UCAM
UNIVERSIDAD CATÓLICA
DE MURCIA

ESCUELA INTERNACIONAL DE DOCTORADO
Programa de Doctorado en Ciencias Sociales

"Methods of Project Management for Lean Development of New Products"

Autor:
Nils Engelhardt

Directores:
Prof. Dra. Mercedes Carmona Martínez
Prof. Dr. Rudolf Jerrentrup

Murcia, Septiembre de 2019

This page is intentionally left blank

UCAM

UNIVERSIDAD CATÓLICA
DE MURCIA

AUTHORIZATION OF THE DIRECTOR OF THE THESIS FOR SUBMISSION

Prof. Dra. Mercedes Carmona Martínez and Prof. Dr. Rudolf Jerrentrup as Directors (1) of the Doctoral Thesis "Methods of Project Management for Lean Development of New Products" by Nils Engelhardt in the Departamento de Ciencias Sociales, Jurídicas y de la Empresa, **authorizes for submission since it has the conditions necessary for his defense**.

Sign, to comply with the Royal Decrees 99/2011, 1393/2007, 56/2005 y 778/98, in Murcia

Prof. Dra. Mercedes Carmona Martínez   Prof. Dr. Rudolf Jerrentrup

(1) IF THE THESIS IS DIRECTED BY MORE THAN ONE DIRECTOR MUST NOTED AND SIGN BOTH

This page is intentionally left blank

ACKNOWLEDGEMENTS

First of all I would like to thank Prof. Dra. Mercedes Carmona Martínez and Prof. Dr. Rudolf Jerrentrup, who guided my work. Their advice regarding the structure of the thesis, the right justifications and conclusions as well as the formal aspects were very helpful for me.

I would also like to thank Prof. Dra. Mercedes Carmona Martínez and Prof. Dr. Alexander Zureck for organizing the doctoral program and the many events in Murcia and Essen and for the great help they gave me with answering my questions.

Next I would like to thank Prof. Dr. Wolfgang Hoheisel and Prof. Dr. Stephan Staudacher for their inspiration to have even started the project of a doctoral thesis and to dedicate myself to the topic of project management.

I would not have been able to finish this work if I had not been so wonderfully supported by my family. So I thank my wife Elke for making it possible for me to sacrifice many hours of our time in common, often putting her needs to the back. And I thank my children Hanna and Karlotta for their patience and understanding that their dad could not always have time for them but also for ensuring that I've stood with both feet on the ground in my life.

Finally, I would like to thank all my friends and colleagues who have always believed that I could successfully complete this project.

I would like to dedicate this work to my parents Dirk and Gerburg Engelhardt. Unfortunately, they cannot witness how their son receives his Ph.D. because they both died last year. I thank you for everything you have done for me during my whole life!

Karlsruhe, 1 August 2019
Nils Engelhardt

This page is intentionally left blank

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

APM	Agile Project Managemant
B2B	Business to Business
DIN	Deutsche Industrienorm (German industry standard)
DOI	Declaration of Interdependence
FBS	Feature Breakdown Structure
ICB	IPMA Competence Baseline
IPMA	International Project Management Association
ISO	International Organization for Standardization
NPD	New Product Development
PD	Product Development
PDS	Product Development System
PERT	Program Evaluation and Review Technique
PM	Project Management
PMBoK	Project Management Body of Knowledge
PRINCE	Projects in Controlled Environments
R&D	Research and Development
TPM	Traditional Project Management
TPS	Toyota's Production System
VBS	Value Breakdown Structure
WBS	Work Breakdown Structure
WIP	Work in Progress

1 INTRODUCTION

Implementation of "lean principles" has been established for a number of years in the production environment. A company's production is geared towards the customer through improving value creation, which means avoiding waste comprises the goal of "lean production."

In the last few years, optimization's focus has expanded to all other areas of a company, with not only the production being considered from a lean perspective. Lean development, for instance, transfers the successful management concept of lean production into the development and integrated product development process.

In addition, different methods and models of project management have evolved in recent years. Beyond the "classic," "traditional" project management (waterfall model, V-model, etc.), the methods of "agile" project management (Scrum, Lean-Kanban, etc.) in particular have increasingly been studied and described nowadays.

This thesis links both topics—lean development and project management methodologies. It further analyzes the different project management approaches for suitability for lean product development in order to determine which approach leads to more successful projects.

1.1 BACKGROUND AND MOTIVATION

Working for years in project management in the automotive industry, the author's daily business consisted of chasing progress of the development projects. A common approach to this corresponded to the waterfall or V-model (depending on the project manager). A great deal of time was spent with planning, but also with "red tape" and bureaucracy in the projects. In this sector, every company is more or less forced to avoid missed deadlines—resulting in numerous (expensive) firefighting actions. However, there must be a better way of performing project management, better planning models and more reliable product development execution.

As head of the manufacturing engineering department at another company in the automation sector, the author has invested considerable time into learning and applying the lean methodology, not only in production, but also in administrative processes. In the beginning, lean development represented a significant subject matter for this company. This also applies to the company in which the author is now employed, which is active in the life science sector.

In recent years, the author has grown increasingly interested in managing development projects the “lean” way. In other words, he sought to determine how the established lean production methods can be successfully transferred to development projects.

Since the beginning of this century, the methods and models of agile project management (APM) have spread more and more, albeit mostly within the context of software development.

Considering and analyzing both approaches, this thesis offers a scientific contribution through a comparison of the "classic" against the "agile" project management from the "lean" perspective. This enables coming closer to answering the above question—namely, how projects can be managed “better," or more successfully.

1.2 RESEARCH GOAL

This thesis examines which project management method—classic or agile—best reflects the lean product development principles. For this study, various boundary conditions and their effects are thus considered. These include, for example, maturation of the development department (both organizational and employee, i.e., experience and error rate) or realization of the customer interaction (e.g., frequency of change requests). Furthermore, the nature of the project can be of vital importance for determining which method produces better results.

All questions concerning the introduction of a lean culture in a company remain beyond the scope of this study, even if, especially in the more creative areas of product development, the human factor cannot be underestimated. Certain boundary conditions are also assumed as a given or are examined only concerning their impact, but no questions regarding their achievement or maintenance are discussed.

The process and time frame from the beginning of the development (concept phase) to the start of a product's production are analyzed, meaning the "real" development. Furthermore, idea generation and project selection processes, as well as product maintenance in later stages of the product life cycle, influence the development area (especially its staff capacity, but also the formation of knowledge and learning). However, these also exceed this investigation's scope.

1.3 RESEARCH METHOD

Due to the long duration of a typical development project, especially of a product in the area under consideration, it is not possible to analyze an ongoing or recently started project. On the other hand, it is also difficult to assess afterwards how a project would have developed under a different project management approach. Instead, simulations are performed that consider both the project management methods and the various constraints. This allows examining whether different project management methods for different parameters of the boundary conditions would best implement the lean methodology. The simulation covers the development of a product consisting of mechanical, hardware and software components. Evaluation of the results and the corresponding figures is based on the success criteria harmonized between the project management approaches.

1.4 STRUCTURE OF THE DISSERTATION

Following the introduction in chapter one, chapters two through five discuss the literature on traditional project management (TPM), APM, lean development and the success of a project or project management.

Chapter six derives hypotheses from the study of literature. Chapter seven describes the simulation employed to compare the various project management approaches. The project's framework conditions, as well as the implementation of the different project management (PM) methods in the models, are described. The mechanisms utilized to determine the learning effects and rejection rates are also discussed. Finally, the simulation's output data is analyzed and statistically examined. Chapter eight interprets and discusses the results.

2 TRADITIONAL PROJECT MANAGEMENT

2.1 THE HISTORY AND IMPORTANCE OF PROJECT MANAGEMENT

In the history of mankind, countless achievements have been made that, from today's point of view, would be understood as a project. The pyramids in Egypt or other important buildings, as well as shipbuilding, have been identified as examples of such.

> *"While projects were conducted by the Pharaohs, under Vauban or by the Vikings, there was no management model at that point. It was only in the second half of the 20th century that project management broke away from other forms of activity to be identified, highlighted and generalized in and of itself. It has now become a management model"* (Garel, 2013, p. 665).

