BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF BUSINESS AND MANAGEMENT

FAKULTA PODNIKATELSKÁ

INSTITUTE OF ECONOMICS

ÚSTAV EKONOMIKY

PROCESS INNOVATION EFFICIENCY EVALUATION IN IT ORGANISATION

PROCESS INNOVATION EFFICIENCY EVALUATION IN IT ORGANISATION

MASTER'S THESIS

DIPLOMOVÁ PRÁCE

AUTHOR Bc. Pavla Vaverková

AUTOR PRÁCE

SUPERVISOR doc. Ing. Ondřej Žižlavský, Ph.D.

VEDOUCÍ PRÁCE

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA PODNIKATELSKÁ

Zadání diplomové práce

Ústav:	Ústav ekonomiky
Studentka:	**Bc. Pavla Vaverková**
Studijní program:	Ekonomika a management
Studijní obor:	Podnikové finance a obchod
Vedoucí práce:	**doc. Ing. Ondřej Žižlavský, Ph.D.**
Akademický rok:	2018/19

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:

Process Innovation Efficiency Evaluation in IT Organisation

Charakteristika problematiky úkolu:

Introduction
Goals of the Thesis and Applied Methods
The Theoretical Background of Thesis
Analysis of Current Situation
Proposed Solution: Innovation Scorecard Design
Conclusion
References
Appendixes

Cíle, kterých má být dosaženo:

The main goal of the master thesis is to design, implement and verify Innovation Scorecard framework for process innovation in Red Hat's RHEL Atomic Host.

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 978-0-13-309-58-5.

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 2018/19

V Brně dne 28.2.2019

L. S.

doc. Ing. Tomáš Meluzín, Ph.D.
ředitel

doc. Ing. et Ing. Stanislav Škapa, Ph.D.
děkan

Abstrakt

Diplomová práce je zaměřena na měření efektivnosti procesních inovací ve společnosti Red Hat Czech s.r.o. Práce je rozdělena do tří částí. První část práce se zabývá teoretickými poznatky. Ve druhé části se jedná o představení společnosti a pochopení interních procesů a systémů firmy. Poslední část obsahuje návrhy na zlepšení tohoto projektu, které vychází z analýzy současné situace společnosti.

Klíčová slova

inovace, procesní inovace, dotazníkové šetření, Innovation Scorecard, KPI, inovační metriky, projektové řízení, podpora rozhodování, efektivnost procesu, úspora zdrojů, cílové hodnoty

Abstract

This master thesis focuses on measuring the effectiveness of process innovation in Red Hat Czech s.r.o. The thesis is divided into three parts. The first part deals with theoretical knowledge. The second part is concerned with the introduction of the company and providing an understanding of the internal processes and systems currently in operation within the company. The last part contains suggestions for improvement by this project. This is based on the outcome of an analysis of the current modus operandi within the software development area of the company.

Key Words

Innovation, Process Innovation, Questionnaire Survey, Innovation Scorecard, KPIs, Innovation Metrics, Project Management, Decision-making support, Process Efficiency, Saving Resources, Target Values

Rozšířený abstrakt

Diplomová práce je zaměřena na měření efektivnosti procesních inovací ve společnosti Red Hat Czech s.r.o. Tato společnost využívá ve svých procesech Agile přístup, který představuje vývoj softwaru na základě iterativního vývoje, kde se požadavky i následné řešení vyvíjí podle spolupráce mezi jednotlivými týmy. Tato práce se zabývá konkrétním procesem ve společnosti a tím je Container Rebuild Process, na kterém se podílí týmy, které jsou složeny z lidí z celého světa. Tento model, který je založen na iterativním vývoji, umožňuje rychlý vývoj software a jeho největší předností je schopnost reagovat na změny požadavků ze strany zákazníků v průběhu vývojového cyklu. Je flexibilní a v rámci Scrumu, což je jedna z agilních metodik, kterou se firma rozhodla používat, se jedná o dodávky v tzv. sprintech (nejčastěji se jedná o období 2 týdnů). Výhody tohoto modelu jsou, kromě již zmiňované rychlé reakce na změny, také rychlost dodávaného výstupu a smyslem tohoto modelu je něco vytvořit již v rámci prvního sprintu.

Pro tvorbu efektivního a správného řešení je nutné nejprve zanalyzovat konkrétní společnost a pochopit její procesy a vnitřní prostředí. Vzhledem k tomu, že se jedná o společnost, která působí celosvětově, bylo žádoucí zvolit pro sepsání této diplomové práce anglický jazyk. Dále bylo zapotřebí nejprve zanalyzovat současný stav a činnosti s ním spojené. Z této analýzy vyplývá, že ve společnosti a v konkrétním Container týmu jsou role rozdělené a náplň práce je přímo naplánovaná. Což může v realitě však někdy znamenat to, že v některých případech nezbývá čas na další úkoly, protože se některá část procesu zpozdila, a tudíž byla časově náročnější, což vyústilo ve zpoždění všech ostatních navazujících činností.

Na základě zanalyzovaných procesů a pochopení, jak společnost funguje, mohl být nadesignován systém Innovation Scorecard, který byl následně aplikován na procesní inovace v Red Hatu. Tento systém měl následující strukturu procesu: Cíle – Kritické faktory úspěchu – Klíčové ukazatele výkonnosti – Metriky – Cílové hodnoty. V první řadě bylo potřeba stanovit cíle, kterých má být dosaženo za pomoci aplikace Innovation Scorecard v praxi. Následovalo stanovení kritických faktorů úspěchu, se kterými jsou dále spojené klíčové ukazatele výkonnosti. Je důležité si uvědomit, že všechny klíčové ukazatele výkonnosti jsou metriky, ale ne všechny metriky jsou klíčové ukazatele výkonnosti. Velmi důležitá část této práce představuje stanovení metrik a jejich cílových hodnot. Správné stanovení metrik a jejich cílových hodnot je totiž alfou a omegou efektivního měření systému procesních inovací.

Podle definované struktury procesu jsou cíle hlavním aspektem pro Innovation Scorecard a také mají relevantní dopad na úspěšnost a implementaci Innovation Scorecard. Dále existují kritické faktory úspěchu, které jsou odvozeny od tohoto definovaného cíle. Charakterizují takovou kritickou situaci, která může reálně nastat. Pro tyto kritické faktory úspěchu jsou vytvořeny klíčové ukazatele výkonnosti, které informují, jak se konkrétní společnost snaží řešit kritické faktory úspěchu. Metriky slouží pro vyjádření inovačních cílů, které musí být jasné a bezchybné a zároveň musí být dosažitelné. Není důležitý počet stanovených metrik, ale jejich kvalita a celkový přínos k úspěšnému zavedení designu Innovation Scorecard. Na základě těchto informací bylo stanoveno celkem 7 metrik, které byly rozděleny podle toho, do které části procesu patří. Jednalo se o metriky input, process, output a outcomes. Jednotlivé metriky byly sledovány podle toho, co bylo požadováno stanovenou metrikou změřit, především byly sledovány trendy. Jednotlivé metriky byly také rozděleny podle toho, zda se měřily mezi jednotlivými verzemi (sprinty) nebo pouze před aplikací a následně až po aplikaci. Tento rozdíl záležel na vypovídací schopnosti jednotlivých metrik.

Na základě zjištěných výsledků v případě metriky struktury pracovních aktivit bylo zjištěno, že pomocí aplikace Innovation Scorecard na procesní metriky měla aplikace automatizace dopad na eliminaci manuální práce u jednotlivých členů týmu. V případě 5 lidí ze 7 se jednalo o totální snížení manuální práce ve prospěch jiných prioritních úkolů. U aktivit 1, 2, 4 a 7 se dokonce jednalo o snížení u všech členů na rozpětí 20 – 0 % po aplikaci automatizace.

U metriky časového zpoždění byla ve verzi 3 a 4 zjištěna hodnota 0, což byla cílová hodnota. Dá se konstatovat, že bylo dosaženo minimalizace nebo případně úplného zrušení časového zpoždění. Podle dostupných informací z verze 7.6.2 bylo ušetřeno minimálně 8 hodin práce. Tento časový fond se dá věnovat jiným aktivitám, případně novým projektům.

S případnými erory se počítalo již na začátku aplikace automatizace do praxe. V případě první verze medium erory přesahovaly stanovené cílové hodnoty, ale v další verzi již byly v požadovaném rozpětí. Nicméně v poslední verzi tyto hodnoty opět stouply, protože byl implementován nový mechanismus, kvůli kterému opět vznikly chyby v systému. Tento mechanismus však nebyl plně implementován a zablokoval Container tým na další dva nebo tři dny. Z tohoto důvodu se opět vrátili k originálnímu mechanismu, který byl implementován na začátku.

Požadavky na změny, které byly vyjmenovány, byly spojeny s funkčností a efektivností aplikace. Byly postupně zavedeny v následujících verzích a dá se říci, že vedly k úspěšnému fungování, protože proces běží automaticky, což byl hlavní cíl implementace automatizace do praxe. Počet prioritních úkolů, na které nezbýval čas, byl minimalizován až na hodnoty 0 téměř u všech členů týmu. To znamená, že již nejsou prioritní úkoly, které by členové týmu zanedbávali. Cílem aplikace bylo redukovat manuální práci jednotlivých členů. Díky tomuto byl ušetřen čas, který může být dále rozdělen na jiné aktivity nebo projekty.

Velmi důležitou metrikou v rámci hodnocení byla spokojenost práce. Ta byla hodnocena před automatizací a následně po ní, a přestože byli členové týmu velmi spokojeni se svojí prací ještě před aplikací, jejich spokojenost byla velmi vysoká i po zavedení aplikace. Výsledky byly hodnoceny na základě Likertovy 10-ti bodové škály a byla potvrzena spokojenost Container týmu s aplikací.

Na základě všech těchto informací je metrika ušetřeného času také velmi důležitá. Podle dostupných informací byl build proces prováděn manuálně a byla snaha tuto práci zautomatizovat. Tento cíl byl splněn. Na základě výpočtů bylo ušetřeno 13 hodin, které může Container tým nyní investovat do nových procesů nebo nových projektů ve firmě. Tyto zjištěné informace a výsledky vedou k závěru, že systém Innovation Scorecard, který byl designován, implementován a ověřen, byl účinný a efektivní v aplikaci automatizace v praxi. Použitelnost konceptu Innovation Scorecard byla demonstrována na procesních inovacích v IT firmě a bylo ověřeno, že tento přístup funguje v praxi.

Bibliografická citace

VAVERKOVÁ, Pavla. *Process Innovation Efficiency Evaluation in IT Organisation* [online]. Brno, 2019 [cit. 2019-05-10]. Dostupné z: https://www.vutbr.cz/studenti/zav-prace/detail/119638. Diplomová práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav ekonomiky. 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 10. května 2019 ....…………………………...

podpis studenta

Poděkování

V této části bych ráda poděkovala vedoucímu své diplomové práce panu doc. Ing. Ondřeji Žižlavskému, Ph.D. za jeho odborné vedení práce, profesionální přístup a čas. Velmi oceňuji jeho připomínky a věcné rady, které mi poskytl při psaní této práce. Dále bych také ráda poděkovala své rodině za jejich podporu při studiu a zpracování této diplomové práce. Velké díky patří také společnosti Red Hat a členům Container týmu za poskytnutí informací, dat a podkladů pro zpracování této práce a také za jejich čas.

CONTENT

INTRODUCTION

Agile management methods have become a de facto standard over recent years for application in IT/Software Development environments and organizations across the world. Agile differs from other approaches. The focus is on how people work together and how they can cooperate with each other. Typical characteristics of Agile are fast decision making, incremental working, flexible development and autonomous working practices. This method is specifically suitable for application in small software development projects such as the project presented and discussed in this thesis. According to Drucker (2009) who claimed, “if you can’t measure it, you can’t manage or improve it”, it is important to recognize if it is possible to introduce a measurement system into an innovative or Agile working environment. The success of innovation at work or in projects is dependent on the application of an appropriate efficient and user-friendly measurement systems.

The combination of concepts of the Innovation Scorecard and Agile Methodology was the reason why many authors considered it as important. The biggest advantage of Agile access is flexibility and ability to react fast to changes. This approach is dynamic and provides quick access to solutions within software development environments. The Innovation Scorecard works on some logical steps process of implementation within organizations.

Red Hat was chosen for this master thesis because of the project I worked on in this company since January 2019. The project duration is 3 years and it is a cooperation between Brno University of Technology and the TAČR which means Technology Agency of the Czech Republic. The name of the Program is “The Program in support of applied social and scientific and humanistic research, experimental development and the innovation of ÉTA”. The title of the Project is “Innovation Scorecard: Evaluation of the Effectiveness of Process Innovations in IT Industry”. The identification code of the project is TL02000007. The results of the master thesis could be used by senior management in Red Hat to improve business decision making and how effectively the company operates in future.

1 GOALS AND APPLIED METHODS OF THE THESIS

The main goal of the master thesis is to report and discuss the design, implementation and verification framework for the introduction of a process improvement system. This system describes the introduction of an Innovation Scorecard concept that acts as a process for innovation within one Red Hat project known as "Atomic Host".

The considered objectives for this master thesis are:

- to understand work processes and how Red Hat operates in general,
- to research the theory,
- to adjust the Innovation Scorecard system to fit the Agile Software development environment,
- 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 implement the proposed Innovation Scorecard system as a part of the process innovation initiative,
- to gain feedback from stakeholders who are participating in the process innovation initiative as part of the "Atomic Host" work,
- to modify the Innovation Scorecard system as required.

The master thesis consists of three parts. Part one defines what is meant by Innovation, what the characteristics of an Innovation Scorecard are and the details of any performance measurement such as KPIs, Metrics and the difference between these two. This section includes a brief description of what is meant by an Agile approach and why many companies that operate in today's competitive software development market, have adopted an Agile way of working. An associated approach known as SCRUM is presented.

Another goal of this master thesis is to analyze the current situation of Red Hat in sufficient detail to support this study. Part two of this study includes the introduction to Red Hat, the company history, project team members and their roles and responsibilities. This includes the basic descriptions of Red Hat's processes and details of the technologies currently in use. This part of the thesis helps to identify any weaknesses and any potential areas for improvement will be elaborated in the final part of this thesis.

The final part of this project contains proposals for the best solution for this project based on the information about the current modus operandi of Red Hat. Proposals will be elaborated on, interpreted and then presented to senior management at Red Hat. A Case Study will be prepared about the application of the Innovation Scorecard during individual Sprints.

The foundation of this master thesis is to understand how Red Hat operates and functions. This includes a good understanding of their current work policies, processes and procedures and to gain better insights how Red Hat generally conducts their business. It is also important to get to know the people who are likely to work on the Innovation Scorecard initiative. It is important for this study to understand all of the processes to make an assessment how to implement any form of innovation in the future. The research team collected both hard and soft data from the outcomes of interviews and important communications with various people in Red Hat.

