BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF BUSINESS AND MANAGEMENT

FAKULTA PODNIKATELSKÁ

INSTITUTE OF FINANCES

ÚSTAV FINANCÍ

DEVELOPMENT AND APPLICATION OF AN INNOVATION SCORECARD SYSTEM TO MEASURE THE SUCCESS OF INNOVATION IN RED HAT

NÁVRH A IMPLEMENTACE SYSTÉMU INNOVATION SCORECARD ZA ÚČELEM MĚŘENÍ ÚSPĚCHU INOVACE VE SPOLEČNOSTI RED HAT

MASTER'S THESIS

DIPLOMOVÁ PRÁCE

AUTHOR AUTOR PRÁCE	Bc. Tetyana Shpilka
SUPERVISOR VEDOUCÍ PRÁCE	doc. Ing. Ondřej Žižlavský, Ph.D.

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA PODNIKATELSKÁ

Zadání diplomové práce

Ústav:	Ústav financí
Studentka:	**Bc. Tetyana Shpilka**
Studijní program:	Ekonomika a management
Studijní obor:	Účetnictví a finanční řízení podniku
Vedoucí práce:	**doc. Ing. Ondřej Žižlavský, Ph.D.**
Akademický rok:	2019/20

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává diplomovou práci s názvem:

Návrh a implementace systému Innovation Scorecard za účelem měření úspěchu inovace ve společnosti Red Hat

Charakteristika problematiky úkolu:

Úvod
Cíle práce, metody a postup zpracování
Teoretická východiska práce
Analýza současného stavu
Návrh controllingového rámce Innovation Scorecard
Závěr
Seznam použité literatury
Přílohy

Cíle, kterých má být dosaženo:

Cílem diplomové práce je návrh a následná implementace controllingového systému Innovation Scorecard v rámci procesní inovace ve společnosti Red Hat Czech s.r.o.

Základní literární prameny:

ASHMORE, S. and K. RUNYAN. Introduction to Agile Methods. Addison-Wesley Professional, 2014. 336 p. ISBN 978-0-133-43521-4.

DAVILA, T., M. J. EPSTEIN and R. D. SHELTON. Making innovation work: How to manage it, measure it, and profit from it. Updated ed. Upper Saddle River: FT Press, 2013. 348 p. ISBN 978--13-309258-5.

PARMENTER, D. Key performance indicators: developing, implementing, and using winning KPIs. John Wiley & Sons, Inc., Hoboken, New Jersey, 2015. ISBN 978-1-118-92510-2.

ŽIŽLAVSKÝ, O. Innovation Scorecard: Conceptual Performance Measurement and Management Framework for Innovation Process. Brno: Vutium Press, 2016. 293 p. ISBN: 978-80-214-5370-8.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2019/20

V Brně dne 29.2.2020

L. S.

doc. Ing. Vojtěch Bartoš, Ph.D. ředitel	doc. Ing. et Ing. Stanislav Škapa, Ph.D. děkan

Abstrakt

Tato diplomová práce představuje návrh a implementaci řídicího systému známého jako Innovation Scorecard, který byl zaveden a aplikován v rámci společnosti Red Hat Czech s.r.o. Účelem bylo vylepšení současného modu operandi společnosti se zvláštním důrazem na zlepšení vývoje a implementačních procesů softwaru. Hlavní zaměření práce je především na měření toho, jak úspěšný byl jakýkoli navrhovaný inovativní způsob práce. Diplomová práce je rozdělena do tří částí. První část obsahuje základní znalosti, které jsou důležité pro pochopení konkrétního tématu výzkumu. Druhá část diplomové práce popisuje společnost a umožňuje tak pochopení interních procesů a systémů v rámci firmy. Třetí část obsahuje související návrhy, jak zlepšit vnitřní procesy. To je založeno na analýze s hlavním zaměřením na to, jak společnost v současné době funguje.

Klíčová slova

inovace, pracovní procesy, týmová práce, komunikace, posílení postavení, efektivita procesů, Innovation Scorecard, agilní prostředí, projektové řízení, cílové hodnoty, inovační metriky, měření metrik,

Abstract

This thesis presents the design and implementation of a control system known as Innovation Scorecard that was introduced and implemented within Red Hat Czech s.r.o. to improve the company's current modus operandi, with particular emphasis on improving software development and implementation processes. The main focus of this research has been on measuring how successful any suggested innovative way of working has been. The thesis is divided into three parts. The first part contains basic knowledge that is important for understanding the particular topic of the research. The second part introduces the company and provides an understanding of the internal processes and systems in operation within the company. The third part contains associated proposals how to improve the internal processes. This is analysis-based with the main focus on how the company currently operates.

Key words

Innovation, Work Processes, Teamwork, Communication, Empowerment, Process Efficiency, Innovation Scorecard, Agile Environment, Project Management, Target Values, Innovation Metrics, Metric Measurement,

Rozšířený abstrakt

Red Hat je předním poskytovatelem takzvaných open source a linuxových operačních systémů, které jsou založeny na zpřístupnění originálních zdrojových kódů. Posláním společnosti Red Hat je *„být katalyzátorem pro zákazníky, přispěvatele a partnery s cílem vytvořit zdokonalenou technologii založenou na principu open source“*. K dosažení tohoto poslání Red Hat uplatňuje takzvaný „agilní“ přístup řízení procesů, jehož největší předností je rychlá reakce na změny požadavků ze strany zákazníků v průběhu celého vývojového cyklu. Denním pracovním jazykem společnosti je především angličtina, což je také důvod, proč byla tato práce napsána v angličtině.

Hlavní zaměření této diplomové práce je projekt nazvaný Continuous Integration (CI), ve kterém jde o zavedení konceptu Innovation Scorecard v rámci pracovního prostředí ve společnosti Red Hat Czech s.r.o. Zahrnuje měření toho, jak úspěšné byly implementované procesní inovace ve společnosti Red Hat Czech s.r.o., s využitím výsledkových dat výzkumu „před a po“ změně. Tento projekt byl jedním ze tří projektů, které tvoří celkový výzkumný projekt Innovation Scorecard.

Proces Innovation Scorecard se skládá z následující struktury: Cíle – Kritické faktory úspěchu – Klíčové ukazatele výkonnosti – Vybrání metriky – Cílové hodnoty. Výchozím bodem bylo stanovení cílů, kterých má být dosaženo prostřednictvím aplikace Innovation Scorecard v praxi. Následujícím krokem bylo stanovení kritických faktorů úspěchů, které jsou nezbytné pro projekt k dosažení svého poslání. S kritickými faktory úspěchu jsou dále spojené klíčové ukazatele výkonnosti, představující měřitelnou hodnotu, která bude demonstrovat úspěšnost plnění vytýčených cílů v průběhu celého projektu. Velmi významná část této práce představuje stanovení metrik a jejich cílových hodnot. Metriky slouží k vyjádření inovačních cílů, které musí být jasné, bezchybné a zároveň dosažitelné. Není důležitý počet stanovených metrik, ale jejich kvalita a celkový přínos k úspěšnému zavedení systému Innovation Scorecard. Právě stanovení správných metrik a jejich cílových hodnot je důležité k dosažení určených cílů.

Model Innovation Scorecard byl realizován v řádném, strukturovaném a logickém pořadí. To byl jediný způsob, jak zaručit úspěch. Z tohoto důvodu byla metodika vybraného Innovation Scorecard rozdělena do pěti fází řízených čtyřmi branami, aby byla zajištěna hladká realizace. Aby byl systém Innovation Scorecard přizpůsoben konkrétnímu procesu

Continuous Integration, byly vyhodnoceny a považovány za nejvhodnější pouze následující čtyři fáze:

- Fáze 1: Generování nápadů
- Fáze 2: Vývoj nápadu
- Fáze 3: Implementace
- Fáze 4: Po implementační fáze

Vzhledem k povaze práce, se tým iScorecard zabýval různými aktivitami řešení problémů, aby zajistil, že projekt byl dodán v dohodnutých časových lhůtách. Aby bylo možné plánovat, organizovat, kontrolovat a řídit celý projekt, provedl tým iScorecard následující činnosti pro řízení tohoto projektu:

- pravidelné měsíční přezkumy pokroku,
- osobní setkání s Red Hat,
- zápisy ze schůzek,
- pravidelné telefonní hovory s projektovým manažerem a Leapp týmem,
- průběžné hodnocení stavu "dashboard report",
- revize souborů.

Největší zisky společnosti byly realizovány v oblasti změn požadavků zákazníků především v průběhu vývojového cyklu. Společnost Red Hat se rozhodla přijmout takzvaný agilní přístup k řízení projektů "Scrum", protože to bylo nejvhodnější pro jejich současný modus operandi (Scrum se skládá z některých pracovních postupů používaných v agilním projektovém řízení, které jsou založeny na každodenní komunikaci a flexibilním přístupu k hodnocení plánů). Každá z fází tohoto výzkumného projektu byla řízena použitím takzvaných "krátkých sprintů", které jsou typické pro použití při vývoji softwaru, aby se věci dělaly rychle a efektivně bez překážek a přerušení externími zásahy. Kompletní analýza současných procesů a postupů Red Hat odhalila potenciální oblasti pro zlepšení procesů v řadě oborů. Tato práce byla zásadní pro rozvoj týmového přístupu Innovation Scorecard a pro návrh, vývoj a zavedení nejvhodnějšího fáze/brány přístupu (rozdělení projektu do zvládnutelných fází, viz popsané výše).

Ve spolupráci s jedním z interních vývojových týmů společnosti Red Hat (známého jako Leapp Team) bylo provedeno šetření s cílem posoudit, jaké chyby je třeba opravit v rámci procesu CI. To vytvořilo základ pro posouzení toho, jaké zlepšení procesů bylo považováno za nezbytné k vyřešení těchto problémů a jak by se výsledky mohly měřit z hlediska jejich úspěšného provádění. Je třeba zmínit, že způsob práce v Red Hat zdůraznil řadu oblastí, které by mohly být zlepšeny týmem Innovation Scorecard, jako je komunikace mezi různými členy týmu, čas na vyřešení hlášených chyb a posílení postavení zaměstnanců tak, aby se rozhodovali sami. První výsledky naznačují (projekt CI stále probíhá v době psaní tohoto rozšířeného abstraktu), že zavedení vyváženého přehledu výkonnostních metrik učinilo některé významné zlepšení procesu a způsoby práce v rámci Red Hat. Úplné výsledky budou známy až přibližně 6 měsíců po dokončení výzkumného projektu (očekává se, že to bude 31. prosince 2021). Tým iScorecard plánuje po dokončení tohoto projektu provést realizaci přínosů, aby zjistil, zda projekt přinesl slíbené výhody.

Výsledek této práce významně přispěl ke zlepšení modus operandi ve společnosti Red Hat. Například doba, kterou zaměstnanci Red Hat v minulosti potřebovali, aby vytvořili nápady na potřeby, byla výrazně snížena. Komunikace mezi členy týmu byla zlepšena tím, že zaměstnanci mohou komunikovat přímočaře, aniž by bylo nutné nejprve odkazovat na vrcholné vedení. A konečně, některé procesy vývoje softwaru byly změněny a posíleny s tím výsledkem, že doba potřebná k dokončení určitých úkolů byla výrazně snížena.

Tato diplomová práce však měla následující uložená omezení. Vzhledem k pracovnímu tlaku nebyli zaměstnanci Red Hat vždy schopni sdílet potřebná data projektu s týmem iScorecard, když to bylo potřeba. To vedlo k určitému zpoždění, v případě dodání třetího projektu známého jako Continuous Delivery (CD). V důsledku toho, nebylo možné dokončit tuto diplomovou práci zcela tak, jak bylo původně plánováno.

Výsledek tohoto výzkumu bude zajímavý jak pro akademiky, tak pro komunitu v praxi. Navrhuje se další výzkum, který by zlepšil přijatý koncept Innovation Scorecard. Odborníci z praxe v jiných odvětvích budou mít z tohoto výzkumu rovněž prospěch, aby zajistili, že inovace nebo změna nebudou jednoduše prováděny, aniž by byl zaveden proces, který měří úspěch těchto inovací.

Bibliografická citace

SHPILKA, Tetyana. *Development of an Innovation Scorecard System in Red Hat to Measure a Process Innovation Performance.* Brno, 2020. Dostupné také z: https://www.vutbr.cz/studenti/zav-prace/detail/127676. Diplomová práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav financí. Vedoucí práce Ondřej Žižlavský.

Čestné prohlášení

Prohlašuji, že předložená diplomová práce je původní a zpracovala jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušila autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 17. května 2020 ....

podpis studenta

Poděkování

Touto cestou bych ráda poděkovala vedoucímu mé práce doc. Ing. Ondřeji Žižlavskému, Ph.D. který přispěl svými zkušenostmi a nápady při psaní této diplomové práce. Velmi oceňuji jeho odborné vedení, profesionální přístup a čas, který mi věnoval. Dále bych chtěla také poděkovat prof., Dr. Eddie Fisher, Ph.D., M.Sc, členu týmu iScorecard, který mi pomáhal svými připomínkami, radami při obtížích nebo otázkách, na které jsem při práci narazila. Mé poděkování patří také společnosti Red Hat a členům Leapp týmu za poskytnuté informace a podklady pro zpracování této práce. V neposlední řadě patří velké díky mé rodině za psychickou a finanční podporu během celého studia.

Obsah

Introduction

It appears that the consumer market is constantly changing. Customer expectations in terms of product quality and delivery times have been rising steadily over the last few years. As a direct result, most businesses had to respond to this new challenge in order to stay competitive and remain in business. The winner is the one who reacts first and presents the consumer with the product or service closest to their requirements. There is a growing need to respond flexibly to market changes.

One way to become competitive is to be creative and strive for innovative products. Innovation is no longer a new phenomenon. Many publications exist that describe and present this topic well. Different authors suggest different definitions of what is meant by "innovation". It all depends on their personal perception. Innovation itself is not just a method or idea. It is an action or process that actually leads to an innovation. It is a way of thought and behaviour that must be reflected across any organisation, including every department, every project and every person who works for the company. The term "innovation" is encountered in many different working environments such as software development and construction. What is currently missing is an approach to measure how successful each innovation has actually been. Therefore, organisations should carry out, for example, continuous evaluations of their current innovation activities including projects, and use this data to decide whether or not to continue with their projects.

The measurement of how successful innovations at work including projects has been depends on the use of an appropriate, user-friendly measurement system. Kaplan and Norton (1996) introduced a so-called Balanced Scorecard concept which focused on measuring an organization's performance in both financial and non-financial terms. This Balanced Scorecard itself was not an appropriate tool for measuring the actual value added by innovations.

This thesis introduces a newly-developed practical approach called Innovation Scorecard that provides an efficient and effective means to measure innovation in different work environments. Innovation Scorecard is based on a Gate Process that sits within a management framework that ensures a consistent and repeatable measurement of innovations.

The IT industry is considered to be one of the most innovative and dynamic sectors in the Czech Republic. The adoption of a so-called “agile way of working” has helped this industry to reduce ineffective and long-winded ways of working in order to deliver solutions to their customers much faster and at higher quality. Key to agile working is its ability to respond fast to changes.

It is for this reason that Red Hat in Brno was chosen to act as the main partner for this innovation scorecard research project. Red Hat is an innovation-driven organisation and is considered to be the world's leading provider of enterprise open source solutions. The expected results from this Innovation Scorecard implementation research project are intended to significantly contribute to improving the efficiency, economies of scale and final competitiveness of organizations throughout the information technology (IT) industry both in the Czech Republic and other countries.

1 Problem Statement and Aims of Thesis

A shortfall exists in business how the outcomes of innovations in organisations and companies are actually measured. The primary objective of this research is to adapt, adopt and implement a measurement system that enables organisations to measure how successful they have been in the implementation of innovations across their businesses. This thesis describes how the concept and its associated processes of an Innovation Scorecard system were rolled out within a specific Red Hat software development project known as "Continuous Integration" (CI).

The considered objectives of this thesis are:

- to gain a better understanding of current Red Hat CI processes and then adapt the considered conceptual Innovation Scorecard system to suit Red Hat's modus operandi
- to conduct some research to ascertain what is already known about the subject matter under investigation, such as the theoretical background of the topic of Innovation Scorecard
- to propose an Innovation Scorecard solution that is based on the following structured process: Goals – Critical Success Factors (CSFs) – Key Performance Indicators (KPIs) – Metrics – Target values,
- to amalgamate the considered Innovation Scorecard system with existing work processes and make it work
- to adapt the Innovation Scorecard system so it can be applied in in yet unknown and unchartered territories within Red Hat
- to collect and act on feedback received from CI project participants

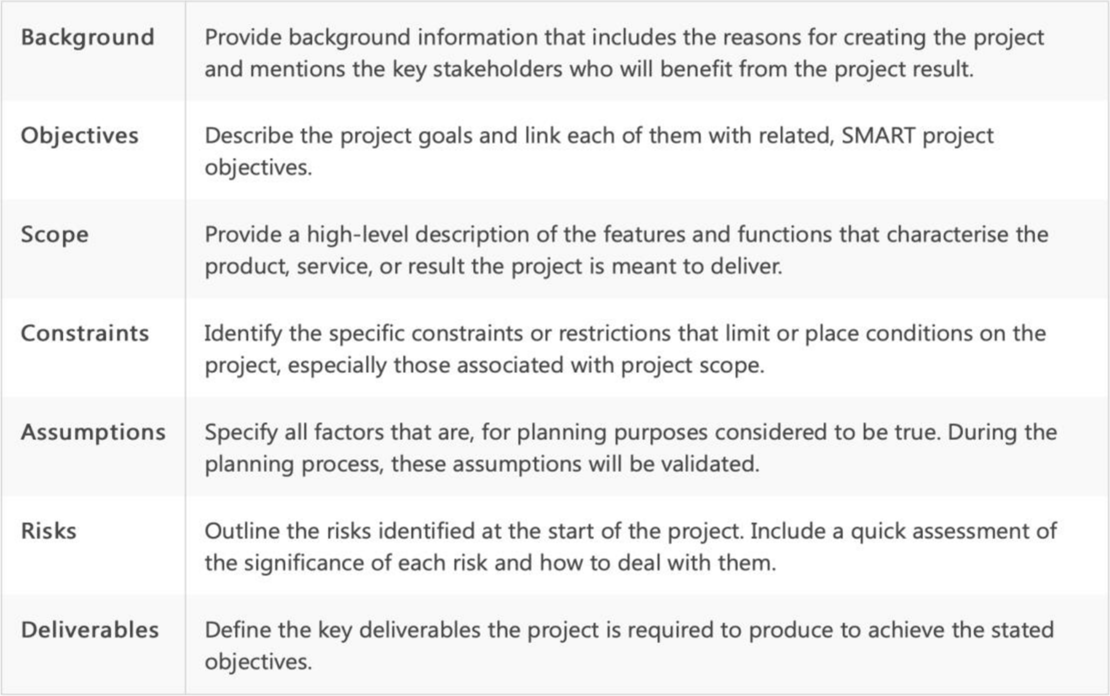
This thesis is divided into three parts. The first part is concerned with a detailed explanation of what is meant by "agile working environments", why this approach is of paramount importance to Red Hat's "way of working" within a software development area and how the concept of agile is applied within so-called and typical "SCRUM" software development environments. Understanding this background information is vital for understanding how the Red Hat team works and how it achieves its stated goals. This section will also include a limited literature review for the purpose of establishing what is already known about the subject under review. This review will focus on the four substructures of agile project management considered to be the most important for this research: Teamwork, Communication, Work Processes and Empowerment. Monitoring of relevant identified areas will be presented in a clear table form. The table will capture opinions from various authors covering the years 2014-2020 to capture only the contemporary thinking in this subject matter. This first part will also include details of what the basic principles of innovation are, what makes a good Innovation Scorecard system and what the various stages and gates are contained within the Innovation Scorecard framework.

The second part of the thesis analyses and provides details of Red Hat's current situation in relation to planned and anticipated research outcomes of this study. Details are provided that relate to the history of Red Hat, its products and services and the relationship with existing customers. This is considered important background information. This research is primarily concerned with one of the three associated projects that form part of the Innovation Scorecard project: Continuous Integration (CI). Detailed descriptions of the processes involved, and tools used and applied are included in this section. Applied tools and techniques are described and there will be a detailed description and discussion of the "before" and "after" status in terms of the Innovation Scorecard introduction. Associated issues and problems emanating from the original CI processes are presented to aid further understanding of the need for an Innovation Scorecard approach to reduce or eliminate these issues and problems.

The final third chapter contains optimum solution proposals how the outcomes from this research can be applied within Red Hat's current modus operandi. The CI project is managed in line with current best project management practice based on the Association for Project Management's (APM, 2019) and simple but effective project management approaches based on the well-known BOSCARD principle (Table 1) and document management have been adopted. A project definition document (PDD) will be produced at the beginning of the project to ensure that all important aspects of the planned and considered work have been captured well in a single location. This document is reviewed and updated at regular intervals and it aids management progress reporting and decision-making. The above-mentioned BOSCARD principle will be adopted by the research team to ensure a consistent and coherent approach in term of managing this research project (Haughey, 2011).

Other associated proposals, presented in the last chapter will be elaborated on, interpreted and then presented to senior management at Red Hat. Case studies will be prepared and presented in relation to the Innovation Scorecard application during the various so-called "software development sprints".

Table 1: BOSCARD template
(Source: HAUGHEY, 2011)



Background	Provide background information that includes the reasons for creating the project and mentions the key stakeholders who will benefit from the project result.
Objectives	Describe the project goals and link each of them with related, SMART project objectives.
Scope	Provide a high-level description of the features and functions that characterise the product, service, or result the project is meant to deliver.
Constraints	Identify the specific constraints or restrictions that limit or place conditions on the project, especially those associated with project scope.
Assumptions	Specify all factors that are, for planning purposes considered to be true. During the planning process, these assumptions will be validated.
Risks	Outline the risks identified at the start of the project. Include a quick assessment of the significance of each risk and how to deal with them.
Deliverables	Define the key deliverables the project is required to produce to achieve the stated objectives.

