UNIVERSITY OF WEST BOHEMIA

FACULTY OF ELECTRICAL ENGINEERING
Department of Materials and Technology

DIPLOMA THESIS
Agile project management methods

Thesis author: **Taras Kovalenko**
Thesis supervisor: **Tomáš Řeřicha**

2021

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta elektrotechnická
Akademický rok: 2021/2022

ZADÁNÍ DIPLOMOVÉ PRÁCE
(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Bc. Taras KOVALENKO**
Osobní číslo:	**E20N0012P**
Studijní program:	**N0713A060011 Materiály a technologie pro elektrotechniku**
Téma práce:	**Agilní metody řízení projektů**
Zadávající katedra:	**Katedra materiálů a technologií**

Zásady pro vypracování

1. Seznamte se s vodopádovým způsobem řízení projektu.
2. Seznamte se s agilními metodami řízení projektu.
3. Porovnejte výhody, předpoklady, přínosy a úskalí obou metod včetně podmínek pro jejich aplikaci.
4. Seznamte se a popište projekt RPN – Regional Nomination Platform.
5. Navrhněte transformaci z vodopádově řízeného projektu do agilního řízení, popište tuto transformaci na projektu RPN.
6. Vyhodnoťte úskalí transformace, doporučte podmínky, při kterých je předpoklad pro úspěšnou transformaci a zhodnoťte své výsledky.

Rozsah diplomové práce:	**40 – 60**
Rozsah grafických prací:	**dle doporučení vedoucího**
Forma zpracování diplomové práce:	**elektronická**

Seznam doporučené literatury:

- DOLEŽAL, J.: Projektový management –Komplexně, prakticky a podle světových standardů. Praha: Grada, 2016. ISBN: 978-80-247-5620-2
- DOLEŽAL, J., KRÁTKÝ, J., CINGL, O.: 5 kroků úspěšného projektu. Praha: Grada, 2013. ISBN: 978-80-247-4631-9
- KŘIVÁNEK, M.: Dynamické vedení a řízení projektů. Praha: Grada, 2019. ISBN: 978-80-271-0408-6
- ŠOCHOVÁ, Z., KUNCE, E.: Agilní metody řízení projektů. Praha: Computer Press, 2019. ISBN 978-80-251-4961-4
- ŠOCHOVÁ, Z.: Skvělý ScrumMaster. Praha: Computer Press, 2018. ISBN 978-80-251-4927-0
- Internetové zdroje

Vedoucí diplomové práce:	**Ing. Tomáš Řeřicha, Ph.D.** Katedra materiálů a technologií
Oponent diplomové práce:	**Ing. Aleš Michálek** firma Unicorn
Datum zadání diplomové práce:	**8. října 2021**
Termín odevzdání diplomové práce:	**26. května 2022**

[illegible] L.S. [illegible]

Prof. Ing. Zdeněk Peroutka, Ph.D.
děkan

Prof. Ing. Aleš Hamáček, Ph.D.
vedoucí katedry

V Plzni dne 8. října 2021

Abstrakt

Agilní techniky – jsou techniky, které pomáhají efektivně řídit projekty. V dnešní době se agilní techniky velmi aktivně používají nejenom v oblasti IT, ale i v jiných oblastech průmyslu. Mezi nejvýznamnější techniky Agilního řízení patří Scrum, XP, FFD. Tato práce je zaměřena na popis problematiky přechodu od tradičních metodik řízení do Agilních technik, zejména do Scrum. V rámci literární rešerše jsou popsané metodiky řízení: Vodopádový model, PRINCE II, Scrum. Praktická část se zabývá přechodem od tradiční metodiky do Agile techniky řízení. Tento popis vypracován na základě zkušeností, které byli získané v rámci spolupráce se společností, která se zabývá vývojem softwaru. Tato firma nabízí kompletní řešení v oblasti bankovnictví, energetiky, médií a v mnoha dalších oblastech. V další časti je prezentován popis Regional Nomination platform. Výsledkem praktické části jsou doporučení pro přechod do AGILE způsobu řízení a implementační checklist. Je důležité si uvědomit, že tento checklist bude orientovaný na přechod právě pro výše zmíněnou společnost. V závěru je uvedeno hodnocení transformace.

Klíčová slova

Agilní metodiky, Scrum, PRINCE II, Vodopádový model, IT.

Abstract

Agile techniques - are techniques that help manage projects effectively. Nowadays, agile techniques are very actively used not only in IT but also in other areas of industry. The most important techniques of Agile Control include Scrum, XP, FFD. This work focuses on the description of the issue of transition from traditional management methodologies to Agile techniques, especially to Scrum. The management methodologies are described in the literature search: Waterfall model, PRINCE II, Scrum. The practical part deals with transition from the traditional methodology to the Agile transition to the Agile management technique. This description is based on experience gained in cooperation with a software development company. Thus, the company offers complete solutions in the field of banking, energy, media and many other areas. Next part provides a practical description of the Regional Nomination platform resulting from the practical part are recommendations for transition to AGILE management method and implementation checklist. It is important to note that this checklist will be transition-oriented for the aforementioned company. The evaluation of the transformation is presented in the conclusion.

Key Words

Agile techniques, Scrum, PRINCE II, Waterfall model,

Declaration

Hereby I declare that I solely wrote this diploma thesis, with the aid of literature and sources referenced in List of References, which is a part of this diploma thesis.

Further I declare that all software used while working on this this diploma thesis is legal. All images which are take over Unicorn internal documents are created in UUBML – Unicorn Universe Business Model Language.

....
Signature

In Plzeň on 2018-05-23 Taras Kovalenko

Acknowledgment

I would like to thank Unicorn for the experience I gained during my collaboration and for the support my superiors gave me along the way.

I would also like to thank Tomáš Řeřícha, the supervisor of my diploma thesis, who helped me a lot and guided me in the elaboration of this thesis.

Contents

List of symbols and abbreviations

Symbol	**Description**
DOD	Definition of Done
DT	Dual track
FAT	Factory acceptance test
NSLDS	North Sea dispatch system
PO	Product owner
SM	Scrum master
SW	Software
UAT	User Acceptance test
US	User Story

Introduction

Since the 1970s various management techniques have developed rapidly in the field of IT. One of the most important techniques is the Waterfall Model and PROMPT (the predecessor of PRINCE II). These methodologies were based on the principle of strict adherence to the plan and also the gradual processing of the assignment. The main disadvantage of these methodologies was that they failed to respond to ever-changing customer requirements. Subsequently, based on these methodologies, new methodologies such as V-model, W-model, PRINCE II were created. Common name used for these methodologies is traditional management methodologies.

Unlike traditional management methods there are agile management methodologies. The main idea of these methodologies is that the basic goal of the project is not the fulfilment of the initial plan, but customer satisfaction. Therefore, less emphasis is placed on documentation and detailed planning. In addition, there is a lot of emphasis on things that really add value, such as writing the code itself. For this reason, we may sometimes come across the view that Agile Control Techniques are just a way for programmers to avoid tedious work with documentation and embark on a job they enjoy - programming. In my opinion, this idea is completely wrong. When switching to Agile, not only the content and scope of the programmer's work will change. The structure of the development team and the approach to the project not only internal management but also customer management should be adapted to agile management.

Today, there are a huge number of publications describing individual management techniques. However, the issue of transition to Agile Management is usually already addressed by each company separately. The aim of this work is to describe transition to Agile in one particular company within a specific project. This work could serve as inspiration for those who make decisions whether the transition to Agile can be successful or whether it will bring more problems than benefits.

The thesis is divided into three parts. The first part, which is theoretical, describes several techniques of the IT project. First, the most basic methodology of traditional programming control is described - the Waterfall Model. The description of this technique is very important because all the projects whose transition to Agile will be described in the practical part were originally managed using WM. The following is a description of another traditional methodology - PRINCE II. The whole work, which describes transition to Agile, is specifically focused on transition specifically to the Scrum methodology. Therefore, the last methodology that is described in the practical part is the Scrum methodology. The aim of the practical part was to briefly describe the key (for the given company) methodologies, to state their strengths, weaknesses and also its properties that have the greatest impact on the project. It is important to realise that the aim of this chapter was not to describe all methodologies of IT projects, whether traditional or Agile. The description of even the most used methodologies would lead either to a dramatic increase in the scope of work, or to the fact that these methodologies could be described only very briefly.

The following section will describe transition to Agile. As already mentioned, the practical part describes the transition specifically in Unicorn and therefore some steps may be missing or, conversely, there may be steps that do not occur during a normal transition. The main goal of this chapter is to describe the transition in general so that the reader has an idea of what a standard transition should look like.

The practical part consists of two parts. The first part describes the RNP (regional nomination platform) project, which belongs to Unicorn. Its general structure will be described, the main business processes will be briefly described, and a description of the structure of the development team will also be given. The second part describes the transition of the RNP project to Agile. Based on this chapter, a checklist will be developed, which should serve as a guide for other projects. All information about the project and the transition steps were obtained through cooperation with Unicorn, either in consultation or throughout documentation.

1 Theoretical introduction

During the second half of the last century, the project management methodology was actively developed. There are several reasons that led to such active development.

The first reason was that during World War II there was a need to solve projects requiring maximum performance with a very limited amount of material and human resources. The same problem existed after the war when the economies of many countries (eg. Japan) were destroyed. Experience gained during and after World War II became the basis of future methodology.

The growing number of international corporations also had a very strong influence on management methodology. There was a need to manage complex projects effectively, which very often involved not only manual work of a large number of people, but also work of experts from various fields.

Another impetus for the development of new management methodologies was development of information technology. On the one hand, the IT sector has accelerated the management of a large number of people, but on the other hand very often the classic management methods were unsuitable for this area. Therefore, many management techniques were originally designed to manage IT-related projects. An example is the Waterfall Model (WM), Extreme programming (XP), SCRUM, PRINCE II (or its predecessor PROMPT II).

1.1 Traditional management methodologies

Traditional management methods have been widespread in many companies since the 1970s. Their accuracy and structure have led to these methods being used as government standards. And their use is not limited to IT.

Common feature of these methods is that a lot of effort is spent in the initial stages of the project: planning, designing. This often led to unnecessary risks, as it was very difficult to fully plan a large project in the long term. Also, any change in the requirement took a relatively long time for the plan to be adjusted. Changes in traditional management requirements are therefore undesirable.