Analysis is a method for gaining new knowledge and its subsequent interpretation. Hendl (2012) claimed analysis is about the organization of data and their description using graph, numbers and any other mathematical tool. Data is analyzed during the data collection process but the whole process is completed after observations, interviews and questionnaires have been finished.

Synthesis is using the Innovation Scorecard for creation the proposals for solving and subsequent improvement the innovation activities in the company. Analysis of content has been applied when associated Red Hat documentation was reviewed and analyzed. This will aid the development of this thesis. Completed questionnaires will be compared in relation to the completed work by the automation process as this will provide before and after date.

Observation plays an important role in this situation, too. Observing how people work will provide valuable insights how people work, which processes they follow and what the value of meetings is to find solutions to general problems.

The questionnaire will be used to measure the current situation in Red Hat before any implementation of innovations and other changes. It can help to understand how the company works generally and what the staff work on and which methods they use to this work. It will also be helpful after any changes and implementation to measure the efficiency of implemented changes in the company or especially in the team. The result from this measurement will be used for the suggestion what implementation can be used. The Questionnaire will be evaluated using a Likert Scale.

A Case Study will also be completed in relation to the Innovation Scorecard which will be applied during various sprints. This Study will contain all information about the individual Sprints and the whole process. This Study is of both practical and theoretical value to a variety of people including practitioners and academia and across industries such as IT and Sciences. Results are valuable for individuals as well as working teams.

2 THE THEORETICAL BACKGROUND OF THE THESIS

This chapter contains the basic theoretical concepts and methods that are needed to understand the subject matter under investigation. It includes a definition of what is meant by innovation. This is important as it includes the necessary details of what will be measured and why, a description of the innovation scorecard system, the detailed characteristic of adopted KPIs and metrics, what the difference is between the two and other concepts that are associated with innovative technologies such as Agile methodology and Scrum.

2.1 Definition of Innovation

Innovation is the term which is interpreted in different ways by different authors. The concept of innovation comes from the Latin “innovare” – to renew or change, and represents something new or the renewal of something human. It can be a new idea or method or the use of new ideas and methods. It forms part of human existence. But like every concept, this concept has many interpretations and therefore has a different meaning to different people (Žižlavský, 2016).

Innovation relates to a new method, custom or device and it can be a new idea, practice or object. Innovation comprises all the activities which are connected with bringing a new product or process to market. The process of Innovation is relatively time demanding transformation process which contains intensive management and significant resources. The process of Innovation consists of the five stages especially recognition, invention, development, implementation and diffusion (Fahrer, 2012).

Innovation is a targeted change which concerns products (new products or improvements to existing products), production methods, the organization of work and production (new types of solutions), and management methods used for the first time by the company. Change and novelty are very important characteristics of innovation according to other definitions (Žižlavský, 2016).

There also appears to be a difference between the definition of what is meant by innovation and invention. Gerald Zaltman found that the main difference is the innovations do not represent something new, but the invention does. Some sources report that inventions or ideas become innovations in course of their transformation into application that is used in practice (Zaltman et al., 1973).

The concept of innovation may not always necessarily be linked to a new product or a new service that comes into the market. It can occur in the approach or process itself. The word innovation evokes something new but not always. Innovation can be understood as the process of upgrading or transforming a resource into something else and therefore even better (Žižlavský, 2016).

Innovation can be understood as an improvement of the production of products and services or production processes and thus the improvement of the economic potential of the company. Today’s innovations work on the assumption of using science and technology. František Valenta was the founder of the innovation theory in the Czech Republic. Innovation is a part of each person’s activity in its approach. It represents all changes in the internal structure of the organism or production unit (Valenta, 1969).

Division of innovation is the most commonly classified according to the Oslo Manual developed by experts in measuring and evaluating innovation in OECD member countries. It is the best international guidelines for the collection and use of data on the concept of innovation and primarily it helps to demystify what is meant by innovation. It was established by experts in the field of measurement and evaluation of innovation from OECD member states. According to this approach innovation is divided into the four basic types:

- product innovation,
- process innovation,
- marketing innovation,
- organizational innovation (OECD, 2018).

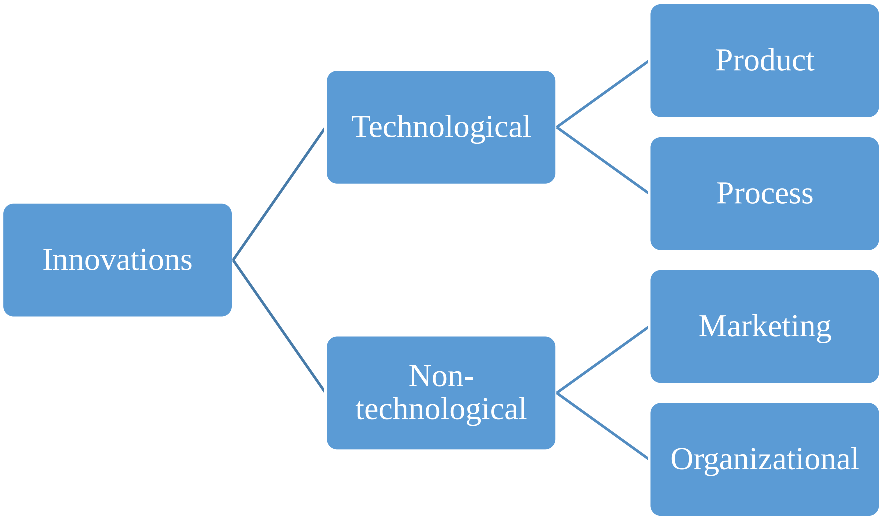

Figure 1: Types of Innovation according to the Oslo Manual

(Source: OECD, 2018)

Innovations are divided into the two types of innovations. The first is technological innovations. These technologies create either completely new products or important changes in them. These innovations can be classified mostly according to the changes in the applicability of individual procedures or changes. This category includes product and process innovations. Product innovations include products and services. They can use new knowledge or technologies. An introduction of new or improved currently production delivery methods represents process innovations (CZSO, 2018).

The second type of innovations are non-technological innovations. These technologies represent innovations that relate to organization, business structure and the social sphere. Marketing innovations and organizational innovations belongs to this category. Marketing innovations are based on marketing method which the company did not used in the past. It is not important if the method has been developed by the innovating company or adopted from another company. Organizational innovations relate to the organization of a wide range of situations such as organization of procedures, organizing relations with other companies as maintaining a certain standard in cooperation and so on (CZSO, 2018).

Innovations in the market represent a certain innovation process and without it the innovation would not happen. There are different types of innovation and it depends on how we approach them because there are also many ways to achieve these innovations.

Innovation begins with the idea of change or innovation. It follows the different phases of the development of this idea and later the realization of the innovation (Žižlavský, 2016).

2.2 To Measure or Not to Measure

It is possible to measure almost anything provided there are opportunities and possibilities, and there is a drive and determination to put measurements into practice. Measurement has a very important role to play in innovation. It focuses on the understanding of what can actually be measured, especially in the initial phase of any innovation process. Innovation, by definition, is a continuous process by itself. Change is driven by customers' forever changing needs and requirements for something new. Companies are constantly creating new changes and acquiring new knowledge and so the measurement of a process becomes more complex (Žižlavský, 2016).

The ability to drive through change by applying performance measurement criteria has become one of the most important skills for managers to have. Effective measuring systems and indicators are important for successful innovation at this provides important information for the management of innovation. Managers who have access to high quality information as a result of effective performance management systems, are more likely to make better and informed decisions. It is important for managers to recognize the results of the analysis as this may have a direct effect on the quality of their decision making. Access to quantitative information, produced as a result of some measurements, supports managers to present data analysis, for example, in statistical format. This provides evidence that the outcome of the data analysis is reliable and valid, and that this information can be used, for example, in presentations to a wider audience (Davila et al., 2013).

The quality of any knowledge suffers if work outputs cannot be measured appropriately. If people do not comprehend something in terms of true meaning, based on words and/or numbers, it is possible that things appear to be beyond people's power and they should make every effort to try and understand the meaning first before taking any further action (Žižlavský, 2016).

The choice of measurement systems plays an important role in management. When incorrect measurements are chosen that are not entirely appropriate for the situation, the results may be inefficient and totally inappropriate. As a result, the efficiency of the work and the measurement itself deteriorates and there may be non-objective analyzes and as a direct consequence, incorrect measures. As a basic rule a combination of the strategy and the measurement of innovations can be considered with several metrics to provide a clean and clear result of performance or measurement (Davila et al., 2013).

It appears that some companies measure too much, and this can be counterproductive. It is important to know when and under what circumstances it is necessary to measure. It is inappropriate when the company begins to measure too much and suddenly finds it has a lot of analyzes, calculations and results and does not really understand what to do with this data. A clear understanding of what needs to be measured and how this needs to be aligned to company strategy is of paramount importance (Žižlavský, 2016).

2.3 Innovation Scorecard

The essence of innovation is that it has to be measurable. One of the most important ingredients of the concept of innovation is the ability to change or improve the current status quo. In 1996 Kaplan and Norton developed the concept of the Balanced Scorecard which became one of the most used system for measuring innovation in performance. Some companies were struggling with the introduction of such a performance management system. Žižlavský (2016), as a result, developed an appropriate concept for the introduction of a Balanced Scorecard system that was fit for the intended purpose for the Czech Republic. The implementation of this system incorporates the theoretical background, the structure of this process and how decisions need to be made within a control framework that includes decision gates. The whole system is made up of task definition, planning, controlling, evaluating or reviewing, and supporting (Žižlavský, Vaverková, 2018).

This framework is based on Balanced Scorecard methodology. Its essence lies in the balance between short-term and long-term goals, inputs and outputs and also financial and non-financial indicators. Short-term goals have usually the form of operative goals and long-term goals are strategic goals. To be effective and successful, an Innovation

Scorecard must be structured and show a logical approach so that implementation can be achieved without a minimum of disruption. According to Niven, a series of individual elements make up a unit with each part playing a vital and important role to make up the whole (Niven, 2014).

Every business model of an innovative project must contain inputs and outputs. Every part of every model should be driven by the processes by which inputs are converted to outputs and the results of the model itself. The composition of such a business model is one of the most challenging parts of innovation as management needs to agree on which innovation model to choose for application and roll out within their company. The "input-process-output-outcomes" model is based on a separation between formulation and implementation and consists of various inputs and outputs in the form of technical, organizational and business activities. Basically, it is about identifying the differences between input, process output and resultant measures. There is a relationship between input and outputs and a closer inspection of this reveals that both these elements are required to develop the concept of innovation (Brown, Svenson, 1988).

2.3.1 Design of the system

Almost every time we try to measure something, we want to get some results. In measuring innovation, the measurements should be effective and economic. Effective measurement is the provision of information for the management of company and economic measurement means financial (for a reasonable price). However, the individual indicators are very inadequate, because they mostly meet only one (economic) condition and therefore only perceive innovation from one side (Žižlavský, 2016).

For this reason, it is important to use a complex system with several indicators to assess the company's capabilities or performance. These indicators explore the innovation process from multiple sides and thus represent the real image of the process complexly. We could not achieve that by using individual indicators. The benchmark of this method as a comprehensive measurement system is the Balanced Scorecard which measures innovation performance for the entire company (Horváth, Partners, 2002).

Figure 2: Implementation of Innovation Scorecard

(Source: adapted from Horváth, Partners, 2002, modified from Cokins, 2009)

According to Keegan et al. (1989) there exist three steps in measurement system. The first defines the strategy and goals of the company. The second develops a set of performance measurements and the last goal focuses on linking the performance to any existing management system. The Innovation Scorecard gains from the long-term experience of the process of implementing a Balanced Scorecard within a company (Horváth, Partners, 2002).

The definition of an innovation strategy forms the basis for the implementation of an Innovation Scorecard concept. It is important to determine the strategic plan, to have clear and unambiguous leadership and to allocate appropriate resources. This particular part of the innovation process is governed by the authority vested in senior management. Strategic Goals are the main aspect for the Innovation Scorecard and also have a relevant and direct impact on the successful implementation of any Innovation Scorecard initiative. Critical Success Factors (CSF) form part of any innovation system. They depict the most important success factors (high level only, at strategic level) that need to be introduced as part of any innovation system roll out. At the level below CSFs, so-called Key Performance Indicators (KPI) need to be developed. These allow the company to focus on the operational level performances in line with the higher-level CSFs. Another useful tool is the established and implantation of innovation maps. These help companies to focus on strategy performance rather than strategy formulation. They provide important communications to assist with translating the innovation strategy into business goals necessary to realize the execution plan. Metrics are tools for expressing innovation goals. These must be clear and free of errors and achievable. Target values describe the innovation goals, and these are determined at the beginning of the process in detail. They should be primarily achievable but also credible. They must be reality-based and have to be measurable (Žižlavský, 2016).

Performance measurement system design is another important part of design of any Innovation Scorecard. Many authors were responsible for its inauguration. Neely et al. (1996) defined it as *"the process of quantifying the efficiency and effectiveness of actions".* But Moullin (2002) on the other hand claimed that it is "*the process of evaluating how well organizations are managed and the value they deliver for customer and other stakeholders".* And Aubrey Daniel (2004) created the term performance management and linked this concept to the art of leading by focusing on managing behaviour and results. In 1997 Bititci et al. defined the performance management as *"the process by which the company manages its performance in line with its corporate and functional strategies and objectives"* (Lehner, 2016).

Neely et al. (1996) also defined a performance measure as *"a metric used to quantify the efficiency and/or effectiveness of action"* and as well as a performance measurement system as *"the set of metrics used to quantify both the efficiency and effectiveness of actions".* He represented all these relationships (Figure 3) at three different levels: at individual, entity and at relationship level (Neely et al., 1996).

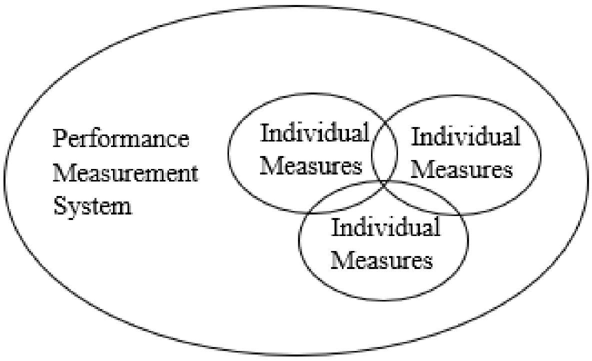

Figure 3: A relationships in Performance Measurement System design

(Source: Neely et al., 1996)

Globerson (1985), Maskell (1991) or Bourne et al. (2003) defined these principles of performance measurement system design and this is presented in detail in Table 1. Some of these principles are similar to those identified by Neely.

