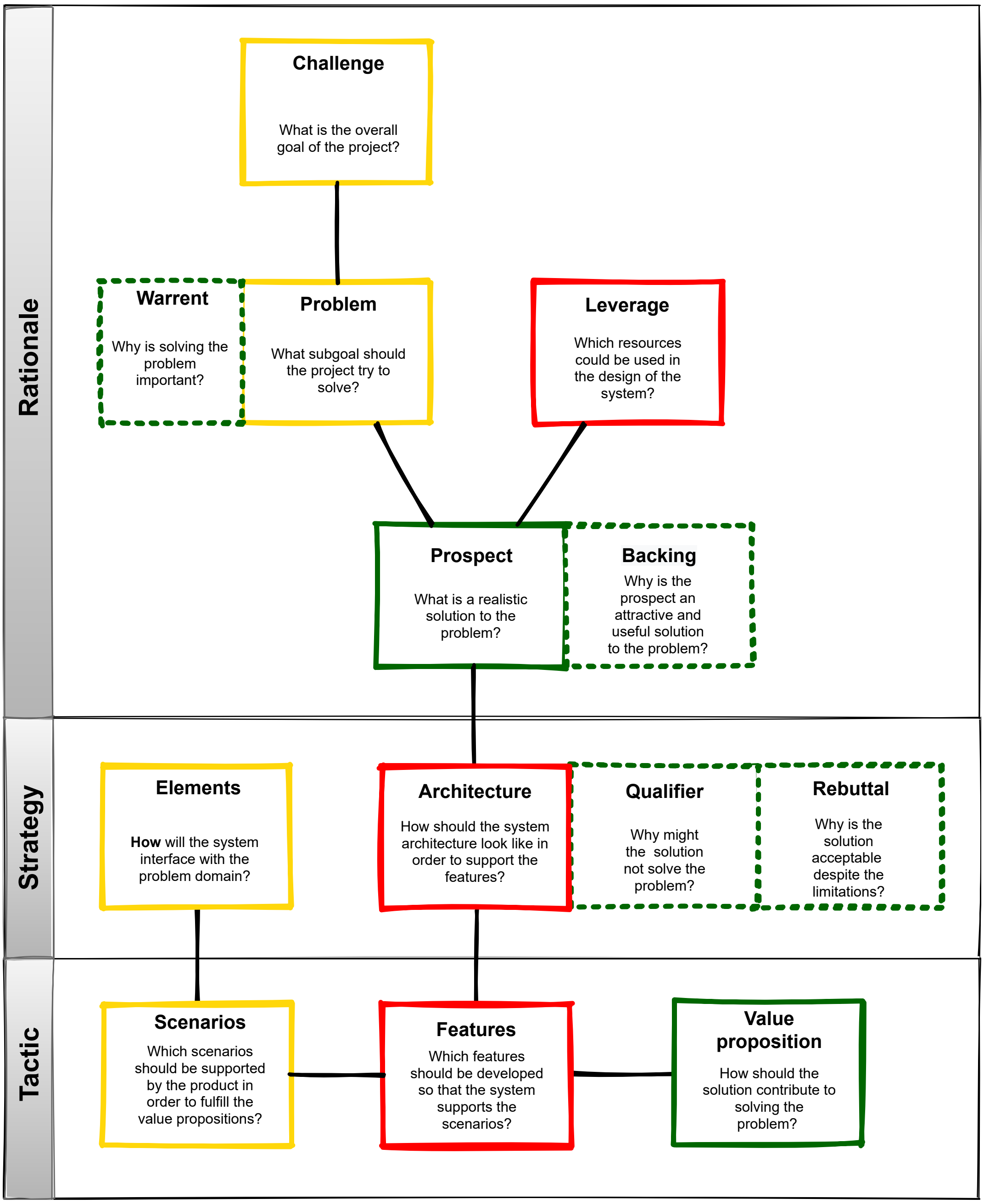

Rationale
Strategy
Tactic
Challenge
What is the overall goal of the project?
Warrent
Why is solving the problem important?
Problem
What subgoal should the project try to solve?
Leverage
Which resources could be used in the design of the system?
Prospect
What is a realistic solution to the problem?
Backing
Why is the prospect an attractive and useful solution to the problem?
Elements
How will the system interface with the problem domain?
Architecture
How should the system architecture look like in order to support the features?
Qualifier
Why might the solution not solve the problem?
Rebuttal
Why is the solution acceptable despite the limitations?
Scenarios
Which scenarios should be supported by the product in order to fulfill the value propositions?
Features
Which features should be developed so that the system supports the scenarios?
Value proposition
How should the solution contribute to solving the problem?

Figure 2.8: Configuration diagram divided into levels.

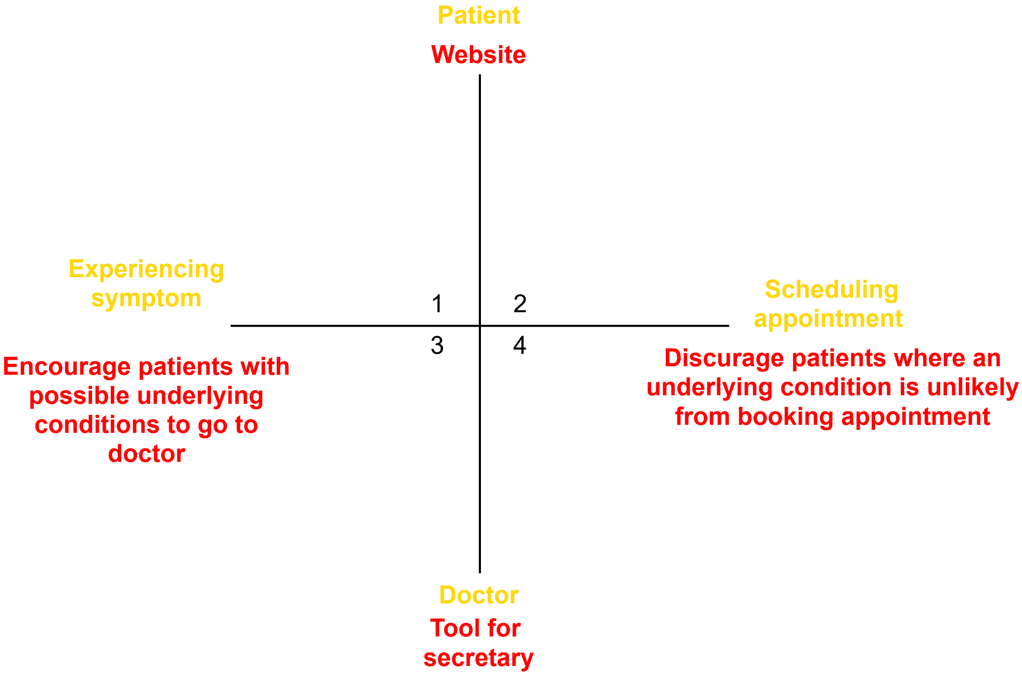

Patient
Website
Experiencing symptom
Encourage patients with possible underlying conditions to go to doctor
1
2
3
4
Scheduling appointment
Discurage patients where an underlying condition is unlikely from booking appointment
Doctor
Tool for secretary

Figure 2.9: Axis for health system prospect scenario.

Chapter 3

Approach

3.1 Designing as reflective conversation

The goal of this project is to evaluate the ESSENCE methodology. In order to understand how to conduct an experiment that aids in understanding ESSENCE, it is useful to have insights into how we design.

3.1.1 Seeing-moving-seeing

Donald Schön describes designing as a reflective conversation with design materials [6, 7]. In a design situation, the designer sees the design through some medium, like paper, updates the design based on his interpretation of what he sees, and then sees the design again. This iterative process is called *Seeing-Moving-Seeing.*

This point can be illustrated by the example of a student writing a report. The student writes a draft containing the overall information that should be contained in the report. This contains three sections, A, B, and C, in that order. **[Seeing]** Later, the student reads the draft and identifies that the structure could be improved if he switched the order of section B and C. **[Moving]** He then switches the order, so the report structure is now A, C and B. **[Seeing]** Later he reads the last part of section C and the first part of section B and identifies that they do not seem coherent. This short example contains all three phases. After seeing the second time, the student will update the design based on his interpretations of the new design state, making the second *seeing* stage equal to the first *seeing* stage of a new iteration,

The interpretation of the student in the *seeing* phases is based on his *appreciative systems.* Each designer has his own appreciative systems based on his preferences and experience. If another designer sees the same design, he might have a completely different interpretation of the design, which would result in a different choice of move. In the previous example, the student could send his original draft to his supervisor. After reading it, the supervisor might not find that the order of sections B and C should be switched but instead find that there should be introduced a completely new section, D between them, making the structure A, B, D, and C.

3.1.2 Methodology design

The approach of designing a methodology should be similar to any other type of design. In the report domain, the most important quality is that the report communicates the correct information to the target reader. Therefore, the student, assuming that he understands the target reader, can interpret the quality of a report by reading it and move based on his interpretation. The main quality of any methodology should be that it is useful for intended users. This usefulness can hardly be measured by only learning the methodology, but should be learned through the experience of uses.

ESSENCE is designed for software development teams to use in development projects. Therefore, it should be tested in real development projects. The interpretation development teams have of ESSENCE will depend on their appreciative systems. Appreciative systems are based on personal experience, so by drawing on the appreciative systems of development teams, we can utilize their experience from previous development projects. In order to examine ESSENCE, I, therefore, use ESSENCE in a development project and document my experience with it. As a software student at Aalborg University, I have been a part of several different development projects. My findings will then be interpretations of the quality of ESSENCE, where the modifications I suggest can serve as suggestions for the direction in which ESSENCE should move.

3.2 Project design

In order to evaluate ESSENCE, I work on a project guided by ESSENCE and report my experience of how ESSENCE affects the development process. As I am only one person, I modify the concept of roles as it is not possible to have both a Challenger, Responder, and Anchor. Instead, each role should be fleeting, as the child role, so that it represents an angle on the problem rather than a permanent role.

In order to add structure in my analysis and evaluation of ESSENCE, I fill out my expectations before each activity and then evaluate the usefulness after. I have developed a list of standard questions that I will answer before and after each activity. For readability purposes, the documentation concerning the usefulness of ESSENCE is contained within pink boxes. This is done so it is easy to follow when text is related to the case and when it is related to the evaluation of the ESSENCE methodology.

Questions to be asked before each activity:

What do I expect my solution will be to the problem?

How useful do I expect this activity to be?

What do I expect to gain from this activity?

Questions to be asked after each activity:

How did ESSENCE affect the course of action?

Was the activity overall helpful for the project?

Which benefits did ESSENCE have in this activity? Would they still be benefits in a multi-person team?

Which drawbacks did ESSENCE have in this activity? Would they still be drawbacks in a multi-person team?

Could something be modified in ESSENCE in order to make it more useful for a one-person team at this point? What?

How was the ESSENCE roles and views utilized in this activity?

Other comments

The project is divided into two parts, the pre-project, including the initial fill-out of the configuration table, and the development part. The development will be done in several sprints and at the end of each sprint I carry out an RST review.

Chapter 4

Case initiation

4.1 Project Initiation

4.2 Preface

4.2.1 Challenge

As in any project, the work in this project will be based on a challenge. The overall challenge in this project is based on the problem that the ESSENCE methodology has a steep learning curve, which can make it difficult for development teams to utilize the potential. During this project, the challenge will be to:

Help development teams use the ESSENCE methodology in development projects.

4.3 Pre-project

After the challenge is identified and a team is gathered, the first step in an ESSENCE project is to carry out the pre-project activity. This section documents the pre-project for this project.

4.3.1 Pre-project Expectations

What do I expect my solution will be to the problem?
I expect that the solution will be a website with a user-interface where it is possible to fill out prospect scenarios and maintain and update a configuration table for each product.

How useful do I expect this activity to be?
I expect that this activity will not change my understanding of the problem much, as it is a relatively trivial challenge.
What do I expect to gain from this activity?
I expect to gain a better understanding of the problem and additional insights into the possibilities in the problem domain as well as what technology I can leverage.

4.3.2 Take stock of the challenge

The project challenge is to help developers to learn and use the ESSENCE methodology in development projects. [1] describes the ESSENCE methodology and mentions *take stock of challenge*s as a sub-activity of the pre-project. However, it is never defined what this actually involves.

Initially, I understood this to be defining the challenge as done in 4.2.1. However, I found it difficult to do Ecology object generation and Leverage point generation without any real understanding of the problem. Therefore, I saw the need to expand more on the challenge before continuing with the generation activities.

First, I defined some questions that needed answering.

1. What do I know about the problem the challenge is based on?
    (a) Why does the problem occur?
    (b) When does the problem occur?
    (c) For whom does the problem occur?
2. What could be possible solutions to the problem?

For question 1, we need to explore the underlying problem, that the ESSENCE methodology has a steep learning curve, which can make it difficult for development teams to utilize the potential. To answer *1.a*, we will have to dive deeper into why the ESSENCE methodology has a steep learning curve. As a developer myself and as new to the ESSENCE methodology, I will answer this question based on my experience. The following list suggests why ESSENCE might have a steep learning curve:

1. ESSENCE includes many new concepts that need to be learned at the same time.
2. It can be difficult to understand how the different concepts fit together.
3. It can be difficult to understand why and how a specific concept is useful.
4. There exists no tutorial or tool that can help understand ESSENCE, only a book. As not everybody learns best from a book, it can therefore be difficult for some people to understand ESSENCE.

5. It is difficult to get feedback on whether the concepts are understood correctly, as there are not many people you can ask.
6. It can be difficult to understand what is important to focus on for the different activities.

For 1.b, we want to understand the circumstances of the problem. At present, ESSENCE is almost exclusively used in Aalborg University, where it is taught in a course. However, in the future, if adopted by enough people, it could be taught both in other universities and in dedicated courses like for the Scrum framework. For 1.c we have two groups, the developers who are trying to learn ESSENCE, and the people who are trying to teach ESSENCE.

Based on the different knowledge I gained from answering question 1, I can now answer question 2 and define possible solutions for the problem:

- A tutorial that introduces and explains new concepts one-by-one. The tutorial can have two forms:
  - A supplement to the existing course that can be used to help explain the different concepts during the course.
  - A stand-alone tool that can explain ESSENCE as a whole, or selected parts of ESSENCE
- An ESSENCE tool that can supplement the book by helping people fill out the configuration table and/or the prospect scenarios.
- A tutorial or tool that can test the user's understanding of ESSENCE and provide feedback for what is understood correctly and what is not.
- A tool that can be used for the entire life-cycle of a project and that can guide the user to where to put his focus.

With an overall understanding of the challenge, it should be easier to begin the generation activities.

4.3.3 Defining Relevant Ecology Objects

The first part of defining relevant ecology objects is generating potential ecology objects.

Ecology object generation

Services:
One way of helping development teams with using ESSENCE could be to enable them to generate configuration tables and prospect scenarios. This could be done by interfacing with a diagram tool like **diagrams.net**.

A quiz website that can be used to test the student's knowledge about ESSENCE.

Different process framework tools that the students want to use together with ESSENCE. This can help show how ESSENCE supplements process frameworks.

Brainstorming tools to help students brainstorm in the different activities.

Artifacts:

The development teams could use a **camera** to take pictures of configuration tables and scenarios created by hand so that the system can store the documents, or provide feedback.

Whiteboard where the teacher can present content from the system for the students.

Microphone that can be used to record questions or explanations.

Big screen, like a TV, that development teams could use to brainstorm and carry out activities together.

Repositories:

Repositories with **information about entities** that could be used to suggest possible ecology objects and leverage points to the students, in order to help them generate ideas.

People:

The system could interface with the **teacher** of an ESSENCE course. The teacher could then provide feedback to students or use the tool as a supplement. This gives two possible interface options, *teacher who provides feedback* and *teacher as a supplement of education*. As these would result in different system designs, I will analyze them separately.

Another people interface could be **other students** who are either currently participating in an ESSENCE course or who have already completed the course. These could provide voluntary feedback, or participate in discussions about ESSENCE.

Ecology object filtering

After identifying potential ecology objects, I did ecology object filtering. This is documented in the appendix A.2. I found that the most relevant objects were the *big screen*, a *teacher as a supplement*, and a *whiteboard*.

The scores in the SWOT analysis are generally low. This is probably partly due to that as a one-person group I did not have any sparring partner, which made it difficult to generate ideas for the ecology objects, and because some of the ecology objects proposed I did identify was quite forced. I think that the lack of sparring partners in brainstorming sessions, at least for ecology objects, is a clear drawback when using ESSENCE in a one-person team.

4.3.4 Defining Strategic Leverage Points

The next part of the pre-project is to consider leverage points.

Leverage point generation

For leverage point generation I use the four categories in order to help the brainstorming.

Technologies Teaching knowledge, how do people retain knowledge best

Develop small **games or tutorials** for the different concepts.

Quiz techniques, in order to uncover whether or not people have learned something

Idea generation techniques in order to help students understand the brainstorming activities.

Chat robot that can ask questions about ideas in order to mature the idea.

Artifacts: **Personal computer** for users to interface with the system.

Cellphone for users to interface with the system. Notebook, quick ideas.

Repositories Repository of *earlier solutions and projects* that the students can use to gain inspiration.

People This does not seem relevant at this point.

Leverage point filtering

In Section A.3 the leverage point filtering is documented. In result, we got the prioritized leverage points, **Teaching Knowledge** and **personal computer**. Some other leverage points could prove useful later, especially *cellphone*, *earlier solutions and projects* and *quiz technologies*, and these will be kept in mind.

Comments on leverage filtering

The leverage filtering was easier than ecology filtering, but it was still difficult to get ideas. However, the leverage generation activity forced me to think out of the box, and think of specific solutions to the problem.

4.3.5 Initial Problem

Based on the leverage points and ecology object that I found have a high priority, I define the initial problem.

- Teacher as supplement
- Whiteboard (class)
- Big screen (group work)
- Teaching Knowledge
- Personal Computer

The selected ecology objects and leverage points generally point in the direction that the system should be able to be used during an ESSENCE course. Therefore, I select the initial problem.

It can be difficult for some student to understand ESSENCE without visual explanation and experimentation with the concepts and activities.

4.3.6 Pre-project Evaluation

How did ESSENCE affect the course of action?
Before the pre-project, I focused mostly on the configuration table and the prospect scenarios. The idea was mostly to just make a website where these could be filled out and explained, with some explanations about the different categories. My idea now is more related to the introduction of concepts and activities as a whole, which will likely involve configuration tables and prospect scenarios, but the focus will be on teaching concepts individually more than on using the ESSENCE methodology.

Was the activity overall helpful for the project?
Yes it was.

Which benefits the ESSENCE activity add to the project? Would they still be benefits in a multi-person team?
The activities forced me to think out of the box and think about elements in the problem domain as well as potential keystones of the solution. This gave me a greater understanding of the problem and possibilities. I suspect that this will be even more useful in a multi-person team, as team members can discuss things from different angles. Also, even though it did not seem intuitive for this project to focus on ecology objects and leverage points, as I viewed this as a project with a rather trivial solution, the activities forced me to reevaluate the objective of the project.

Which drawbacks did ESSENCE have in this activity? Would they still be drawbacks in a multi-person team?
Ecology object and leverage point generation were difficult and time-consuming. I imagine that this would be easier if I had one or more sparring partners.

Could something be modified in ESSENCE in order to make it more useful for a one-person team at this point? What?
As it was quite difficult do do the brainstorming alone, there could be added some idea generating questions and procedures that the developer could try out when running out of ideas.

How was the ESSENCE roles and views utilized in this activity?

I did not find a use for the ESSENCE roles and views in this activity.

Other comments

It was difficult in the ecology object generation phase to know where to start and how to identify ecology objects, especially as it was not clear what the system would actually do in the beginning, so it was difficult to be creative and get ideas for how to interface with the ecology. It would probably be useful with some help to get started.

4.4 Prospect Scenarios

After the pre-project and before the development begins, we use prospect scenarios to contrast alternative ideas and perspectives for the project prospect.

4.4.1 Prospect Scenario Expectations

What do I expect my solution will be to the problem?

A tool that can help students get introduced to the central concepts in ESSENCE.

How useful do I expect this activity to be?

I expect that this activity will give me some additional understanding of the problem and put the problem and solution in a new light. However, I also expect that selecting useful axis and filling out the information about representational forms will be difficult.

What do I expect to gain from this activity?

A slightly changed course of action.

4.4.2 Scenarios

Generating prospect scenarios

For the problem scenario, there is an obvious axis, the *who* axis. This axis defines who the main user of the system will be, the teacher or the students. For the second axis, a good suggestion could be the *when* axis, as it is not clear if the tool should be more focused on individual learning at home, or collective learning during the course.

For the solution domain, we can align the *when* axis with a **what** axis where the solution can either be designed for *individual* or *collaboration*. The *who* can then be aligned with a *why* axis, offering either *course supplement* or *learning by doing*.

We therefore have the following axis and discrete options:

- **Who/why**: *Teacher/course supplement, Student/ learning by doing*

- **When/ what**: *At home/ individual, At school/ collaboration*

We now have four quadrants:

- Teacher/course supplement + At home/ individual
- Teacher/course supplement + At school/ collaboration
- Student/ learning by doing + At home/ individual
- Student/ learning by doing + At school/ collaboration

For each quadrant I will fill out the four representational forms; prototype, metaphor, proposition and icon. ESSENCE suggests that each person in the group should select one representational form, and fill out for each quadrant. As I am only one person, I will be filling out all four representational forms for each quadrant.

Teacher/course supplement + At home/ individual

- **Prototype**: The system could be a website, containing resources that the teacher can assign as homework when reaching specific topics in the course.
- **Metaphor**: Moodle quiz - During my education, some teachers have had quizzes that we should fill out either before or after lectures.
- **Proposition**: Contributes to explain concepts and detect when students have not understood them.
- **Icon**: Thermometer - measure the temperature of the students

Teacher/course supplement + At school/ collaboration

- **Prototype**: The website or program that contains resources that the teacher can use during lectures.
- **Metaphor**: PowerPoint diagrams - often it is easier to understand a concept if shown visually like in a PowerPoint presentation.
- **Proposition**: Helps the teacher make concepts less abstract.
- **Icon**: Picture - Help illustrate concepts

Student/ learning by doing + At home/ individual

- **Prototype**: The system can be a website where the students can experiment with ESSENCE concepts individually at home.
- **Metaphor**: Tutorials point - Website with description of concepts and small tasks to back them up.

- **Proposition**: Let student learn about ESSENCE in his own pace and on demand.
- **Icon**: Car - Let students control pace themselves

Student/ learning by doing + At school/ collaboration

- **Prototype**: The system can be a website that can help students use ESSENCE in a development team.
- **Metaphor**: Diagrams.net - website where multiple people can maintain and edit diagrams together.
- **Proposition**: Help individuals learn ESSENCE concepts while also learning how to apply them.
- **Icon**: Hand - Sharing knowledge and helping each others

ESSENCE is meant to be used in development projects. Therefore it would probably be easier to help students understand the value of ESSENCE and how to apply it, if they learned it in relation to development projects, either real or a case. In my opinion, the **Student/ learning by doing + At school/ collaboration** is the most promising quadrant. Therefore I chose to elaborate on this. The result of the elaboration is displayed in 4.1. In order to avoid redundancy, I only show the elaboration result in the figure.

Elaboration on selected prospect

For each representation in the selected prospect I elaborate on it in order to develop the initial configuration table.

Elaboration on prototype:

The prototype tend to focus on the project tactics. More specifically, it is related to the *scenarios*, *features* and *value propositions* categories.

The prospect scenarios and the configuration tables are central tools in ESSENCE, and ideal candidates for development. I therefore center the scenarios around these.

Elaboration on Metaphor:

Elaboration of the metaphors are related to the strategy of the project. This includes elaborating on elements, architecture and qualification.

Elaboration on proposition:

The proposition is related to the rationale of the project. The leverage and problem were identified during the pre project, and still fits with the current understanding of the project. Proposition elaboration includes Problem, leverage and prospect.

Elaboration on Icon:

The icon represents qualities for the design, and the main purpose here is to identify overall themes for the criteria used to determine if the design has the desired qualities. These criteria should be for each of the three levels. Icon elaboration includes rationale, strategy and tactics.

4.4.3 Prospect conclusion

With the elaboration of the four representational forms, I now have the initial configuration table. This will be used as the starting point for sprint 1.

4.4.4 Prospect Scenario Evaluation

How did ESSENCE affect the course of action?
The course of action is more or less the same as stated in the expectations. Therefore, no significant changes were made to the course of action. However, I do have a much clearer vision about the project direction and have considered the options carefully.

Was the activity overall helpful for the project?
Yes, it was much more helpful than I expected. Before I started this activity, I expected that it would be difficult to identify useful axis, and I did not understand the purpose of the representational forms and the division for the different levels. However, it was actually very intuitive, and provided the right amount of consideration for the different alternatives.

Which benefits did ESSENCE have in this activity? Would they still be benefits in a multi-person team?
ESSENCE enabled me to see the problem from multiple different views, and consider multiple options that I likely would not have considered otherwise. However, I think that it was really useful to be involved in the elaboration of all representational forms, as they are very connected, and I am not sure that making one person in charge of each in parallel, as is the case in multi person teams would provide the best result.