The basis of this thesis is to understand Red Hat's current operational work approaches including policies, processes and procedures. Also included are detail of the attitudes and behaviours of staff. Both qualitative and quantitative data will be collected, analysed and evaluated to generate a better understanding of how Red Hat could implement innovation in future in the most time and cost-effective way. The research data will be made up of both financial and non-financial information. Qualitative research is used to understand basic reasons, opinions and motivations. It provides insight into problems or helps develop ideas/hypotheses for potential quantitative research. For example, quantitative methods of data collection take place based on a group discussion or an individual call. The sample size is usually small, and respondents are selected to meet the quota.

In contrast, quantitative research is used for the collection of numeric data, or data that can be transformed into usable statistics. Quantitative methods of data collection include various forms of research such as face-to-face interviews, phone conversations or systematic observations. The research team collected this data from the results of interviews and important communications with different people in Red Hat. The data is then analysed during the data collection process when the entire process will not be completed until observations, interviews, and questionnaires are complete. The iScorecard team will discuss the extent or scope of the research work with Red Hat including what Red Hat expects to see delivered by the iScorecard team. Included in the scope of this research project are all design elements that make up the CI project, any improvements to the data collection process and the Innovation Scorecard principle introduction within the CI project within Red Hat. Not included in the project scope are any kind of technical solutions associated with the CI project (APM, 2019).

2 Theoretical Background of Thesis

The basic concepts and methods needed to understand the subject matter under review are discussed in this Chapter. The concept of software development in an agile work environment and the basic principles of Scrum are explained in sufficient detail to create a general awareness in the reader. The next section presents a Literature Review associated with this research that is primarily focused on measuring performance in an agile environment. The last section covers the definition of what is meant by innovation, innovation scorecard, a description what is meant by the concept of innovation scorecard and the steps that typically constitute the implementation of an innovation scorecard system.

2.1 Agile software development

The idea of "agile business" (doing things fast and dexterous) was developed in 2001 when a group of business people, in a totally relaxing environment, united to create an alternative to a less flexible "waterfall model" of software development. This model was not able to respond quickly to changes and constant changing requirements by the customer. This group created a document known as the "agile manifest" containing 12 principles based on an agile approach:

1. *"Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.*
2. *At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly. 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. *Business people 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 method of conveying information 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"* (Beck, Kent, 2001).

Beck, Kent (2001), based on these 12 principles of agile software development, defines the four values that need to be followed to make this work successfully:

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

The agile manifesto does not constitute an official document. It is a statement written by a group of men dedicated to software development who were convinced that this activity could be carried out better than at present. They had a desire to find an alternative solution to improve current best practice. This was driven by the frustration of people working in software development who were slowed down by long lead times and early decisions made not being able to be changed later in, for example, projects. In practice, the principles described may or may not work. It depends primarily on the environment into which the methodology is introduced. What is of paramount importance is that these principles can be changed even at short notice if required. It is this flexibility that makes the agile approach such an effective approach (Beck, Kent, 2001).

Currently, large companies such as Google, Amazon, Yahoo or eBay, are more or less using agile methodologies as they are aware that they operate in a very dynamic environment where it is necessary to accept and address changes with great speed and flexibility in order to stand up in a competitive environment (Myslín, 2016).

2.1.1 Agile methodology

The agile approach came in response to an increase in the size of development teams and with this increasing complexity in development. This increase caused a long delay between when the programmer programmed the work and when the user first saw it on the other side. Thus, agile methods support the close cooperation of programmers with target system users to prevent unwanted results (Ashmore, Runyan, 2014).

Practically no agile methodology is strictly determined by the guaranteed and verified procedure. The word “agile” is very close to the word “flexible”, therefore there is always a place to adapt the methodology to a particular project (Šochová, Kunce, 2019). Agile comes from the Latin word agilis, meaning quick, rapid or dexterous.

A great benefit from the traditional method, which is, for example a “waterfall model”, is the focus on the amount of activities that are realistic to be completed within a typical so-called 2 – 4 weeks sprint (to get something done quickly and efficiently by being totally focused on the activity, without any hindrance or disruption). By setting a short time, it will simplify scheduling and ensure that the newly created part can be tested soon, and feedback can be received swiftly. Another advantage of the agile approach is almost constant cooperation between the project team and the customer who gives feedback to the project team. Team members have diverse skills and experiences and must be motivated to work together. Mutual trust and non-blaming are important. At the same time, for an agile method, interactive planning plays an important role in this approach. It is very easy to adapt changing requirements quickly, particularly when these are identified during routine or regular checks (Novoseltseva, 2017).

Gradually, the agile approach proved successful not only in the development of software, but overall in product or solution development and has become very popular in other sectors. It has now also started to play a key role in areas such as finance, telecommunications, marketing and human resource management (HRM), according to Ashmore and Runyan (2014).

2.1.2 Scrum

It appears that the early origin of Scrum arises from a project that was managed by Jeff Sutherland in 1993 (Krishnamurthy, 2012). Working together with Ken Schwaber, both developed Scrum as a formal process in 1995. Ken Schwaber and Mike Beedle developed this method further in 2002. Other scholars suggest that Hirotaka Takeuchi and Ikujiro Nonaka invented Scrum in 1986. The history of Scrum is a topic of regular discussion. The name Scrum was adopted "*from the game of rugby to stress the importance of working as a team in complex product development*" (Verheyen, 2017). Verheyen claims that Scrum is the most used methodology or framework within the agile product delivery area. In contrast, the principle of "lean" is often used in business environments that include software development. It simply means delivering greater value with fewer resources. Another well-known approach, particularly used in software development, is known as Kanban and is primarily concerned with process improvements. Scrum, on the other hand, is focused on getting more work done in less time (Ashmore and Runyan, 2014).

Scrum is a comprehensive managerial methodology that is made up of a complete solution how teamwork should be organised to get the most out of people and to deliver work faster. Scrum was developed to support managing development processes in the most efficient way. This method is well-known for not containing specific tools, technologies and procedures relating to how scrum developers should use it. Instead, it shows how the whole team should work together and how they should communicate to deliver work optimally. Scrum is based on the knowledge that development brings with it a lot of unpredictable events and thus becomes complex. It is a largely managerial methodology which focuses on monitoring and addressing all obstacles that could lead to successful software development (Myslín, 2016).

Working process of Scrum

To understand scrum, it is necessary to understand the roles of people within its framework/processes and to gain a good understanding of the individual parts that make up the scrum framework and its associated processes (Figure 1). The whole point of scrum is to do whatever it takes to get the job done successfully. A typical scrum team is made up of three people with clearly and specifically defined roles and responsibilities: a product owner, scrum master and the development team. Scrum teams are cross-functional and are therefore made up of a variety of people who hold different roles in the organisation such as developers, testers, designers or operational support staff. People performing these roles work closely together to ensure a smooth flow of information and quickly solve problems (Šochová, Kunce, 2019)

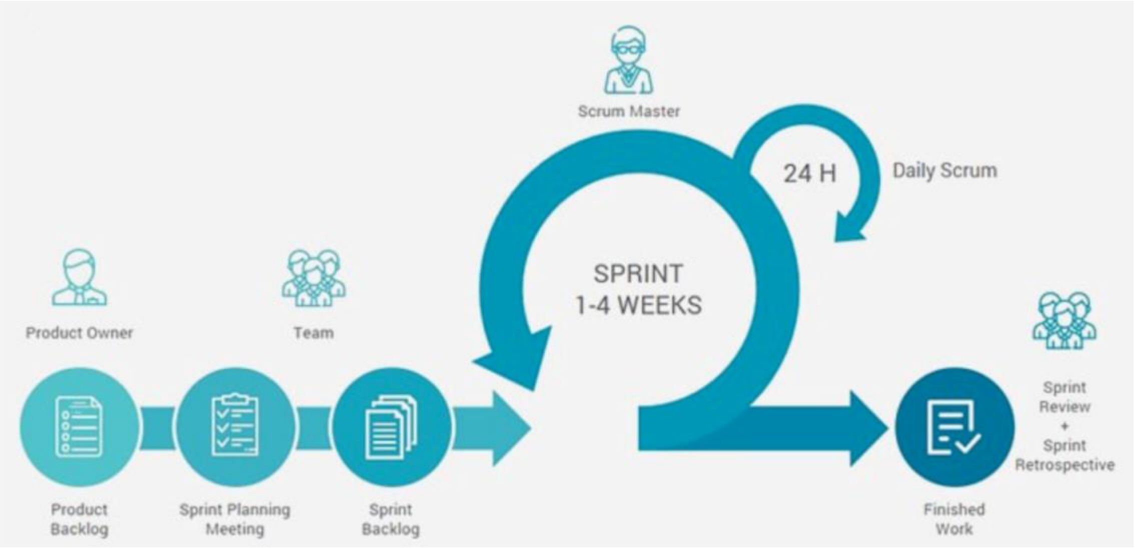


Figure 1: Scrum Framework

(Source: John, 2020)

The scrum team executes on the product owner’s vision. They decide what gets built and the order in which it gets built. The product owner represents the best interest of the end user and ensures that what the customer wants or expects will be delivered. The product owner has the final veto as far as the end delivery is concerned. The product owner is in charge of preparing, for example, any product backlog, a list of tasks and the requirements of what the final product should look like. The product owner is the only point of contact for all questions relating to product requirements. At this stage, it is necessary for the product owner to prioritize the work load to ensure tasks are performed in the right sequence. This is generally referred to, within software development, as “backlog”.

Sprint Planning Meetings form an essential part of the scrum process or framework. Product Owner, team and scrum master meet at regular intervals to sort out and agree what the main priority tasks are and to select which of these priority tasks will be actioned to go forward to the next stage. The outputs from the Sprint Planning effectively become the sprint backlog referred to earlier, in effect describing the requirements the end customer expects (Sutherland, 2014).

The next stage is a sprint that represents a predetermined timeframe in which a team must complete task sets from a specific backlog. The length of time depends on the needs of the team, but the duration is typically between one and two weeks. Sprints are vital to the successful delivery of the intended outputs. It is for this reason that the main emphasis is placed on team empowerment. The team knows what the best solutions are, for example, how to make things work. They are given the autonomy to do whatever it takes to get the job done. Team size usually ranges from five to nine team members. So-called daily scrum team meetings are held to share and discuss progress made. This is important. It allows the team to take immediate corrective action to fix any problems, thus avoiding any lengthy and unnecessary delays. Another valuable contribution these sprints make to the whole process: they assist to produce what are considered to be potentially shippable products (products ready for distribution to the customer). The product owner has the final say as to whether these products have all the features the customer asked for and whether set and agreed quality standards have been achieved.

Each sprint ends with a review. This is sometimes called a "lessons learned" review and in agile projects this is referred to as a "Mini Post Implementation Review". The team reviews the outputs with the product owner so they can identify potential areas for improvement for the next sprint. This process is repeated until the final product has been created to the desired and expected quality standards.

The whole process is overseen by the scrum master who is responsible for the smooth running of the process, for dissolving obstacles/issues that affect productivity/quality and for organizing and facilitating critical follow-up meetings (Myslín, 2016).

2.1.3 Agile project management tool

The most widely used tool in an agile environment is JIRA. It is a tool for registering errors and problems in software development or project management developed by Atlassian, an Australian company. The JIRA control panel consists of many useful functions and features that make it easy to solve problems. JIRA helps support and facilitate the process of project and requirements management, offers flexible and user-friendly tools and monitoring of workers in the performance of tasks to achieve expected performance on the project.

2.2 Literature research

This part of master thesis is about limited literature review, to find out what is already known about the subject matter under investigation. It was a necessary condition to ascertain details of the current theoretical thinking and knowledge in the area of measuring successful change (innovation) within agile projects. This will assist the researcher to bring theory and practice together in the Conclusion part of this master thesis. Four sub constructs of agile project management were investigated, considered to be the most appropriate for this research: Teamwork, Communications, Work Processes and Empowerment (there may be others, but they do not form part of this research). The outcome from this literature review is presented in Tables 2 – 5. The research focused on the work of a number of authors (typically three) covering the years 2014 to 2020 to ensure that contemporary thinking only was captured. Presented first are details of the authors first (names and year of publication, with the full references shown in the overall Reference section of the dissertation). This is followed by, for each area under investigation, of what the characteristics are for each of the four areas. For example, in the area of communication within agile projects, communications are less formal and spontaneous. The final column of the table shows details of how successful, for example, what the impacts of the considered changes/innovations or improvements have been against each area of research, for example, in the area of teamwork "actual versus intended output".

The reviewed literature that relates to teamwork (Table 2) reveals that teamwork is considered essential for today's fast moving and forever changing work environments. It has a direct impact on the successful delivery of planned or expected results. Two of the chosen authors conclude that the adoption of an agile working method makes positive contributions to effective and efficient team working. Another author takes this further by suggesting whether only agile working methods should be adopted as it appears that these produce the most productive results. As each organisation has a different cultural working environment ("This is how we do things around here"), it is necessary for each organisation to assess whether the adoption of an agile working method is most appropriate for them. As the saying goes in Management: “One size does not fit all”.

Table 2: Literature research about teamwork in agile projects

Author	**Observed characteristics of the research**	**The result of the research**
Hidalgo (2019)	The study addresses the extent to which key principles and tools usually used in scrum, due to their potentially positive influence on team dynamics and efficiency, can contribute to the collaborative management and coordination of tasks in research processes a group development model taken from social psychology.	Lessons learned from this case study point to the need to reconsider the suitability of the scrum framework as the best agile approach for distributed research management.
Freire et al. (2018)	According to the agile principles and values, as well as recent research articles, teamwork factors are critical to achieve success in agile projects. Assess the practical usefulness of agile methods through a case study	Within the context of the associated case study, the model can help agile teams in assessing the quality of their teamwork, identifying opportunities for improvement and confirming the positive cost-benefit of its adoption.
Lindsjørn et al. (2016)	Spontaneous communication. Not strong leadership. Self-organizing teams. The team makes decisions; estimates, prioritizes, and delegates tasks in particular. Large team focus, daily meetings. Facilitator helps protect team members from tasks outside the team.	This research confirms that how well teams are managed and looked after is a contributing factor to improving team performance and in the case of projects, improving the quality of delivered products and services. It appears that team performance generally can be improved when the views of team members “how this can be achieved” are taken into consideration.

The outcome from this specific and topic related literature review suggests that the level of relevant and effective communications within an agile working environment (Table 3) is much higher than, for example, compared to non-agile working environments. Whilst the level of communications is much higher, this does not necessarily mean that the relevance of communications is appropriate. It is for this reason that companies who adopt agile working practices, need to adopt performance measurement metrics to validate how effective these communications actually are. This should include areas such as team and stakeholder communications.

Table 3: Literature research about communication in agile projects

Author	Observed characteristics of the research	The result of the research
Loiro et al. (2019)	This research suggests that a typical Agile Project Management team, consisting of a product owner, a team leader and team members, is the optimum way to manage work in an agile work environment. For this to work effectively, an effective workflow management system also needs to be in place that complements the agile work team.	The results reveal that, for instance, less detailed documentation leads to problems such as communication lapses, rework and product inconsistencies in agile settings
Yagüe et al. (2016)	In this research work observations were obtained from three perspectives: communication among team members, communication of the status of the development process, and communication of the status of the progress of the product under development. The research question to which this study responds concerns how development teams perceive the communication infrastructure while developing products using agile methodologies.	Team members consider that the use of appropriate media tools such as smartboards and video tools make them feel in practice that teams are co-located. This allows for the effective sharing of relevant project/process/product information during the development process. This also allows to overcome some of the still existing communication problems.
Stray et al. (2016)	The study investigated how daily meetings are conducted and how team members feel about these meetings. A Grounded Theory study with 12 software teams in three companies in Malaysia, Norway, Poland and the United Kingdom was conducted. 60 people were interviewed, 79 daily meetings were observed, and the resulting data was analysed, evaluated and presented.	The factors that contributed most to a positive attitude towards the need for daily meetings were: information sharing with the team and the opportunity to discuss and solve problems. The factors that contributed the most to a negative attitude were: status reporting to the manager and that the frequency of the meetings was perceived to be too high and the duration too long

By definition, empowerment implies that people need to take an active part in the daily decision-making across an organisation (Table 4). IT teams focus on software development which means they have to have a certain degree of autonomous decision-making. If this is not practised, then the result would be constant consultation with the line manager. This defeats the whole objective of being autonomous. This explains why some traditional strict line management-based organisations appear to be inefficient. Any performance or "innovative way of working" measurements must include the positive and negative effects empowerment has had on people's ability to become more autonomous.

Table 4: Literature research about Empowerment in agile projects

Author	**Observed characteristics of the research**	**The result of the research**
Alsaquaf (2019)	This paper focused on whether agile methods are suitable for large scale distributed projects.	Organisations need to understand at the earliest opportunity how mature their agile teams actually are. This insight will actually help organizations, for example, to define the level of autonomy of their staff. Agile teams that lack maturity need senior management control to coordinate the collaboration between the teams, while more mature agile teams are self-organized teams, and they would therefore be less productive if closer management supervision were introduced.
Hoda, Murugesan (2016)	Based on a Grounded Theory study of 21 participants from six different companies, this paper presented the issues and constraints associated with practicing agile project management in a self-organizing team context	It appears that there is a disparity between how project managers manage their teams and how people are being managed within self-organizing teams. Empowered people still require some guidance and assistance when needed but autonomy implies that, generally, people are expected to make decisions and get on with the work.
Tessem (2014)	This research aims to get a better understanding of how empowerment is enabled in software development teams, both agile and non-agile, to identify differences in empowering practices and levels of individual empowerment.	Agile developers, in contrast to non-agile developers, appear to better placed to achieve higher levels of contribution towards achieving organisational goals. For non-agile teams, higher empowerment can be obtained by systematically applying low-cost participative decision-making practices in the manager–developer relation and among peer developers. For agile teams, it is essential to follow the empowering practices already established more rigorously.

It is essential that organisations choose, roll out and apply an appropriate management approach and method how to run and manage the organisation as effectively and efficiently as possible. Internal work processes (Table 5) and procedures need to reflect how the organisation, in terms of operations, wishes to be perceived by customers and key stakeholders. Adopted processes need to be fit for their intended purpose. It appears that agile working methods are particularly suited to software and IT development areas. This industry is very dynamic and has a need to be able to respond to constantly changing environments very quickly.

Table 5: **Literature research about work processes in agile projects**

Author	Observed characteristics of the research	The result of the research
Serrador, Pinto (2015)	A data sample of 1002 projects across multiple industries and countries was used to test the effect of an agile way of working in organizations, measuring the efficiency and overall stakeholder satisfaction against organizational goals.	Their findings suggest that agile methods do have a positive impact on project success as far as the application of work processes is concerned.
Younas et al. (2018)	The research considers if, for example, a merger of agile and cloud computing could provide infrastructure optimization and automation benefits to agile practitioners.	The study concludes that agile development in a cloud computing environment is an important area in software engineering. There are many open challenges and gaps.
Lei at al. (2017)	This research statistically compares the effectiveness of two of the most applied agile methods: Scrum and Kanban. These methods are highly regarded for their contributions to manage software development projects effectively and efficiently	Results suggest that both Scrum and Kanban lead to the development of successful projects, and that the Kanban method can be better than the Scrum method in terms of managing project schedules.

In summary, the conducted literature review, with its main focus on Teamwork, Communication, Work Processes and Empowerment, showed how changes in agile projects/scrum working environments, have developed over the past six years. Innovation in agile working environments is well documented. In the next section, the concept of Innovation is described in more detail and how it relates to this research. Next the concept of what is meant by an Innovation Scorecard, considered most suitable for use within an IT company that already operates within an agile work environment.

2.3 Definition of Innovation

It is almost impossible to find a general definition of innovation, since there is simply no clear definition. Different authors interpret this term in different ways, for which it has a different meaning for different people. The perception of the concept of innovation can therefore be very subjective. Therefore, I would like to point out a few innovation interpretations according to different authors.

One of the first to address this topic was J. A. Schumpeter, whose theory is still considered the basis of a modern approach to innovation. Schumpeter understood innovations such as product, procedural or organizational changes that may not stem from scientific discoveries but can arise by combining existing technologies or applying them in a new context (Schumpeter, 2004).

Another of the authors, who is considered an expert in innovation and whose work is based on a number of well-known authors is Peter F. Drucker. He interpreted innovation as "*the specific tool of entrepreneurs, the means by which they exploit change as an opportunity for a different business or a different service. It is capable of being presented as a discipline, capable of being learned, capable of being practiced*". Just as Schumpeter explains this concept of Ducker from the perspective of an entrepreneur (Drucker, 2006).

In the Czech Republic, the founders of innovation theory include František Valenta, who sees innovation "as any change in the internal structure of the production organism. That is, any transition from the original to the new state". Compared to Schumpeter and Ducker, Valenta's concept is broader. Innovation in its conception can therefore be understood in such a way that improving the production of products or services or production processes brings about an improvement in the economic potential of the company (Valenta, 2001).

From current authors concerned by innovation can be called Žižlavský, who says that innovation is a targeted change related to products (new products or improvements to existing products), production methods, organization of work, production process (new types of solutions) and methods used in the company for the first time (Žižlavský, 2016).