Table 1: Principles for Performance Measurement System Design

Globerson (1985)	Maskell (1991)	Bourne et al. (2003)
Performance criteria must be chosen from the company's goals. Performance criteria must make possible the comparison of companies that are in the same business. The purpose of each performance criterion must be clear. Data collection and methods of calculating the performance criterion must be clearly defined. Ratio based performance criteria are preferred to absolute numbers. Performance criteria should be under the control of the evaluated organizational unit. Performance criteria should be selected through discussions with the people involved (customers, employees, managers, etc.). Objective performance criteria are preferable to subjective ones.	The measures should be directly related to the company's manufacturing strategy. Non-financial measures should be adopted. It should be recognized that measures vary between locations – one measure is not suitable for all departments or sites. It should be acknowledged that measures change as circumstances do. The measures should be simple and easy to use. The measures should provide fast feedback. The measures should be designed so that they stimulate continuous improvement rather than simply monitor.	Performance measurement refers to the use of a multi-dimensional set of performance measures. Performance measurement should include both financial and non-financial measures, internal and external measures of performance and often both measures which quantify what has been achieved as well as measures which are used to help predict the future. Performance measurement cannot be done in isolation. Performance measurement is only relevant within a reference framework against which the efficiency and effectiveness of action can be judged. Performance measures should be developed from strategy. Performance measurement has an impact on the environment in which it operates. Starting to measure, deciding what to measure, how to measure and what the targets will be, are all acts which influence individuals and groups within the company. Once measurement has started, the performance review will have consequences, as will the actions agreed upon as a result of that review. Performance measurement is an integral part of the management planning and control system of the company being measured. Performance measurement is being used to assess the impact of actions on the stakeholders of the company whose performance is being measured.

(Source: Žižlavský, 2016)

2.3.2 KPIs

Key Performance Indicators are used for the efficient performance of work. Managers use KPIs to understand how well and effective any business performs and how successfully any business is managed. They can help to clarify how organizations work with, for example, CSFs. This information is very important for the effective and efficient management of any company. KPI helps to highlight the critical importance of metrics and two of the popular sayings are *"What gets measured gets done"* and *"if you can't measure it, you can't manage it"*. Many managers do not take KPIs into consideration and this causes them problems in the long run (Marr, 2012).

Niven defined seven characteristics of KPIs. First of all are non-financial measures when he claimed that all of KPIs indicators are definitely non-financial because they cannot be expressed in financial terms. Secondly, there appears to be a general opinion that KPIs should be measured frequently such as daily or weekly. If KPIs are monitored monthly, quarterly or annually, they cannot be a KPIs that are relevant to any business. Thirdly, the most senior staff of companies need to be made aware of the results of having KPIs in their companies so they can improve strategic decision-making and act upon the results of any KPIs. It is equally important that staff within companies understand why it is important that KPIs are employed within their organization and what it is they need to do to act and support KPIs for their own work areas. This is also true to say for all teams including any cross-functional working that may be in place such as Projects and Software Development "sprints" within Agile/SCRUM companies such as Red Hat. The main focus of attention should always be on CSFs and KPIs that provide the greatest return on investment and that have the greatest positive impact on how the company performs (Parmenter, 2015).

According to Harold Kerzner (2017) *"all KPIs are metrics, but not all metrics are KPIs"*. The biggest difference is that metrics are general but KPIs are specific. Most companies use just metrics alone because they do not know what the purpose is of using the KPIs in the company. KPIs help companies to make decisions. This explains why effective KPIs have specific performance targets associated with them. KPIs should be able to help predict the future and also be measurable to express the results. KPIs should be able to help predict the future and also be measurable to express the results. KPIs should be able to used as catalysts for changing the future and results should show both negative and positive outcomes so that lessons can be learned and appropriate changes made in future. KPIs should be automated to eliminate human error as much as possible and to have the right number of KPIs for the business (Kerzner, 2017).

2.3.3 Innovation Metrics

Innovation metrics are tools for measuring the innovation processes in the company. They help to find out if the company is prepared to innovate or not and metrics also measure the effectiveness of applied innovation strategies. Innovation metrics are a diagnostic tool that can predict the possibility of future capacity of innovate. They have other uses, too. For example, in the area of communication where it is important to define and describe the strategy. The next part is control which is focused on monitoring the implementation of innovation strategy and its impact. And the last is learning which contains the identifying the new opportunities (Trias de Bes, Kotler, 2011).

These metrics are divided into three types. First is Input metrics which include financial and other things which are put at the beginning of the innovation processes, for example R&D. Output metrics are the second type and they include the products of the innovation processes or system, especially the results of innovation effort. The last metrics are Outcome and they represents the goal or aim of the process. It can be the impact of the successful innovations into the company. All these metrics can be intangible in case of ideas or practical problems, tangible or human in case of laboratories or scientists (Grubler, Wilson, 2014).

There is another different category of indicators used in R&D performance measurement. Each of these authors in the table below divide the indicators according to their views and opinions. Werner and Souder separate the performance indicators into qualitative which means objective indicators and also quantitative, or alternatively subjective indicators. Brown and Svenson had the same title as Grubler and Wilson but they added a process which is described as an analyzing tool during activities. Technometric and Bibliometric are special because first is based on patent data and the second is based on research publications. Financial and Non-financial indicators are described below the table (Chiesa, Frattini, 2009).

Table 2: Classification of Performance Indicators

Indicators	Authors
Qualitative and Quantitative	Werner, Souder (1997)
Input, Process, Output, Outcomes	Brown, Svenson (1988)
Technometric	Debackere et al. (2002)
Financial and Non-financial	Bremser, Barsky (2004)
Bibliometric Indicators	Verbeek et al. (2002)

(Source: Chiesa, Frattini, 2009)

There are financial and non-financial metrics which were selected for this thesis. The Innovation Scorecard tries to find a balance between these two metrics. It is also about the balance between operative and strategic goals, input and outputs and also internal and external factors (Žižlavský, 2016).

Financial indicators can provide relevant evidence whether the company is achieving value and it can be used to inform senior management, for example, if the decisions they made had any positive or negative impact on business performance. They are connected with goals which are determined for a short time and they also have a base on accounting data from history. It means that they are not suitable indicators for strategy and future development (Kislingerová, 2008).

There are three groups of the financial indicators. Some indicators can help the company to check how effective innovation has been as far as competitive advantage is concerned. This includes topics such as return on sales, liquidity and productivity. Second group are the indicators which introduce the economic results of implementation the innovation plan in the company. This category includes profitability indicators. The last group of indicators inform about the financial effects of innovations. This group contains the profit ratio or the rate of return (ROI). The percentage of revenues from new products or services, the profits which is generated from new products or the percentage of growth in new products or services are next innovation metrics (Cooper et al., 2004).

Non-financial indicators help managers understand the connection between innovation goals and associated resource allocation. These metrics can identify key factors that represent the development of financial indicators. These are much more sensitive to changes than the financial indicators (Kaplan, Norton, 2001). They are very important because of their role in internal processes and also at the corporate level. Long cycles of development benefit from cycles of innovation much more than operating cycles because of the longer time it takes to create a value (Bremser, Barsky, 2004).

2.4 Agile Manifesto

Agile is an approach to software development and includes the evolution of requirements and solutions. It emphasizes teamwork, customer collaboration and the ability to respond quickly to change. It represents the flexible and iterative process in which the customer engages and thus gains very fast feedback. It is a set of behaviors, concepts and techniques that represent this agile approach. Essential points include cooperation, mutual trust and self-organization. An important part is the planning of iterative and incremental approaches to resolve issues/develop solutions and planning work at certain time intervals known as sprints. The communication is very important because it can help to understand what needs to be done, by when, how and by whom (Šochová, Kunce, 2014).

Development methods have been used since the mid-1990s. Then an alternative way of developing software was developed to compete with a waterfall model that was less flexible at the time. The document known as The Agile Manifesto which contains the 12 agile approach principles, was created by a group of software developers in 2001. At present, this approach is still being used by more organizations even outside IT (Mathis, 2013).

Principles behind the Agile Manifesto:

1. *"Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.*
2. *Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.*
3. *Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.*
4. *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 behavior accordingly."* (Millett, Blankenship, Bussa, 2011).

In addition to these 12 principles of agile software development, the authors also defined four values that are recommended to follow and are suggested to be considered to be agile:

- *"Individuals and interactions over processes and tools*
- *Working software over comprehensive documentation*
- *Customer collaboration over contract negotiation*
- *Responding to change over following a plan"* (Šochová, Kunce, 2014).

It is suggested that communications and individuals have greater significance and efficiency than processes. Documentation is usually lengthy, so it is better to use a working software that need a timely update. Working with a customer perceived as part of a team is based on communication for better results and user satisfaction. In response to changes, the plan may be affected, but in the end, it will improve the quality of the final product (Šochová, Kunce, 2014).

2.4.1 Agile methodologies

The decision about using Agile in company means that the company uses tools and techniques of Agile to develop their products. It depends what methodology the company choses, but it is possible to combine these. The first methodology is called Extreme Programming (XP) which was founded by Kent Beck and its advantage is an efficiency, customer focus and feedback and also quality. It has five values which are communication, simplicity, feedback, courage and respect. Scrum methodology introduces a teamwork and the organizing in the team and was founded by Jeff Sutherland and Ken Schwaber. A benefit of Lean Software Development from Mary and Tom Poppendieck is that it can eliminate work which cannot create customer value. The final one is known as the Kanban Method which can present just-in-time development and workflow and was developed by David J. Anderson. There are other Agile Methodology which are not so famous, but for example, Feature-Driven Development, Dynamic Systems Development Method (DSDM) and Crystal Family (Ashmore, Runyan, 2014).

2.4.2 Scrum

Scrum or sometimes referred to as “an agile project management framework“, its focus is on the use of an empirical process, which means rapid and effective reaction to change. It also helps to introduce value products while still controlling and adapting the process. Scrum is one of the agile approaches to product development that uses feedback and collective participation on the project. The right product can only be created by integrating the whole company into the project as well as teamwork with the customer. This method provides guidance on recommendations, but it is not specified how to achieve this. If this approach is desired, a specific team must be set up to manage this (Sliger, 2011).

This method was first discussed in a Harvard Business Review document in 1986 in Japan. Hirotaka Takeuchi and Ikujiro Nonaka introduced new possibilities for project team management on examples from the automotive industry. The name "Scrum" was given based on a Rugby term which describes the process of re-starting the game after an interruption due to an unwanted interruption. Software development companies began using the agile scrum process in 1993. Ken Schwaber used Takeuchi and Nonaka's thoughts how to develop software were shared in 1995 by presenting these experiences with Jeff Sutherland on a program convention in Austin (Zikmund, 2010).

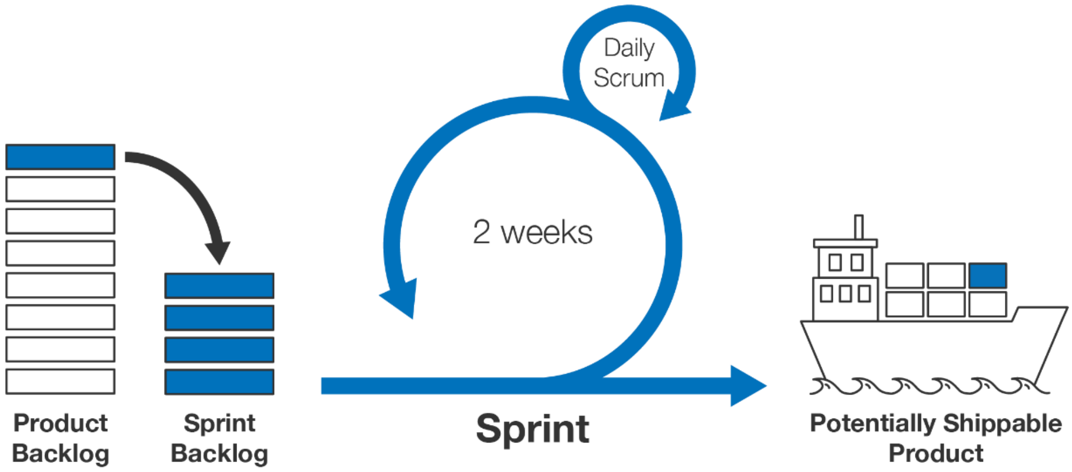

Figure 4: Sprint Planning Scrum

(Source: Kunz, Leigh, Associates, 2018)

At the beginning of the work, a list of tasks to be completed at the end of the project is prepared. Depending on which task is to be done earlier, this task gets more priority. Tasks so ordered, sorted by priority, are called product backlogs. In order for team-customer cooperation to be better, user stories are created which is a presentation of what the user wants and what activities they want. Teamwork is the basis of the sprint (iteration) which is a repeating time unit that has a fixed length, usually lasting 2 weeks. Before the start of the sprint, a sprint backlog is set, a list of tasks that should be fulfilled, with which the team is working throughout the sprint. The everyday stand is a daily stand-up, where each member of the team reports on what they have done and what they plan to do on that day, or reporting what has failed. This meeting should not take longer than 15 minutes. When the work is completed, a sprint review follows when the team presents the work to the customer (Šochová, Kunce, 2014).

Typical Scrum activities are Sprint Planning Meeting, Daily Scrum (or Stand-up), Sprint Review and also Sprint Retrospective. Scrum does not have typical team roles like in classical project management methods. The following are the most important three specific roles in Scrum:

Product owner

The person who is responsible for defining the features and the owner of the product. They are in contact with the team every day to answer questions and define items that will lead to the right goal. Primarily, this person must understand the product and then describe it to the whole team, including how to achieve it (Šochová, Kunce, 2014).

Scrum master

The leader of the team and also the person who is responsible for team communication and with people outside the team. They try to remove any obstacles that would lead to an unsuccessful project and at the same time its main objective is to create an independent and efficient team that would achieve the highest level of performance. It is the person who is part of the team and should be available at any time and they need to share their workspace directly with the team (Šochová, Kunce, 2014).

Development Team

These are the members of the team who work on the product. They are collective responsible for delivering the product on the basis of tasks that are entirely in their charge. This team is multifunctional and mutually substitutable. Each member of the team is an expert on other issues and collectively forms the effective whole. The development team includes testers, designers or UX specialists and also operational engineers in addition to developers. This is the reason why the Scrum Teams are called cross-functional (Šochová, Kunce, 2014).

3 ANALYSIS OF THE CURRENT SITUATION

This part provides a detailed business overview of Red Hat and its current operations, including detailed descriptions about the company history, business overview, its link with Brno University of Technology, products and services, with particular emphasis on the Atomic Host initiative and the related introduction of an Innovation Scorecard. Red Hat's internal processes are analyzed to gain a better understanding how the business functions and operates. One of the main objectives of this study is investigate and establish any gaps that may exist within current business processes that would benefit from improvement through the adoption of an Innovation Scorecard system.

3.1 History of the company

The ACC Corporation was founded in 1993 by Bob Young. This company sold Linux and Unix software complement. In 1994 Red Hat Linux (RHL) was founded by Marc Ewing. Young bought Ewing's company in 1995 and they both created Red Hat Software together. Young became CEO. In 1999 the company became a publicly traded company. Marc soon left and Bob began looking for his own replacement for himself.