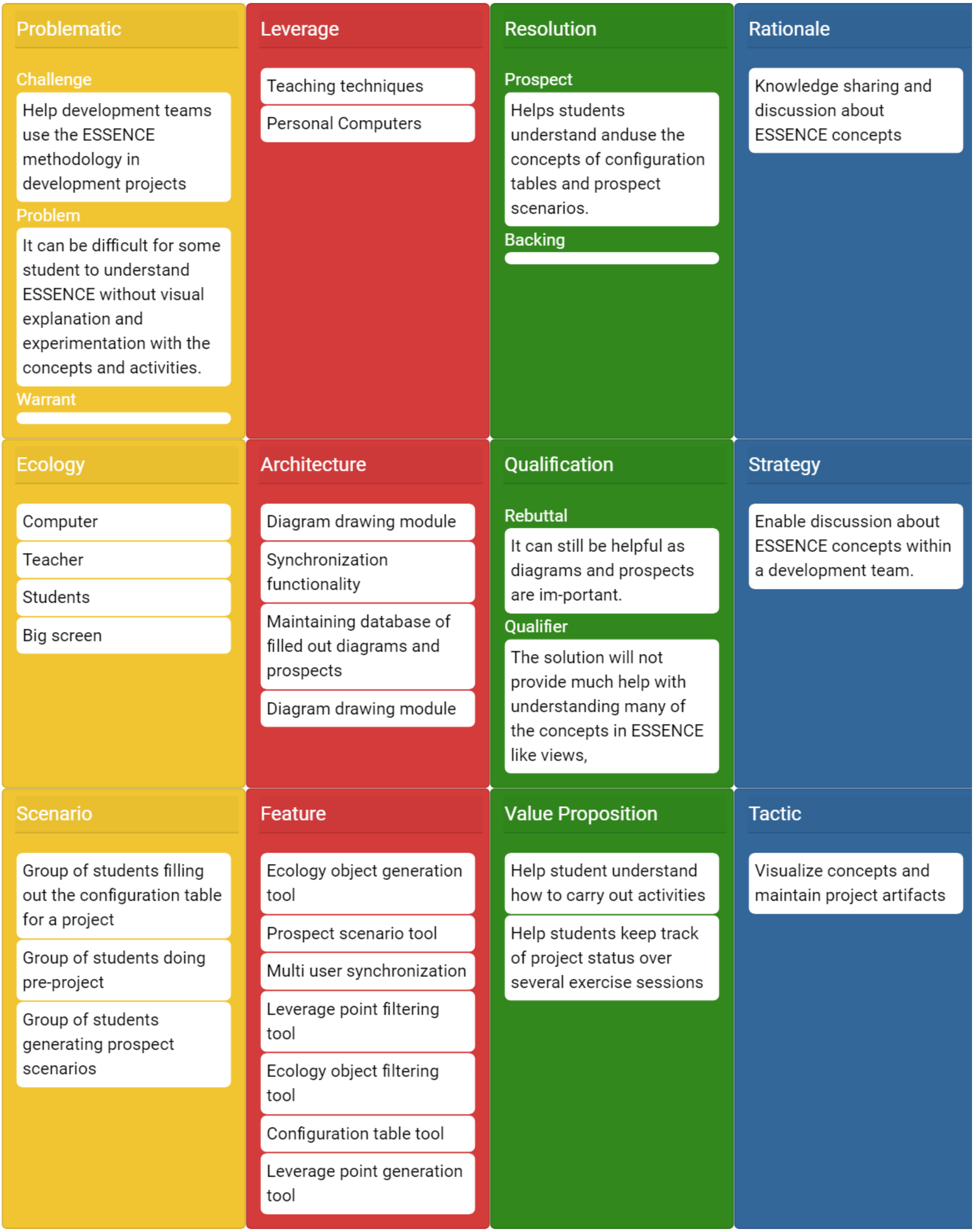
Which drawbacks did ESSENCE have in this activity? Would they still be drawbacks in a multi-person team?
I did not identify any drawbacks for this activity. It was very useful.

Could something be modified in ESSENCE in order to make it more useful for a one-person team at this point? What?
I did not identify anything that should me modified.

How was the ESSENCE roles and views utilized in this activity?
The views were an important part of this activity, as the categories are divided into which view they represented, and understanding which view a category represents can help understanding how it should be filled out. I did not explicitly use the roles in this activity.

Other comments
No

Figure 4.1: Configuration table after elaboration.



Problematic Challenge Help development teams use the ESSENCE methodology in development projects Problem It can be difficult for some student to understand ESSENCE without visual explanation and experimentation with the concepts and activities. Warrant	**Leverage** Teaching techniques Personal Computers	**Resolution** Prospect Helps students understand anduse the concepts of configuration tables and prospect scenarios. Backing	**Rationale** Knowledge sharing and discussion about ESSENCE concepts
Ecology Computer Teacher Students Big screen	**Architecture** Diagram drawing module Synchronization functionality Maintaining database of filled out diagrams and prospects Diagram drawing module	**Qualification** Rebuttal It can still be helpful as diagrams and prospects are im-portant. Qualifier The solution will not provide much help with understanding many of the concepts in ESSENCE like views,	**Strategy** Enable discussion about ESSENCE concepts within a development team.
Scenario Group of students filling out the configuration table for a project Group of students doing pre-project Group of students generating prospect scenarios	**Feature** Ecology object generation tool Prospect scenario tool Multi user synchronization Leverage point filtering tool Ecology object filtering tool Configuration table tool Leverage point generation tool	**Value Proposition** Help student understand how to carry out activities Help students keep track of project status over several exercise sessions	**Tactic** Visualize concepts and maintain project artifacts

Chapter 5

Initial development

5.1 Sprint 1

With the initial configuration table in place, I can now begin the development of the product. Each sprint will be about two weeks, and use the configuration table from the previous activity as a starting point. The goal of each sprint is to develop a part of a solution, measure how appropriate the solution is, and update the configuration table based on what I learned. At the end of each sprint, I will do an RST Review.

5.1.1 Sprint 1 scope

As I cannot develop every part of the system at once, I select a scope at the beginning of each sprint. The purpose of the sprint is develop a product that support the feature(s) selected, and measure if the feature(s) fulfill the value proposition, and attempt to learn something additional about the problem domain.

In the configuration table we have the following features:

- Leverage point generation tool
- Leverage point filtering tool
- Ecology object generation tool
- Ecology object filtering tool
- Prospect scenario tool
- Multi-user synchronization
- Configuration table tool

All the features provide important functionality for using and understanding ESSENCE; Leverage points, ecology objects, and prospect scenarios are important for initiating the project, configuration tables are important for maintaining an overview of the project and multi-user synchronization is essential for collaboration in the development team.

Multi-user synchronization do not make much sense without any of the other features, so this would be a poor starting point. I imagine that leverage point and ecology object activities are more difficult to support, as it is not the overview of leverage points and ecology objects that is most difficult, it is the generation. Therefore, these will wait until future sprints. The initial configuration table is a direct result of the prospect scenarios, so implementing the configuration table could be a natural part of implementing the prospect scenarios. I believe that implementing configuration tables is a good starting point for the project, and after this I can move to other features in later sprints.

5.1.2 Leverage choices

The product for sprint 1 should be some minimal diagram tool. However, even a minimal diagram tool requires me to make decisions about the basic structure of the solution that will support other parts of the product. Therefore, I make an overall analysis of the leverage choices and architecture of the solution. More specifically, I analyze the following points:

- Primary device
- Interface type
- Backend technology
- Frontend technology

Primary device

I identified three potential primary device types: Tablets, Computers, and Smartphones. Table 5.1 shows the SWOT analysis for the primary device.

I did not do a quantitative comparative SWAT analysis, assigning values to each category, but instead a qualitative analysis comparing each device to the others as a whole. At this point in time, the benefits of the PC clearly out weights the others as it is almost guaranteed available for all computer science students during exercise sessions, has a big screen which enables overview and a keyboard which makes it easier to type. Therefore, the primary device for the prototype is the PC.

Interface type

For interface type, there is the option of either developing a program that should be installed on each PC or a website. The SWOT analysis is displayed in 5.2.

	Tablet	PC	Smartphone
S	• Big screen • Touch-screen - Content can be moved around by hand	• Big screen • Keyboard • All developers has a computer	• Most people have a smartphone • Quick to open
W	• Not always a keyboard - writing text can be difficult • Many different OS • Many different formats	• Can be slow to open and start-up	• Small screens • Many different formats • Development is very OS-dependent
O	• Tablets can physically be put side by side in order to provide a complete image	• If the solution works on most OS, it can be adapted to other settings later. • Prospect Scenarios could also benefit from the big screen	• Additional functionality might benefit from the fast start uptime
T	• Probably only few students actually have tablet available	• Some computers have very poor batteries and need to be close to a charger. Therefore there can be settings where a team member cannot contribute.	• The format of cellphones changes often • Some students might not have a smartphone

Table 5.1: Primary device SWOT analysis

	Website	Program
S	• Not OS-dependent, we only need to be concerned with browser • Popular choice, many framework, and forums	• Potential performance benefits
W	•	• Needs installation, many students will find this annoying
O	• Can probably be used on tablets with minor modification • Can likely be adapted to cellphones with little work	•
T	•	• When a new OS is developed, we need to support that also

Table 5.2: Interface type SWOT analysis

It is clear that a website will be the better choice, as this has a lot of benefits, no obvious drawbacks, and good opportunities. Also, the only benefit of a program, performance, is likely not relevant for this project, as it is not performance heavy.

Backend technology

For the backend, I have the option of either setting up a server or using a serverless solution. I do not see setting up a physical server as an attractive alternative, as this requires too much maintenance and is not easily scalable, so a potential server would still be hosted online. With a hosted option there should be no down-time. However, setting up the functionality on a machine, virtual or not, still require some overhead. For the diagram tool, the only functionality required by the backend is to serve the website and maintain a database. Therefore I will go with the fastest possible solution for this, which should be a serverless option. If more backend functionality is required in the future, I will migrate the backend when appropriate. As I am familiar with the Google Cloud platform from a previous project, I will use this as a starting point.

The Google Cloud platform offers several alternative options for website development. At this point in time, the backend should be able to do the following two tasks:

1. Serve the front-end
2. Maintain data

Google Cloud offers the following service types[4]:

1. Preconfigured and click to deploy
2. Static websites
3. Dynamic websites

Type (1) is used to help people develop a basic website quickly. However, using type one restricts the website content and development freedom of the front-end. As it is important in this product to have much freedom in the front-end design, this is not an appropriate option.

The decision is then between a static website and a dynamic website[9]. A static website only serves the code that was written originally, with no user customization. The freedom when using this option is rather limited. However, it is quick to get up and running with a static website[5]. The static solution is called *Firestore*.

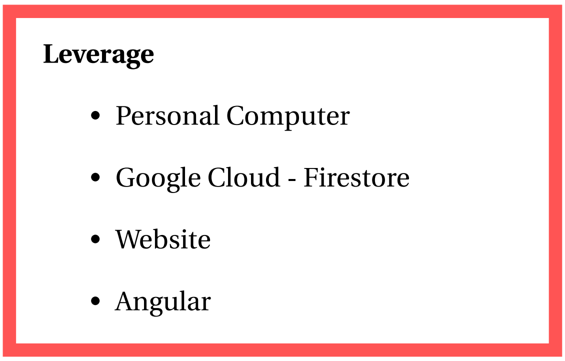
If the content of the solution gets more complex in the future, the solution can be migrated to another serving option. However, for developing the diagram tool, the static options seem better for getting something up and running, especially considering my limited development experience.

Frontend technology

I develop the frontend using the Angular framework as I am familiar with this framework from an earlier project and this is a popular web development framework developed by Google. Therefore there are likely many online resources for both Angular development and interoperability with Google cloud.

Leverage table

An updated version of the leverage table is displayed in 5.1 and the architecture is displayed in 5.2.



Leverage

- Personal Computer
- Google Cloud - Firestore
- Website
- Angular

Figure 5.1: Sprint 1 beginning leverage category.

Architecture

- Firestore database
- Firestore hosting
- Angular frontend
  - Diagram drawing module
  - *Prospect scenario module*
  - *Leverage point generation module*
  - *Leverage point filtering module*
  - *Ecology object generation module*
  - *Ecology object filtering module*
- *Synchronization module*

Figure 5.2: Sprint 1 beginning architecture category.

5.1.3 Sprint 1 feature expansion

The goal of sprint 1 is to develop a configuration table tool. This feature is an overall feature that can be viewed as a collection of many sub-features. ESSENCE does not provide any clear guidelines for sub-features, but after being in dialog with the ESSENCE author, Ivan Aaen, I decided to expand the feature category with sub-features for overall features. I name this activity **feature expansion** and view it as a part of the product design. An exhaustive description of all extracted features and the rationale can be found in Appendix A.4.

Figure 5.3 shows the updated feature category including sub-features. During this sprint, it is unlikely that I will be able to implement all the sub-features, especially as I am also getting used to the development tools. Therefore, I start with 1.a through 1.e, as these are the fundamental functionality without which the solution would not make sense. The other items are also more or less important for a good solution and will be considered in future sprints.

Features

1. Configuration table tool
   - (a) Separate containers for each category
   - (b) Category items separated into distinct units
   - (c) Add new items
   - (d) Delete items
   - (e) Edit items
   - (f) *Connect category items across categories*
   - (g) *Make users aware when an item is not properly backed*
   - (h) *Be able to make items active and inactive*
   - (i) *Maintain configuration tables*
   - (j) *Sub-items*
   - (k) *Change ordering of category items*
   - (l) *Keep track of item status*
2. *Leverage point generation tool*
3. *Leverage point filtering tool*
4. *Ecology object generation tool*
5. *Ecology object filtering tool*
6. *Prospect scenario tool*
7. *Multi-user synchronization*

Figure 5.3: Updated feature category. Italic items are reserved for future sprints.

5.1.4 Product Build

During sprint 1 I implemented a website with all the features I selected in 5.1.3. The resulting UI is shown in 5.4 and contains a single page. Additionally, I implemented Firebase hosting and maintain a single configuration table globally.

5.1.5 Product evaluation

At this point, it would be useful to collaborate with real students of an ESSENCE course in order to evaluate the usefulness of the product at the current time. However, I have only implemented a basic configuration table layout at this point. As I do not personally know any students on the ESSENCE course, it seems like a waste of resources to recruit test persons at this point, and given the covid lockdown, it is doubtful that I will be able to meet with anyone in person.

Instead, I get comments from the ESSENCE author Ivan Aaen in the first part of the RST review, and use the product in practice for the second part of the RST review and note my thoughts. The resulting evaluation is described in A.5.

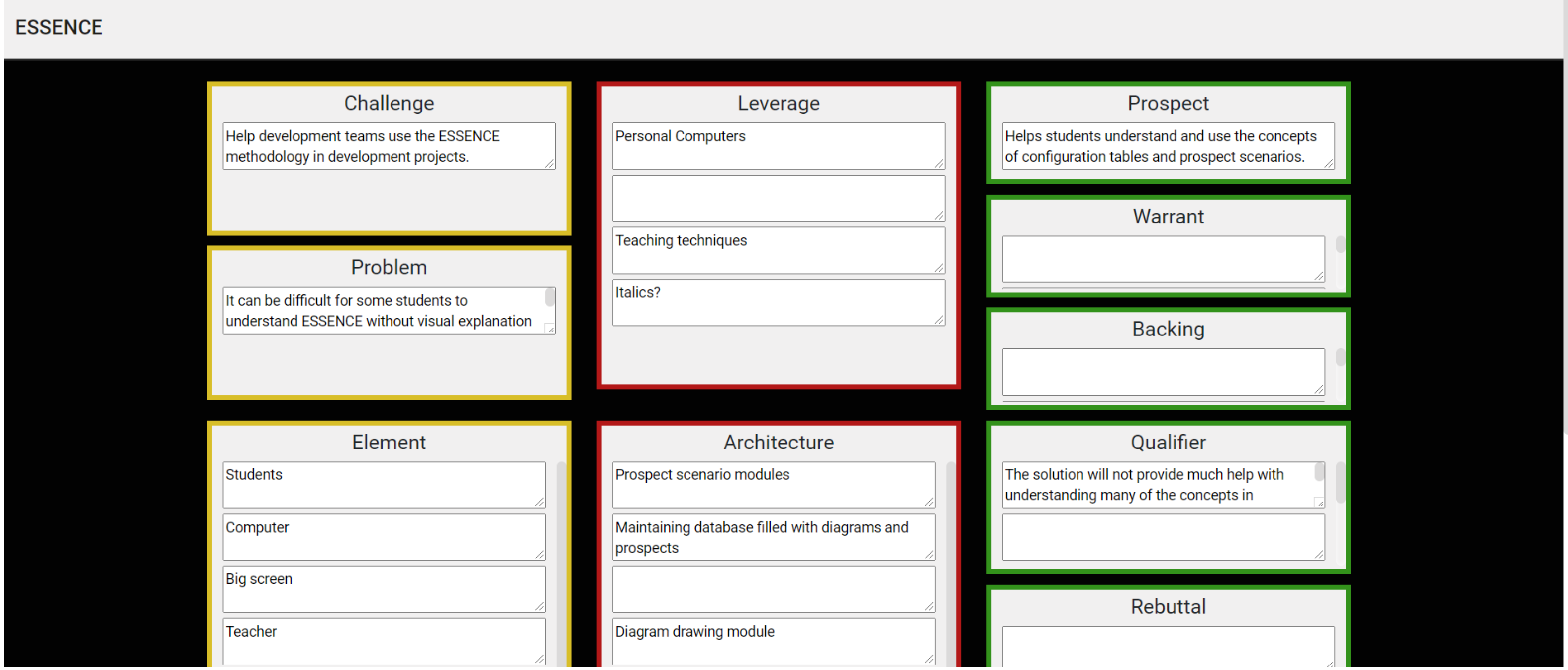
5.1.6 Reflections about ESSENCE

Before sprint 1 I did not write down my expectations, as I did not expect ESSENCE to influence the sprint significantly before the RST Review activity. However, during the sprint, the features and leverage categories, and comparative strategies, have played a fundamental role in the product design and prioritization, and helped structure the development and encourage reflection about the choices.

The reevaluation of ESSENCE categories outside the RST review is however not explicitly encouraged by the ESSENCE methodology. In my opinion, ESSENCE could benefit from activities like the feature expansion activity, in the beginning of each sprint, like sprint planning.

Without any activity like this, the configuration table is not directly linked to the actual work output in the sprint, but is mostly useful to get an overview of what the project is. However, directly connecting the features category to a product backlog provides a direct link between the conceptual understanding of the project, in the configuration table, and the actual project work during the sprint.

Additionally the leverage category and the architecture category are useful when selecting the basic structure of the product. Therefore it would in my opinion be beneficial to update the categories during the sprint and use ESSENCE both to understand and guide the project.



Challenge

Help development teams use the ESSENCE methodology in development projects.

Problem

It can be difficult for some students to understand ESSENCE without visual explanation

Element

Students

Computer

Big screen

Teacher

Leverage

Personal Computers

Teaching techniques

Italics?

Architecture

Prospect scenario modules

Maintaining database filled with diagrams and prospects

Diagram drawing module

Prospect

Helps students understand and use the concepts of configuration tables and prospect scenarios.

Warrant

Backing

Qualifier

The solution will not provide much help with understanding many of the concepts in

Rebuttal

Figure 5.4: Product UI in the end of sprint 1

5.1.7 RST Review expectations

What do I expect my solution will be to the problem?
I expect that I will continue the development of the product, and expand it with the other features and better backend support. I also expect that I will add support for actually using the configuration table.
How useful do I expect this activity to be?
I think that I have a relatively clear understanding of the scope of the project at this point, so I am not sure that this activity will be that useful.
What do I expect to gain from this activity?
A sharper vision for the project and possibly identify some shortcomings that I have not yet considered.

5.1.8 RST Review

The input for this review is:

- Product: The website
- Configuration table: The initial configuration table from 4.4.2 with updated leverage from 5.3, architecture from 5.2 and features from A.4. I include the subfeatures as these can be useful to get a complete overview of my mental model for the project.
- Checklist based on the criteria outlined in the configuration table from 4.4.2.

Developing checklist
Before I can begin the activity, I define the checklist explicitly. For the rationale criteria to be fulfilled, the prospect should support *knowledge sharing and discussion about ESSENCE concepts*. The goal solution should *enable discussion about ESSENCE concepts within a development team*, and this should be done by *visualizing the concepts and maintaining project artifacts*.

Participants

As I am only one person, I have to adjust the procedure so that it fits for a one person team. In principal, I am both a producer, the review leader, and a user representative. The only external participant is Ivan Aaen, who participated in the first part of the activity by offering comments on the product state.

Review

This activity has two parts, (1) feedback from stakeholders and (2) updating the configuration table.

For (1) I showed the product to Ivan Aaen and noted his comments.

For part (2) I updated the configuration table and noted my thoughts.

The notes from (1) and (2) can be found in Appendix A.5.

Before ending the session, I consider each criterion. **Rationale criteria**: Do the prospect support knowledge sharing and discussion about ESSENCE concepts? **Answer**: It does support knowledge sharing and discussion about the activities, however, there has not been proper consideration about other concepts. For instance, views are introduced in colors in the configuration table, but there is no real consideration about explaining what they actually mean. One feature that could improve this is to include view ideation as a feature. This gives us a new scenario; *Students exploring different views of the project*, that is related to a new value proposition; *Help students see the project from different views*. This adds views to the prospect. Views are very important in ESSENCE but were neglected in the previous prospect. With the updated configuration table, the prospect does in my opinion support knowledge sharing and discussion about ESSENCE concepts.

Strategy Criteria: Does the designed solution enable discussion about ESSENCE concepts within a development team? **Answer**: The designed solution includes synchronization of working artifacts which is very important for discussion in a development team. With my current understanding, I estimate that, these criteria is fulfilled.

Tactics question: Do the features visualize the concepts and maintain project artifacts? **Answer**: Yes.

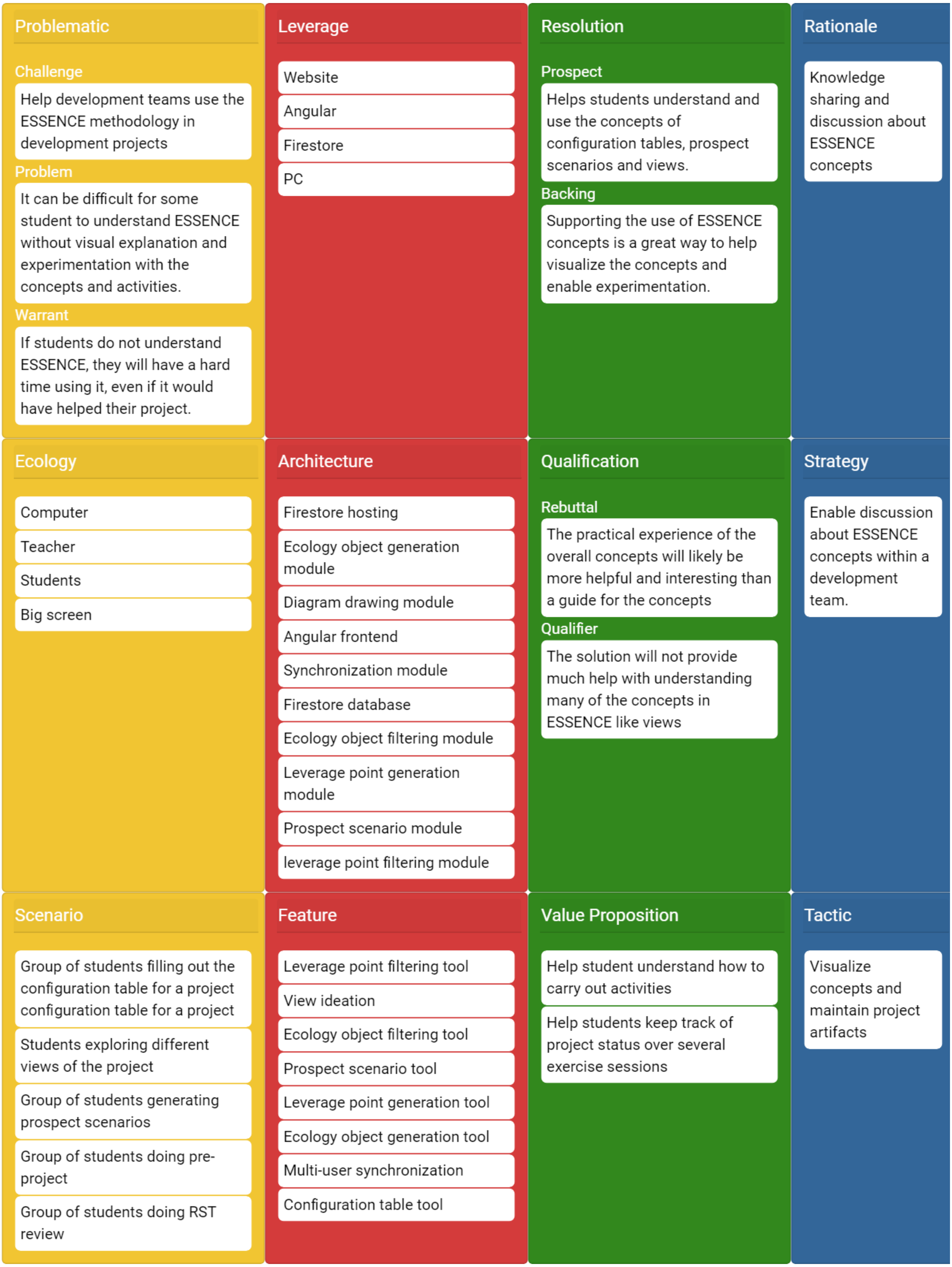
Resulting configuration table

To get an overview of the current status of the project, the configuration table for the project at this point is displayed in 5.5.

