# BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

## FACULTY OF BUSINESS AND MANAGEMENT
FAKULTA PODNIKATELSKÁ

## INSTITUTE OF INFORMATICS
ÚSTAV INFORMATIKY


# EXPERIMENTATION IN AGILE DEVELOPMENT AND HUMAN-CENTERED DESIGN
EXPERIMENTOVANIE V AGILNOM VÝVOJI A DIZAJNE ZAMERANOM NA ČLOVEKA

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE


AUTHOR — Hugo Chrást
AUTOR PRÁCE

SUPERVISOR — Ing. Lenka Širáňová, Ph.D.
VEDOUCÍ PRÁCE



BRNO 2023

BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF BUSINESS AND MANAGEMENT

# Assignment Bachelor's Thesis

| | |
|---|---|
| Department: | Institute of Informatics |
| Student: | **Hugo Chrást** |
| Supervisor: | **Ing. Lenka Širáňová, Ph.D.** |
| Academic year: | 2022/23 |
| Study programme: | Managerial Informatics |

Pursuant to Act no. 111/1998 Coll. concerning universities as amended and to the BUT Study Rules, the degree programme supervisor has assigned to you a Bachelor's Thesis entitled:

## Experimentation in Agile Development and Human – centered Design

**Characteristics of thesis dilemmas:**

Introduction
Goals of thesis and methods
A theoretical review of a problem
Analysis of the contemporary situation
Proposal of solution
Conclusion
References

**Objectives which should be achieve:**

The aim of this thesis is to use knowledge and theoretical insights from project management, experimentation and human–centred design.

**Basic sources of information:**

ASHMORE, S. and K. RUNYAN. Introduction to Agile Methods. Addison-Wesley Professional, 2014. ISBN 978-0-321-92956-3.

COLE, R. and E. SCOTCHER. Brilliant Agile Project Management: A Practical Guide to Using Agile, Scrum and Kanban. Pearson, 2015. ISBN 978-1-292-06358-4.

DOLEŽAL, J. a kol. Projektový management: Komplexně, prakticky a podle světových standardů. Praha: Grada Publishing a.s., 2016. ISBN 978-80-247-5620-2.

MARTIN, R. Agile Software Development- Principles, Patterns and Practices. USA: Prentice Hall, 2003.

STAPLETON, J. DSDM: Business Focused Development. Pearson Education, 2003. 2nd edition. ISBN 9780321112248.

Deadline for submission Bachelor's Thesis is given by the Schedule of the Academic year 2022/23

In Brno dated 5.2.2023

L. S.

Ing. Jiří Kříž, Ph.D.
Branch supervisor

doc. Ing. Vojtěch Bartoš, Ph.D.
Dean

## Abstract

This Bachelor thesis compares linear and iterative project management methodologies and explores the use of human-centered design principles in software development. It analyses the application of Experiment Canvas frameworks in the development process of RedHat Company's Agility & Continuous Improvement Team. Based on the research findings, the thesis proposes the adoption of Experiment Canvas frameworks tailored to the needs of RedHat Company. The study demonstrates that Experiment Canvas implementation can improve the efficiency and effectiveness of software development practices at RedHat.

## Abstrakt

Táto bakalárska práca porovnáva lineárne a iteračné metodiky riadenia projektov a skúma využitie princípov dizajnu zameraného na používateľa pri vývoji softvéru. Analyzuje použitie Experiment Canvasu v procese vývoja tímu Agility & Continuous Improvement spoločnosti RedHat. Na základe výsledkov výskumu práca navrhuje prijatie Experiment Canvasu prispôsobených potrebám spoločnosti RedHat. Štúdia dokazuje, že implementácia Experiment Canvasu môže zvýšiť efektívnosť a účinnosť postupov vývoja softvéru v spoločnosti RedHat.

**Key words**

Experimentation, Human-centered Design, Agile Development, Project Management, Experiment Canvas

**Klíčová slova**

Experimentovanie, dizajn zameraný na človeka, agilný vývoj, projektový manažment, Experiment Canvas

**Rozšírený abstrakt**

Táto bakalárska práca poskytuje komplexné preskúmanie metodík riadenia projektov, konkrétne porovnanie lineárnych a iteračných procesov a skúma uplatňovanie princípov dizajnu zameraného na používateľa pri vývoji softvéru. Primárnym cieľom tejto práce je analyzovať súčasnú situáciu v spoločnosti RedHat s konkrétnym zameraním na jej tím Agility & Continuous Improvement a využitie Experiment Canvas v procese vývoja softvéru.

Na dosiahnutie tohto cieľa sa v tejto štúdii využíva teoretický prehľad metodík riadenia projektov vrátane hĺbkovej analýzy vodopádového modelu a agilných prístupov k vývoju. Okrem toho práca poskytuje rozsiahle preskúmanie princípov dizajnu zameraného na používateľa vrátane použiteľnosti a rámca Experiment Canvas.

Analýza súčasnej situácie spoločnosti RedHat zahŕňa prehľad histórie, cieľov a obchodných činností spoločnosti. Okrem toho táto práca skúma tím Agility & Continuous Improvement v spoločnosti RedHat a spôsoby, akými do svojej práce začleňuje agilné metodiky vývoja a rámce Experiment Canvas.

Na základe výsledkov výskumu táto práca navrhuje riešenia na zlepšenie procesu vývoja softvéru v spoločnosti RedHat. Navrhované riešenie využíva rámec Experiment Canvas, špeciálne prispôsobený potrebám spoločnosti RedHat.

V závere táto práca poskytuje komplexné preskúmanie metodík projektového riadenia, princípov dizajnu zameraného na používateľa a ich aplikácií pri vývoji softvéru. Výsledky výskumu dokazujú, že implementácia rámca Experiment Canvas do procesu vývoja softvéru môže zlepšiť celkovú efektívnosť a účinnosť vývojových postupov spoločnosti RedHat.

BIBLIOGRAPHIC CITATION OF THE THESIS

CHRÁST, Hugo. *Experimentation in Agile Development and Human – centered Design* [online]. Brno, 2023 [cit. 2023-05-15]. Available at: https://www.vutbr.cz/studenti/zav-prace/detail/152607. Bachelor's Thesis. Brno University of Technology, Fakulta podnikatelská, Ústav informatiky. Supervisor Ing. Lenka Širáňová, Ph.D.

**Affidavit**

I declare that the present bachelor project is an original work that I have written myself. I declare that the citations of the sources used are complete, that I have not infringed upon any copyright (pursuant to Act. no 121/2000 Coll.).

Brno dated 15th May 2023

Hugo Chrást

author's signature

**Acknowledgements**

I would like to express my sincere gratitude to my thesis supervisor Ing. Lenka Šiřáňová, Ph.D. for her valuable guidance and advice that she has provided throughout the writing process of the thesis. Also, I would like to say thank you to all my friends and family members that gave me support during whole my studies.

# CONTENTS

# INTRODUCTION

Project management plays a crucial role in today's fast-paced and dynamic business environment. It provides a structured approach to planning, executing, and controlling projects to achieve specific objectives within defined constraints. Effective project management ensures that projects are completed on time, within budget, and to the satisfaction of stakeholders. This introduction will discuss the importance of project management and provide an overview of the bachelor thesis's focus.

The significance of project management stems from the complex nature of modern projects. Organizations across various industries undertake projects to implement new systems, develop innovative products, or improve existing processes. However, without proper management, projects can become chaotic, leading to missed deadlines, budget overruns, and unsatisfactory outcomes.

Project management practices help in mitigating these risks by providing a systematic framework for project planning, resource allocation, and risk management. It ensures that project objectives are clearly defined, roles and responsibilities are assigned, and progress is monitored regularly. By coordinating and aligning various project activities, project management fosters effective communication, collaboration, and coordination among team members.

The bachelor thesis aims to explore key concepts and methodologies related to project management and their application in real-world scenarios. It consists of two main sections: a theoretical review of relevant topics and an analysis of a contemporary situation.

The theoretical review section (Chapter 1) provides a comprehensive understanding of project management and its methodologies. It begins with an overview of project management (Section 1.1), highlighting its fundamental principles and benefits. It then delves into the comparison between linear and iterative processes (Section 1.2), discussing the traditional Waterfall model (Section 1.2.1) and the more adaptive Agile development approach (Section 1.2.2). Additionally, the importance of human-centered design is explored, focusing on usability (Section 1.3.1) and the Experiment Canvas framework (Section 1.3.2).

The analysis of a contemporary situation (Chapter 2) centers around the RedHat Company. This section aims to evaluate a specific project undertaken by the organization and analyse its project team's dynamics. By utilizing the Experiment Canvas framework, the thesis explores how the project team effectively managed the project, addressing challenges and ensuring successful outcomes.

Through the examination of theoretical concepts and their practical application in a real-world context, this bachelor thesis seeks to contribute to the existing knowledge and understanding of project management principles and their relevance in contemporary business environments. By doing so, it aims to provide valuable insights and recommendations for future project management endeavours.

# GOALS OF THESIS AND USED METHODS

The aim of this thesis is to use knowledge and theoretical insights from project management, experimentation and human-centred design.

Abstraction Laddering:

The first method utilized in this thesis is abstraction laddering. Abstraction laddering is a technique widely used in human-centered design that facilitates a deeper understanding of the problem by gradually uncovering the underlying motivations and goals. This method involves systematically breaking down the problem into smaller components, identifying their interconnected relationships, and exploring the underlying factors driving the issue at hand. Through abstraction laddering, the thesis aims to gain insights into the fundamental aspects of the problem and create a solid foundation for subsequent research and experimentation.

Experiment Canvas:

The second method employed in this research is the experiment canvas. Experimentation is a crucial aspect of project management and human-centered design, as it allows for iterative testing and validation of proposed solutions. The experiment canvas is a visual tool that helps outline and structure experiments effectively. It enables researchers to define clear objectives, identify target metrics, design the experiment methodology, and determine the key success criteria. By utilizing the experiment canvas, the thesis aims to develop and execute a series of experiments to test and refine potential solutions, ensuring their feasibility, usability, and effectiveness.

In summary, this bachelor thesis aims to leverage knowledge and theoretical insights from project management, experimentation, and human-centered design to address a specific problem. By employing abstraction laddering and the experiment canvas, the research strives to achieve a comprehensive understanding of the problem, propose effective solutions, and provide practical insights and recommendations for future application. Through this interdisciplinary approach, the thesis aims to contribute to the fields of project management, experimentation, and human-centered design, promoting innovation and enhancing problem-solving capabilities.

# 1 THEORETICAL REVIEW OF A PROBLEM

The chapter describes the theory framework of this bachelor thesis. It serves as a theoretical foundation for the field of agile approach in project management, experimentation in agile development and human-centered design.

## 1.1 Project management

A project management process is necessary to ensure that the engineering process ends up meeting the real-world objectives of cost, schedule, and quality. The project management process specifies all activities that need to be done by the project management to ensure that cost and quality objectives are met. Its basic task is to ensure that, once a development process is chosen, it is implemented optimally. A proper management process is essential for success [1].

The activities in the management process for a project can be grouped broadly into three phases:

1. **Planning** – initial phase which is perhaps the most critical project management activity. Project planning is the single most important management activity, and it forms the basis for monitoring and control. A plan is usually produced before the development activity begins and is updated as development proceeds and data about progress of the project becomes available [1].
2. **Monitoring and control** – the longest phase in terms of duration. It encompasses most of the development process. It includes all activities the project management has to perform while the development is going on to ensure that project objectives are met and the development proceeds according to the developed plan (update the plan, if needed). The major driving forces, most of the activity of this phase revolves around monitoring factors (cost, schedule, and quality) that affect these. Monitoring potential risks for the project, which might prevent the project from meeting its objectives, is another important activity during this phase. If the objectives may not be met, necessary actions are taken in this phase.

In iterative development, this analysis can be done after each iteration to provide feedback to improve the execution of further iterations [1].

3. **Termination analysis** – the last phase of the management process is performed when the development process is over (monitoring and control last the entire duration of the project). The basic reason for performing termination analysis is to provide information about the development process and learn from the project in order to improve the process. This phase is also often called postmortem analysis. In iterative development, this analysis can be done after each iteration to provide feedback to improve the execution of further iterations [1].