In 2000 Matthew Szulik became the next CEO of the company. For a long time RHL was sold as a box product right next to Microsoft Windows and Lotus Notes in retail stores. The new version was released every six months and the company hoped that customers would buy it regularly to gain new features. The year 2002 is the latest issue of RHL and when it entered the RHEL market. It became flagship and at the same time offered the most demanding data centres in the world. RHEL has led to more than ten years of business growth and has provided resources and flexibility to invest and participate in other open source communities. This made it possible to add more features and options to meet customer demands, needs and requirements.

In 2005 Young left the board and officially stepped down from the company. A year later Red Hat acquired JBoss which first expanded its product portfolio outside Linux. In December 2007 he became President and CEO of Red Hat Jim Whitehurst. Under his leadership the company was still expanding. In 2012 the company became the first open source company to make more than 1 billion dollars in revenue. In 2013 the company joins the open stack project and another year is released by RHEL 7. Over the 20 years Red Hat has grown from a room to a skyscraper.

The name Red Hat comes from the red Cornell lacrosse cap which belonged to Marc Ewing (one of Red Hat's co-founders). When Marc worked at his job, he wore his grandfather's red hat. He helped fellow students in the computer lab and he was known for his cheerful nature. The advice to find a right person was "If you need help, look for the guy in the red hat." When he started distributing his own version of Linux, he chose Red Hat as the company's name.

The company focuses on providing open-source software solutions. The term open-source is something that can be changed because its design is publicly accessible. It basically means sharing codes in software development where accessibility concerns the source code. It is a cooperative and creative process that solves common problems and ensures that these solutions will be known.

3.2 Red Hat Czech s.r.o.

Red Hat Czech s.r.o. has been operating in the Czech Republic since 2004. In 2006 it was registered in the Commercial Register and the first branch in Brno was opened. It focuses on testing and software development. The Red Hat company has over 500 employees in 2012 and over 700 employees in 2014. At present the company has over a thousand employees. The company currently has four buildings that Red Hat rented in Brno. The interiors were created in collaboration between architects from Great Britain and the team of global solutions for Red Hat's work environment. The environment is in line with the Red Hat operation model and the state-of-the-art building also has its own gym, wellness centre, library and relaxation and game zones.

This company has become Red Hat's largest development center in the world and has also received a number of awards. For example, the tittle Best Employers of the Czech Republic from 2010/2011 according to the company Aon Hewitt. The company's headquarters won 3rd place in the competition CzechInvest agency and foreign investment association – AFI business property of the year 2010 in the category of high-tech real estate of the year. In the Czech Republic has been operating for several years with the support of CzechInvest. The largest development center of the American software company Red Hat was also created in Brno thanks to the subsidy from the Operational Program Enterprise and Innovation.

3.2.1 Red Hat University Program

Red Hat supports the work of engineers, university scientists and students on open source research projects. The aim of this project is to give students the opportunity to work on real projects. This makes it possible to discover talents, and it is a recognized possibility to realize their ideas in reality. Red Hat participates not only in student activities but also in applied research. The software product portfolio provides many opportunities to use the latest methods and techniques, particularly in software and security. Working with Red Hat is a simple procedure. Just select a project. Candidates do not need to be developers to join the project. They have adopted flexible recruitment approach to gain and retain the best talent in this industry. They are not limited by semester terms or team size. It does not matter whether the project is only for a short period of time or long-period cooperation. Associated Partner Universities are Boston University, Czech Technical University (Faculty of Electrical Engineering), Brno University of Technology (Faculty of Information Technology) and Masaryk University (Faculty of Informatics).

3.2.2 Development Conference

Red Hat organizes free community conferencing annually for developers, administrators, DevOps engineers, testers, documentation writers and other contributors. The conference welcomes speakers from a wide range of backgrounds and experiences. It provides speakers with perfect opportunities to improve their presentation skills. Its primary purpose is to provide opportunities for developers and contributors who handle the future development of previous projects and share updates and announcements of recent progress. Red Hat offers a variety of scholarships that cover a range of non-speakers who are students or return to full-time work within Red Hat’s technology business section. DevConf is region-oriented for Europe, the Middle East and Africa, and scholarships are available to speakers from these locations.

3.3 Red Hat Industries

Red Hat technology is already used in many industries such as Financial Services, Government and the Public Sector, Healthcare and Life Sciences and Telecommunications.

Financial Services

The financial services industry is a very competitive environment. Critical are speed, safety and agility. Companies are causing changes across the industry by changing the competitive environment. These companies use enterprise open source software to keep up with demand, adopt innovative technology and manage costs while maintaining security.

Government and Public sector

Red Hat products are designed to help them become more efficient and meet critical IT requirements. RH takes the best innovations from the open source community and stabilizes it for a public sector mission. The company helps to complete associated accreditation and certification processes. This Red Hat solution is used throughout the federal government and in all 50 states.

Healthcare and life sciences

Red Hat as the world's leading open source provider, has the expertise to create and integrate solutions to the current healthcare environment that is constantly evolving. These organizations must adhere to the changes that are linked to the new regulations and the law on affordable care.

Telecommunications

Red Hat offers a comprehensive open platform that helps service providers deliver innovate and faster new services to the market safely and efficiently. Telecommunication Services providers are looking for agile IT/software solutions in readiness of the expected 5G network roll-out expected during 2019. Other areas include the Internet, Artificial Intelligence (AI) and robotics.

3.4 Red Hat Products

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

Linux platforms can manage tens, hundreds and thousands of servers as easily as a single one. Platforms are easy to use for system management for growing Linux infrastructure. They are based on open standards and functional modules that enhance Red Hat Enterprise Linux's (RHEL) management capabilities.

Middleware is a multipurpose software that provides services to applications beyond those offered by the operating system. It covers everything between kernel and the user applications. It works on a software connection to other software. It can be considered as an application due to data flow from one application to another. Middleware is divided into Accelerate, Integrate and Automate.

Virtualization platforms are technologies that allow to create more simulated system environments. The software then connects to the hardware and allows the division of one system into multiple ones. The essence is to separate the device source from the hardware and distribute it appropriately.

Cloud computing is a set of principles and approaches to deliver computing infrastructure, services, platforms and applications-sourced from clouds to users on demands across a network. Clouds are supplies of virtual resources that are controlled by software and can be accessed by users on demand.

Storage is the process by which information technology archives, organizes and shares the bits and bytes. Computers have a short-term memory that is processed by RAM. It remembers all past activities. This memory is cleared by transferring all entries to the storage volume.

DevOps forms part of **Management**. It describes approaches to speeding up processes, from development to deployment, in a production environment where the user can provide value. These approaches require communication between development teams and operational teams. Empathy among co-workers is important. DevOps developers collaborate with information technology operations to speed up software creation and testing.

Available Services is an Open Innovation Lab where people will learn, with the help of Red Hat experts, to use agile methodology and an open source tool to work on their enterprise's business problems. There are also Certified Training Courses. These courses help master Red Hat technologies and certificates will be issued to those who achieved the necessary pass rate following the exam at the end of courses. Company Consulting is a service offered to companies who wish strategic consultants to analyse their organization's operational and strategic issues and find solutions to fix these.

3.4.1 Linux

Linux is a free and accessible open source operating system and IT infrastructure platform. It was originally created in 1991 and it is based on a Unix. Its name is derived from its creator Linus Torvalds and the ending "x" in Linux refers to mentioned Unix. In the world of operating systems, Linux is the best-known and most-used open source system. The source code for Linux is free and available, there are several different distributions, including Red Hat Enterprise Linux, which is flagship product of RH, and Fedora Linux, which is a community project founded by Red Hat to develop a desktop version of Linux. There are licenses to protect the source data. The developers of software create applications and other services inside Linux containers. These applications let them code once and then run their code anywhere. All containers applications contain some part of a Linux distribution.

3.4.2 Containers

Linux containers are technologies which enable to package and isolate applications with their runtime environment – all of the files that are needed to run. Containers are boxes where everyone works within that box. It means they concentrate only on the part which with they work with and thus separating the area of responsibility. This helps to reduce conflicts between development teams and operating teams. They work on all levels and at all stages. They help to simplify and accelerate application development. Due to the fact that Linux containers are based on an open source technology, news can be accessed as soon as it becomes available.

3.4.3 Red Hat Enterprise Linux

Red Hat Enterprise Linux is an operating system platform that runs on a range of hardware. It is used in physical, virtual, container and cloud environments and it is available in multiple variants. RHEL became a flagship product that became very attractive to the world's most demanding datacentres. This Linux distribution is intended for commercial sphere solutions including mainframes. New versions are published every 1 ½ to 2 years. The source codes of the open source programs are freely accessible including updates.

The first version of the Enterprise was released in 2000. The first version of the RHEL series was launched in 2002 and its current name was established a year later. The RHEL life cycle is divided into three levels. The first phase is full support for new hardware, fixing all bugs through updates and creating update ISO images. The second phase includes extended support, followed by maintenance. One of the variants is also Red Hat Enterprise Linux Atomic Host which is designed to run applications with Linux containers.

3.4.4 Red Hat Enterprise Linux Atomic Host

Red Hat® Enterprise Linux Atomic Host is a secure, lightweight, and minimal-footprint operating system optimized to run Linux containers. A member of the Red Hat Enterprise Linux family, Red Hat Enterprise Linux Atomic Host couples the flexible, modular capabilities of Linux containers with the reliability and security of Red Hat Enterprise Linux in a reduced footprint, to decrease the attack surface and provide only the packages needed to power hardware and run containers.

With this offering, Red Hat combines:

- An enterprise-class container-specific host.
- New container capabilities in the world's leading enterprise Linux platform.

- A certified program for containerized applications.
- An extensive ecosystem of support and services.

Red Hat’s vision for containerized application delivery on an open hybrid cloud infrastructure is comprehensive, including portability across bare metal systems, virtual machines and private/ public clouds.

By choosing Red Hat Enterprise Linux Atomic Host, customers can take advantage of the fast pace of innovation from open source community projects like the Docker project and Project Atomic while maintaining a stable platform for production deployment. Customers can concentrate on customizing and developing containerized applications while Red Hat maintains the underlying Linux platform on which these depend.

3.5 Atomic Host

This part describes the Atomic Host system in detail. It is a variation of Red Hat Enterprise Linux 7 optimized to run Linux containers. The Atomic Host process will be explained in sufficient detail including how it works in a practice.

3.5.1 Process

Each updating batch is planned to be developed, tested and released over a period of 6 weeks. Work starts with a planning phase that typically lasts for 10 days. It is based on the outputs from Sprint Planning Meeting(s). These meetings are attended by the Product Owner, Product Manager and team members. The Product Owner presents a set of features he/she would like to see completed in the sprint (the “what”). The team then determines the tasks needed to implement these features (the “how”). Work estimates are reviewed to see if the team has the time to complete all the features requested in the sprint. If yes, the team commits to the sprint. If no, the lower priority features go back into the product backlog until the workload for the sprint is small enough to obtain the team's commitment.

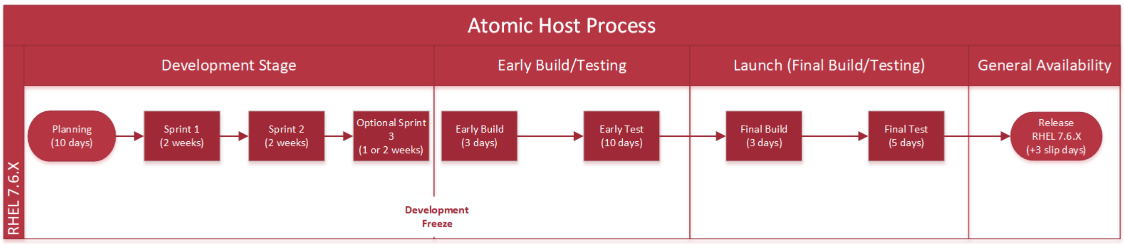

Figure 5: Atomic Host Process

(Source: own processing)

The development phase follows. This includes individual sprints and usually lasts for 30 days. Development work is managed and completed applying an agile method known as Scrum. It consists of two sprints and each takes 2 weeks. It can also include a third optional sprint if it is needed (this decision is made during the planning phase).

In the development phase, everything is based on the plan (backlog(s)) coming from preceding planning phase. At the end of the development phase, there is a development freeze. It means that the development is closed for any further work to be undertaken. Developers confirm and provide relevant information that all packages are ready and available to proceed to the testing stage. The testing phase consists of two steps. First, Early build and testing is performed. The building process takes 3 days. Its purpose is to put together all bits and associated packages. Early testing follows which takes 10 days. If there is a delay in the building process such as 3+2 days, then Early testing has to be shortened such as 10-2 days. Some lateness in Early testing is not a big issue. Early testing is deployed to identify any issues/problems as early as possible and then take corrective action. Any delay in this phase does not have a big impact on the general availability (GA) of updated batches.

When early testing is completed successfully, Final build and testing (also referred to as launch) follows. The process of this launch is the same as Early build and testing, for example, to build final packages and appropriate containers and then test them. Time allowed for the Final build is 3 days. Testing takes 5 days. If passed successfully then an Atomic Host is created. Any issues during the launch phase have a huge impact on the final GA. An agreement is in place that GA cannot slip by more than 3 days.

3.5.1 Communication flow

With the exception of senior management communication flows in Red Hat start with the Programme Manager who has five technical teams (QE, Release Team, Development Lead, Container Owners and QE Lead) directly reporting to him. In addition, a Red Hat Project Manager reports to the Programme Manager. There Each team consists of a mix of team members of different countries from across the world. This means that communications can be difficult at times due to prevailing different time zones. However, the team mainly communicates via e-mails and also video conferences. Delays can be expected due to these inherent limitations.

The iScorecard Team works closely with the external team and this enables them to communicate simultaneously with both the Programme Manager and individual technical teams. Direct communications with senior management are also an integral part of the communication process. They usually communicate via e-mails as a daily routine and arrange meetings with the Red Hat Teams when it is needed. All lines of communication follow the hierarchical structure below (Figure 6).