1.2 Waterfall model

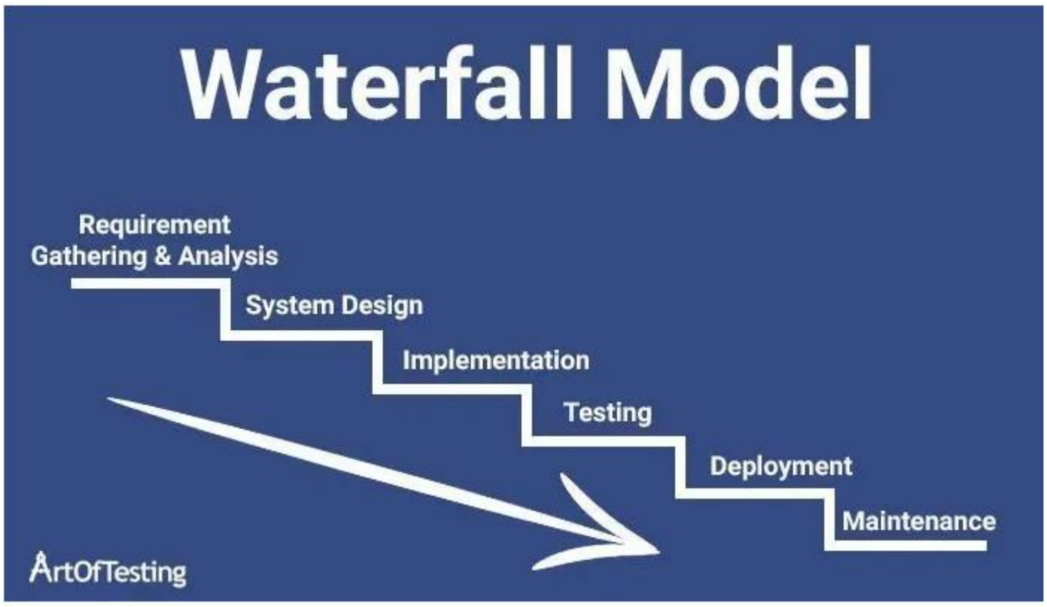


Figure 1 Waterfall model [1]

One of the oldest and most used traditional management methodologies is the Waterfall Model. This is based on the principle of a sequential approach to project work.
The first detailed description of the Waterfall Model was made by W.W. Royce in "Managing the Development of Large Software Systems" in 1970[2]. This model was created as an improvement on the simple phase models introduced by Herbert D. Bennington in 1956, which contained 9 phases.
W.W. Royce defined six phases of the development process so that each phase was dependent on the previous one. According to Royce, the main stages of software development are:

- Requirements specification
- Proposal
- Implementation
- Integration
- Testing
- Installation

As already mentioned, the main principle of the waterfall models is a sequential approach to the work on the project. This means that the project should gradually go through all phases of the project. You can proceed to the next phase only if the previous one is completely

finished. According to the WM between the individual phases, no iterations should take place.
In practice, it has proven that if the project is managed based on a waterfall model, it is more effective if the programmers are divided into several highly specialised teams (developers, testers …).
The United States Department of Defence adopted DOD-STD-2167 A "Defence Systems Software Development" in 1988[3]. This standard used the main principles of the waterfall model. This standard was abandoned in 1994, when MIL-STD-498 was adopted [4]. The model was used as a standard in Europe and was actively used by Toyota [5].

1.2.1 Criticism

It is important to realise that the main idea of WM is that the initial phases of the project (requirements specification, design) are as important as the implementation phases (the phase that brings the most added value to the project). Therefore, the effort that must be made during the first stages must be comparable to the effort that has been made to implement the source code. This approach makes it easier to detect errors during the initial phase, which helps save time and money needed to correct the error. It is notoriously known that if an error occurred in the final stages, it would lead to greater financial losses.
However, the main disadvantage of waterfall models is that testing is done at the end of the project. And it is clear that eliminating all the errors which are found is very costly and time consuming. It is often stated in the literature that if too many critical errors occur during testing, the only suitable solution is to start working on the project from the very beginning.
According to WM, it is necessary to devote a lot of time to creating complete and detailed documentation, right at the beginning of the development process. On the one hand, quality documentation is very useful during the training of new employees. Because if new team members read the documentation, they will get most of the information they need without having to train it. This makes the WM-driven system more resistant to staff changes. On the other hand, very detailed documentation is often unnecessary and only makes the project more expensive.
Another disadvantage of WM is that the customer only cooperates with the development team in the first phase. Afterward, the finished software is handed over to the customer only in the final phase. As a result, the development team will not receive any feedback during software development. This is a very big disadvantage which results in other factors that limit the use of WM in practice. First, the customer very often gets an idea of what the final

product should look like when the software is at least partially accepted. In addition, it is clear that for a project managed by WM, any change in requirements is undesirable and involves large financial and time losses. The waterfall model is therefore suitable for projects where the customer knows exactly what the final product should look like and knows that its requirements will not change during development. Otherwise, it is better to use another control method.

In my opinion it is a very important disadvantage of waterfall models that the user does not participate in the development process at all. All requirements, validation, changes to the original project are addressed with the customer. As a result, the development team receives all the feedback from the customer, not the user. Often missing communication with the user has a very negative effect on the quality of the resulting program.

1.2.2 V model

As it can be seen, the waterfall model has several disadvantages, due to which its use is very limited. Therefore, there has always been an effort to modify this model in order to at least partially eliminate its weaknesses. The best-known modified models are: spiral model, V - model, W - model, sashimi model.

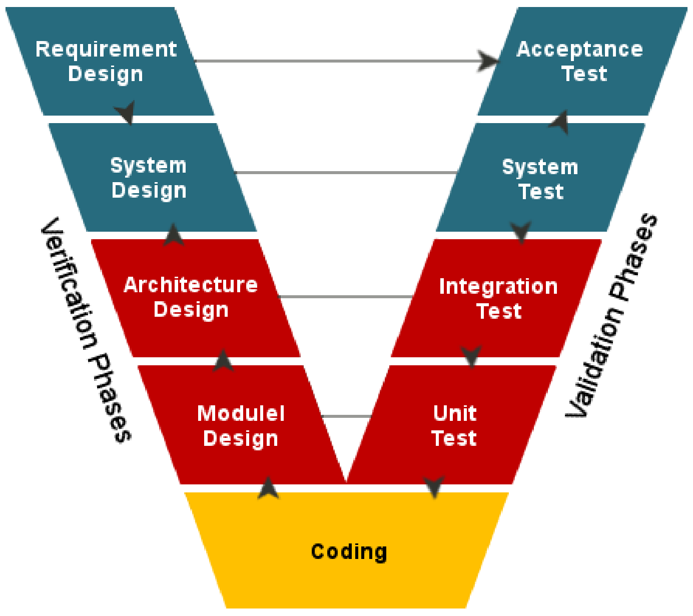


Figure 2 V model in programming [6]

For example, the V model uses a dual approach to development: the design process should be mirrored in the testing process. So, the Design strings

- Requirement design
- System design
- Architecture design
- Model design

match the test strings

- Acceptance test
- System test
- Integration test
- Unit test

The central point of the projects is the development process (coding). This model can be seen in Fig. 2. The shift from the left to the right corresponds with the time shift. The higher the objects are, the greater their level of abstraction. The main difference between WM is that the testing process should be started together from the corresponding phase. Thus, integration tests should be written at the time Architecture Design is developed.

This approach helps to fix one of WM's main problems that the testing process only begins in the final stages of projects. Such structure helps to reduce the risk of an error occurring in the initial stages which will only be detected in the final stages. In addition, the total number of errors that occurred in the initial stages (stages that precede coding) is smaller. It is also important that each level of abstraction of the model corresponds to its level of tests.

Despite the fact that the V-model corrects some of the disadvantages of the Waterfall model, the V-model also "inherited" many of WM's weaknesses. An example is the inability to respond effectively to changing requirements, the large amount of effort that needs to be put into the initial stages. In addition, the risk of finding a mistake in the last phase of projects is smaller compared to WM but still greater if compared to Agile techniques. Another disadvantage is that the total time devoted to the initial stages of the project (which precede the coding) is greater. It is associated with the fact that in addition to design, it is necessary to write tests.

1.3 PRINCE II

Another important traditional methodology is PRINCE II (PRoject IN Controlled Environment ver.2) - is a methodology for project management that originated in the United Kingdom in 1996. This methodology was created as a result of a revision of the previous

PRINCE methodology. The PRINCE methodology was also created as a result of a revision of the previous methodology, which was called PROMPT II.

1.4 Older versions

The PROMPT II methodology (PRoject Organization, Managing and Planning Techniques) was created in 1975 by Simpact Systems Ltd. A group of project managers actively participated in the development of the methodology who gained very extensive knowledge during their work at IBM. The PROMPT II methodology was widely used in the management of state projects but its use was limited only to IT projects.

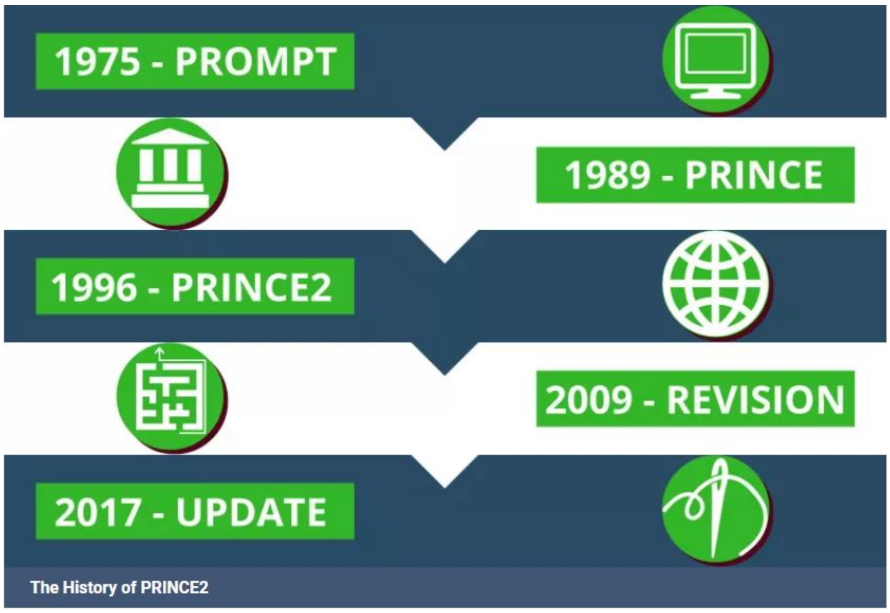


Figure 3 The history of PRINCE II [7]

The PROMPT II methodology defines the stages of the project here:

- Feasibility stage
- Initial stage
- Stage specification
- Design stage
- Development stage
- Installation stage
- Operation stage

As mentioned in 1989 the PROMPT II methodology was revised by the Central Computer and Telecommunications Agency (CCTA) [8]. The main purpose of this revision was to create a new version that would be suitable for use as a state standard for information technology project management. The new version was named PRINCE, which originally meant PROMPTIn CCTA Environment.

The PRINCE methodology comprised a very innovative principle, which was that the work on the project should be provided from different perspectives: user, business and technical. In addition, each of these perspectives should be coordinated by the responsible person: Business Perspective Coordinator, Technical Perspective Coordinator, User Perspective Coordinator. These roles also occur in the PRINCE2 methodology.

Due to its clearly defined structure, the PRINCE methodology made it possible to manage the project very efficiently. Therefore, it soon began to be used not only as an official state standard but also as a standard for many IT companies.

1.5 Main pillars of PRINCE II

The new PRINCE II methodology was created in 1996 as a result of revision of the PRINCE methodology. The resulting methodology functioned as a system of Processes, principles and processes integrated into the development environment.