As it can be seen that the concept of innovation is interpreted differently from many perspectives (Figure 2). Therefore, the distribution of innovation is most often classified according to the Oslo Manual, which was developed by experts in measuring and evaluating innovation in OECD Member States. This is the best international directive for the collection and use of innovation concept data and, above all, helps to define what innovation is signifying. According to this approach, innovation is divided into four basic types:

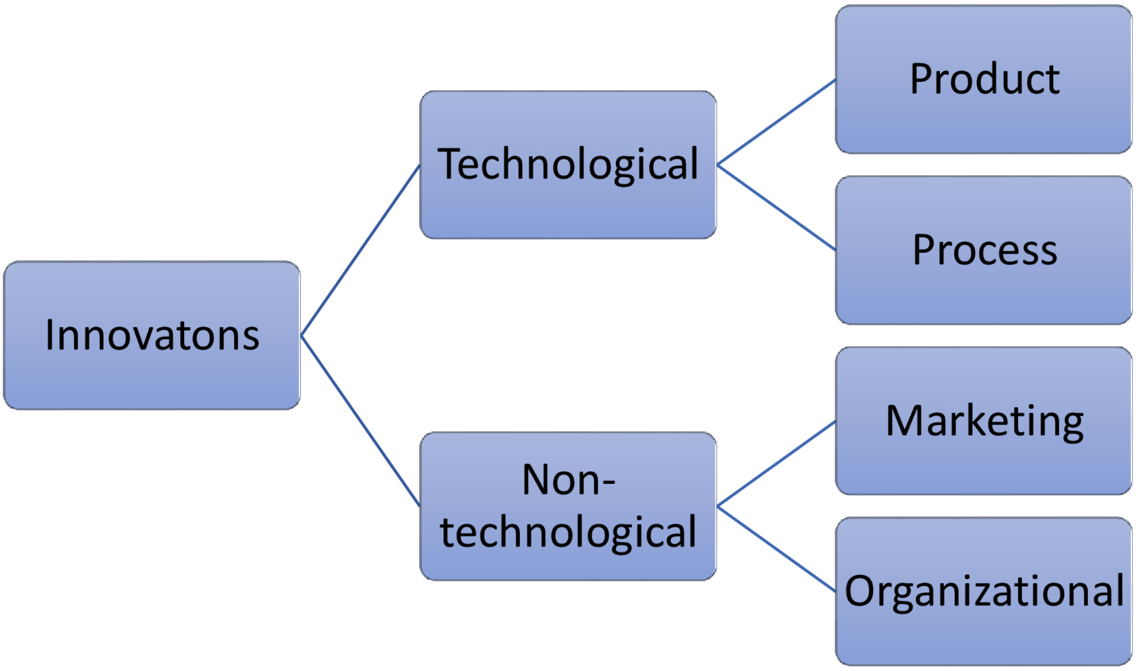


Figure 2: Types of Innovation according to the Oslo Manual

(Source: OECD, 2018)

Innovation is divided into two types of innovation. The first type of innovation is technological innovation, which is then divided into product and process innovation. Product innovation includes both the introduction of new goods and services, as well as significant improvements in the functional or user characteristics of an existing goods or service. Process innovation represents the introduction of a new or substantial improvement in the method of production or distribution. These include significant changes in procedures, technology, equipment or software (CZSO, 2018).

The second type of innovation is non-technological innovation, which includes marketing and organizational innovations. Marketing innovation strives to better address customer needs, open new markets, or place a business product on a new market to increase sales.

Compared to other marketing tools, marketing innovation must be based on the introduction of a new marketing method that has never been used before by the enterprise. The last type of innovation is organizational, expressing the introduction of a new organizational method into business practices, jobs, organizational and external relationships. Taking compared to other organizational changes in the company where such an implementation that has never been used in the company in the past (CZSO, 2018).

It can be revealed from the above that innovation in the company is a key driver for economic development. Innovation is important and gives space, especially in a constantly changing market, where competition with new innovations comes every day. One of these markets is the IT. Innovation in this sector is an integral part of the sector, but its measurement sits not very much into account. Therefore, this dissertation will apply an innovation measurement process called Innovation Scorecard.

2.3.1 Innovation Scorecard

This concept has given space to create ever-increasing demands for new products and services to meet customer needs and requirements. In order for the company to remain competitive, it must respond positively and adopt new approaches to how to become, and above all, how to remain innovative in its outlook.

Innovation Scorecard is not a new project. It was created from a concept known as Balanced Scorecard made by Robert Kaplan and David Norton in the early 1990s. It was created as an attempt to help companies measure their business performance using both financial and non-financial data. Their goal was "*to align business activities to the vision and strategy of the business, improve internal and external communications, and monitor business performance against strategic goals.*" The Balanced Scorecard provides a relevant range of financial and non-financial information that promotes effective business management (Kaplan, Norton, 1996).

Over the years, a new theory has emerged that has taken Balanced Scorecard to a new level, thus Innovation Scorecard. Its primary orientation was directed to innovations that formed part of change management. The essence of the functioning of this model is based on the creation of a framework for measuring performance and management in such a way that it can measure all things of innovation. These two models fit well together and bring clear benefits to enable businesses to cope better and easier control the accelerated range of changes that have recently taken place in different industries (Žižlavský, Fisher, 2019).

The merger of these two concepts was created on the basis of a research project supported by the Czech Scientific Foundation in 2013 – 2015. One of the main tasks of the research was to find out whether organizations in the Czech Republic actually measured effective and efficient innovations. The research also looked at what performance metrics were used, how those metrics were used, and what efficiency they had. At the end of this study, it was a finding that the companies that effectively managed innovation soured important and reliable data on innovative performance, including the advantage of application and innovation management. The correct use of innovations in line with existing corporate strategies gives the space for managers and employees to properly *"plan, organize, monitor and control"* all innovative activities for the benefit of the organization (Žižlavský, Fisher, 2019).

Currently, the main question for many organizations isn't whether to innovate or not, but how to innovate efficiently and effectively. For this reason, it is entitled that organisations are able to continuously evaluate their current innovation projects and use this data to decide whether to continue their projects or not.

2.3.2 Innovation Scorecard Core

Model Innovation Scorecard must happen in an orderly, structured and logical sequence (Figure 3). Only a strictly followed approach can ensure that all characteristics and essentials are respected in this way for an activity such as innovation. When measuring innovation, measurements should be dependent on two conditions: efficient and economic. Often, individual indicators do not meet because they point only to economic status and efficiency are no longer attracted to as much attention (Žižlavský, 2016).

It is therefore important that a comprehensive system with several indicators is used to assess the capabilities and performance of the company. Multiple indicators can examine the innovation process from several sides, thus comprehensively displaying the actual picture of the process. For this reason, the basic Innovation Scorecard structure is designed according to Horvath's log-term experience with the Balance Scorecard model, which can measure the innovative performance of the whole company (Horváth and Partners, 2002).

The basic structure of the Innovation Scorecard implementation process includes the stages shown in Figure 3.

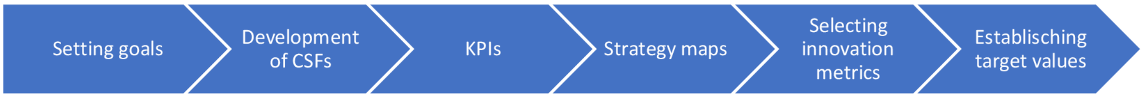


Figure 3: Implementation Scorecard design process
(Source: Žižlavský, 2016)

Setting goals

The first step is to set the objectives of the project and how this objective can be achieved. It is important to think that Innovation Scorecard cannot contain a large number of goals, because there would be confusion and it would not be possible to focus on what is important. Quality objectives affect the functioning of Innovation Scorecard as a whole. A poorly defined target can affect successful execution and implementation (Žižlavský, 2016).

Development of Critical Success Factors (CSFs)

The next step is to identify appropriate CSFs for individual objectives. CSFs represent circumstances and influences that can make a major contribution to the success of the project if properly secured. Identifying critical success factors will lead to monitoring and measuring progress towards achieving. Strategic goals and ultimately to the mission of the company. A CSFs is a high-level goal that is very essential for a business to meet (Žižlavský, Fisher, 2019).

Key performance Indicators (KPIs)

According to Parmenter, KPIs are expressed as “*those indicators which focus on the aspect of organizational performance that are the most critical for current and future success of the organization*” (Parmenter, 2015)

KPIs are a measurable value that shows how effectively a company achieves key organizational goals. KPIs can evaluate the performance of an individual, as well as the performance of the entire organization. This is a form of communication through which we can communicate to stakeholders what is to be achieved. It is important that stakeholders understand what the organizational goals are, how they are planned to achieve them and who can act on the basis of this information. KPIs also provides a focus on operational improvement, creates an analytical basis for decision-making, and helps focus attention on what matters most. Very fitting is Peter Drucker's statement, "*What gets measured gets done*" (Drucker, 2004).

Strategy maps

Another tool you need is implementing a strategic map. This is a diagram used for innovative projects to document the objectives that have been created by the organization. Strategic maps allow you to better transfer targets to operating conditions and align performance with established organisational objectives. The innovation strategy map also helps the organization implement CSFs/KPIs strategies in practice (Kerzner, 2017).

Selecting innovation metrics

Determining metrics is important for measuring innovation processes in society. Metrics serve to clearly express innovative goals, while allowing you to track the extent of their achievement. Measuring innovation goals makes it possible to control the behaviour in the desired direction. It is important that the correct target value is set for each metric. In order to make the measurement of innovation objectives clear, more than two or, exceptionally, three metrics should not be set for each individual innovation target. If a target requires multiple metrics, the target must be broken down. The selection of metrics depends on their effectiveness, i.e. their effectiveness and cost-effectiveness. Efficiency means that it is intended to bring relevant information to corporate management and economy, it must be carried out at reasonable costs (Žižlavský, 2016).

For this dissertation, both financial and non-financial metrics would be used. Innovation scorecard is trying to find a balance between these metrics. Financial indicators show relevant evidence that the company achieves value, which can then be used to inform senior management. Non-financial indicators are much more sensitive to change than financial indicators. They should be defined, so that in the future we can say whether they have changed, whether desirable or not (Davila, et al., 2013).

Establishing target values

The target values describe innovative objectives that are detailed at the beginning of the process. They should be challenging and ambitious, but at the same time credible and real. In general, they should be determined according to the SMART method. Organisational objectives should therefore be specific, measurable, achievable, relevant and time bound.

2.3.3 Innovation scorecard Framework

Developed by Innovation Scorecard by Žižlavský, it was specially designed for to fitting in most environments. It is based on the considerations of Kaplan and Norton (1996), namely the Balance Scorecard approach, where it focuses on the balance between operational and strategic objectives, required inputs and outputs, internal and external factors and lagging/leading indicators (including financial and non-financial) (Žižlavský, 2016).

„*This effective and efficient approach to introduce the concept of Innovation Scorecard into organization is vital for moving innovations from the idea to launch phase in a systematic, managed and controlled way* “(Žižlavský, Fisher, 2019).

As can be seen from Figure 4, the methodology of the proposed Innovation Scorecard is divided into five stages controlled by four gates, where an appraisal is being evaluated as to whether the new product should move to the next gate or be terminated. This system is designed to act as a funnel that starts from the idea to the entire duration of the project, where the head of the innovation project is responsible for meeting all the required criteria before entering the next stage.

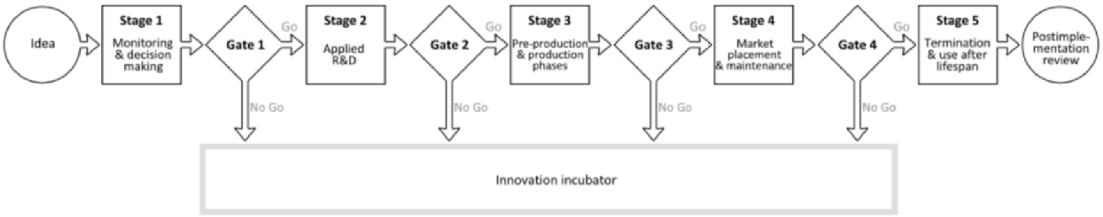

Idea
Stage 1
Monitoring & decision making
Gate 1
Go
No Go
Stage 2
Applied R&D
Gate 2
Go
No Go
Stage 3
Pre-production & production phases
Gate 3
Go
No Go
Stage 4
Market placement & maintenance
Gate 4
Go
No Go
Stage 5
Termination & use after lifespan
Postimplementation review
Innovation incubator

Figure 4: Innovation Scorecard Framework

(Source: Žižlavský, 2016)

The first gate contains the inspiration for measurements related to activities dedicated to identifying ideas for an innovative project. At this stage, it depends on whether ideas have been created from existing sources and whether they come from internal or external stakeholders. The idea at this stage is assessed by whether it is in line with the company's strategy. Ideas sophisticated enough can go to the following R&D stage.

In the second gate, the project is re-evaluated on the basis of criteria from Gate 1 and other variables such as market potential. At this stage, the potential return on the new product is evaluated, followed by a more detailed definition, e.g. the risks, necessary efforts and time horizons. Technical aspects of the product are further assessed, and a detailed financial analysis is carried out. This stage is very important, but it is often neglected.

The third stage is re-testing the product for overall functionality, which includes testing the product and the market, e.g. in the product, preliminary market tests to assess customer reactions. The quality of the measures taken from the second phase is also evaluated in this gate. In the third gate, the product is evaluated the last time, before it is placed on the market.

Gate four implements production and marketing plans for innovation. Innovation is marketed and after 6-18 months the innovation project is terminated as the product becomes a common product. At the end of this phase, the product is evaluated once more. Real performance is being evaluated compared to predictions, e.g. customer satisfaction.

The project is reviewed after implementation in the last fifth gate. Focus is on finding errors that could be avoided in the next project (Žižlavský, 2016).

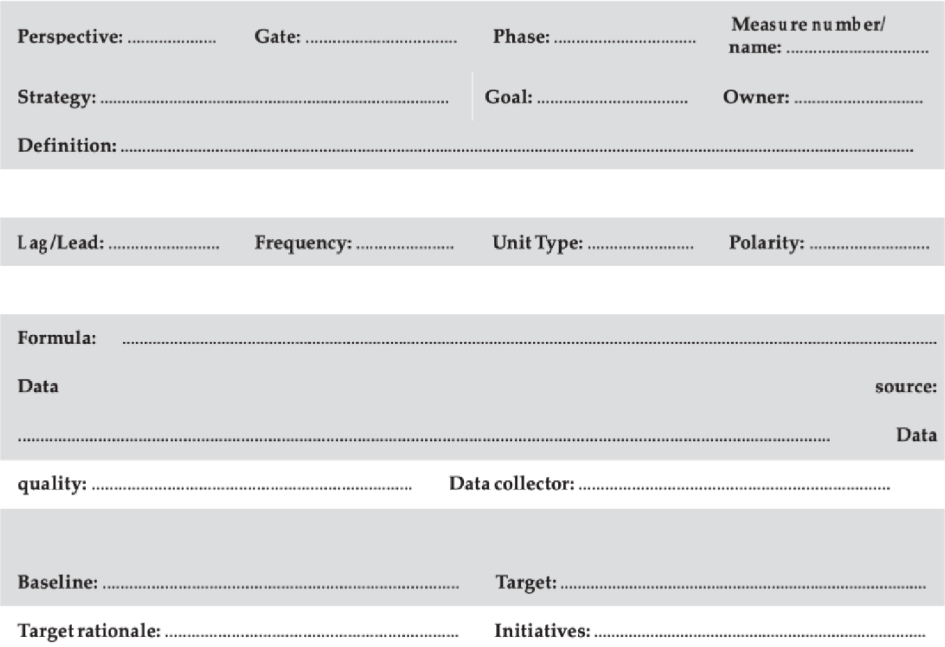
After we have described all the theoretical background, we move to the analytical part. The next chapter of the dissertation will be described the company, where will be implemented model Innovation Scorecard.

2.3.4 Creating Innovation Scorecard Data Sheet

Once possible metrics are agreed for each gate, the next step is to specify the properties of each of them in the Innovation Scorecard Data Sheet, which represents a document that provides all users with a detailed review of Innovation Scorecard measures. including a detailed list of characteristics.

Figure 5 shows the Innovation Scorecard Data Sheet (This framework ensures that the measures are clearly defined and based on an explicitly defined formula and data source.

Based on Niven's (2014) work, there are four parts of the template that must be finished. In the first section, shown at the top, employees provide essential background material on the measure. The second one lists specific measure characteristics. Calculation and data specifications are outlined in the third component of the dictionary. Finally, in the bottom section, space is provided to outline performance in formation relating to the measure (Žižlavský, 2016)



Perspective: .... Gate: .... Phase: .... Measure number/ name: ....

Strategy: .... Goal: .... Owner: ....

Definition: ....

Lag/Lead: .... Frequency: .... Unit Type: .... Polarity: ....

Formula: ....

Data source: .... Data quality: .... Data collector: ....

Baseline: .... Target: ....

Target rationale: .... Initiatives: ....

Figure 5: Innovation Scorecard Data Sheet
(Source: Žižlavský, 2016)

Measurement background

Perspective: Displays the perspective (financial/customer/internal process/learning and growth/innovation) under which the measure falls.

Gate: Displays the gate (idea screening/project selection/innovation preparation and market test/analysing test market results, after-launch assessment/post- implementation review) under which the measure falls.

Phase: Displays the phase (input/process/output/outcome) under which the measure falls.

Measure Number/Name: All performance measures should be provided by a number and name. The name of the measure should be clear. A good name is one that explains what the measure is and why it is important. It should be self- explanatory and not include functionally specific jargon. The number is important

as well and should managers later choose an automated reporting system. Many will require completely unique names for each measure, and since managers may track the same measures at various locations or departments, a specific identifier should be supplied.

Strategy: Displays the specific strategy that will positively influence the measure.

Goal: Every measure was created as a translation of a specific objective. Use this space to identify the relevant goal. The aim of the Innovation Scorecard requires succinct and simple formulation. It is however often necessary to add detailed explanation to it – a legend that facilitates the clarification and communication of the significance and background of individual goals. For this reason, for each goal a short commentary should be prepared (three to four lines). First and foremost, there needs to be an explanation of why the goal is seen as significant. This should avert the danger that is a few weeks or months later discussion as to what actually the intent of this is or that innovative goal.

Owner: The person who is to act on the data should be identified. The owner is the individual responsible for results. Should the indicator's performance begin to decline, it is the owner we look to for answers and a plan to bring results back in line with expectations.

Description: After reading the measure name, most people will immediately jump to the measure description, and it is therefore possibly the most important piece of information on the entire template. Challenge is to draft a description that concisely and accurately captures the essence of the measure so that anyone reading it will be able to quickly grasp why the measure is critical to the company (Žižlavský, 2016).

Measure characteristics

Lag/Lead: Outline whether the measure is a care outcome indicator or a performance driver.

Frequency: The frequency with which performance should be recorded and reported is a function of the importance of the measure and the volume of data available. Most companies have measures that report performance on a daily, weekly, monthly, quarterly, semi-annual, or annual basis.

Unit Type: This characteristic identifies how the measure will be expressed. Commonly used unit types include numbers, currency (Euro, USD, CZK, etc.), and percentages.

Polarity: When assessing the performance of a measure competent managers need to know whether high values reflect good or bad performance. In most cases, this is very straightforward. We all know that lower costs and increased employee satisfaction are good, while a high value for complaints reflects performance that requires improvement. However, in some cases the polarity issue can prove quite challenging.

Calculation and data specifications

Formula: The formula box provides the specific elements of the calculation for the performance measures.

Data Source: The source of the raw data should be specified. The importance of this question lies in the fact that a consistent source of data is vital if performance is to be compared over time. In this section employees should rigorously attempt to supply as much detailed information as possible. The more information provided here, the easier it will be to begin actually producing Innovation Scorecard reports with real data. However, if employees provide vague data sources, or no conformation at all, managers will find it exceedingly difficult to report on the measure later.

Data Quality: This area of the template should be used for comments on the condition of the data expected to use when reporting Innovation Scorecard results. If the data is produced automatically from a source system and can be easily accessed, it can be considered high quality and vice versa (Žižlavský, 2016).

Data Collector: The person who is to collect and report the data should be identified. In the first section of the template the owner of the measure is identified as that individual who is accountable for results. Often this is not the person expected to provide the actual performance data.

Performance information

Baseline: Users of the Innovation Scorecard will be very interested in the current level of performance for all measures. For those owning the challenge of developing targets the baseline is critical in their work.

Target: Following the Innovation Scorecard methodology, target values should be established in the fifth phase. For those measures that do not currently have targets, this section could be left blank and completed once the targets have been finalised. In this example, some companies may find it difficult to establish monthly or quarterly targets and instead opt for an annual number, but track performance toward that end on a monthly or quarterly basis.

Target Rationale: As above, this will only apply to those measures that currently have a performance target. The rationale provides users with background on how the particular target(s) has been arrived at. For people to encourage the achievement of a target they need (Žižlavský, 2016).

3 Current Situation and Analysis of Problem

This section provides a detailed overview of Red Hat and its current operations, including a detailed description of the company's history, customers and provided products. It was necessary to analyse Red hat's internal processes in order to better understand how business functions and operates. The essence of this part of survey is to identify any gaps that may exist in current business processes that would be beneficial for the adoption of an Innovation Scorecard system.

3.1 Red Hat

Red Hat is the world's leading provider of open source and Linux operating systems. The headquarter is located in Raleigh, North Carolina, USA. Currently the company has more than 60 branches in 28 countries and more than 2,500 employees in the world. Red Hat focuses mainly on Linux and open-source solutions, Middleware, applications, system management solutions, training consulting services and the support of its customers around the world.

Red Hat's mission statement *"to be the catalyst in communities of customers, contributors, and partners creating better technology the open source way"*. It also emphasizes the outing of the community, where everyone should be fully encouraged to use their voices and talents. The company's vision is to become a defining technology company of the 21st century (Book of Red Hat, 2019).

3.1.1 History of Red Hat

The name Red Hat was first used in 1993. It was at a time when a gentleman with the name Bob Young formed a corporation called ACC in 1993. At that time, ACC sold Linux and Unix software add-ons. In 1994, another gentleman by the name of Marc Ewing began distributing his personal version of Linux which he named Red Hat Linux. The creation of Red Hat Linux took place in October 1994 and it became known as the "Halloween Release". The following year, after the release of a successful version of Red Hat Linux, Young bought Ewing's business and then merged it with his ACC corporation, creating a new company called Red Hat Software.