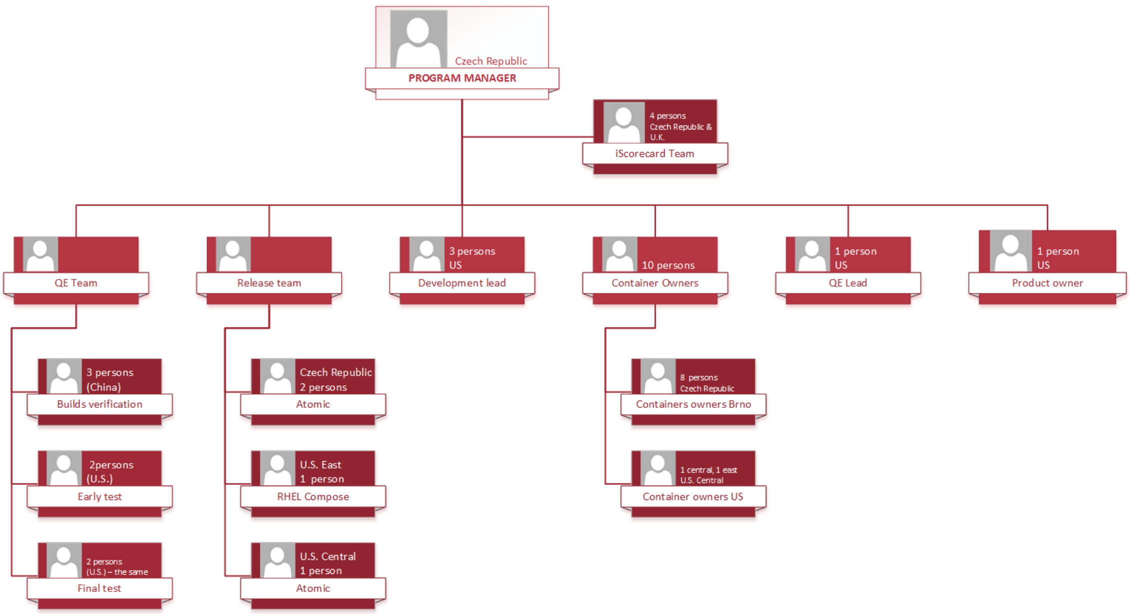

Figure 6: Communication Flow Diagram

(Source: own processing)

3.5.2 Build

The build process includes two steps: Early and Final (launch) build. The process in both steps is identical. At the beginning, the program manager (PgM, located in the Czech Republic) gives instructions to RHEL release team where Compose and three Images are created by the release team (located in the US). These are further verified by QE (also in the US). If the created images are not correct or acceptable, they have to be returned to RHEL. If they are verified and accepted, they will be submitted to Extras Compose (source code – the components that are open source so they can make everything available to customers) plus YML file (The YML file is the information coming from RHEL and extras.). Extra compose is completed in the Czech Republic. Once this has been completed, their roll-out commences via the containers at global level ('ovirt' is created specifically for use in the Czech Republic and 'openvm' specifically for use in the US. It can also be launched via containers outside of the US and/or Brno. The process is complete and the Atomic Compose emanates from this.

The Build Process (Figure 7) has two main challenging areas. Location appears to be a major concern due to the different time zones involved. Overall, the process starts in the Czech Republic and is triggered by the Program Manager, but the release team is located in the US. As a result, the team starts preparation work (compose and images) minus 6 or 7 hours behind Czech time. The verification by QE is also conducted in the US. Then some activities that follow are completed in the Czech Republic, at global level and/or US. Working in different time zones causes undesirable delays. Teams have to wait until working hours commence in countries other than their own. There are dependencies between working activities in different countries (sequential). This causes some repetitive manual work being done during periods of time overlaps. This causes frustration and appears to demotivate team members in various locations. It also reduces team members' time to be creative and proactive. The second area is the flow of Communication. The person mainly affected by this is the Program Manager. Over 40% of their time is spent communicating effectively with the various parties involved in the project. This is exacerbated by the need to communicate with team members who work in the different time zones mentioned earlier.

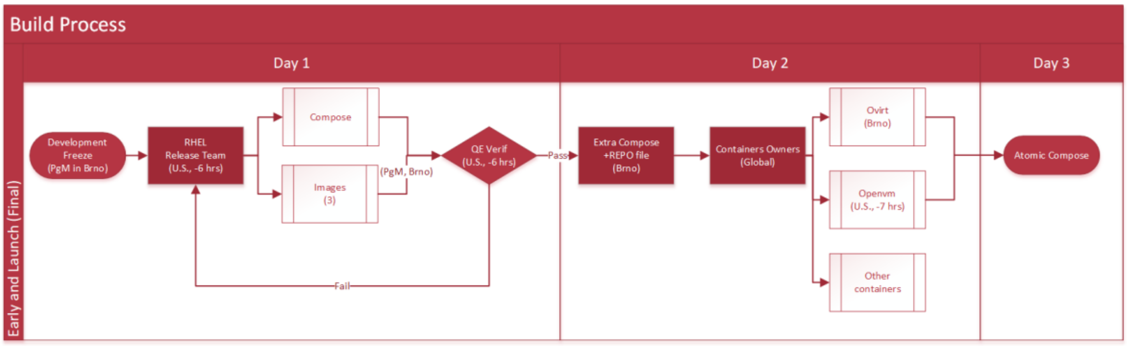

Figure 7: Atomic Host Build Process

(Source: own processing)

Risks

The following areas that form part of the build process have been identified by Red Hat senior management to carry high risks. These risks need to be managed by the Red Hat project manager. The identified risks form part of the build process:

- QE
- Extra Compose
- Containers
- Atomic Host Compose

3.6 Containers Rebuild Process

This master thesis focuses on Containers Rebuild Process in support of the required process innovation automation tool, to provide support in terms of information, to provide metrics and to show impact/improvements of the automation system. The whole process begins on the first day when the release engineering team creates ‘compose and base’ images based on available latest packages. The output from this activity is passed on to the QE team who will check if everything works. If yes, the process continues in the area of layered container images. Follow on phases are completed on the following day. If there are any issues, the previous process of ‘composing and base images’ is repeated and then re-tests commence. The release team will inform the Program Manager once this work has been completed successfully. This includes details of which compose URL is available and what the name is of the Base IMG. The Program

Manager will then inform relevant parties of what it is they need to do next. Rebuild developers can commence their tasks once the Dockerfile has been updated. The build process can now be ‘run’. Once the build process has been successfully completed, the owner will review the Errata tool and create a new rebuild bug in the Bugzilla service. This is the end of this phase. Next phase is testing (QE). Once all testing has been completed and the release of the new build has been approved, the new and updated container image is uploaded to the public repository so it can be accessed by customers.

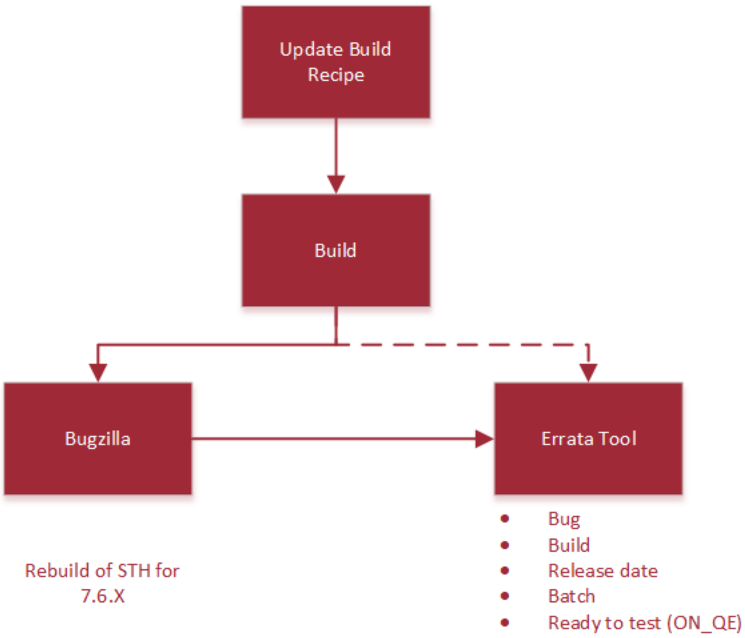

Figure 8: Update Build Recipe

(Source: own processing)

The objective of the process innovation within the ‘Containers Rebuild’ process is to eliminate repetitive manual work conducted by container image owners within RHEL’s Atomic Host. The result of the automation work will shorten the Container Rebuild process and provide team members with more time to focus their attention on fixing customers’ problems and becoming more creative. The automation process needs to respond positively in the event of failure. Should this occur, the new system will automatically, without any need for human intervention, return to the beginning of the process and resolve any issues. Another objective is to improve the current Release Engineering processes. The dedicated automation team will be engaged in this change work.

The above-mentioned areas need to be clearly and fully understood by the whole Container Team, with regard to the Atomic Host project so that there is no confusion in "what we mean", "what this is used for", "who is the owner / point of contact" and "what is the tool there to help". Horvath's (2002; 2016) suggests that it is important to use and apply experience and the power of words and using numbers to clarify meanings and intentions. Put differently, goals, critical success factors (CSF) and key performance indicators (KPIs) are described verbally in combination with metric and any target values. It appears that this will ensure that goals can be achieved in a quality way. This is the main quality criterion for the Innovation Scorecard, and it has a significant influence on any successful launch and implementation of, for example, software development initiatives.

The original intention of Target Values was to set hard values. These values would be identified on the basis of traffic lights system known as RAG (Red, Amber and Green). Red highlights significant issues, Amber is used to highlight some issues and Green means all is well. It was not possible to use this system within the Agile environment due to the inherent nature of Agile (discussed earlier in this thesis). As a result, a new method was developed that was based on setting monitoring and following trends (increasing/decreasing).

An automation tool is going to be implemented in the Container Rebuild Process. The defined goals and critical success factors are:

1. Introduce an Innovation Scorecard System for Atomic Host: Container Build Process
    a. Produce high level project documents
2. Container Automation Build Process
    a. Develop/buy automation tool and implement it
    b. Improve modus operandi
    c. Improve the Design and Container Build process reporting to improve the communication flow
    d. Effective Dependency Management during Container Build Process

Table 3: Workflow Comparison (Red Hat)

No Automation	Automation
• If owner makes changes, they do not receive any response about potential breakages/problems • PgM sends out email to request new build • Owner has to update all Dockerfile versions • Owner creates new Bugzilla • Owner creates new erratum • Owner creates new image • Owner might test the image manually • Owner attaches new image to erratum • Owner switches the erratum to ON_QE	• Bot runs a scratch build, immediate feedback loop secured • Bot scans Dockerfile for error • Dockerfile versions template • Owner merges changes • Bot syncs changes to dist-git • Bot makes a production build • Bot creates new Bugzilla, erratum, attaches production build and switches the erratum

(Source: own processing)

4 PROPOSED SOLUTION: INNOVATION SCORECARD DESIGN

The implementation of the automation tool (Table 4) is planned to be completed in three rounds. The early build in Round 1 is going to be done manually in order to ensure that, overall, the RHEL Atomic Process will not be interrupted. The final build will be completed non-manually (automatically).

Table 4: Time Schedule of Container Rebuild Innovation Process

	Round 1 RHEL Atomic Host version 7.6.2	**Round 2** RHEL Atomic Host version 7.6.3	**Round 3** RHEL Atomic Host version 7.6.4
Early build	January 8th 2019	February 19th 2019	April 2nd 2019
Final build	January 22nd 2019	March 4th 2019	April 16th 2019

(Source: own processing)

The quality of effectiveness and efficiency of the current process will be measured by through the application of a research questionnaire and by conducting some face to face interviews. It is anticipated to ascertain some basic and some more in-depth data so it possible for the iScorecard team to develop and roll-out appropriate performance measures to show evidence of improvements 'before' and 'after' changes to working practices were introduced. A statistical analysis will be carried out to enable the production of statistical and graphical performance presentations.

Creating Casual Links

Innovation goals presented in previous section are not separate and independent. On the contrary, they are connected and influence each other. The success of an innovation project is thus dependent on the collective action of many factors and is visualised by Strategy map (e.g. Cokins, 2009; Kaplan, Norton, 2001). It is a diagram that depicts how the innovation project creates value by connecting the partial goals with those of each stages of the Innovation Scorecard, i.e. how goals work together in an integrated, cause and effect sequence to build innovative culture, behaviour, processes and results.

Figure 9 illustrates an innovation strategy map for capacity planning project with its five stacked stages. Each rectangle represents an innovation goal plus their appropriate measures and targets. There are dependency linkages in an innovation strategy map with an upward direction of cumulating effects of contributions. The derived metrics are not in isolation, but rather have context in the innovation mission and vision. Innovation strategy maps and their derived scorecard are navigational tools that guide the company to execute the innovation strategy, but not to formulate the strategy. They are first and foremost a communication tool translating innovation strategy into vital goals necessary to execute the plan. With all the information contained on a single page, it is possible to visualise the cause-effect relationships described in Innovation Scorecard.

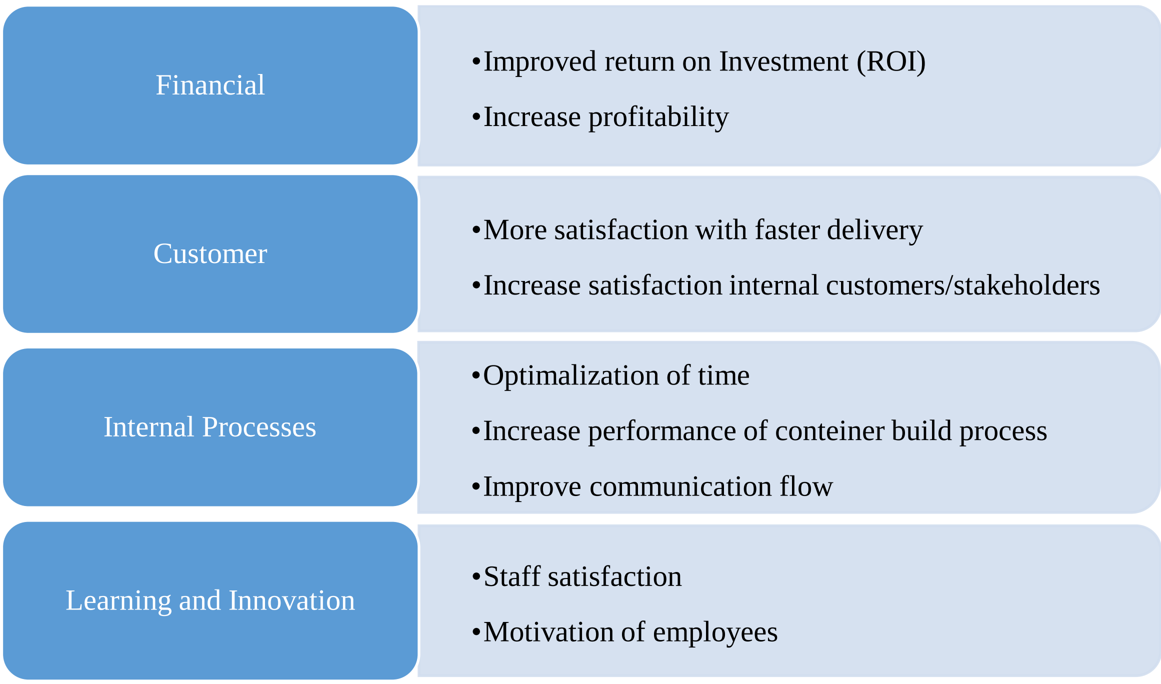

Figure 9: Strategy map for Atomic Host (Container Rebuild Process) innovation project

(Source: own processing)

4.1 Introduction and Implementation of Innovation Scorecard

The first step of an Innovation Scorecard System for Atomic Host is to introduce the Container Build Process. This means to determine the resource requirements of the project so that the project can be managed and completed successfully. Clear roles and responsibilities had to be determined to everyone in the team knew clearly and unambiguously what they were expected to do, when, how and with whom they had to interact. All team members within the Innovation Scorecard team were measured using the adopted definition of full-time equivalent (FTE). This is content of the project and at the same time it is the aim of the work of Research team. So, the personal capacity is determined. There is an Innovation Scorecard Leader of the whole project and at the same time the most important person of the team. The Innovation Scorecard Student of the Brno University of Technology and also the member of the team participates on the project and also she works on master thesis on the same title as this project. The Innovation Scorecard Professor which cooperates on the project from England and he is also indispensable member. Team Red Hat does not have express FTE specifically. Project Atomic Host and their cooperation on the project RHEL is part of their daily work which means they have to work on it anyway. Expression of FTE does not make sense.