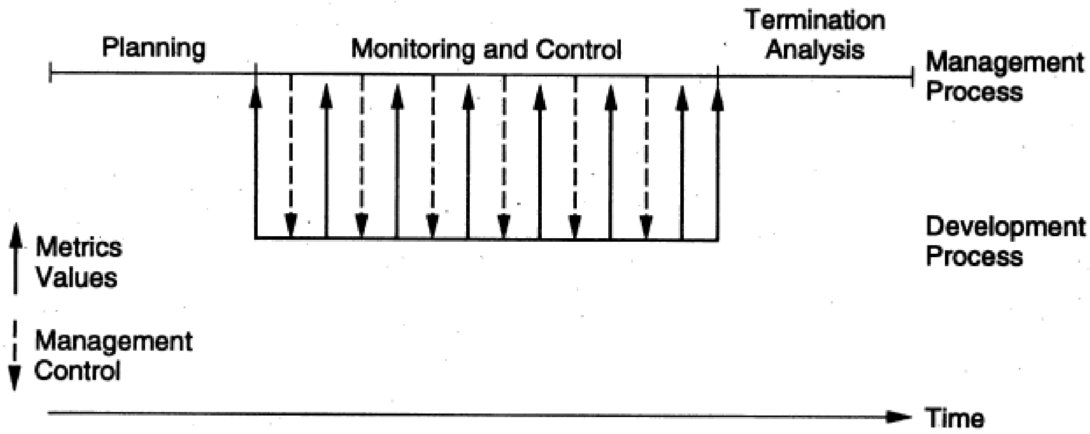



**Figure 1 Idealized temporal relationship between development and management process**

(Source: Jalote, 2008)

An idealized temporal relationship between the management process and the development process is shown in Figure 1. The figure shows that planning is done before development begins, and termination analysis is done after development is over. During the development, from the various phases of the development process, quantitative information flows to the monitoring and control phase of the management process, which uses the information to exert control on the development process [1].

## 1.2 Linear versus iterative process

The differences between linear and iterative process are illustrated in Figure 2:

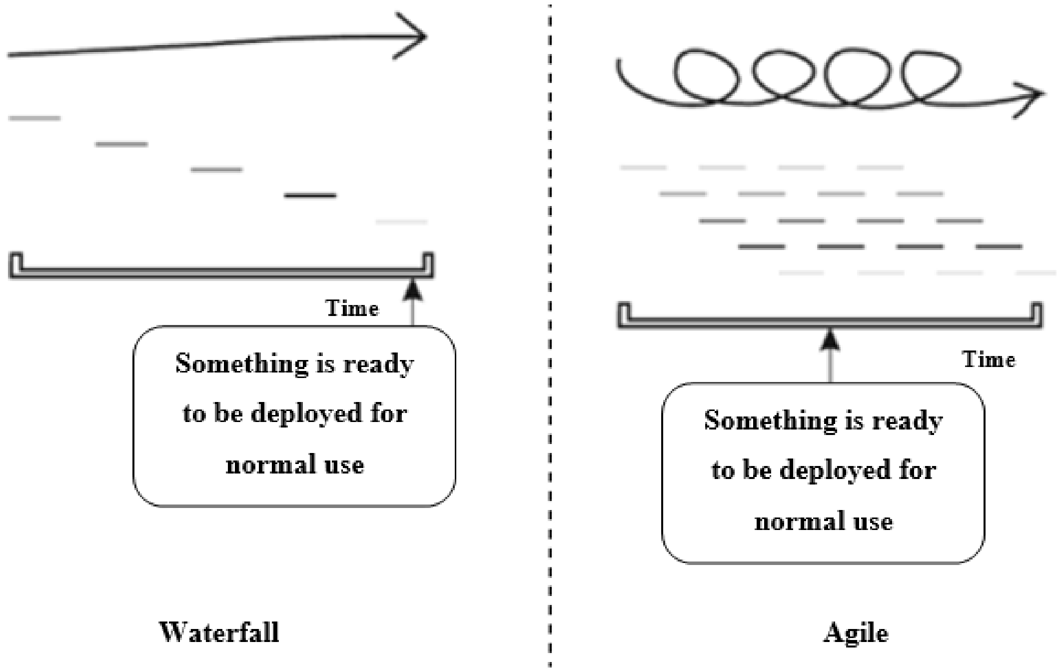



**Figure 2 Linear vs iterative process**

(Source: Doležal 2016, edited by author)

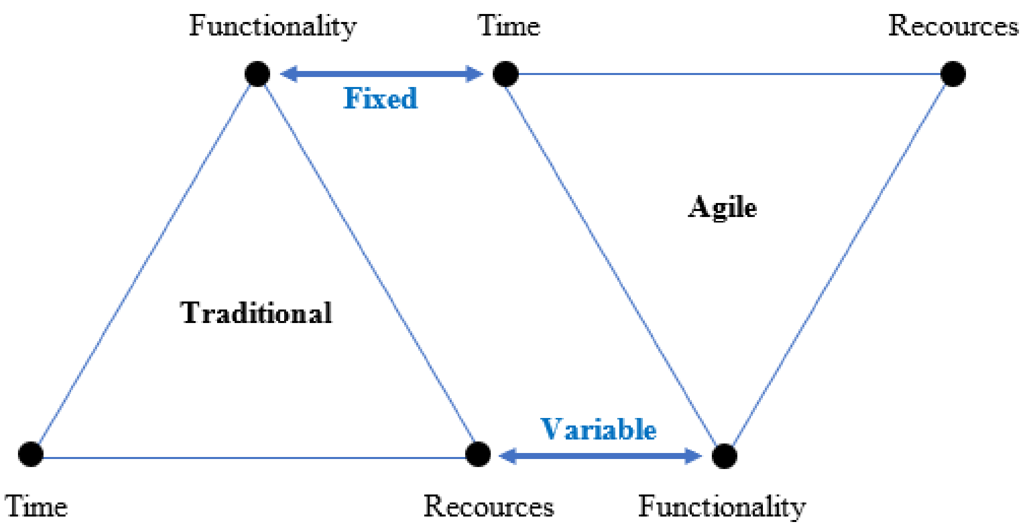



**Figure 3 Comparison of traditional and agile methodologies**

(Source: Pejchal 2015, edited by author)

Traditional and agile methodologies still differ in the variables of the triple-imperative. Traditional project management initially defines a set of requirements (scope) that it considers to be immutable. According to these, it estimates the time and cost required for implementation. Agile project management, on the other hand, considers time and resources as fixed and the variable is the scope, which is adapted to the customer's priorities [2].

### 1.2.1 Waterfall model

Waterfall is one of the traditional methods of project management. When using it, it is assumed that at the beginning of the project we know the requirements well and at the same time these requirements are stable.

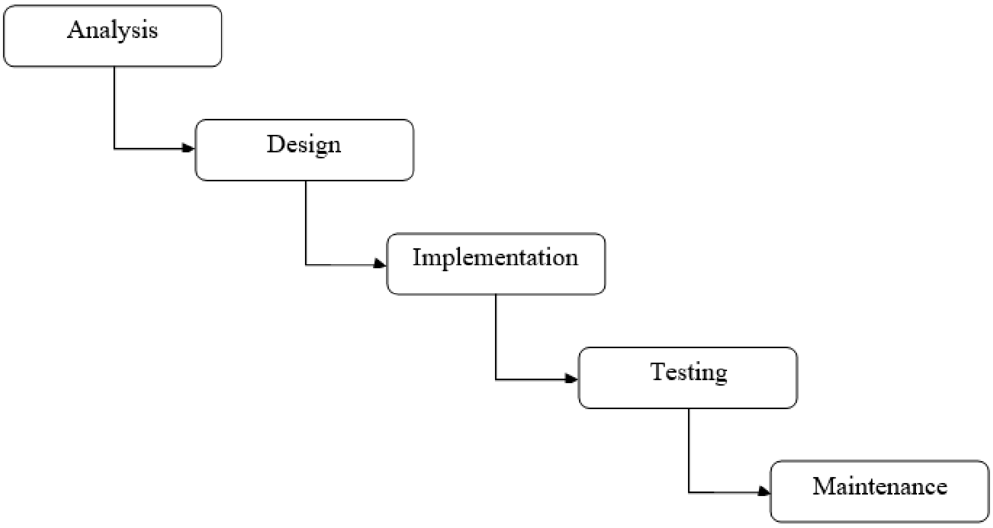



**Figure 4 Waterfall model**

(Source: Doležal, 2016, edited by author)

The waterfall method is based on five phases:

1. Analysis,
2. Design,
3. Implementation,
4. Testing and
5. Maintenance,

that each phase depends on the previous phase being completed before the next phase is started [3].

**Analysis phase** (also called the software requirement specification (SRS)): In this phase a complete and comprehensive description of both the functional and non-functional requirements are conducted. Usually, the method for describing the functional requirements is by use of user stories, where the user's interaction with the software is described. One of the outcomes from this phase is the requirements document [3].

**Design phase**: In this phase the design for the whole product is conducted through planning and problem solving. For example, the plan contains solutions for the software architecture design, database conceptual scheme, graphical user interface design and data structure definition. One of the outcomes from this phase is the design document [3].

**Implementation phase**: In this phase the real software code is written and compiled into the product. In other words, this is the process of converting all the requirements and blueprints into a product. The outcome of this phase is the source code [3].

**Testing phase**: In this phase the outcome of the previous phase is tested, i.e., the source code. The purpose of the testing phase is to debug the source code and to correct the bugs that were found. This phase is also known as the verification and validation phase because it tests if the requirements are fulfilled in the product. The outcome of this phase is for example the test document [3].

**Maintenance phase**: This phase is the process of modifying the product (source code) after delivery. The maintenance includes for example refinement, correction of errors and quality improvement of the code [3].

### 1.2.2 Agile development

Agile development approaches evolved in the 1990: s as a reaction to the more documents – driven methods, especially the waterfall approach [1]. The term agile development is an umbrella term for different kinds of agile methods (Figure 5) [6].

Agile methods have the characteristics that they are iterative, focus on teamwork, collaboration between customer and development team and also feedback from customer throughout the development lifecycle. The key difference between agile development and more heavily-driven development is that in agile development things like requirements and solutions to different problems will evolve through the development process [6].

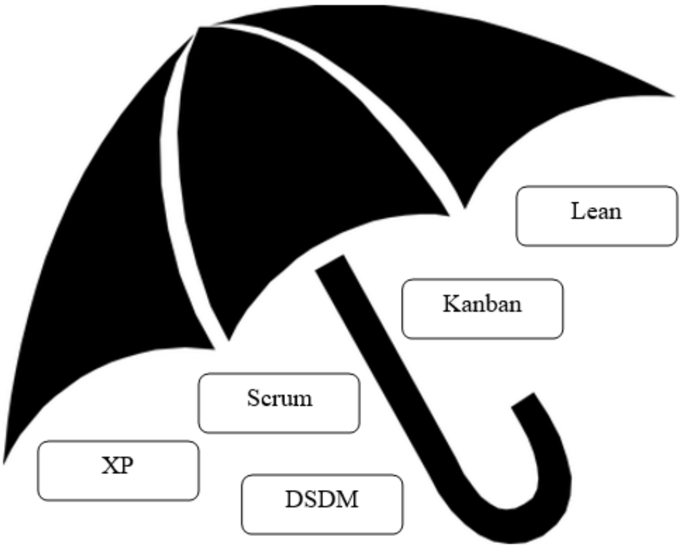



**Figure 5 Agile umbrella**

(Source: Cole 2015, edited by author)

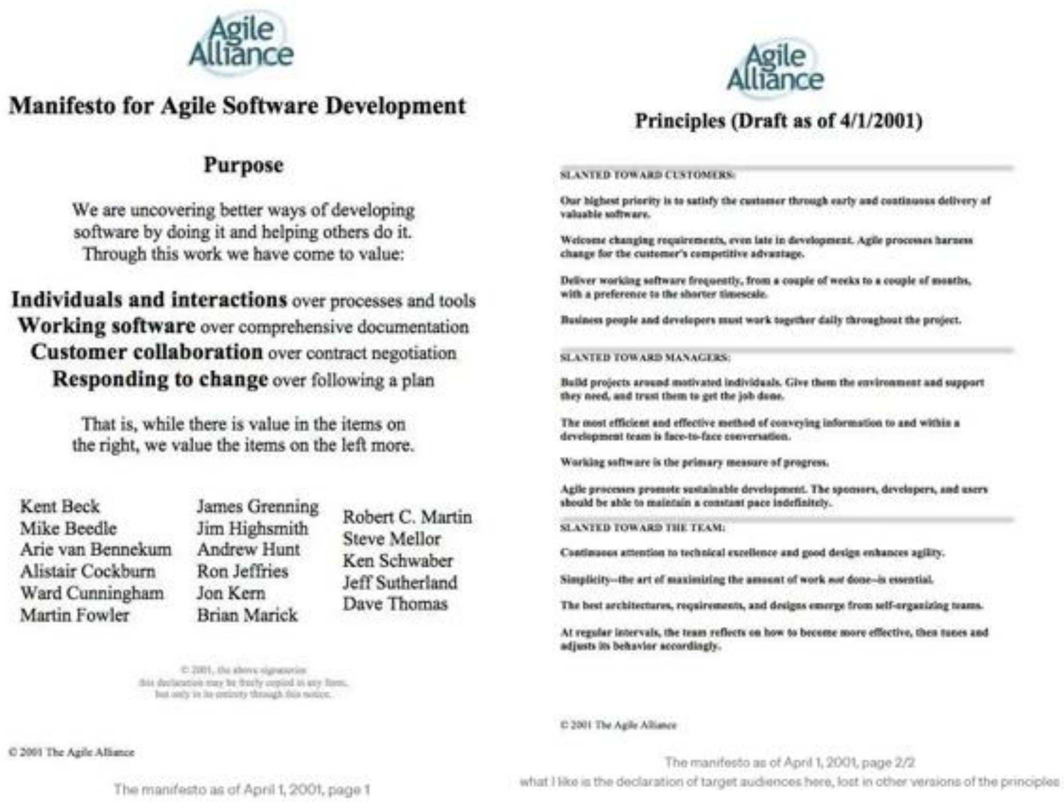
#### 1.2.2.1 Agile manifesto

In the beginning of 2001, the common characteristics for agile development was formed in the Agile Manifesto to better understanding what agile development is about. The Agile Manifesto was established at an informal meeting in Utah, USA, where people from the industry representing the different agile development methods were represented. The purpose was to discuss an alternative way of developing software besides the heavy document–driven methods, like the Waterfall model. The result of the meeting was the *The Manifesto of the Agile Alliance* (Figure 6) and a formation of a new organization, the Agile Alliance [4].