The name Red Hat was originated by Marc Ewing. He often worked in a computer lab in a red lacrosse cap, which he had from his grandfather. He was known for his willingness and ability to share his knowledge. It was even said *"if you need help, look for the guy in the red hat"* this phrase has been transferred to the Red Hat logo, where is also red cap (Figure 6).

Figure 6: Logo of Red Hat

(Source: Book of Red Hat, 2019)

The first major success that led the company to public awareness was in 1999, when Red Hat made eighth biggest gain in Wall Street history. In 1999, Red Hat bought Cygnus Solution, making it the largest opensource company in the world. Cygnus provided commercial support for free software and brought together software programmers. One of the founders of Cygnus, concretely Michael Tiemann, became chief technical officer of Red Hat and until 2008 served as a president of open source affairs. In addition to Cygnus Solution, Red Hat has acquired other companies such as WireSpeed, C2Net, and Hells Kitchen Systems (HKS). With these acquisitions, Red Hat wanted to secure the following benefits;

- embedded devices to the internet (WireSpeed),
- web server security through StrongHold, which was the first commercial server which supported and made available source code to the development community (C2Net),
- payment processing software that is a necessity in the e-commerce (HKS).

The year 2002 was particularly significant as it marked the launch of the first Enterprise version called Red Hat Enterprise Advanced Server, later renamed as Red Hat Enterprise Linux (RHEL). Dell, IBM, HP, and Oracle had also announced support for this platform. In 2006 and 2007, Red Hat acquired JBoss and MetaMatrix.

Another significant event for Red Hat was the expansion of Raleigh's headquarters in 2011, which was recorded in two stages. The first stage included addition of 540 employees and an investment in operations in Raleigh of 109 million USD. The second stage was expansion into new technologies such as visualization software and cloud technology offering.

The turnaround came in 2011, in which the company achieved annual sales of over $1 billion. It was the first company in the history of IT and Software Development that achieved such revenue levels purely based on income from open source software. In 2016, Red Hat reached another milestone, with revenue surging $2 billion.

In addition, another change occurred in 2019, when the world's second largest "technology deal" was reached. IBM's acquisition of Red Hat, under which IBM acquired Red Hat shares at a cost of approximately USD 34 billion. *“Joining forces with IBM gives Red Hat the opportunity to bring more open source innovation to an even broader range of organizations and will enable us to scale to meet the need for hybrid cloud solutions that deliver true choice and agility”,* said Jim Whitehurst, CEO President, Red Hat. Red Hat will act as a separate unit within IBM and will be announced as part of the IBM Cloud and Cognitive Software segment.

3.1.2 Red Hat Products

Red Hat products are Linux platforms, Middleware, Virtualization platform, Cloud computing, Storage, Management and Available Services.

The first product to be presented is **Linux platforms** that leading alternative to open source for a reliable modern IT platform that has the capability to deploy applications to Virtual Machines, hardware and cloud environments. This platform increases efficiency, is easy to manage, control and support all major hardware platforms and thousands of applications. These are based on open standards and functional modules that enhance Red Hat Enterprise Linux's (RHEL) management capabilities.

Some software known as **Middleware** (its main purpose is to glue together separate, complex and existing programmes) acts as a bridge between an operating system or database and applications, particularly in network environments. Middleware provides a portfolio of products and components for creating, integrating, and automating modern business applications and processes. With Red Hat Middleware, organizations can accelerate the development and delivery of business solutions to spend more time to innovate and maintain their competitive advantage. Data management, application services, messaging and authentication are addressed through middleware. It helps developers build applications more efficiently. It is a so-called "black hole" bridge between applications, data and users.

Another product named as **Virtualization platform** is a technology that enables to create useful IT services using resources that are traditionally limited to hardware. It allows to use the full capacity of the physical machine by spreading its capabilities among the many users or the operating environments.

Additional production offered by Red Hat is a **Storage.** Through storage information technology archives, organizes, and shares bits and bytes. Data storage is a central component of big data. The short-term memory is handled by random-access memory (RAM). There are many types of storage such as cloud storage, object storage, file storage and block storage.

Cloud Computing is the availability of computer system resources on demand, especially data storage and computing power, without direct active user management. It stores everything that works in the network cluster such as data, application and services. Administrative tools are typically run as platforms that represent software used to manage this data, service applications. There are four kinds of clouds: public, private, hybrid and multi clouds. The private cloud is specifically and exclusively “run” for one organization, regardless of whether it is controlled internally or by a third party. It is very demanding and expensive for maintenance and requires constant recovery, otherwise there is a risk of serious vulnerabilities. Organizations use this cloud if they don't want sensitive data to be accessible to external users. The public cloud provides services offered to an external provider over the internet so that they are available to anyone who wants to buy them. Unlike the private cloud, there is no need to spend large amounts of money on management and maintenance. This is managed by the cloud service provider. The private and public cloud combination demonstrates a hybrid cloud that enables data and application sharing between the private and public cloud. This cloud is used by those companies that often undergo short-term fluctuations in demand. Thus, companies pay only for resources that they temporarily use. There is another cloud option called multi cloud. This cloud consists of more than one cloud service from more than one cloud vendor, whether public or private. The purpose is to eliminate dependence on any provider or cloud instance. Companies choose from these four clouds, depending on individual cost, performance, reliability, and deployment needs.

The maintenance of the infrastructure is focused product **Management**, which contains several components for their streamlining. The first is Smart Management representing software that includes cloud management tools supported by RHEL. The following software named as Satellite helps users provide, configure, and update the Red Hat infrastructure by automating most system maintenance tasks. Infrastructure management can't do without a comprehensive CloudForms platform that provides security for virtual and cloud infrastructure that can easily deliver services across all cloud environments. The Ansible Automation Platform is also a tool for managing infrastructure, which is basis for building and operating automation throughout the organization. The last tool for managing infrastructure is Insights, which allows to predict and prevent problems before they occur.

The last product is **Available services** that is offered by Red Hat. One such service is known as Open Innovation Labs. Red Hat experts teach others how to use agile methods and open source tools to work on their enterprise's business problems with a view to resolve these in the most efficient and cost-effective way. In addition, there is a training certification course, which helps people to understand and then master Red Hat technologies. Those who achieve the necessary pass rate following the exam at the end of courses will receive a certificate of competence. Red Hat also offers consultations with strategic advisers to analyse current problems in their organizations and help staff to overcome these by applying comprehensive and cost-effective solutions.

3.1.3 Red Hat customers

Currently, Red Hat services are no longer used just for the IT sphere, but they are also used in Financial Services, Healthcare and Life Sciences, Government and the Public Sector and Telecommunications.

The **Financial services** sector is in a very competitive environment where speed, safety and dexterity are critical. With the use of Red Hat services, financial services can optimize business processes, modernize technology and evolve towards a more agile culture. Among other things, it can better manage more complex technologies, reduce risks and maintain compliance.

Many foremost **Healthcare companies** are dependable on Red Hat, that rely on protecting their sensitive data. These organisations must ensure that this huge amount of data is protected and that it complies with the Health Insurance Portability and Accountability Act (HIPAA) and other standards. Red Hat joined forces with OpenSCAP (leader in open source security protocols) to provide a solution to help prevent and mitigate security threats.

Government leaders are expected to promote efficiency and innovation, hire and maintain top talent, and implement effective IT initiatives. The key to achieving these goals is to adopt an open and agile culture in government organizations. That is why Red Hat designed products to help them streamline and meet critical IT requirements and stabilize the best innovations from open source communication to public sector mission.

Traditional operators in **Telecommunications** service must "reinvent" themselves to thrive in this ever-changing market landscape. Red Hat offers an open platform for the industry to help service providers deliver innovative and faster new services to the market safely and efficiently.

3.1.4 Red Hat Enterprise Linux (RHEL)

All of the above-mentioned Red Hat products form part of the Red Hat Enterprise Linux platform. This platform represents the Linux open distribution developed by Red Hat, which is focused on the commercial sphere. RHEL is the result of a collaborative development process that originated in open source communities. It works with community members, customers, and even competitors in thousands of downstream projects before integrating the best features and bug fixes into the Fedora Linux distribution, which is made up of Red Hat Enterprise Linux. This process leads to stabilization of open source functions. The latest version of Red Hat Enterprise Linux 8 was released in May 2019. For each new version, updates are issued twice a year.

Figure 7 shows that RHEL is divided into four footprints. The first footprint named as **Bare metal** is the commander of all computers which can connect to any hardware or software with the editing of the source code. The following **virtualization** footprint consists of a native RHEL component that contains operating system components (process schedulers, security administrators, and more) that needs to create and run virtual machines (VM). With the aid of a virtualization machine, the application written for one operating system (OS), can be used on a machine running another OS. Each VM is implanted with dedicated virtual hardware such as central processor units, memory, and disks. Next footprint is **containers** which are like a box that helps to focus only on the part that is being worked with to separate the area of responsibility. The last important footprint is **cloud**, where all clouds are unique and requires a flexible OS like Linux. RHEL creates a stable and consistent structure that stretches over these 4 traces of IT, regardless of the underlying hardware, service, or provider.

Currently, RHEL uses over 40,000 companies, most often in the United States. Red Hat customers are aware of the value of open technology, so they use Red Hat products to overcome the big challenges. Customers also have added value in keeping costs low and open options.

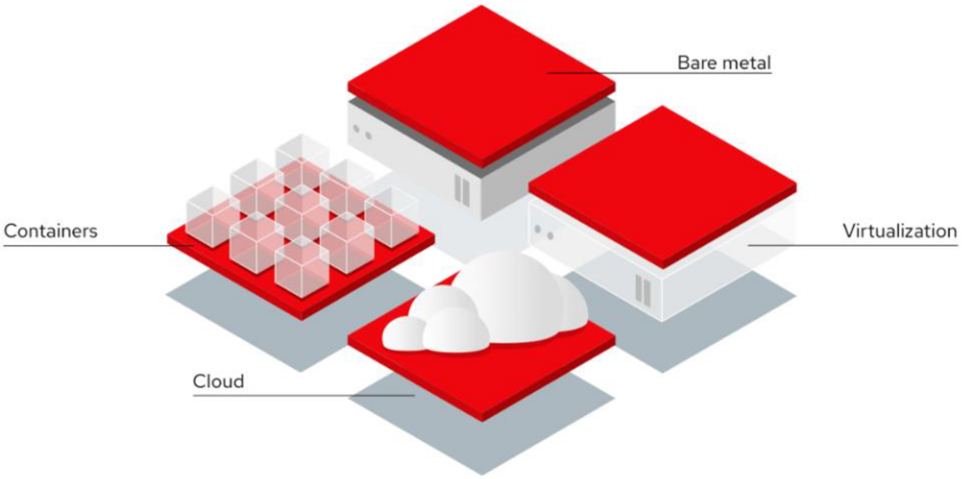


Figure 7: How the RHEL works

(Source: Red Hat, 2019)

Looking at RHEL customers by industry, the Figure 8 shows that the biggest segments are computer software and hospital and health care.

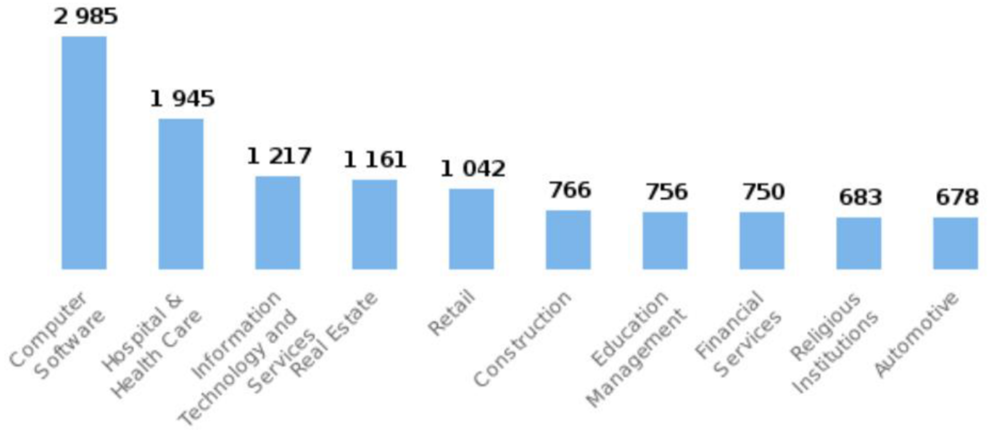


Figure 8: Distribution of companies using RHEL server by industry

(Source: Enlyft, 2020)

3.2 Continuous Integration (CI)

This part describes the Continuous Integration process will be explained in sufficient detail including how it works in a practice and what CI tools Red Hat used.

3.2.1 Continuous Integration Background

The current long-term target presented by Red Hat for the Continuous Integration and Continuous Delivery (CI/CD) project is based on providing the customer with a response to the new code as quickly as possible. However, in order to achieve this objective, CI process must form reliable testing of embedded new codes. CI is important to provide rapid feedback if the defect is introduced to the code based, could be rectified and correctified as soon as it is possible. In order to conduct deeper tests by CI process that have greater reporting value. After CI provides a good quality code that shows no negative feedback, the code can be delivered by the CD process to the customer. As this is a long-term goal, it needs to be achieved step by step, so this thesis will focus on Continuous Integration improvements. In the future, there is also a planned CI/CD project that will move Red Hat closer to achieving their long-term goal.

3.2.2 Process

Continuous Integration is a process for finding errors quickly and locate these more easily. As each change introduced is typically small, pinpointing the specific change that introduced a defect can be done quickly and efficiently. The practice of merging everyone's code together including "build it and test it", is conducted several times per day. CI ensures the timely detection and eradication of software defects. Defects in this sense are those defects that could otherwise not be detected until days, weeks or even months after they were created. Preliminary detection of defects at the beginning of the development process can translate into lower costs and shorten the timeline (Heller, 2020).

3.2.3 Continuous Integration process in Red Hat

In order to understand why Red Hat wanted to introduce changes, it is necessary to describe how the CI process works and what problems occurred by default. As part of the Continuous Integration process, Red Hat uses a distributed versioning system known as GitHub to record changes in files. Git is version control system that allows to work together with other developers. With git it can be seen what others are working on, review their code without traveling a thousand miles. Unlike centralized systems, the user is not downloading only the latest version of the file but represents the entire repository. In a collapse (this term is used in software development and means, in this context, to compress or shorten a hierarchy so that only the roots of each branch are visible), the data can then be restored from the user. Red Hat approach is to share the project details, for example, with developers and it is for this reason that they use GitHub. GitHub is automatic and can remember any changes that have occurred. The issue with GitHub is that it is an automated system and there appears to be some room for further development to improve this system. If a new code is generated (Figure 9) GitHub takes a "picture" of all the files at any given time and save references to that snapshot. If there is no change in the code, it does not create a new record, but merely refers to the previous identical record that has already been saved. The next step in the CI process is to perform testing. Testing is a process for obtaining information about the characteristics and status of a system to determine if the system is operating as specified in the proposal.

A typical test performed on integration servers is the Unit test. If there were some change in any of the used codes, the test would start automatically. This test provides detailed feedback on how the code works. If the code passes the unit tests, the "Triggers Build" runs, which allows to set how often the task will run. Then all what have to be done is run the tests. The exact path depends on the selected names in the project. If it is misspelled, the task will end up in error.

The last step before final saving is adding a “Post-Build” step, where needs to be specify the output (e.g. email address) where the results should be sent. After this step, it is determined whether the code is active or not. If it is negative, GitHub is update. In case it is active, it goes to the last stage of testing using “end to end” (E2E) tests. Running E2E testing enables the identification of any complex code functionalities.

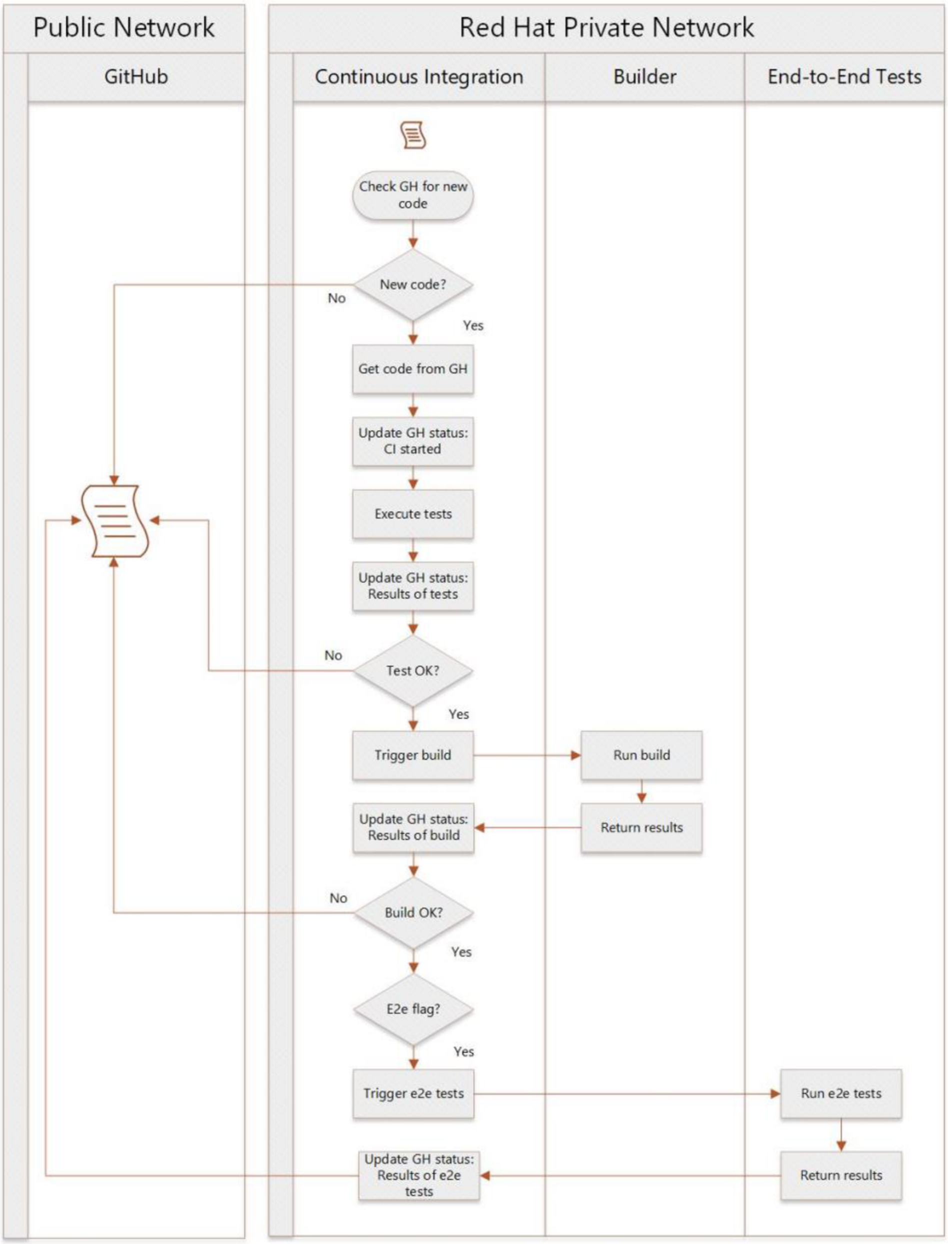


Figure 9: CI process for the Leap project in Red Hat
(Source: own processing)

3.2.4 Continuous Integration Tools

Once, programmers used to be solely responsible for integrating their own code, but now CI tools running on a server handle integration automatically. These tools can therefore be set up to be created at scheduled intervals or when the new code enters the storage location. Some CI tools can even automatically generate documentation to help you control quality and version management. Choosing a suitable tool for CI depends on the size of the company, the type of product offered, the goals and the overall philosophy of the company. Among the most famous instruments for continuous intermigration are:

- Buddy
- TeamCity
- Jenkins
- Travis CI
- Bamboo
- GitLab CI
- Circle CI
- Codeship

Two CI tools – Travis CI and Jenkins – will be relevant to this particular thesis.

Jenkins

Jenkins is an award-winning tool for continuous open source integration that is able to organize a string of actions so that it can support and improve the continuous integration process, including automation-related processes. Its main benefit lies in the ability to accelerate the software development process through automation. Jenkins supports a complete life cycle of software development from building, documenting, testing, deploying and other stages of software development lifecycle. Jenkins is a Java-based tool, which means that only Java Runtime Environment is required to run it. For this reason, Jenkins can be installed on any operating system where Java is being used. Jenkins' installation is free and is used worldwide by more than 30,000 users and grows every day (Heller, 2020).

Travis CI

Travis CI is a very popular service for automating Continuous Integration (CI). Travis CI was the first who introduce a new approach to creating code in the cloud. This CI tool allows the user to log in, link their storage, and then create and test their applications. Travis CI is provided free of charge and offers automatic run tests after writing changes to the repository. Travis CI is easily integrated into common Cloud storage sites such as GitHub and Bitbucket. It offers a lot of automated CI options that eliminate the need for a dedicated server because Travis Ci is hosted in the cloud. This allows to test in different environments, on different machines and on different operating systems (Maskovsky, 2014).

The following Table 6 provide an overview of the main differences between Jenkins and Travis CI. Table shows that Travis is much less maintenance intensive than Jenkins. Travis CI is more suitable to work on an open source project. If a new private business project is already developing, having its own server is a more acceptable alternative to Jenkins. The next section explains why Jenkins and Travis CI are important for the CI innovation process in Red Hat.

Table 6: **Comparison of Travis CI & Jenkins**
(Source: Heller, 2020)

Baseline	**Travis**	**Jenkins**
Free download of source code	yes	yes
Initial installation	Minimal	Requires extensive setup.
Hosting	Free, but if the project is private, it is paid for the business plan.	It requires developers to run and maintain their own hosting server
Hosting maintenance	Minimal configuration.	It must run its own server constantly, otherwise Jenkins does not work
Performance	Comparable	

3.2.5 Issues in the CI process

Many errors happen within the Continuous integration process. This requires a lot of time to repair the errors. The Leapp team documented the known issues and this led to the development of new ideas how to resolve these. One problem is the lack of disk space. Almost all disk space is consumed by old builds, so there is no space for new ones. When Jenkins detects that the disk has run out of capacity, copying and E2E tests that merge the system with the public are blocked. This block must be fixed by freeing up disk space. Given the growing number of new external contributors, building constraints for each task do not help, so the severity of this problem is high. Another problem, representing medium severity is "Post-Build" step, as described in the CI process, represent output where E2E test results supposed to be sent. If anyone from the team wants to know the results of some E2E test, its needed to be checked in Jenkins manually.