Garel (2013) stated that project management was employed starting in the 1930s, but no management model was yet available. It was not until the engineering projects conducted at the end of the 1950s that standardized tools, practices and roles were developed, leading to the emergence of a true model.

Kerzner (2009) identified three major phases in changing the focus on project management beginning in the 20th century, with the current phase beginning in the 1990s: "By the 1990s, companies had begun to realize that implementing project management was a necessity, not a choice. The question was not how to implement project management, but how fast could it be done" (Kerzner, 2009, p. 45).

Regarding the need for project management, especially for companies involved heavily in R&D, he claimed that, assuming that only a small percentage of R&D projects ever regain their cost, project management constitutes a necessity. Project management can also be employed to cancel a project in time (Kerzner, 2009, p. 47).

Based on the driving forces behind implementing project management, Kerzner (2009) described numerous benefits of project management. These include better profitability, increased efficiency and effectiveness, better customer cooperation and improved quality.

Garel (2013) examined the history of project management models. In so doing, he listed four management models (the entrepreneurial model, the engineering model, the Taylorist model and the concurrent engineering model) that have developed successfully over time, with one model not necessarily replacing another. Garel further distinguished between "managerial practices" and "management models." He stated that a management model involves more than just a management technique. Project management possesses a multidisciplinary character in contrast to functional hierarchies. It therefore goes far beyond the technical management of deadlines and costs (Garel, 2013).

2.2 THEORY ON NEW PRODUCT DEVELOPMENT (NPD)

This research focuses on R&D projects—more precisely, the development of a more complex product consisting of both hardware and software components. Therefore, the theory of new product development (NPD) needs to be reflected. As previously described, "New product development is a multi-stage process. Many different models with a varying number of stages have been proposed in the literature" (Murthy, Rausand, & Østeras, 2008, p. 15) (see "2.4.5 Stage-gate process and project life ").

According to Murthy et al. (2008), the NPD process covers the activities that companies undertake when developing and introducing new products. A newly launched product develops in a series of steps, starting with an idea or initial product concept that is evaluated, developed, tested and brought to market.

Bhuiyan (2011) proposed the following stages for an NPD model:

- “New Product Strategy: Links the NPD process to company objectives and provides focus for idea/concept generation and guidelines for establishing screening criteria.
- “Idea generation: Searches for product ideas that meet company objectives.
- “Screening: Comprises an initial analysis to determine which ideas are pertinent and merit more detailed study.
- “Business Analysis: Further evaluates the ideas on the basis of quantitative factors, such as profits, return-on-investment, and sales volume.
- “Development: Turns an idea on paper into a product that is demonstrable and producible.
- “Testing: Conducts commercial experiments necessary to verify earlier business judgments.
- “Commercialization: Launches products.” (Bhuiyan, 2011, p. 749)

This research concerns project management, so it includes the development and testing phases only. The commercialization phase needs consideration when the success of a project is discussed (see “5 Success of a Project”).

Regarding the question of what a "product" is in the first place, Murthy et al. discussed the terms “tangible” and “intangible”: In the classical meaning, products are physical and tangible. This contrasts with services that are intangible. However, this separation becomes blurred, as a product usually comprises a combination of tangible and intangible components (Murthy et al., 2008). The focus of this research is also on a product comprised of both tangible (mechanics and electronics) and intangible parts (software).

In order to understand the context of this work, the different classification possibilities of projects are described below. Murthy et al. (2008) distinguished the following:

Classification 1: Based on the type of consumer

First, certain products are purchased by households. The next group concerns off-the-shelf standard products utilized by other enterprises for their activities. The range of the products' complexity is quite wide, spanning from complete units (vehicles, machines) to individual components (spare parts, tools, electronic components).

Finally, specialized products (e.g., military technology, ships and power stations) are usually highly complex and expensive, representing state-of-the-art technology with high research and development costs for the manufacturers. Its customers usually consist of governments or industrial companies (p. 16).

This research focuses on consumer, industrial and commercial products (e.g., car navigation system or car sound systems).

Classification 2: Standard versus custom-built

Standard products are manufactured with a future demand in mind. Thus, the products are manufactured based on market studies. Standard products cover all consumer goods, as well as most industrial and commercial products.

Customer-specific products are manufactured at the customer's request and include special products (p. 16-17).

This research includes both aspects.

According to this categorization, Parsaei and Sullivan (1993) further differentiated the following classification.

Classification 3: Based on the nature of design and design process

Creative Design: Creative design describes an abstract decomposition of the design problem into a series of levels representing decisions for the problem. No fixed plan is developed for the problem.

Innovative Designs: The problem decomposition is known, but the alternatives for each of its subparts do not exist and must be worked out. The result can include a new combination of existing components. Innovative design further requires a certain amount of creativity.

Redesign: An existing design is adapted to reflect the necessary modifications to the original functionality demands.

Routine designs: The framework for the solution (i.e., the subparts and their alternatives) are known in advance. The aim is to identify alternatives for each part that meets the given constraints (p. 116-117).

Classification 3 strongly influences the product development difficulty, particularly due to the widely varying uncertainties associated with it. This could represent an important factor in selecting the appropriate project management approach or method.

Further classifications have been developed, but these are not relevant for the chosen focus.

The next question concerns what a *new* product describes. Murthy et al. (2008) stated that "the 'newness' of a new product can vary from high to low and depends on whose perspective and what is new" (p. 18).

In terms of perspective, they distinguished between "New to the world" (e.g., the first aircraft, radio, computer, car), "New to the industry" (first application in an industry of a product well established in some other industry), "New to the manufacturing firm" (and familiar to competitors in the industry), "New to the market" and "New to the customer" (p. 18).

Viewed in terms of what is new, the following possibilities are presented: "New technology (digital computer replacing analogue computer), New process (which reduces the production cost and/or increases the quality of conformance), New features (this is most dramatic in consumer electronics, such as, cellular phones), New uses (chips designed for computers being used in domestic appliances), [and] New design (which reduces the production cost)" (p. 18).

Regarding the degree of newness, this offers an "indicator of the difference between the new product and the existing one" (Murthy et al., 2008, p. 18).

The "newness" relates to the product classification based on the design nature. Therefore, it can be assumed that products possessing a major or radical change lead to greater uncertainty regarding the product itself, in turn leading to a more complex development process. This should considerably influence project management approach or methodology, too.

Pons (2008) examined the intersection of the PMBoK Guide (Project Management Institute, 2013) with new product development (NPD):

> *"There is a need for project management methods that can handle NPD. The problem of course is that some forms of NPD, especially those involving a high degree of innovation, are notoriously difficult to manage. Many NPD projects use project management tools, at least elements thereof. However, these are not always fully satisfactory, and the formal project management method in its entirety is not a complete solution for the management of all NPD projects. … Conventional project management requires relatively complete initial definition of outcomes and scope, which can be problematic for NPD: the uncertain outcomes cause the scope of work to be dynamic. This is especially the case when research is involved." (p. 82)*

Shenhar and Dvir (2007, p. 13) introduced another framework to analyze projects. In it, the following aspects determine the project characteristics:

- "Novelty" – How intensely new are crucial aspects of the project?
- "Technology" – Where does the project exist on the scale from low-tech to super-high-tech?
- "Complexity" – How complicated are the product, the process and the project?
- "Pace" – How urgent is the work? Is the timing "normal, fast, time-critical or blitz"?

These points represent the basis for examining how different framework conditions affect projects in this thesis. As such, they are further elaborated below.

Novelty: New creations

Shenhar and Dvir (2007, chapter 4) distinguished between varying degrees of newness or novelty, identifying different project stages for the freeze of requirements: A "derivative product" offers a revision of a successful product in which only small details (dimensions, individual features) are changed. The requirements for these products are clear. It is often a matter of financially capitalizing on the product for a while longer. These projects are relatively low risk and provide little cause for concern. The requirements for the new product should be frozen at the beginning of the project.

A new version of an existing product comprises a "platform novelty." Such projects require a high degree of analysis and market research. Product requirements usually continue changing until the middle of the project, as they adapt based on market research findings.

"Breakthrough products" face greater challenges, as they do not yet possess defined markets that the manufacturer can use. Market research is also less reliable, as potential customers are not yet familiar with the product and are unable to judge it. The product design is based on the developers' expertise. Trial and error represents an important part of the design and development process, and the project team must gather feedback from generations of prototypes.