5.1.9 RST Review Evaluation

How did ESSENCE affect the course of action?
ESSENCE forced me to consider whether the current configuration actually lived up to the goal of the project. I did introduce a few new items for the configuration table, but my understanding remains almost the same as before this activity.
Was the activity overall helpful for the project?
Not that much. At this point in the project, it would probably have been more helpful to just continue development. Even though it is not a long activity, it still takes time and energy from other tasks.

Figure 5.5: Configuration table after sprint 1.



Problematic

Challenge

Help development teams use the ESSENCE methodology in development projects

Problem

It can be difficult for some student to understand ESSENCE without visual explanation and experimentation with the concepts and activities.

Warrant

If students do not understand ESSENCE, they will have a hard time using it, even if it would have helped their project.

Leverage

- Website
- Angular
- Firestore
- PC

Resolution

Prospect

Helps students understand and use the concepts of configuration tables, prospect scenarios and views.

Backing

Supporting the use of ESSENCE concepts is a great way to help visualize the concepts and enable experimentation.

Rationale

Knowledge sharing and discussion about ESSENCE concepts

Ecology

- Computer
- Teacher
- Students
- Big screen

Architecture

- Firestore hosting
- Ecology object generation module
- Diagram drawing module
- Angular frontend
- Synchronization module
- Firestore database
- Ecology object filtering module
- Leverage point generation module
- Prospect scenario module
- leverage point filtering module

Qualification

Rebuttal

The practical experience of the overall concepts will likely be more helpful and interesting than a guide for the concepts

Qualifier

The solution will not provide much help with understanding many of the concepts in ESSENCE like views

Strategy

Enable discussion about ESSENCE concepts within a development team.

Scenario

- Group of students filling out the configuration table for a project configuration table for a project
- Students exploring different views of the project
- Group of students generating prospect scenarios
- Group of students doing pre-project
- Group of students doing RST review

Feature

- Leverage point filtering tool
- View ideation
- Ecology object filtering tool
- Prospect scenario tool
- Leverage point generation tool
- Ecology object generation tool
- Multi-user synchronization
- Configuration table tool

Value Proposition

- Help student understand how to carry out activities
- Help students keep track of project status over several exercise sessions

Tactic

Visualize concepts and maintain project artifacts

Which benefits did ESSENCE have in this activity? Would they still be benefits in a multi-person team?

I did consider some aspects that I might not have otherwise.

Which drawbacks did ESSENCE have in this activity? Would they still be drawbacks in a multi-person team?

The criteria were difficult to use, and the description of the checklist is not very clear. Also, at this point, the use for elements, leverage, and architecture is not very clear, which made it difficult to actually update the configuration table. It is not clear what the purpose of the Architecture category is, as it is only described in the book as which modules it contains.

Could something be modified in ESSENCE in order to make it more useful for a one-person team at this point? What?

In my opinion the Architecture category should describe the overall structure of the solution rather than just which modules it contains, as this does not seem that useful. It is possible that this is really the intention, but it does not appear this way when reading the book.

Also, it is unclear what the elements category is exactly. It seems to be a mixture of *users*, *interface types*, *situations*, and ecology. I think this is way too broad to really understand what the purpose is, which makes it difficult to use it. I think that it would be reasonable to have a completely separate category, or maybe a subcategory, for users, as most systems are designed for a specific user, and it is always helpful to have a clear understanding of who the primary user of the product really is. For this project it is *students*. Connected to this category could be another category called *situations*, which describes for which situations the users will need the product. In this project, this could be *during exercise sessions*. Being mindful of the situation could then help the ESSENCE user generate one or more specific scenarios for each situation.

How was the ESSENCE roles and views utilized in this activity?

The views were used when considering the different category items. I don't think the concepts of roles played any real part in the project at this point.

Other comments

In this case, I did not think the activity was really useful. However, the project is at a very early stage, so not much has happened since I filled out the initial configuration table. Therefore it seems reasonable that the activity did not have a huge impact. It is realistic that this will be very different in future sprints.

5.2 Sprint 2

5.2.1 Feature expansion

Before beginning sprint 2, I update the features based on my comments from the RST review in sprint 1. In order to limit cluttering the feature table, I delete already implemented functionality sub-items. Figure 5.6 shows the updated sub-features for the configuration table tool. As this is the only part of the product that is implemented, I only included the configuration tool sub-features for simplicity. The items are ordered by priority.

5.2.2 Scope

As the configuration diagram tool only contains basic functionality at this point, and is not that much more helpful than a whiteboard, I continue on the configuration diagram tool in this sprint. Much of the testing will be easier if I have user functionality, and as I am sure user functionality will be necessary at some point in order to make this tool useful in any way, I chose this as the basis for this scope. As the items in figure 5.6 are ordered by the priority, additional time during the two week sprint will be used for adding features further down the list.

5.2.3 Product Build

During sprint 2 I implemented the following features:

- User functionality
- Maintaining several diagrams for each user
- Highlighting of connected categories when holding mouse over category
- Ability to connect and disconnect category items in related categories
- Ability to make item active/passive
- Ability of user to create and delete projects

The UI now contains login related screens, a project page and the diagram overview page. This is shown in 5.7b.

5.2.4 RST Review expectations

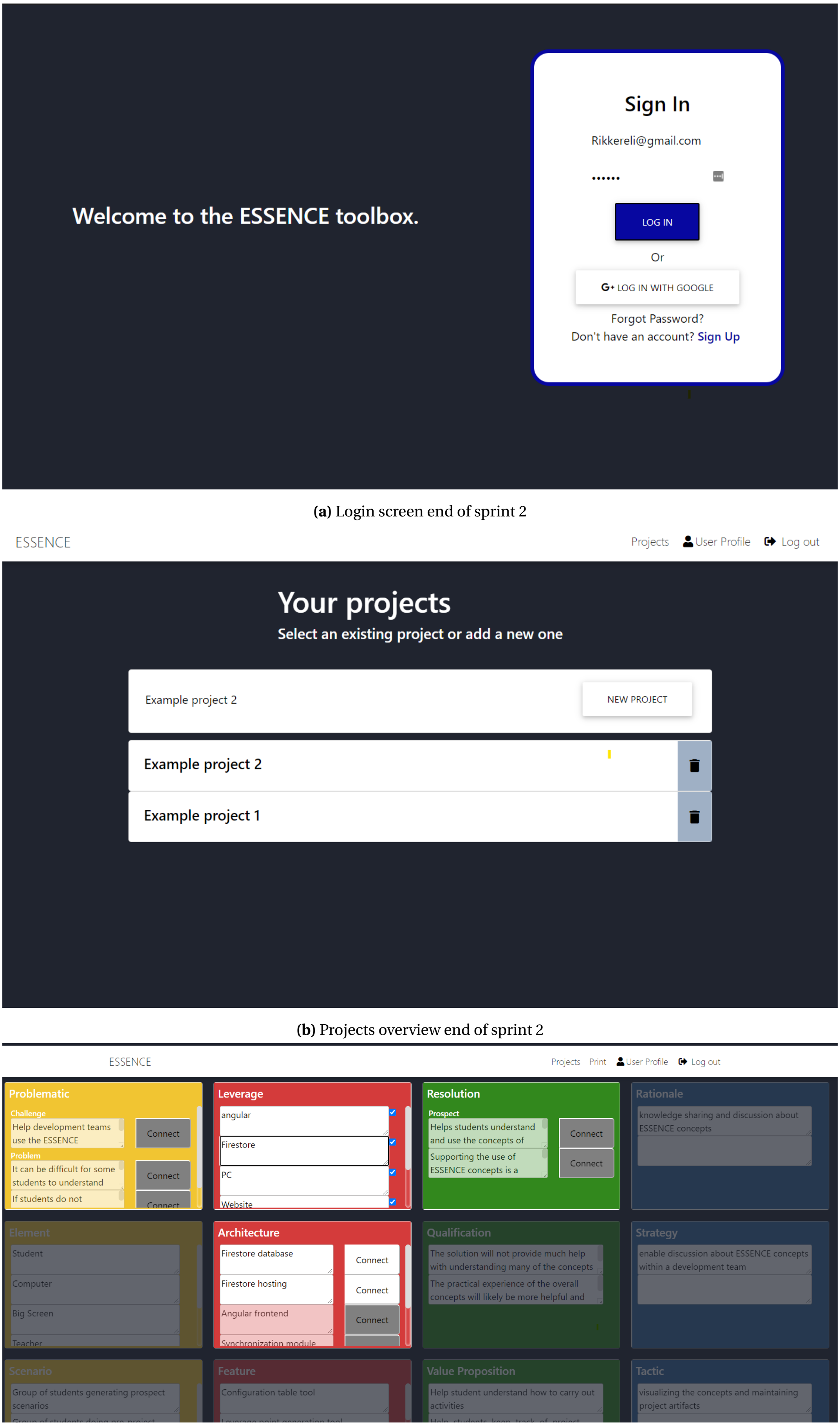
What do I expect my solution will be to the problem?
I expect that the solution will be a tool that can assist developers in maintaining an ESSENCE development project.
What do I expect to gain from this activity?

Features - Configuration table tool

1. Maintain configuration tables for separate users
2. Maintain multiple configuration tables for each user
3. Process view column
4. Connect categories that are directly connected. Dim irrelevant categories when focusing on specific category
5. Connect category items across directly connected categories
6. Make users aware when an item is not properly backed
7. Highlight relevant items when focusing on specific item
8. Give category items status <active|inactive>. Keep track of item status (e.g., italic when inactive)
9. "print" functionality for the configuration table for easy inclusion in report
10. Change ordering of category items
11. Connect each row (tactics, rationale and strategy) by border in order to give visual connection
12. Sub-items
13. Item limit
14. Information to inspire filling out the categories
15. Visibility connected to the frequency with which a category is typically changed
16. Enable the user to add a description about each item

Figure 5.6: Configuration table tool sub-features

5.2. Sprint 2



Welcome to the ESSENCE toolbox.

Sign In

Rikkereli@gmail.com

••••••

Forgot Password?

Don't have an account? Sign Up

(a) Login screen end of sprint 2

Your projects

Select an existing project or add a new one

Example project 2

Example project 2

Example project 1

(b) Projects overview end of sprint 2

Problematic

Challenge

Help development teams use the ESSENCE

Problem

It can be difficult for some students to understand

If students do not

Leverage

angular ☑

Firestore ☑

PC ☑

Website ☑

Resolution

Prospect

Helps students understand and use the concepts of

Supporting the use of ESSENCE concepts is a

Rationale

knowledge sharing and discussion about ESSENCE concepts

Element

Student

Computer

Big Screen

Teacher

Architecture

Firestore database

Firestore hosting

Angular frontend

Synchronization module

Qualification

The solution will not provide much help with understanding many of the concepts

The practical experience of the overall concepts will likely be more helpful and

Strategy

enable discussion about ESSENCE concepts within a development team

Scenario

Group of students generating prospect scenarios

Group of students doing pre-project

Feature

Configuration table tool

Leverage point generation tool

Value Proposition

Help student understand how to carry out activities

Help students keep track of project

Tactic

visualizing the concepts and maintaining project artifacts

(c) Diagrams overview in the end of sprint 2

Figure 5.7: UI in end of sprint 2.

Until now, the focus has mainly been on the diagram tool, as this is a part of the project where the solution is obvious. However, the next steps will probably be more related to supporting the project, and ESSENCE, as a whole, with more learning strategies and where more thought needs to be put on all elements. I expect to get a more concrete vision for that part of the project during this RST Review, as the vision right now is relatively fuzzy.

How useful do I expect this activity to be?

I expect the RST Review to be much more useful this time.

5.2.5 RST Review

The input is:

- Product: The website at sprint 2 end
- Configuration table 5.5.
- Checklist from RST Review in sprint 1

Part 1 of the RST Review is to show the product to stakeholders and note any feedback they provide. At this point, the only external stakeholder is still Ivan Aaen. I again used part 2 to evaluate the diagram tool. The comments from Aaen and my thoughts are noted in A.6.

5.2.6 RST Review Evaluation

How did ESSENCE affect the course of action?

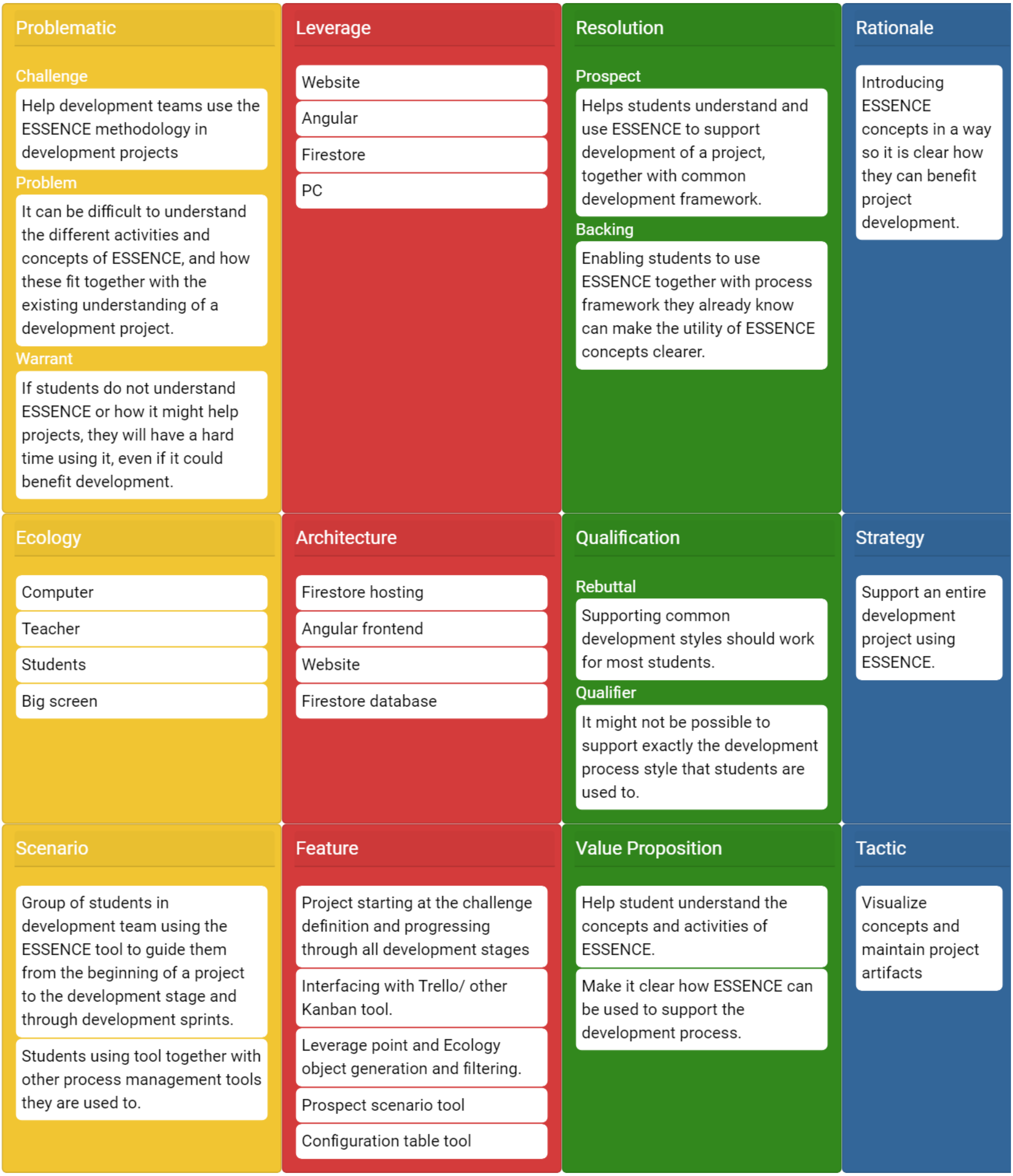
Before this activity, the main focus on the project was to introduce the concepts and support students using them in exercise sessions. Now, the focus is more on letting students use the tool to guide the development of a project framework and project management tools the are used to. One of the reasons I want to go in this direction is, that the activities and concepts in ESSENCE seem strange before use. Therefore I imagine that the best way to introduce them is to make students use them in a real project instead of a short example.

Was the activity overall helpful for the project?

Yes, it was very helpful.

Which benefits did ESSENCE have in this activity? Would they still be benefits in a multi-person team?

Even though the vision for the project is mostly the same as before the activity, I did do a small pivot in the project problem, where the solutions for the old and the new problem are likely very similar, but the focus for the new problem is more defined. If I had not carried out this activity, the project would likely have progressed with



Problematic

Challenge

Help development teams use the ESSENCE methodology in development projects

Problem

It can be difficult to understand the different activities and concepts of ESSENCE, and how these fit together with the existing understanding of a development project.

Warrant

If students do not understand ESSENCE or how it might help projects, they will have a hard time using it, even if it could benefit development.

Leverage

- Website
- Angular
- Firestore
- PC

Resolution

Prospect

Helps students understand and use ESSENCE to support development of a project, together with common development framework.

Backing

Enabling students to use ESSENCE together with process framework they already know can make the utility of ESSENCE concepts clearer.

Rationale

Introducing ESSENCE concepts in a way so it is clear how they can benefit project development.

Ecology

- Computer
- Teacher
- Students
- Big screen

Architecture

- Firestore hosting
- Angular frontend
- Website
- Firestore database

Qualification

Rebuttal

Supporting common development styles should work for most students.

Qualifier

It might not be possible to support exactly the development process style that students are used to.

Strategy

Support an entire development project using ESSENCE.

Scenario

- Group of students in development team using the ESSENCE tool to guide them from the beginning of a project to the development stage and through development sprints.
- Students using tool together with other process management tools they are used to.

Feature

- Project starting at the challenge definition and progressing through all development stages
- Interfacing with Trello/ other Kanban tool.
- Leverage point and Ecology object generation and filtering.
- Prospect scenario tool
- Configuration table tool

Value Proposition

- Help student understand the concepts and activities of ESSENCE.
- Make it clear how ESSENCE can be used to support the development process.

Tactic

Visualize concepts and maintain project artifacts

Figure 5.8: Sprint 2 updated configuration table.

the former problem.

Which drawbacks did ESSENCE have in this activity? Would they still be drawbacks in a multi-person team?

I find it difficult to carry out the activity, as it can be difficult both to decide the level of granularity for category items, what the ideas for some individual categories are and how different categories fit together. Also, it is very difficult to get started as there is no real starting point and no suggestions to begin the activity. The biggest drawback is in my opinion the criteria, as they are created based on the former understanding of the project, but are not updated as the understanding of the project is updated. When the criteria are updated during the review, they stop making sense. It is not really clear what the point of them is. The distinction between the categories could likely be clearer in a multi-person team, as vocalizing and discussing makes concepts more understandable.

Can I imagine some modifications to ESSENCE that could improve it?

I think that ESSENCE could benefit from clear instructions on how to update the configuration table, and how to use the criteria. It would also be nice with some simple questions for each category and item that could get the thought process going when the development team runs out of ideas.

How was the ESSENCE roles and views utilized in this activity?

I did not give either the views or roles any thoughts during this activity.

Other comments

5.3 Sprint 3

5.3.1 Scope

In the two first sprints, the scope has been the diagram tool. In this sprint, I will focus more on the overall project life-cycle, as the diagram tool now contains the overall functionality. Therefore, the scope of sprint 3 is based on the feature: *Project starting at the challenge definition and progressing through all development stages.*

5.3.2 Product Build

During sprint 3 I implemented a simple UI for each activity in order to get the basic functionality working. The flow is displayed in figure5.9, and contains the following UI pages:

1. Challenge definition
2. Ecology object generation and filtering
3. Leverage point generation and filtering

4. Initial problem and warrant definition
5. Prospect scenario axis definition
6. Prospect scenario representation definition
7. Prospect scenario representation expansion
8. Sprint initiation - feature selection
9. Sprint in progress - Kanban board
10. Stakeholder comments
11. Criteria definition for RST review
12. Configuration table update

Each of these pages is very simple implementations without any real assistance for understanding ESSENCE besides the gradual introduction of concepts. Besides pages for the basic ESSENCE activities, I have introduced pages for workflow management during sprints. Additionally, I implemented a print/convert to PDF function and did a lot of refactoring in order to make the product more reliable.

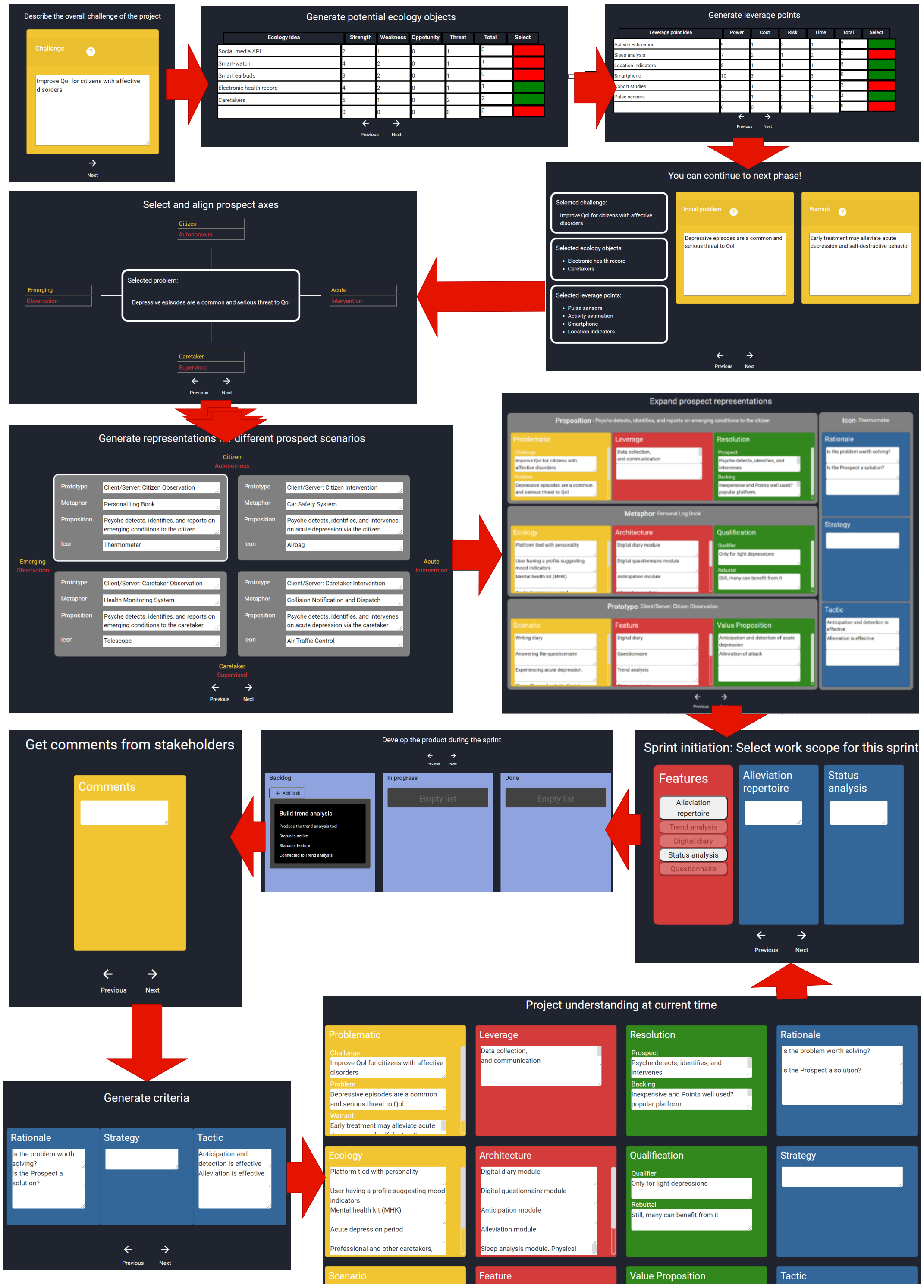
With the basic structure in place, the next sprints will focus on implementing support for specific activities.

5.3.3 RST Review expectations

What do I expect my solution will be to the problem?
I expect the solution to be a tool that slowly introduces students to ESSENCE concepts while working on a project.
What do I expect to gain from this activity?
At this point, I think some reflection on the prospect is needed. At this point, it includes working with a common development framework, but I expect that there is a need to specify that they will be using the Scrum development framework to be more specific, as "common development framework" is too broad for me to implement realistically.
How useful do I expect this activity to be?
I expect the review to be useful this time, as I think the prospect at this point should be modified at least slightly.