The following medium severity issue is related to GitHub status. When a job fails before any test is executed, GitHub status is not updated and is stuck on “pending”. This problem often occurs when is an error in the code. By looking at the public network, it is not clear what the problem has occurred, and it is necessary to be check in Jenkins, which affected build needs to be consulted to know what is happening. One of the biggest problems is the time spent by engineers to continually fix any errors in the system. Only one engineer is deployed for this work and is not compensated for in case an inability to work. Test results need to be made available to anyone who opens the pull request, particularly in the case of external contributors. This is not yet possible to implement because the unit tests run on an internal infrastructure, so it is impossible to publish these tests for security reasons. This issue needs to be fixed as it is time consuming and this in turn could also save money. The iScorecard team joined the project at this moment in time. The conceptual performance of this team has the future potential to improve efficiency and ultimate competitiveness within the IT and Software Industry in companies other than Red Hat.

4 Proposed Solution, Contribution of Proposed Solution

The issues with Continuous integration described in the previous chapter 3.2.5 creates a change request for an upgrade in the form of an innovation of the process that could solve or at least eliminate the issues addressed. These problems need a lot of maintenance time. This is why Red Hat decided to work with the iScorecard team so that this team can help to resolve this issue. The iScorecard Innovation team adopted the Innovation Stage gate model by Copper (1998) for the ongoing project management of the CI project. In addition, the performance measurement system design by Žižlavský (2016) was also adopted for the purpose of managing this project. The theoretical background for this framework is described in chapter 2.3.3.

4.1 Logical Framework Approach

As this is a concept of the project, the preparatory phase of the project cannot be neglected either. This phase involves the creation of basic assumptions for the implementation of the project. For that reason, at the first meeting with the Project Manager (PM) and Leapp team, it was agreed that the preparatory phase of the project would include the creation documents. Some of these will be presented in the next section.

4.1.1 The Project Definition Document (PDD)

The document defines how the CI project will be managed. The final version of the PDD was subsequently passed to the Leap Team and PM so they could review and update the document as was necessary. For the purposes of the thesis, only important parts have been selected from this document. The sufficient components listed in the following paragraph are Objectives, Critical Success factors (CSFs) and Communication Flow.

Objectives

The following objectives have been set for the Continuous Integration project;

- make the process easy to update and fit for its intended purpose,
- reduce or minimize maintenance,
- improve the speed of managing filing issues,
- reduce engineering input time.

Critical Success Factors (CSFs)

Following a discussion between the Red Hat and the iScorecard team, two critical success factors were evaluated to get all team members to focus on what really matters. The first factor is to reduce the overhead of the CI process. The second factor is to improve the current development and maintenance modus operandi.

Communications flow

There will be a Continuous Communication flow in this project between Red Hat and the Innovation Scorecard team (Figure 10). Regular interactions will take place in the form of face to face, telephone and electronic communications. The Red Hat project manager is in charge of a team of 9 people. The Red Hat deputy project manager looks after a team known as the LEAPP team. Red Hat advised that their current team structure could change in line with changing operational requirements. The iScorecard team, made up of members of Brno University of Technology and an external PM specialist consultant, work with the Red Hat Project Managers. All lines of communication follow this hierarchical structure. The flow of communication (right information at the right time in the right format to the right people) forms an important part of this project.

The following communication methods were applied in the project:

- Meetings (regular and irregular) with Red Hat and the iScorecard Team
- E-mails
- Monthly Project Progress Reports (Appendix 1)
- Formal and Informal Verbal, Written and Visual Communications
- Workshops, Presentations and Publications
- iScorecard Website

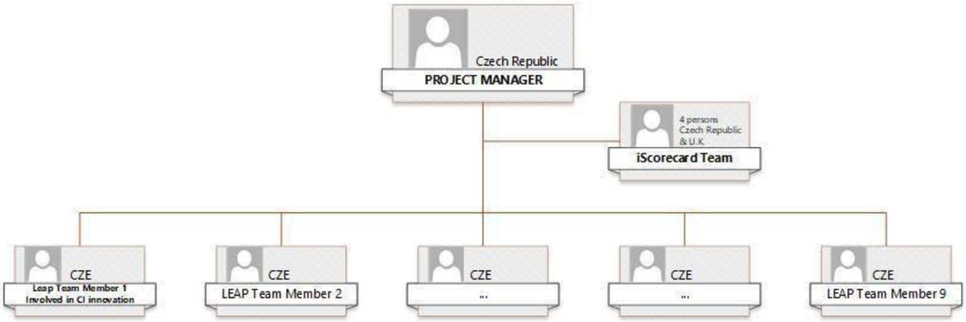


Figure 10: Communication Flow Diagram for the iScorecard Project

(Source: own processing)

4.1.2 Action list, Risk and Issues Register

The Team iScorecard has a weekly call to check the project's progress and assess what needs to be done during the following week. All call information is recorded in the project action list. This document contains the work that the iScorecard team has done, including data, task owner, detail, and term (Appendix 2) The table describes what's happening and possibly where the problem is. This and other documents operate on the basis of traffic light systems known as RAG (Red, Amber and Green). Red highlights significant issues that the project team cannot be solve by own and will need help to solve problems. Amber is used to highlight some issues, for example, that doesn't go all according to plan. And the last green means that everything is fine, and the project goes according to plan. The role of RAG in this project is based on project status identification and reporting.

The Risk and Issues Register has also been created for this project because each product development project involves uncertainty about what will happen. The success of the project can be ensured through risk management. When something goes wrong and doesn't go according to plan, it ceases to be a risk and becomes an issue that needs to be solved to ensure success. In (Appendix 3), it can be seen that even in this project the risk has become a problem, therefore it has been moved from the risk register to the Issue registry (Appendix 4).

In addition to the named documents, other documents were created during the project. These were the Project Schedule (Appendix 5), Scorecard Guide, Change Control Process and Minutes of meetings.

4.2 Innovation Scorecard Design

The Innovation Scorecard concept developed by Žižlavský was specifically designed to fit most working environments. It is based on the considerations of Kaplan and Norton (1996), the so-called Balanced Scorecard approach, where it focuses on the balance between operational and strategic objectives, required inputs and outputs, internal and external factors and lagging/leading indicators (including financial and non-financial) (Žižlavský, 2016).

The CI innovation process is divided into distinct stages. This includes a number of management decision gates. The adoption of this process ensures that innovations can be managed in an efficient as well as efficient way. It is therefore possible to manage the project from idea through implementation and close-down in a systematic and controlled way.

The adopted stage gate process divides the innovation process into different stages and gates. It was decided at a meeting with Red Hat on 6th August 2019 to combine the previous pre-implementation and implementation stages under the umbrella of the combined SCRUM and 2-weekly SPRINT approaches. This meant that the previous Stage 3 and Stage 4 were combined, resulting in the following 4 – Stage process:

- Stage 1: Idea generation
- Stage 2: Idea Development
- Stage 3: Implementation
- Stage 4: Post – Implementation

4.2.1 Stage 1: Idea generation

Idea generation represents a collection of ideas designed to improve the Continuous Integration (CI) process. These ideas are analysed in detail to check and confirm if they meet all the right criteria to be considered for adoption. Red Hat created a document called "Leapp Continuous Integration Status". The main purpose of this document was to summarize the advantages and disadvantages of all considered ideas. The next step included an analysis of the chosen ideas and to check if these ideas actually met Red Hat's requirements of "what makes an acceptable new idea to achieve innovation". Further checks then confirmed if any considered ideas could actually be realistically implemented. Any ideas considered not fit for their intended purpose and not meeting Red Hat's requirements were discarded. Any idea that did not pass Stage 1 of the adopted Stage Gate process would not be moved forward to Stage 2. The iScorecard team presented a number of suitable and fit for intended purpose measurement metrics to Red Hat staff at a project review meeting held on 2nd August 2019. The suggested metrics were discussed and agreed during a project review meeting with the Red Hat team on 2nd August 2019. Several metrics were adopted as suitable to form the core of the Gate 1 performance and outcome measurements. Following on from some further and final discussions with the Red Hat team, the final set of metrics, most suitable for use in the Continuous Integration process, were selected and adopted by both the Red Hat and the iScorecard team. Table 7 is a summary of the adopted metrics and the associated targets that each metric will measure, with the main focus on how successful each innovation has been. The quality of effectiveness and efficiency of the current process were measured by conducting face to face interviews.

Table 7: **Metrics for Gate 1**
(Source: own processing)

	Metric number/name	Target
Inputs	G1IO1 – Work Effort for Given Tasks	iScorecard max 0.8 Full Time Equivalent (FTE) Red Hat max 0.25 FTE
	G1I02 – Quality of current CI process	Minimize blocks
Process	G1P01 – Time of systematic idea generation and evaluation	Max. 1 week
Outputs	G1O01 – Quality of generated idea(s) – percentage of the problems solved by each idea generated	Min. 75%
Results	G1R01 – Milestone/Deadline	9th July 2019
	G1R02 – Total cost of idea generation phase	Maximum 50,000 CZK

Inputs

G1IO1 –Work Effort for Given Tasks

This metric is designed to help determine how effectively and efficiently human resources have been deployed and how productive and successful the application of the iScorecard team's innovation stage gate process has actually been. The main purpose for deploying and tracking this metric was to focus on the responsibilities of project team members and to establish how they were able to execute their clearly defined responsibilities, using typical project management performance criteria such as time, cost and quality. In addition, the following FTE metric (Table 8) was agreed at a follow-on meeting with the Red Hat team.

The earlier referred to and adopted "Leapp CI Status" document was created during the Idea Generation phase of this project. It took two team meetings to develop and finalise this document. The first meeting lasted approximately 10 minutes and was mainly concerned with some initial "in principle" agreements what information the document should contain and how this document should be used. The second meeting appeared to be much more productive. It lasted for just an hour and the teams involved focused their attention more on, for example, creating error lists (also known as bug lists). This list contained relevant details of all occurring errors within the CI project. In the past, it took one engineer nearly 4 hours to produce the same information. It was evident from the outcome of this one exercise that the introduction of innovative ways of working already showed early positive results that suggested what the potential resource savings in Red Hat could be at the end of this project, based on the results of measuring how successful this one innovation has been. Overall, this iterative approach helped to produce a valuable document that summarized the identified CI project issues and how to mitigate/resolve the issues. This was of great help to both the project manager and Red Hat's Leapp Team. In parallel, the iScorecard team was already working on creating a necessary document for the CI project: A Project Definition Document (PDD). This document describes high details of how this project will be managed, what assumptions have been made and what the expected outcomes of the project are, to name but a few headings. The PDD also described in sufficient detail the suggested metrics for each of the process Gates.

Table 8: **Work Effort for Given Tasks**
(Source: own processing)

Baseline (FTE)	**Inception**	**Termination**
PM	0.03	0.05
Leapp Team	0.13	0.20
Red Hat Team	0.00	0.00
Red Hat Summary	**0.16**	**0.25**
Team Leader	0.30	0.30
Professor	0.30	0.20
Student	0.20	0.30
iScorecard Team Summary	**0.80**	**0.80**

Process

G1I02 – Quality of current CI process

This is one of the core metrics that have been identified for application within the CI project. Its efficiency will be evaluated in two stages: before and after the CI innovation implementation and then at quarter-yearly intervals. Measures that will be applied include maintenance time, number of blockages and average time spent on managing blockers. Table 9 is a good example of how much time both the Leapp and Red Hat teams spent over a given period of time to fix errors/bugs including any duplication of effort due to the way the teams were working at the time. Some errors/bugs were of such a nature that it was not possible for the teams to follow the CI process as the errors/bugs needed to be fixed before they could do so. It was frustrating and very time consuming. Some so-called "features", associated with software development work, also slowed down the teams as far as the CI process was concerned. As can be seen from Table 8, bugs occur more often than features. In this example, eight bugs were recorded during this reporting period and it took the Leapp team 180 hours to repair these, plus the Red Hat team spent an additional 20 hours to do the same. In contrast, there were only two features the Leapp team needed to fix and this took them 32 hours in total. Both areas provide ample opportunities to reduce the time it takes to fix bugs/manage features. This is where the innovation scorecard and associated metrics will make a substantial difference in terms of reducing error/bug fixing times and proving how successful these changes have been compared to the current way of working in both the Leapp and Red Hat teams.

Table 9: Lists of errors from January to May
(Source: own processing)

Name	Frequencies	Leapp team (hrs)	Red Hat team (hrs)	Blockage (hrs)
Bugs	8	180	20	200
Features	2	32	0	32
Summary				**232**

G1P01 – Time of Systematic Idea Generation and Evaluation

This metric (Table 10) was developed to assess and measure how much time it took to generate new ideas that could successfully solve identified issues and problems within the CI process. It was an absolute necessity that any new ideas had to be of high quality and fit for intended purpose and use throughout the CI process. Initially, the Red Hat project manager started the process of coming up with innovative ideas how to improve current processes. He wanted to make sure that the Leapp Team's primary focus was on creating new codes rather than get side-tracked by any distractions that could delay their main work. As a result of this approach, it was possible to create new ideas far more constructively, making the best use of available resources. Two ideas (Table 9) from the list of created options were selected. They were considered most suitable to fix the majority of the identified problems. The first idea was to migrate the maximum number of CI processes to a hosted continuous integration service called Travis CI. This service is used to build and test software projects that are hosted with the help of something known as GitHub (development platform used by software developers to review and build codes). It appeared to the team that this approach could potentially resolve the following associated issues:

- test results would be visible to everyone involved in the pro
- reduced need for any repairs compared to the current CI process
- tests run automatically and can run at individual or large group levels
- reduced likelihood of malicious codes being introduced into the process

The second idea was concerned with improving the original thinking of staying with an open-source automation server (known as Jenkins). This approach, sometimes referred to as the non-human part of the software development process, enables software developers to reliably build, test and deploy their software. This solution has the advantage of allowing for an endless customization of the associated server. The disadvantages of adopting this approach are twofold. Red Hat associates only would have sight of the test results and it carries a need for substantial regular maintenance related to its infrastructure. The measurement of the idea generating process established that the project manager spent 30 minutes on developing Idea1 and 70 minutes developing Idea 2. In contrast, the Leapp Team spent 130 minutes to assess whether Idea 1 was feasible for application within the CI process. They spent 180 minutes on assessing whether Idea 2 was feasible. This provided the iScorecard Team with significant statistical evidence on which to base suggestions how this process could be improved, working collaboratively with all parties involved in the CI process.

Table 10: Systematic idea generation and evaluation
(Source: own processing)

Baseline (hrs)	**PM**	**Leapp team**	**Red Hat team**
Idea 1 Generation	0.17	2.00	0
Idea 1 Evaluation	0.33	0,17	0
Idea 2 Generation	0.17	2.00	0
Idea 2 Evaluation	1.00	2.17	0
Summary Idea1	**0.50**	**2,17**	**0**
Summary Idea2	**1.17**	**3.00**	**0**

Outputs

G1O01 – Quality of generated idea(s)

This metric (Table 11) is intended to illustrate the expectation, expressed in percentage terms, of how many problems in the CI process could be solved in relation to the number of generated problems. The adoption of this metric carries another benefit. It is possible to measure resource commitment levels in terms of time and inputs needed to resolve problems. This metric's primary focus is on measuring all functionalities irrespective of their difficulty level such as easy or difficult. This ensures that easy functionalities are not completed first at the expense of difficult functionalities. It is also possible that measurement results would be distorted. This would "send the wrong message to senior management". The iScorecard team was asked to apply some form of "weighting" to each functionality to manage this challenge as objectively and effectively as possible. Contrasting views between the main parties concerned developed during the early process stage. The project manager expected that Idea 1 should solve 71 % of the identified problems. The Leapp Team's view was different. They considered that only 64% of the identified problems associated with Idea 1 would be resolved successfully. As far as Idea 2 was concerned, the project manager did not expect any problem resolution. His reasons for holding this view were manifold;

- open visibility of unit testing-everyone could see everything at any time resulting in too many interferences,
- the automation of build creation may not suit this particular process,
- a reduction of available engineering time due to spending too much time on fixing errors/bugs.

The project manager held the view that the Jenkins server could not provide any improvements. The Leapp Team did not share this view. They considered that adopting a Jenkins approach would yield at least 50% improvements to solve problems. This team also rated the second Idea much higher that the project manager. It was based on having access to far more data than the project manager has. This provided the Leapp Team with supportive evidence on which to base their views. This team was more focused on finding innovative solutions that would ultimately improve their modus operandi.

Table 11: **Expected quality of the generated ideas**
(Source: own processing)

Baseline		**Idea 1: Migration from Jenkins to Travis CI**		**Idea 2: Jenkins tune-up**	
Problem	**Weight**	**PM** (%)	**Leapp Team** (%)	**PM** (%)	**Leapp Team** (%)
Unit tests	0.5	100	100	0	50
Build creation	0.1	75	50	0	50
E2E tests	0.1	0	0	0	50
Complexity	0.3	45	30	0	50
Summary		**71**	**64**	**0**	**50**

Results

G1R01 – Milestone/Deadline

Pace and effectiveness are two of the many components that determine whether a project is going well. It is suggested that there is a need for a metric that measures whether the development phase of an idea is being implemented in time and how successful that implementation has been. This idea started on 3rd June 2019 and its implementation was completed on 7th June 2019. The outcome of the applied metric met expected results.

G1R02 – Total cost of the idea generating phase

This metric measures the money spent during the first phase of the project. This cost tracking can be used as a benchmark for future projects and associated phases. The total cost of the idea creation phase was calculated based on the time to complete this project stage. It is measured by multiplying, for example, the hours a person has spent on work related to this project. An average hourly associate/senior manager hourly rate is used to calculate the actual cost to the business for conducting this work. It was then possible to compare the total cost for work done with the set financial limit Red Hat has agreed for this project such as 50,00 CZK. As can be seen from Table 12, the total cost incurred did not exceed the budget limit. The iScorecard team ensured, through regular reviews and calculations, that this limit was not exceeded and that it was managed accordingly. This adopted approach ensured that the true cost of innovation in Red Hat was measured for the benefit of senior management so informed business decisions, for example, can be made.

Table 12: Total cost of idea generation phase
(Source: own processing)

Baseline	FTE	Weekly costs (CZK)	Costs (CZK)
iScorecard	0.60	16,500	13,200
Red Hat	0.16	19,500	3,087
Summary			**16,287**

4.2.2 Stage 2: Idea Development

The concept of Idea Development plays an important role within the Continuous Integration project. One of its major functions is to explore and establish how the generated ideas from the previous project stage can be suitably integrated into the whole CI process. Another of its functions is to verify and confirm that adopted ideas are fit for their intended purpose and that accepted and implemented ideas actually improve Red Hat's operational performance. It is possible to measure how successful and effective adopted ideas have been through the application of appropriate measurement techniques such as metrics and statistical analysis. It is thus possible to provide evidence of performance improvement or deterioration on a work area by work area basis. Similar to the approach considered in Stage 1, the effectiveness and efficiency of investigated current processes will be measured objectively through the application of face to face interviews with Red Hat staff who are engaged in the associated work activities. The iScorecard Team expects that data of different depth levels will be generated by this research. This will enable the team to develop and roll-out appropriate performance measures to show evidence of improvements such as "before" and "after" changes to working practices were introduced. Continuity is an important factor within the CI process. This means that any considered and ultimately selected ideas can only proceed to Stage 3 when all selection criteria have been met. In order to succeed with this approach, the iScorecard Team agreed with Red Hat senior management that the development of ideas would be managed through "sprints" that typically last for two weeks. So-called integration testing was carried out after every two sprint events. This type of testing is part of a testing sequence that includes unit, integration and user acceptance testing to ensure that integrated components work together as expected and intended. Table 13 shows a summary of developed and adopted performance metrics and the target values the iScorecard Team set for each of these. Further metrics were considered necessary in order to measure how successful each innovation has actually been, a kind of measuring the measurement.

Table 13: Metrics for Stage 2
(Source: own processing)

	Metric number/name	Target
Inputs	G2IO1 – Work Effort for Given Tasks	iScorecard max 2 FTE Red Hat max 1,6 FTE
	G2I02 – Number of proposals from stage 1	Min. 1
Process	G2P01 – Interventions within the Development Stage by the innovation team	Intended max. 1 Unintended max. 5
Outputs	G2O01 – Quality of proof of concept offered	Min. 75%
Results	G2R01 – Milestone/Deadline	30th July 2019
	G2R02 – Total cost of idea Development Stage	Maximum 100.000 CZK

Inputs

G2IO1 – Work Effort for Given Tasks

This metric is very similar to the metric used for Stage 1. Because this is a different process than in the previous Stage, it is appropriate to measure and determine how effectively and efficiently human resources have been deployed. The essence of tracking this metric is to focus on project team members and see how much effort they've made to meet the goals set during Gate 2. This metric will measure, with the time individual team members spent on project related activities, expressed in Full Time Equivalent (FTE) terms. The target value of this metric was to reach 1.60 FTE for Red Hat and 2.00 FTE for iScorecard. Table 14 includes “extension common”. This was agreed at the meeting with Red Hat staff. The reason for this inclusion that it appeared that the time spent on this extension common was greater than the work associated with both ideas separately. Extension common is an expression used in software development. It describes an ability interchange hardware or software in a given environment without any other code or configuration changes being required and resulting in zero negative impacts. This extension, used for Travis and Jenkins applications, was “tried and tested” to confirm if the adoption of these applications would generate more stability. Members of the project (Leapp team) spent in total 18 hours working on the idea 1 and 5 hours for idea 2. Most of the work was done in the first week of the sprint, the second week included only a "pull request", which is important for assessing whether the ideas involved are compatible for the process.