The participants found consensus around four key values. Through this work they have come to value:

1. Individuals and interactions over processes and tools.
2. Working software over comprehensive documentation.

3. Customer collaboration over contract negotiation.
4. Responding to change over following a plan [6].



Agile Alliance

**Manifesto for Agile Software Development**

**Purpose**

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

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

That is, while there is value in the items on
the right, we value the items on the left more.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick
Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas



The manifesto as of April 1, 2001, page 1

Agile Alliance

**Principles (Draft as of 4/1/2001)**

SLANTED TOWARD CUSTOMERS:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

SLANTED TOWARD MANAGERS:

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

SLANTED TOWARD THE TEAM:

Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work *not* done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.



The manifesto as of April 1, 2001, page 2/2
what I like is the declaration of target audiences here, lost in other versions of the principles

**Figure 6 The Manifesto of the Agile Alliance**

(Source: Martin 2003, s. 4)

To further explain the values behind the manifesto, twelve principles were outlined [4]:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer´s competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Businesspeople and developers must work together daily throughout the project.

5. Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.
6. The most efficient and effective methods of conveying to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity – the art of maximizing the amount of work not done – is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly [4].

#### 1.2.2.2 Scrum

According to the Webster dictionary definition, a "scrum" is a rugby play in which the forwards of each side come together in a tight formation and struggle to gain possession of the ball. Rugby is a game that cannot be won by a single superstar; it takes a full closely working team, to be successful. The same rule applies to Scrum software development. Each member of the Scrum team is vitally important and must contribute if the team wants to be successful [7].

Scrum emphasizes the importance of organizing a project into specific durations, known as time boxes, that helps the team to know what they need to focus on each day and encourage a sense of urgency. Work requirements are grouped into chunks of work that can be completed in one- to six-week time frames called sprints. All design, development, test, and customer validation work is contained within the sprint; the goal is that by the end of the sprint, the new functionality is ready to deliver to customers [7]. The product owner selects work requirements for each sprint. The product owner maintains a prioritized list of requirements for the product called a product backlog.

The requirements are written in the form of user stories or stories about the problem that needs to be solved by the requirements [7].

Tracking is also an important part of the Scrum approach. Scrum teams meet daily in what is commonly referred to as a daily stand-up or daily Scrum meeting. A Scrum master is assigned to lead these meetings to ensure that they remain brief and focused and that all team members have a chance to contribute. These meetings are designed to be short (usually around 15 minutes), and each team member answers the following questions:

- What have I done since yesterday?
- What am I doing today?
- Any roadblocks?

The main point of these daily stand-ups is to get rid of anything that gets in the way of completing tasks and projects in the manner and by the time they are needed.

The team also uses burn down or burn up charts to track progress through the sprint [7].

#### 1.2.2.3 Kanban methodology

Kanban (Japanese for signal card) is a method that is an alternative or complement to Scrum. Like Scrum, it meets the three factors for success: focus, prioritise and align workload with capacity. Unlike Scrum, which divides time into iterations, Kanban sees development as a continuous flow of tasks, just like Lean. While Scrum limits the workload by limiting how much work is done in an iteration, Kanban limits the maximum number of things that can be worked on in parallel [9].

Kanban limits the number of tasks developers work on simultaneously. A development department can decide that they only work on two things at the same time. Nothing new may be started before something ongoing is completed. There is a constant flow of tasks to be completed [9].

Kanban does not set a time limit for how long one thing can be worked on. What Scrum and Kanban both have in common is the board that visualizes the work in progress [9].

### 1.2.2.4 Dynamic Systems Development Method (DSDM)

DSDM was first introduced in 1994 as a framework to provide more structure to rapid application development (RAD) [7].

The DSDM development lifecycle has five phases and known as "the three pizzas and a cheese" (Figure 7). The forward path follows the dark arrows, and recognized routes back to evolve the system are shown by the lighter arrows. The DSDM lifecycle has seven phases:

1. Pre-project;
2. Feasibility study;
3. Business study;
4. Functional model iteration;
5. System design and build iteration;
6. Implementation;
7. Post-project [8].

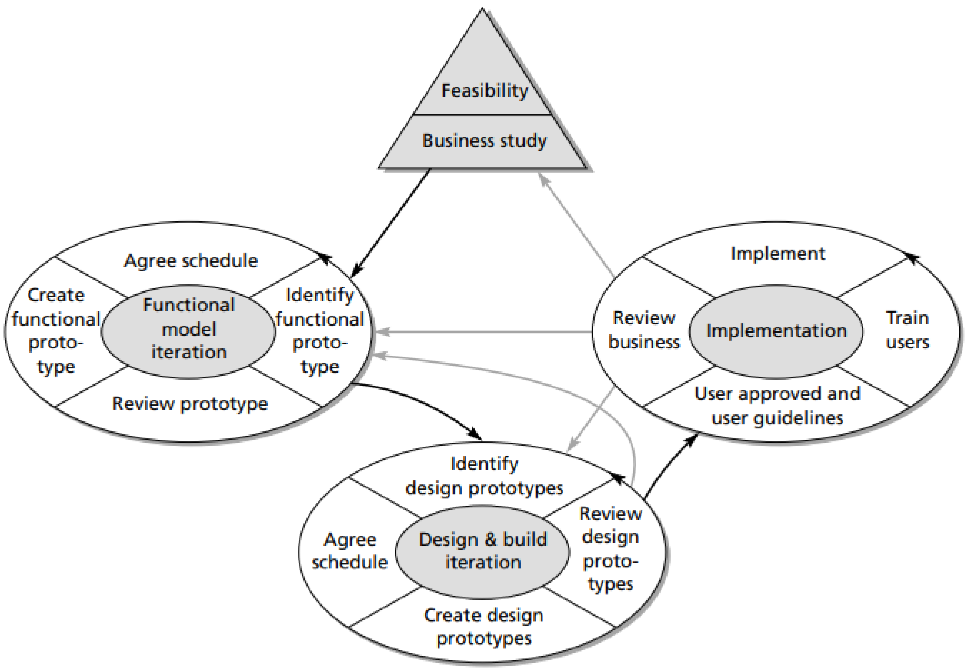



**Figure 7 DSDM process diagram**

(Source: Stapleton 2003, s. 4)

#### 1.2.2.5 Extreme Programming (XP)

The Practices of Extreme Programming Extreme programming is the most famous of the agile methods. It is made up of a set of simple, yet interdependent practices. These practices work together to form a whole that yields better greatness than its parts [7].

**Customer Team Member**

It is necessary that the customer and developers to work closely with each other so that they are both aware of each other's problems and are working together to solve those problems.

The customer of an XP team is the person or group who defines and prioritizes features. In an XP project, whoever the customers are, they are members of, and available to, the team [4].

**User Stories**

In order to plan a project, it is needed to know something about the requirements, but it isn't mandatory to know every detail.

We need to have some sense that there are details, and we must know roughly the kinds of details that will be involved, but we don't have to get to the specifics. When using XP, we get the sense of the details of the requirements by talking them over with the customer, but we do not capture those details.

A user story is a planning tool that the customer uses to schedule the implementation of a requirement based upon its priority and estimated cost [4].

**Short Cycles**

An XP project delivers working software every two weeks. Each of these two-week iterations produces working software that addresses some of the needs of the stakeholders. At the end of each iteration, the system is demonstrated to the stakeholders in order to get their feedback [4].

**Acceptance Tests**

The details about the user stories are captured in the form of acceptance tests specified by the customer. Together, they act to verify that the system is behaving as the customers have specified. Once an acceptance test passes, it is added to the body of passing acceptance tests and is never allowed to fail again. This growing body of acceptance tests is run several times per day, every time the system is built. If an acceptance test fails, the build is declared as a failure. The system migrates from one working state to another and is never allowed to be inoperative for longer than a few hours [4].

## 1.3 Human-centered design

The International Organization for Standardization (ISO) in standard 9241-210:2010 says that Human-Centered Design (HCD) is *"characterized by: the active involvement of users and a clear understanding of user and task requirements; an appropriate allocation of function between users and technology; the iteration of design solutions; multidisciplinary design."* [12].

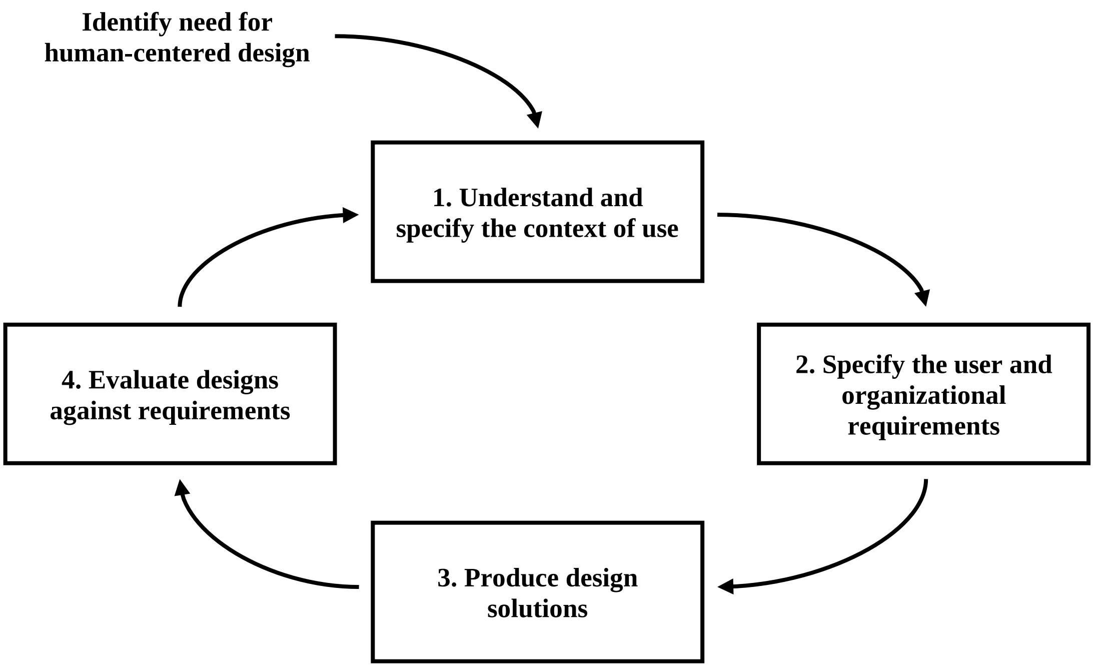



**Figure 8 The human-centered design**

(Source: Seffah, 2006, s. 115, adapted by author)

HCD represents the techniques, processes, methods, and procedures for designing usable products and systems, but just as important, it is the philosophy that places the user at the centre of the process [10].
In HCD, the development of a product or service, the focus lies on the user and the entire experience during the products or service's lifecycle. HCD shares the iterative design thinking of Agile development processes [10].

### 1.3.1 Usability

There are several ways to ensure that a system and its user interface (UI) is usable. One of the most commonly known methods is usability testing. However, there are several more tools with their own strengths and weaknesses:
Heuristic Evaluation is a discount usability engineering' method for evaluating user interfaces to find their usability problems [10].
Interviews with users is a good complement to understand their needs. Interviews can be performed by themselves or as a part of a debriefing during usability testing [10].
Surveys can be used at any time in the lifecycle but are most often used in the early stages to better understand the potential user. Asking people about what they do or have done is no substitute for observing them do it in a usability test [10].
Qualitative Usability Tests is a controlled form of testing the usability of a product or service that involves representative users that perform realistic tasks. Usage is observed and data can be gathered in different ways like note taking or video recording of the session [10].
Analysing Real Usage Statistics (with tools like Google Analytics and Google Tag Manager), real usage can be collected as statistical data or single usage. This is a good complement to qualitative usability tests [10].