5.3.4 RST Review

The input is:



Describe the overall challenge of the project

Challenge ?

Improve Qol for citizens with affective disorders

Generate potential ecology objects

Ecology idea	Strength	Weakness	Oppotunity	Threat	Total	Select
Social media API	2	1	0	1	0	🟥
Smart-watch	4	2	0	1	1	🟥
Smart earbuds	3	2	0	1	0	🟥
Electronic health record	4	2	0	1	1	🟩
Caretakers	5	1	0	2	2	🟩
	0	0	0	0	0	🟥

Generate leverage points

Leverage point idea	Power	Cost	Risk	Time	Total	Select
Activity estimation	9	1	2	1	5	🟩
Sleep analysis	7	2	1	2	2	🟥
Location indicators	8	1	1	1	5	🟩
Smartphone	10	3	4	3	0	🟩
Cohort studies	8	1	3	2	2	🟥
Pulse sensors	7	1	2	1	3	🟩
	0	0	0	0	0	🟥

You can continue to next phase!

Selected challenge:
Improve Qol for citizens with affective disorders

Selected ecology objects:
- Electronic health record
- Caretakers

Selected leverage points:
- Pulse sensors
- Activity estimation
- Smartphone
- Location indicators

Initial problem ?
Depressive episodes are a common and serious threat to Qol

Warrant ?
Early treatment may alleviate acute depression and self-destructive behavior

Select and align prospect axes

Citizen
Autonomous

Emerging
Observation

Selected problem:
Depressive episodes are a common and serious threat to Qol

Acute
Intervention

Caretaker
Supervised

Generate representations for different prospect scenarios

Citizen
Autonomous

Emerging
Observation

Acute
Intervention

	Citizen Observation	Citizen Intervention
Prototype	Client/Server: Citizen Observation	Client/Server: Citizen Intervention
Metaphor	Personal Log Book	Car Safety System
Proposition	Psyche detects, identifies, and reports on emerging conditions to the citizen	Psyche detects, identifies, and intervenes on acute depression via the citizen
Icon	Thermometer	Airbag

	Caretaker Observation	Caretaker Intervention
Prototype	Client/Server: Caretaker Observation	Client/Server: Caretaker Intervention
Metaphor	Health Monitoring System	Collision Notification and Dispatch
Proposition	Psyche detects, identifies, and reports on emerging conditions to the caretaker	Psyche detects, identifies, and intervenes on acute depression via the caretaker
Icon	Telescope	Air Traffic Control

Caretaker
Supervised

Expand prospect representations

Proposition: Psyche detects, identifies, and reports on emerging conditions to the citizen

Problematic
Challenge
Improve Qol for citizens with affective disorders
Problem
Depressive episodes are a common and serious threat to Qol

Leverage
Data collection, and communication

Resolution
Prospect
Psyche detects, identifies, and intervenes
Backing
Inexpensive and Points well used? popular platform.

Icon: Thermometer

Rationale
Is the problem worth solving?
Is the Prospect a solution?

Metaphor: Personal Log Book

Ecology
Platform tied with personality
User having a profile suggesting mood indicators
Mental health kit (MHK)

Architecture
Digital diary module
Digital questionnaire module
Anticipation module

Qualification
Qualifier
Only for light depressions
Rebuttal
Still, many can benefit from it

Strategy

Prototype: Client/Server: Citizen Observation

Scenario
Writing diary
Answering the questionnaire
Experiencing acute depression.

Feature
Digital diary
Questionnaire
Trend analysis

Value Proposition
Anticipation and detection of acute depression
Alleviation of attack

Tactic
Anticipation and detection is effective
Alleviation is effective

Sprint initiation: Select work scope for this sprint

Features
Alleviation repertoire
Trend analysis
Digital diary
Status analysis
Questionnaire

Alleviation repertoire

Status analysis

Develop the product during the sprint

Backlog	In progress	Done
+ Add Task	Empty list	Empty list
Build trend analysis Produce the trend analysis tool Status is active Status is feature Connected to Trend analysis		

Get comments from stakeholders

Comments

Generate criteria

Rationale
Is the problem worth solving?
Is the Prospect a solution?

Strategy

Tactic
Anticipation and detection is effective
Alleviation is effective

Project understanding at current time

Problematic
Challenge
Improve Qol for citizens with affective disorders
Problem
Depressive episodes are a common and serious threat to Qol
Warrant
Early treatment may alleviate acute

Leverage
Data collection,
and communication

Resolution
Prospect
Psyche detects, identifies, and intervenes
Backing
Inexpensive and Points well used? popular platform.

Rationale
Is the problem worth solving?
Is the Prospect a solution?

Ecology
Platform tied with personality
User having a profile suggesting mood indicators
Mental health kit (MHK)
Acute depression period
Professional and other caretakers,

Architecture
Digital diary module
Digital questionnaire module
Anticipation module
Alleviation module
Sleep analysis module. Physical

Qualification
Qualifier
Only for light depressions
Rebuttal
Still, many can benefit from it

Strategy

Scenario

Feature

Value Proposition

Tactic

Figure 5.9: The UI flow in the end of sprint 3

- Product: Website version shown in figure 5.9
- Configuration table: Configuration table from sprint 2 end, shown in 5.8
- Checklist: For the checklist, I am using the process view column in the configuration table.

Part 1 of the RST Review is to show the product to stakeholders and note any feedback they provide. At this point, the only external stakeholder is still Ivan Aaen. I again used part 2 to evaluate the diagram tool. The comments from Aaen and my thoughts are noted in A.8. The updated configuration table is shown in 5.10

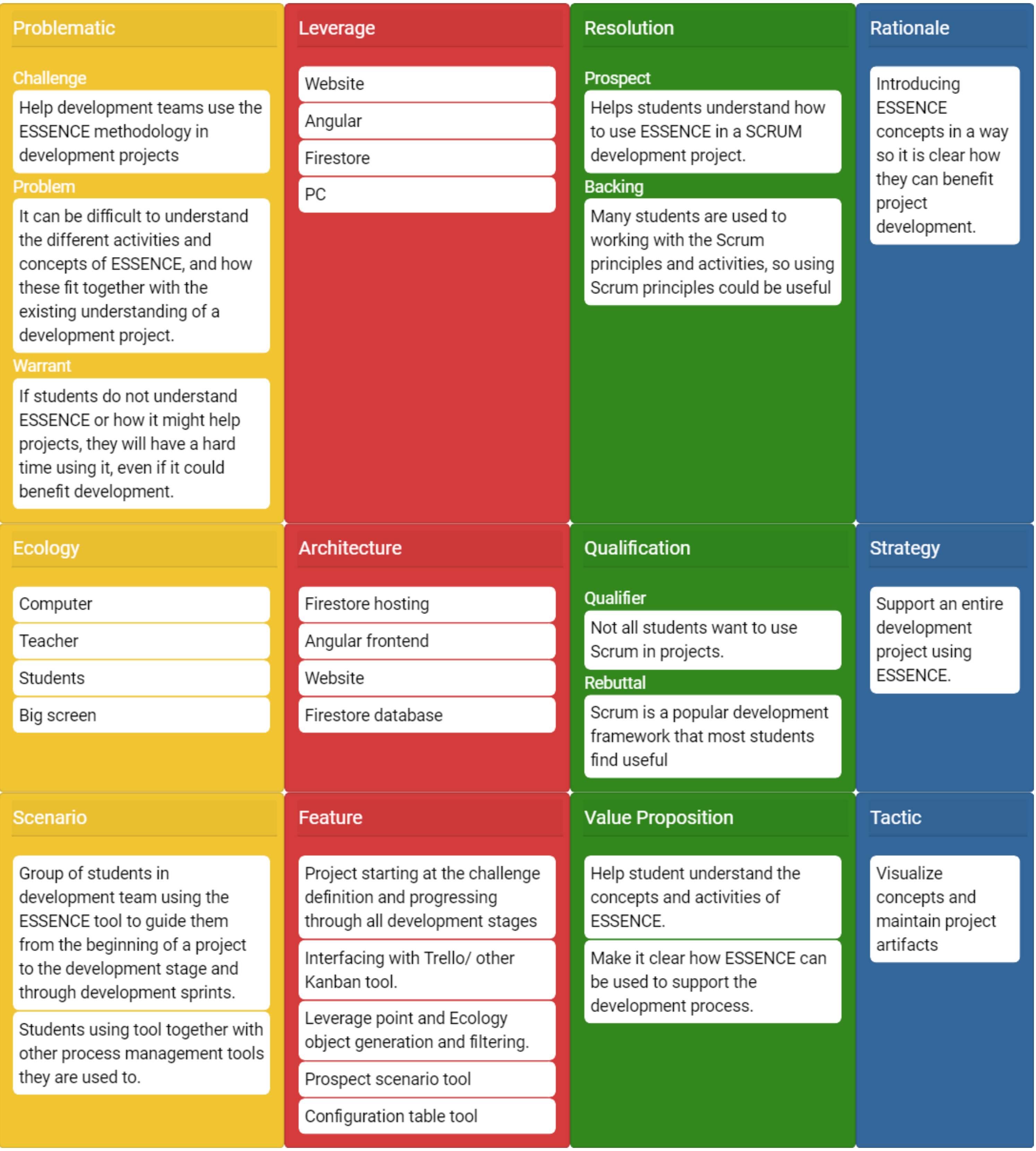
5.3.5 RST Review Evaluation

How did ESSENCE affect the course of action?
ESSENCE did not significantly change the course of action. However, it was useful because it made me think critically about my view on the project, and slightly change the course of action.

Was the activity overall helpful for the project?
Yes, it ensured that I noticed my updated understanding and therefore changed the prospect.

Which benefits did ESSENCE have in this activity? Would they still be benefits in a multi-person team?
Filling out the categories ensures that each aspect of the project understanding is considered and reflected upon. In my opinion, the greatest advantage of this is, that no major aspect of the project understanding is taken for granted, and the process of defining brings a lot of nuance to the project.

Which drawbacks did ESSENCE have in this activity? Would they still be drawbacks in a multi-person team?
The biggest drawback of the RST Review is in my opinion the lack of any real procedures on where they begin and how to update the categories. I think the idea is to use the criteria for questioning the categories, but the definition of criteria is very fuzzy, and it is still not clear how to carry out the diagram update. As there are many categories, and multiple items in each category, it is very difficult to understand where to begin, and in which order each category and each item should be evaluated. Also, even after having evaluated an item, it is not clear how to proceed. Another related issue is, that the utility of many categories still does not appear very intuitive to me, especially how they are connected to each other. Also, having all categories in a table in the same text format does not appear to be a good solution as they are conceptually very different from each other. In my opinion, the greatest drawback of ESSENCE at this point is the purpose and application of the criteria is very fuzzy. This makes it difficult to carry out the review.



Problematic

Challenge
- Help development teams use the ESSENCE methodology in development projects

Problem
- It can be difficult to understand the different activities and concepts of ESSENCE, and how these fit together with the existing understanding of a development project.

Warrant
- If students do not understand ESSENCE or how it might help projects, they will have a hard time using it, even if it could benefit development.

Leverage

- Website
- Angular
- Firestore
- PC

Resolution

Prospect
- Helps students understand how to use ESSENCE in a SCRUM development project.

Backing
- Many students are used to working with the Scrum principles and activities, so using Scrum principles could be useful

Rationale

- Introducing ESSENCE concepts in a way so it is clear how they can benefit project development.

Ecology

- Computer
- Teacher
- Students
- Big screen

Architecture

- Firestore hosting
- Angular frontend
- Website
- Firestore database

Qualification

Qualifier
- Not all students want to use Scrum in projects.

Rebuttal
- Scrum is a popular development framework that most students find useful

Strategy

- Support an entire development project using ESSENCE.

Scenario

- Group of students in development team using the ESSENCE tool to guide them from the beginning of a project to the development stage and through development sprints.
- Students using tool together with other process management tools they are used to.

Feature

- Project starting at the challenge definition and progressing through all development stages
- Interfacing with Trello/ other Kanban tool.
- Leverage point and Ecology object generation and filtering.
- Prospect scenario tool
- Configuration table tool

Value Proposition

- Help student understand the concepts and activities of ESSENCE.
- Make it clear how ESSENCE can be used to support the development process.

Tactic

- Visualize concepts and maintain project artifacts

Figure 5.10: Sprint 3 updated configuration table.

Can I imagine some modifications to ESSENCE that could improve it?
I think that there is a great need for:

- More instructions, or suggestions, for how to carry out the diagram update activity. It does not have to be specific rules that have to be followed, but simple guidelines that could be used when the team gets stuck would go a long way.
- Some alternative to the diagram format that makes conceptual sense could improve the experience.
- Reconsidering the criteria, either removing these or adding additional explanation about what the idea is and how they should be used, as the current format is, in my opinion, too recursive.

E.g., the criteria are developed based on the current understanding, and reasoning, of the project. As the understanding of the project often changes during a review, it does not seem obvious that specific criteria based on the former understanding of the project are useful when updating the categories. For me, these seem to be more in the way of the project understanding naturally unfolding, than helpful. Instead, it could be useful with a general list of questions that could be asked at every review like *does the backing still hold?*.
How was the ESSENCE roles and views utilized in this activity?
At this point, I am very skeptical about the ESSENCE roles, as I don't understand the use of them, and I think that they add more confusion and complexity to the ESSENCE methodology without really contributing anything useful. In my opinion, it is a very constraining way of viewing a development team, as each person is determined to be only one thing. It should be up to each individual development team to decide what the roles in the team should be. I do think that the concept of views is useful, as this conceptually divide the project into areas.
Other comments

Chapter 6

Late development

6.1 Sprint 4

6.1.1 Sprint scope

During sprint 3 I developed an overall flow of the tool. In this sprint, I will try to reflect on how to illustrate ESSENCE concepts, and how ESSENCE could be modified to improve the intuitiveness.

6.1.2 Design of part 1

Up until now the product design have been straight forward implementation of my vision for the ESSENCE tool. I now have a solution that cover the basic activities of ESSENCE, and the next step will be to improve the learning experience, specifically the ease of leaning, for students using the tool. In my opinion, one major obstacle for learning ESSENCE is, that it is not always intuitive what the ideas behind the concepts are, and how the concept should be used. The ESSENCE tool should help conceptually explaining both the idea behind and the purpose of each concept. In this sprint I will therefore mainly aim make design designs for the ESSENCE tool without actually implementing anything.

In order for the design to be successful, the design should support some type of learning for each of the following concepts:

1. Views
2. Pre-project
   (a) Challenge definition
   (b) Ecology object activities
   (c) Leverage point activities
   (d) Initial problem

   (e) Prospect scenario activities
      i. Axis generation
      ii. Prospect representation generation
      iii. Prospect representation expansion

3. Categories
   (a) Challenge
   (b) Problem
   (c) Backing
   (d) Leverage
   (e) Prospect
   (f) Warrant
   (g) Ecology
   (h) Architecture
   (i) Qualifier
   (j) Rebuttal
   (k) Scenario
   (l) Feature
   (m) Value proposition
4. Criteria
   (a) Strategy
   (b) Tactics
   (c) Rationale
5. RST Reviews

For each concept and activity, the design should make it clear:

1. How it is related to other concepts and activities.
2. What it is.
3. How should it be used.
4. Why it is important.

One big issue I have with the current design of the ESSENCE tool is, that all categories are in text format, and that they are not introduced in any other way. This provide almost no visual aid for understanding the idea behind the category. At this point, the structure of the ESSENCE tool is already designed to help students understand a structure that can support ESSENCE. However, the structure is still not clear.

This sprint will be divided into three parts:

- Part 1: Gather material about learning.
- Part 2: Make design suggestions about the ESSENCE tool
- Part 3: Evaluate the design.

6.1.3 Design research

First of all, it is relevant to chose the overall structure of the tool. [11] describes how software tools can be used to support the use of methodologies in software projects. Each methodology has a set of rules or guidelines. If any of these are violated, the tool should either:

1. Restrict workflow until the violation is corrected
2. Give informative feedback that states the violation

The paper suggest three different design philosophies related to these violation responses:

1. Restrictive
2. Guided
3. Flexible

The restrictive philosophy is designed to force the developer to use the methodology correctly, and do the correct steps in the correct order. The guided approach offer more freedom and only suggests, instead of enforces, what to do at different stages. In a flexible tool, the user has complete freedom on how to use the tool.

Each philosophy has it's own benefits and drawbacks. As the tool gets more restrictive, it gets easier to get users to use the methodology the correct way. However, people differ in creative processes, and a too restrictive approach will prevent development teams from using the tool to support the development process that they prefer.

In itself, ESSENCE is mend to be a very flexible, lightweight, methodology where any definitive rules are avoided and the concepts and activities are more thought as a set of helpful guidelines. In my opinion, it makes sense that when a development team understands ESSENCE, they can take the concepts and activities and use them however they would like.

However, a tool has to be mastered before it can be wielded freely, and for now, the biggest obstacle for ESSENCE is the steep learning curve. As the primary goal for the product developed in this project is to mitigate this steep learning curve, I imagine that a hybrid between the restrictive and guided philosophy would prove most useful.

In my design, I will aim to use the restrictive philosophy whenever I am introducing a new concept and activity, and then use the guided approach when the students have been introduced. Ideally, I would introduce a learning mode and a developer mode to the tool later, and design the developer mode based on the flexible philosophy. However, it is very unlikely that I will have time for this in this project.

In order to get some inspiration for how to introduce the ESSENCE concepts in a manner that support learning, I did some research about teaching techniques, and found a general list of learning strategies that I will use as inspiration[10]:

1. Present material in different styles, both visual and verbal.

2. Give both concrete examples and abstract explanations for introduced concepts, and discuss the connection between characteristics of the examples and key elements of the abstract representations.

3. Use repeated exposure and practice over time.

4. When students are introduced to concepts, make them work on problems where they use the concepts. As they get better at using the concepts, make them spend less time studying examples and more time using the concepts to solve new problems.

5. Use quizzes and exams to learn.

For technique 1 I have until now mainly presented concepts verbally with long pieces of text. In the new design I will aim at presenting more visibly. For technique 2 I did include both concrete examples and abstract explanations for some concepts, but I did not discuss the connection between the examples and key elements for the abstract representations enough. Technique 3 is naturally included in the tool, as concepts are used over the span of a project. Technique 4 is also naturally a part of the design, as students are using the concepts to fill out the information in the tool. Technique 5 is not a part of the design at this point, but quizzes could be included in the learning part of the design when new concepts are introduced. This gives me three areas of concern when designing the tool:

1. Introduce visual aid to understand the concepts

2. Improve the verbal description of the concepts and activities

3. Include quizzes after introducing new concepts

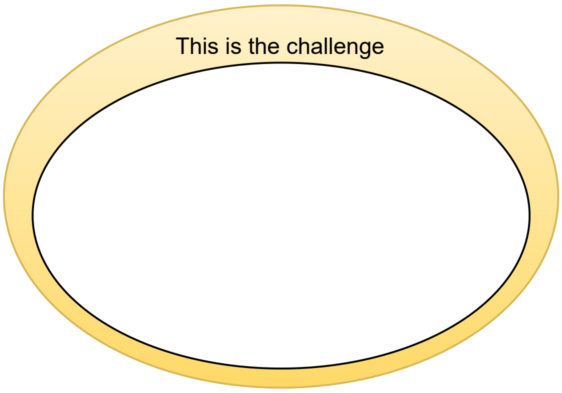


Figure 6.1: Conceptual representation of the challenge.

The verbal descriptions are very important for the finished product, but should be written in order to support the visual design and the flow of the solution. Therefore I will not focus on the verbal design at this point. The quizzes can be useful to support the learning, but formulating useful questions is not what I am educated for. Therefore I will not include quizzes at this point in the product. For the rest of this sprint I will focus on modifying the design so that it explain the concepts visually. ¨

6.1.4 Design choices

In this section I will experiment with how to represent each concept visually. I will begin with each category.

Challenge: Purpose of using category: Get a clear vision of the reason behind the project. Answer questions:

- Where is the project going to fit into the real world?
- What is the reasoning behind the project?

In ESSENCE, the challenge is described as a *The broader domain in which the problem resides*. Therefore, a visual representation of the challenge should be able to contain at least the problem. Additionally, ecology objects should also be able to be represented being within the challenge domain.

As challenge is within the paradigm view, it should be yellow. The container could in theory be of any shape. However, the shape that I find most intuitive is an ellipse, as this represents a more dynamic and fluent form compared to the restricted form of a square and circle. An illustration of how the challenge could be displayed is illustrated in 6.1

Problem: Purpose of category:

The problem resides within a challenge and does have a span of its own. However, this span is more restricted and controlled, as the problem is a solvable subset of the challenge. Therefore I think the most intuitive way to illustrate the problem, and the

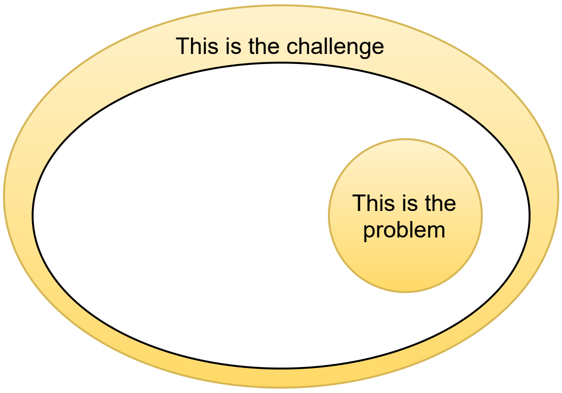


Figure 6.2: Conceptual representation of the problem.

relationship between the problem and the challenge is with a cirkle within the challenge. This is illustrated in 6.2.

Warrant: The warrant is related to the problem in that it is something that justifies why the problem should be solved. The warrant is less straight forward when trying to visually represent it, but one way to do it could be as the foundation of the problem. The warrant and the suggested representation is illustrated in 6.3.

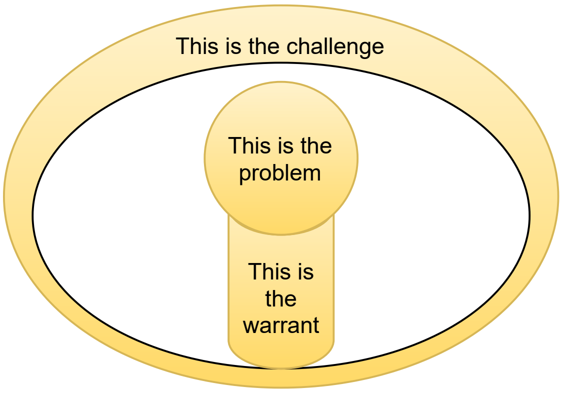


Figure 6.3: Conceptual representation of the warrant.

Ecology: Ecology objects can also easily be visualized as something within the challenge domain. The tricky part is to make ecology objects distinguishable in relation to the problem, as the problem is more of an idea where the ecology objects are concrete objects within the domain. However, selecting a concrete visual representation of the objects that is similar to a real object could conceptually distinguish ecology objects enough from problem and warrant, that it want confuse students. I think the most challenging aspect of designing a visual representation for ecology objects is to find a proper visual representation of an artifact and a service type. Figure 6.4 shows a suggestion for ecology objects.

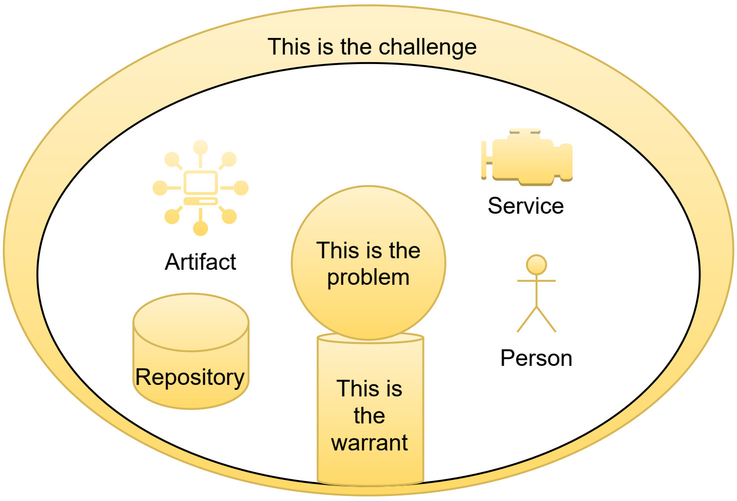


Figure 6.4: Conceptual representation of the ecology.

Leverage: Leverage is conceptually external to the challenge (although it would be possible in some circumstances to have leverage points within the challenge domain). In order to show the contrast between the ecology objects and the leverage points, it would make sense to introduce leverage points outside the challenge. A visual representation of leverage can be seen in 6.5

Scenario: Scenarios are situations where the solution should be used. A visual way to describe scenario could be as a user story. I think this would make much sense. An example of a scenario is displayed in 6.6. The scenarios category could then be a collection of these scenarios.

Prospect: The prospect can be very broad or very narrow, but the most intuitive understanding of the prospect I have been able to form is a short description that includes both the problem and the technical solution.