Technology: Technical difficulty

Another source of uncertainty concerns technical difficulty. Here, too, they recommended shifting the requirements' definition over time as the degree increases: Uncertainty in a project stems from both the required technology and the knowledge available within the company. Novelty for the market and novelty for the enterprise are different sources, but ultimately lead to a certain degree of technological uncertainty.

Low-tech projects involve almost no technical risks, but require high efficiency to be economically viable in the end. In contrast, high-tech projects involve much greater risk, and are therefore more vulnerable to errors, cost overruns and deadline delays.

For low-tech projects, the requirements must be defined at an early stage to fully realize their efficiency, while high-tech projects should refrain from being frozen for as long as possible in order to benefit from learning effects during the project period. (Shenhar & Dvir, 2007, chapter 5).

Complexity: Measuring the complications

Shenhar and Dvir (2007, chapter 6) understood a higher degree of complexity as a greater interaction between the parts, requiring more formality in project management.

Pace: A sense of urgency

Last, they mentioned that some projects are more time-critical than others. The urgency, in turn, determines the project’s success. Of course, competition forces new products onto the market quickly, but other factors are often more important than the time schedule. However, time-critical projects also occur, wherein a delay is synonymous with the project’s failure, such as because a window of opportunity closed. The extreme case consists of crisis projects with extremely urgent timing (Shenhar & Dvir, 2007, chapter 7).

2.3 DEFINITION OF THE TERMS PROJECT AND PROJECT MANAGEMENT

In order to be able to examine different project management approaches, the term "project" should first be defined. To this end, different norms and guidelines define the term “project” as follows (see also (Dr. Angermeier, 2016)):

DIN 69901-5:2009: “Vorhaben, das im Wesentlichen durch Einmaligkeit der Bedingungen in ihrer Gesamtheit gekennzeichnet ist.” [A project is essentially characterized by uniqueness of the conditions as a whole] (*DIN 69901-5:2009-01 Project management—”Project management systems” Part 5: Concepts*, 2009)

PMBoK Guide: “A temporary endeavor undertaken to create a unique product, service or result” (Project Management Institute, 2013, p.3).

Competence Baseline 3.0 (IPMA): “A time- and cost-constrained operation to realize a set of defined deliverables (the scope to fulfil the project’s objectives) up to quality standards and requirements” (Gaupin & International Project Management Association, 2006, p. 13).

PRINCE2:2017: “A temporary organization that is created for the purpose of delivering one or more business products according to an agreed Business Case” (“PRINCE2” Official Website, 2009).

Lewis (2002) described a project as follows:

> *“a multitask job that has performance, time, cost, and scope requirements and that is done only one time. If it is repetitive, it's not a project. A project should have definite starting and ending points (time), a budget (cost), a clearly defined scope-or magnitude-of work to be done, and specific performance requirements that must be met.” (p. 2)*

Meanwhile, Kerzner (2009) stated the following:

> *“In order to understand project management, one must begin with the definition of a project. A project can be considered to be any series of activities and tasks that:*

- *“Have a specific objective to be completed within certain specifications*
- *“Have defined start and end dates*
- *“Have funding limits (if applicable)*
- *“Consume human and nonhuman resources (i.e., money, people, equipment)*
- *“Are multifunctional (i.e., cut across several functional lines)” (p. 2)*

Lester (2014) cited the British Standard BS 6079-2:2000 Project Management Vocabulary, which described a project as “A unique process, consisting of a set of coordinated and controlled activities with start and finish dates, undertaken to achieve an objective conforming to specific requirements, including constraints of time, cost and resources” (p. 1).

After the term "project" is defined, the term "project management" can also be described.

The distinction between project and project management is important to distinguish between project success and project management success (see "5.1 Project Success versus Project Management Success"). Accordingly, Munns and Bjeirmi (2015) provided the following definition:

> *"A project can be considered to be the achievement of a specific objective, which involves a series of activities and tasks which consume resources. It has to be completed within a set specification, having definite start and end dates. In contrast, project management can be defined as the process of controlling the achievement of the project objectives." (p. 81)*

Lester (2014) distinguished between project management and functional management:

> *"Project management is essentially management of change, while running a functional or ongoing business is managing a continuum or 'business-as-usual'. (…) There is a fundamental difference between project management and functional or line management where the purpose of management is to continue the ongoing operation with as little disruption (or change) as possible. This is reflected in the characteristics of the two types of managers. While the project manager thrives on and is proactive to change, the line manager is reactive to change and hates disruption." (p. 1-2)*

Kerzner (2009, p.3) referred to the PMBoK Guide (Project Management Institute, 2013): "Project management, on the other hand, involves five process groups as identified in the PMBoK Guide, namely:

- "Project initiation
- "Project planning
- "Project execution
- "Project monitoring and control
- "Project closure

"Successful project management can then be defined as having achieved the project objectives:

- "Within time
- "Within cost
- "At the desired performance/technology level
- "While utilizing the assigned resources effectively and efficiently
- "Accepted by the customer." (Kerzner, 2009, p.3)

The IPMA Competence Baseline stated the following about goal and objective:

> *"The project goal is to provide value to the interested parties (...). The project objective is to produce the agreed end results, especially the deliverables, in the time-frame required, within budget and within acceptable parameters of risk. The project objectives are the set of targets that the project, programme and portfolio managers should attain to provide the expected project benefits to the interested parties." (Gaupin & International Project Management Association, 2006, p. 44)*

For the discussion about interested parties, see "5.4 Stakeholders."

Unfortunately, project management's benefits cannot be achieved without overcoming obstacles such as the following (Kerzner, 2009):

- Project complexity
- Customer's special requirements and scope changes
- Organizational restructuring
- Project risks
- Changes in technology
- Forward planning and pricing (p. 4).

Kerzner (2009) summarized the following:

> *"The following would be an overview definition of project management: Project management is the planning, organizing, directing, and controlling of company resources for a relatively short-term objective that has been established to complete specific goals and objectives. Furthermore, project management utilizes the systems approach to management by having functional personnel (the vertical hierarchy) assigned to a specific project (the horizontal hierarchy)." (p. 4)*

The term "project" relates to the deliverable (which, in this research, means a tangible product with intangible content). Furthermore, "Projects exist to produce deliverables. (…) Deliverables are measurable, tangible outputs and can take such form as:

- "Hardware Deliverables: These are hardware items, such as a table, a prototype, or a piece of equipment.
- "Software Deliverables: These items are similar to hardware deliverables but are usually paper products, such as reports, studies, handouts, or documentation. Some companies do not differentiate between hardware and software deliverables.
- "Interim Deliverables: These items can be either hardware or software deliverables and progressively evolve as the project proceeds. An example might be a series of interim reports leading up to the final report." (Kerzner, 2009, p. 5-6)

The IPMA Competence Baseline provided the following definition:

> *"The deliverables of a successful project, programme or portfolio are tangible or intangible assets created by the project, programme or portfolio for the customer. They are represented by drawings, schematics, descriptions, models, prototypes, systems and products of various kinds. Deliverables are not only the product sold or service put into use after project closure, but also the operational processes, organizational changes and human resource changes needed for a successful organization to operate. The project deliverables may be classified in terms of their priority (must have; nice to have; if there is time), by agreement with the*

> *interested parties. Those of lower priority may not be delivered if there are time constraints." (Gaupin & International Project Management Association, 2006, p. 58)*

2.4 TRADITIONAL PROJECT MANAGEMENT (TPM) METHODOLOGY

In the PM literature, the terms "method," "framework" or "guideline" are not used uniformly. In this paper, TPM is understood as referring to methods that, in the form described in the following subchapters, determine the requirements at the beginning, then break down the project into individual work packages, which are then executed according to the planning. Methods that count among these include the waterfall method or the V-model. These are described in detail in the standards and guidelines cited so far.