Table 14: Work Effort for Given Tasks
(Source: own processing)

Baseline (FTE)	**Inception**	**Termination**
PM	0.01	0.30
Leapp Team	1.18	1.00
Red Hat Team	0.00	0.30
Summary	**1.19**	**1.60**
Team Leader	0.70	0.80
Professor	0.55	0.40
Student	0.60	0.80
iScorecard Team	**1.85**	**2.00**

G2I02 – Number of proposals from Stage 1

This metric measures the number of ideas generated in Stage 1 that have actually been considered and moved forward to Stage 2 (Idea Development). It will only be applied once during Stage 1. The execution is typically conducted prior to the start of Stage 2. The considered ultimate target is to have at least one idea approved and selected. Any selected idea must be justified. The following two ideas passed the selection process and were considered fit for use throughout the CI process in order to:

1. Migrate the maximum number of CI processes to a hosted CI service called Travis CI
2. Improve the original thinking of staying with an open-source automation server known as Jenkins

Each of these ideas incorporates expected benefits that are congruent with the idea generating process criteria. As a result, once the idea generating process was completed, a decision was made by the iScorecard Team to proceed to Stage 2 as all requirements to move to the next stage were fulfilled.

Process

G2P01 – Interventions within the Development Stage by the innovation team

This metric is intended to record any intentional or unintentional intervention that may arise either from a key stakeholder group in the Red Hat team or from the iScorecard Team. It is also important to mention that any intervention should have a positive effect on the CI process and not harm it in any way. The maximum target values for these metrics were set at five intentional and one unintended intervention. No thoughtful or unintended interventions were recorded during the idea development phase that would significantly change the CI process. As a result, it is reasonable to assume that, in principle, the desired target has been achieved.

Outputs

G2O01 – Quality of proof of concept offered

This metric's primary focus is on the completion of activities that lead to the confirmation and demonstration that this pilot study project's design concept in the area of innovation scorecard will actually work when implemented. This proof of concept will show that the Innovation Scorecard Team's approach to deliver the "Continuous Integration Project", from a technological point of view, is feasible. This particular metric's starting point was based on two ideas that were developed in the previous Stage 2 (Idea Development). These ideas were considered fit for intended purpose and it was now necessary to confirm that this consideration proved to be true. Proof of concept typically involves the application of two sequential actions. The initial action is concerned with establishing the potential research feasibility. One such review covers the area of application mapping. This is a process used in areas such as IT and software development. It deals with and establishes, for example, what the components and interdependencies are within a certain software development area and then "maps" these. It provides a kind of "helicopter" view of any particular total process. This will aid informed decision making. Provided this first step is completed successfully and produces a positive evaluation in terms of feasibility, the research project will be allowed to continue to the second action. This involves moving any considered and confirmed suitable ideas to the implementation phase (Stage 3).

To ensure that the proof of concept continues to stay true to its nature, it is necessary to review the status of the concept after agreed changes have been integrated. Essential integration testing will be conducted to achieve this. This ensures continuity of quality control and assurance. It is generally considered more effective to conduct measurements after changes have been implemented. This is based on the knowledge and experience that not all sprints may generate changes. This metric is based on the use of a so-called flexible job board solution that provides the Red Hat team with full visibility of all planned activities. This approach provides opportunities to maximise work output with minimum effort. This particular metric (Table 15) is intended to show how many problems are solved during the proof of concept phase. It provides supportive evidence by taking into account how many ideas were generated in relation to the number of identified associated issues.

An important benefit of this metric is its ability to focus on all features regardless of the difficulty level (easy or difficult). This provides assurance that easy functions are not completed in preference to more difficult functions. Red Hat applied some "weighting" to certain functions in accordance with their levels of priority. This enabled the iScorecard Team to manage this challenge as unbiased and as objectively as possible. Based on the outcome of the feasibility study for considered improvement ideas, the project manager decided that Idea 1 could solve 59% of the identified problems. Red Hat's Leapp Team shared the Project Manager's view leading to an agreed implementation of the adopted Idea 1. There was a difference of opinion as far as Idea 2 was concerned. The Project Manger maintained a strong view that the implementation of Idea 2 would not lead to a resolution of identified problems associated with Idea 2. The Leapp Team considered that the "Jenkins" enhancement could potentially solve 88% of the identified issues. This issue has been resolved. The Project Manager decided to proceed with his view and that therefore the implementation of Idea 2 did not go ahead. The issue was closed successfully.

Table 15: Offered proof of concept Quality
(Source: own processing)

Baseline		**Idea 1: Migration from Jenkins to Travis CI**		**Idea 2: Jenkins tune-up**	
Problem	**Weighting**	**PM** (%)	**Leapp Team** (%)	**PM** (%)	**Leapp Team** (%)
Unit tests	0.5	100	100	0	100
Build creation	0.1	0	0	0	75
End2End tests	0.1	0	0	0	75
Complexity	0.3	30	30	0	75
Summary		**59**	**59**	**0**	**88**

The iScorecard Team experienced a similar problem during Stage 1 (Idea Generation) as far as the evaluation of the quality of the generated ideas is concerned. It is therefore possible to compare the two metrics (Table 16) and extrapolate relevant information that helps to determine which idea, for example, meets the feasibility criteria better in relation to the proof of concept.

In contrast, Migration from Jenkins to Travis CI (Table 16) has a 12% lower success rate than the project manager expected. The Leapp Team agreed with the Project Manager. In addition, there was a negative deviation between the expected quality of the idea and the actual quality of the idea (5%). The Project Manager maintained his view that Idea 2 cannot improve any of the identified issues. The Leapp Team considered that the adoption of the "Jenkins tune-up" could improve the resolution of issues by 38% more than they expected. Once again, both parties did not agree on a potential resolution due to holding different views and perspectives on what might work. It is important, in this context, that evaluations are conducted to generate factual inputs into the decision-making process. It is vital that informed decisions are made rather than making decisions based on assumptions and feelings. It should also be noted that both parties hold different perspectives, for example, based on customer expectations (Project Manager) and the views of software or system developers (Leapp Team). The target values for both Idea 1 and Idea 2 were initially set at 75% and according to the Leapp Team only Idea 2 actually met this condition. Although each idea contained sufficient benefits to be considered before making a decision whether to adopt the idea, it was ultimately up to the Project Manager to make the final decision.

Table 16: Difference between Quality of Generated Ideas and Proof of concept
(Source: own processing)

Baseline	Quality of Generated Ideas (%)	Proof of concept (%)	Variance (%)
Idea 1	71	59	- 12
	64	59	- 5
Idea 2	0	0	0
	50	88	+ 38

G2R01 – Milestone/Deadline

The purpose of this metric is to measure whether the idea development stage was performed in a timely manner. An initial milestone was agreed and set for the completion of the Idea Development Stage on 30th July 2019. The actual completion of this work was finished on 24th July 2019. The outcome of the applied metric met expected results.

G2R02 – Total cost of the Idea Development Stage

This metric provides evidence of the actual cost associated with the implementation of Stage 2 (Idea Development). The total cost for the idea generation phase is calculated based on actual time spent during this phase, measured by multiplying hours/days spent by Full Time Equivalents (FTE) and the average hourly rate used within Red Hat at associate/senior leadership level. It was then possible to cost the total amount of work conducted against the financial limit (CZK 200,000) set by Red Hat for this Stage. The total budgeted cost for this Stage did not exceed the limit, on the contrary, the total amount spent was well below the budgeted cost (Table 17). The iScorecard Team had the responsibility of ensuring that the budget was not exceeded. This was achieved by conducting regular financial performance reviews.

Table 17: **Total actual cost for the Idea Development Stage**
(Source: own processing)

Baseline	**FTE**	**Weekly costs (CZK)**	**Costs (CZK)**
iScorecard	1.85	16,500	30,525
Red Hat	1.19	19,500	23,156
Summary			**53,681**

4.2.3 Stage 3: Implementation

The implementation phase demonstrates if generated ideas actually work as expected within the CI process. In contrast to Stage 2 (Idea Development) when a lot of exploration and testing is conducted, this Stage is concerned with putting the adopted theory into practice. An important and essential activity often associated with this Stage is the need to maintain control and communicate well with all parties involved. Appropriate metrics fit for their intended purpose for use in this Stage were developed and subsequently selected. This enabled the iScorecard Team to record and monitor relevant data from Red Hat to verify if the adopted idea was compliant or non-compliant with the current CI process used in Red Hat. One important aspect of this approach was to check, for example, if earlier rejected ideas could perhaps still be considered and implemented, based on collecting and analyzing new data that provided new insights. Each generated project output was checked for quality and measured to see if it met all the criteria for proceeding to Stage 4. Similar to the other Stages, the outcomes from Stage 3 (Implementation) will be measured (Table 18) at the end of two sprints that each last for two weeks (four weeks in total).

Table 18: Metrics for Gate 3
(Source: own processing)

	Metric number/name	Target
Inputs	G3IO1 – Work Effort for Given Tasks	iScorecard max 1.6 FTE Red Hat max 0.8 FTE
	G3I02 – Senior Management commitment	Min. 1 from CI team Min.1 from Leapp or Red Hat team
Process	G3P01 – Number of meetings/calls within the Innovation Project	Min. 1 at inception + 1 within stage
Outputs	G3O01 – Number of change requests relating to proof of concept	Radical max. 1 Enhancement max. 2
Results	G3R01 – Milestone/Deadline	28th October 2019
	G3R02 – Total cost of idea generation phase	Maximum 300,000 CZK

G3IO1 – Work Effort for Given Tasks

This metric measures how productive engaged human resources have been to support the successful roll-out of the concept of Innovation Scorecard within the CI process at Stages 1 and 2. This metric, as part of Gate 1 and 2 activities, measures the time individual team members spent on project related activities, expressed in Full Time Equivalent (FTE) terms (Table 19). The target value for this metric was to reach 0.35 for Red Hat and 0.8 for the iScorecard Team. It should be noted that the FTE used in this metric was slightly distorted by associated and related time spent on a component known as "Linter". This component was related to work conducted in association with CI project extension work for both the Travis and Jenkins platforms. This was an important change in terms of actual efforts required to complete all planned work in a timely manner. This was difficult to achieve for a number of reasons.

The Leapp Team, during the first two sprints, spent one hour (0.0125 FTE) checking platform issues relating to Travis. This slight diversion lasted one hour for each sprint. The first sprint for the other platform known as Jenkins meant that the Leapp team needed to perform so-called “base of infrastructure stabilization” activities. This involved the correction of regular errors. The second sprint required even more time (24 hours or 0.3 FTE). This included changing end to end testing that did not work as needed and rearranging testing that was appropriate and fit for the intended purpose. An unexpected situation arose during the planned third sprint in the form of an external intervention. This intervention included a radical change that allowed the Red Hat to migrate the entire CI process to another cluster (the system on which Jenkins operates). There was no way to prevent intervention, the only compromise they could achieve was to shift the deadline from weekly to monthly. The fourth sprint involved the migration of an existing cluster (backing up the resource cluster, migrating the backup data and then restoring the data to a target cluster) to a new cluster. This took the Leapp team 24 hours (0.30 FTE).

The intervention was completed during the fifth sprint. This took the Leapp team another 48 hours (0.6 FTE) and an additional 8 hours by the Red Hat team (0.10 FTE). Once the fourth and fifth sprints were completed, there were still many errors in the infrastructure that caused system blackouts. After each blackout, the help of an outsourcer was required to detect errors, correct them and provide information on how to resolve the problem in the future. However, even this solution did not prevent the blackouts from happening again after a few days. The summary (Table 19) shows that the required FTE values have not been reached. This stage took more time than expected as a direct result of unpredictable intervention. It was Red Hat’s intention to maintain the performance quality of the Jenkins platform before the intervention happened to avoid further financial spending.

Table 19: Work Effort for Given Task
(Source: own processing)

Baseline	**1. Sprint**	**2. Sprint**	**3. Sprint**	**4. Sprint**	**5. Sprint**
Leapp team	0,06	0,31	External Intervention	0,30	0,70
Red Hat team	0,00	0,00		0,00	0,10
PM	0,00	0,00		0,002	0,05
In Total	0,06	0,31	0,00	0,302	0,35
Red Hat Summary		1,022			

Baseline	1. Sprint	2. Sprint	3. Sprint	4. Sprint	5. Sprint
Team Leader	0.30	0.25	0.05	0.05	0.20
Professor	0.10	0.20	0.00	0.00	0.10
Student	0.20	0.22	0.05	0.05	0.20
In Total	0.60	0.67	0.10	0.10	0.50
iScorecard Summary		1,970			

G3I02 – Senior Management commitment

This metric measured how many senior management staff (who supported the innovation project during previous stages and made final "Go" decisions), were still actively participating in the Implementation Stage. This metric partially fulfils the set goal to ensure that support for the project was provided and maintained within Red Hat. The target measurement value for this metric has been set up to measure if at least one senior management Red Hat team member is still supporting the project after they made a "Go" decision during the Initial Stage of the project. This metric is executed after each 2 sprints (monthly). Following an analysis, it is confirmed that at least one of the two project managers involved in this process, are still continuing to support the project. The first project manager, who managed the entire CI process from the beginning, continues to support the implementation phase, in which both ideas are tested. The second project manager appears not to support the ongoing project. On the contrary, his intervention caused many problems in the third, fourth and fifth sprints (already discussed before).

G3P01 – Number of meetings/calls within the Innovation Project

This metric measures the number of meetings and/or telephone/conference calls held/completed during the CI Innovation Project (Table 20) to ensure that relevant project information and progress reports are produced for sharing between the Red Hat Project Manager, the CI Team and the iScorecard Team. The aim of this metric is to check that clear task responsibilities and commitments to complete these are in place within the project and that support for and awareness of the Innovation Project is maintained within Red Hat. It was agreed with Red Hat that a minimum of one meeting should be held before the Implementation Stage begins and one meeting during the Implementation stage.

It appears that people who work in IT and Software Development prefer to communicate by electronic means such as a so-called ticketing system. The reason for this is that they do not wish to be interrupted. They need to focus their attention, without interruption, on the tasks ahead. They view interruptions to be some kind of interventions that they see as "errors in the program", perhaps regarding these as "we have done something wrong". Communication levels during the first two ideas' work activities (including Travis and Jenkins) were relatively low as the work was conducted in accordance with the project plan. This changed as soon as interventions occurred. This generated a need for increased communications between members of the Leapp Team. Most communications took the form of face to face meetings and electronic means. The Jenkins platform had to be migrated to a new "cluster" as mentioned before, without causing any major concerns or issues. This explains the need to communicate more to avoid these problems.

Table 20: Number of meetings/calls within the Innovation Project
(Source: own processing)

Baseline	**Travis**		**Jenkins**		**Intervention**	
	Inception	Stage	Inception	Stage	Inception	Stage
Meeting	2	2	0	0	0	5
Telephone calls	0	0	0	0	0	0
Conference calls	0	0	0	2	0	0
E-comm	0	0	2	2	3	10
Summary	**2**	**2**	**2**	**4**	**3**	**15**

G3O01 – Number of change requests relating to Proof of Concept

It is essential to manage any change requests in a controlled manner to maintain project control. Therefore, this metric has been developed to measure how successful change requests were managed and controlled by the Leapp Team and make recommendations how to keep change requests to an absolute minimum. This is an important approach. Change requests, in any project, can occur at any time during the life cycle of the project.

To make sure proper control over these change requests is exercised, it is essential to control this process.

The metric target values were set are based on the assumption that no more than five change requests would be received during this stage. It was further assumed that this would include one request for a major or substantial change and five requests for minor changes. Three change requests only were recorded during this whole stage. The first two requests for change were received during the first and second sprints. These were essential as they related to improving the functionality of the Jenkins platform. The first change requested a "stabilization of the underlying infrastructure" and the second change request requested changes to the end to end testing process as the initial tests did not produce the desired test results. It was therefore necessary to revisit the end to end testing by making necessary changes to the test programme to meet the Leapp Team's requirements. Some change requirements were relatively easy to roll out and integrate into the existing end to end test process. One change request was quite complex as it demanded some radical changes to the test process, including the overall migration of an existing cluster to another cluster. All of these changes are "declassified" as described in metric G3IO1.

G3R01 – Milestone/Deadline

This metric measures the timely performance of the implementation stage. An initial milestone for the completion of the Implementation Stage was set to 30th October 2019. It was difficult to complete the Implementation Stage within the set time parameters and target.

The radical intervention that occurred during this stage produced some negative impact on the CI process that appeared to make the process less efficient and effective after the innovation was integrated. This explains why the iScorecard Team could not achieve the set and expected completion time target.

G3R02 – Total cost of the Idea Generation Phase

Similar to Gate 1 and 2, this metric measures the cost of the implementation phase. The total cost for the idea generation phase is calculated based on actual time spent during this phase, measured by multiplying hours/days spent by Full Time Equivalents (FTE) and the average hourly rate used within Red Hat at associate/senior leadership level (Table 21). It was then possible to cost the total amount of work conducted against the financial limit set by Red Hat for this Stage. The highest cost associated with the Implementation Stage relates to the fifth sprint. This was due to the fact that an existing cluster was transferred and then implemented in another cluster to guarantee the desired high-quality level of integrity of the Jenkins platform.

Table 21: **Total cost of Idea Generation Phase**
(Source: own processing)

Baseline (CZK)	FTE	1. Sprint	2. Sprint	3. Sprint	4. Sprint	5. Sprint	Summary
Red Hat	1.48	2,438	12,187	-	11,782	31,200	57,606
iScorecard	1.97	19,800	21,945	3,300	3,300	16,500	64,845
In Total							122,451

4.2.4 Evaluation of the Implementation Stage for Ideas 1 and 2

Ideas 1 and 2 appeared to be the optimum solutions for fixing the problems identified with the help of the Leapp Team and the Project Manager. When some external intervention occurred that caused some significant damage to the overall system infrastructure, it was no longer possible to continue with the smooth operations of the considered Idea 2 as far as the Jenkins platform was concerned. This intervention also impacted Idea 1 (Travis CI). Travis failed to provide key features to address classified problems (described in G1P01). For this reason, ideas 1 and 2 have been evaluated as inadequate, implying that they cannot advance to Stage 4 and are evaluated as being a “No Go” decision (Figure 11). In the next section, a new way of solving the CI process issues will be considered and introduced to fix this issue. As it is no longer possible to proceed to Stage 4, it will be a necessary requirement for measurements must to start from Stage 1 (Idea Generation).

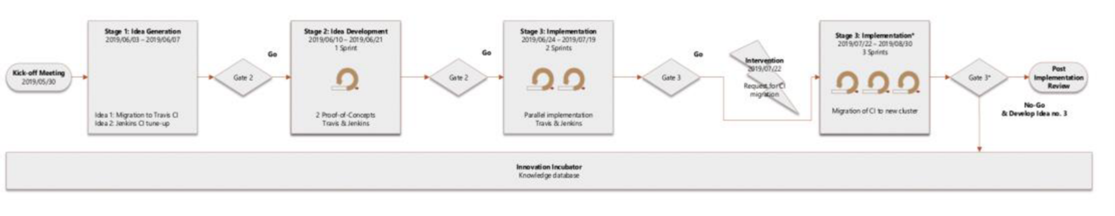


Figure 11: “No Go” decision for Ideas 1 and 2
(Source: own processing)

4.3.5 Stage 1: Idea generation for Idea 3

This stage reports the evaluation of generated ideas that Red Hat considers having the potential to address some of their classified problems. The metrics agreed by the Red Hat team (Table 22) will be used and tracked by the iScorecard team, to exclude any ideas that do not have the potential to advance to the next stage of the project (Stage 2: Idea Development).

Table 22: Metrics for Gate 1
(Source: own processing)

	Metric number/name	Target
Inputs	G1IO1 – Work Effort for Given Tasks	iScorecard max 0.8 FTE Red Hat max 0.25 FTE
	G1I02 – Quality of current CI process	Minimize blocks
Process	G1P01 – Time of systematic idea generation and evaluation	Max. 1 week
Outputs	G1O01 – Quality of generated idea(s) – percentage of the problems solved by each idea generated	Min. 75%
Results	G1R01 – Milestone/Deadline	9th September 2020
	G1R02 – Total cost of idea generation phase	Maximum 50,000 CZK

Inputs

G1IO1 –Work Effort for Given Tasks

The idea generation phase required substantial inputs in terms of time from the project manager (PM) who was often considered to be the author of ideas that have been created to solve problems associated with the CI process. In addition, inputs from a new manager, bringing in some different perspectives as far as innovation was concerned, helped the current PM to analyse and evaluate the proposed new idea and to check its suitability for use within the Red Hat team. With the help of the inputs from both managers, exchanging different views and ideas, it was possible for the PM to decide that the newly-generated idea could potentially solve the identified problems in the associated area of how much time people spent on the completion of tasks (Table 23). Entering the task in the so-called

JIRA system (shows the cycle or lead time for any product, version or sprint, see Chapter 2.1.6) helped to support and facilitate Red Hat's project management process and any specified requirements that formed part of this process.

Table 23: Work Effort for Given Tasks
(Source: own processing)

Baseline (FTE)	**Inception**	**Termination**
PM	0.08	0.05
Leapp Team	0.03	0.20
Red Hat Team	0.00	0.00
Red Hat Summary	**0.11**	**0.25**
Team Leader	0.20	0.30
Professor	0.20	0.20
Student	0.20	0.30
iScorecard Team Summary	**0.60**	**0.80**

Process

G1I02 – Quality of current CI Process

This particular performance metric (Table 24) is most suitable to demonstrate the effectiveness of the current CI process in terms of quality, showing the "before" and "after" of any adoption of any new idea how to improve it. The table shows a summary of how much time both the Leapp and Red Hat teams spent fixing errors between August and September 2019. Errors were of two kinds. Less severe errors forced the Leapp team to slow down operations whilst the more serious errors stopped all work until errors were resolved or eliminated. There was an external intervention during August which resulted in the adoption of the so-called Jenkins server in order to improve the CI process. As can be seen from Table 23, the Leapp team spent a total of 72 hours and the Red Hat team spent a total of 8 hours to fix the errors generated during this time period. Despite best efforts, an additional 15 new errors emerged after the migration. The PM instigated the creation of a new idea how errors could be fixed more effectively and efficiently in future.