Presentation to Red Hat stakeholders/associates/senior leadership on the changes and assurance that their teams know what to expect must be done. It is about the number of workshops on Innovation Scorecard framework and its implementation in Atomic Host in order to provide internal PR of innovation project. It secures support and awareness about innovation project within Red Hat. It is required to give at least 1 workshop for senior leadership and Container Team associates before innovation project launching. This workshop was arranged and, on the workshop, the iScorecard team met the team of Red Hat. Here it was presented what the project aims and objectives were, what we were working on right now, what the goals of the project were and what the customer expectations were. A follow up meeting was required to agree further activities in relation to the innovation team. Its content can be meeting with process of measurement and result of the project Atomic Host at the end. The part of this meeting can be the proposal, alternatively a discussion how to apply the Innovation Scorecard on other projects in RHEL which means QE, Extra Compose and Atomic Host Compose.

Number of meetings and calls on Innovation Scorecard in order to ensure transmission of information and progress report between Red Hat program manager, Container Team and iScorecard team is also too important to improve and maintain good and effective communications. It ensures clear responsibility and time commitment within the project as well as support and awareness about innovation project within Red Hat. Meetings of the Innovation Scorecard team were arranged every week at the beginning of the project. The essence of these meetings was too meet iScorecard team, Container Team and also programme manager. Other important thing was to understand the process in the company. This part was slightly demanding for iScorecard team because it has not so significant knowledges in IT industry as such extent as team Red Hat has. It was important to find out the way how to understand the process of the company at least on basic level which is needed to understand the whole process.

iScorecard team have periodically calls every week. Agenda is created the day before every call. The essence of this calls is reminder what is done and what has to be done next week. All of the information from the calls are recorded in the Project Action List. This document contains all work the iScorecard team has done including the date, owner of the tasks, detail and deadline (see Table 5). There is also describe what is in process and alternatively where is the problem if it is. In a fact that one of the project management skills is to monitor and control the project to find out it works on time there is a document which works on RAG status describe previously. The role of the RAG system is based on identifying and reporting against the project status. As explained previously, Green colour means that the project works to the plan and there is no difficulty or any trouble. Amber colour represents that something goes wrong and not to the plan and in the future the project may need an assistance. And the worst is the red colour which means that there is any problem and the project team need help to solve the problem.

Table 5: Tasks from the Action List

Date	RAG status	Action No.	Detail	Owner	Deadline
2.1.2019		1	Produce high level project documents – Project Definition Document, Issues and Risks Register, Change Control Process, Project Action List	Student	8.1.2019
9.1.2019		2	To see the senior manager and PGM to talk about the documents	Leader	13.1.2019
14.1.2019		3	Get in touch with expert on statistics, arrange consultation on questionnaire	Professor	18.1.2019
16.1.2019		4	Creation Questionnaire	Student	20.1.2019
23.1.2019		5	Creation Metrics	Leader	27.1.2019
27.1.2019		6	Feedback on Metrics and complete target values	Student	11.2.2019
1.2.2019		7	To arrange The Project Definition Workshop	Leader	16.2.2019
12.2.2019		8	To meet with Red Hat and asked them if they want a monthly Status Report	Student	19.2.2019
20.2.2019		9	Creation of monthly Status Report	Student	28.2.2019
1.3.2019		10	Creation history of the tool and describing of metrics	Student	20.3.2019

(Source: own processing)

It was the responsibility of the iScorecard team to arrange and manage all important project-related phone calls. They agree day and hour of the call. Member of the team send the invitation card by e-mail to other members the day before call including the agenda of the call. Communication by e-mail is daily routine as well as work on activities which are needed to another development of the project. Cooperation by Skype and WhatsApp is also part of team's communication tool.

Measure whether container rebuild process innovation will be performed in a timely manner. Milestones when core activities must be terminated are set as follows:

- Project documents in place by 21st January 2019;

- Container automation build process has to be implemented and be fully functional by end of June 2019.

The Project documents are Project Plan, Project Schedule, Project Definition Document, Issues and Risks register, Documentation register, Change Control Process, Project Action List, Scorecard Guide, Status Report, Communications and Reporting line Plan, Minutes of meetings. Measures number of produced, agreed and signed high level project documents in order to ensure smooth Innovation Scorecard introduction in container automation build process. Target value is to produce, agree and sign off all documents by 31st January 2019.

4.2 History of the tool

Container Team solved container development for a long time in the past. There was an effort to develop 10-12 new container images for RHEL 8 almost for two years ago. The Container Team was not sure how the maintenance of images will happen. The next thing which the Container Team solved was the fact what to do after Container development. It was appropriate technology which was using for a long time. The Container Team decided to improve an infrastructure which means to make easier the work of developers. There existed a specific version of the tool but the process for developer was still demanding because of too much e-mails, communication and work generally. They decided for implementation the tool in a practice. Meanwhile the team became the part of Cyborg team, they solved an automatization and an improvement of the process. Management of the company suggested a proposal for develop it for RHEL 8. Initiator of this idea was senior manager of the Container Team. For simplification the Container Team continued with their work, but they wanted to improve the work of developer and also to improve their current work.

At the beginning there were few requirements on bots (basic unit – container which do tasks which can be done by human on junior position) and the Container Team found out how to implement the units and how to connect them to the pipeline (at the end will be a product). One of the Bots secure mutual communication between bots and the team use the Celery for allocation tasks. Individual bot does various tasks and also can do a human work. They have human names because they can replace human work. Bots connect something which already exist. Despite the fact there was a tool for build image and also errata tool which is for deliver the product to the customer. They did not still have a tool which can connect them together. Bots were created for this reason.

4.2.1 Application of the tool in practice

RCM (Release Configurator Management) created Jira ticket in the past. It was used in container develop and also it included the information about base image and compose which were developed by RCM. These data were sent to PgM. His role was in reading the information and he just put them into the email and send to the owners (in this time 10 people). The conversation started. Image owners made build on the basis of the information and then they created administrative things, Bugzilla and errata. They put the build to the errata and move the errata to QE testing. QE gave information to the PgM if the testing was right and they sent the link on errata. The process was finished and the product was created. If there was some error, it was in build which was created by image owner or in compose and base image which were created by RCM team. All image owners must rebuild their images in this case and whole communication was again by emails. They often must to check where exactly is the problem. If in their work or in input from the RCM team. When the problem was the number of communications multiplied with the number of owners (it is 10 on 14 image).

Tool is for automation of build process and it was first applicated on the version RHEL Atomic Host 7.6.2. This version was launched on the beginning of 2019. The aim of the tool is to minimize and whole manual work on build. Jira ticket was used in this version and probably it will be used in the next version too. His essence is for RCM because they work on these tickets and they recorded work and process via them. Ticket is for many people and for many tasks. It was used in creation of image and compose by RCM in the past. And also PgM used it for control. Image owners neither automation team do not need them.

4.3 Creating Innovation Scorecard Data Sheet

Once managers have set goals and created casual links, the next step is to catalogue specific characteristics of each in an Innovation Scorecard Data Sheet – a document that provides all users with a detailed examination of Innovation Scorecard measures, including a thorough list of characteristics. Creating the measure-data sheet is not necessarily a fun or glamorous task, but it is an important one, because it provides the background for measure choices.

Figure 9 shows the template – Innovation Scorecard Data Sheet – which seeks to specify what a "good" innovation performance measure constitutes. The framework ensures that the measures are clearly defined and based on an explicitly defined formula and source of data. Based on Niven's (2014) work, there are four sections of the template that must be completed. 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.

Measure number/name: …………………………………………	Phase: …………………………	Owner: …………………………
Goal: ……		
Definition: ……		
Lag/Lead: …………………………………	Unit type: …………………………………	Frequency: …………………………………
Formula: ……		
Data source: ……		
Data quality: …………………………………………………	Data collector: …………………………………………………	
Baseline: …………………………………………………	Target: …………………………………………………	

Figure 10: Innovation Scorecard Data Sheet

(Source: own processing)

Measurement background

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 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.

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

Goal: Every measure was created as a translation of a specific objective, including the identification of relevant goals. 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 is the intent of this 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.

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, time (FTE, hours), currency (USD, CZK, etc.), and percentages.

Calculation and data specifications

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

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 confirmation 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.

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 of for an annual number, but track performance toward that end on a monthly or quarterly basis.

4.4 Selecting Innovation Metrics

Following on from the Innovation Scorecard background, the next step is innovation indicators selection. Metrics presented in Table 7 are catalogue according to an Innovation Scorecard Data Sheet template presented in Figure 9. iScorecard Leader and iScorecard Student are established as data collectors for all metrics in Container Rebuild Process innovation project.

As the innovation project is in the area of software development (including short sprints), the team considers that the application of a 'gate process' to manage this each small project is not justified as it does not add any value to the overall process.

Inputs lead to outputs. Processes are used to get from input to output. The purpose of the innovation scorecard initiative is not to improve Red Hat's current quality management system. This system includes all processes. The goal/purpose is to improve the management of inputs and outputs through innovative approaches to improve areas such as job satisfaction, staff motivation and morale, customer perception and associated outcomes such as improving productivity, reduce business overheads and generate new/repeat business for Red Hat.

Kaplan's "twenty is plenty" suggests that twenty metrics is sufficient for any project. It has to be justified so suggest to only use Kaplan's suggestions as a guideline.

His work was done in the 1980s/90s and can only act as a guideline as the business world has changed, in particular areas such as IT and Software Development. We can justify using as many metrics as we think appropriate to ensure that Red Hat receive value for money and return on their initial investment in the form of innovative ways of working, reducing unnecessary processes and creating a new confidence in their customer base that Red Hat are delivering industry best practice solutions to customers desired or required business needs.

It is not the number of metrics we use, what really matters is the quality of these metrics and their overall contribution to the success of introducing an innovation scorecard design. It is not important 'how many' but what is the quality of the contribution of each metric.

Table 6: Innovation Metrics

	Metric number/name	Target
Inputs	Working activities structure	Reduce manual repetitive work
	Blocked time	Minimize
Process	Number/weight of errors during implementation	Minimize
Outputs	Number of requests for automation tool changes	Max. 1 radical/10 incremental
	Number of due (priority) activities	Minimize
Outcomes	Job Satisfaction	Increase
	Saved resources	Maximum of Time

(Source: own processing)

The following structure of each metric description includes the name of the metric, characteristics of each metric or information about when and under what conditions it is measured. Tables contain measured values from individual versions or values before and after automation. As far as possible there are comments on the results of each version or the results before and after the introduction of the tool.

4.4.1 Working activities structure

Investigates typical working activities structure of Container Team before and after automation tool implementation. The purpose is to present the impact of automation on the structure where repetitive manual work should be eliminated in favour of other (priority/non-priority) tasks. This qualitative data is going to be gained by questionnaire and structured interviews held before and after innovation project. Target values are set by trends (=to reduce manual repetitive work in favour of other priority tasks) because of the Agile access Red Hat uses in the area of development and in the company world-wide.

Table 7: Working activities structure (range in %)

Activities	CTM 1		CTM 2		CTM 3		CTM 4		CTM 5		CTM 6		CTM 7	
	Before	**After**	**Before**	**After**	**Before**	**After**	**Before**	**After**	**Before**	**After**	**Before**	**After**	**Before**	**After**
Activity 1	80 – 60	20 – 0	100-80	20 – 0	40 – 20	20 – 0	20 – 0	20 – 0	80 – 60	20 – 0	60 – 40	20 – 0	20 – 0	20 – 0
Activity 2	40 – 20	20 – 0	40 – 20	100-80	20 – 0	20 – 0	40 – 20	20 – 0	60 – 40	20 – 0	40 – 20	20 – 0	40 – 20	20 – 0
Activity 3	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	40 – 20	20 – 0
Activity 4	20 – 0	20 – 0	40 – 20	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	40 – 20	20 – 0
Activity 5	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	80 – 60	60 – 40	20 – 0	20 – 0	20 – 0	20 – 0
Activity 6	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	60 – 40	40 – 20	20 – 0	20 – 0	20 – 0	20 – 0
Activity 7	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0	20 – 0

(Note: CTM – Container Team Member) (Source: own processing)

The individual activities in table 7 specifically include these activities:

Activity 1 – Developing new features in the automation system

Activity 2 – Resolving issues in the automation

Activity 3 – Performing manual steps in case of issues

Activity 4 – Communicating with programme management

Activity 5 – Making sure that the builds are done on time

Activity 6 – Building container images using the latest content available

Activity 7 – Making sure the latest build is ready to be shipped (advisory has the build attached and is in state ON_QE

It was found out, based on the outcome of a questionnaire, how much time Container Team members spend on individual activities. It was interesting to investigate what their working hours/structure was before and after the implementation of the automation tool. The assumption was a range in % based on 8 hours working day/40 hours week from Monday to Friday. The idea is to show the impact of automation on the structure where repetitive manual work should be eliminated by the implementation of the tool in practice.

Before the automation all team members had at least one activity in range in % bigger than 20 – 0. The purpose of the tool was to minimize these activities and get them to work automatically. For activities 1, 3, 4 and 7, all members of the team were reduced to a range of 20 – 0 after automation of the tool. For the rest activities, the number 2, 5 and 6 was reduced by 6 members of 7.

According to the collected and analysed results, it was found that a total of 5 people of 7 experienced a total reduction in manual repetitive work in favour of other priority tasks. This means that people can spend the time they have saved on other activities thus improving productivity within their work areas. It will also enable them to focus their attention on new projects or other work activities. In addition, with the extra time now being available for other activities, it is also possible to help and support other colleagues with the work they need to deliver.

The impact of automation through the implementation of a tool in practice on the work of individual Container Team members was that it eliminated manual labour, which was one of the main goals of this implementation. It is possible to state that the implementation of a innovation scorecard was successful and effective, fulfilled the goal for which it was designed and met expectations.

4.4.2 Blocked Time (Delays)

Measures blocked time by waiting for another team member work/reaction. This metric is going to be measured before innovation project and after each cycle. Target value is set up to 0 hours after 3 cycles.