### 1.3.2 Experiment Canvas

Experiment Canvas takes problems and solutions. Once the experiment trial is done, we can review it and most importantly learn from it.

Depending on the team whether it's a sprint or some kind of release, at the end of each phase we run the retrospective to figure out if things could have been set up in a better way and eventually learn from the experimentation we ran (Figure 9) [13].
An Experiment Canvas is a tool that allows teams to design a tailor-made experiment at the right time. The first step to solve the idea of implementation or problem statement needs to be well addressed and understood before we come up with the right solution [13].

Experiment Canvas consists of eight sections, they don't have to be filled out in this particular order [13].

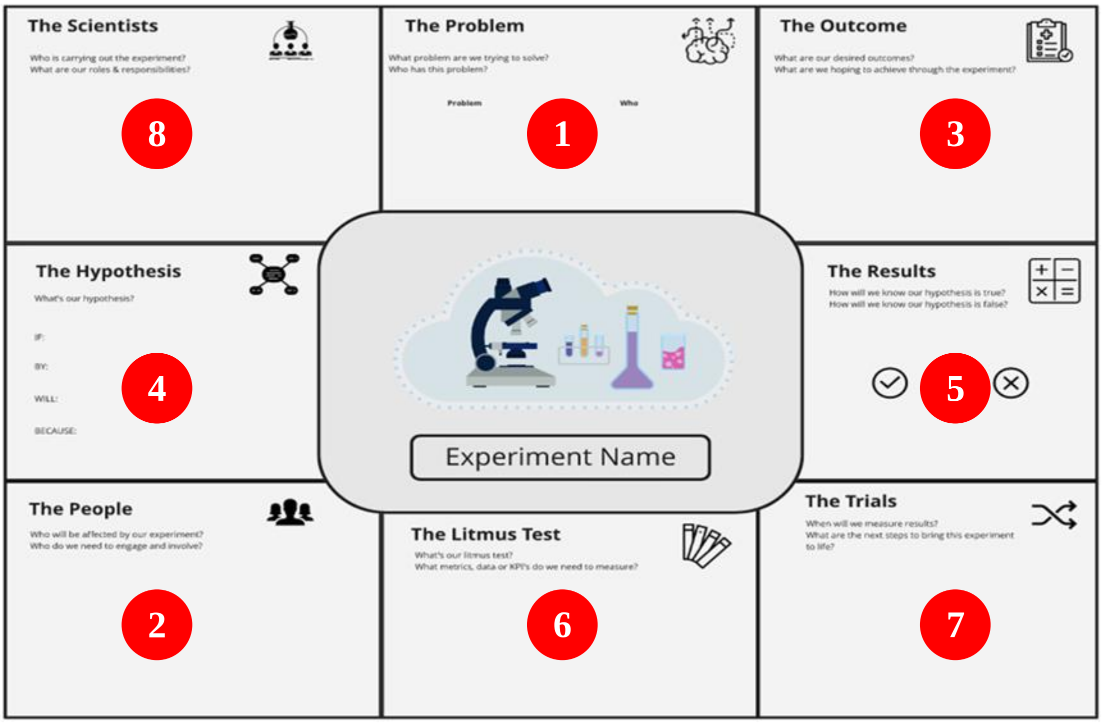



**Figure 9 Experiment Canvas Template**

(Source:Becker, 2022, adapted by author)

### 1. The Problem

The first section is the problem, which refers to the scientific question or problem that the experiment aims to address. It is important to clearly define the problem before beginning the experiment, as this will guide the design of the experiment and ensure that it is relevant to the scientific community [13].

If the team understand the problem, they can come up with the right solution. This section focuses on what problem are the technical teams trying to solve and who are the participants [13].

**2. The People**

The second section is the people, which refers to the individuals or groups that will be involved in the experiment. This includes both the researchers themselves and any participants or subjects who will be involved in the study. It is important to consider the ethical implications of the experiment and to ensure that all participants are treated with respect and dignity [13].

Need to identify who is the user and prioritise who else needs to be worked with by answering these questions:

- Who has this problem?
- Who will be affected by our experiment?
- Who do we need to recruit and engage? [13].

**3. The Outcome**

The third section is the outcome, which refers to the specific result or outcome that the experiment aims to achieve. This may be a specific measurement or observation, or it may be a more general goal, such as understanding a particular phenomenon or testing a specific hypothesis [13].

At this phase, the problem has already been determined and who is concerned by the problem.

Two fundamental questions now need to be answered:

- What changes do we expect to see if we try to solve the problem?
- How will it be different and better? [13].

**4. The Hypothesis**

The fourth section is the hypothesis, which is a statement that proposes a potential explanation for the problem being studied. The hypothesis should be testable and specific, and it should be based on existing scientific knowledge.

Section that tends to take the longest. Because it's one of the most important sections, it needs to be clear about what we do and why.

**IF** – what you could do is go to your problem section and take that problem section, and reward it as it is already accomplished,

**BY** – you are going to put your solution there, what you are going to do differently,

**WILL** – you want to capture benefits and outcomes of the solution if it is successful,

**BECAUSE** – a belief statement, in order to make the team ideate around why the hypothesis is true [13].

**5. The Results**

The fifth section is the results, which refers to the data or observations that are collected during the experiment. These may include quantitative data, such as measurements or statistics, or qualitative data, such as observations or interviews. This section revolves around measuring and how will we know if the hypothesis is either true or false. We would like the technical teams to have a group conversation to come up with one or two specific numbers. At this point the facilitation of the Experiment Canvas is close to be done [13].

**6. The Litmus Test**

The sixth section is the litmus test, which refers to the criteria that will be used to evaluate the success or failure of the experiment. This may include statistical significance, practical significance, or other relevant measures.

If the team has decided on metric in order to decide if the hypothesis is true or false then the next step is to move to this section. If those are the numbers that we need to determine the success or failure, what kind of data do we need to collect during this experiment in order to be able to make the determination in the end? Facilitating a group conversation in this section will bring us all of the desired answers [13].

**7. The Trials**

The seventh section is the trials, which refers to the specific methods and procedures that will be used to conduct the experiment. This includes the selection of participants, the design of the study, and the methods used to collect and analyse data.

Three fundamental questions now need to be answered:

- When will we measure results?
- What are the next steps to bring the experiment to life?
- If we know how long we are going to run this for, what are the steps we need to get this experiment to life?

Good to assign responsibility to people. Here comes the decision to settle down when the team will measure the results [13].

### 8. The Scientists

The final section is the scientists, which refers to the individuals or teams responsible for conducting the experiment. This includes the researchers themselves, as well as any assistants or collaborators who may be involved in the study [13].

The Trials and The Scientists sections order can be swapped depending on the team, sometimes the team who is going to run this experiment is already fixed and the team members might indicate how long the experiment needs to be run for [13].

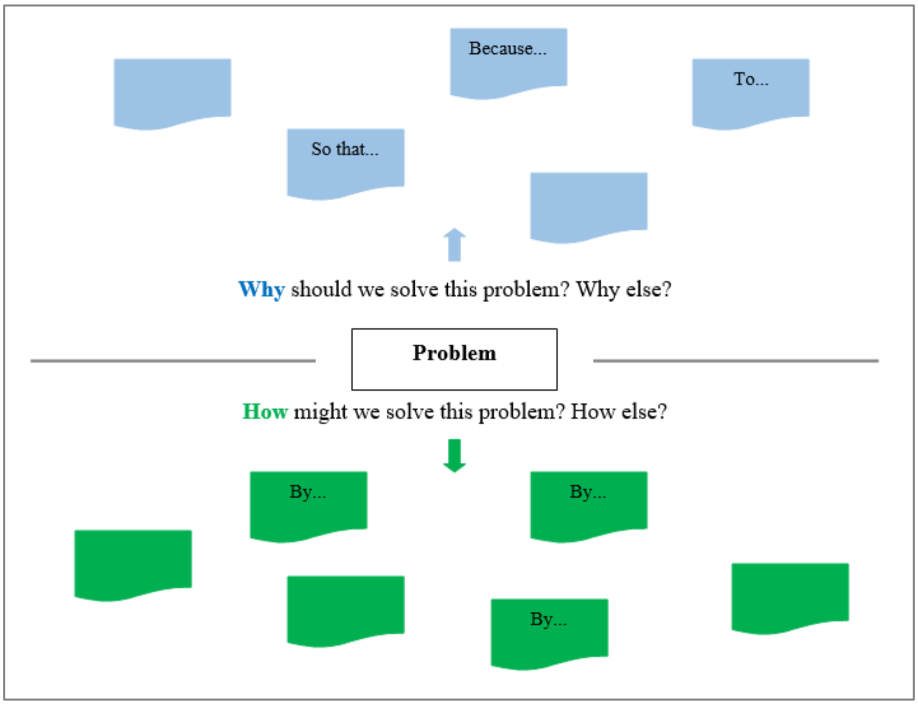



**Figure 10 Abstraction laddering**

(Source:Becker, 2022, adapted by author)

Complete The Trials first if we will know for how long we will collect the data. Later we will determine the core team that will be running the experiment [13].

Overall, the Experiment Canvas provides a structured approach to planning and conducting scientific experiments, which can help to ensure that experiments are conducted in a rigorous and systematic manner, and that they produce valid and reliable results [13].

After each iteration comes the retrospective to learn from the experiment and adjust the metrics/sections if needed. Unless the team decides that they no longer need to run the experiment [13].

Before the teams gets into the Experiment Canvas it is very important that they either have their problem well-structured or have done a problem-framing activity such as Abstraction laddering (Figure 10) [13].

Abstraction Laddering is a brainstorm and the participants explore "why" and "how" statements generated from the initial problem statement in order to explore context, possibilities, and potentially reframe the original problem statement to one that is more compelling or broader in nature [13].

# 2 ANALYSIS OF CONTEMPORARY SITUATION

The following section describes the RedHat company where the practical part of the bachelor thesis was carried out. It further introduces the project itself and its solution.

In this Bachelor thesis, we delve into a real-world problem-solving scenario involving RedHat, a prominent open-source software solutions provider. Specifically, we focus on the RedHat Satellite Team, responsible for the development and maintenance of the RedHat Satellite product. When faced with complex technical challenges, the team seeks assistance from the Agile & Continuous Improvement (ACI) team through Jira tickets.

To address these challenges, the ACI team adopts the Experiment Canvas method, an iterative problem-solving framework. By employing this approach, the teams aim to identify the root cause, propose and test solutions, and achieve effective resolutions. Through this case study, we showcase how the Experiment Canvas method is applied in real problem-solving, highlighting the importance of collaboration and experimentation in the software development environment.

## 2.1 About the company

**Figure 11 Company logo**

(Source: RedHat Source Page)

RedHat is an American multinational software company that specializes in the development of open-source software products. It was founded in 1993 and is headquartered in Raleigh, North Carolina. RedHat is a leading provider of enterprise Linux and cloud solutions, including operating systems, middleware, applications, management tools, and support services [14].

One of RedHat's most notable products is the RedHat Enterprise Linux (RHEL) operating system, which is widely used in data centers, cloud computing, and virtualization environments. RedHat also offers other popular open-source solutions such as OpenShift, a platform for containerized applications and Kubernetes orchestration, Ansible, an automation and configuration management tool, and JBoss Enterprise Middleware, a suite of Java-based middleware products [14].

RedHat has a strong focus on open-source software development and community involvement. It has a vast network of contributors who collaborate on its various projects, and it sponsors many open-source initiatives, such as the Fedora Project, a community-driven Linux distribution, and the OpenJDK project, which develops the Java SE platform [14].

In 2019, RedHat was acquired by IBM in a deal worth approximately $34 billion. Despite the acquisition, RedHat operates as a distinct entity within IBM and continues to develop its open-source software products and services [14].

RedHat has a significant presence in the enterprise software market and counts many large organizations among its customers, including Fortune 500 companies. Its open-source approach and commitment to community involvement have contributed to its success and popularity within the industry.

## 2.2 Project team and the solved project

RedHat – Agility & Continuous Improvement Team (ACI)

**What is the ACI Team?**

The ACI team coaches, educates, and guides teams in applying modern development practices in an effective and valuable manner. The organization grew as a result of investment that product organizations made into their Digital Transformation journeys. Staffing is directly linked to product cost centres, and therefore is not considered a shared service. As a result of this funding, we are primarily focused on CCX, OpenShift, RHEL and RHV [14].

**What do we do?**

The ACI Team is engaged in improving RedHat's Product and Technologies' organizational health and maturity. We support Digital Transformation by working with teams to create and implement improvement strategies that maximize the removal of process waste in the system, employee engagement and well-being, and ensure RedHat delivers high return on investment. The key to the team's success is to work impartially with all roles in the organization to drive measurable improvements in organizational performance and health [14].