Table 24: Lists of Errors from August to September 2019
(Source: own processing)

Baseline	Frequencies	Leapp team (hrs)	Red Hat team (hrs)	Blockage (hrs)
August		72	8	80
September	15	8	4	12
Summary				**92**

G1P01 – Time of systematic idea generation and evaluation

The metric from Table 25 was also used to assess how long it took in general terms to create any new development idea suitable for the CI process. Although the intention was to create new ideas internally, it sometimes was necessary to respond to so-called "external interventions" even if this meant that the Leapp team had to spend more time on fixing errors and problems. In this case, the PM was able to develop a new idea that could mitigate the impact of the external migration and solve other CI process problems. This approach had the advantage that it ensured that the Leapp team were able to focus their attention more on improving Red Hat products and services. This was a challenge as the current CI process still required a lot of maintenance work to be completed. This was managed through the application of the so-called Jenkins tool. With the previous manager having left this process and the new manager just having started, there was a unique opportunity to generate a new idea, namely: "To transfer the infrastructure work responsibility to another team". According to Table 25, the PM spent 10 minutes developing idea 3 and 30 minutes assessing whether idea 3 was feasible for application in the current CI process. In contrast, the Leapp team spent 30 minutes to ponder the feasibility of applying the idea in the CI process.

Table 25: Systematic idea generation and evaluation
(Source: own processing)

Baseline (hrs)	PM	Leapp team	Red Hat team
Idea 3 Generation	0.17	0.00	0
Idea 3 Evaluation	0.50	0.50	0
Summary Idea 3	**0.67**	**0.50**	**0**

Outputs

G1O01 – Quality of Generated Idea(s)

This metric was primarily concerned with the quality of the generated ideas. Four potential areas within the current CI process were identified as likely "candidates" to resolve the issues around Idea 3. These are:

- Unit Testing-the visibility of unit testing: everyone could see all test results at any time, resulting in too many interferences
- Build Creation-reduced likelihood of malicious codes being introduced into the process
- E2E Testing-tests run automatically and can run at individual or large group levels
- Complexity- a reduction in available engineering time due to spending too much time fixing errors/bugs

Table 26 shows that Idea 3's assessment by stakeholders (PM and Leapp team) is almost identical which confirms that both parties feel strongly about this. Using this example, it can be seen from the results that the PM expected Idea 3 to solve 43% of the identified problems. In contrast, the Leapp team considered that Idea 3 could solve 59% of the identified problems. The two parties displayed different priorities which explains the difference in the % problem resolution. The PM was focused on output, irrespective how this was achieved. In contrast, the Leapp team considered that they may not fix all of the problems but that they will have higher levels of commitment (25% more) to get things done. It is for this reason that the Leapp team viewed Idea 3 more positively. Their main concern was that they might be able to fix the complexity problem but that this could lead to a reduction in their available time to fix bugs/errors. This team took into account that Idea 3 could solve 75% of these problems. This was also aligned with the target values set for this metric. Irrespective of the expected outcome, the PM insisted that Idea 3 should move forward to the next stage in the cycle: Development.

Table 26: Expected Quality of the Generated Idea (Stakeholder Assessment)
(Source: own processing)

Baseline		**Idea 3: Migration of Jenkins to Another Team**	
Problem	**Weight**	**PM** (%)	**Leapp Team** (%)
Unit Testing	0.5	0	50
Build Creation	0.1	100	50
End2End Testing	0.1	100	50
Complexity	0.3	75	80
Summary		**43**	**59**

Results

G1R01 – Milestone/Deadline

This idea started on 2nd September 2019 and its implementation was completed on 9th September 2019. The outcome of the applied metric is that the development phase of an idea needs to be implemented within set timescales.

G1R02 – Total Cost of the Idea Generating Phase

By using this metric, the funds used during the first stage of the project were tracked. The financial costs for the first stage were agreed to be no more than CZK 50,000. As can be seen from Table 27, the total "spent cost" did not exceed the budget limit.

Table 27: Total cost of idea generation phase
(Source: own processing)

Baseline	**FTE**	**Weekly costs (CZK)**	**Costs (CZK)**
iScorecard	0.60	16,500	9,900
Red Hat	0.16	19,500	1,950
In Total			**11,850**

4.2.6 Stage 2: Idea Development for Idea 3

The next phase of the project is known as Idea Development. It provides an examination and identification of how the generated ideas from the previous phase of the project can be integrated appropriately into the entire CI process.

Table 28 shows a brief summary overview of the developed and received performance metrics (Inputs, Process, Outputs and Results) and target values that the iScorecard team has set for each of them. Included are also metrics to measure how successful upgrades have been.

Table 28: Metrics for Stage 2
(Source: own processing)

	Metric number/name	**Target**
Inputs	G2IO1 – Work Effort for Given Tasks	iScorecard max 2.00 FTE Red Hat max 1.6 FTE
	G2I02 – Number of proposals from Stage 1	Min. 1
Process	G2P01 – Interventions within the Development Stage by the Innovation Team	Intended max. 1 Unintended max. 5
Outputs	G2O01 – Quality of Proof of Concept offered	Min. 75%
Results	G2R01 – Milestone/Deadline	17th November 2019
	G2R02 – Total cost of idea Development Stage	Maximum 100,000 CZK

G2IO1 – Work Effort for Given Tasks

This particular metric focuses on measuring the efforts required in terms of time to perform given tasks. In this example, the Full Time Equivalent (FTE) target for Red Hat staff was 1.6 FTE and the figure for the iScorecard team was 2.0 FTE. A new Quality Engineering (BaseOS QE) team joined the project at this stage. Their main role was to provide technical quality-related advice and services relating to the migration of the existing infrastructure to another team. As can be seen from Table 29, there was no Red Hat involvement during the first sprint. It appears that there was a lack of potential work capacity in the first two weeks of this process to make any progress during the sprint. This changed during the second sprint. Progress could be made and the Leapp team was engaged for twelve hours doing work in relation to Idea 3. A similar amount of time was spent by the Leapp team in relation to the last sprint of this process. The majority of this work was in relation to so-called “pull requests” (a method of submitting contributions, for example, in an open software development project when a developer typically asks for an external repository to be considered for inclusion within the main system’s repository). It was thus possible to assess if an idea was actually compatible with the existing work process. It is also interesting to note that the PM did not make any contributions during the Idea Development phase. The likely reason for this could be that the PM did not want to get involved in or interfere with technical issues but focus his attention on ensuring that his expectations were met.

Table 29: Work Effort for Given Tasks
(Source: own processing)

Baseline (FTE)	**1. Sprint**	**2. Sprint**	**3. Sprint**
PM	0.00	0.00	0.00
Leapp Team	0.00	0.30	0.30
QE Team	0,10		
Red Hat + QE Team Summary			**0,70**
Team Leader	0.05	0.30	0.30
Professor	0.00	0.25	0.20
Student	0.05	0.30	0.25
Summary (iScorecard Team)			**1,70**

G2I02 – Number of Proposals from Stage 1

The adoption of this metric enabled the evaluation of captured ideas from Stage 1 (Idea Generation) and assess how many of these ideas were actually moved forward to Stage 2 (Idea Development). The main goal for this metric was to have at least one idea from Stage 1 approved and selected. The outcome from this exercise confirmed that only one idea passed the suitability selection process and that this idea was considered suitable for use throughout the CI process to achieve the following objective:

1. To migrate the responsibility for the existing infrastructure of Jenkins to another Red Hat team

This idea carried many benefits so the PM and the Leapp team decided to move this idea forward to the next stage of this research project.

Process

G2P01 – Interventions within the Development Stage by the Innovation Team

The iScorecard team set the target value for this metric as five intended and one unintended intervention. Interventions in this context mean to help and motivate development teams to improve the streamlining of operations and driving productivity forward. In addition, it was a necessary requirement to consider the significant impacts interventions can have on the CI process, both at negative and positive levels. During the Idea Development Stage, no intended or unintended interventions were recorded that would significantly change the CI process. As a result, the desired objective mentioned above has not been achieved.

Outputs

G2O01 – Quality of Proof of Concept Offered

The target value for this metric was set to solving a minimum of 75% of the earlier identified issues by applying Idea 3. Based on the results (Table 30) of quality of proof of concept offered, the project manager decided that Idea 3 can improve the project by a total of 97 % in the four examined areas. The Leapp Team believed that Idea 3 could potentially solve 93 % of the identified problems.

Table 30: **Offered Proof of Concept Quality**
(Source: own processing)

Baseline		Idea 3: Infrastructure Migration to Another Team	
Problem	**Weighting**	**PM** (%)	**Leapp Team** (%)
Unit Testing	0.5	100	100
Build Creation	0.1	100	100
End2End Testing	0.1	100	100
Complexity	0.3	90	75
Summary		**97**	**93**

The iScorecard team recorded a similar metric during Phase 1 (Idea Generation), as far as the evaluation of the quality of the generated ideas is concerned. Therefore, it is possible to compare two metrics (Table 31) and induce relevant information to help determine whether the idea under scrutiny has a positive impact on key stakeholders' expectations. The PM's opinion was that Idea 3 had a 54% higher success rate than initially expected. The Leapp team's conclusion was that adopting Idea 3 could improve the issue of problem solving by 34% more than expected. Once the proof of concept exercise was completed, both the PM and the Leapp team agreed that Idea 3 could bring the following benefits;

- an improved flexible infrastructure capable of meeting business and technological changes,
- a reduced workload,
- a reduction in the need for error correction within Jenkins,
- less input knowledge requirements by team members (Jenkins is complicated and requires expertise),
- testing can be conducted from many locations without the need for a hosted service.

The outputs from this metric confirm that Idea 3 can be considered as a means to resolve current and future problems associated with this particular Red Hat working environment. The quality assessment of the idea exceeds the previously-set target value of 75% and as a result the idea can automatically advance to the third phase of the project (Implementation).

Table 31: Difference between Quality of Generated Ideas and Proof of Concept
(Source: own processing)

Baseline		Quality of Generated Ideas (%)	Proof of concept (%)	Variance (%)
Idea 3	PM	43	97	+ 54
	Leapp Team	59	93	+ 34

G2R01 – Milestone/Deadline

The initial milestone for the completion of Idea 3 was agreed and set for completion on 17th November 2019. The actual completion of this phase was achieved on 21st January 2020. This delay was caused by workforce unavailability due to operational overriding business needs.

G2R02 – Total Cost of the Idea Development Stage

A total amount of CZK 100,000 was set to complete the Idea Development phase in a timely manner. As can be seen from Table 32, the total expenditure for this phase was achieved at a below budget level, resulting in a cost saving of 18,550 CZK.

Table 32: Total Actual Cost for the Idea Development Stage
(Source: own processing)

Baseline	FTE	Costs (CZK/Week)	1. sprint	2. sprint	3. sprint	Summary (CZK)
Red Hat	0.60	16,500	-	11,700	11,700	25.350
iScorecard Team	1.19	19,500	3,300	28,050	24,750	52,800
In Total						**81,450**

4.2.7 Stage 3: Implementation for Idea 3

The implementation phase determines which generated ideas actually work as planned and expected within the CI process. This stage involves a move from theoretical research to practical application of the research outcomes. Communications with key stakeholders is an important activity during the implementation phase. This is the reason why most of the selected metrics shown in Table 33 were designed to measure how successful communications have been within the associated CI process.

Table 33: Metrics for Gate 3
(Source: own processing)

	Metric number/name	Target
Inputs	G3IO1 – Work Effort for Given Tasks	iScorecard max 1.6 FTE Red Hat max 0.8 FTE
	G3I02 – Senior Management commitment	Min. 1 from CI team Min.1 from Leapp or Red Hat team
Process	G3P01 – Number of meetings/calls within the Innovation Project	Min. 1 at inception + 1 within stage
Outputs	G3O01 – Number of change requests relating to proof of concept	Radical max. 1 Enhancement max. 2
Results	G3R01 – Milestone/Deadline	30th June 2020
	G3R02 – Total cost of idea generation phase	Maximum 300,000 CZK

G3IO1 – Work Effort for Given Tasks

The target value for this metric was 0.8 for Red Hat and 1.6 for the iScorecard team. It should be noted that the full-time (FTE) used in this metric was distorted due to unplanned time being spent on “debugging” as part of the CI process. While this component was not related to the migration of the “Jenkins infrastructure” to another team (Idea 3), it was necessary to complete this work in order to maintain the integrity of the CI process. Table 34 shows that the implementation of Idea 3 took distinctly more Red Hat time than the determined target value. This process has proved significantly more complicated than was initially expected. The Leapp team spent 4.75 FTE migrating Jenkins to another team. It was important for the Leapp team that the Jenkins migration retained all relevant functionalities and that none of these were lost during the migration. Reality was different. The migration did not go as planned. One of the key functionalities could not be migrated to another team. The absence of this functionality forced the Leapp team to spent a considerable amount of their time to fix this problem which had a negative impact on the time available to do their planned work. Table 34 also shows that during the seven sprints there was no involvement of the Red Hat team or the PM.

Table 34: Work Effort for Given Task
(Source: own processing)

Baseline (FTE)	1. Sprint	2. Sprint	3. Sprint	4. Sprint	5. Sprint	6. Sprint	7. Sprint
Leapp team	0.30	0.60	0.80	0.75	0.80	0.70	0.80
PM	0.00						
Red Hat Team	0.00						
Red Hat Summary	4,75						
Team Leader	0.10	0.30	0.20	0.25	0.20	0.20	0.20
Professor	0.05	0.10	0.05	0.20	0.10	0.05	0.10
Student	0.01	0.20	0.15	0.22	0.20	0.15	0.20
iScorecard Summary	3,12						

G3I02 – Senior Management commitment

The target value for this metric was set to include that at least one member of the Red Hat senior management team was still supporting the CI project. Following a team meeting held on 24th April 2020, the Red Hat approved that the CI project manager would continue to actively support the implementation phase which included the testing of Idea 3. This metric does not cover the time another manager spent on this project during the implementation phase. It is possible that the Red Hat team did not consider this necessary as this manger's time was already accounted for in his normal "day job".

G3P01 – Number of meetings/calls within the Innovation Project

It was agreed with Red Hat that the target value for this metric would be one meeting before the implementation phase and one meeting during the implementation phase. The level of communication during the implementation of Idea 3 was relatively high because the work carried out did not comply with the project plan. As can be seen from the outcomes of the previous stages, people working in IT and software development prefer to communicate electronically. As a result, most of the communications between the project team members were in electronic format. During the implementation phase there were a total of 10 meetings and the Leapp team spent a total of 42 hours attending these meetings. The time of the Quality Engineering (BaseOS QE) team which helped the Leapp team with technical quality-related advice and services related to the migration of the existing infrastructure to another infrastructure, was also included in the "total time spent" during the implementation phase. This team spent most of their time attending review meetings that typically lasted about 2 hours. Table 35 shows that a total of 2 pre-implementation meetings and 34 meetings/E-comm during implementation were held which confirms that the set target for meetings was achieved (see above).

Table 35: Number of meetings/calls within the Innovation Project
(Source: own processing)

Baseline	**Idea 3**	
	Inception	Stage
Meeting	2	10
Telephone calls	0	0
Conference calls	0	0
E-comm	0	24
Summary	**2**	**34**

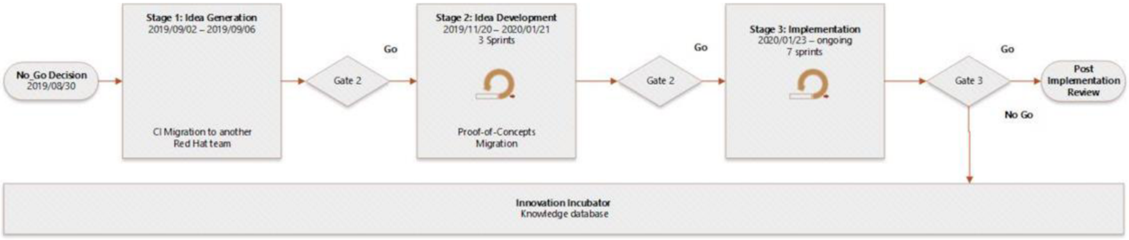
G3O01 – Number of change requests relating to Proof of Concept

The tracked metric targets have been set to assure that no more than five change requests will be received during this phase. However, no change requests were recorded during the entire implementation phase. For now, Idea 3 represents only an initial concept that is not yet complete. In the course of this concept, it is not possible to identify specific change requests that could potentially improve the process. Any ideas on how to improve the process can only be specified after the implementation is complete. Therefore, all expected customer requests will be feasible only after the successful completion of the implementation.

G3R01 – Milestone/Deadline

An initial milestone for the completion of the Implementation Stage was set to the end of June 2020. For the purpose of this thesis, related data was captured only until the set milestone of 30th April 2020. It should be noted that this limitation does not allow for the completion of this research, which is still in progress (Figure 12).

Figure 12: Ongoing Idea 3
(Source: own processing)



G3R02 – Total cost of the Idea Generation Phase

The total set amount for the implementation phase was set at CZK 300,000. Table 36 shows that as of 30th April 2020, it can be considered that the planned target rate has been reached. However, the real completion of the implementation phase is expected as of 30th May 2020. With the remaining CZK 11,955, there is a low probability that the set limit will be reached.

Table 36: **Total cost of Idea Generation Phase**
(Source: own processing)

Baseline (CZK)	FTE	1. Sprint	2. Sprint	3. Sprint	4. Sprint	5. Sprint	6. sprint	7. sprint
iScorecard	3.15	8,250	19,800	6,600	21,945	16,500	13,200	16,500
Red Hat	4.75	11,700	23,400	31,200	29,250	31,200	27,300	31,200
Summary							**288,045**	

Conclusion

The innovation process represents a wide range of activities carried out from the initial idea itself, through development to its expected application into practice. The success of this complex process at all stages is conditional on the ability of the entity that implements the innovation process. However, each innovation is unique, specific and intended to gain competitive advantage and business growth. This brings with its revenue growth but at the same time it raises costs. In today's business there appears to be a lack of measurement of how successful innovations have been. This could prevent wasting unnecessary resources. The following is a summary of the main conclusions that can be drawn from the outcomes of this research.

The initial assumption for the successful writing of this thesis (Chapter 1) was the determination of methodological starting points. In the field of methodology, most of the methods or procedures that were selected in the beginning as suitable for the focus of this work have been used.

The second chapter contained the theoretical background and presents research of Czech and foreign professional literature and other relevant information sources. This chapter was divided into three parts. The initial section deals with an agile environment that is very important for the IT industry. Software development is a very dynamic environment where it is necessary to accept and address change with great speed and flexibility in order to compete in a competitive environment. In response to this dynamic environment, agile methods have been developed that support the close cooperation of programmers with target users to prevent unwanted results. One of the most commonly used methods in an agile environment is Scrum. With the help of Scrum, managers are able to organize team work, focusing on monitoring and solving any obstacles that might arise in the way of successful software development. The second part contains a limited overview of the literary research in order to determine what is already known about the four identified areas. These four sub constructs of agile project management were selected as the most suitable for this thesis: teamwork, communication, work processes and Empowerment. For example, this research found that the level of relative and effective communication in an agile environment is much higher than in a non-agile work environment. For this reason, it is important that companies accept the right performance measurement metrics to verify how effective this communication is in reality. The last part contains the definition of innovation and is an introduction to the new practical innovation scorecard approach which provides conceptual measurement of the effectiveness of innovations. It also acts as a management framework for innovation in different sectors.

The third chapter focuses on the characteristics of Red Hat s.r.o., which is considered to be the world leader in the provision of open source and Linux operating systems. This chapter dealt primarily with the understanding of internal working processes and how Red Hat works in general. It was necessary to examine the theory of what was already known about the subject under investigation. This way, the Innovation Scorecard system can be modified and adapted to be suitable for a thoughtful purpose and used in the Agile Software Development work environment. The reason Red Hat decided to work with the iScorecard team was the high frequency of problems arising in the process called "Continuous Integration".

The last design chapter focuses on the practical application of knowledge from theoretical backgrounds where the Innovation Scorecard system was introduced in detail. After reviewing internal processes, the iScorecard team was able to implement a proposal to introduce the Innovation Scorecard system into Red Hat's real work environment without disrupting work activities. The use of the Stage Gate model for ongoing project management has proven to be a very effective and productive tool to measure how successful the deployment of innovation within their business has been. Four stages (Idea Generation, Idea Development, Implementation and Post-implementation) were created for this model. Each stage has been tracked with five metrics corresponding to the project stage. With the help of this model, it was possible to measure the generated ideas 1,2,3. The measurement showed in the implementation phase that the generated ideas 1 and 2 are not the optimal solution for the identified problems, therefore, according to the decision of the project manager, they were evaluated as a "No Go" decision. Since ideas 1 and 2 could no longer proceed to the fourth stage, it was necessary to generate a new idea which had to be measured from the first Stage.

Idea 3 that followed has already shown better outputs and positive results based on completed measurements. At the implementation stage, this process turned out to be much more complicated than Red Hat expected, thus extending the implementation time.

The resulting evaluation of whether idea 3 could solve the identified problems could not completed because the project could not be finished in accordance with the original project plan. In spite of all this, it can be concluded that the Innovation Scorecard helps businesses achieve better results and thus increase their overall performance. This tool also highlights that it is a big mistake for managers to focus solely on scoreboards and try to influence them directly, instead of influencing the quality of the functioning of the innovation process. The benefit of this thesis for Red Hat is to serve as a guide for preparing metrics and starting to measure the success of their innovations/changes. This will significantly improve the company’s readiness for the changes that lie ahead and respond and cope better with competition and market condition changes.