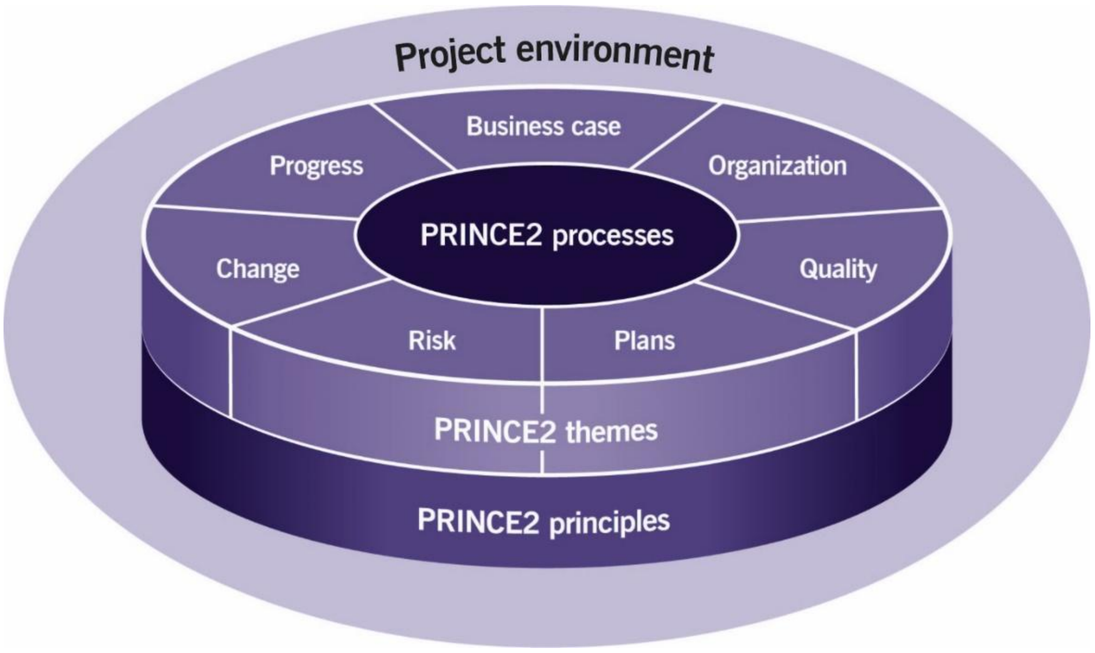


Figure 4 Main PRINCE II pillars [9]

1.5.1 Principles

According to the PRINCE II methodology, 7 main processes should take place throughout the project. These processes help to manage the entire project effectively and increase the quality of the final product. These principles are as follows:

- Continued business justification

According to PRINCE2, it is very important that when working on a project, all changes have business rationale. Otherwise, there may be situations where the development team puts a lot of effort into work that brings none or almost no added value to the project. This principle (or an alternative to this principle) can be seen in many agile methodologies.

- Learn from experience

Each project brings unique experience to the development team. If the development team is able to learn useful lessons from each project then the quality of the development process rises with each project. Therefore, it is important to go through all similar processes at the beginning of each project and get useful lessons that may be applicable to the project. The PRINCE2 methodology defines a clear criterion against which to assess whether a lesson is useful for a given project: if a lesson brings about change, then that lesson is useful.

- Defined roles and responsibilities

It is very important for a successful project that every employee involved in the project has a clear idea of what is expected from him and what he or she is responsible for.
There are 3 areas of interest in each project
- Business
- Users
- Suppliers

As it can be seen, a similar view of the division of roles already existed in the PROMPT methodology.

- Manage by stages

According to the SMART method goals should be precisely defined and be measurable. It is clear that staged management helps to define the objectives very precisely and define goals very accurately and determine whether these goals have been achieved. In addition, if we divide the development process into several stages we are not forced to plan in detail the processes that will not take place in the near future. It brings some flexibility to the development process.

- Manage by exceptions

This principle is that most of the senior management powers should be delegated one management level lower. Top management should intervene in delegated activities only when necessary.

PRINCE2 defines a very precise procedure for detecting situations where problems have occurred in a delegated job. Each process has 6 basic goals:

- Time
- Costs
- Quality
- Scope
- Risk
- Benefits

Tolerance limits must be set for each of these objectives. If it were probable that the tolerance limits might be exceeded, the intervention of senior management would be necessary.

- Focus on product

The big problem with the waterfall model was that this model was primarily focused on processes. This led to the already mentioned problem, when a lot of effort is put into a thing that brings almost no added value. However, if the goal of the project is to deliver quality software with a minimum of time delay, great emphasis should be placed on the product.

- Tailor to suit project environment

All the principles mentioned so far help to integrate this methodology into the development process to a certain extent. In addition, the effectiveness of these processes depends on how successful the integration into the environment will be. Adaptation of the project to the development environment is a feature of PRINCE2, which ensures its wide use in various projects (not only in IT projects, unlike PRINCE).

1.5.2 Processes (Stages)

The PRINCE2 methodology defines the following stages:

- Starting up a project

During this stage, it is necessary to obtain all the necessary information that is needed to start the project. It is very important that this process is completed as soon as possible.

- Directing a project

Within this stage, most of the managerial work should be delegated to middle management. This process is closely related to the principle of manage by exceptions

- Initiating a project

During this phase, a business rationale for the entire project should be made. According to the relevant principle, commercial justification should take place on an ongoing basis. In addition, project parameters should be defined.

- Managing a stage boundary

During this phase, the current phase of the project should be completed and the next phase should be planned.

- Controlling a stage

According to the manage by stages principle, the stages should be managed and controlled. It is a process (or rather a group of processes) that fulfils the daily work of a manager. Specifically, it focuses on the processes by which the manager manages individual activities

- Managing product delivery

During this stage, processes take place in order to ensure the timely delivery of a product that has satisfactory properties.

1.5.3 Themes

Themes are main managerial tasks that should be fulfilled continuously during the project. There are 6 topics defined for the PRINCE II methodology, which need to be addressed throughout the project.

- Business Case

Creating and maintaining a record of the business rationale of the project.

- Organisation

Defining individual roles.
Assigning roles to the job for which they will be responsible. And the subsequent provision of communications between roles.

- Quality

Defining the product requirement and ensuring that the delivered product will meet these conditions.

- Plans

Elaboration of the plan and all used methodologies PRINCE II. This includes product description, its Breakdown Structures, product flow description.

- Risk

Identification of risks and their causes. Development of a risk management plan.

- Change

Problem identification and so-called project bottlenecks. Implement changes that would eliminate these issues.

- Progress

Creating a record of the main parameters of project management. Reports on key stages of projects as well as reports on the final stage and project completion.

1.5.4 Criticism

As already mentioned, one of the basic features of the PRINCE II methodology is the ability to adapt to almost any project. That is why this methodology, which was originally intended for IT projects, has been actively used in a wide range of different projects.

The success of the methodology was also due to the fact that PRINCE II offered a clearly defined framework. This means that the management of the system is relatively simple, all roles are precisely defined, in addition controlling all stages of projects is very effective (especially due to stage management and exception management). Other advantages of PRINCE II include efficient control of resources and increasing quality of the final product.

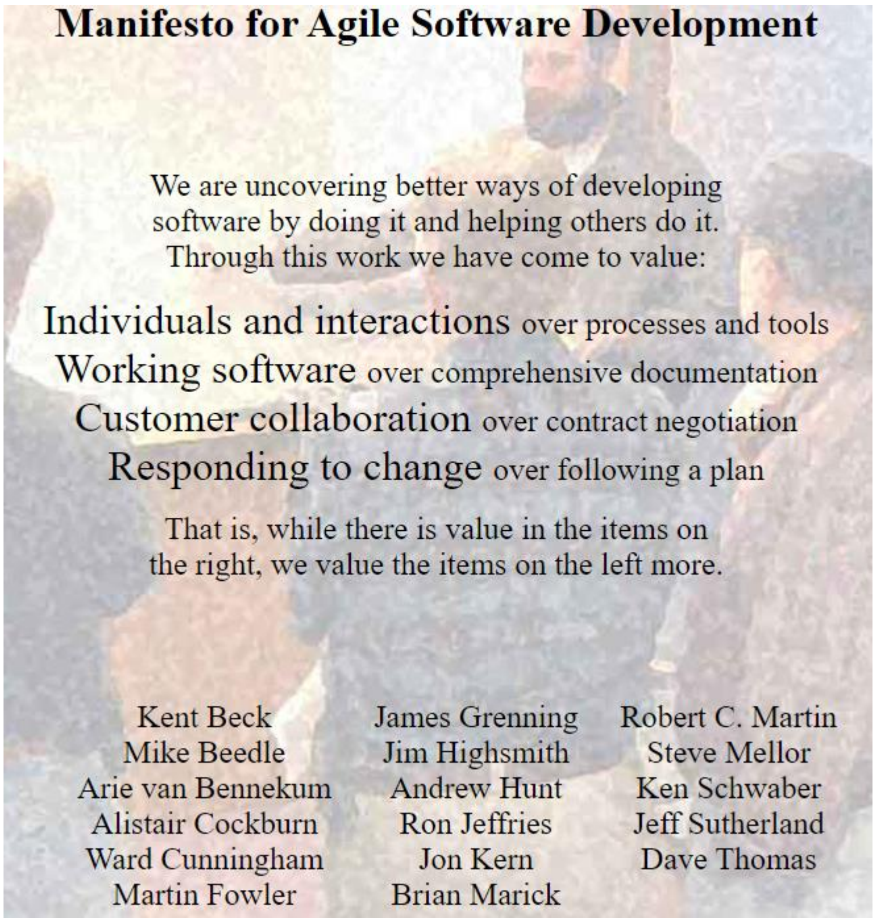
However, the PRINCE II methodology has several unfavourable features that limit the use of this methodology in real life. One of the main disadvantages is the classical disadvantage that occurs in almost all classical management methods. It is mainly a very small ability to respond to changes in the assignment.

Another important disadvantage is the absence of any methodology regarding the methodology of approaches to the management of supply contracts, project participants and other processes that the creators excluded outside the framework. It is assumed that each project manager chooses his own methods and approaches to such work.

Paradoxically, one of the main features, which was considered one of the biggest advantages of this model, led to one of the biggest disadvantages of the system. As already mentioned, PRINCE II can be adapted to the environment of any project. However, the methodology itself does not describe the adjustment process so that it is possible to assess accurately whether the adjustment was made correctly. In other words, the manager should not only determine how the adaptation shall take place but also decide how the success of the adaptation shall be evaluated. As a result, almost any failure of projects can be justified by the fact that in these projects the PRINCE II methodology was not properly adapted to the environment of the given projects. As an example we can mention failed projects of the British government [10][11][12]. And it is very ironic that methodology that places great emphasis on project control is set in such a way that it is not possible to check whether the cause of the failure is the methodology itself.

1.6 Agile management methodologies

Due to the above-mentioned shortcomings (limited cooperation with the customer, inability to respond effectively to changes in requirements, etc.), using traditional methodologies was very risky. In the 1970s, therefore, new methodologies began to emerge. These methodologies were able to be adapted to frequently changing requirements (XP, Scrum, Crystal). In addition, according to the new methods, the product should not be delivered at once, at the end of the project, but continuously, in parts. As a result, the customer received at least part of the product earlier. And that's why they get an idea of what the final product would look like sooner.



Manifesto for Agile Software Development

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

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

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

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

Figure 5 Manifesto for Agile Software Development [13]

These techniques received a common name - agile management methodologies. The main principles that should guide any agile methodology are described in the Agile Manifesto (see Figure 5). This manifesto was created in 2001 during a meeting of 17 managers using various agile methodologies.

Today, the popularity of Agile is constantly growing. For example, in 2020, the percentage of teams using Agile was 37%. By 2021, 86% of the team was using Agile [14].

1.7 Scrum

Scrum is one of the most frequently used agile techniques these days. The Scrum methodology was first described in "The New Product Development Game" by Hirotaka Takeuchi and Ikujiro Nonaka in 1986. The term scrum was previously used in rugby and means the initial state of the team before the ball was thrown.