The team is divided into 4 pillars (Figure 12) where each covers different types of activities. Membership in each of these pillars is fluid [14].

I am a member of Experiments & Continuous Improvement pillar.

| Experiments & Continuous Improvement | Strategic Alignment | Technical Enablement | Training Delivery |
|---|---|---|---|
| We help teams be more focused and methodical in their continuous improvement efforts.<br><br>Create a culture of experimentation and change that leads to improved product value, team performance, and job satisfaction. | Helping organizations deliver the right features at the right time. This team works on aligning company strategy with feature work and helping teams set up OKRs. | Through the lens of improvement, this pillar enables the Agility and Continuous Improvement teams as well as Product Development Teams to get the most out of our selected communication tools, reduce waste from duplicative tooling, and add tools in where we are missing functionality. | This pillar focuses on the creation of training and workshops needed for improvement within product development teams, providing expertise to help scale our impact through empowering teams, and overall be the curators of all of our team's content. |

**Figure 12 Four pillars of the team**

(Source: RedHat Source Page)

Nowadays what our pillar offers is that when somebody reaches out for help, we let them fill out the request form. After receiving the filled-out request form, we do pre discovery just to find out more about their request, and what we currently offer are workshops to design and run the experiments. Eventually after each iteration we tend to do the retrospective.

### 2.2.1 The Problem

The Satellite Value Stream Team in RedHat contacted our ACI Team us with a problem and a potential solution. Initially, we considered convening the entire group, but one team member had already devised a solution and wanted to test it. To ensure collaboration and fairness, we opted to involve the entire team in the decision-making process. This allowed everyone to contribute their perspectives on the problem at hand and explore the most optimal solution. By operating in this manner, we ensured that the group acted cohesively, rather than being led by a single individual.

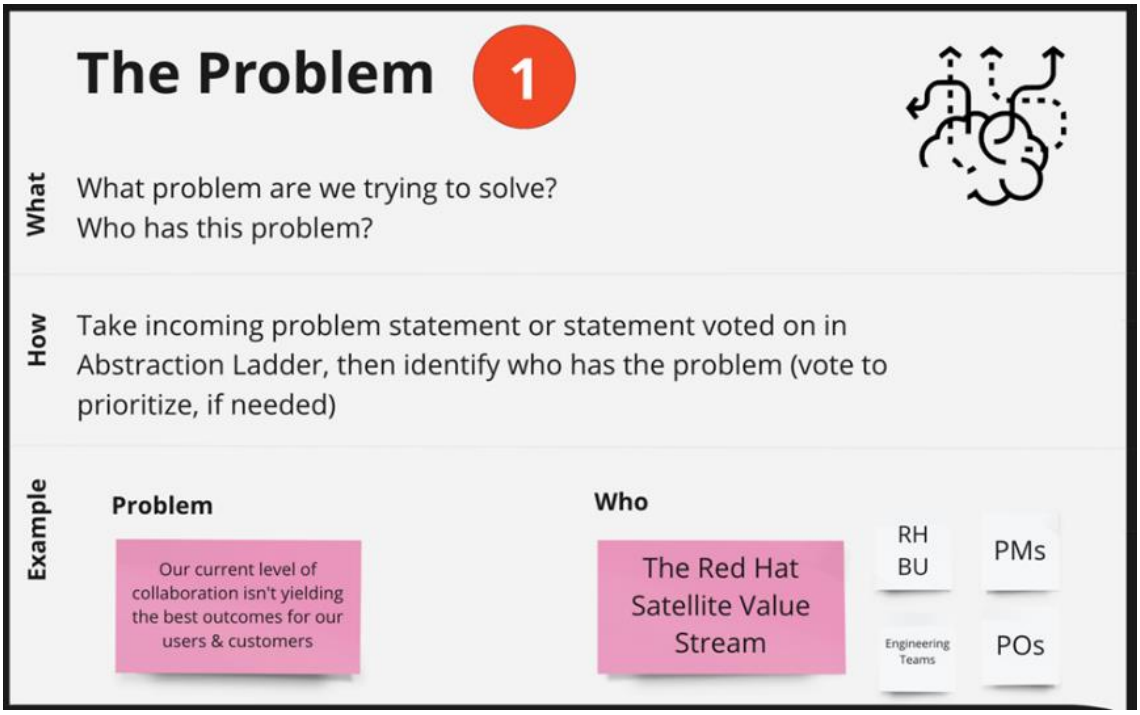



**Figure 13 The Problem**

(Source: Author)

The problem, as described by the team, was that their current level of collaboration was not producing the best outcomes for their users, customers, and partners. This is a robust statement that can help the team to work together more effectively, resulting in better decisions as a team.

### 2.2.2 The People

On this occasion, we completed the Experiment Canvas in a non-standard sequence. Our reasoning behind this decision was due to the fact that we discussed the RedHat Satellite Value Stream as the individuals who are affected by this issue. Consequently, we aimed to broaden our scope and determine who else would be involved in this problem. Our goal was to identify all parties that would be impacted by this challenge and ensure that we informed them of our efforts to address it.

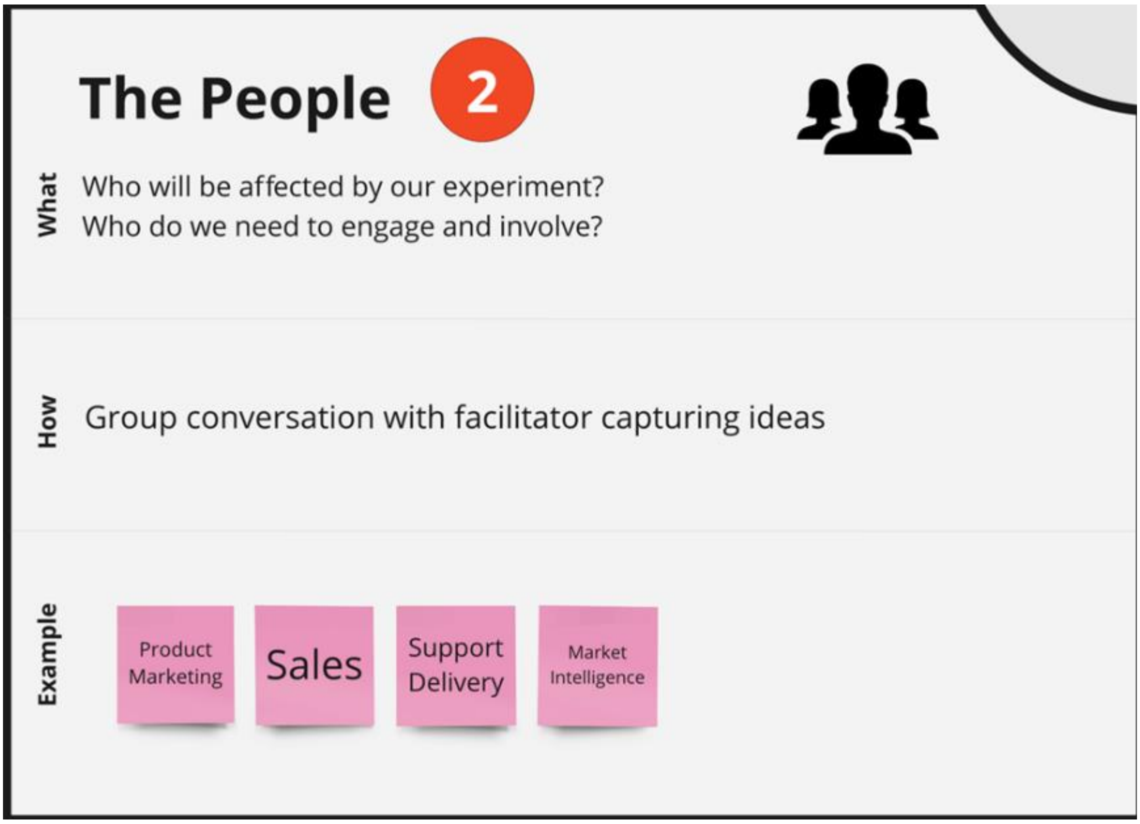



**Figure 14 The People**

(Source: Author)

### 2.2.3 The Outcome

Subsequently, we proceeded to the outcomes section, wherein we aimed to address the question: "What results should we expect to see?" This portion of the discussion was an open and unstructured dialogue where we aligned our thoughts on what could happen if we resolved this problem.

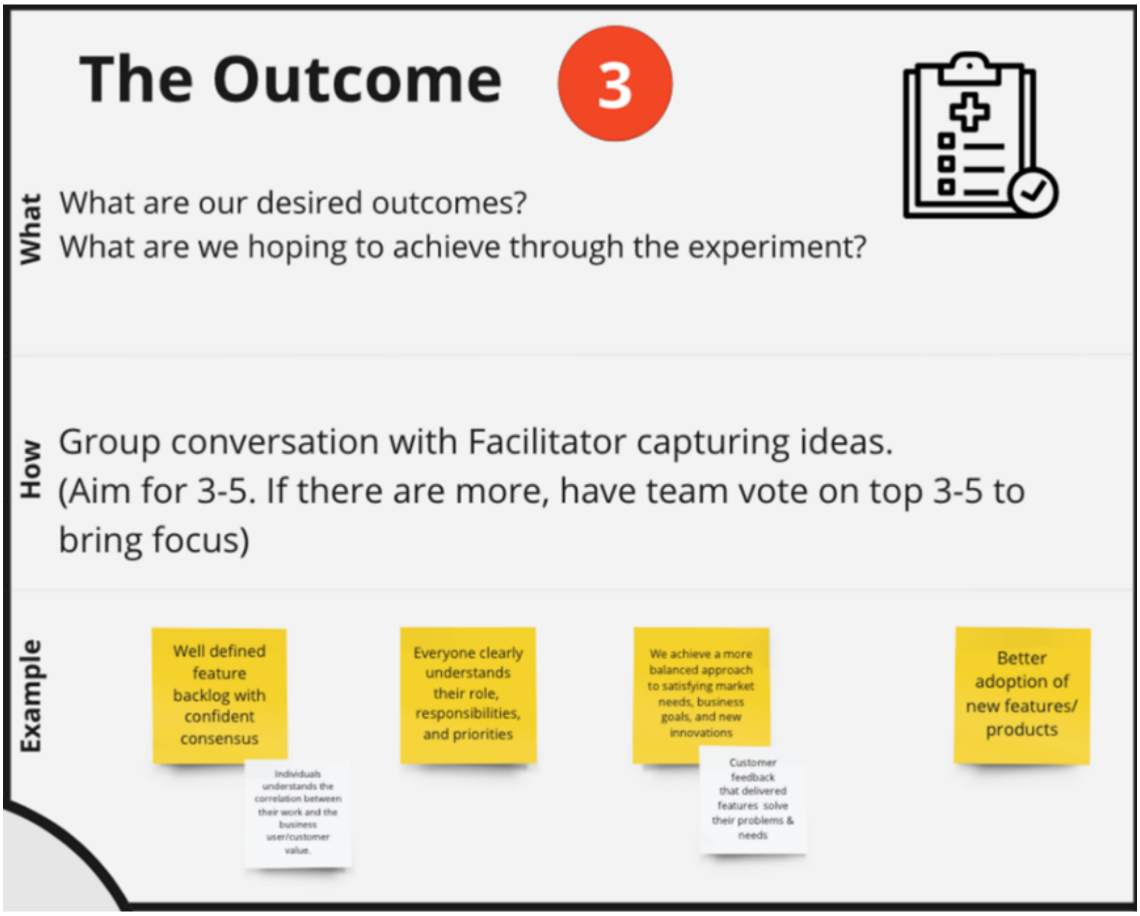



**Figure 15 The Outcome**

(Source: Author)

We identified the following four potential outcomes:

1. We hoped to have a well-defined feature backlog that instilled confidence within the team regarding its content, prioritization, and all relevant details. This means

that by collaborating on the backlog and work processes, team members could develop trust in why the backlog appears as it does.

2. We aimed to ensure that everyone understood their role, responsibilities, and priorities. As a team, we would take collective ownership of this problem and determine who would handle which aspects of the solution.
3. We anticipated achieving a more balanced approach to meeting market needs, business goals, and fostering new innovations. By engaging in collective discussions and incorporating customer feedback into our planning process, we expected to create better problem statements in our backlog that would result in a more balanced approach to meeting market needs and working on new innovative ideas.
4. We envisaged that the better backlog, enhanced team understanding, and balanced approach to meeting market needs and business goals would lead to better adoption of new features and products. By delivering the right things, team members could increase the adoption and satisfaction levels of the products, especially when introducing new features or products, resulting in better hard metrics for the products. Thus, by understanding what we were doing and why, we could achieve a higher level of adoption for our offerings.