References

ALSAQAF, W., DANEVA, M. and WIERINGA, R., 2019. *Quality requirements challenges in the context of large-scale distributed agile: An empirical study. Information and Software Technology*, 110, pp.39-55., [On-line]. [2020-05-23]. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0950584918300739

ASHMORE, S., RUNYAN, K., 2014. *Introduction to Agile Methods. Addison-Wesley* Professional. 336 p. ISBN 978-0-133-43521-4.

ASSOCIATION FOR PROJECT MANAGEMENT, 2019. *APM Body of Knowledge*, Seventh Edition, British Library Cataloguing, ISBN 978-1-903494-82-0.

BECK, KENT, 2001. *Manifesto for Agile Software Development*. [On-line]. [2020-03-07]. Available at: http://agilemanifesto.org

COOPER, J.R., 1998. *A multidimensional approach to the adoption of innovation*: Management Decision, ISSN 0025-1747

CZECH STATISTICAL OFFICE, 2008. *Příklady inovací pro TI 2008* [On-line]. [2020-03-01]. Available at: http://www.czso.cz/csu/redakce.nsf/i/prakticky_seznam_prikladu_inovaci_verze_2008/$File/priklady_inovaci_2008.pdf

DAVILA, T., et al., 2013. *Making innovation work*: How to manage it, measure it, and profit from it. Updated ed. Upper Saddle River: FT Press, 348 p. ISBN 978-0-13-309258-5.

DRUCKER, P.F., 2006. *Innovation and Entrepreneurship*, Harper Business; Reprint edition.

ENLYFT, 2020. *Companies using Red Hat Enterprise Linux Serve*r. Enlyft. [On-line]. [2020-04-11]. Available at: https://enlyft.com/tech/products/red-hat-enterprise-linux-server

FREIRE, A. et al., 2018. *A Bayesian networks-based approach to assess and improve the teamwork quality of agile teams*. Information and Software Technology, 100, pp.119-132. [On-line]. [2020-04-23]. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0950584917300204

HAUGHEY, D., 2011. *BOSCARD*: Terms of Reference. Project Smart. [On-line]. [2020-04-02]. Available at: http://www.projectsmart.co.uk/boscard.html

HELLER, M., 2020. *What is Jenkins?* The CI server explained. [On-line]. [2020-05-11]. Available at: https://www.infoworld.com/article/3239666/what-is-jenkins-the-ci-server-explained.html

HIDALGO, E.S., 2019. *Adapting the scrum framework for agile project management in science: case study of a distributed research initiative.* Heliyon, 5(3). [On-line]. [2020-04-02]. Available at: https://linkinghub.elsevier.com/retrieve/pii/S2405844018340635

HODA, R., MURUGESAN, L.K., 2016. Multi-level agile project management challenges: A self-organizing team perspective. Journal of Systems and Software, 117, pp.245-257. [On-line]. [2020-04-23]. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0164121216000807

HORVÁTH and PARTNERS, 2002. *Balanced Scorecard v praxi.* Prague: Profess Consulting. 386 p. ISBN 80-7259-018-9.

JOHN, A., 2020, *What is Scrum? How can it help your Business?* Amazing Viral News. [On-line]. [2020-04-23]. Available at: https://www.amazingviralnews.com/what-is-scrum-how-can-it-help-your-business/

KAPLAN, R.S., NORTON, P.D., 1996. *Using the balanced scorecard as a strategic management system*, Boston: Harvard Business Review.

KERZNER, H., 2017. *Project Management*: Case Studies. Hoboken, N.J: John Wiley. ISBN 978-1-119-38597-4.

KRISHNAMURTHY, V., 2012. *A Brief History of Scrum*. TechWell Insights. [On-line]. [2020-03-07]. Available at: https://www.techwell.com/techwell-insights/2012/10/brief-history-scrum

LEI, H. et al., 2017. *A statistical analysis of the effects of Scrum and Kanban on software development projects*. Robotics and Computer-Integrated Manufacturing, 43, pp.59-67. [On-line]. [2020-04-23]. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0736584515301599

LINDSJØRN, Y. et al., 2016. *Teamwork quality and project success in software development: A survey of agile development teams*. Journal of Systems and Software, 122, pp.274-286. [On-line]. [2020-04-23]. Available at: https://linkinghub.elsevier.com/retrieve/pii/S016412121630187X

LOIRO, C. et al., 2019. *Agile Project Management: A Communicational Workflow Proposal*. Procedia Computer Science, 164, pp.485-490. [On-line]. [2020-04-23]. Available at: https://linkinghub.elsevier.com/retrieve/pii/S1877050919322574

MASKOVSKY, E., 2014, *GitHub*: Continuous Integration s Travis CI. [On-line]. [2020-05-09]. Available at: https://devsectorcs.wordpress.com/2014/11/09/github-continuous-integration-s-travis-ci/

MYSLÍN, J. *Scrum: průvodce agilním vývojem softwaru*. Brno: Computer Press, 2016. 167 s. ISBN 978-80-251-4650-7

NIVEN, P.R., 2014. *Balanced Scorecard Evolution*: A Dynamic Approach to Strategy Execution. Hoboken: Wiley. 368 p. ISBN 978-1-118-72631-0.

NOVOSELTSEVA, E., 2017. *The benefits you get by doing Agile Project Management*. [On-line]. [2020-03-07]. Available at: https://apiumhub.com/tech-blog-barcelona/benefits-of-agile-project-management/

OECD/Eurostat (2018), *Oslo Manual 2018: Guidelines for Collecting, Reporting and Using Data on Innovation, 4th Edition*, The Measurement of Scientific, Technological and Innovation Activities, OECD Publishing, Paris/Eurostat, Luxembourg. [On-line]. [2020-03-08]. Available at: https://doi.org/10.1787/9789264304604-en

PARMENTER, D., 2015. Key *Performance Indicators: Developing, Implementing, and using winning KPIs.* John Wiley & Sons, Inc., Hoboken, New Jersey. ISBN 978-1-118-92510-2.

RED HAT, 2019. *Book of Red Hat*. [online]. [2020-03-07]. Available at: https://www.rdhat.com/en/book-of-red-hat

RED HAT, 2019. *Linux platforms*: Red Hat Enterprise Linux. Red Hat. [online]. [2020-04-07]. Available at: https://www.redhat.com/en/technologies/linux-platforms/enterprise-linux

SERRADOR, P., PINTO, J.K., 2015. *Does Agile work? — A quantitative analysis of agile project success*. International Journal of Project Management, 33(5), pp.1040-1051. [online]. [2020-04-22]. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0263786315000071

STRAY, V., et al., 2016. *The daily stand-up meeting: A grounded theory study*. Journal of Systems and Software, 114, pp.101-124. [online]. [2020-04-22]. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0164121216000066

SCHUMPETER, J.A., 2004. *Kapitalismus, socialismus a demokracie*. Jiří Ogrocký. Brno: Centrum pro studium demokracie a kultury, 470 s. ISBN 8073250446.

SUTHERLAND, J., 2014. *Scrum: The art of doing twice the work in half the time*, Crown. ISBN-13: 978-1847941107

ŠOCHOVÁ, Z., KUNCE, E., 2014. *Agilní metody řízení projektů*. Brno: Computer Press. ISBN 978-80-251-4194-6.

TESSEM, B., 2014. *Individual empowerment of agile and non-agile software developers in small teams*. Information and Software Technology, 56(8), pp.873-889. [online]. [2020-04-23]. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0950584914000470

VALENTA, F., 2001. *Inovace v manažerské praxi*, Praha: Velryba. ISBN 80-85860-11-2

VERHEYEN, G., 2017. *Scrum, What's in a Name?* Agile Zone. [online]. [2020-03-07]. Available at: https://dzone.com/articles/scrum-whats-in-a-name

YAGÜE, A., et al., 2016. *An exploratory study in communication in Agile Global Software Development.*, 48, pp.184-197. [online]. [2020-05-23]. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0920548916300381

YOUNAS, M. et al., 2018. *Agile development in the cloud computing environment: A systematic review*. Information and Software Technology, 103, pp.142-158. [online]. [2020-04-23].Available at: https://linkinghub.elsevier.com/retrieve/pii/S0950584918301319

ŽIŽLAVSKÝ, O., FISHER, E., 2019. *Innovation Scorecard: Theoretical Background and Principles. International Project Management Association*. [online]. [2020-03-07]. Available at: http://www.spm-hq.jp/

ŽIŽLAVSKÝ, O., 2016. *Innovation Scorecard:* Conceptual Performance Measurement and Management Framework for Innovation Process. Brno: Vutium Press. 293 p. ISBN: 978-80-214-5370-8.

LIST OF TABLES

LIST OF FIGURES

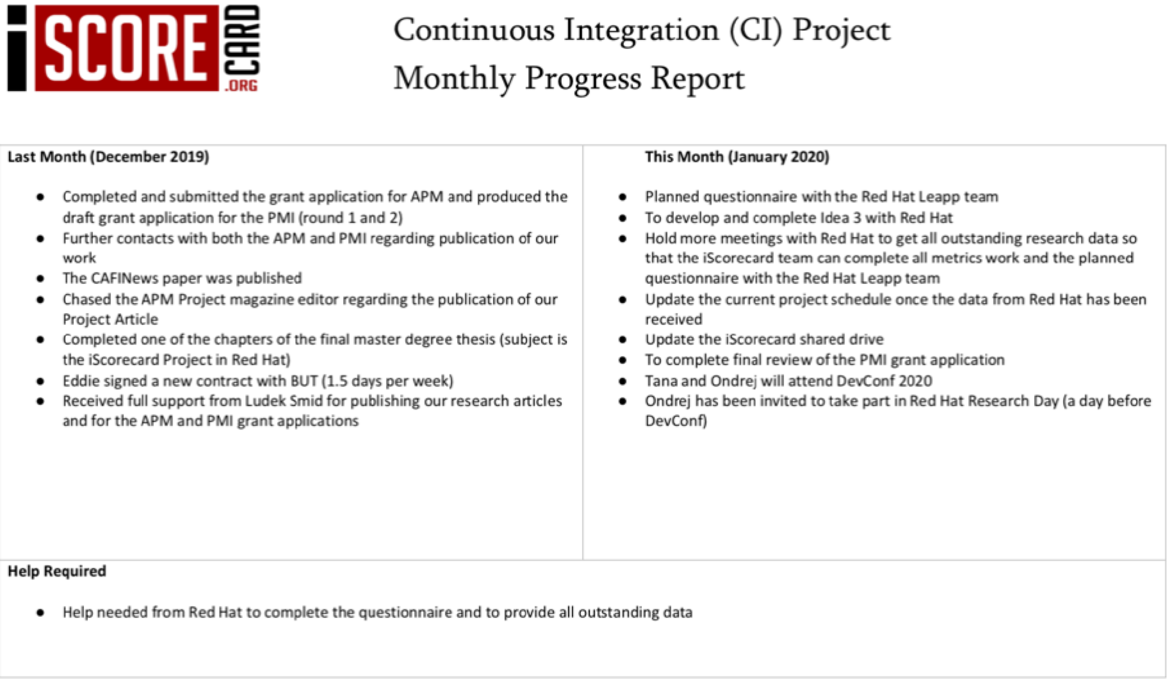
LIST OF APPENDCES

LIST OF ABBREVIATIONS

CI	Continuous Integration
CSFs	Critical Success Factors
KPIs	Key Performance Indicators
HRN	Human Resource Management
IT	Information Technology
OECD	Organisation for Economic Co-operation and Development
CZSO	Czech Statistical Office
HKS	Hells Kitchen Systems
RHEL	Red Hat Enterprise Linux
IBM	International Business Machines
RAM	Random–access Memory
VM	Virtual Machines
OS	Operation System
CD	Continuous Delivery
E2E	End to End tests
e.g.	for example
et. al.	and others
PDD	Project Definition Document
RAG	Red, Amber and Green
PM	Project Manager
FTE	Full Time Equivalent
CZK	Czech koruna

Appendices

Appendix 1: Progress report



Continuous Integration (CI) Project
Monthly Progress Report

Last Month (December 2019)

- Completed and submitted the grant application for APM and produced the draft grant application for the PMI (round 1 and 2)
- Further contacts with both the APM and PMI regarding publication of our work
- The CAFINews paper was published
- Chased the APM Project magazine editor regarding the publication of our Project Article
- Completed one of the chapters of the final master degree thesis (subject is the iScorecard Project in Red Hat)
- Eddie signed a new contract with BUT (1.5 days per week)
- Received full support from Ludek Smid for publishing our research articles and for the APM and PMI grant applications

This Month (January 2020)

- Planned questionnaire with the Red Hat Leapp team
- To develop and complete Idea 3 with Red Hat
- Hold more meetings with Red Hat to get all outstanding research data so that the iScorecard team can complete all metrics work and the planned questionnaire with the Red Hat Leapp team
- Update the current project schedule once the data from Red Hat has been received
- Update the iScorecard shared drive
- To complete final review of the PMI grant application
- Tana and Ondrej will attend DevConf 2020
- Ondrej has been invited to take part in Red Hat Research Day (a day before DevConf)

Help Required

- Help needed from Red Hat to complete the questionnaire and to provide all outstanding data

Appendix 2: Action list

Action list						
No.	RAG	Date	Detail	Owner	Planed completion	Actual completion
1.	🟩	28.6.2019	Produce monthly dashboard report	Leader	2.7.2019	2.7.2019
2.	🟩	28.6.2019	Produce high level project documents - PDD, Issues and Risks Register, Change Control Process, Project Action list	Leader/ Student	2.7.2019	2.7.2019
3.	🟩	28.6.2019	Produce Schedule for the project	Professor	6.12.2019	19.12.2019
4.	🟩	28.6.2019	Update the PDD and review it by Red Hat	Leader	5.7.2019	5.7.2019
5.	🟩	30.7.2019	Discuss and develop metric for CI project	Leader	31.8.2019	31.8.2019
6.	🟩	2.9.2019	Get all research data from Red Hat	Leader/ Student	15.10.2019	18.10.2019
7.	🟩	2.9.2019	Write to CEO of Red Hat to get more support	Professor	2.9.2019	2.9.2019

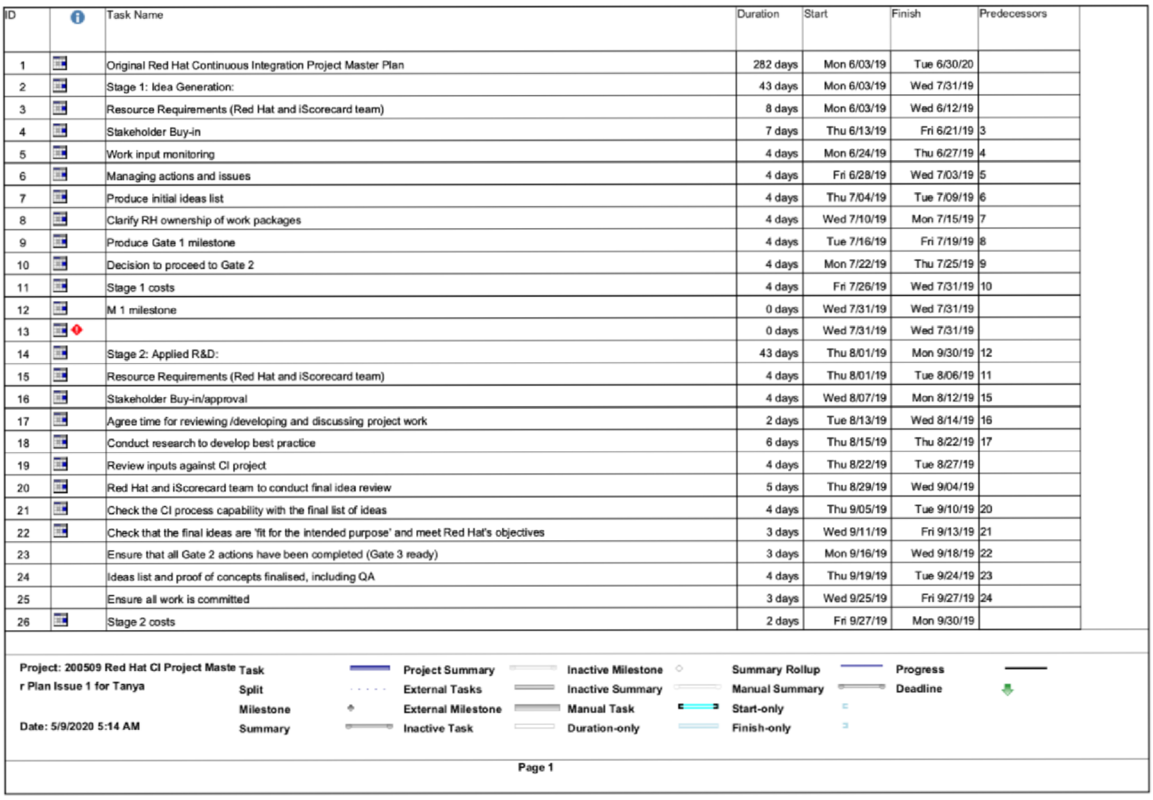
Appendix 3: Risks register

Risks Register					
No.	RAG	Date	Detail	Value	Measure
1.	🟧	5.8.2019	Red Hat may decide to stop the CI project and restart	The iScorecard team needs to spend more time to restart the project	Will be reviewed monthly
2.	🟩	2.9.2019	The key member of Red Hat may leave the company	Project costs may increase	This no longer a risk by 30.11.2019
3.	🟥	2.9.2019	The iScorecard team cannot complete all measurements	More time needed and costs may go up	Risk closed. This is now an issue
4.	🟧	19.12.2019	The impact of IBM on the availability and commitment of Red Hat stuff	The iScorecard team needs to make more available to get information from RH	Will be reviewed monthly

Appendix 4: Issues Register

Issues						
No.	RAG	Date	Detail	Owner	Planed completion	Actual completion
1.	🟩	5.7.2019	Access into Red Hat internal network	Leader/ Student	7.8.2019	7.8.2019
2.	🟧	13.8.2019	Lack of research data from Red Hat	Leader/ Student	18.10.2019	
3.	🟩	2.9.2019	Problems with shared drive documents	Leader	30.10.2019	30.10.2019
4.	🟩	2.9.2019	Current time spent on project by iScorecard team member	Leader	30.11.2019	15.12.2019
5.	🟧	19.12.2019	Lack of commitment by Senior Management in Red Hat	Leader/ Professor	31.1.2020	

Appendix 5: Original Project Plan



ID	ⓘ	Task Name	Duration	Start	Finish	Predecessors
1	▦	Original Red Hat Continuous Integration Project Master Plan	282 days	Mon 6/03/19	Tue 6/30/20	
2	▦	Stage 1: Idea Generation:	43 days	Mon 6/03/19	Wed 7/31/19	
3	▦	Resource Requirements (Red Hat and iScorecard team)	8 days	Mon 6/03/19	Wed 6/12/19	
4	▦	Stakeholder Buy-in	7 days	Thu 6/13/19	Fri 6/21/19	3
5	▦	Work input monitoring	4 days	Mon 6/24/19	Thu 6/27/19	4
6	▦	Managing actions and issues	4 days	Fri 6/28/19	Wed 7/03/19	5
7	▦	Produce initial ideas list	4 days	Thu 7/04/19	Tue 7/09/19	6
8	▦	Clarify RH ownership of work packages	4 days	Wed 7/10/19	Mon 7/15/19	7
9	▦	Produce Gate 1 milestone	4 days	Tue 7/16/19	Fri 7/19/19	8
10	▦	Decision to proceed to Gate 2	4 days	Mon 7/22/19	Thu 7/25/19	9
11	▦	Stage 1 costs	4 days	Fri 7/26/19	Wed 7/31/19	10
12	▦	M 1 milestone	0 days	Wed 7/31/19	Wed 7/31/19	
13	▦ ◆		0 days	Wed 7/31/19	Wed 7/31/19	
14	▦	Stage 2: Applied R&D:	43 days	Thu 8/01/19	Mon 9/30/19	12
15	▦	Resource Requirements (Red Hat and iScorecard team)	4 days	Thu 8/01/19	Tue 8/06/19	11
16	▦	Stakeholder Buy-in/approval	4 days	Wed 8/07/19	Mon 8/12/19	15
17	▦	Agree time for reviewing /developing and discussing project work	2 days	Tue 8/13/19	Wed 8/14/19	16
18	▦	Conduct research to develop best practice	6 days	Thu 8/15/19	Thu 8/22/19	17
19	▦	Review inputs against CI project	4 days	Thu 8/22/19	Tue 8/27/19	
20	▦	Red Hat and iScorecard team to conduct final idea review	5 days	Thu 8/29/19	Wed 9/04/19	
21	▦	Check the CI process capability with the final list of ideas	4 days	Thu 9/05/19	Tue 9/10/19	20
22	▦	Check that the final ideas are 'fit for the intended purpose' and meet Red Hat's objectives	3 days	Wed 9/11/19	Fri 9/13/19	21
23		Ensure that all Gate 2 actions have been completed (Gate 3 ready)	3 days	Mon 9/16/19	Wed 9/18/19	22
24		Ideas list and proof of concepts finalised, including QA	4 days	Thu 9/19/19	Tue 9/24/19	23
25		Ensure all work is committed	3 days	Wed 9/25/19	Fri 9/27/19	24
26	▦	Stage 2 costs	2 days	Fri 9/27/19	Mon 9/30/19	

Project: 200509 Red Hat CI Project Master Plan Issue 1 for Tanya

Date: 5/9/2020 5:14 AM

Task	▬	Project Summary	▭	Inactive Milestone	◇	Summary Rollup	▬	Progress	▬
Split	·····	External Tasks	▬	Inactive Summary	▭	Manual Summary	▬	Deadline	⬇
Milestone	◆	External Milestone	▬	Manual Task	▬	Start-only	⊏		
Summary	▬	Inactive Task	▭	Duration-only	▭	Finish-only	⊐		

Page 1