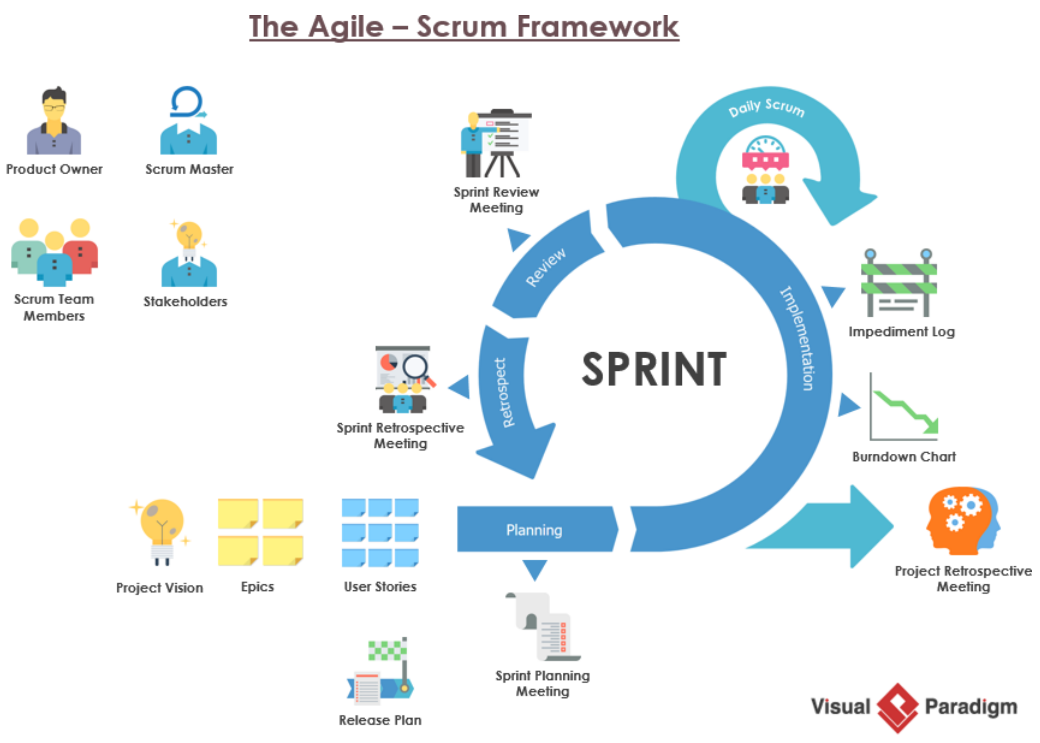


Figure 6 Scrum Framework [15]

The main idea of this methodology is to divide the development process into development cycles - sprints. Each of these sprints should have well-defined goals and, in addition, should bring some added value to customers. The typical sprint time is 1 to 4 weeks. During each sprint, there should be five-minute daily meetings of the development team (Daily Meeting).

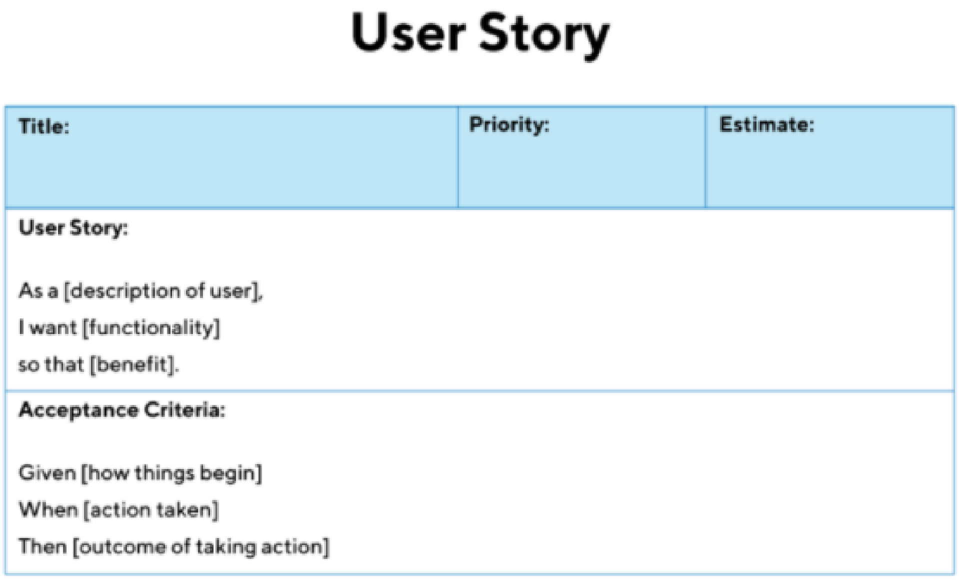
As already mentioned, the name of the methodology was taken from sports terminology. One of the main features of Scrum is just following the philosophy of sports wrestling. Other important principles that should be followed in a project managed by Scrum are transparency and active feedback with the customer.

1.7.1 Artefacts

The Scrum methodology defines several important artefacts using which helps to effectively plan and regulate project work.

The first artefact that arises during software development is the **Backlog project**. This is a list of all requirements that should be done within the project. All requirements should be prioritised. Prioritisation is done on the basis of risk, business value, dependencies, size, and date needed. Each sprint has a separate backlog called a sprint backlog.

Project increment is a ready-to-use part of a product that must be implemented by the end of a sprint.



User Story

Title:	Priority:	Estimate:
User Story: As a [description of user], I want [functionality] so that [benefit].		
Acceptance Criteria: Given [how things begin] When [action taken] Then [outcome of taking action]		

Figure 7 User Story template [16].

The main means for describing project requirements are User Stories(US). There is only one task that needs to be described in the Users Story. The description should be made using simple language and should be as concise as possible (typical pair of sentences). The following information is usually provided:

- User type
- What functionality he needs
- For what reason

The **Burndown chart** is used to visualise the work done during the sprint and also the work that is to be done. It is a diagram that shows, for each day of the Sprint, how much work was done on that day and how much work remains (see Fig. 8). Burndown charts can be used to visualise periods of time longer than a single sprint. The Burndown chart is not directly part of the Scrum methodology.

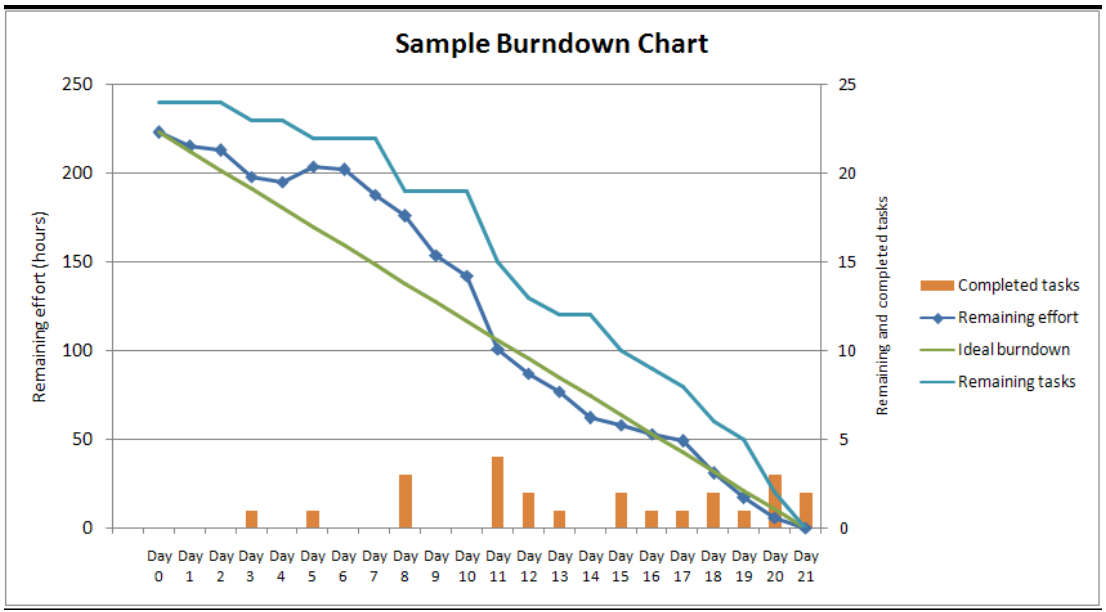


Figure 8 Burndown chart, example [17]

1.7.2 Roles

One of the main pillars of the Scrum methodology is clearly defined roles. The Scrum methodology not only defines new roles, but also provides a completely new perspective on the structure of the development team. As stated, the core philosophy of Scrum is based on the principle of wrestling.

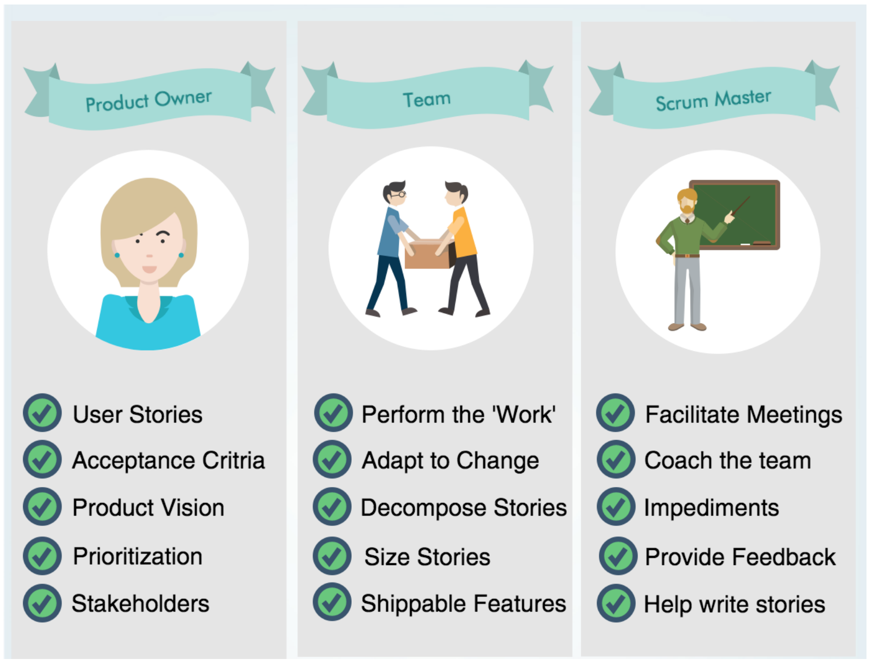


Figure 9 Roles in Scrum [18]

This means that the development team must be:

- Self-organising and independent. Probably, it could not be a successful team that only listened to the teams of coaches without individual decisions throughout the match.
- Multifunctional. Because it is difficult to imagine a team which has goalkeepers or attackers only.

As already mentioned, Scrum also defines new, unique roles that should support the development team: Product Owner (PO) and Scrum Master (SM).

Main responsibilities of the members of the project team are following.

Product owner

The main idea of this role is that an external employee (provided by the customer) shall be involved in the development process. This employee should be able to continuously define new requirements during development and then evaluate the results of the development team. A similar idea occurs in many other agile techniques. For example, according to the methodology of Extreme Programming, the customer shall provide an employee who shall be the person who will use the developed software.

In the SCRUM methodology, this role is more complex. The Product Owner is responsible for many other things, and they are much more demanding.

The main work for which the PO is responsible is the fulfilment of the Backlog tasks and their prioritisation. Only the PO is responsible for ensuring that all backlog points are accurately and clearly expressed [19]. In addition, the PO shall ensure that all members of the development team understand these points to the extent necessary to ensure satisfactory quality of the final product.

The PO should work not only with the development team. In the simplest case, the PO should communicate with its superior. Based on this communication, new requirements should arise and customer feedback shall take place. However, it often happens that we have more interested people on the customer's side who "want to talk". In this case, it is often considered whether several Product Owners should work on the project. However, the official guide says unequivocally: There shall be only one product owner. In this case, there should be one PO that should represent the wishes of the commission of all stakeholders.

The secondary objectives of the tasks that the PO deals with are:

- release notification
- active participation in customer tests
- education dev. team in the field of business
- training of stakeholders in SW development

In essence, PO is a mediator between the development team and the customer.

Given that the PO has the right to decide what will be done on a particular sprint, it is expected that:

- has a clear idea of what the final product should look like. Ideally, it should be the user of this project.
- is well versed in the market situation, especially in terms of competition
- has a vision for developments in the project area

At Scrum, the product owner is the only person who is ultimately responsible for the return on investment (ROI) of the product development efforts.

Scrum Master

The Scrum master is responsible for ensuring that the work on the project proceeds exactly according to the Scrum methodology. To achieve this goal, it is necessary that each member of the Scrum team clearly understands the basic principles of Scrum. This means that the Scrum master should, if necessary, provide training for the team member. It is logical that the SM is responsible for checking that all project participants follow the rules and procedures set by the Scrum methodology.