Kerzner (2009) stated the following:

> *"Achieving project management excellence, or maturity, is more likely with a repetitive process that can be used on each and every project. This repetitive process is referred to as the project management methodology. ... During the 1990s, the following processes were integrated into a single methodology:*
>
> - *"Project Management: The basic principles of planning, scheduling, and controlling work*
> - *"Total Quality Management: The process of ensuring that the end result will meet the quality expectations of the customer*
> - *"Concurrent Engineering: The process of performing work in parallel rather than series in order to compress the schedule without incurring serious risks*
> - *"Scope Change Control: The process of controlling the configuration of the end result such that value added is provided to the customer*
> - *"Risk Management: The process of identifying, quantifying, and responding to the risks of the project without any material impact on the project's objectives." (p. 74-75)*

The following chapters discuss several aspects of project management in greater detail. It should be noted that the literature sources employed in the following subchapters should be viewed within the context of TPM. Therefore, the focal points of the statements and their content tendencies are to be assigned to this approach. Some statements are relativized following discussion of the agile approach.

2.4.1 Requirements management and change management

The IPMA Competence Baseline offers the following definition:

> *"Requirements management consists of the identification, definition and agreement of the project to meet the needs and expectations of interested parties, especially those of the customers and users. Project requirements are derived from customer needs, which are driven by opportunities and threats." (Gaupin & International Project Management Association, 2006, p. 44)*

Regarding requirements management, Lester (2014) wrote that different stakeholders possess their own requirements. All these requirements wield their respective effects on time, cost and performance. It is the project manager's job to reach agreement with stakeholders concerning the requirement's priority. From this point, "Once agreed, these requirements become the benchmark against which the success of the project is measured" (p. 18). In the event that a stakeholder subsequently wishes to change a requirement, the project manager must review the effect on costs, time and scope, as well as inform and obtain the approval of all other stakeholders. The new requirements are then subject to change management (Lester, 2014, p. 18).

Wysocki (2006, p. 479) emphasized the importance of requirements: In order to avoid project failure, the collection of requirements represents one of the first and most important milestones. This involves identifying what the customer really needs and questioning his assumptions. Later, he discussed the question of what should be done if one or both of those is not possible (see "3.4.1 Requirements management and change management").

Wysocki (2006, p. 484-485) listed three types of requirements:

- Functional requirements, which specify what the product or service must do
- Non-Functional requirements, or properties that the product or service should possess in order to do what it must
- Global requirements, which describe properties of the system as a whole.

Fulfilling these requirements can be seen as the quality of the project (IPMA (Gaupin & International Project Management Association, 2006)). This quality ensures long-term business success through customer satisfaction. Furthermore, "The intended functionality of the product should be validated during the course of the project. Normally, the customer or user will be involved in these reviews to ensure compliance with the product requirements" (Gaupin & International Project Management Association, 2006, p. 48).

Typically, the project is influenced by its environment and other uncertainties. Therefore, "Changes are often necessary in a project due to unanticipated occurrences. It may be necessary to change the project specification or the contract terms with suppliers or customers. Changes must be monitored against the original project goals and objectives" (Gaupin & International Project Management Association, 2006, p. 70). The IPMA competence baseline includes a special chapter concerning changes: A change management process should be agreed upon with all relevant stakeholders proactively at the beginning of a project, and not when the need arises. Change in project scope or specification is performed through a formal process. Such changes can be requested by any party, but both proposed and approved changes must be managed and properly communicated to all stakeholders. How these changes affect project content, time, costs and risks is determined through a comparison with the original plan. When the changes are accepted, the project plan is updated to reflect the changes (Gaupin & International Project Management Association, 2006).

Regarding change, Lester (2014) mentioned that it is common for projects to change during their execution. These changes usually affect time, cost and/or quality. Therefore, it is important to manage all changes to ensure that the originator acknowledges the impact so that, for instance, an extension of the original specification is compensated (p. 84).

Lewis (2002) also suggested making changes in an orderly fashion. Without change control, the project is likely to end above budget, behind schedule and hopelessly inadequate, but with no warning of such until it is too late. Change control is thus necessary to protect the project from the effects of uncoordinated extensions or scope changes that lead to additional work (p. 29).

Wysocki (2006) discussed management of changing requirements. Although the requirements have been defined as carefully as possible, they are still rather likely to change. However, this is in fact the opposite of a problem. This indicates that the stakeholders' interests will be kept under constant observation. The project team is likewise demonstrated to react flexibly to changed conditions, increasing the chances of project success. As Wysocki described, "Change is not the enemy; unmanaged change is" (2006, p. 486). To manage requirement changes, a baseline must be generated to perform version control. A change process, including an approval process, must also be established (Wysocki, 2006, p. 486).

Lester (2014) also examined the difference caused by who initiates the change. If a change has been agreed upon, the modified costs and times are incorporated into the budget. This represents the new target value to be monitored. However, matters differ depending on whether the change was requested by the customer or caused internally (e.g., to correct an engineering error). The second case leads to additional costs, which are usually not charged to the customer (i.e., they reduce the profit) (p. 100).

For this reason, the latter simulation also identifies the initiator of change requests, indicating whether these were caused internally or externally (see "7.3.4.7 Reject rates – Calculation of the values for iterations or obsolete tasks and determination of the threshold for the acceptance of a task").

2.4.2 Planning

Kerzner (2009) cited two proverbs that affect project planning:

> *"Failing to plan is planning to fail.*
>
> *"The primary benefit of not planning is that failure will then come as a complete surprise rather than being preceded by periods of worry and depression." (p. 412)*

He described planning "as the function of selecting the enterprise objectives and establishing the policies, procedures, and programs necessary for achieving them. Planning in a project environment may be described as establishing a predetermined course of action within a forecasted environment" (Kerzner, 2009, p. 412).

Lewis (2002), meanwhile, stated that "projects seldom fail at the end. Rather, they fail at the definition stage" (p. 11).

Murthy et al. (2008) wrote that "The project plan deals with planning the remainder of the new product development project in detail and deals with issues such as time and resource allocation, scheduling of tasks, and so on."

Kerzner (2009) identified four basic reasons for project planning:

- To eliminate or reduce uncertainty
- To improve operation efficiency
- To obtain a better understanding of the objectives
- To provide a basis for monitoring and controlling work.

He further described planning as defining what must be performed when and by whom in order to fulfil their assigned responsibility (Kerzner, 2009, p. 415).

Lewis (2002) derived the necessity for planning from the definition of control. To recognize deviations, it is necessary to compare the current position with the position one should be in. Only then can corrective action be taken. A plan is thus necessary to know what one’s situation is, "so planning is not an option" (p. 29).

From this, the question arises as to whether exact planning is possible at all. Even in the literature reflecting the TPM approach, this is doubted. As Kerzner (2009) stated, “Successful project managers realize that project planning is an iterative process and must be performed throughout the life of the project” (p. 412).

Lewis (2002) further suggested that the team members who will later implement the plan should be involved in the plan’s preparation. This can improve estimates and reduce the likelihood that tasks will be forgotten. It is also important to keep in mind that it is always likely that the plan will be revised, as unexpected obstacles often force re-planning. Therefore, if the probability of change is high, planning should not be too detailed, because this would involve unnecessary effort (p. 30).

2.4.3 Work breakdown structure

The first step in planning a project is structuring or breaking down the entire project into smaller elements. Kerzner (2009) described these elements as follows:

- Manageable, in that specific authority and responsibility can be assigned
- Independent, or with minimum interfacing with and dependence on other ongoing elements
- Integratable so that the total package can be seen
- Measurable in terms of progress (p. 434).

The system of these elements is called the work breakdown structure (WBS). Kerzner (2009) described WBS as a product-oriented structuring of the hardware components, services and data required to manufacture the end product. The WBS is organized based on how the work is conducted and represents how project costs and data are aggregated and finally reported (p. 434).

Regarding WBS, Kerzner (2009) stated that the work package represents the decisive level for WBS implementation. A work package comprises a basic task or allocation. It describes the activities to be performed and offers an instrument for monitoring and reporting on work progress. "Work package" is the generic term employed in identifying discrete tasks with definable outcomes. Work packages typically last 80 hours or between two and four weeks. However, this may not be possible for large projects. Furthermore, the larger the work packages are, the more problematic and subjective the work in progress rating will be, unless the packages are broken down by objective indicators, such as milestones with specified budget values or completion percentages (p. 437).

Lewis (2002) claimed that rationale of a WBS is to track all tasks. The WBS should thus be created prior to the schedule. However, a WBS does not describe the order in which the work is executed. Such sequencing is defined during the preparation of a plan, since, “Until everyone has agreed that all tasks have been identified, it is misleading to develop a schedule” (p. 47-49) (see “2.4.4 Scheduling and networking scheduling techniques”).