Table 8: Blocked Time Delays

Before	RHEL 7.6.2	RHEL 7.6.3	RHEL 7.6.4
8 - 10	8	0	0

(Source: own processing)

On development software work in cooperation lots of people from the whole world. Because of this fact in past it was lots of time delay (8 – 10 hours) in waiting on somebody else's work. Time zone is the problem in communication because they are different in America, Europe and Asia. In the past it was sent the e-mail to 40 people. Moreover, there were communication needed to solve issues. For example, they communicated via IRC to image owners about requests to repair configurations in images. Team had to must communicate with RCM because they received incorrect input data. They sent regular status update to their PgM too.

RHEL 7.6.2

This problem was solved by applying the innovation tool. It saved lots of time in communication which meant that they were able to get on with the work. This improved time delays when they were waiting for responses from others, leading to reductions in wasted time such as waiting for answers to issues from others. A good example how the innovation approach improved wasting time was when team members were waiting almost 8 hours for information relating to open vm container issues during January 2019 (version 7.6.2). It had to be done manually at the end and it was done after 10 minutes finally. Time of container owners (10 people) was saved because of installation the automation. Their work is done by two members of automation team for this time. Aim of the process is full automation or just work of release team which assign the beginning of the process. Next aim is the vision of one summary e-mail at the end of the whole process which automation team send.

RHEL 7.6.3

This time delay is zero in version 7.6.3. Communication time lasts for 10 minutes because of the one e-mail which is sent to 10 people after successful process. It related only to the process. This work do two members from automation team and all of the process is fully automatic. There is no possibility to wait to another people because of the automatic process of tool. This was the main reason for implementing the tool in practice. It can be noted that the delay time achieved the required value 0 already in this version.

RHEL 7.6.4

As well as in the previous version there was no delay time. This fact was caused by the application of automation in practice and it was assumed. This was also the main goal that tool was fully automated, and the delay time was minimized or completely canceled. Given that the delay time was zero in this and the previous version it can be observed that the application of the tool in practice achieved the desired goal that the Container Team had set and expected it to do.

4.4.3 Number/Weighting of errors during implementation

Measures number and importance of failures within automation tool implementation. This metric is going to be measured after each cycle on implementation by using following scale to evaluate importance of instances:

- Critical – automatic tool is stopped, and Container Build Process has to be finalized manually;
- Medium – automatic tool is stopped but the Container Team is able to reactivate the process again;
- Low – some bugs which can be fixed immediately, and the process continues.

Table 9: Number/weigh of errors during implementation

	RHEL 7.6.2		RHEL 7.6.3		RHEL 7.6.4	
	Baseline	**Target**	**Baseline**	**Target**	**Baseline**	**Target**
Critical	2	0	0	0	1	0
Medium	7	5	1	5	3	0
Low	4	8	1	8	1	8

(Source: own processing)

RHEL 7.6.2.

First application of tool had few errors. Primarily there were two critical errors which represented total collapse of tool. It was stopped because of these errors and it had to be done manually again. First problem was the fact that some images needed to add custom compose for their image build, and they had no support for that. It had to be done initial task pipeline manually. Other images were not influenced. And the second one was incomplete support for pushing to CDN repos. Whole pipeline is composed from few tasks which were made and the last cannot be done because they did not have the information needed to completion (missing support for pushing to CDN repos). They did not expect that and they asked all to finish it manually and save it to configuration file.

First application included few medium errors:

1. Missing authority (responsible person) for filling input information about product, version, release, and components.
2. There was an issue in a library the automation system is using which prevented us from creating new errata.
3. Failed to add bugs to errata - variable typing error in the code.
4. Bugzilla 'fixed in version' not updated - NVR comparisons needed to be improved.
5. Our automation system did not have permissions to change errata which were in ON_QA state.
6. In final build errata that were updated (not created) failed to be switched to QE.
7. Missing support for updating errata that are in state RHEL_PREP.

Image owners had to regulate configurations to successful launching automatization. During this Container Team found out that half owners made a mistake and they had to tell them to correct it. After this action the team was able to continue with the automatization. Low error was the fact that the automation was unintentionally lowered many times because of using OpenShift job instead of pod. Next error was a description of bug in Bugzilla which included irrelevant information and also name of components occurred errors in errata summary. The last low error was missing template for sending e-mails.

The targets were determined in this version as 0 critical error, 5 medium errors and finally 8 low errors. This version fulfilled these targets only in low errors. It was first application of the tool so it was expected that there will be critical and medium errors in bigger size.

RHEL 7.6.3

The version 7.6.3. was the second application of the tool. There were three problems in this version but just one task related to the automation tool. One medium and one low error were produced by wrong configuration of input data (specifically unexpected and unwanted changes in Bugzilla in three images). Medium issue in automation related to failure of change one image build in errata and its subsequent switch into QE. They solved all issues inside the team and it was not needed to contact image owners. All in this version was expected and all of the issues were small.

There was not a critical error in this version which is a good signal of the right implementation of the tool. The targets for this version were determined as well as in the version before. This version fulfilled these all targets.

RHEL 7.6.4

In this version Container Team implemented a new mechanism in some experimental images. Until now, to run the pipeline, there was a need to someone in the container Team to manually launch a boot that was responsible for building the build for each image called as Solenya-batch (which requires a few small changes in one file and uploading it into the infrastructure). In the future, they have the ability to remove this step and rely on automation to activate rebuild, which has already existed in Red Hat infrastructure and it is used (known as OSBS OpenShift Build Servise auto-rebuild. Some time ago, the OSBS came to a state where it could be used for a Container Team, and 7.6.4 was in the release section where they tried for the first time the new OSBS functionality. However, they encountered a problem that blocked them for two to three days. More time is needed to permanently fix the problem, so it returns to the original – manually triggering Solenya-batch.

This version had one critical error and one medium error which related to new mechanism. Critical was when new functionality (OSBS) made activation some builds impossible and medium was when it changed the way to get data for build, also one image contained an unexpected package – corrected on the maintainer’s side of the team that specified the version he expected.

Other two medium errors were that three build activities were probably not going to be run because of a lack of computing resources. And it was also necessary to ask RCM to modify the input data to run the build. There was one low error when Solenya-batch encountered an unsuccessful build it did not record all results.

Targets of this version were determined as 0 critical, 0 medium and 8 low errors. Because of the new mechanism which Container Team implemented in this version, there was one critical error and one medium error which related to this implementation of the new mechanism. Other errors were related to the automation tool.

4.4.4 Number of requests for automation tool changes

The metrics serves to evaluate the implemented automation tool towards current state. It is going to be measured after cycle. Target values is set up to none requests for any changes in automation tool delivered to innovation team. In other words, automation tool follows all requests from all stakeholders which were delivered to innovation team.

Table 10: Number of requests for automation tool changes

	RHEL 7.6.2		RHEL 7.6.3		RHEL 7.6.4	
	Baseline	**Target**	**Baseline**	**Target**	**Baseline**	**Target**
Radical	2	1	1	1	1	1
Incremental	2	10	0	10	0	10

(Source: own processing)

RHEL 7.6.2.

There were two proposals on changes in version 7.6.2 after first activation. These proposals were permitted, but after few images they need to do another things. These changes significantly changed used tool and they are considered as radical. Specifically:

1. To add own compose,
2. support of additional product.

Next two changes were incremental.

1. To add another thing to the Bugzilla and it would be great if the tool can do that,
2. change the component ‘s name.

These proposals have been made because they have been identified as appropriate and could also help to improve the next applied version in practice.

RHEL 7.6.3

For version 7.6.3 was the proposal for more detailed and united reporting. It can be considered like radical change. The point is greater emphasis on team awareness of the actual functioning of the tool in practice as well as its achievements in the event of full system automation. This proposal has been adopted and implemented because sufficient team awareness is one of the cornerstones for the future successful operation of the tool. On the other hand, there is no incremental request.

RHEL 7.6.4

In this version there was one proposal for change, and it can be marked as critical. Until now the pipeline had to be run manually which was responsible for building the builds for each image. Container Team wanted to change this and so there was a suggestion to remove this step and rely on running rebuilds that already exist in Red Hat infrastructure. Specifically, it was OSBS (OpenShift Build Service) auto-rebuild. As well as in the previous version there is no incremental request.

4.4.5 Number of due (priority) activities

Measures progress in working activities structure, resp. impact of saved time on due (priority/non-priority) activities compared to current state. The metric is going to be measured before innovation project launching and after automation tool implementation. Decreasing trend is set as target value for particular stakeholders.

Table 11: Number of due (priority) activities

	Before	After
Container Team member 1	5	0
Container Team member 2	2	0
Container Team member 3	4	1
Container Team member 4	3	1
Container Team member 5	1	0
Container Team member 6	1	0
Container Team member 7	2	0

(Source: own processing)

Based on the application of the tool, it is possible to measure the impact on the saved time on the priority tasks that the team members had in the job description and also the non-priority tasks they had no time for. It is therefore a measure of progress in working activities structure.

Thanks to the application the time of all members of the Container Team was saved. Some have used this saved time for a new project and others have focused on leadership or project management. A total of 5 members have reached the final value of 0 which means that they did not negligent of any non-priority tasks after automation. The purpose of automation was to reduce the manual work of the individual members and automate it so that their time fund saved could be used elsewhere.

4.4.6 Job Satisfaction

Measures satisfaction with their working activities. Data is going to be gained by qualitative questions within structured interviews and by questionnaire. Satisfaction is measured by 10-point Likert scale before innovation project and when automation tool is fully implemented in container rebuild process. Number 1 is very dissatisfied and number 10 is very satisfied.

Table 12: Job Satisfaction

	Before	After
Container Team member 1	8	8
Container Team member 2	9	8
Container Team member 3	8	10
Container Team member 4	7	10
Container Team member 5	6	7
Container Team member 6	8	10
Container Team member 7	9	10

(Source: own processing)

Satisfaction with their working activities was evaluated on the basis of a questionnaire that was conducted with individual participants, respective members of the Container Team. The evaluation is measured by 10-point Likert scale. According to the results almost all members were fully satisfied with their working activities before the implementation of the tool. And even though all this was done manually in practice.

Satisfaction was evaluated by questionnaire again when automation tool is fully implemented in container rebuild process. The introduction of tool automation had high expectations. According to the results, more than half of the participants are fully satisfied with the functionality and therefore with the fact that this work is now done automatically, which was the purpose of the implementation of the tool application in practice.

According to the information which were found out on the face to face interview, the tool application had a high impact on the fact that they do less to do things now and they have more time to work on features. Of course, there is always something for improve and even in this case, automation can be a little more improve and secured so that the process is completely error free. However, the Container Team agreed that this might work for the time being, because the main goal was to make the tool replace manual work as automatic and that was accomplished.

4.4.7 Saved resources

Improving the efficiency of overall container rebuild process (improvements in effective communications and build process duration), its comparison with initiated state allows us to calculate saved time. This final information is crucial for negotiating with senior leadership about upcoming innovation (automation) project(s) in Atomic Host.

- **Improvements in Effective Communications** is measured by saved time, which is achieved by automation of reporting system.
- **Build process duration** measures time needed to rebuild all containers in each cycle. The aim is to reduce time needed to build 1 container. Then number of containers in each cycle is multiplied by the time and build process duration se calculated.

Build process duration = time per 1 image * number of images

Assumptions: ca 14 images at the beginning, then increasing in the future

Saved resources (time) = saved time in communication flow + saved time in build process duration

Table 13: Saved resources.

	Before	After
Build process duration	45 * 14 = 630 min (10,5 h)	45 minutes
Communication	15 * 14 = 210 min (3,5 h)	15 minutes
Total	840 minutes = 14 hours	60 minutes = 1 hour
Saved resources (time)	**13 hours**	

(Source: own processing)

The result is based on the assumption that there are 14 images that can be increased in the future. If they did not consider any delay or problems, then build process duration of 1 image before automation lasted for 45 minutes. Communication lasted for 15 minutes for each build. Total time was 60 minutes, which was 1 hour to build 1 image. After multiplying this time by 14 images, it was total of 14 hours. In the case that there were any problems, or another team member was waiting for the time shift, this time could be increased. According to the information from Container Team this time was often 3 or 4 days.

After the automation, the build of all 14 images lasts for 45 minutes because it is done in parallel. This was one of the main reasons for implementation of the tool in practice. Thus, the communication involving all these containers is for 15 minutes. Total time is 60 minutes, which is 1 hour to build all 14 images including communication. It was possible that more time could be needed for the application of the new tool if there were any issues with the tool. Provided there were no issues, then delays would become a thing of the past as everything was going to be done automatically. Comparing the before and after automation "saved time" results, it is possible to see that 13 hours of time were saved by applying the tool in practice. It can be stated that the build has saved a total of 585 minutes (9,75 hours) and within the communication it is 195 minutes (3,25 hours). This saved time after recalculation corresponds to a total of one day and a half which was saved by simply putting the application into practice. Based on the information from the face to face interview it was found that thanks to the fully automation tool the communication improved and also the time that they had to spend in general was improved.

4.4.8 Statistical survey

The findings of the application of tool in practice can be evaluated also on the basis of statistical research. The point is that the results can be proved by a different method than by simply comparing the data before and after. In the case of statistical research this is a technique based on calculations. Two metrics were randomly selected for the demonstration, first metric is Job Satisfaction which will be evaluated based on the t-test. The second metric is Working Activities Structure which will be evaluated based on McNemar's test.

The paired t-test is used to compare the mean values of two populations, comparing observation samples, for example, before and after on the same object (selection). The McNemar's test is designed to be used in conjunction with a PivotTable in the case of a paired experiment to monitor the occurrence of a random variable on the same sample. It is therefore similar to a paired t-test. Based on this test it is evaluated whether the experiment differs between the two repetitions of the probability of occurrence of individual variants of a random variable.

The main goal for selecting a test procedure is to make the mistake as small as possible. The Statistics software program was used to calculate these tests.

For testing it is necessary to formulate a null and alternative hypothesis and also to determine the level of significance. The null hypothesis claims that the experimental intervention as not effective. This can also be interpreted as the fact that the values did not shift before and after the change. However, this hypothesis can be disproved. If the test proves that the null hypothesis is invalid, then it is inclined to an alternative hypothesis. This hypothesis claims that the intervention was effective and there was a shift in value. Depending on the nature of the experiment, it is possible to determine the direction of the shift (increase/decrease).

Subsequently a level of significance is selected which was selected at 10 % (0,1). This above represents the probability of committing a type 1 error. There is an attempt to not make this mistake. Based on the calculations in the program, the p value comes out and is compared with the significance level. If p value is ≥ the significance level, the null hypothesis is confirmed. If p value is < significance level, an alternative hypothesis is confirmed. There is a huge risk that the null hypothesis is rejected, although it is correct. It is very important to be careful about the results of these tests.