As part of the work on the project, SM should cooperate not only with the Product Owner and the development team but also with people who are not members of the Scrum team. It is the SM that should ensure that these people know how to proceed in order to achieve maximum efficiency in a Scrum environment.

Another very important work that SM is responsible for is planning and organising the implementation of Scrum into the project. This means that he is a member of the team that has been working on the project from the very beginning.

Development team

Main purpose of the development team is the creation of increments.

The authors of the methodology stated that success of a team in the development process can be achieved if the same ideas are applied to the organisation of the development team as apply to the sports team. This means that all team members should be able to make decisions and organise independently. It is obvious that if members of the rugby team follow instructions which they received from the coach during the match, such a team will never be able to win. Therefore, according to the Scrum methodology, the main reason why traditional methodologies often fail is their strict management where each member of the team is very independent.

First, the team must be independent and self-organising. This means that only the development team can decide:

- how to convert backlog product to increment
- who will be responsible for the implementation of individual User Services
- in what order the individual US will be implemented, etc.

It is also important that it is a basic structural unit. There should be no sub-teams within the development team. The development team itself should not be dependent on other development teams. This requirement implies that the team should be multifunctional (to be able to solve its problems on its own).

1.7.3 Meetings

To ensure the proper functioning of the Scrum methodology, it is important to ensure the correct flow of information. This is ensured by a number of meetings which help not only to acquaint project members with the emergence of new or meeting existing requirements but also to ensure that changes that increase the efficiency of the entire company are implemented within the company.

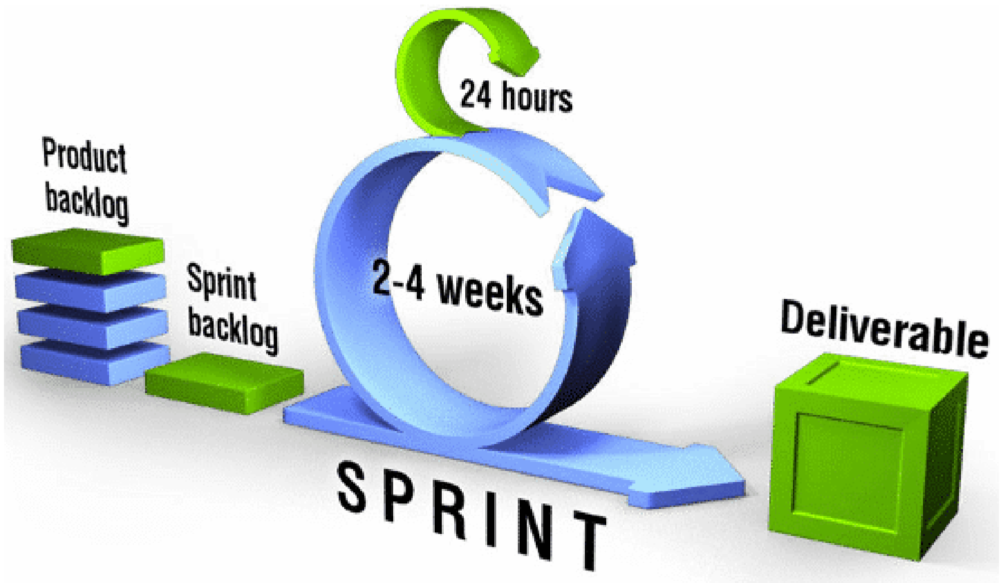


Figure 10 Sprint in Scrum [20]

he Scrum methodology defines both regular meetings which should take place in each meeting as well as one-off meetings taking place at the beginning and end of projects. In this chapter, we will focus only on regular meetings. One-time meetings will be described in the following chapter.

Sprint planning meeting

The sprint planning meeting should take place before the start of each sprint. All SCRUM Team participants should attend this meeting. From time to time, this meeting may be attended by external experts who will help to address certain specific aspects of the project.

The full amount of the next sprint's work should be scheduled for this meeting. The main purpose of this meeting is to answer the following questions:

- What are the goals of the sprint, ie. all the work that should be done during the sprint?
- How shall these goals be met?

The first question should be discussed by all members of the SCRUM Team. Planning should take into account the work done during previous sprints as well as the experience gained during these sprints. It is important that each participant in the meeting gets a clear idea of the work that will be done during the next sprint.

The second issue should be dealt with mainly by the development team. Once all the requirements have been set, the developer team should think about how they will achieve these goals. At the end of the Sprint planning meeting, the development team should inform the Product Owner and Scrum Master which ways the set goals will be achieved.

Feedback plays a very important role throughout the meeting. During the meeting, the developers had to communicate actively with the customer. The scrum master should be able to adjust the scope of the planned work if a party considers the current scope to be too small or, conversely, too large. Alternatively, the Scrum master should be able to find a compromise solution.

Daily Scrum

As the title suggests, this is a daily meeting attended by the development team. This meeting should be very short, in no case longer than 15 minutes. It is important that this meeting takes place at the same time each working day. The meeting should start even if one of the participants is not present.

During the meeting, each member of the development team should answer the following questions:

- What did I do yesterday?
- What will I do today?
- What obstacles do I see that prevent me from reaching the sprint goal?

It is logical that the Daily Scrum will be much more effective if all its participants are ready.

Scrum master guarantees the implementation of daily meetings. However, the development team is responsible for ensuring that these meetings take place every day.

SPRINT REVIEW

This meeting takes place at the end of each sprint. The aim of this meeting is to check what requirements were already implemented or to modify the backlog. All stakeholders are attending this meeting.

First, the product owner should announce what part of the backlog was met. In addition, the PO should express its views on the current state of the backlog and on how the individual elements of the backlog could meet or change.

Subsequently, the development team should show the entire product implementation. In addition, developers should comment on what they think were successful and what they were having trouble with. They should also say how these problems were eliminated or what changes should be made to minimise their occurrence in the future.

During the sprint review, the market analysis should also be discussed. Also analysis of the target audience and their influence on the implementation of customer requirements. The result of this meeting is a restored backlog.

SPRINT RETROSPECTIVE

The main purpose of this meeting should be to develop a proposal to improve the development process. It is assumed that all the necessary knowledge and ideas should be gained during the previous sprint. This meeting should be attended by the entire development team, which should be managed by the Scrum Master. The meeting should take place between the Sprint Review meeting and the start of the new sprint. This sprint should take place in a few hours.

At the beginning of the meeting, it is necessary to perform an analysis of the previous sprint. Development team members evaluate the efficiency of the use of resources and inputs used in the previous sprint. Subsequently, all the ideas that might lead to the improvement of the project should be discussed. How to implement these ideas should also be discussed. It is important that the suggestions of all team members are taken into account.

During the final phase, the development team should focus on establishing a plan for implementing changes that might lead to the improvement of the development process. This is a very important step, as ideas to improve the process can arise at any time during the sprint.

1.7.4 Criticism

As already mentioned while planning a Sprint care should be taken to ensure that each Sprint brings some added value to the customer. This principle can be seen in Figure 11.

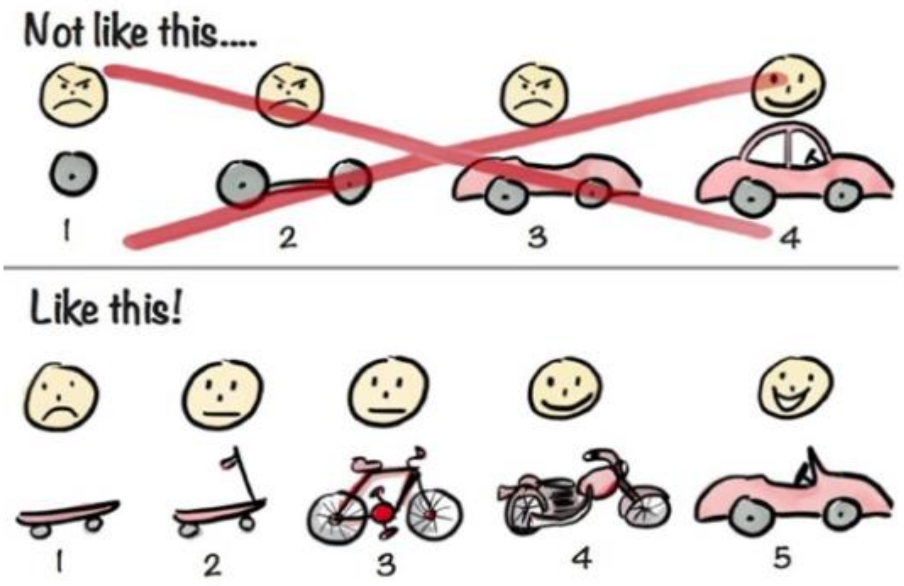


Figure 11 Principle of the developing in Scrum [21]

This principle can be applied relatively easily, for example, to the eshop. In this case, you are able to deliver a product with a minimum number of functionalities very quickly. You are then able to add to this product in each subsequent sprint. However, there are numbers of projects where it is not possible to proceed in the same way at all. In the publication work Scrum black book, one can come across a very interesting idea: "it is not possible to build a bridge using the Scrum methodology"[21]. It is logical that when building a bridge, the only useful product is a completely finished bridge. Therefore, it is important that the project itself is such that it is possible to continuously deliver useful (for the customer) parts of the project.

Another very important problem of Scrum is that all its processes are mandatory and forever. We cannot say that if we devote a lot of time to work on some aspect of it (such as analysis), it would be possible not to do this work in the future. All principles, rituals and meetings should take place at each sprint. This idea contradicts the main idea of programming, which says that often repetitive work should be automated.

While describing the PRINCE II methodology I stated that the main contradictory argument against the criticism was the answer that the source of the error is not the methodology itself but its unhealthy implementation. A similar situation can be found in the Scrum methodology. All the disadvantages of Scrum described below can be assessed to a large extent as a consequence of incorrect implementation of the methodology.

As already mentioned in Scrum, everything is evaluated by the number of completed User Stories. Unfortunately, this leads to many developers trying to postpone work over the US that may require more time than originally specified by the Product Owner.

Another problem we may encounter while working on a project is that very often the only thing that evaluates the success of the project is the number of completed User stories. Unfortunately, the quality of the code is very often forgotten. In this case, development team members are not motivated to improve the code quality. After all, high-quality and low-quality tasks will be evaluated in the same way.

Postponing the processing of errors is also a very important negative effect which you can deal with. Unfortunately, very often found bugs are simply added to the Backlog of the following sprint. Therefore, most problems are solved with a delay of one Sprint. The reason for this is that according to the sprint, the Sprint Backlog should be fixed at the beginning and not change. It means that no one with the developer will be willing to do extra work which no one appreciates.

Given the above information, it is clear that the transition to Agile may not produce the expected results. According to the IT Project Success Rates Survey TM from 2018, 36% of the team had difficulty switching to Agile. And in 3% of cases there was a complete failure [22] [23].

2 Scrum and Waterfall model comparison

As already mentioned, the main advantage of Agile techniques is the ability to actively respond to changing requirements. And therefore Agile is actively used in a project where the customer does not have an exact and complete idea of what the final product should look like. In such projects, it is very common to encounter a situation where the customer can (or plans to) get an idea of the final product only when they see a part of the product.

Another important advantage of Agile over traditional methods is the very fast delivery of at least partially finished products. For example, according to the Scrum methodology, the first version of the software should be released during the first month. This is a huge advantage for small and medium projects. According to the waterfall model, the result of the work should occur only at the end of the project. Which is a disadvantage on the one hand, but it is important to realise that there are projects that can only be solved this way. The above information also shows that agile-managed projects undergo morning testing which helps to detect errors faster.