According to Kerzner (2009), work package characteristics include the following:

- “Represents units of work at the level where the work is performed
- “Clearly distinguishes one work package from all others assigned to a single functional group
- “Contains clearly defined start and end dates that are representative of physical accomplishment (This is accomplished after scheduling has been completed)
- “Specifies a budget in terms of dollars, man-hours, or other measurable units

- "Limits the work to be performed to relatively short periods of time to minimize the work-in-process effort." (p. 437)

Meanwhile, Wysocki (2006) listed six criteria for completeness:

- Status/completion is measurable
- Start/end events are clearly defined
- Activity possesses a deliverable
- Time and cost are easily estimated
- Activity duration is within acceptable limits
- Work assignments are independent (p. 504).

The WBS procedure can be found in the TPM. However, similar structures can also be found in the APM and lean developing approach, though under different aspects (see "3.4.3 Work breakdown structure" in APM and "4.5 Comparison between Lean Production and Lean Development").

2.4.4 Scheduling and networking scheduling techniques

The IPMA Competence Baseline includes a chapter regarding time and project phases. For project phases, see "2.4.5 Stage-gate process and project life cycle." Concerning time, it stated the following:

> *"Time covers the structuring, sequencing, duration, estimating and scheduling of activities and/or work packages, including the assignment of resources to activities, establishing project deadlines and monitoring and controlling their timely execution. These aspects should be displayed on a critical path diagram." (Gaupin & International Project Management Association, 2006, p. 60)*

After the project is structured, in order to analyze what needs to be done, the next step involves determining when the task should be accomplished and by whom.

Lewis (2002, p. 67) suggested that time schedules should be developed according to what is logically feasible, with resources being allocated later. If initial timetable calculations are conducted under the assumption that unlimited resources are available, this results in the optimal schedule. The simulation follows this approach (see "7.3.4.1 Size of the resource pool").

The IPMA Competence Baseline stated that the aim of scheduling is to identify which activities need to be executed and at which time, and to logically order these tasks along a time axis. Scheduling includes the relationships among subprojects and between work packages, as well as the duration and timing of the tasks. The schedules are based on relative prioritization of the activities and the resources available with appropriate capabilities. In cases of uncertainty regarding the time necessary for a certain phase or activity, a time buffer or float should be included in the schedule (Gaupin & International Project Management Association, 2006, p. 60).

According to Kerzner (2009), activity planning offers one of the most important tools for defining how corporate resources are to be incorporated. The schedules are employed to determine time-based resource utilization requirements and provide the basis for visual performance tracking and cost estimation. These schedules help both the customer and management obtain an actual overview of the projects (p. 455).

Various planning techniques have been developed for creating the schedule (GANTT, PERT, Milestone Charts; Networks, etc.). The plan's concrete form is not relevant for the later simulation, since it is modeled on a more abstract level (see "7.3.2 Structure of the development process and mapping in the simulation"). To create the simulation, typical projects were reflected, typically planned with milestone plans and GANTT charts.

Because costs are more or less directly related to efforts in R&D projects, schedule and resource planning influence the planning of costs. Accordingly, the IPMA Competence Baseline stated that project cost management estimates the costs for each work package, subsystems, and the entire project and determines its overall budget. This includes comparing planned versus actual costs at different points in the project, estimating remaining costs and updating the final cost projection. The costs for the deliverables should be both measurable and quantifiable. The costs of changes should also be estimated, approved and reported (Gaupin & International Project Management Association, 2006, p. 64).

2.4.5 Stage-gate process and project life cycle

The traditional approach structures the project into a life cycle with stages and gates. According to Kerzner (2009), the stage-gate process represents the common approach when companies identify the importance of developing project-management processes. The stage-gate process was developed because organizations' traditional structure was hierarchical, with centralized management, control and communication, which are no longer practicable for project management with its horizontal workflows. The stage-gate process then progressed to life cycle phases. (p. 66).

The IPMA Competence Baseline (Gaupin & International Project Management Association, 2006) defined a project phase as a discrete period in the course of a project that is distinct from other periods. A project phase comprises both important project achievements and decisions that form the basis for the next phase. Phases further possess defined objectives and deadlines. Different kinds of (sub-)projects can utilize distinct phase models, making coordination more complex. Milestones can be employed to aim for specific objectives or time intervals. In the real world, project phases may also overlap (p. 60).

Kerzner (2009) further stated that the process consists of stages and gates. Stages describe a set of activities that can be executed either in series or in parallel, depending on the level of risk the project team can be exposed to. The phases are led by cross-functional teams, while the gates form the structured decision points located at the end of each phase (p. 66).

Gate reviews are not the same as status reports. Lewis mentioned that response to control data must be timely. In some cases, project status data can even be delayed by four to six weeks, thus becoming obsolete as a starting point for any corrective action. At best, project status information should be provided in real time (Lewis, 2002, p 94).

Regarding the decisions at the gates, Kerzner stated that “Gatekeepers must be willing to make decisions. The four most common decisions are:

- “Proceed to the next gate based upon the original objectives
- “Proceed to the next gate based upon revised objectives
- “Delay making a gate decision until further information is obtained
- “Cancel the project” (Kerzner, 2009, p. 67).

Sponsors also require the bravery to stop a project. The gates’ aim is not only to obtain permission to continue, but also to detect failure early so that no more resources are wasted in this project, instead being allocated to more promising activities (Kerzner, 2009, p. 67).

Lester (2014) also used the terms “decision points,” “milestones” or “trigger points” for the gates of the end of the life cycle phases. A review should be performed at every gate to ensure that the project remains profitable, that it is on track, that costs remain within budget, that adequate resources are allocated for the next phase and that identified risks can be controlled (p. 49).

Kerzner (2009) conclusively stated that the purpose of life cycle phases is control. Upon completion of each phase, a review meeting is held with the project leader, sponsor, senior management and customer to evaluate that stage’s achievements and obtain authorization for the next phase (p. 418).

2.5 CRITICS OF THE TRADITIONAL PROJECT MANAGEMENT

Although project management is described in detail in the standards and guidelines, a large number of projects are not successfully completed. According to the Standish Group Chaos Report, only 16% of projects succeed, while 31% fail completely (Clancy, 1995). Meanwhile, software development in particular has generated discussion regarding how its product development should take place.

First, some authors have stated that software is different in this matter. Specifically, Stepanek (2005, p. 22) listed the following points, among others:

- Software is unique in that its most significant issue is its complexity.
- It is uniquely difficult to define a complete set of requirements for software before beginning development.
- Software development is not just a process of creating software; it also involves a learning process for how to create the software that is best suited for its purpose.
- Software can be modified rapidly, and this pace is expected, but it is better to implement the changes properly.
- No software is perfect as first envisioned; it will always require changes to make it best suit its role.

Because software development represents a large aspect of the kind of project investigated in this research, these points are worth discussing when evaluating the project management approaches.

Stepanek (2005, p. 49) further listed hidden assumptions behind the project management guidelines (especially the PMBoK) and stated that these are not valid for software development:

- Scope can be completely defined.
- Scope definition can be performed before the project starts.
- Software development consists of distinctly different activities.
- Software development activities can be sequenced.
- Team members can be individually allocated to activities.
- The project team size does not affect the development process.
- There is always a way to produce meaningful estimates.
- Acceptably accurate estimates can be obtained.

- One developer is equivalent to another.
- Metrics are sufficient to assess software quality.

This criticism, which originated in software development, finally led to the emergence of new PM methods, which found their echo in the Agile Manifesto (see "3.2 The Agile Manifesto, the Agile Principles and the Declaration of Interdependence").

However, the points of criticism concerning the TPM also apply to projects outside the IT environment. Cumbersomeness when new framework conditions arise, or a slow reaction to necessary changes, also present an issue in the hardware development environment. The goal of developing exactly the product that the end customer requires also applies here, but this is made more difficult by the early definition of the requirements. The explicit focus on customer benefit (and not on adhering to a plan established at the beginning) is reflected in both APM and the lean philosophy. Therefore, both topics are discussed in greater detail in the following chapters.