### 2.2.4 The Hypothesis

Moving forward to the hypothesis, we dedicated a significant amount of time, nearly half an hour, refining the hypothesis statement by iterating and discussing various aspects to reach a meaningful conclusion.

Eventually, we agreed that if we enhance our level of collaboration by functioning as a cross-functional "Pod," with all facets of the delivery process working together daily, and collaboratively defining our backlog with context based on human-centered design research, we will be able to communicate precisely why we are working on something and how it benefits our users, customers, and partners. If we implement these collaborative efforts to make decisions collectively as a team, we can reduce our feature delivery time by improving understanding and implementation, enhancing innovation and value, and ultimately producing better outcomes for our users, customers, and partners. Our belief is rooted in the idea that if we operate as one team, collaborating with clear context and purpose, we can deliver better products and services. Therefore, we believe that collaboration is the cornerstone for generating the best ideas and delivering successful outcomes.

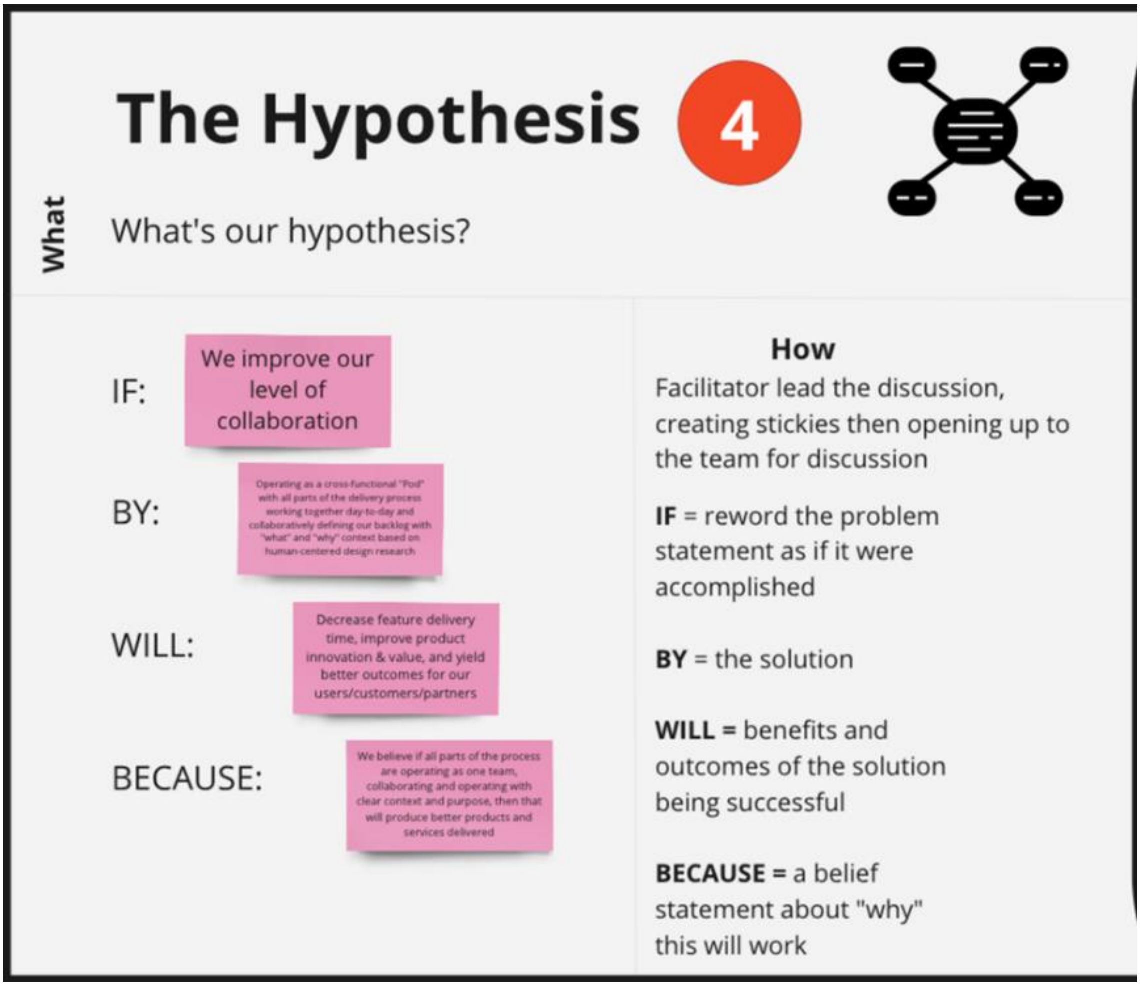



**Figure 16 The Hypotesis**

(Source: Author)

Arriving at the hypothesis statement, we proceeded to define the specifics of the experiment. Our next step was to switch the order and tackle the litmus test section before focusing on "The Results" section.

### 2.2.5 The Litmus Test

One area we are considering measuring is the time it takes to translate market problems into product-specific epics. This involves assessing the speed and effectiveness of our collaboration, which was one of the team's key responsibilities. Another area we are looking to measure is the level of confidence in our product backlog, with a particular focus on improving Jira confidence. In the long run, we want to measure feature and product adoption rates, as we anticipate that this new way of working will result in better outcomes, specifically the improvement of our new feature and product rate, as discussed in our outcomes. Finally, we aim to measure our understanding and confidence in our goals, market problems, and features.

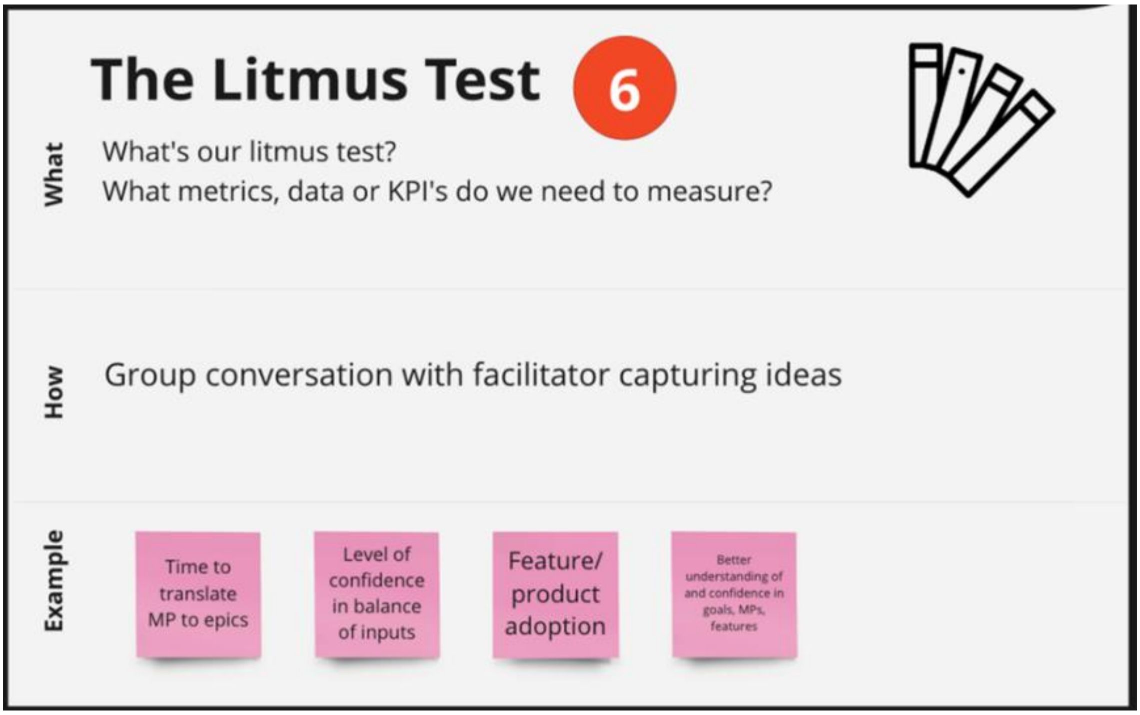



**Figure 17 The Litmus Test**

(Source: Author)

Once we had set the litmus test, we proceeded to prioritize our actions and draw a clear boundary. This marked a crucial turning point in the experiment. We then began to contemplate the potential outcomes if we were to conduct this experiment. Specifically, we deliberated on two key questions:

1. What specific changes can we anticipate to see as a result of this experiment?
2. What are the key metrics we must attain to have confidence in asserting whether this experiment was a success or a failure in facilitating the accomplishment of our objectives?

### 2.2.6 The Results

We established two critical metrics that we must achieve to determine the success of our experiment.

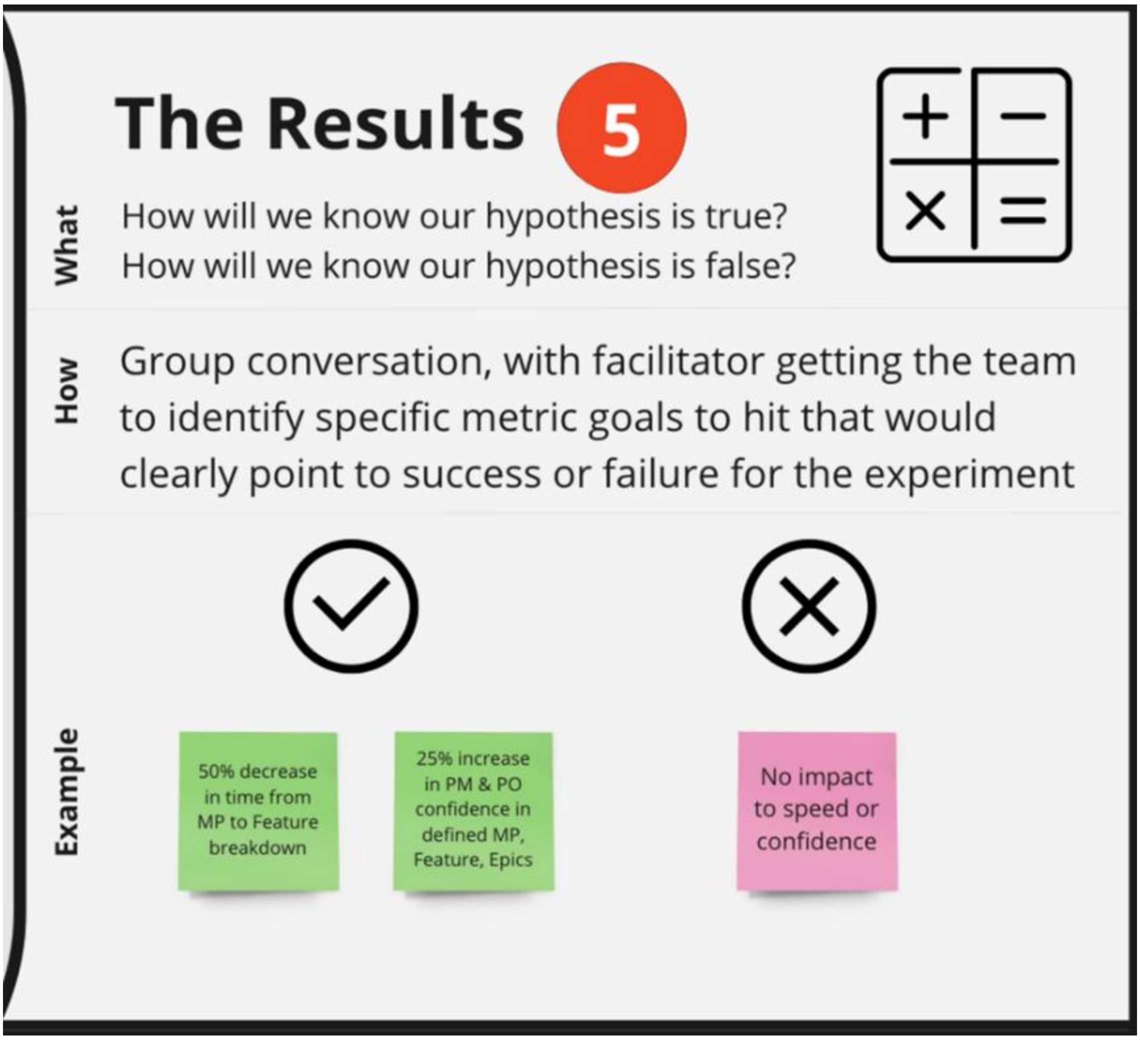



**Figure 18 The Results**

(Source: Author)

Firstly, we aim to achieve a 50% reduction in the time it takes to transform Market Problems into Feature breakdown. This is based on the concept in the Litmus Test section, where we translate market problems into epics. We hypothesize that efficient collaboration would facilitate quick alignment due to a better understanding, and hence, we could efficiently break down market problems into the next level of work that needs to be done.

Secondly, we aim to achieve a 25% increase in the confidence of product management and ownership in defined Market Problems, Features, and Epics. We identified Product Managers and Product Owners as key individuals that need to have confidence in the Jira and our backlog. The combination of these metrics would indicate that we are working faster, more clearly, and building greater trust and understanding in our backlog and its rationale.