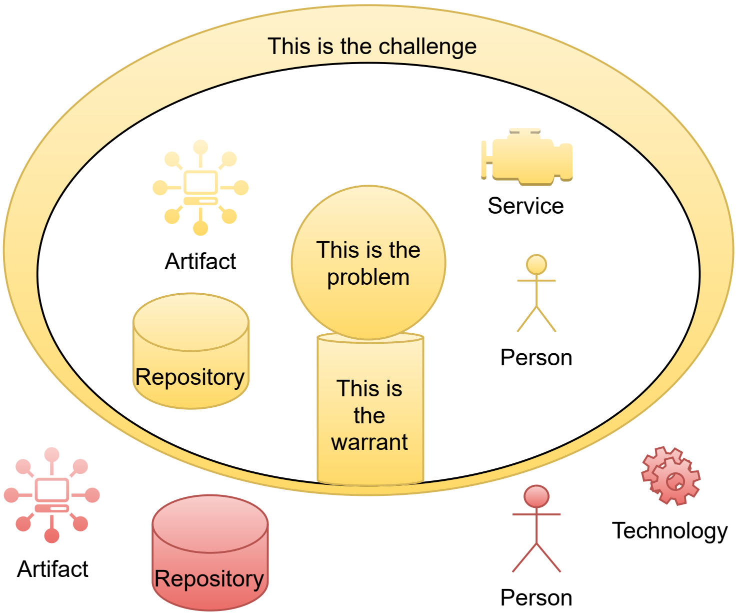


Figure 6.5: Conceptual representation of leverage points.

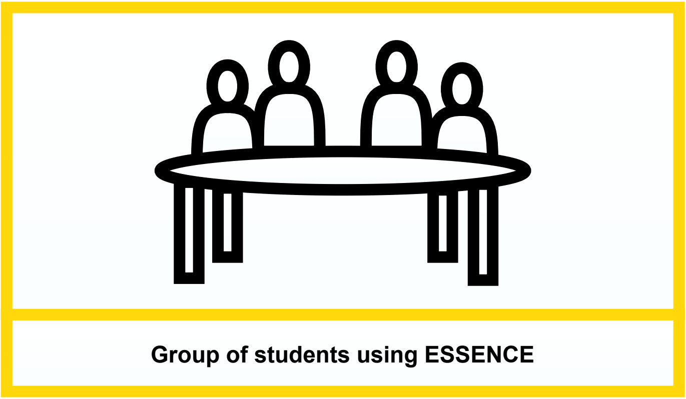


Figure 6.6: Conceptual representation of the scenarios.

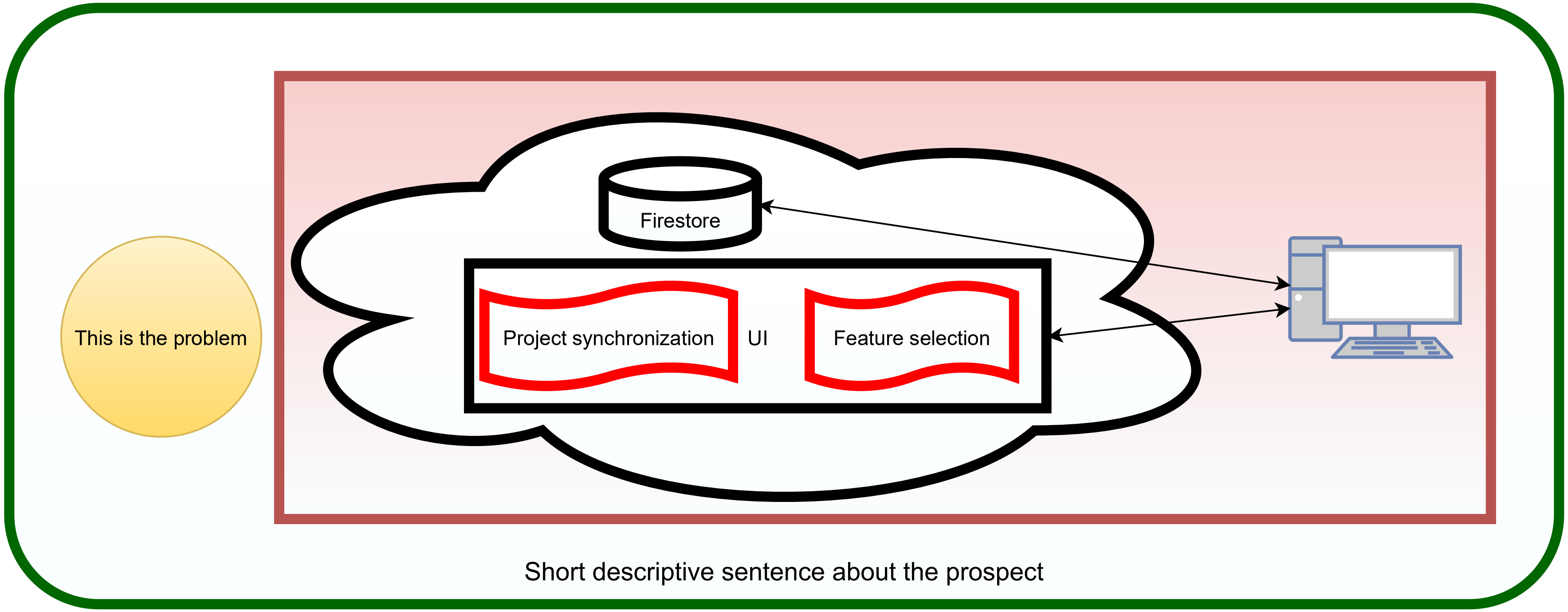


Figure 6.7: Conceptual representation of the prospect.

Backing: The backing is an explanation that works as the foundation of the prospect, and why the prospect is a desirable solution to the problem. Therefore the visual representation of the backing should be similar to that of the warrant only related to the prospect instead of the problem. The backing is represented in 6.8.

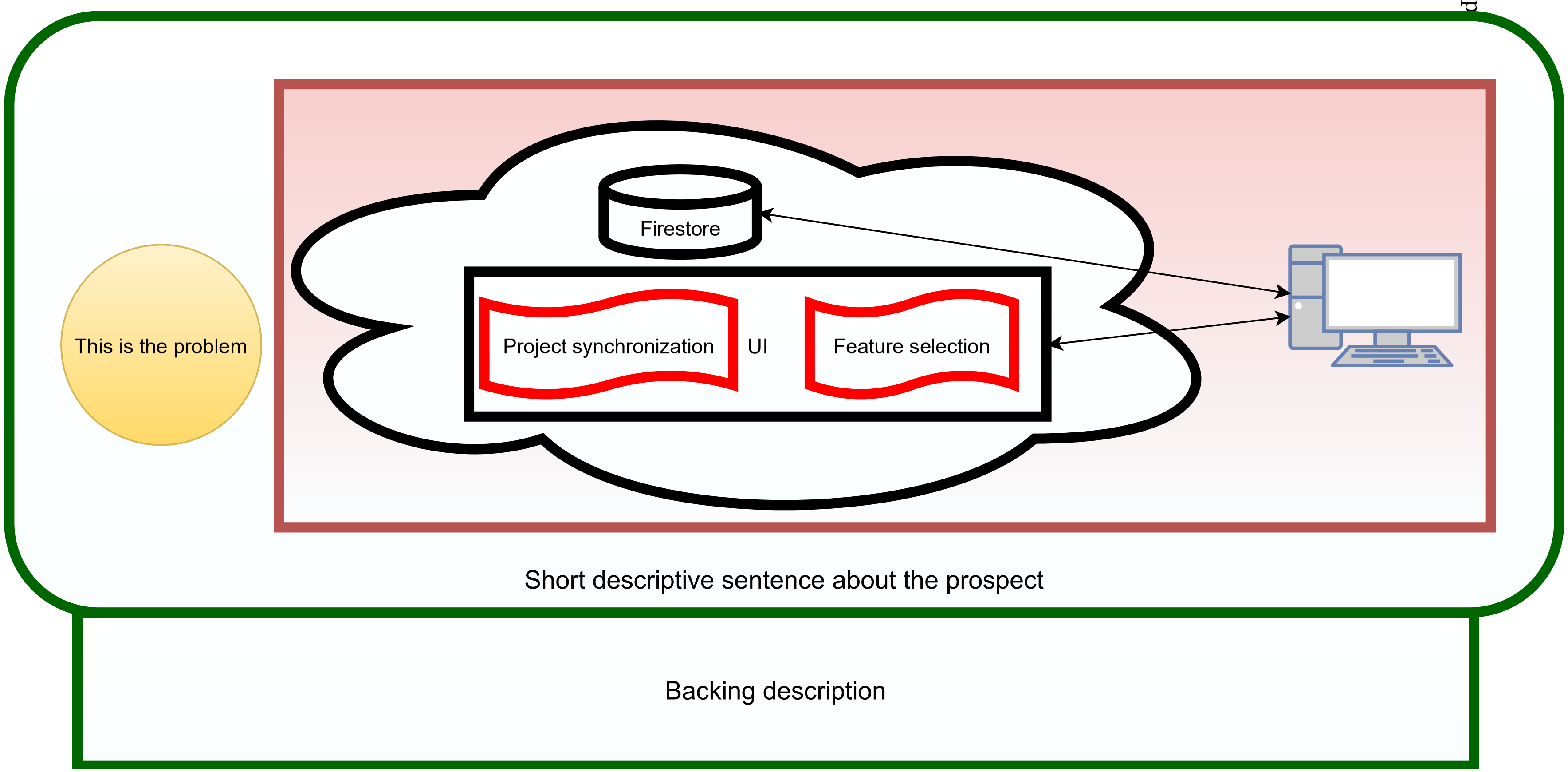


Figure 6.8: Conceptual representation of the prospect.

Architecture: The architecture is the conceptual structure of the product. This could both include the type of product that should be developed, and the internal structure of it. The best conceptual way I can think of for this is to actually picture the structure. This way, the architecture picturing will be different for each project. For this project, architecture could be pictured as in 6.9

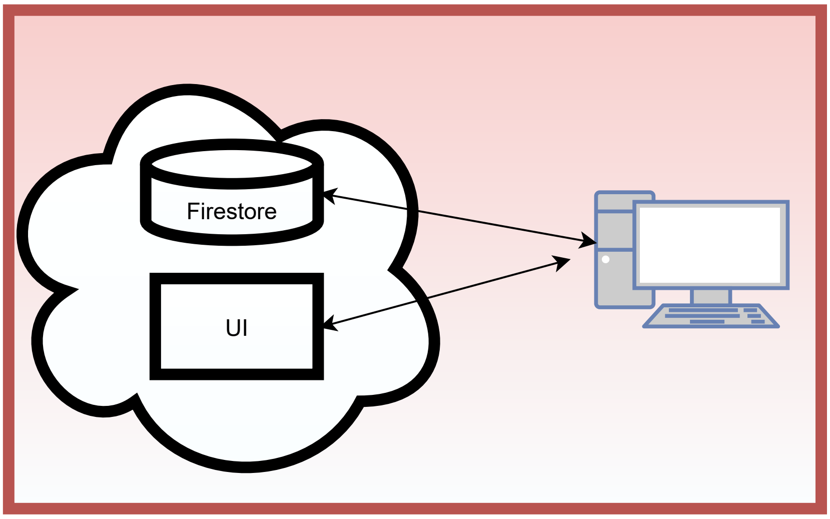


Figure 6.9: Conceptual representation of the architecture.

Feature: I cannot think of an intuitive way to visually display features, other than individual boxes with feature descriptions and names. The interesting way to visualize features would be to connect them to relevant scenarios and place them within modules in the architecture. A mock up of this is shown in 6.10

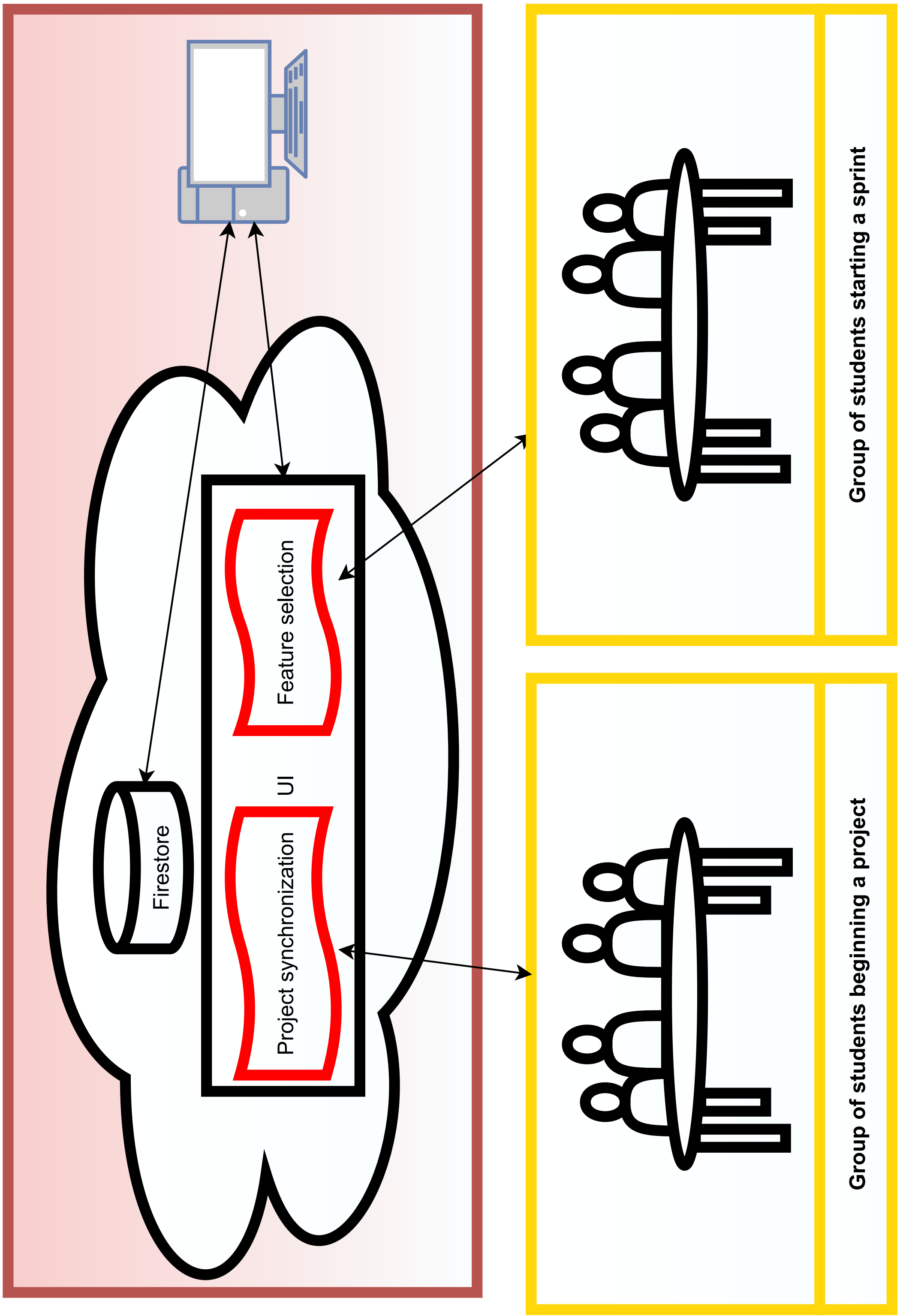

Firestore
UI
Feature selection
Project synchronization
Group of students starting a sprint
Group of students beginning a project

Figure 6.10: Conceptual representation of the feature.

Value proposition: The value proposition signifies how each feature contributes to solving the problem. This could be symbolized by a gift, connected to the feature.

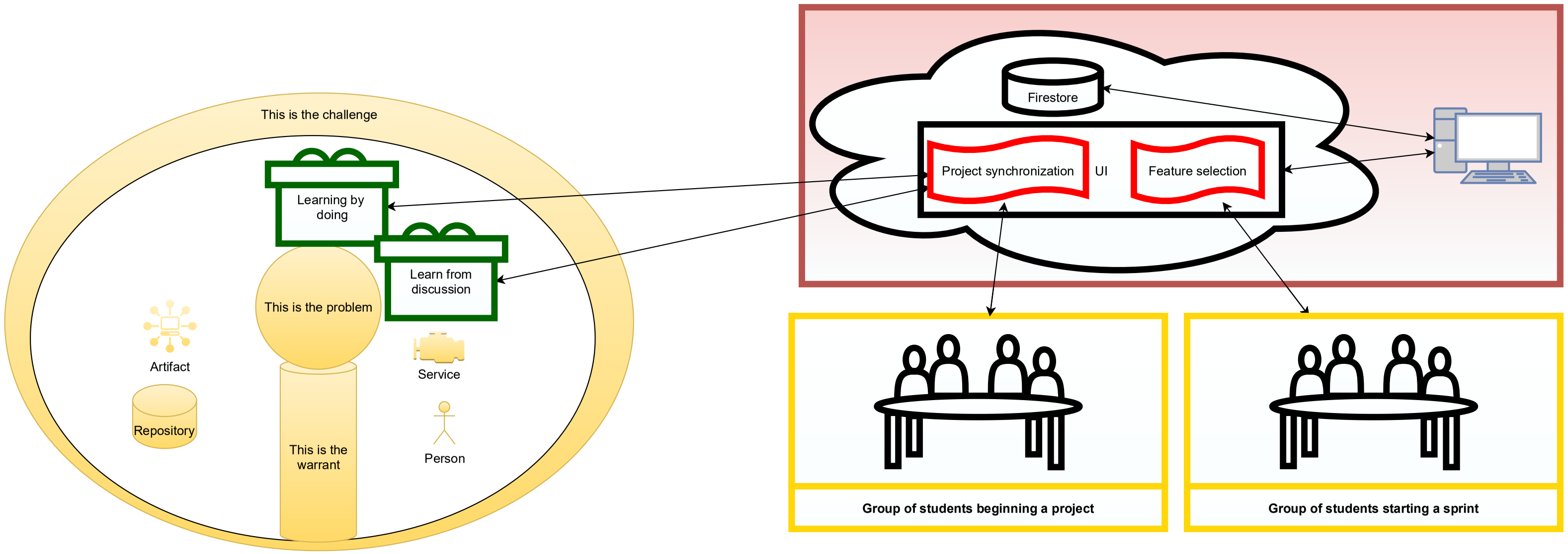


Figure 6.11: Conceptual representation of value propositions.

For the last two categories, **Qualifier**, and **Rebuttal**, I was not able to think of an intuitive visual representation. Therefore I omit this at this point.

6.1.5 Thoughts about ESSENCE

During the design activity, I got to see ESSENCE from a different perspective. First of all, I finally understood why Ivan kept insisting that ESSENCE should not be in a particular sequence, and I thought about different starting places for the project understanding. A project does not necessarily have to begin with the challenge as I previously thought, instead it could begin in potentially all other categories as well, providing different drawbacks but also different benefits.

During this analysis, I made some observations about the configuration table structure. In the following, I have added a list of suggestions for changes that could be done to the default configuration table. For each change I argue why I find this to be a good change:

The challenge category should be renamed to domain My problem with the name *challenge* for this category is that the associations I draw when hearing the word challenge is relatively similar to my associations for the word problem. In ESSENCE, problem and challenge are conceptually very different and have different utilities, but the similarity makes it difficult to really distinguish the purpose of the categories. Also, if the development team already identified the problem, it seems redundant to write down a broader challenge. Instead, it would in my opinion make more sense to change the challenge category to a domain category. The way it is described in the book is already as *the broader domain in which the problem resides*, so when considering that description, the word domain seems more related. Also, the domain might be an equally useful place to begin a project.

With the challenge category version, you could begin a project by defining the challenge *it is difficult to keep track of the household economy*. Changing the *challenge* category to *domain* would make an equivalent starting point named *household economy* possible. In my opinion, this is not just an equally good starting point, it also makes it more intuitive to define the ecology. E.g, for service ecology objects it would make at least as much sense to generate ecology objects related to *the household economy*, like online banking solutions as it does to generate service ecology objects for the challenge *it is difficult to keep track of household economy*. Therefore **I suggest the challenge category should be renamed domain**.

The problem category should be moved to the project view Another issue I identified is with the *problem* category. The problem category is currently placed within the *paradigm* view. However, the idea of the paradigm view should in my opinion be related to understanding the world related to the project, that is the *domain*, *ecology*, and *scenarios*. The problem is however a fundamental choice in the project, that should be made by stakeholders. Therefore, if the problem is not located in the *project* view, then I don't understand what the project view is actually useful for.

The Value propositions and warrant categories should be merged Value propositions and warrant are two separate categories. However, in practice they solve much of the same job - to understand how solving the problem will bring value. Therefore it would, in my opinion, be better to merge them together, as two categories with almost the same content confuse more than it adds value. I like the name *Value propositions*, as it works well, both from the feature perspective - *how would adding this feature bring value in relation to solving the problem*, and from the problem perspective - *how would solving the problem add value*.

The prospect and backing should be removed from the configuration table and applied outside of it The reason that I want to remove the prospect category is that in the current form it does not really add any information to the configuration table, but only works to add more clutter. I think that it would make much more sense as a description, or identity, of the configuration table, rather than a part of it. Then the backing can be used as a short summary of why the configuration is appropriate.

Qualification should be renamed to limitations The word *qualification* or *qualifier* does in my opinion not make sense in the context. In the book, it is described as the limitations to what the solution offers. However, the dictionary definition of the words does not seem related to limitations[2, 3]. Therefore I don't understand what the relation between the category name and the book description of the category is. Instead, it would make more sense to rename it *limitations*.

Merge the limitations category and the rebuttal I understand the reasoning behind the rebuttal category, as we should always be aware not only what the limitations are, but also why the solution is reasonable despite the limitation. However, having this in two different categories adds extra clutter to the table. As the limitations and rebuttals are tightly coupled, meaning that each limitation should have a rebuttal, simply adding them together to one category seems reasonable. E.g., in the psyche project in the ESSENCE book[1], we have the limitation (or qualifier) stating *Only for light depressions*, and the rebuttal stating *Still, many can benefit from it*, which could be rewritten to the limitation: *Only for light depression but still, many can benefit from it.* This solution contains the same amount of information as the current solution, but the configuration table would be less cluttered.

Add suggestions for how to use one category to generate a new one As previously mentioned, I have now understood that the configuration table should not always be filled in a specific order, but can more or less start in any category. However, it is not clear how you would use the information in one category to fill out the next. Therefore, I think it would be greatly beneficial to add some suggestions for how to get from one category to the next. I added a suggestion on how to do this in 6.12. The idea is that given information from the category on the horizontal axis, we try to fill out the category in the vertical axis using the suggestion in the common cell. If we have information about several categories, we can use the joined cell suggestion from one or more of the filled-out categories to fill out the new category. The blue cell suggests what the category should contain.

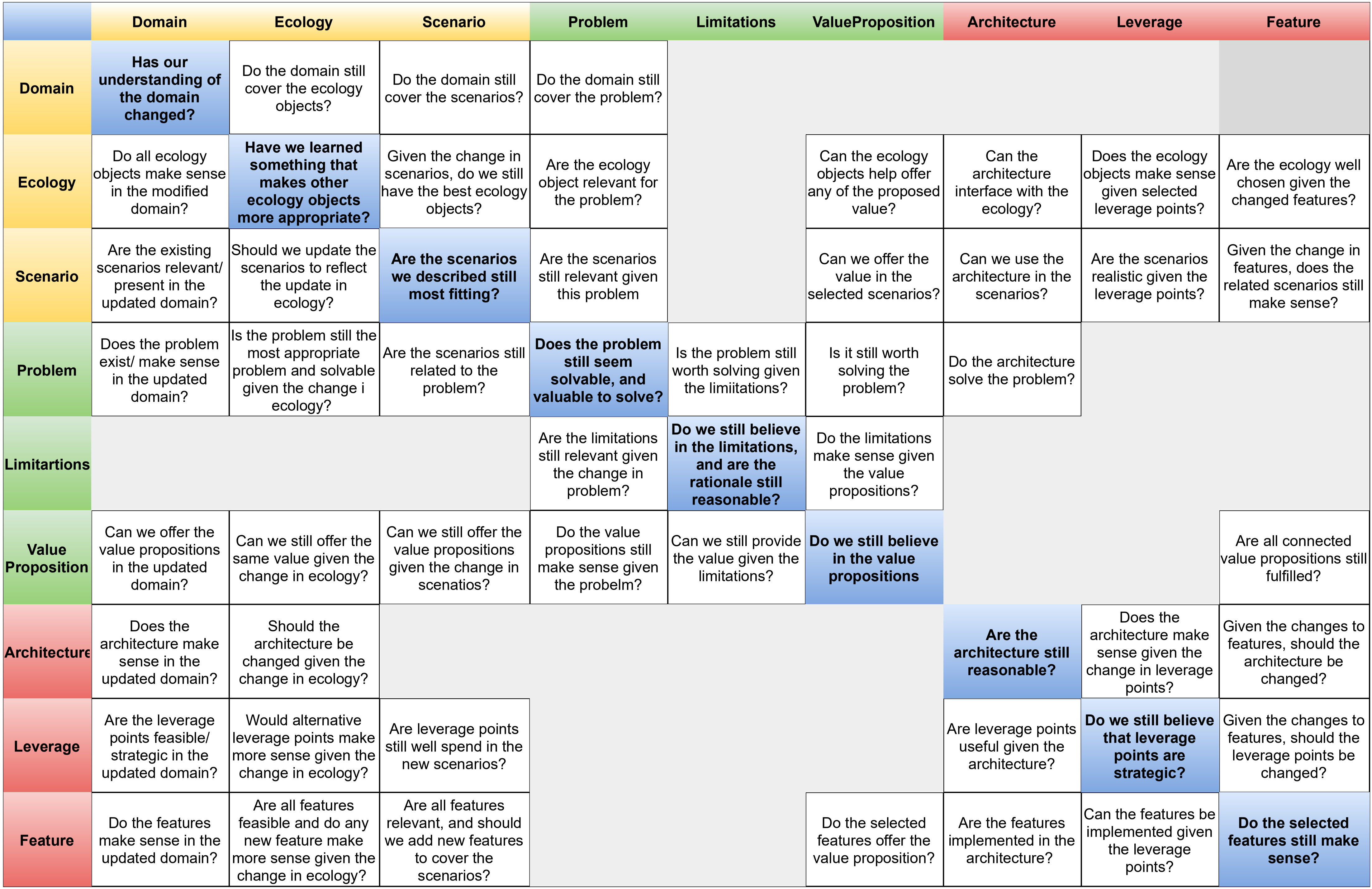
This table is just a suggestion, but I imagine that something similar might be helpful when learning how to fill out the configuration table.