2.6 CONCLUSION OF THE DEFINITIONS FOR THIS RESEARCH

The definitions and deeper examination of the project management theory helps to distinguish the traditional approach from the agile alternative. Continuing, chapter three discusses the concept of APM and relates this approach to the general definitions of project management. In chapter four, both approaches are further related to the concept of lean development, forming the basis for comparison of both approaches in chapter 4.5.

3 AGILE PROJECT MANAGEMENT

3.1 THE HISTORY OF AGILE PROJECT MANAGEMENT (APM)

Seibert (2007) described APM's history, including its different predecessors, such as the following:

- The iterative and incremental approach models already presented in the 1970s by Basili and Turner, in which a software system is split into subsystems and piecewise functionality is added and improved.
- The spiral model developed by Barry Boehm in the mid-1980s.
- The evolutionary project management model developed by Tom Gilb.
- The term "agile software development" was first used by the Japanese Mikio Aoyama in 1993.

Leutze (2014) analyzed which theories and practices influenced the creation of the APM approach, mentioning the following:

- Complexity theory
- Deming's quality theory (quality management)
- Theory of constraints
- Lean
- Education and dissemination of knowledge.

The history of the Agile Manifesto itself was described at the website (Highsmith, 2001). There, it was stated that a meeting was held on February 2001 in Utah, with leading representatives from Extreme Programming, SCRUM, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development, and Pragmatic Programming exchanging their experiences and developing an alternative to documentation-driven, heavyweight software development processes.

3.2 THE AGILE MANIFESTO, THE AGILE PRINCIPLES AND THE DECLARATION OF INTERDEPENDENCE

The Agile Manifesto's authors stated the following on their website (Highsmith, 2001):

> *"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."*

It must be mentioned that, in this original version, the manifesto related solely to software development.

The authors further described 12 principles behind the Agile Manifesto (Highsmith, 2001):

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

- *Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.*
- *Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale.*
- *Business people and developers must work together daily throughout the project.*
- *Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.*
- *The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.*
- *Working software is the primary measure of progress.*
- *Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.*
- *Continuous attention to technical excellence and good design enhances agility.*
- *Simplicity—the art of maximizing the amount of work not done—is essential.*
- *The best architectures, requirements, and designs emerge from self-organizing teams.*
- *At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.*

Beyond this, "A less well-known descendant of the Agile Manifesto is the Declaration of Interdependence (DOI), which seeks to extend the Agile Manifesto to non-software products, project management and management in general" (Harris, 2007).

The DOI stated the following:

> *We are a community of project leaders that are highly successful at delivering results. To achieve these results:*

- *We increase return on investment by making continuous flow of value our focus.*

- *We deliver reliable results by engaging customers in frequent interactions and shared ownership.*
- *We expect uncertainty and manage for it through iterations, anticipation, and adaptation.*
- *We unleash creativity and innovation by recognizing that individuals are the ultimate source of value, and by creating an environment where they can make a difference.*
- *We boost performance through group accountability for results and shared responsibility for team effectiveness.*
- *We improve effectiveness and reliability through situationally specific strategies, processes and practices (Anderson et al., 2005).*

The authors further explained that "the title 'Declaration of Interdependence' has multiple meanings. It means that project team members are part of an interdependent whole and not a group of unconnected individuals. It means that project teams, their customers, and their stakeholders are also interdependent" (Anderson et al., 2005).

3.3 METHODS OF APM

Seibert (2007) listed APM's methods as follows:

- Extreme Programming (XP) from Kent Beck and Ward Cunningham
- Adaptive Software Development (ASD) from Jim Highsmith
- Crystal-Clear from Alistair Cockburn
- Scrum from Ken Schwaber and Jeff Sutherland
- Feature-Driven Development (FDD) from Jeff DeLuca and Peter Coad
- Lean Software Development from Mary and Tom Poppendieck.

He stated that, "Although these approaches all have different focus and working methods, they are based on common beliefs and basic principles, which they use to delineate themselves from the plan- and specification-driven approaches based on the waterfall model. Most agile approaches are based on the fact that they try to minimize the pure design phase and to arrive as early as possible to executable software in the development process, which can then be submitted to the customer at regular, short intervals (iterations) for joint co-ordination. In this way, it should be possible at any time to respond flexibly to customer requirements, thus increasing overall customer satisfaction“ (Seibert, 2007).

Seibert (2007) named Scrum as the representative method for APM, given that Scrum focuses more on project management than other agile approaches.

3.4 COMPARISON OF TPM AND APM

In the literature, opinions diverge regarding the relation between TPM and APM approaches. Some authors have discussed how APM is reflected in current norms and guidelines, while other authors have attempted to integrate TPM and APM (see “3.6 Hybrid Models”). Still other authors have claimed that agile methods contradict traditional methods and that these methods are “not compatible” (Seibert, 2007), or else that APM is superior to TPM, at least for projects that are complex, uncertain and time limited (Fernandez & Fernandez, 2008).

Griffiths (2004) described a method for “using Agile alongside the PMBOK." Sliger (in (Sliger & Consulting, 2008) and (Sliger, 2006)) identified a high level of compatibility between the PMBOK and agile practices, while Binder, Aillaud and Schilli (2014) balanced agile and ISO 21500. Conversely, Koskela and Howell (2002) argued that “the underlying theory of project management is obsolete” (p. 293).

Wysocki (2006) compared the different perspectives of the enterprise, the customer, the project manager and the development team regarding projects. This aided comparing the different project management strategies. From the enterprise’s perspective, it is “a good strategy for the project manager to build a project plan that delivers value early rather than late in the project life cycle” (p. 420). Regarding the customer’s perspective, the greatest challenge for many project managers and their development teams is to retain the customer throughout the complete project lifecycle. The intensity of customer retention varies with the chosen approach. With linear and incremental PM strategies, the effort is minimal, as it is assumed that requirements, functions and features have been completely recognized and documented. If one changes the adopted strategies to iterative, adaptive and extreme, this participation increases (Wysocki, 2006, p. 421-422).

The project manager's perspective also involves the customer: It is important that the project manager and the team understand what the customer really needs and that the customer really understands what will be delivered. The more uncertain one of the parties is regarding what will be delivered, the more the strategy should focus on the agile approaches (iterative, adaptive or extreme). This minimizes the estimation or even guessing of future undefined requirements, functions or features. In so doing, it minimizes wasted time due to wrong assumptions, because "any waste that can be prevented must be prevented. And that means no guessing" (Wysocki, 2006, p. 422) (see also "4.3 The transfer from Lean Production to Lean Development: The motivating reasons" discussing the lean goal of avoiding waste).

From the development team's perspective, different types of developers can be identified in terms of uncertainty and customer contact: At one end, some do not like to think for themselves. They expect a clear and complete specification to work with without fearing changes caused by the customer. Many possess no intention of discussing it with the customer at all. On the other end of the spectrum are those who possess a positive relationship with the customer and want to work together with him to achieve the best solution. They experience no problem working on a complex, poorly defined project. Many actually thrive on exactly this kind of projects (Wysocki, 2006, p. 423).

Seibert (2007) introduced the APM by mentioning that global competition, growing market dynamics and increasingly complex technologies since the 1990s have required ever shorter project durations and more frequent technical changes during the project. Many customers were also unable to formulate their requirements for a new system precisely at the start of the project. The waterfall model was not geared to such boundary conditions and, in conjunction with widespread quality management systems (ISO 9001, CMM, V-model, etc.), became increasingly documentation-intensive and cumbersome.

Based on the PMBoK's hidden assumptions (see "2.5 Critics of the Traditional Project Management"), Stepanek (2005, p. 94-95) discussed how agile methodologies avoid these invalid assumptions:

With regard to whether the scope can be defined completely and already at the beginning of the project, he stated that APM does not "require all of the scope to be defined up-front: scope definition is done throughout the course of the project."

For sequencing and specialized allocation of activities, the rule for the APM is that "Agile methodologies treat development as a single activity, so design, coding, and testing are done concurrently. The developers are all doing the same job."

In APM, the team's size is of relevance. In fact, "Crystal defines explicit changes to the methodology depending upon the size of the project team. In XP, it's implicit that the methodology is intended for small teams or subteams of 2 to 12 developers."

Derivation of estimates also becomes more accurate, because "estimates can be based on the experience of prior iterations, which allows you to build up a history of real metrics to make estimation more accurate" (all in (Stepanek, 2005, p. 94-95).