Job Satisfaction

The Job Satisfaction metric was subjected to a paired t-test and was calculated using Statistics software. A null and an alternative hypothesis was established, and the level of significance was set at 10 %. Reliability is 95 %. It was following calculation, which was calculated by the program:

Table 14: Calculation of paired T-test in Statistics software

	t-test pro závislé vzorky (Tabulka1) Označ. rozdíly jsou významné na hlad. p < ,10000									
Proměnná	**Průměr**	Sm.odch.	N	Rozdíl	Sm.odch. rozdílu	t	sv	p	Int. spolehl. -95,000%	Int. spolehl. +95,000%
Prom1	7,857143	1,069045								
Prom2	9,000000	1,290994	7	-1,14286	1,345185	-2,24781	6	0,065639	-2,38695	0,101232

(Source: own processing)

According to the results of the Statistics software in which the Job Satisfaction values were compared before automation and after the implementation of the tool in practice, a p value 0,065639 was issued. This value was compared with the significance level which is 0,1. This comparison of values can be written as 0,065639 < 0,1. Thus it can be stated that since the p value is less than the selected 10% significance level, the alternative hypothesis is confirmed and the null hypothesis is rejected. The fact that the test is significant also indicates the red colour of the font in the table.

Working Activities Structure

The Working Activities Structure metric was subjected to McNemar's test. A null and an alternative hypothesis was established, and the level of significance was set at 10 %. Reliability was 95 %. This test was calculated based on the Pivot Table (Frequency Table) and related relative frequencies. It was calculated for each activity and this will continue into future planned activities.

Table 15: McNemar's test for activity 1

	Pivot Table				Frequency Table				Sign. level	P value
Act.		**B1**	**B2**	**Total**		**B1**	**B2**	**Total**		
1	**A1**	3	0	**3**	**A1**	0,429	0,000	**0,429**		
	A2	4	0	**4**	**A2**	0,571	0,000	**0,571**		
	Total	**7**	**0**	**7**	**Total**	**1,000**	**0,000**	**1,000**		
Total									**4,000**	**3,841**

(Source: own processing)

According to the statistical research calculation for activity 1 (Developing new features in the automation system) the significance level was 4,000 and p value was 3,841. The significance level in this case is > p value so it can be stated that the alternative hypothesis is confirmed. On the basis of this test it was confirmed that there is a significant difference and thus the application of tool in practice had an impact on the activity 1 in working activities structure.

Table 16: McNemar's test for activity 2, 5, 6

	Pivot Table				Frequency Table				Sign. level	P value
Act.		**B1**	**B2**	**Total**		**B1**	**B2**	**Total**		
2, 5, 6	**A1**	6	0	**6**	**A1**	0,857	0,000	**0,857**		
	A2	1	0	**1**	**A2**	0,143	0,000	**0,143**		
	Total	**7**	**0**	**7**	**Total**	**1,000**	**0,000**	**1,000**		
Total									**1,000**	**3,841**

(Source: own processing)

According to the statistical research calculation for activity 2, 5 and 6 (Activity 2 – Resolving issues in the automation, Activity 5 – Making sure that the builds are done on time, Activity 6 – Building container images using the latest content available) the significance level was 1,000 and p value was 3,841. The significance level in this case is < p value so it can be stated that the null hypothesis is confirmed. In this case the application of tool in practice did not have such an impact or the change was not so significant.

Table 17: McNemar's test for activity 3, 4, 7

	Pivot Table				Frequency Table				Sign. level	P value
Act.		**B1**	**B2**	**Total**		**B1**	**B2**	**Total**		
3, 4, 7	**A1**	7	0	**7**	**A1**	1,000	0,000	**1,000**		
	A2	0	0	**0**	**A2**	0,000	0,000	**0,000**		
	Total	**7**	**0**	**7**	**Total**	**1,000**	**0,000**	**1,000**		
Total									**xxx**	**3,841**

(Source: own processing)

According to the statistical research calculation for activity 3, 4 and 7 (Activity 3 – Performing manual steps in case of issues, Activity 4 – Communicating with program management, Activity 7 – Making sure the latest build is ready to be shipped (advisory has the build attached and is in state ON_QE) the significance level cannot be expressed. This calculation is impossible. If the value is only in one line, the resulting effect cannot be calculated.

CONCLUSION

The Innovation Scorecard was designed, implemented, has been verified that it works in practice and its viability and reliability have been established. Innovation Scorecard was implemented in the area of process innovation within one Red Hat project known as "Atomic Host". The partially considered objectives for this master thesis were to understand work processes and also how Red Hat operates in general. It was necessary to research the theory of what was already known about the subject matter under investigation and to modify the Innovation Scorecard system and approach to make it fit for the intended purpose and use within an Agile Software Development work environment. This included modifications relating to the existing Agile Methodology in operation within Red Hat in addition to further adjustments as far as the innovation approach was concerned. This approach ensured that the concept of an innovation scorecard was aligned with the day to day operations of Red Hat's Agile Methodology. It was thus possible to implement the proposal for the introduction of an Innovation Scorecard system into the live working environment of Red Hat without disrupting the "business as usual" work activities. Early feedback from the Atomic Host team indicated that the implementation of the Innovation Scorecard system yielded some positive results and that no further modifications were considered necessary to improve it.

One of the Atomic Host teams, known as the Container Team, benefited from the implementation of the Innovation Scorecard. Performance reviews were conducted and the outcomes, based on applying some measurement metrics, suggest that the modus operandi within this team improved significantly. Manual operations were integrated into existing fully automated working practices with the result that duplications of effort were eliminated and or at least improved upon. This resulted in reduced efforts required by the team to carry out certain activities. In turn, this made it possible to team members to be freed up to undertake other or additional tasks, adding further value to the introduction of the innovation scorecard within Red Hat. It resulted in significant time savings in terms of efforts employed and it was possible for the team to re-deploy these resources elsewhere in the business.

For example, previously mentioned activities 1, 2, 4 and 7 now benefited from a reduction in team members after the tool was automated. This was one of the main goals of the tool application in practice and results suggest that this goal was achieved. For example, a set time delay metric was employed to assess how much time could be saved through the application of the Innovation Scorecard concept. The practical application of this tool met customer expectations leading to a total cancellation of any time delay. A by-product of this experience was the creation of new user perspectives in areas such being creative and innovative. It appeared that Red Hat staff started to talk to each other more. This created new opportunities for working together much more closely and better understand how to build better innovative products for the benefit of Red Hat's customer base.

It can be concluded that the Innovation Scorecard tool has made a major contribution towards reducing the number of errors experienced in some areas of software development such as the Atomic Host. Error reduction target values were achieved over and above any set limit. When new mechanisms were introduced by the Container Team, outside the Innovation Scorecard scope, it became obvious that error values increased. This is a normal and expected behaviour and this provides further evidence in support of an Innovation Scorecard system to combat these issues. It is suggested that an Innovation Scorecard approach adds value to the modus operandi of Container Teams in software development areas.

In the case of requirements, they were those that would in the next version facilitate or improve the further functioning and effectiveness of the tool itself. These were operational changes, but they were successful because the automation of the system went well and the entire process is now being performed automatically, which was the goal of implementing automation info practice.

Due priority activities were gradually reduced as a result of saved time. The aim of the tool application in practice was to reduce the manual work of individual members. This goal was achieved, and the time individual members saved can be used on other projects or any related work areas. Another priority metric was Job Satisfaction. It appeared that the Container Team were relatively satisfied with the work they were doing before any changes were introduced. After the introduction of the Innovation Scorecard tool, greater Job Satisfaction levels were identified amongst staff following the outcomes of some research that included team members. Being allowed to be creative and to become innovative leads to higher levels of Job Satisfaction.

The build process was a manual operation that greatly benefited from the Innovation Scorecard process introduction. All work until then was conducted manually. Fully automating this process led to significant reduction in efforts needed by the Container Team opening up new opportunities to reassign resources to other areas of the business without affecting the Container Build work area.

The Innovation Scorecard tool has made a significant contribution to improve the efficient and effective operations within Software Development in IT companies such as Red Hat. The outcomes from this research suggest teams such as Container Teams would benefit from a wider roll out of this system. Economies of scale can be achieved in terms of reducing manual efforts and then re-allocating saved resources to support other business areas. It is confirmed that the concept of Innovation Scorecard works both in theory and practice. This will be of great value and benefit to both the community of practice and academia. It is hoped that this research has brought theory and practice much closer together.

REFERENCES

AFUAH, A., 2014. *Business Model Innovation: Concepts, Analysis and Cases.* New York: Routledge. ISBN 978-0-415-81739-4.

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

BREMSER, W.G., BARSKY, N.P., 2004. Utilizing the Balanced Scorecard for R&D Performance Measurement. *R&D Management*, vol. 34, no. 3. ISSN 1467-9310.

COKINS, G., 2009. *Performance Management: Integrating Strategy Execution, Methodologies, Risk, and Analytics.* Hoboken, N.J.: Wiley. 240 p. ISBN 978-0-470-44998-1.

COOPER, R.G., 2008. Perspective: The Stage-Gate Idea-to-Launch Process-update, What's new and NextGenSystems. *Journal of Product Innovation Management*, vol. 25, pp. 213-232. ISSN 1540-5885.

COOPER, R.G., EDGETT, S. and KLEINSCHMIDT, E., 2004. Benchmarking best NPD practices. *Research Technology Management*, vol. 47, no. 1. ISSN 0895-6308.

CZECH STATISTICAL OFFICE, 2008. *Příklady inovací pro TI 2008* [On-line]. [2018-11-14]. Retrieved from: http://www.czso.cz/csu/redakce.nsf/i/prakticky_seznam_prikladu_inovaci_verze_2008/$File/priklady_inovaci_2008.pdf

DAVILA, T., EPSTEIN, M.J. and SHELTON, R.D., 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.

DONELLY, G., 2000. A P&L for R&D. *CFO Magazine*, February, pp. 44-50.

FAHRER, N., 2012. *Innovation & Other Useless Things.* Norman Fahrer. 108 p. ISBN 978-1-938578-01-4.

GRUBLER, A., WILSON, CH., 2014. *Energy technology innovation: Learning from Historical Successes and Failures.* Cambridge University Press. ISBN 978-1-107-02322-2.

HENDL, J., 2012. *Přehled statistických metod: analýza a metaanalýza dat.* Prague: Portál. 736 p. ISBN 978-80-262-0200-4.

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

CHIESA, V., FRATTINI, F., 2009. *Evaluation and Performance Measurement of Research and Development.* Cheltenham: Edward Elgar Publishing. 301 p. ISBN 978-1-84720-948-1.

ITTNER, C.D., LARCKER, D.F., 2003. Coming up short on Nonfinancial Performance Measurement. *Harvard Business Review*, vol. 81, no. 11, pp. 88-95. ISSN 0017-8012.

KAPLAN, R.S., NORTON, P.D., 2001. *The Strategy focused Organization.* Boston, MA: Harvard Business School Press. 416 p. ISBN 978-57851-250-8.

KEEGAN, D.P., EILER, R.G. and JONES, C.R., 1989. Are your Performance measures obsolete? *Management Accounting*, vol. 70, no. 12., pp. 45-50. ISSN 0025-1682.

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

KISLINGEROVÁ, E., 2008. *Inovace nástrojů ekonomiky a managementu organizací.* Prague: C.H.Beck. 293 p. ISBN 978-80-7179-882-8.

KUNZ, LEIGH, ASSOCIATES, 2018. *Our approach.* [On-line]. [2019-02-07]. Retrieved from: https://kunzleigh.com/about/our-approach/

LEHNER, S., 2016. *Performance Measurement of an Internal Service Team: Design and Implementation of a Balanced Scorecard.* GRIN Verlag. 71 p. ISBN 978-3-668-12170-6.

MARR, B., 2012. *Key Performance Indicators (KPI): The 75 measures every Manager needs to know*. Pearson UK. ISBN 978-0-273-75038-3.

MATHIS, B., 2013. *Agile Project Management for Beginners: Mastering the Basics with Scrum.* CreateSpace Independent Publishing Platform. ISBN 978-1-500-89264-7.

MILLETT, S., BLANKENSHIP, J. and BUSSA, M., 2011. *Pro Agile.NET Development with SCRUM.* Apress. ISBN 978-1-430-23534-7.

NEELY, A., MILLS, J., PLATTS, K., GREGORY, M. and RICHARDS, H., 1996. Performance Measurement System Design: Should Process based Approach be adopted. International Journal of Production Economics, vol. 46-47, no. 1, pp. 423-431. ISSN 0925-5273.

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

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 Paris: OECD Publishing. 258 p. ISBN 978-92-64-30460-4.

OSTERWALDER, A., PIGNEUR, Y., 2012. *Tvorba business modelů: příručka pro vizonáře, inovátory a všechny, co se nebojí výzev.* V Brně: BizBooks. ISBN 978-80-265-0025-4.

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.

POVILAITIS, S. 2014. *Acceptance Criteria.* [On-line]. [2018-11-12]. Retrieved from: https://www.leadingagile.com/2014/09/acceptance-criteria/

Red Hat. 2019 [On-line]. [2019-03-12]. Retrieved from: https://www.redhat.com/en

SLIGER, M. *Agile Project Management with Scrum*. [On-line]. 2011 [2018-11-12]. Retrieved from: https://www.pmi.org/learning/library/agile-project-management-scrum-6269

SYLVER, B. *What does "Innovation" really mean? Businessweek. News and Features* [On-line]. 2006 [2018-09-28]. Retrieved from: http://www.businessweek.com/innovate/content/jan2006/id20060131_916627.htm.

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

THOMASCHEWSKI, D., TARLATT, A., 2010. Determinants for Failure and Success in Innovation Management. In A. GERYBADZE et al., eds. *Innovation and*

International Corporate Growth. Berlin: Springer, pp. 127-149. ISBN 978-3-642-10822-8.

TIDD, J., BESSANT, J. R. and PAVITT, K., 2005. *Managing Innovation: Integrating Technological, Market and Organizational Change.* Chichester: John Wiley. 582 p. ISBN 0-470-09326-9.

TRIAS DE BES, F., KOTLER, P., 2011. *Winning at Innovation: The A-to-F Model.* Palgrave Macmillan. ISBN 978-0-230-34343-6.

VALENTA, F., 1969. *Tvůrčí aktivita – inovace – efekty*. Prague: Svoboda. 258 p.

WERNER, B.M., SOUDER, W.E., 1997. Measuring R&D Performance: State of the Art. *Research Technology Management*, vol. 40, no. 2, pp. 34-42. ISSN 0895-6308.

ZALTMAN, G., DUNCAN, R. and HOLBEK, J., 1973. *Innovations and Organizations.* Wiley. 212 p. ISBN 978-0-4719-8129-9.

ZIKMUND, M. *Řízení a optimalizace – Agilní projektové řízení.* [On-line]. 2010 [2018-11-19]. Retrieved from: http://www.businessvize.cz/rizeni-a-optimalizace/agilni-projektove-rizeni

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.

ŽIŽLAVSKÝ, O., VAVERKOVÁ, P., 2018. *Innovation Scorecard Design: A Process Innovation Case Study*. International Advances in Economic Research, roč. 24, č. 3, pp. 299-300. ISSN: 1083-0898.

LIST OF FIGURES

LIST OF TABLES