Another important difference is the customer feedback. According to traditional methodologies, the customer will see the project at the end. As a result, the development team does not receive any feedback from the customer during the processes of development and testing. According to Agile methodologies, the customer should constantly work with the development team. On the one hand, this is a great advantage, because it is possible to correct a large number of errors at the beginning. On the other hand this is a challenge as the Development Team member is required to have specific skills that allow him to work effectively with an external employee.

Furthermore, a very important difference between the methodologies is the approach to the cooperation of the development team with the customer. First, it is important if the development team is well experienced and motivated so that they can organise their activities independently. In addition, training new employees for Agile projects is more complicated. As far as traditional methodologies are concerned in this case requirements for the experience of the members of the development team are smaller. In addition, training new developers is easier in this case, as a new member of the development team can read the company's documentation. In general, in traditional methodologies as the main means of information sharing use documentation, Agile techniques rely more on knowledge.

2.1 Comparison of the success of large and small projects managed by traditional and Agile methods

Size of the project has a very important effect on the success of any methodology. In general, the probability of success of the methodology of large projects is lower compared to small projects. This dependence can be seen in Figure 12. While talking about small projects, the probability of success of Agile and traditional methodologies is practically the same. [24].

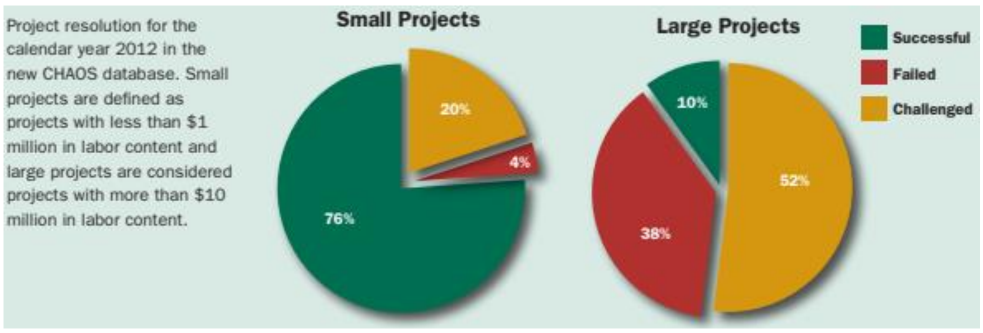


Figure 12 Comparison of the success of Agile of the large and small projects [24].

A research of Riga Technical University brought practically the same results [25]. According to this research, for small projects, the success rate of the waterfall model is slightly lower than the success rate of Agile Techniques. For large projects, this difference is much more significant. This difference can be seen in Figure 13, which shows the difference between the expected and the actual budget. As it can be seen, management of large projects by the waterfall model causes a larger difference between the expected and the actual budget (compared to Agile). A similar dependence can be observed when comparing planned and actual person-days.

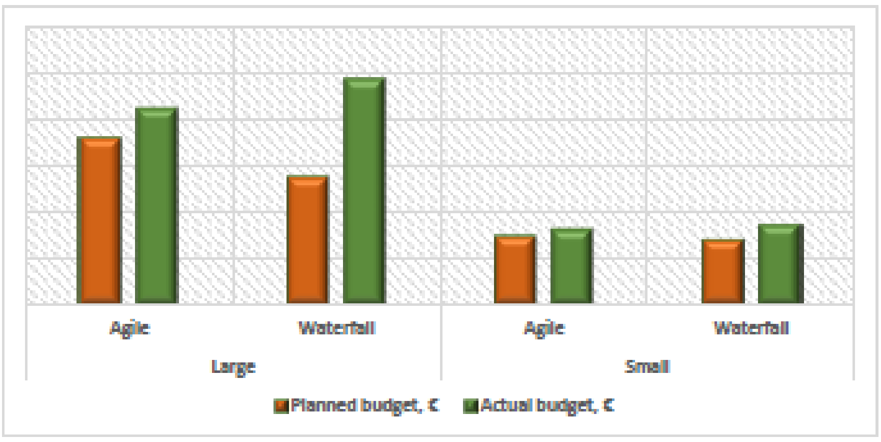


Figure 13 Deviation from the planned budget for Agile and Waterfall [25]

3 Transition to Agile

3.1 Initial phase

There are various reasons to move to Agile. The reason may be both management's efforts to ensure rapid response to changing requirements and for example efforts to reduce risks and maximise project profits. For example, a project whose transition will be described in this work had a rather trivial cause. Project which was originally managed using a waterfall model became more and more complex over time. Therefore, some customer requirements have already become difficult to manage. For example, planning a large project was already very risky. It was already very clear that the project managed by WM would be economically disadvantageous and could not provide the required quality.

It is important to realise that there are a number of obstacles to implementing Scrum in a project. If we look at the issue of moving to Agile in general (whether it's Scrum, XP or any other method), then we can identify some of the most common reasons:

- Inconsistent processes and practices across teams
- Organisational culture at odds with agile values
- General organisation resistance to change

3.1.1 Set goals

In my opinion, one of the basic conditions for a successful transition to Agile is as follows. Management should have a clear idea of what they expect from the transition. Like any control technology, agile approaches have specific advantages and disadvantages. Therefore, before the beginning of transition, it is necessary to define specific goals and also make sure that the chosen management technique will help us to achieve these goals. Therefore, one of the steps in the initiation phase is to study the literature and compare individual methods.

Another important step should be a feasibility study. A rough transition plan should be drawn up on the basis of this study. This plan is very important for communication with the customer.

3.1.2 Communication with the customer

Once a rough migration plan for Agile has been created, this migration plan needs to be agreed by the customer. It is very important to realise that almost all Agile technicians have completely different access to collaboration with the customer. For example, according to the extreme programming methodology, the customer should provide the development team with a person who will work with the development team throughout the product development. We cannot expect that changes concerning the customer will be so small that he will not notice them. And that's why the customer needs to know what we want from him. It may happen that customers have their ideas about the transition or something may not suit him. That is why it is important to find a compromise solution. And also adjust the transition plan as needed.

This step can cause some complications in case a customer has no previous experience with Agile. In this case it is good to think about how to share some information with the customer to mitigate adoption of the new model.

3.1.3 Team preparation

Most agile techniques have a completely different view not only of communication with the customer but also of the development team structure. Therefore, it is important to organise the appropriate structure and provide the necessary training for the development team. Due to the lack of programmers on the labour market today, the best strategy will be to avoid changes in the development team and try to train existing employees so that they are able to adapt to the needs of the Agile environment.

One very specific feature of this step is that it is not firmly connected with the other steps. Staff training may take place outside the transition. In this case, it is important to make sure that all team members are fully trained.

3.2 Create a team responsible for the transition

During this stage a team who is responsible for the transition should be created. It is expected that the responsible team will decide which part of the project will adopt Agile first. Subsequently, this team shall create a detailed transition plan. And the team is also responsible for controlling all stages.

3.3 Demonstration of the plan to the customer, approval

Once a specific transition plan is created, it is important to present it to the customer. As I already mentioned it is necessary to take into account that the customer may want to implement some changes to the plan. At this stage, it is also important to check whether the customer has met all the requirements (made at the start of the adoption) entered so far .

3.4 Methodology adoption

This is a very important phase of the transition. In most cases, we anticipate that the Project will gradually, step by step, move to Agile. However, this transition is very specific to each technique and will therefore not be described in more detail in this chapter.

3.5 Conclusion

The following issues shall be discussed during the final stage:

What went successfully?

What went wrong?

Was the transition successful?

What's next?

4 Practical part

4.1 Description of RNP system

North Sea Link is the first submarine cable between the UK and Norway with a capacity of up to 1400 MW. This system works successfully with the help of the North Sea Link Dispatch System. The main purpose of the Regional Nomination Platform (RNP) system is to provide functionalities (for example support allocation process) for HVDC Interconnector operators. One of these operators is National Grid which operates the NSDS (North Sea Dispatch Communications System) project.

The primary aims of the RNP system can be perceived in the following areas:

Regulate the allocation processes in the external systems and operate the capacities as allocated in external systems. Also, this system should support the multi-interconnector concept.

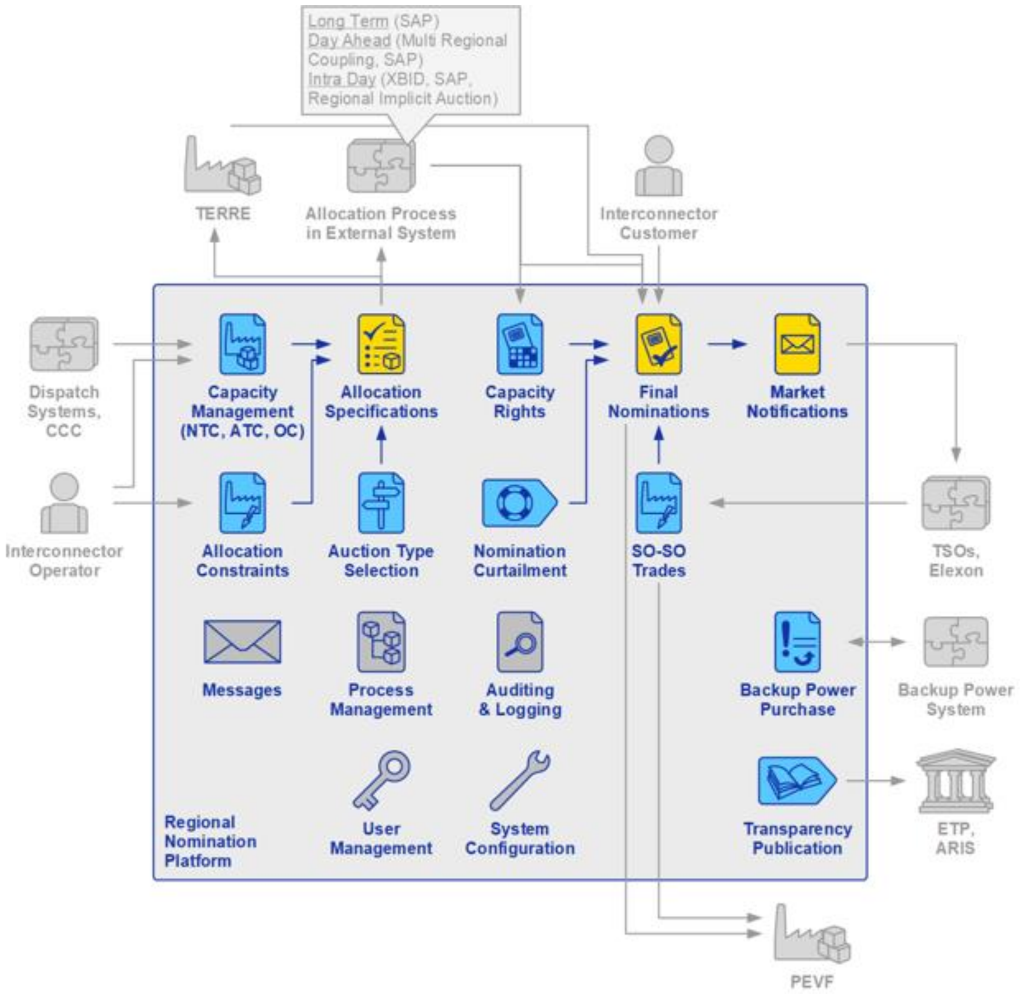


Figure 14 High-level design of RNP [26]

At Figure 14 you can see the High-level concept of RNP.

The following outline describes in introductory terms the general scope of the RNP system.