We would consider the experiment a failure if there were no impact on speed or confidence in the defined Market Problems, Features, and Epics.

### 2.2.7 The Trials

After extensive discussion, we resumed our meeting with an essential question. How long do we anticipate running this experiment to test our hypothesis, achieve our desired outcomes, and collect data?

To plan a follow-up, we needed to establish a time limit. We agreed on a 30-day period as the team was new, and members still had their day jobs, making it challenging to commit full-time. To achieve our objectives, we decided to schedule weekly collaborative sessions to work together on Market Problem statements for a month and evaluate the outcome. We estimated that working together and collaborating in a live workshop would take a month, so we set a midway checkpoint to adjust if needed after two weeks. We immediately scheduled a retrospective in the calendar for 30 days to hold ourselves accountable and assess progress. Next, we discussed the necessary steps and assigned them to team members to prepare for the experiment. Nora was responsible for identifying one Market Problem that we would work on together. Jenny scheduled live sessions for us to explore the problem space and gain a better understanding as a team. Jimmy handled some tasks, such as developing a plan to structure a workshop and translating data and understanding into an improved Market Problem. In the third week, we would hold the workshop and break down the results in the fourth week to create the next level of epic breakdowns for the team. We would then review the outcomes, assessing whether the collaborative effort resulted in a better Market Problem that effectively communicated the product's purpose and goals.

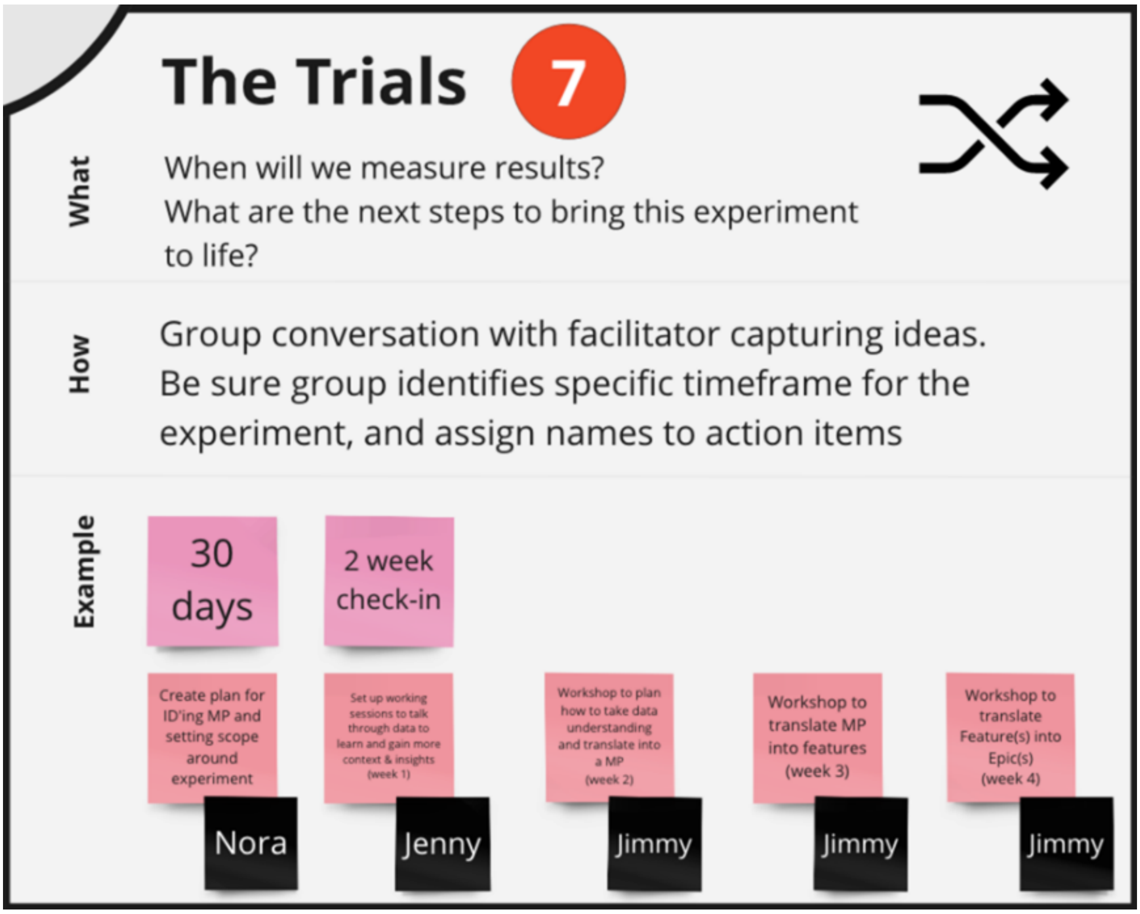



**Figure 19 The Trials**

(Source: Author)

### 2.2.8 The Scientists

Discussing the details of the experiment, we moved on to determine who would be involved in driving it forward. To guide our decision, we referred to Section 8, "The Scientists", which outlines the necessary skills and expertise for running experiments successfully. Based on our previous discussions, we identified specific roles that would be essential to include in the experiment. Jenny, the UXD team member, would be responsible for conducting research and bringing data to the table. Nora, the Product Manager, would provide valuable insights into the product roadmap and goals. Dimitry, the Product Owner, would contribute their expertise in defining and prioritizing features. Some engineers would be involved to provide technical knowledge on the feasibility of breaking down features. Finally, Jimmy was assigned as the facilitator to ensure smooth coordination and execution of the experiment.

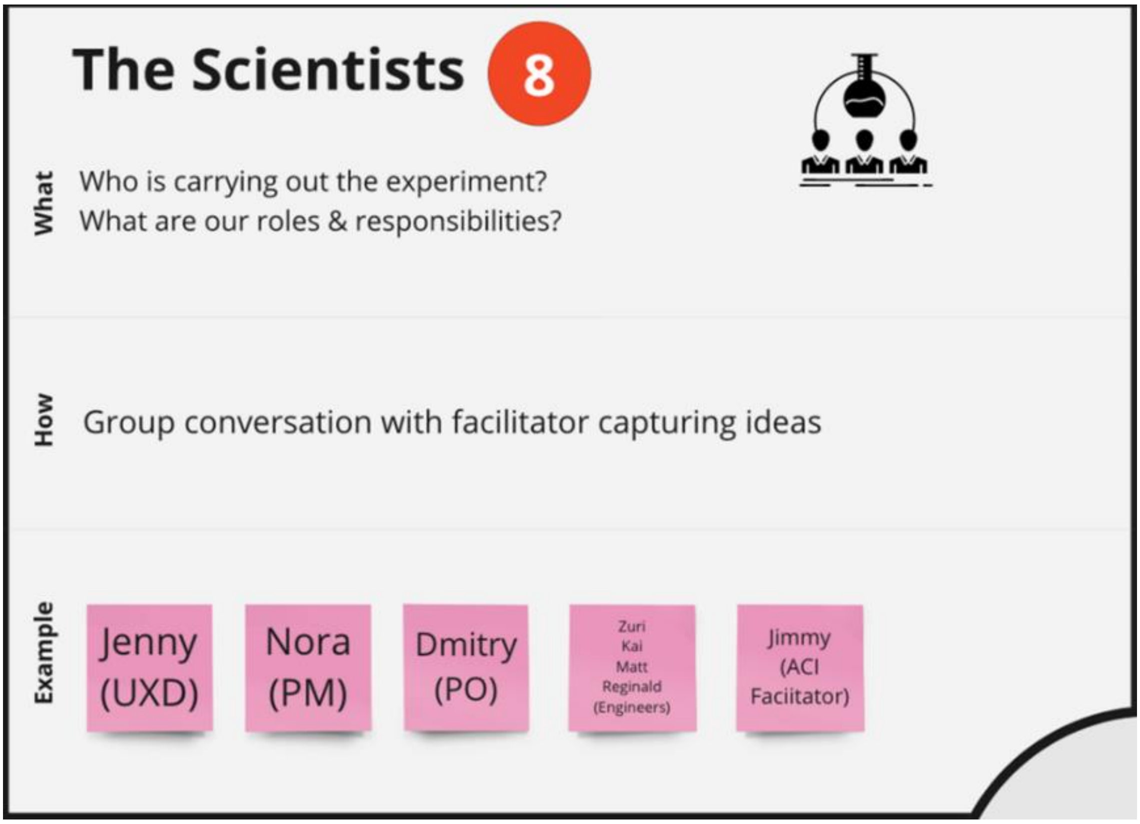



**Figure 20 The Scientists**

(Source: Author)

Upon the completion of our discussion, we had achieved a great deal of clarity on our goals, the duration of the experiment, the necessary data, the individuals who would participate, and their respective responsibilities. At this point, our attention turned to the next steps that we had identified during our trials. We reviewed the preliminary tasks that needed to be completed to ensure a successful execution of the experiment for a 30-day period. We monitored the progress of these tasks in our meetings, and once everything was completed, we were ready to proceed with the experiment.

After running the experiment for a month, we conducted a retrospective to assess its effectiveness. We ensured psychological safety and asked everyone to fill out a questionnaire to gauge their level of confidence in the Market Problem statement before and after the experiment. The results were positive, with most participants reporting an increase in confidence and knowledge of the problem. Feedback was received on the usefulness of the experiment, and some roles found it more helpful than others. Overall, the team considered the experiment a success as it achieved the metrics, we set out to accomplish: a clearer understanding of the Market Problem in a shorter time frame, resulting in the decision not to conduct further experiments on this particular problem. The specifics of the retrospective will not be disclosed as we prioritize maintaining trust within the team.

# 3 PROPOSAL OF SOLUTION

In this proposal, we outline a potential approach to resolving the identified problem. The experimentation process can unfold in various ways. For our specific example, the problem was successfully resolved after the first iteration, wherein the team felt that they had achieved their primary objective of improving team collaboration. However, in cases where the desired outcome is not achieved, certain decisions need to be made based on the results obtained and the success or failure of the hypothesis.

In the event of failure, it is necessary to revisit the creation of the experiment canvas and make adjustments to the sections and metrics involved. This iterative process allows for improvements to be made, providing an opportunity to try again in subsequent iterations. Alternatively, if the desired outcome proves consistently unattainable, it may be necessary to consider discontinuing the current approach.

On the other hand, in the case of success, there is still potential for further refinement. It is important to identify areas that can be adjusted and improved upon to enhance the overall effectiveness of the solution. By continuously evaluating and adapting the solution, it is possible to maximize its impact.

The proposed solution of enhancing collaboration within the RedHat Satellite Team through the implementation of the Experiment Canvas method offers several benefits. By adopting a cross-functional "Pod" approach, where all aspects of the delivery process work together on a daily basis, and by collaboratively defining the backlog based on human-centered design research, the team can achieve the following advantages:

**Increased Collaboration**

The solution fosters a higher level of collaboration among team members. By working closely together and sharing knowledge and expertise across different areas, team members can leverage their collective skills and experiences, leading to improved problem-solving and decision-making.