Dybå and Dingsøyr (2008) further listed the main differences between TPM and APM:

	TPM	APM
Assumption	Systems are fully specifiable and predictable	Continuous design improvement based on rapid feedback and change

	TPM	APM
Management	Command and control	Leadership and collaboration
Knowledge Management	Explicit	Tacit
Communication	Formal	Informal
Development model	Life-cycle model	Evolutionary, iterative
Organization	Large, mechanistic	Small, organic
Quality control	Heavy planning, late heavy testing	Continuous control of requirements, continuous testing

Table 1: Main differences between TPM and APM[1]

To investigate APM's applicability in the non-software area, Owen, Koskela, Henrich and Codinhoto (2006) presented the differences between APM and TPM, which can be found on Cockburn's website (Cockburn, n.d.).

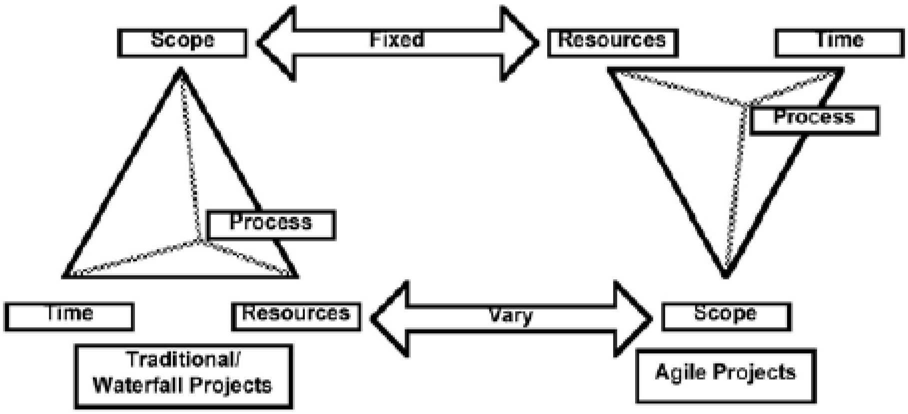


Figure 1: Difference between TPM and APM – Fixed and vary parts of the iron triangle (Owen et al., 2006)

---

[1] Derived after (Dybå & Dingsøyr, 2008)

3.4.1 Requirements management and change management

For a number of authors, the main difference between TPM and APM concerns whether all information is already available at the beginning of the project in order to be able to specify the requirements precisely at this point. In APM, the requirements are reassessed in each iteration, far more frequently than in TPM. According to its philosophy, APM welcomes changes, unlike TPM, where any deviation from the original plan constitutes a problem.

Wysocki (2006, p. 480) discussed difficulties in gathering requirements (see also "2.4.1 Requirements management and change management" in TPM). He doubted that the customer will be able to specify exactly what the product will look like at the start of the project. Furthermore, there are

> *"customers or clients who can't seem to make up their minds or are known to constantly change their minds, and you have a significant problem to deal with. … The root cause of many problems that come up in the course of doing a project originate in a disconnect between what the client says they want and what they really need. … Traditional Project Management (TPM) forces them into specifying what they want when that is the absolute wrong thing to do." (Wysocki, 2006, p. 480)*

Salameh (2014) also discussed the limitations of the TPM approach: Projects seldom follow a sequential process, since it is usually difficult for customers to define a project's requirements completely, correctly, and above all, at the beginning. TPM requires disciplined planning and control methods based on the assumption that project objectives and activities are predictable, and that the events and risks affecting the project are manageable. TPM is driven by linear processes and practices enabling the project leader and the team to define and complete the project through detailed pre-planning. Furthermore, TPM expects that a phase will not be repeated after completion.

The APM methodology forms a contrast: The APM approach enables immediate adjustment of the project as the requirements are checked and evaluated in each iteration. In addition, APM pursues a feature-driven management approach; the project's scope is defined by prioritizing project features and requirements according to value or importance. Therefore, customer involvement in the scope and analysis of project requirements is critical. Customer engagement helps the agile project team to avoid investing a great deal of effort in developing low-value, costly functions or requirements.

In the TPM, scope deviations identified during the project plan's execution are not to be expected and should not occur. This constitutes a major difference between TPM and APM: APM expects deviations in scope, cost and time as a normal property of projects, and thus designs its processes to support and address them; TPM, meanwhile, assumes that all executed plans are in place and all requirements are properly defined and understood, meaning any deviation is flagged as negative. (Salameh, 2014).

Collins (2014) mentioned that requirements often change considerably during a development process. The longer the time interval from requirement capturing to fulfillment, the more likely the customer will state that what has been delivered does not match their actual needs. It is also unlikely that many of the requirements will still be considered important after a few months. Reducing the risk that development will be disconnected from what the customer really needs is possible by providing the customer more frequent feedback, including what is being developed. APM will speed up the feedback cycle and actively engage the customer in prioritizing the requirements and in designing the product. At the heart of agile methods is the regular production of tangible software or product with a constant customer-centric communication cycle.

Salameh (2014) highlights the different view on change: TPM and APM differ significantly regarding how they view and approach change. According to TPM, any change to the project details and clearly defined scope is perceived as a threat that should be managed; therefore, change management is defined as the process of avoiding scope fluctuations or changes. In the APM perspective, on the other hand, change management is expected and facilitated. APM expects that costs, time and quality are fixed, and only the scope can be modified. In APM, the project team is committed to delivering at a fixed date, at a fixed price, but with a flexible scope. Therefore, the APM project team concentrates on working only on high-priority project deliverables and requirements that offer the greatest business value. If a new feature needs to be included within the project scope, it has to be replaced with an element with the same effort. This approach differs from the TPM approach, where new functionality is added at the expense of cost and delivery date.

Collins (2014) concluded that the APM approach ensures that more important subjects are processed first. Traditional planning usually centers on the principle of implementing a project within a defined budget and timeframe. The agile philosophy, by contrast, is to deliver high-quality products or software as quickly as possible. This value is increased by ensuring that this delivery will benefit from regular feedback. This also helps to deliver the greatest value in terms of customer-prioritized functionality within a given timeframe. As such, there is a shift away from delivering within budget and time to delivering the greatest possible customer benefit within that time.

3.4.2 Planning

Part of the agile method ethos is that less initial planning is better and an evolutionary process is more efficient (Dybå & Dingsøyr, 2008) and (Serrador & Pinto, 2015). Agile methods differ from TPM methods (e.g., waterfall) in that they rely on continuous design, flexible scope, and freezing design features as late as possible. This includes the inclusion of uncertainty and customer involvement as well as a modified project team structure. In addition, agile is characterized as iterative and incremental so as to avoid the standard approaches of freezing design and specifications at an early stage, a fixed project scope, and limited customer engagement (Serrador & Pinto, 2015).

However, agile methods do not mean not planning. It is important to note that agile does not eliminate front-end planning as part of the project development methodology. In fact, agile methods demand planning in advance. To meet the project requirements for the first release, extensive communication and collaboration with the customer is required. Following this, even more planning is performed in agile environments, though planning is fragmented throughout the development lifecycle rather than being conducted in advance and only once (Serrador & Pinto, 2015).

It would therefore be a fallacy to claim that no more planning is performed in the APM. It simply becomes more short-cycled, meaning more frequently planned, and only as necessary for the next step. A balance between traditional methods and agile methods is usually suitable for project planning. A number of parameters, such as project size, security requirements and known future demands require advance planning, even for agile projects, while turbulent, rapidly changing environments require less advance planning and more agile methodologies (Serrador & Pinto, 2015).

3.4.3 Work breakdown structure

Of course, the APM also divides the entire project into partial aspects. Whereas product development is broken down into work packages in TPM, for APM, the requirements and features are separated instead.

Tolbert (2012) elaborated that APM utilizes a structure similar to a WBS: In TPM, the project structure plan forms the basis for all further planning processes, from which, among other things, the schedule, budget and risk management are derived. In the WBS, the lowest, most detailed level is the "work package." The engineering view of the product prevails in the project structure plan, with the various teams being responsible for their respective requirements. In the APM approach, the focus is not on the work packages, but rather on an FBS (Feature Breakdown Structure). However, the entire team works together across all iterations. Customer orientation is thus permanently maintained.