The business functionalities of the system are represented by the following components:

- Capacity Management – The System receives the basic input, and calculates the ATC (Available Transfer Capacity) as the basis for the further allocation process. The ATC value can be manually modified if needed.
- Allocation Constraints – The allocation constraints, as defined by the individual Interconnector Operators, form an important part of the allocation specification – i.e., actually driving the set-up of the respective allocation process.
- Auction Type Selection – The system supports multiple means of allocation process, including a parallel run of multiple allocation methods. The RNP system performs as a "switchboard" for selection of the respective auction type, further triggering the respective relevant processes and activities within the system.
- Capacity Rights – Depending on the carried-out allocation process (observing the available and permissible modes for the individual timescales), the System receives the allocation results either in a form of capacity rights, or Results values.
- Nominations Curtailment – In case of unexpected operational circumstances, the nomination curtailment handles the reduction of nominations to reflect an unplanned outage.
- SO-SO Trades (System operator- System operator trades), as agreed between the respective Transmission System Operators can be entered into the System, and are further reflected into the final nomination values.

4.2 Requirements

As already mentioned, it is necessary to precisely define the goals before starting the Transition. Subsequently, we need to make sure that the transition to Agile will really help us to achieve these goals. In most cases, the transition is conditioned by the fact that the waterfall model is not able to ensure the required quality.

In most cases, it is possible to define several common benefits that are expected from the Agile adoption.

- More stable team – the team continuously works on the Backlog

- Higher flexibility regarding to changes – ability to break down huge changes to small pieces
- Higher team performance – the team is staying focused, there is a queue with Stories
- Higher quality – testing is performed during the sprint, smaller pieces of changes are preferred
- Better and transparent prioritisation – everyone can add/remove User Story
- More transparent planning – knowledge about what was implemented in last Sprint, we know the quality, we plan next Sprint

If these goals cannot be achieved within the framework of acceptances, then the transition to Agile is unnecessary.

4.3 Dual track

A detailed model describing the original Waterfall model can be seen in Figure 15. The model that should arise as a result of the transition to Agile can be seen in Figure 16. This system is called dual track (DT). It is a hybrid model that is partly Agile and partly traditional.

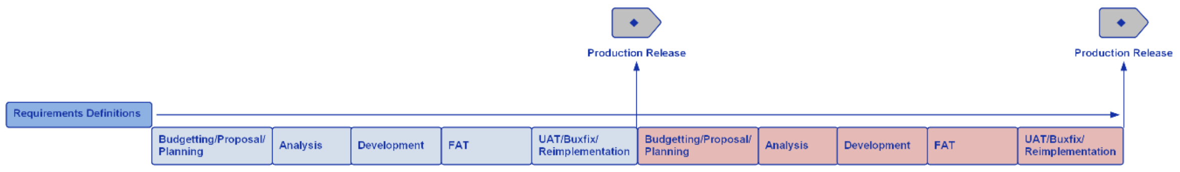

Production Release
Production Release
Requirements Definitions
Budgetting/Proposal/ Planning
Analysis
Development
FAT
UAT/Buxfix/ Reimplementation
Budgetting/Proposal/ Planning
Analysis
Development
FAT
UAT/Buxfix/ Reimplementation

Figure 15 Current state of the RNP [26]

Currently, the project is managed using the Waterfall Model. As it can be seen in Fig. 15, the project gradually goes through the following phases:

- Proposal
- Analysis
- Development
- Factory Acceptance test (FAT)
- User Acceptance test (UAT)

As already mentioned, dual track is a combination of Agile and Traditional approach to project management. The basic idea of dual track is that the project should meet two conflicting requirements. The first requirement is that the project be Agile for requirement definition, analysis, development and testing. The second requirement is that release management and UAT should remain managed traditionally.

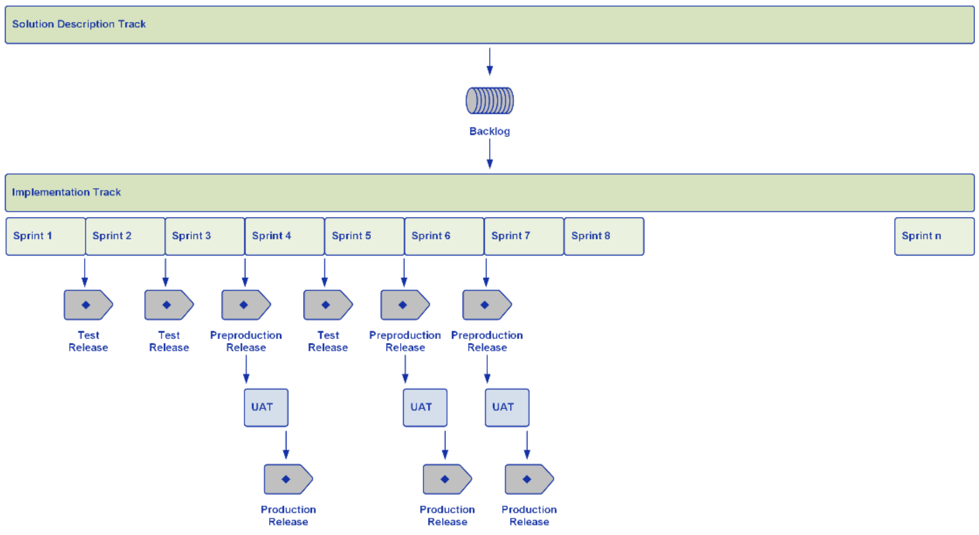


Figure 16 Dual track [26]

For the purpose of effective management, the project is divided into two tracks (see Fig. 16). Requirements definition, US creation and prioritisation should take place in the Solution Description Track. The development of the project itself should take place in the Implementation Track. This development should be managed according to Scrum (taking place in sprints). A closer look at the Implementation Track can be seen in Figure 17.

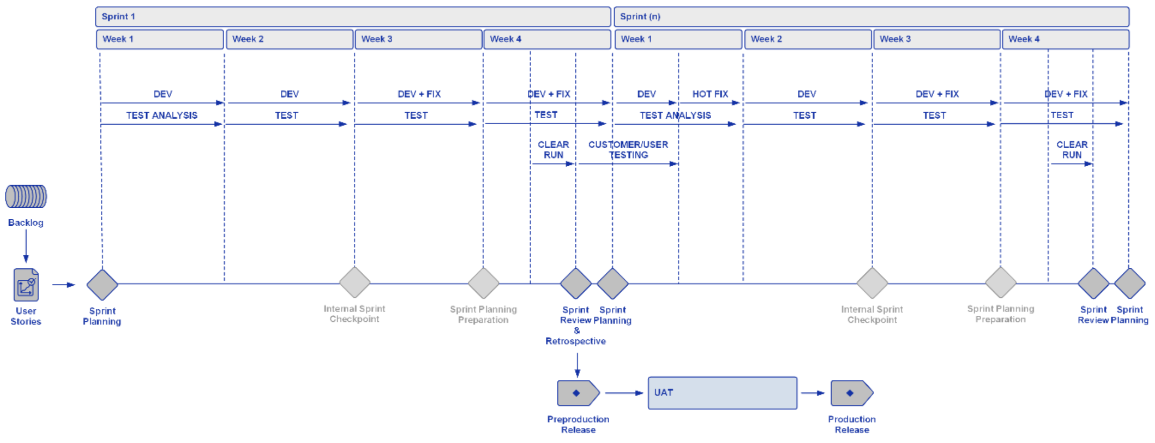


Figure 17 Scrum planning during the Dual Track [26]

Fig. 17 shows the diagram of the Implementation track under the condition that the sprint will take place. The development procedure is as follows.

The development process is divided into two parallel paths. Development should take place as part of the first journey. If a bug is detected during development a user story shall be created. Which means that the fix will not run immediately. Therefore, according to DT, fixing bugs should take place in the second half of the sprint. As part of the second trip, testing should take place. Obviously, the analysis of the test should take place during the first week. This is based on the need to prepare "solid ground" for testing. First, not enough US will be completed in the first week. And that's why the team can't start testing right away, otherwise they would have to wait until the US is completed.

A clear run should take place at the end of the last week. After the Clear Run, customer testing should begin, which should end during the first week of the next sprint.

Another important thing that should take place over the last day of each sprint is Sprint Retrospective and Sprint Review. After these meetings, the process of releasing a new version of the software should start. This process is special and it should actively involve the customer. The whole process consists of three points: pre production release, User testing, production release. User testing should be attended mainly by the Key User, who should be the person who should be the target user for the software (analogue of User in XP).

4.4 Rough transition plan

Based on information obtained during the implementation of Agile in some projects, it can be expected that transition should take place in three time intervals.

The first interval should run from the acceptance of the project until the beginning of the first Sprint. Most organisational and procedural changes should be made during this interval. The most important step in this interval is to create a Solution Description Track. The second interval should take place in the first and second sprints. Only a limited number (1-2) of development teams should be managed by Agile. Remaining teams should be managed traditionally.. It is assumed that the transformation should take place gradually during the first two time intervals. I.e., the first development teams at the end of the second sprint should be managed according to the dual track model. In the third interval, all remaining teams should switch to dual track.

4.4.1 Initial phase

First of all, it is important to realise what is the cause of the transition to Agile in company. The RNP project is not the first project that would like to adopt Agile. In most cases, the need to adopt Agile arose in response to customer requirements. In this case the project management should agree with the customer that the required quality will be met, but there is a need to manage the project in the Agile way.. As this section is not covered by the checklist, this chapter will not be discussed in detail.

4.5 Development team preparation

As already mentioned, the preparation of the team does not have to take place only during the transition to Agile. Today there is a widely used practice for a company to invest in its employees. Free training is one of the most common benefits that companies offer to their future employees today. This benefit can be used to familiarise your employees with the Agile environment.

This principle also applies to Unicorn. Employees are trained on an ongoing basis before the need arises in the transition. Therefore, the whole step is shortened to the fact that the development team should be informed about the future transition. How to find the Scrum Master is solved in the same way (SM must be trained). For this reason, there will be no points in the checklist regarding staff training.

4.6 Organisational changes

As already mentioned, the changes should concern not only at the side of the project approach of the specific project that is moving to Agile but also at the customer's side.

The most important member of the team that the development team should work on is the Product Owner. The term Business PO is used in company documentation. Since PO is very actively involved in the creation of the Solution Description Track, this role should be filled at the beginning of the first time interval.

Another important role is the Technical Consultant. The term Technical Product Owner is used in company documentation. It is obvious the main responsibility of this team member is to consult on all the technical details of the project. This role should also be cast at the beginning of the first phase.

To successfully start the first sprint, the role of Key User should also be filled. This is an employee who is very actively involved in User Acceptance Tests (UAT). If in the project his role is limited only by the acceptance of UAT, then this role should be filed no later than 3 weeks after the start of the first sprint (see Fig. 17). However, it may happen that KU should be an opponent in the design. In this case, this role needs to be filled at the beginning of the first time interval.

If all the preparatory steps described in chapter 3.1.3 are taken, the organisational changes within the organisation should be minimal. The main change should be the creation of self-organising teams. These teams should include everything needed to develop the role: analyst, tester, developer… 1-2 Scrum teams should be prepared before the start of the first sprint. During the first two sprints (the second time interval), these teams should gradually move to the Agile Development Principle.

An Agile Dashboard should be created in parallel.

4.7 Solution Description Track creation

One of the most important process changes is the creation of the Solution Description Track, which includes the creation of a new system for entering requirements, their transformation into US and prioritisation. This track should run in parallel with the Implementation Track. A very important point that connects these two tracks is the beginning of each sprint. It is important that a complete Sprint Backlog is created before the sprint begins.

The most important difference between traditional methods and Agile is that the requirements should not arise at the beginning of the project, but should be formulated continuously throughout the project. At the beginning, you need to create a backlog of the first Sprint..

4.7.1 US, DOD creation

Subsequently, each Epic should be divided into individual User Stores. When creating a US, it is important to follow certain rules. These rules are described in more detail in chap. 1.7.1. It is important to realise that the PO is aware of this difference.