	Domain	Ecology	Scenario	Problem	Limitations	ValueProposition	Architecture	Leverage	Feature
Domain	**Choose a domain**	Which domain does these ecology objects belong to?	Which domain does these scenarios belong to?	Which domain does this problem belong to?	Which domain do these qualifiers belong to?	Which domain do the value propositions share?	In which domain could we use this architecture to solve a problem?	In which domain could we use these leverage points so solve a problem?	In which domain could we use these features to solve a problem?
Ecology	Which ecology objects could be interesting in this domain?	**Select interesting ecology objects**	Which ecology objects could be relevant for this scenario?	Which ecology objects could be relevant when solving the problem?	Which ecology objects would make sense given the limitations in the design?	Which ecology objects could provide some of the value for this solution?	Which ecology objects would be useful to interface with the architecture?	Which ecology object would interface well with the leverage points?	Are there any existing ecology objects that offer this feature?
Scenario	Which scenarios could be relevant in the domain?	Which scenarios are related to the ecology?	**Select interesting scenarios**	In which scenarios does the problem arise?	Which scenarios could be interesting despite limitations?	In which scenarios could we offer the value proposition?	In which scenarios could we use the designed architecture?	In which scenarios would the leverage points be useful?	In which scenarios could we use the features?
Problem	What is a solvable problem within this domain?	Which problem could the ecology objects contribute to solving?	Which problem could be solved in the scenarios?	**Chose a solvable problem**	Which problem could be solvable despite the limitations?	Which problem would these value propositions contribute to solving?	Which problem could the architecture be used to solve?	Which problem could leverage points help solve?	Which problem could features be used to solve?
Limitations	Which areas in the domain would not be possible to solve?	Which limitations does the ecology objects bring?	Which limitations are there related to the scenario?	Which areas of the problem would not be realistic to solve?	**Define acceptable limitations, and reason why these are acceptable**	Is it realistic that we can offer the value to everybody involved?	Which technological limitations do we face?	Which limitations do we get from selecting the leverage points?	Are there any limitations for how well the feature should work?
Value Proposition	How can we bring value to the domain?	How could the solution interfacing with the ecology bring value?	What value could we add to the scenario?	How will solving the problem bring value?	What value could be added despite limitations?	**Define which value the solution could bring**	What value could the architecture add?	What value could leverage points help add if used correctly?	How does the feature bring value?
Architecture	Which solution would fit into the domain?	How could the architecture interface with the ecology?	How should we design the solution so it is useful in the scenarios?	Which solution would solve the problem?	Which architecture is possible despite limitations?	Which solution design could bring the values?	**Design an architecture for the solution**	How should leverage points be organized in order to create an architecture?	How should the solution be organized in order to provide the features?
Leverage	Which resources could be strategic in the domain?	Which leverage points would interface well with the ecology?	Which leverage points could be strategic in relation to scenario?	Which leverage points could be strategic for solving the problem?	Which leverage points could mitigate the limitations?	Which leverage points could be useful for providing the value?	Which leverage points could be strategic given the architecture?	**Choose strategic leverage points**	Which leverage points could be strategic for implementing the feature?
Feature	Which features could be useful in the domain?	Which features would make sense given the ecology objects?	Which features would make sense in the scenario?	Which features could help solve the problem?	Which features could mitigate the limitations?	Which feature could provide this value?	Which features would make sense given the architecture?	Which features would be possible given the leverage points?	**Select features for the solution**

Figure 6.12: Fill out matrix, suggests how to fill out a category, and use the information in one category to fill out another. The already known category/categories is selected on the horizontal axis(text direction) and the , on the category we want to fill out is on the vertical axis.

Change the current version of pre-project When I read the ESSENCE book, the pre-project seemed like the correct way to begin a project. However, the most appropriate approach for initiating a project depends on the type of project. For some projects, it makes sense to select a challenge (or domain), explore what the possibilities are for that domain, and then select an initial problem. However, some projects might use some specific technology as a starting point, and then select an appropriate problem based on what the leverage point can be used for. Instead of the current form, the pre-project could be described as the *initial phase of the project, where the direction for the project is decided.*

Add configuration table update suggestion At this point in time, there exists no suggestions for how to update the configuration table, and how to react when a category changes. In my opinion, it would be very useful to have suggestions for how to proceed when a category is changed. Figure 6.13 illustrates a suggestion of how this could be done, with suggestions on how to reflect on the category in the vertical axis, given changes in the category on the vertical axis. The blue cells suggest how to reflect on the category itself. The grayed-out cells suggest that reflection directly on the vertical category might not be necessary. This is only a draft and not my final suggestion.



	Domain	Ecology	Scenario	Problem	Limitations	ValueProposition	Architecture	Leverage	Feature
Domain	**Has our understanding of the domain changed?**	Do the domain still cover the ecology objects?	Do the domain still cover the scenarios?	Do the domain still cover the problem?					
Ecology	Do all ecology objects make sense in the modified domain?	**Have we learned something that makes other ecology objects more appropriate?**	Given the change in scenarios, do we still have the best ecology objects?	Are the ecology object relevant for the problem?		Can the ecology objects help offer any of the proposed value?	Can the architecture interface with the ecology?	Does the ecology objects make sense given selected leverage points?	Are the ecology well chosen given the changed features?
Scenario	Are the existing scenarios relevant/ present in the updated domain?	Should we update the scenarios to reflect the update in ecology?	**Are the scenarios we described still most fitting?**	Are the scenarios still relevant given this problem		Can we offer the value in the selected scenarios?	Can we use the architecture in the scenarios?	Are the scenarios realistic given the leverage points?	Given the change in features, does the related scenarios still make sense?
Problem	Does the problem exist/ make sense in the updated domain?	Is the problem still the most appropriate problem and solvable given the change i ecology?	Are the scenarios still related to the problem?	**Does the problem still seem solvable, and valuable to solve?**	Is the problem still worth solving given the limiitations?	Is it still worth solving the problem?	Do the architecture solve the problem?		
Limitartions				Are the limitations still relevant given the change in problem?	**Do we still believe in the limitations, and are the rationale still reasonable?**	Do the limitations make sense given the value propositions?			
Value Proposition	Can we offer the value propositions in the updated domain?	Can we still offer the same value given the change in ecology?	Can we still offer the value propositions given the change in scenatios?	Do the value propositions still make sense given the probelm?	Can we still provide the value given the limitations?	**Do we still believe in the value propositions**			Are all connected value propositions still fulfilled?
Architecture	Does the architecture make sense in the updated domain?	Should the architecture be changed given the change in ecology?					**Are the architecture still reasonable?**	Does the architecture make sense given the change in leverage points?	Given the changes to features, should the architecture be changed?
Leverage	Are the leverage points feasible/ strategic in the updated domain?	Would alternative leverage points make more sense given the change in ecology?	Are leverage points still well spend in the new scenarios?				Are leverage points useful given the architecture?	**Do we still believe that leverage points are strategic?**	Given the changes to features, should the leverage points be changed?
Feature	Do the features make sense in the updated domain?	Are all features feasible and do any new feature make more sense given the change in ecology?	Are all features relevant, and should we add new features to cover the scenarios?			Do the selected features offer the value proposition?	Are the features implemented in the architecture?	Can the features be implemented given the leverage points?	**Do the selected features still make sense?**

Figure 6.13: Update matrix, suggests how to update categories and proceed when a category is updated. Given that the category in the horizontal axis is updated, how will this affect the category in the vertical axis?.

6.1.6 Sprint 4 conclusion

When I began Sprint 4 I expected that the visual design I defined in this sprint would work as a basis for Sprint 5, where I would change the design according to the design choices I made in this sprint. However, during this sprint I got inspired to design flow matrices for the ESSENCE categories. In doing this, I reflected on whether the real difficulty of ESSENCE is to understand what the different categories are in themselves, or whether it is actually to understand how the categories are related, and how information in one category can affect information in others. My conclusion is, at this point, that it is more important for people who learn ESSENCE to get an idea of how they use the *flow of information* to learn something new about the project than it is to explain the basic concepts with visual support. Therefore, I believe that the goal for sprint 5 will be to:

- Help users enter known information in the correct place
- Help users utilize filled out categories to fill out additional categories
- Help users identify categories that have changed
- Help users identify other categories that should be revisited based on the change in one category

In order to do this, I will most likely change the flow of the tool as it is. Currently, the tool follows a suggestion of the project lifecycle, but I want it to instead follow the flow of knowledge and ideas.

6.1.7 RST Review expectations

What do I expect my solution will be to the problem?
I expect that my solution will be a tool that can be used by both beginners and advanced users of ESSENCE to facilitate project understanding exploration.
What do I expect to gain from this activity?
I expect that the RST Review will help me understand how my vision has changed, and how that should affect my solution.
How useful do I expect this activity to be?
I expect that the review will be very useful.

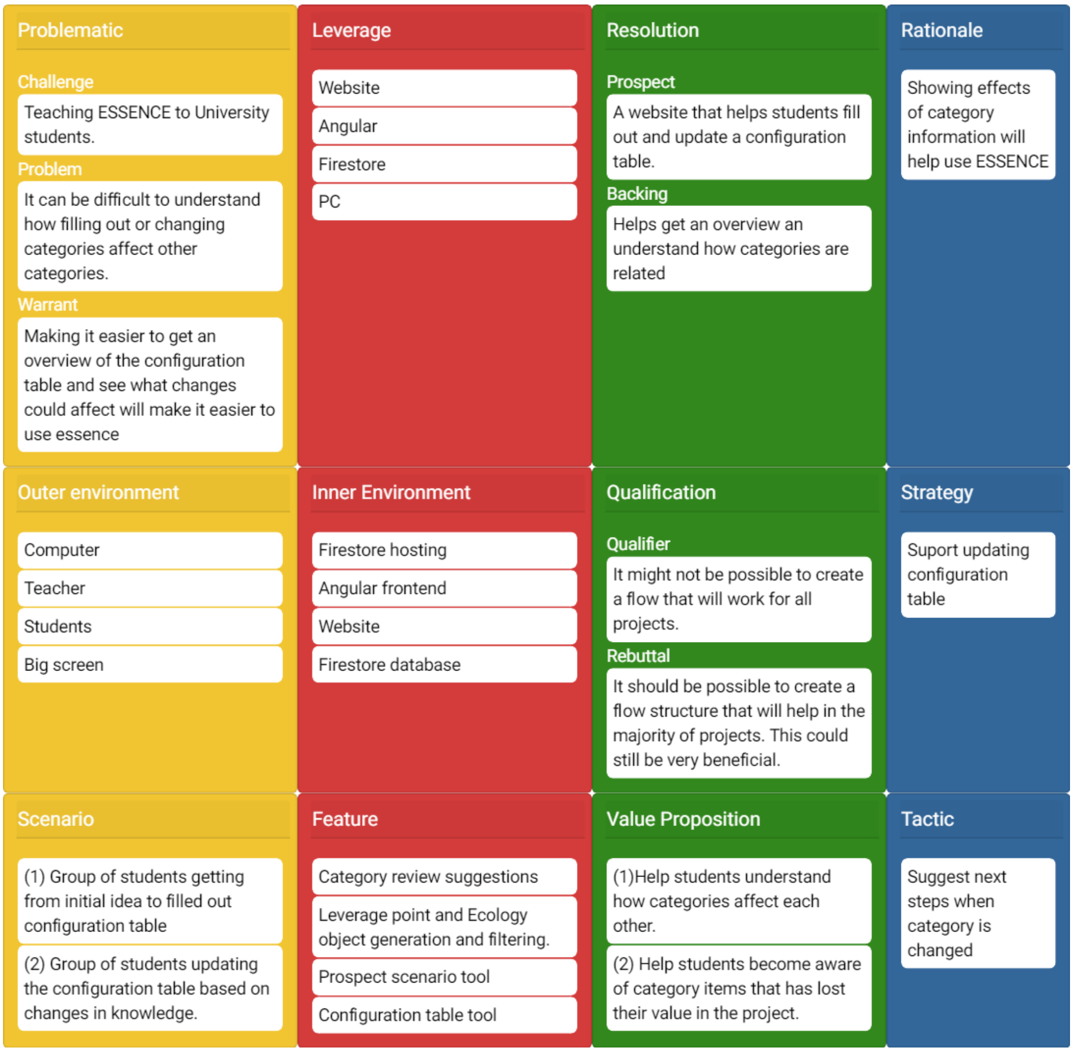
6.1.8 RST Review

As I have not made any real updates to the product in this sprint, collecting comments will not be a part of the review. Instead, I will use the update flow matrix6.13 as a starting point for my updates.

I will do this to both try out if this is useful, and to identify any obvious changes that could be made.

During this sprint, Aaen sent me an updated version of his vision for the configuration table. The resulting configuration table will reflect these changes. The updated layout is described in A.9.

The configuration table update is described in A.10. The resulting configuration table is displayed in 6.14.



Problematic Challenge Teaching ESSENCE to University students. Problem It can be difficult to understand how filling out or changing categories affect other categories. Warrant Making it easier to get an overview of the configuration table and see what changes could affect will make it easier to use essence	Leverage Website Angular Firestore PC	Resolution Prospect A website that helps students fill out and update a configuration table. Backing Helps get an overview an understand how categories are related	Rationale Showing effects of category information will help use ESSENCE
Outer environment Computer Teacher Students Big screen	Inner Environment Firestore hosting Angular frontend Website Firestore database	Qualification Qualifier It might not be possible to create a flow that will work for all projects. Rebuttal It should be possible to create a flow structure that will help in the majority of projects. This could still be very beneficial.	Strategy Suport updating configuration table
Scenario (1) Group of students getting from initial idea to filled out configuration table (2) Group of students updating the configuration table based on changes in knowledge.	Feature Category review suggestions Leverage point and Ecology object generation and filtering. Prospect scenario tool Configuration table tool	Value Proposition (1)Help students understand how categories affect each other. (2) Help students become aware of category items that has lost their value in the project.	Tactic Suggest next steps when category is changed

Figure 6.14: Sprint 4 configuration table

6.1.9 RST Review Evaluation

How did ESSENCE affect the course of action?
It did not directly affect the course of action, but did help me reflect on how my

project has changed.
Was the activity overall helpful for the project?
Yes.
Which benefits did ESSENCE have in this activity? Would they still be benefits in a multi-person team?
I think the RST Review helped me understand how the scenarios and related features have changed. I think it would be at least as beneficial in a multi person team, as the discussion would be very useful.
Which drawbacks did ESSENCE have in this activity? Would they still be drawbacks in a multi-person team?
The update flow matrix was of great help as it gave me a starting point and relevant questions to get going. In previous sprints it has been very difficult to keep an overview of all categories. The drawback of ESSENCE is, that it does not have a similar, native, matrix.
The criteria still does not make sense to me.
Can I imagine some modifications to ESSENCE that could improve it?
I think that there is a great need for a useful update matrix.
Additionally, I would need some more help or explanation to use the criteria, as these still does not make any sense to me.
How was the ESSENCE roles and views utilized in this activity?
Views were utilized as usually. I would like to remark, that I like the paradigm and product view, but it is not clear to me what exactly the project view should be, and how to use the process view.
Other comments

6.2 Sprint 5

6.2.1 Scope

In sprint 4 I learned that the fill out flow should be central to the solution. In this scope I will implement this in the solution. The first step will be to remove the pre project and replace it with a *select what you know about the project at this point* page. Then I will implement help for dynamically filling out the categories, move the SWOT and PCRT analysis and the prospect scenarios.

6.2.2 Final product

Sprint 5 is the last sprint in this project. The product developed at this point is therefore the final product of this project. Figure 6.15 illustrates the UI for the resulting product.

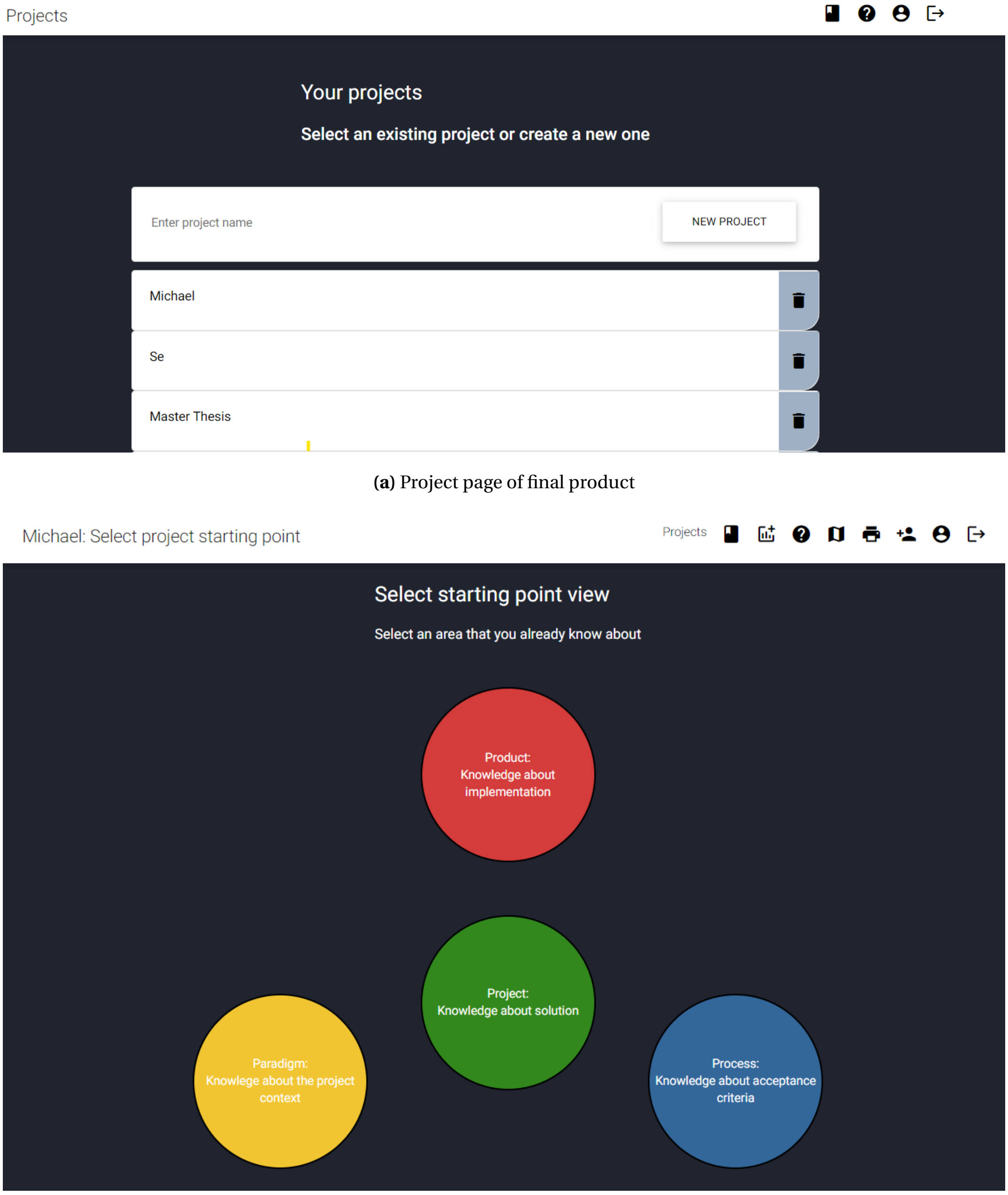
The project flow in this product is:

- Create and select projects, figure 6.15a
- When a project is selected for the first time, the user select a starting point for the project by specifying which area they have knowledge about, figure 6.15b
- When the overall area of the initial knowledge (view) is selected, the user selects a sub category of that area, figure 6.15c
- After a category is selected, the user is able to fill out that category in an empty configuration table, figure 6.15d. The user can edit different categories, one at a time. When a category is edited, related categories gets lighter, as a sign that they should be examined. In the figure we have entered the solution proposition and the problem, and therefore the scope proposition, leverage and rationale, which are directly related are highlighted. When the initial configuration is done, it can be accepted (bottom right corner).
- When the initial configuration table have been accepted, the user is lead to an update configuration screen, figure 6.15e, where there is a gray box over each category. This is used to guide the user so that the user can check that the category is still reasonable. Like previously, categories are highlighted when connected categories are edited.
- Figure 6.15f shows how item connections are displayed. When the arrow to the right side of an item is clicked, the user is able to make connections to items in other categories by clicking on them. When an item is selected, connected items are highlighted if the parent category is opened. If an item does not have the required connections, it is written in cursive, to illustrate that the item is not backed properly.
- Most categories are able to have multiple items. For these categories, a PCRT analysis can be opened, so the user is able to prioritize the items. This is shown in 6.15g. Likewise, it is possible to do a SWOT analysis and switch between these analysis as the user sees fit.
- As the pre-project is removed from the product, the prospect scenario was not naturally part of the product. As this tool still might be useful, I moved it to the toolbar, so the user can use it at any point of the project. This is displayed in figure 6.15h.

The categories in the diagrams different than the categories in the updated version of ESSENCE described in 6.1.8, as the domain, warrant, backing and rebuttal categories are gone, and the qualification category changed name to scope proposition. After talking to Aaen, he accepted most of these changes, but did not agree to the scope proposition name change.

Even though the general outline of the product is done, there are still many details that could be added to improve the product. A list of suggestions can be found in A.11.

Figure 6.15: Sprint 5 final product



Projects

Your projects

Select an existing project or create a new one

Enter project name

Michael

Se

Master Thesis

(a) Project page of final product

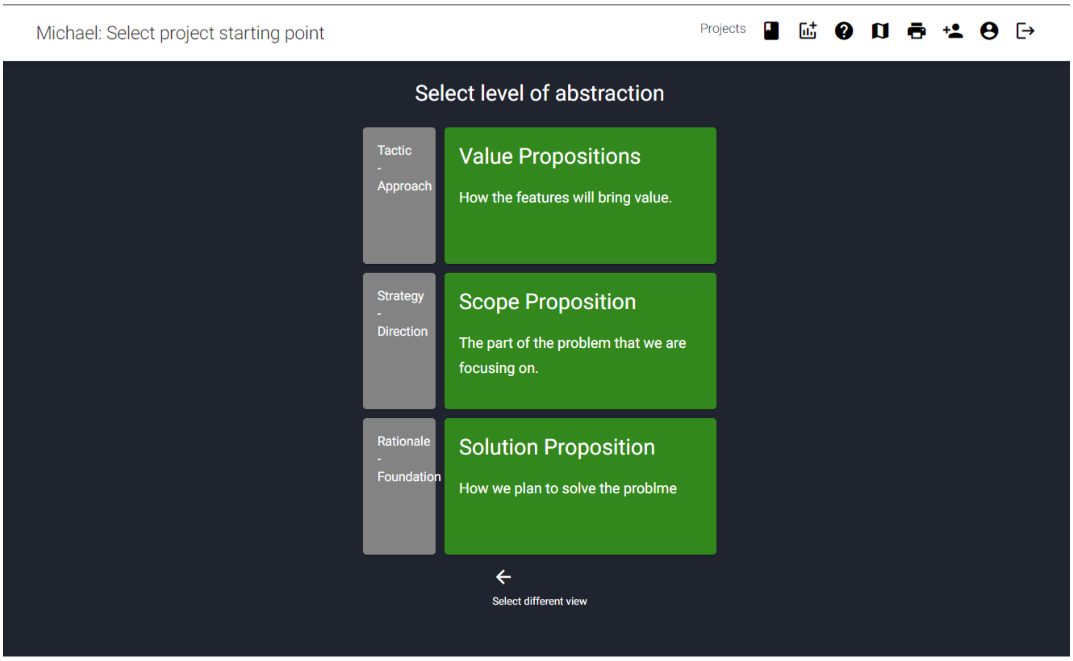
Michael: Select project starting point

Select starting point view

Select an area that you already know about

Projects
Product:
Knowledge about implementation
Project:
Knowledge about solution
Paradigm:
Knowlege about the project context
Process:
Knowledge about acceptance criteria

(b) Select starting point



Michael: Select project starting point

Select level of abstraction

Tactic - Approach

Value Propositions

How the features will bring value.