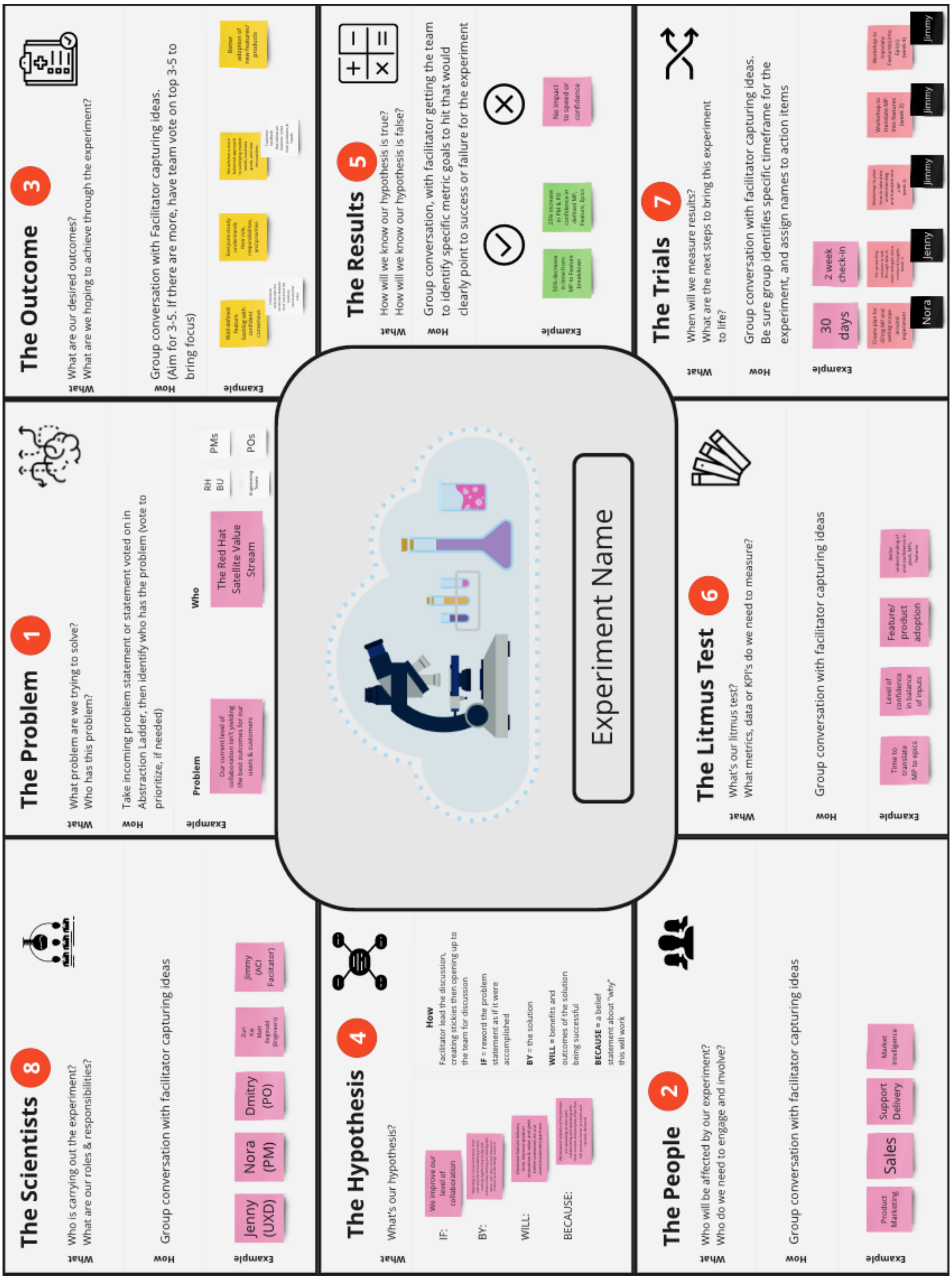



**Figure 21 Experiment Canvas**

(Source: Author)

**Effective Decision-Making**

With enhanced collaboration, the team can make better decisions collectively. By involving all stakeholders in the decision-making process, the team can gather diverse perspectives and insights, leading to more informed and well-rounded choices. This, in turn, enables the team to prioritize effectively and align their efforts towards delivering valuable outcomes.

**Reduced Feature Delivery Time**

The collaborative approach facilitates a better understanding and implementation of features. By working together and clarifying the context behind the work, the team can streamline their development process and reduce any potential misunderstandings or delays. This ultimately results in faster delivery of features, enabling the team to be more responsive to user needs and market demands.

**Enhanced Innovation and Value**

Through collaboration, the team can tap into a wider pool of ideas and expertise. By encouraging open communication and cross-functional cooperation, innovative solutions can be generated, leading to the creation of valuable and impactful products and services. This approach helps the team stay ahead of the competition and deliver greater value to users, customers, and partners.

**Better Outcomes**

By operating as one team with clear context and purpose, the proposed solution aims to produce better outcomes for users, customers, and partners. Collaboration allows for the alignment of goals, improved coordination, and a shared understanding of the value being delivered. This, in turn, enhances customer satisfaction, strengthens partnerships, and drives overall business success.

In summary, the RedHat Satellite Team can benefit from the proposed solution by increasing collaboration, improving decision-making, reducing feature delivery time, fostering innovation, and ultimately achieving better outcomes for their users, customers, and partners. By operating as a cohesive and collaborative unit, the team can leverage their collective capabilities and create a more effective and successful work environment.

### 3.1.1 Duration of activities

Overall, the duration of this project was 30 days with 8 people being involved in the project.

**Abstraction Laddering**

Estimated duration: 1 MD

This one-hour session involves breaking down the problem into its components, analysing each part separately, and then integrating the knowledge to inform experimental design.

**Experiment Canvas**

Estimated duration: 2 MD

The facilitation of the Experiment Canvas took place over the course of two one-hour sessions. Upon completing the Experiment Canvas, during the final session, the team reached a consensus to conduct a check-in in two weeks' time and carry out a retrospective analysis after a span of 30 days.

**Check-in**

Estimated duration: 0,5 MD

The purpose of this check-in was to provide us with updates on any issues encountered during the experiment canvas. All eight individuals participated in this 30-minute activity.

**Retrospective**

Estimated duration: 1 MD

After 30 days of experimentation, a retrospective was conducted to reflect on the progress and outcomes of the activities. The retrospective allowed the team to analyse the effectiveness of the experiments, identify strengths and weaknesses, and gather insights for future improvements. Through open discussions, the participants shared their experiences, highlighted key learnings, and discussed potential adjustments to optimize future experiments. The retrospective provided a valuable opportunity to evaluate the overall impact of the 30-day experimentation period and generate actionable recommendations for continued growth and success.

### 3.1.2 Determination of costs

As the salary rates of the company's employees involved in the project cannot be disclosed, salary costs were estimated based on projected man-days, internal knowledge and the average salary in Europe for the job. For reasons of confidentiality, specific data on personnel costs could not be disclosed. In the end, the costs of the project were only related to the salary costs of the staff and the cost of the resources needed by the staff to do the four activities mentioned above. The employer pays statutory contributions on behalf of the employees and these costs were not reflected in the project because it is an internal company process and the customer is not affected by these costs.

**Table 1 Salary costs**

| **Salary costs** | |
|---|---|
| Project Manager | 144,0 € |
| Product Owner | 157,5 € |
| User Experience Designer | 108,0 € |
| Agile Practitioner (Facilitator) | 135,0 € |
| Product Experience Engineers (3x) | 98,0 € |
| **Total** | **642,5 €** |

Source: Author

### 3.1.3 Benefits of the proposed solution

The team can achieve the following advantages:

**Increased Collaboration**

The solution fosters a higher level of collaboration among team members. By working closely together and sharing knowledge and expertise across different areas, team members can leverage their collective skills and experiences, leading to improved problem-solving and decision-making.

**Effective Decision-Making**

With enhanced collaboration, the team can make better decisions collectively. By involving all stakeholders in the decision-making process, the team can gather diverse perspectives and insights, leading to more informed and well-rounded choices. This, in turn, enables the team to prioritize effectively and align their efforts towards delivering valuable outcomes.

**Reduced Feature Delivery Time**

The collaborative approach facilitates a better understanding and implementation of features. By working together and clarifying the context behind the work, the team can streamline their development process and reduce any potential misunderstandings or delays. This ultimately results in faster delivery of features, enabling the team to be more responsive to user needs and market demands.

**Enhanced Innovation and Value**

Through collaboration, the team can tap into a wider pool of ideas and expertise. By encouraging open communication and cross-functional cooperation, innovative solutions can be generated, leading to the creation of valuable and impactful products and services. This approach helps the team stay ahead of the competition and deliver greater value to users, customers, and partners.

**Better Outcomes**

By operating as one team with clear context and purpose, the proposed solution aims to produce better outcomes for users, customers, and partners. Collaboration allows for the alignment of goals, improved coordination, and a shared understanding ofthe value being delivered. This, in turn, enhances customer satisfaction, strengthens partnerships, and drives overall business success.

In summary, the RedHat Satellite Team can benefit from the proposed solution by increasing collaboration, improving decision-making, reducing feature delivery time, fostering innovation, and ultimately achieving better outcomes for their users, customers, and partners. By operating as a cohesive and collaborative unit, the team can leverage their collective capabilities and create a more effective and successful work environment.

# CONCLUSION

In conclusion, this bachelor thesis has shed light on the significance of experimentation in project management, with a particular focus on the agile approach. Through the exploration of agile principles and practices, two key benefits have emerged.

Firstly, agile methodologies encourage specificity and a clear understanding of project goals. By emphasizing the importance of defining objectives and desired outcomes, experimentation within project management becomes more purposeful and targeted. This ensures that teams are not just engaging in random trials but rather are fully aware of what they are trying to achieve and why. The ability to articulate precise goals allows for better planning, resource allocation, and decision-making throughout the project lifecycle.

Secondly, the agile approach to experimentation aligns with its broader philosophy of adaptability and continuous improvement. Running a retrospective after each experiment enables project teams to reflect on the outcomes and learn from their experiences. By analysing the gathered data, teams can identify strengths, weaknesses, and areas for improvement. This feedback loop facilitates an iterative process where adjustments can be made, and insights gained from one experiment inform the planning and execution of subsequent ones. This continuous learning and adjustment cycle aligns with the fundamental principles of agile project management.

The findings of this thesis emphasize the value of experimentation in project management, particularly within an agile framework. By embracing experimentation, teams can leverage its benefits to drive innovation, optimize processes, and enhance project outcomes. The ability to be specific in defining objectives and agile in executing experiments ensures that project teams remain adaptive and responsive to changing circumstances, thereby increasing the chances of success.

As organizations navigate an increasingly complex and dynamic business landscape, the integration of experimentation within project management becomes imperative. By embracing an agile mindset and adopting experimentation as a core practice, project teams can harness the power of continuous learning, improvement, and adaptability. This thesis serves as a reminder of the significant role that experimentation plays in project management and provides a foundation for future research and application in this field.

# BIBLIOGRAPHY


[1] JALOTE, P., 2008, A Concise Introduction to Software Engineering, UK: Springer-Verlag.

[2] PEJCHAL, J. 2015. Agilní a tradiční metodiky v projektovém řízení. Diplomová práca. Masarykova univerzita. Fakulta informatiky [online]. [cit. 28.11.2022]. Available from: <https://is.muni.cz/th/ht6wr/Agilni_a_tradicni_metodiky_v_projektovem_rizeni.pdf>.

[3] ŤUREKOVÁ, B. 2018. Projektové řízení v konkrétní IT spoločnosti [online]. [cit. 12.11.2022]. Available from: <https://is.muni.cz/th/ig9t3/turekova_bakalarskapraca.pdf>.

[4] MARTIN, R., 2003, Agile Software Development- Principles, Patterns and Practices, USA: Prentice Hall.

[5] DOLEŽAL, J. a kol, 2016. Projektový management: Komplexně, prakticky a podle světových standardů. Grada Publishing a.s., 2016. 424 p. ISBN 978-80-247-5620-2.

[6] COLE, R., SCOTCHER, E. 2015. Brilliant Agile Project Management: A Practical Guide to Using Agile, Scrum and Kanban. Pearson, 187 s. ISBN 978-1-292-06358-4.

[7] ASHMORE, S., RUNYAN, K. 2014. Introduction to Agile Methods. Addison-Wesley Professional. 328 p. ISBN 978-0-321-92956-3.

[8] STAPLETON, J. 2003. DSDM: Business Focused Development. Pearson Education, 2003, 2nd edition. 239 p.. ISBN 0-321-11224-5.

[9] BJÖRKHOLM, T., BRATTBERG, H. 2010. Prioritera Fokusera Leverera. Din Snabbguide till Lean, Agile, Scrum, Kanban och XP. Published by Vulkan, 2010. ISBN 978-91-978630-5-6. Available from <https://www.crisp.se/wp-content/uploads/2019/01/Prioritera_fokusera_leverera_v1_5.pdf>.

[10] RUBIN, J., CHISNELL, D. 2008. Handbook of Usability Testing, Second Edition: How to Plan, Design, and Conduct. Wiley Publishing, 2008. 386 p. ISBN 978 0 470 18548 3.

[11] SEFFAH, A., GULLIKSEN, J., DESMARAIS, M. C. 2006. Human-Centered Software Engineering - Integrating Usability in the Software Development Lifecycle. Springer Science & Business Media, 2006. 391 p.. ISBN 978 1 4020 4113 6.

[12] ISO 9241-210:2010 Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems.

[13] Becker, J. 2022. *Experiment Canvas*. In Open Practice Library powered by RedHat Open Innovation Labs. Available from <https://openpracticelibrary.com/practice/experiment-canvas/>.

[14] RedHat Source Page (internal).

# ABBREVIATIONS AND SYMBOLS LIST

| | |
|---|---|
| **ACI** | **A**gility & **C**ontinuous **I**mprovement |
| **CCX** | **C**ontact **C**entre **E**xpress |
| **DSDM** | **D**ynamic **S**ystems **D**evelopment **M**ethod |
| **HCD** | **H**uman-**c**entered **d**esign |
| **ISO** | The **I**nternational **O**rganization for **S**tandardization |
| **RAD** | **R**apid **a**pplication **d**evelopment |
| **RHEL** | **R**ed**H**at **E**nterprise **L**inux |
| **RHV** | **R**ed**H**at **V**irtualization |
| **UI** | **U**ser **i**nterface |
| **XP** | **E**xtreme **P**rogramming |

# FIGURE LIST

## TABLE LIST