All User Stories are subject of Backlog Refinement: prioritisation, evaluation if the User Story is ready to be implemented. Appropriate methods such as planning poker should be used for planning. As a result, the number of days required to process one US can be expected to be derived from the Fibonacci sequence. Sometimes it is not necessary to use the number of days as the main criterion. User Points may be used.

In addition to the standard information, User Stories should include information on the conditions under which it will be considered completed successfully. The waterfall model defines an easy way to evaluate the success of tasks. The number of errors found during testing is evaluated. If the resulting value does not exceed a certain value, the project is successful. The agile approach has a completely different approach. A DOD (definition of done) needs to be created for each User Stories. The definition of done can contain criteria such as:

- Is it accepted by customers?
- Is it tested?
- Is it documented?

In addition, each US must be designed so that it can be implemented independently of other USs.

Once the project backlog is created you need to create a backlog for the first two sprints. It should be taken into account that it is not possible to start working in a new environment immediately.

4.7.2 Budget of Epic

One important reason to create an Epic is that a budget project (or Epic's budget) should be made before the start of the first Sprint. According to the Scrum methodology, it should be planned for each Sprint separately and only before the start of the respective sprint. In the case of dual track, the customer needs to have at least a rough idea of the costs before starting the project. Obviously, such a budget is only indicative. As a result, instead of planning the budgets of individual sprints, only the budget specification will take place. It is obvious that when creating a budget, it is also necessary to plan the time and all the necessary resources.

4.8 Documentation

The Scrum methodology defines that the documentation should be as concise as possible and, if possible, shorter. The creation of large documents should be avoided. However, according to DT, this solution is unsatisfactory. For large projects, trying to reduce documentation can be too risky. In addition, if we have extensive documentation, we can save a lot of time on training a new team member.

Therefore, the documentation should not change when switching to dual track. What should change is the approach to creating the documentation itself. A document describing the project should be created at the beginning of the project. Unlike a traditional document, a newly created document should not contain a complete description of the entire project. The description of the project should be gradually supplemented during each sprint.

4.9 Bug fix planning

Another integral part of the transition is allocation of resources, which should be used to correct errors that may occur during the transition. Time and financial reserves should be allocated. In addition, you need to plan what will happen to these resources if no errors occur. Making time reserve means that it is needed to add some reserve when team velocity is planned.

4.10 First Sprint planning

Another important step is planning the first sprint. A Sprint Planning meeting, Sprint Review, Sprint Retrospective, etc. should be scheduled. The Scrum master should also prepare and organise the Daily Scrum.

Each sprint should end:

Test Release (to non UAT environment called “Control Gate Environment”)

the scope of the sprint is provided to the Customer, who can evaluate the functionality

this kind of release is not planned to be released to Production nor UAT

Pre Production Release (to UAT environment) –

the scope of the sprint (and previous sprints) is a subject of Production Release

such release is ready for full User Acceptance Testing and Integration Testing

this version is planned to be released to Production Environment

These releases should be matched at least 6 weeks before the start of the second stage (beginning of the first sprint).

4.11 Checklist

1st stage	Organisational changes on the customer's side
	• Product owner (occupy the role, train) • Key User (occupy the role, train) • Technical consultant (occupy the role, train)
	Prepare the first development teams
	Organisational changes in the company
	• Scrum master (occupy the role, train)
	Agile dashboard
	Create Solution description track
2nd stage	Create Backlog
	• Create epic • Create User stories • Create DOD • Prioritise US • Create budget of epic
	Create a basic document
	Planning of eliminating errors
	• Allocate resources to eliminate errors • Create a plan in case resources have not been used
3rd stage	First sprint planning
	Sprint planning meeting Daily Scrum Sprint review Scrum Retrospective
After 2nd sprint	Prepare other development teams

Conclusion

As a part of my diploma thesis I made a description of some control models: Waterfall model, V-model, PRINCE II, Scrum. I listed strengths and weaknesses of all the models. I then compared the Agile and traditional management methodologies. I also evaluated the impact of project size on the success of traditional and Agile methodologies.

The main advantages of traditional techniques are structure and simplicity. In addition, according to these methodologies, a large amount of time and effort should be spent during the first phases of the projects. Which helps eliminate a lot of bugs right from the start. In addition, due to detailed documentation, training new employees is very easy.

The great advantage of Agile Techniques is the delivery of the result not at the end of the projects but continuously. As a result, according to the Agile approach the development team receives regular feedback from the customer. It is also important that there is earlier testing.

The size of projects in particular has a great influence on the success of projects. If we talk about small and medium-sized projects, the success of the agile and waterfall model is comparable. However, for large projects, management using the Waterfall Model is very risky, very often there may be delays.

In the practical part I focus on describing the transition of the RNP system from the traditional management model to Agile. I obtained information from internal documentation and consultations with Unicorn. First, I made a brief description of the RNP system and its original management system. Subsequently, the target management model was described. This model, called dual track, is a synthesis of the traditional management model and the Agile approach. The basic principle of the target system is to manage the project traditionally in User Acceptance testing and release management and be agile during development requiring definition and analysis. Then I created a description of the transition from the original to the target state. Subsequently, a checklist was created that describes the transition to dual track. Dual track is the division of the development process into two tracks: Implementation track and track in which the creation of new requirements is performed.

The main problems I encountered during the transition are as follows:

Creating a development team. Scrum's methodology looks at the structure of the development team and its approach to work. Due to the fact that this is a different "philosophy", an obstacle for a team member can be not only lack of knowledge but also the inability to work in a new environment. As an example, I can mention the need for closer cooperation with the customer and the need for self organisation.

A very important problem with the transition described in this work is that the target system is not pure Scrum. First, it means that the apartment management should be able to manage the project both traditionally and Agile. Second, when implementing, it is important to have vision and know whether implementation of Agile approach in a particular part of the project will help to achieve the set goals, or vice versa. The documentation is an example. According to the basic description of the dual track, it can be expected that it will follow Agile when working with documentation. However, such an approach might lead to unnecessary complications. That is why during the whole project we work with classic "waterfall" documents.

In addition, according to the transition plan, most problems should be identified during the first two sprints. As part of these sprints, there is some "testing" of the new system. Therefore, it is necessary to take into account that the speed of development team work at this stage may be less. In addition, resources need to be allocated that can be used if development teams are unable to develop a Sprint Backlog in a timely manner. It is also necessary to take into account that if there is no delay, it is necessary to decide how to use these allocated resources.

LIST OF THE BIBLIOGRAPHIC REFERENCES

[1] KUIDEEP, Rana, 2019. Waterfall Model in Software Engineering [online]. [cit. 19.3.2022] Accessed: https://artoftesting.com/waterfall-model.

[2]] ROYCE, Winston, 1970. Managing the Development of Large Software Systems. Proceedings of IEEE WESCON [online]. [cit. 19.3.2022]. Accessed: https://leadinganswers.typepad.com/leading_answers/files/original_waterfall_paper_winston_royce.pdf/

[3] DOD-STD-2167A, MILITARY STANDARD: DEFENSE SYSTEM SOFTWARE DEVELOPMENT [online], 1988. [cit. 19.3.2022]. Accessed: http://everyspec.com/DoD/DoD-STD/DOD-STD-2167A_8470/

[4] "MIL-STD-498, MILITARY STANDARD: SOFTWARE DEVELOPMENT AND DOCUMENTATION [online], 1994. [cit. 19.3.2022]. Accessed: http://everyspec.com/MIL-STD/MIL-STD-0300-0499/MIL-STD-498_25500/

[5] KNIBERG, Henric, 2010. Toyota's journey from Waterfall to Lean software development. [online]. [cit. 19.3.2022]. Accessed: https://blog.crisp.se/2010/03/16/henrikkniberg/1268757660000/

[6] Does the V-Model have a verification and validation stage for every stage, [online]. 2019.[cit. 19.3.2022]. Accessed: https://softwareengineering.stackexchange.com/questions/394860/does-the-v-model-have-a-verification-and-validation-stage-for-every-stage/

[7] The History of PRINE2|EUR, [online]. 2017.[cit. 19.3.2022]. Accessed:

https://www.prince2.com/eur/blog/the-history-of-prince2/

[8] PRINCE2 Methodology Overview: History, Definition & Meaning, Benefits, Certification, [online]. 2011.[cit. 19.3.2022]. Accessed: view-source: https://mymanagementguide.com/prince2-methodology-overview-history-definition-meaning-benefits-certification/

[9] what-is-prince2, [online]. 2020.[cit. 19.3.2022]. Accessed: https://www.qrpinternational.ch/en/qrp-news/what-is-prince2/

[10] MEYER, David, 2007.The Blair IT projects [online]. [cit. 19.3.2022]. Accessed: https://www.qrpinternational.ch/en/qrp-news/what-is-prince2/

[11] KING, Anthony; CREWE, Ivor. 2013 The Blunders of our Governments. ISBN 978-1780742663.

[12] The costly trail of British government IT and 'big bang' project disasters [online], 2014. [cit. 19.3.2022]. Accessed: https://www.theguardian.com/technology/2014/aug/19/costly-trail-british-government-it-disasters-universal-credit/

[13] The Agile Manifesto, 2001. Manifesto for Agile Software Development[online]. [cit. 19.3.2022]. Accessed: https://agilemanifesto.org/

[14] The 15th annual State of Agile Report, 2022. [online]. [cit. 19.3.2022]. Accessed: https://digital.ai/resource-center/analyst-reports/state-of-agile-report#ufh-c-473508-state-of-agile-report/

[15] The Agile – Scrum Framework, 2021. [online]. [cit. 19.3.2022]. Accessed: https://medium.com/@muhammad.zahran/clicks-with-agile-and-scrum-38d61c912475/

[16] User Story, 2022. [online]. [cit. 19.3.2022]. Accessed: https://www.productplan.com/glossary/user-story/

[17] The 15th annual State of Agile Report, 2022. [online]. [cit. 19.3.2022]. Accessed: https://cs.wikipedia.org/wiki/Burndown_chart/

[18] Know Your Role [Infographic], 2017. [online]. [cit. 19.3.2022]. Accessed: https://medium.com/generation-agile/know-your-role-infographic-28403ae230e6/

[19] The 2020 Scrum Guide, 2020. [online]. [cit. 19.3.2022]. Accessed: https://scrumguides.org/scrum-guide.html/

[20] Scrum checklist, 2022. [online]. [cit. 19.3.2022]. Accessed: https://ru.itpedia.nl/2019/04/06/de-dagelijkse-scrum-checklist/

[21] Scrum black book, 2018. [online]. [cit. 19.3.2022]. Accessed: https://infostart.ru/1c/articles/949714/

[22] S. Ambler. [2018]. IT Project Success Rates Survey Results. [online]. [cit. 19.3.2022]. Accessed: http://www.ambysoft.com/surveys/success2018.html/

[23] U. Telemaco, T. Oliveira, P. Alencar and D. Cowan, "A Catalogue of Agile Smells for Agility Assessment," [online]. [cit. 19.3.2022]. Accessed: in IEEE Access, vol. 8, pp. 79239-79259, 2020, doi: 10.1109/ACCESS.2020.2989106.

[24] CHAOS MANIFESTO [2017]. [online]. [cit. 19.3.2022]. Accessed: https://www.immagic.com/eLibrary/ARCHIVES/GENERAL/GENREF/S130301C.pdf/

[25] Bormane, Līga & Gržibovska, Jūlija & Bērziša, Solvita & Grabis, Janis. [2016]. Impact of Requirements Elicitation Processes on Success of Information System Development

Projects. Information Technology and Management Science. 19. 10.1515/itms-2016-0012

[26] Unicorn internal documents