Strategy - Direction

Scope Proposition

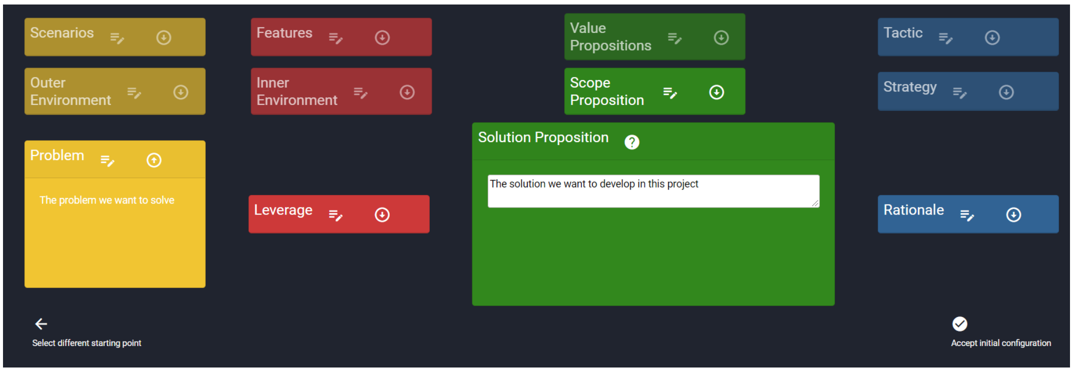
The part of the problem that we are focusing on.

Rationale - Foundation

Solution Proposition

How we plan to solve the problme

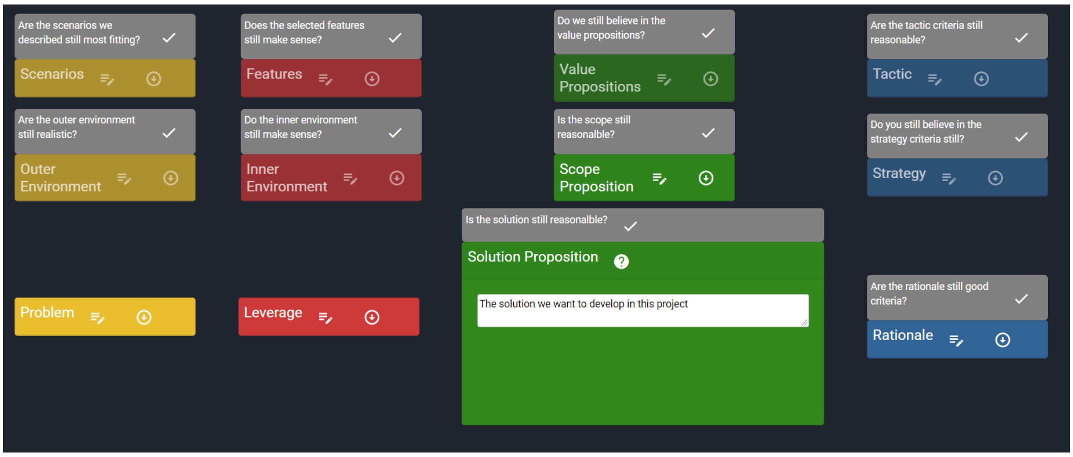
(c) Select category



The problem we want to solve

The solution we want to develop in this project

(d) Initial configuration



Are the scenarios we described still most fitting?

Does the selected features still make sense?

Do we still believe in the value propositions?

Are the tactic criteria still reasonable?

Are the outer environment still realistic?

Do the inner environment still make sense?

Is the scope still reasonalble?

Do you still believe in the strategy criteria still?

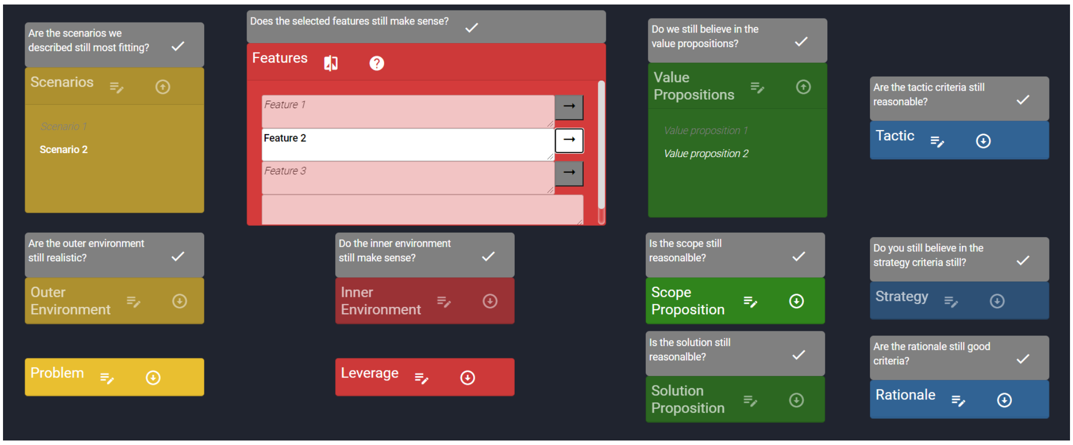
Is the solution still reasonalble?

The solution we want to develop in this project

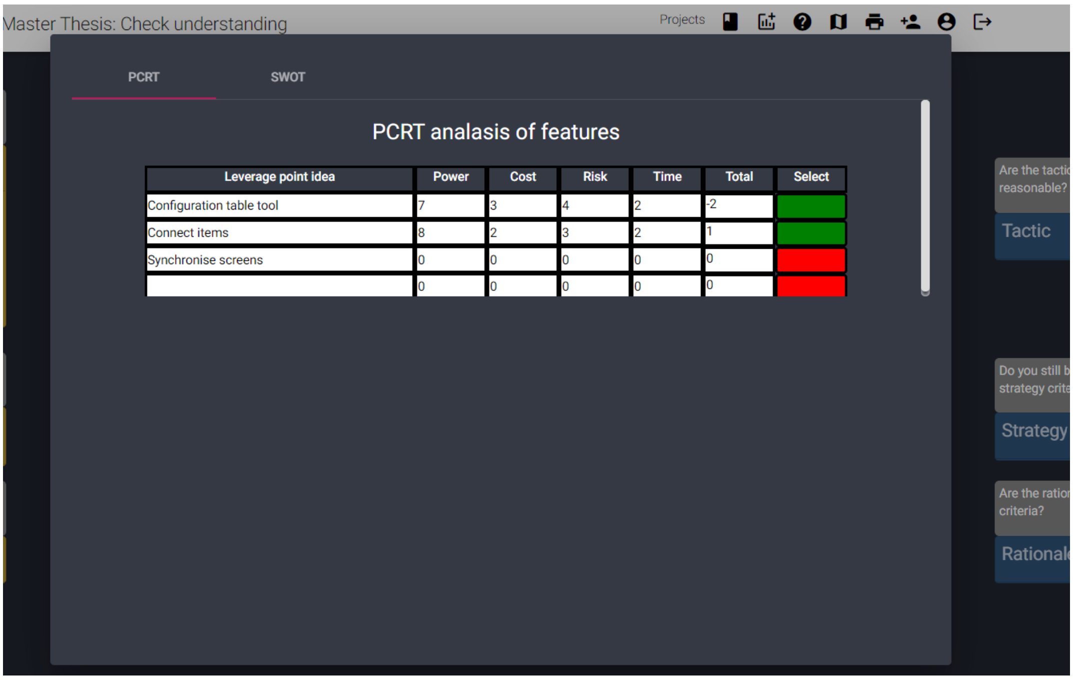
Are the rationale still good criteria?

(e) Update configuration

6.2. Sprint 5


Are the scenarios we described still most fitting?
Scenarios
Scenario 1
Scenario 2
Does the selected features still make sense?
Features
Feature 1
Feature 2
Feature 3
Do we still believe in the value propositions?
Value Propositions
Value proposition 1
Value proposition 2
Are the tactic criteria still reasonable?
Tactic
Are the outer environment still realistic?
Outer Environment
Do the inner environment still make sense?
Inner Environment
Is the scope still reasonalble?
Scope Proposition
Do you still believe in the strategy criteria still?
Strategy
Is the solution still reasonalble?
Solution Proposition
Are the rationale still good criteria?
Rationale
Problem
Leverage

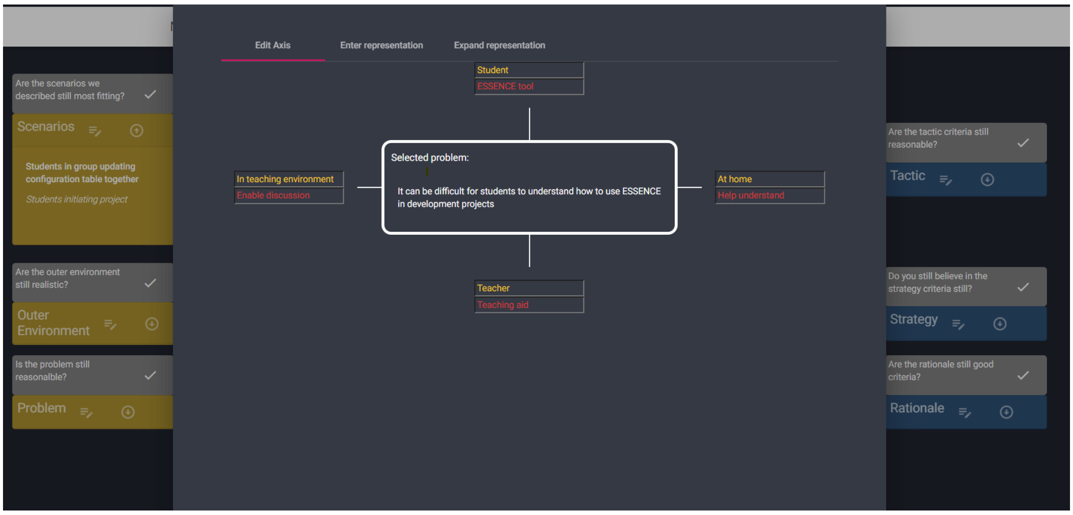
(f) Connect items in configuration table


Master Thesis: Check understanding
Projects
PCRT
SWOT
Are the tactic criteria still reasonable?
Tactic
Do you still believe in the strategy criteria still?
Strategy
Are the rationale still good criteria?
Rationale

PCRT analasis of features

Leverage point idea	Power	Cost	Risk	Time	Total	Select
Configuration table tool	7	3	4	2	-2	🟩
Connect items	8	2	3	2	1	🟩
Synchronise screens	0	0	0	0	0	🟥
	0	0	0	0	0	🟥

(g) PCRT analysis


Edit Axis
Enter representation
Expand representation
Student
ESSENCE tool
In teaching environment
Enable discussion
At home
Help understand
Teacher
Teaching aid
Are the scenarios we described still most fitting?
Scenarios
Students in group updating configuration table together
Students initiating project
Are the outer environment still realistic?
Outer Environment
Is the problem still reasonalble?
Problem
Are the tactic criteria still reasonable?
Tactic
Do you still believe in the strategy criteria still?
Strategy
Are the rationale still good criteria?
Rationale

Selected problem:

It can be difficult for students to understand how to use ESSENCE in development projects

(h) Prospect scenarios

Chapter 7

Findings

7.1 Discussion

7.1.1 How did my understanding of ESSENCE evolve during the project?

Initially, I understood ESSENCE to be mainly instructions about how to carry out a development project so that the development team has an overview of the project and notice incorrect assumptions as early as possible. A good example of this can be found in my expectations for the pre-project 4.3.1, were my expectations for the pre-project mostly focused on how it would change my understanding of the problem, and giving me insights into the possibilities. However, already during the pre-project I found that the benefit of ESSENCE was more related to reflecting on different views on the project. This can be seen in the evaluation 4.3.6. Another example of this is in the prospect scenario activity. Here I found that even though the course of action did not change significantly, I gained a much clearer vision for the project 4.4.4. By sprint 4, the RST review 6.1.9 was able to help me gain a deeper understanding of what the project is, and the process of updating the configuration table, especially with the help of the update matrix (figure 6.13) enabled me to reveal weak points of the project. At this point, my understanding of ESSENCE is completely different from what it was initially in the project. At this point, I do not view ESSENCE as a set of instructions, but instead, as a tool that can be used, when appropriate, to enable reflection and guide the creative process. This is done when the categories are filled out or updated, as the task of filling out the categories requires a lot of thought. In other development projects I have generally worked from a somewhat fuzzy idea of what the project is and what I wanted to do, but never really put it into words. However, by putting it into words, the ideas have to be concrete. In my experience, it is this process of getting from fuzzy ideas to concrete descriptions that brings value in ESSENCE and facilitates reflection. And continuously updating the concrete descriptions ensures that the team never develops projects based on only a fuzzy idea.

7.1.2 Benefits, drawbacks, and modifications of ESSENCE concepts

During the project, I gained much experience on how the different concepts, in ESSENCE, contribute to the reflective process. In the following, I discuss central ESSENCE concepts and describe benefits, drawbacks and suggest modifications.

Configuration table: **Benefit**: The most important concept of ESSENCE is that of the configuration table. In my experience, the configuration table benefited the project in two ways. The first, and maybe most obvious way, is that it helped get an overview of the project understanding and made it clear when something significant had changed in the understanding of the project. The second, and in my opinion most important way, is that it forced me to formulate my description and thereby organize my thoughts. Often, I had the feeling that I knew exactly what the project was, but it was still difficult to fill out, or update, the configuration table, and the process of filling it out made my understanding more concrete. **Drawback**: The definition and utility for some categories are not very clear. This made it difficult to actually gain any utility from them and made them more like clutter than anything else. This is especially true for architecture(inner environment), where both the purpose and connection to other categories are still unclear to me, but also ecology(inner environment) and qualifier. In my experience, none of these generally added any benefit and felt more in the way. **Modification**: There are two possible ways to mitigate this problem. Either, by removing the strategy level altogether, or by providing clear guidelines and better descriptions for the different categories. However, even with the best descriptions, I still think that is necessary to investigate if all categories are really necessary and do provide value in a development project.

RST review:
Benefit: I found that the RST review generally proved helpful, and several times I was surprised that my understanding had actually changed significantly. I think the timing of the RST review is good, as it is carried out frequently enough to catch changes in understanding quickly, but not so often that it is in the way. **Drawback**: The guidelines for the RST review were very confusing. There were specific guidelines for how to get feedback from stakeholders, but none for how to actually carry out the review. This made the feedback seem like it was much more important than updating the configuration table, and made it hard to figure out how to update the table. Also, the lack of guidelines for when to update the configuration table and how changes in different categories affect other categories, combined with a large number of categories, made it difficult to get an overview when updating. **Modification**: There should be clear guidelines for updating the configuration table, both for the individual categories and for how changes in a specific category should affect other categories.

Pre-project: **Benifit**: When I did the pre-project I found it to be very useful. It helped me add some reflection on the project before actually doing any work. **Drawback**: The way the pre-project is described at this point does not support all types of development projects, as the reflective process begins in a specific category. **Modification**: The pre-project should be more open, and support beginning in different categories.

Roles: **Benefit**: I did not use the roles at all, only the views. **Drawback**: I found that the concept of roles did not have any practical utility in this project. This could be because I was the only team member. However, as they did not seem necessary in this project, and the ESSENCE book provides no specific guidelines for how to use them, it is not clear to me that they are really needed. **Modification**: If this is the case, they should be excluded from ESSENCE as all concepts that do not bring value should be considered clutter.

Views Benefit: The views were useful for separating concepts conceptually. **Drawbacks**: I did not find the process view very intuitive, and am still not sure how to use it. **Modification**: There could be better explanations about the process view.

7.1.3 Did I succeed in reducing the learning curve of ESSENCE?

During the project, I developed several hypotheses for how to reduce the learning curve of ESSENCE. Initially, these were mainly focused on reducing the complexity of ESSENCE or presenting ESSENCE in an intuitive way for students. Two hypotheses worked as the foundation for the initial development of the product. The first was that *students would find ESSENCE more intuitive if they see how it can be used in a Scrum development project.* This hypothesis was especially important during my third sprint and can be inferred from the value propositions of sprints 2 and 3 in figure 5.8 and 5.10. The second was that *students find it difficult to understand the individual activities of ESSENCE, and would benefit from a slow introduction of concepts.* This can be inferred from the scenario and value propositions of sprint 3 5.10.

As my understanding matured in sprint 4, I came to realize that these hypotheses were not coherent with my understanding of ESSENCE at this point. I found that it would be more important for students to *understand how to use the flow of information in ESSENCE* than to introduce individual concepts, as this is where ESSENCE is the most powerful (section 6.1.6). Therefore I restructured the development tool. My hypothesis at this point is, that the tool would make it more intuitive for students to learn ESSENCE. This is based on the idea that the configuration table is the main contribution of ESSENCE. Therefore, it is essential that students understand how to properly use it. The more intuitive students find the configuration table, the more comprehensible they should find other concepts and activities of ESSENCE, as they are mainly designed to support the configuration table. E.g., the purpose of both the PCRT analysis and the prospect scenarios is to aid in filling out the categories.

The only valid approach I have identified for testing this hypothesis and thereby to show whether or not the tool reduces the learning curve involves students from the software innovation course. Either by getting feedback from students using the tool or by conducting an experiment to see if students using the tool learn ESSENCE faster. I would have liked to conclude my work this semester with an evaluation of my product. In order to do this, I reached out to current students of the Software Innovation course. However, only one student, Theresa Walker Junker, volunteered, making it impossible to conduct a general experiment. Junker did respond positively to the product, exclaiming: *this is just what I needed*, but in order to show whether or not the learning curve is really reduced, there should be conducted a proper experiment with, or collect feedback from, multiple students.

7.2 Conclusions

In this project, I carried out a development project where I used ESSENCE. The aim was to answer the following questions:

- **Q1**: Which benefits and drawbacks can ESSENCE have in a development project?
- **Q2**: How could ESSENCE be modified in order to improve the utility?

Generally, I find that ESSENCE did benefit the development project, mainly by promoting reflection. However, it was not a straightforward process, and ESSENCE is something completely different to me now than when I began the project. I, therefore, imagine that my experience with ESSENCE would be different if I began a development project with my current understanding. In section 7.1 I discuss my overall experience with ESSENCE and provide a detailed description of the benefits, drawbacks, and proposed modification for many of the ESSENCE concepts. However, I found that the main benefit of ESSENCE was to improve reflection through the configuration table. Therefore, the conclusion will focus on this.

The configuration table promoted reflection on multiple levels. Filling out the individual categories of the configuration table helped to reflect on them each. Initially, I only had a fuzzy idea of the content of categories like *scenarios* and *value propositions*. In previous projects, I would find a fuzzy idea of the scenarios to be sufficient, and not even consider value propositions. However, in order for me to be able to actually fill out the categories, I had to have a clearly defined understanding. This helped me get a clear understanding of what the project is.

On another level, the categories are connected, and reflecting on their connections helped me to learn more about the individual categories. One example is during my RST review for sprint 4 (section A.6). Here I learned that updating the scenarios that my product should support had a cascading effect on other categories including the problem. This lead to a redesign of the entire structure of the product.

The main drawback I experienced for ESSENCE is that it is very difficult to understand and use the configuration table. There are no clear guidelines both for filling out individual categories and for reconsidering other categories when one is changed.

The lack of guidelines for individual categories made it difficult to understand how that category should be used right away. I found the categories to of varying levels of intuitiveness. Categories like *problem* and *scenarios* were very intuitive to me, and consequently easy to fill out and maintain. However, I found categories like *qualification* and *architecture* were very difficult to fill out, and my attempts did not provide much value to me. Part of this was mitigated with experience, and as categories became more intuitive I found them to be more helpful. However, at this point, it is still unclear to me how to use the *architecture(inner environment)* category, even after discussing this in-depth with ESSENCE author Ivan Aaen. In my opinion, each category should be intuitive for the development team in order to be helpful. It is possible that the *architecture* category might become clear to me after more experience, or make more sense in different development projects. However, the less intuitive categories were to me, the more time and energy I used on them. I find it likely that other developers will face the same problems. Therefore I find that the categories in the configuration table should either be reconsidered in order to make them more intuitive, or ESSENCE should be updated with clear guidelines to help use the categories. During sprint 4 I introduced the *fill out matrix* (figure 6.12) in an attempt to mitigate this.

I find the lack of guidelines for reconsidering other categories when one is changed at least as problematic as for the individual categories. I was unable to simultaneously remember which categories and items had been changed, and identify which other categories and items should be reconsidered. This limited the utility of the connections between categories. During sprint 4 I introduced an update matrix6.13, which I used in the sprint 4 RST review6.1.8. I found this to be very helpful 6.1.9, and based on this I imagine that introducing such a matrix into ESSENCE would reduce this issue significantly.

ESSENCE can be useful in a development project. However, the steep learning curve and the lack of guidelines makes it difficult to utilize. Therefore, I find it much more likely that development teams will adopt the methodology in the future if they were provided with an easily comprehensible description of ESSENCE and clear guidelines for the main activities of ESSENCE.

7.3 Future work

Even though I found ESSENCE to be useful in my development project, it is clear to me that it still need work to be applicable in real development projects. Therefore I make the following proposals for future work on ESSENCE.

More experiments As my experiment only involves one developer on one development project, this is not nearly enough to conclude anything general about ESSENCE. Therefore there should be done several additional studies into the usefulness of ESSENCE. Some ideas for focus in these projects could be:

- How ESSENCE affect communication between team members in multi-person development team?
- How other developers experience ESSENCE?
- How ESSENCE affect different types of development projects=

- How useful ESSENCE is for experienced users vs new users?
- How does ESSENCE affect creativity in a development team?

All these experiments could provide valuable information about ESSENCE.

Configuration table One major issue I have experienced in this development project is the lack of clear guidelines for the configuration table and confusion about how individual categories should be used. Therefore I suggest that future work should focus on:

- Developing an update matrix that reflect how different categories affect each other
- Developing a procedural approach for updating the configuration table
- Ensuring that the categories in the configuration table really reflect a useful overview of the project

Additionally, there could be work into other areas like the pre-project or the RST review. However, in my opinion, the best way forward is through experimental research with real development teams. By getting the real experience of development teams we can understand how ESSENCE can benefit development projects and how we can improve ESSENCE in the future.

7.4 Acknowledgements

I want to thank my supervisor Ivan Aaen for being exceptionally supportive and always answering my criticism of ESSENCE with a smile and a long discussion. I also want to thank software student Theresa Walker Junker for volunteering to share her experience of ESSENCE with me and providing me with feedback for my product.

Bibliography

[1] Ivan Aaen. *Essence Problem Based Digital Innovation.* 2021.

[2] Oxford dictionary. *qualification noun.* URL: https://www.oxfordlearnersdictionaries.com/definition/english/qualification?q=qualification.

[3] Oxford dictionary. *qualifier noun.* URL: https://www.oxfordlearnersdictionaries.com/definition/english/qualifier?q=qualifier.

[4] Google. *Google Cloud Web Hosting.* 2020. URL: https://cloud.google.com/solutions/web-hosting.

[5] Google. *Serving websites.* 2020. URL: https://cloud.google.com/solutions/web-serving-overview.

[6] Donald A. Schon and Glenn Wiggins. "Kinds of seeing and their functions in designing". In: *Design Studies* 13.2 (1992), pp. 135–156. ISSN: 0142-694X. DOI: https://doi.org/10.1016/0142-694X(92)90268-F. URL: https://www.sciencedirect.com/science/article/pii/0142694X9290268F.

[7] D.A. Schön. "Designing as reflective conversation with the materials of a design situation". In: *Knowledge-Based Systems* 5.1 (1992). Artificial Intelligence in Design Conference 1991 Special Issue, pp. 3–14. ISSN: 0950-7051. DOI: https://doi.org/10.1016/0950-7051(92)90020-G. URL: https://www.sciencedirect.com/science/article/pii/095070519290020G.

[8] Scrum.org. *What is SCRUM?* scrum.org. 2021. URL: https://www.scrum.org/resources/what-is-scrum.

[9] WP Amelia Staff. *Static vs Dynamic Website: What Is the Difference.* 2019.

[10] Claudia J. Stanny. *Facts and Fantasies about How Students Learn.* URL: https://ctl.byu.edu/tip/facts-and-fantasies-about-how-students-learn.

[11] Iris Vessey, Sirkka L. Jarvenpaa, and Noam Tractinsky. "Evaluation of Vendor Products: CASE Tools as Methodology Companions". In: *Commun. ACM* 35.4 (Apr. 1992), 90–105. ISSN: 0001-0782. DOI: 10.1145/129852.129860. URL: https://doi.org/10.1145/129852.129860.

Appendix A

Appendix

A.1 Appendix

This is the appendix

A.2 SWOT analysis for ecology objects

With a number of potential ecology objects identified in Section 4.3.3, I analyze their relevance using the SWOT framework. For each potential object, I list the strenghts, wearknesses, oppotunities and threats. In order to compare the objects to each other, I use the modified version defined in [1].

Diagram tool: 0

We could implement a diagram tool window or something similar into the interface, that users could use to fill out the diagrams and scenarios

Strengths: 3

- Simple to implement
- Likely free or very cheap
- We can make a standard diagram that users can fill out
- Can likely be integrated into the solution

Weaknesses: -4

- Probably not possible to customize enough so that users will have a seamless experience
- Would likely add a lot of unnecessary functionality, which would only confuse users

- It would probably be easy to implement an alternative myself

Opportunities: 3

- Can be used as a stand-in until a dedicated option can be implemented

Threats: -2

- Support can be removed
- There could be problems when integrating it into the site

Quiz tool: -1

Implementation of ta quiz tool in order to provide feedback on learning goals. **Strengths**: 3

- Possible additional features
- Easy to implement

Weaknesses: -5

- Not necessarily designed to identify gaps in knowledge
- Would probably be easy to develop something like this myself, so it could be not worth the trade-off

Opportunities: 3

- Dedicated version can be implemented later

Threats: -2

- Support can be removed
- There may not be an option that can be integrated directly into the solution, so users could be forced to switch between interfaces, which could be annoying.

Process framework tools: 1

Interfacing with process framework tools for selected process framework in order to show how ESSENCE can be used together with process framework. **Strengths**: 5

- Easy implementation
- Can help put ESSENCE into perspective
- Would take a lot of work to develop all options

Weaknesses: -2

- Possibly many different interfaces

Opportunities: 1

- Proper integration could possibly lead to collaboration with other sites, and increased exposure for ESSENCE. However, this is probably unlikely, and it is not clear that increased exposure would have that big of an effect.

Threats: -3

- Support can be removed
- People can be frustrated if the specific framework they want to use is not supported
- It may be difficult to figure out how the tools could work together

Brainstorming tools: 1

Tools to help students in brainstorming sessions which is an essential part of ESSENCE.
Strengths: 4

- Help with brainstorming aspects, which is an essential part of ESSENCE
- Easy implementation

Weaknesses: -3

- Can be difficult to include elegantly

Opportunities: 2

- Can be implemented later directly into the system, leaving a minimum of wasted work.

Threats: -2

- Support can be removed
- It may not be possible to integrate into the website, in which case the user would have to switch interfaces

Camera: -2

We can use a camera to take pictures of filled out diagrams. **Strengths**: 3

- Easy to implement
- Some might prefer filling out the diagrams by hand.
- Free to implement
- Almost all developers would has a camera available in their phone

Weaknesses: 5

- It can be difficult for other's to read content
- It would be difficult to edit later
- Some people prefer filling out diagrams digitally

Opportunities: 0

-

Threats: 0

-

Whiteboard :2

The solution could be used by a teacher who during his lectures could show content. This could e.g., be when he is explaining concepts, then he could use content from the system as visual aid.

Strengths: 3

- Could help the students concentrate and follow the teaching

Weaknesses: -2

- Requires a teacher

Opportunities: 3

- Does probably not require that much additional effort to provide the visual aids if the solution offer similar aid to individual students learning ESSENCE.
- Could be used in many courses if ESSENCE became widespread

Threats: -2

- During Covid-19, many students and teachers have gotten used to online courses, so in the future the number of physical lectures might be limited

Microphone: -4

We could enable students to ask questions and others to answer the questions in audio format. **Strengths**: 2

- Can be more personal than writing
- Could be preferable for students as they would not have to think about spelling

Weaknesses: -4

- Audio quality can be very varied
- It can be difficult to find information fast in audio format

Opportunities: 0

-

Threats: -2

- Many people might not want to use the microphone, but instead write, as this can be more anonymous.

Big screen :4

Like a TV, that development teams could use to brainstorm and carry out activities together.

Strengths: 5

- Help students work together and discuss concepts.
- Cheap

Weaknesses: -2

- Need several students.

Opportunities: 4

- Can be used both in teaching and in real development projects

Threats: -3

- Many groups will not have a big screen available.

Microphone: -4

We could enable students to ask questions and others to answer the questions in audio format. **Strengths**: 2

- Can be more personal than writing
- Could be preferable for students as they would not have to think about spelling

Weaknesses: -4

- Audio quality can be very varied
- It can be difficult to find information fast in audio format

Opportunities: 0

-

Threats: -2

- Many people might not want to use the microphone, but instead write, as this can be more anonymous.

Entity information: 1

We can collect entity information about possible ecology objects and leverage points in order to help boost creativity in brainstorming sessions. **Strengths**: 4

- Can help enable creativity

Weaknesses: -4

- Requires some work to figure out how to suggest entities

Opportunities: 4

- Could become a valuable tool when using ESSENCE

Threats: -3

- Suggestions may not be relevant

Teacher providing feedback: 1

During an ESSENCE course, students could ask questions in the solution that the teacher can answer **Strengths**: 4

- Gives relevant, tailored feedback from an expert
- Easy to implement

Weaknesses: -3

- Not very scalable

Opportunities: 4

- In the future if ESSENCE gets more adopted, there could be many experts
- Feedback could be saved and maintained so other students in the future can gain from this

Threats: -4

- There are not many experts in ESSENCE at present, so the waiting time could be very long

Teacher as course supplement: 3

The teacher could use the solution to make the student practice theory or get introduced to new concepts as a supplement to the education. **Strengths**: 4

- The solution does not have to explain everything alone, and any questions can be taken to the teacher
- It can help visualize concepts and support the overall learning

Weaknesses: -2

- Requires a teacher

Opportunities: 4

- Students can use the system to explain concepts to each other

Threats: -3

- Quality of learning depends a lot on the teacher
- It can be difficult to design the solution so that everybody will understand different concepts if it is not a stand alone solution

Other students: -1

Students can interact with each other in the solution and help each other understand concepts **Strengths**: 4

- Free
- Very scalable

Weaknesses: -3

- We do not have any guarantees that the other students understand it correctly themselves

Opportunities: 4

- Can promote student involvement and engage students in good discussions

Threats: -5

- Requires volunteers, so maybe none or only few will get involved
- No guarantee that a question will be answered

A.3 Leverage PCRT analysis

After generating potential leverage points, I did a comparative PCRT analysis in order to determine which leverage points could be of high priorty.

Teaching knowledge: 6

Use acknowledged techniques for teaching in order to make sure the concepts are retained properly.

Power: 8 Introducing a concept can give very different results depending on how the concept is introduced. Therefore, designing the introduction based on clear guidelines can provide much better results.

Cost: 0 The gain/cost ratio is low, as the benefits of this can be really high, while the cost is very low.

Risk: 1 The biggest risk for researching teaching techniques is that no technique will be useful and the research will be wasted. This does not seem that problematic.

Time: 1 The time is mostly related to doing the research before the system is designed. This should take at most one work day, and be fairly superficial.

Games or tutorials: 1

Develop small games and tutorials for the different concepts.

Power: 10 Games and tutorials are great for learning.

Cost: 2 This could require quite a lot of effort.

Risk: 3 It is very unclear if I will be able to be successful in this, and I have very little experience.

Time: 4 This will probably take a lot of time.

Quiz techniques: 2

Used to uncover whether or not students have understood concepts properly.

Power: 8 This can be very useful, both for the teacher in order to identify what he needs to focus on, and for students in order to learn whether or not they have properly understood a concept.

Cost: 1 The gain/cost ratio seems low.

Risk: 2 It is really important that a quiz is properly implemented, else it will not bring any value to the solution.

Time: 3 This requires time for both research, design and implementation.

Idea generation techniques: 0

In order to help students understand the brainstorming activities. **Power**: 5 As it can be difficult to generate ideas, especially for the ecology objects and leverage points, it would be useful for students to have some help for this. However, it is not clear how much value this would really add.

Cost: 2 As it is doubtful how much this will bring to the solution, the gain/cost value is not low.

Risk: 1 This would likely be possible to implement

Time: 2 This would take some time

Chat robot: -5

That can ask questions about ideas in order to mature the idea.

Power: 6 Given a great implementation this would probably be helpful.

Cost: 4 A good implementation of this could be really expensive.

Risk: 4 It is likely that I will not be able to implement this in the foreseeable future.

Time: 3 For any useful version of this, both research, design and implementation would take a lot of time that I don't have.

Personal computer: 8

For users to interface with the system.

Power: 10 This is probably the most powerful platform option, as all developers own a PC and it is a great option.

Cost: 0 As this is a great platform to use for development, the gain is very high compared to the cost.

Risk: 0 There is no real risk.

Time: 2 Developing an interface takes some time.

Cellphone: 3

For users to interface with the system. Notebook, quick ideas.

Power: 8 A cellphone would be a great interface for quizzes and for entering quick ideas that developers have while off work. However, it is not great for diagrams and scenarios, but can be used to supplement the PC.

Cost: 2 As there would almost certainly have to be developed a PC interface with or without a cellphone interface, and development can take time. However, it might be possible to develop a common design using web framework.

Risk: 1 As cellphones have very different formats, it can be difficult to make a design that fits many. **Time**: 2 This can take some extra time.

Earlier solutions and projects: 3

Repository of earlier solutions and projects that the students can use to gain inspiration.

Power: 7 Getting inspired by previous project is a great way of getting ideas and understanding the framework

Cost: 0 The cost is very low.

Risk: 3 There is no guarantee that earlier solutions are correct, and they might confuse more than help. Also, it is not guaranteed that students will want to make their work available.

Time: 1 This should not take much time.

A.4 Extracted features for sprint 1

The first sub-feature I identified are *containers* for each category where students can write text. This is the essential feature, as this should be present in any reasonable solution. The category items should be divided into *separate units* so it is easy for the students to get an overview of which items the category contains. While working with the ESSENCE methodology, Something I have learned while using the ESSENCE methodology is, that it is important to note how the different categories and category items are connected. E.g., each feature should be supported by a scenario and a value proposition, and each scenario should be supported by at least one feature. Therefore it would be useful to be able to *connect* the category items to each other across different categories. Additionally, the students should be *made aware* if an item is not supported the way it should be. This could help to get an overview of the solution. However, as the categories are also very useful as brainstorming tools, the table could become bloated if the students are made aware of any items that are not connected, even if it is just early ideas. Therefore it should be possible to have *active and inactive* items, so that active items are items that the students are actively including in the sprint, while inactive items are used for brainstorming. Additionally, the students should be able to *maintain* configuration tables across sessions, and across multiple users. Sharing the progress across multiple users is related to multi user synchronization. This is a different feature that will probably require a great deal of work and thought, so this is not included in the sub-feature list. Another feature that could be nice to have is sub-items, as I find them quite useful in this project. Additionally it should be possible to change the ordering of the items, as this can be useful in brainstorming. The feature category can be used as a backlog for the project, and during the course of a project, this and several other categories can be filled with a large number of items, some active, some inactive and some finished. However, just deleting finished items would make the configuration table not reflect the actual project, but maintaining all items can make the table unmanageable. Therefore it should be possible to keep track of the item *status*. Additionally, the user should be able to add, delete and edit items.

A.5 Sprint 1 evaluation comments

This section contains all comments from myself and Ivan on the sprint 1 resulting product.

Comments from Ivan Aaen:

- There should be a fourth row for the process view¨
- Each row could have a green or blue border, conceptually connecting the items of that row
- As each category can potentially contain an unlimited amount of items, it can become cluttered with too many items.
- Items should be italic when they are not part of the current scope but only maintained.
- The Strategy, Rational, and Tactics connection is not visible
- When a category is highlighted, irrelevant categories should be dimmed

Thoughts from using the tool:

- It would be nice if categories with only one item filled the entire box.
- The box order sometimes jumps
- The criteria should be visible
- There should be some help for updating the boxes
- There could be some kind of acceptance mark for items that live up to the criteria so that it is easier to identify items that have not yet been checked
- The text size for the category title should scale
- Problem and warrant should be directly connected
- Challenge rarely changes. As this is the first category that will be filled out, it should be less visible, as it draws attention from what is actually being filled out right now.
- The Clear connection between challenge and problem should be very clear. Maybe they should be one unit.
- Each item should have the possibility for a longer description, some kind of reasoning.
- Each category should have some description to help the thought process.
- It should be possible to save the state of a configuration table in a readable format for documentation in reports
- Maybe the Architecture would benefit from some drawing
- Maybe pictures would be useful in elements
- Some of the categories are fundamental for making sure that the solution is actually a good idea, but when filled out, they appear in the way of other categories.

A.6 Sprint 2 RST Review

Part 1 of the review was to get feedback from Ivan Aaen. Here is his comments:

- It would be a good idea to remind people of the three levels(strategy, rationale and tactics).
- It would be nice with a two way interface with Trello, where subfeatues are shown when holding the mouse over a category item.
- It would be nice if the subfeatures from Trello included development status.

- Likes the idea of questions before an RST Review that asks how users intend of living up to criteria.

For part 2, I again used the tool for the RST review and not my comments:

- The boxes are not stretched evenly. One box is much longer than others on each row in certain formats.
- Some sub category titles tend to disappear
- Boxes jump around when clicked on, it is really annoying
- It is really difficult to use the checklist
- There is no real place to begin
- The configuration table quickly become cluttered with many items. Especially for the features, it is difficult to really visualize how the items are connected.
- I am not completely sure how categories are and should be connected, and how to use the criteria properly.
- For each RST Review, you should be able to comment on how the criteria are met.
- There are many categories, so things become so cluttered. Some of the categories could be collapsed or reviewed first to get out of the way
- Besides the exit criteria, there should be multiple criteria for the category items
- There should be some kind of help button when user get lost.

A.7 Sprint 3 feature extraction

For sprint 3 feature extraction I extract sub-features from this feature:*Project starting at the challenge definition and progressing through all development stages.*

If the tool should reflect project progression though the development stages, the solution should support the following activities.

1. Challenge initiation
2. Ecology object generation and filtering
3. Leverage point generation and filtering
4. Defining initial problem
5. Filling out prospect scenarios and selecting quadrant

6. Filling out representational forms and expanding on them in order to get initial configuration table
7. Updating the configuration table during RST reviews.

The solution should carry over the progress between activities.

These stages are mostly features by themselves too, but this feature is mostly concerned with the overall flow between the features

1. Fill out challenge at project beginning
2. Update project status at each activity
3. Carry over project status from former stage
4. See project status
5. See current stage
6. See stage map

A.8 Sprint 3 RST Review

Comments from Ivan: The comments are divided into the page they concern.

Challenge definition.

- Does not think that developers should be forced to use ESSENCE in any specific way or do the activities in any specific sequence. Therefore the relatively locked sequence in the application suggest should be more open.

Ecology objects:

- The objects should be ranked by total
- It could auto select the highest ranking objects
- Negative values should be red, this is very important

Leverage points:

- Same as with ecology objects

Initial problem definition:

- Ivan did not see the selected column right away. This indicates that the column should be more visible
- Should not be called selected challenge, as the challenge is not something that is selected

- Ivan debated whether warrant should be yellow or green. Decided on yellow.

Scenario Axis Definition:

- Ivan was skeptical about the fact that users have to define the axis before entering representation

Representation definition:

- The quadrants should be named
- Representations are only text. It should be possible to select pictures for icons instead.

Representation expansion:

- Very filled
- Checkbox is not intuitive
- Icon should be to the right instead of on bottom (I fixed this before making screenshot)

Feature selection:

- Too many boxes without content when many features are selected

We only quickly went through the last screens as they are largely the same as last time. Therefore I did not note any comments for these.

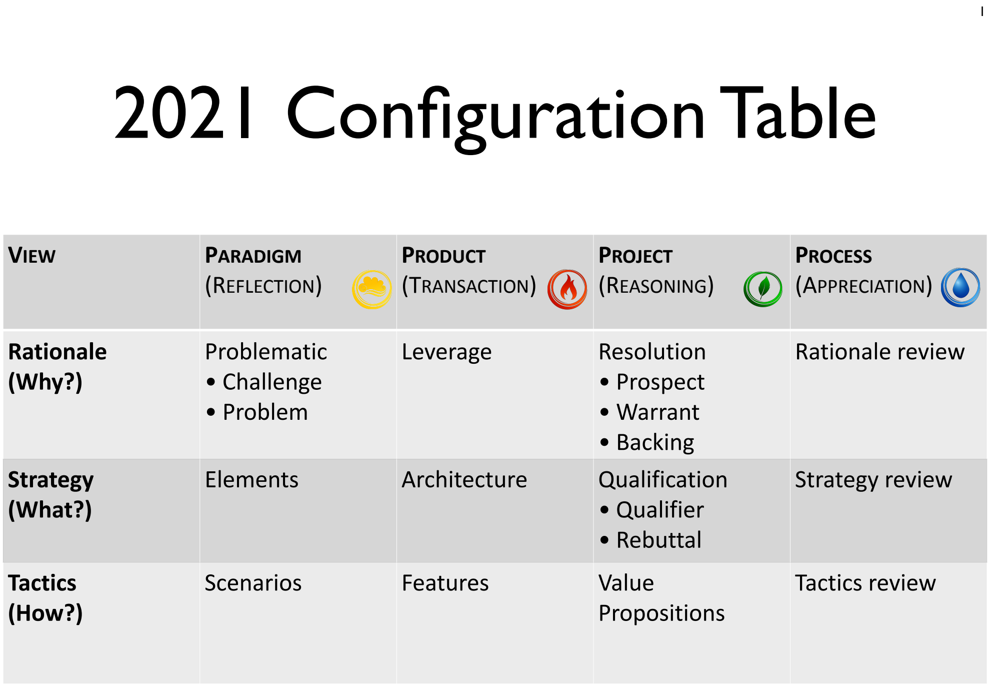
Part 2:

- Comments page is not very usable at this point. Could be great if multiple lists could be made e.g., one for each page, and if we could automatically generate bugfix for a comment.
- It is very difficult to know how to get started with the diagram update. It would be very helpful with some assistance.
- The diagram overview has too little space for each box, so it it difficult to get an overview. The print version actually gets a way better overview. Maybe I should use a design that is more like that.
- Editing a category item should force you to consider all related fields again
- The connection between items should be clear
- Maybe the design choices should be clear?
- Maybe we should define some questions that are useful in most development projects

- The architecture category is still strange. It should really be updated to reflect the way students currently design architecture.
- Maybe all categories in ESSENCE are really related to some commonly used software development practices. Like feature is related to the backlog, architecture is related to the design of a system architecture... This should be explored!
- Ecology and leverage should be somehow directly connected with the ecology objects and leverage points tables, and it should be possible to do a SWOT and PCRT analysis later on.
- It should be more clear what the purpose is with each category and how the are related
- Items in single item categories should not be deleted when empty

A.9 Updated configuration table

In the beginning of may, Aaen sent an updated layout for the configuration table, as he imagines it will be in 2022. This is displayed in A.1.



2021 Configuration Table

VIEW	**PARADIGM** (REFLECTION)	**PRODUCT** (TRANSACTION)	**PROJECT** (REASONING)	**PROCESS** (APPRECIATION)
Rationale (Why?)	Problematic • Challenge • Problem	Leverage	Resolution • Prospect • Warrant • Backing	Rationale review
Strategy (What?)	Elements	Architecture	Qualification • Qualifier • Rebuttal	Strategy review
Tactics (How?)	Scenarios	Features	Value Propositions	Tactics review

Figure A.1: Updated layout for configuration table

A.10 Configuration table update Sprint 4

I update the configuration table from sprint 35.10.

First step of the diagram update is to reflect on the blue cells in the matrix6.13. If answering any of these questions lead to a change in a category, I will follow column and investigate changes there.

Has our understanding of the domain changed? No. The domain is the same, but the wording is a little different, as the domain was called challenge before.

Have we learned something that changes the outer environment? No, there is no obvious reason to change the

Are the scenarios we described still most fitting? No, scenario 1 has changed. The scenario text is *Group of students in development team using the ESSENCE tool to guide them from the beginning of a project to the development stage and through development sprints.* This is not really what I want to do at this point. Instead, I want to help the students (1) get from an initial idea of what the project should be to a filled out configuration table, and (2) identify changes in category understanding and how that might affect the other categories.

For scenario 2, *Students using tool together with other process management tools they are used to.* , I don't really think that this is relevant for the tool at this point, as this is more related to project life cycle.

As I have changed the entire scenario category, I will now update the other categories based on this.

Do the domain still cover the scenarios? Yes.

Given the change in scenarios, do we still have the best outer environment Yes.

Are the scenarios still related to the problem? The problem *It can be difficult to understand the different activities and concepts of ESSENCE, and how these fit together with the existing understanding of a development project.* is more related to understanding concepts and visualizing than for understanding flow. Therefore this should be modified so: *It can be difficult to understand how filling out or changing categories affect other categories.*

Are the qualifier still relevant given the change in problem? The qualifier *Not all students want to use Scrum in projects.* is not related to the problem at all at this point. Therefore this should be changed. The new qualifier is *It might not be possible to create a flow that will work for all projects.*

The change in qualifier should always result in update of the rebuttal. The current rebuttal is *Scrum is a popular development framework that most students find useful* which does not answer the qualifier at all. A relevant rebuttal could be *It should be possible to create a flow structure that will help in the majority of projects. This could still be very beneficial.*

The next question I would like to answer is **Do the value propositions still make sense given the problem?**. This should be answered for each value proposition. For the first value proposition we have *Help student understand the concepts and activities of ESSENCE.* I don't think that this is really fitting at the moment, as it is mostly categories we will be explaining. However, this can be modified to work with the new problem *Help students understand how categories affect each other.* The second value proposition is *Make it clear how ESSENCE can be used to support the development process.*. This should be removed as this is not at all related to the problem any more. A new value proposition could be introduced. This is *Help students become aware of category items that has lost their value in the project.*

Do the selected features offer the value proposition? This should be answered for each feature, together with the question *Are all features relevant, for the scenarios and should we add new features to cover the scenarios?*.

- *Configuration table tool* The configuration table tool is relevant to both value propositions and both scenarios. This should therefore stay.
- *Leverage point and Ecology object[outer environment] generation and filtering.* This could still be relevant when filling out the configuration table, and when updating the categories. However, as I am removing the pre-project, this could become less relevant. However, at the moment I will keep this in the configuration table.
- *Prospect scenario tool* This could be very relevant for helping get from a few categories to a filled out configuration table, helping fulfill value proposition 1. I will also keep this.
- *Project starting at the challenge definition and progressing through all development stages* This is not relevant more.
- *Interfacing with Trello/ other Kanban tool.* This is again not relevant any more.

In order to cover all scenarios, we should add one new feature: *Category review suggestions*, where relevant categories are suggested for review when a category is updated.

Given the changes to features, should the leverage points be changed? I could go through each leverage point. However, when reading the items it was rather obvious that the changes does not affect the leverage points.

Given the changes to features, should the inner environment be changed? Looking at the inner environment it is also clear that this will not change.

Now, I have been through the flow matrix. However, there were some categories that were not included in the flow matrix. I will go through them now.

Proposition: What is the proposed solution for the problem?: The proposition (prospect) from last table was *Helps students understand how to use ESSENCE in a SCRUM development project.*. The new proposition is *A website that helps students fill out and update a configuration table.* Given a new proposition, there should be a new backing. The backing is *Helps get an overview an understand how categories are related.*

Warrant: why should we solve the problem: *Making it easier to get an overview of the configuration table and see what changes could affect will make it easier to use essence.*

A.11 Further work on the product

After sprint 5, my work on the product is done. However, that does not mean that the product will need in order to make it good. Below is a backlog of tasks that could improve the product greatly. The list is divided into areas.

SWOT and PCRT analysis

- Sort items after total value in SWOT and PCRT.
- Do not show items in the category overview if they are disabled in the SWOT or PCRT analysis
- Make item enabled + disabled in both SWOT and PCRT analysis if toggle disable button is clicked
- Restrict values to valid input values
- Make values that count negatively red

Category overview

- The empty input field should always be at the bottom of the category items
- It should be possible to move the category item order
- It should be visually clear for which categories the items should connect
- It should be clear why category items should be connected
- When a category is highlighted after a connected category is edited, there should be an explaining description of why the category should be updated. E.g., if a scenario is edited, instead of just highlighting the features category, some text should appear, stating *do the features still make sense given the change in scenarios?*. The description should be placed close to the category
- There should be a button that activates the reflect boxes
- The descriptions of the categories and the images related should be updated with relevant images and text.
- It should be possible to limit and expand category boxes
- It should be possible to exit the edit mode of a category without editing another category.

Projects overview

- It should be possible to view related users for a project

- It should be possible to rename projects
- It should be possible to reorder projects
- It should not be possible to create a project without a name
- It should not be possible to have several projects with the same name, or there should be a last accessed field to tell identically named projects apart

Backend

- When a project is deleted, it should also be properly deleted in the backend
- The way users are connected to projects currently in the backend is not good. This should be updated
- The reflection boxes and the "should reflect on" information should be stored in the database instead of the browser for multi-user synchronization and maintenance between sessions.