The requirements can be structured, such as via so-called epics and stories up to individual features, and they are monitored accordingly in the backlog. In APM, the traditionally used WBS, with which the project plan is realized, is replaced by dynamic backlogs. Usually, "stories" are developed first, describing the more general business value priorities. This makes it easier to develop a rough estimate and illustrate how the value is validated (Owen et al., 2006).

3.4.4 Risk management

Risk management plays a major role in TPM. Salameh (2014) established the connection to the PMBoK (Project Management Institute, 2013) and listed the corresponding topics: A project risk describes a possible event or uncertain condition that, when it occurs, usually negatively affects project objectives, such as scope, timing, costs and quality. This is why active risk management must be practiced in the TPM throughout the project's entire duration. This includes identifying and analyzing risks, as well as planning and controlling countermeasures.

In contrast, explicit risk management plays a minor role in APM. As Salameh argued, the need for formal risk management is often disputed in APM. Due to its iterative approach and the resulting limited, well-controlled scope, explicit risk management becomes irrelevant if a project pursues an agile approach. Frequent customer deliveries while focusing on user acceptance help project teams to minimize the greatest risk most projects are exposed to: failing to deliver or delivering the wrong product (Salameh, 2014).

3.4.5 Stage-gate process and project life cycle

Salameh (2014) further linked the TPM and APM approaches to the project management process groups and knowledge areas (see Guide to the Project Management Body of Knowledge (Project Management Institute, 2013)). The distinguishing feature of agile development is that each iteration is completed in itself, as it covers all activities from requirements analysis to design, implementation and testing. At the end of each iteration, a working product is delivered to the customer; the customer can evaluate this product and then refine the requirements and functions as needed, to be included in future iterations.

This means that the usual TPM process steps are also performed in APM, though not just once for the entire project; rather, they are performed in small steps in each iteration. In APM, the process groups are also not simply executed linearly, as in TPM. If APM is utilized, the process groups are executed repeatedly at the iteration level. The main difference between the two project management methods is that TPM is distinguished by extensive but inflexible planning and control procedures, task allocation and assignment, as well as strict adherence to milestones. In APM, by contrast, planning occurs incrementally throughout the project's lifecycle in the planning of each iteration (Salameh, 2014).

3.5 APM FOR NON-SOFTWARE PROJECTS

The Agile Manifesto originally focused on software development. However, the question was eventually raised concerning whether the approach can also be applied to non-software projects.

Stare (2014) mentioned that APM is mainly established in the area of IT projects, but not yet in other project types (e.g., engineering). This is probably due to how frequent changes seem too expensive, partial deliverables are difficult to use, or the implementation in a multi-project landscape with shared resources is more difficult.

Cooper and Sommer (2016) stated that agile development methods are traditionally only used for software projects. However, new findings have illustrated that agile methods can be integrated with traditional and known approaches to obtain an Agile-Stage-Gate® hybrid model. In addition, research on recent industry experience suggests that this new hybrid model offers significant potential for manufacturers of physical products, ranging from heavy industrial equipment to food and toys, leading to surprisingly positive results.

They further discussed how agile-stage-gate must be customized for manufactured products: Although APM offers numerous advantages in product development, the methods for software development cannot be transferred to hardware development without adjustments. The concept of a "done sprint" in particular has to be considered, given the short iteration time. Therefore, an important difference between agile for software and agile-stage-gate for physical products is the definition of a sprint and what "done" means. Developing a new chemical or pharmaceutical product or a new machine cannot simply be incremented. For complex products, it remains difficult to build a single part that demonstrates the system's functionality within a short time, nor is it possible to market it. Thus, in contrast to software APM, an iteration's result may not be a functioning product, but it is still something that can be demonstrated. These could include, for instance, virtual prototypes, 3D-printed prototypes or working models. This also provides the customer something to try out and evaluate (Cooper & Sommer, 2016).

Figure 2 Illustration of deliverables in non-software projects (Cooper & Sommer, 2016)

For this thesis, it is generally assumed that it is possible to also master such a project as is considered here (i.e., not only software, but also mechanical and electronic components) using agile methods.

3.6 HYBRID MODELS

In order to balance the advantages and disadvantages of TPM and APM, the literature has also described so-called "hybrid models." The project as a whole is based on a phase model, but the individual phases are handled agilely.

Binder et al. (2014) stated that the hybrid approach strives to combine the strengths of both APM and TPM to simultaneously meet the requirements of corporate policies and procedures. Here, the aim is not on comparing processes. Instead, the TPM methods embodied in the standards describe which processes a project manager should follow (what to do), while APM establishes principles for the way the project team works (how to do it).

According to Stare (2014), rather than describing the entire product lifecycle (initiation, planning, execution, closing), the APM instead focuses solely on the execution phase. In principle, the product lifecycle remains the same, with only some parts being moved to the execution phase. For example, the specification and part of the planning is also conducted in the execution phase. At the beginning, a rough plan is created, which is certainly necessary to obtain an overview of the whole project. The planning is then detailed for each iteration, improving accuracy.

Cooper and Sommer (2016) offered a similar argument: The stage-gate process consists of a sequence of five or six individual phases, starting with the "Idea Generation" and continuing to "Product Launch." The end of each phase contains a gate in which the project's continuation is judged. This is where poor projects are stopped so that resources can be diverted to more promising projects. The stage-gate process is cross-functional, involving technical developers as well as decision-makers from marketing, sales and operations. Gating models are generally regarded as planning-based approaches. APM now divides the project's development phase into a series of short iterations. Thus, the agile process is highly responsive; it is not plan-based, but rather "planned on the fly," iteration for iteration. Agile is therefore primarily used for the project's "technical phases" (development and testing).

This is in line with this thesis' approach of limiting the project horizon to the development and testing phases.

To integrate agile into stage-gate, Cooper and Sommer (2016) mentioned that "The differences emerge from the different intents of the two systems: Stage-Gate is a comprehensive idea-to-launch system and a macroplanning process, whereas Agile is a microplanning project management methodology" (p. 169).

Boehm and Turner (2005) discussed the management challenges for implementing agile processes in traditional development organizations—namely, how agile, light processes can be combined with standard industrial processes without affecting agility or damaging systems and software development processes that have been defined and refined over years of work.

They also identified difficulties in integrating the different timeframes, with it being problematic to reconcile different life cycles. Agile processes concentrate on providing functionality immediately, while traditional methods aim to optimize development over time. The traditional longer lifecycles demand modifications to the agile processes. For instance, the documentation that traditional methods require for training and support is not necessarily considered in APM (Boehm & Turner, 2005).

3.7 SUITABILITY OF THE PM APPROACHES FOR DIFFERENT TYPES OF PROJECTS

Boehm and Turner (2004), as well as Vinekar, Slinkman and Nerur (2006), analyzed the "home ground" for the different approaches. The different characteristics of application, management, technical aspects and personnel are accordingly considered. The choice between TPM and APM for a specific project thus depends on the following factors:

- The size of the system's development project and team
- The consequences of failure (i.e., criticality)
- The degree of dynamism or volatility of the environment
- The competence of personnel
- Compatibility with the prevailing culture (Vinekar, Slinkman, & Nerur, 2006, p. 32)

In essence, they offered a "strategy for choosing a particular approach for a particular project based on the risks posed by the five factors mentioned above, implying the simultaneous pursuit of agile and traditional development approaches" (Vinekar et al., 2006, p. 32). The authors thus strove to integrate agile and traditional "subunits" and compared the different approaches:

	Traditional Subunit	Agile Subunit
Management and Organizational	Command and control	Leadership and collaboration
	Autonomous	Cooperative
	Disciplined	Flexible
	Manager as planner	Manager as facilitator
	Explicit knowledge	Tacit knowledge
	Individual reward systems	Team reward systems
People	Individual work	Collaborative work
	Specialized skills	Multidisciplinary skills
	Managerial decision making	Pluralist decision making
	Low customer involvement	High customer involvement
	Large teams	Small teams
Process	Process centric	People centric
	Standardized	Speculative
	Measure progress	Assess progress
	Life-cycle development	Evolutionary development
	Write code prior to tests	Write tests prior to code
	Unified approach to projects	Individual approach to projects
	Preplanned	Adaptable
	Linear	Iterative
	Long durations	Short durations
Technology	Structured or object oriented	Object oriented
	Standardized tools	Tools for iteration

Table 2: Comparison of traditional vs agile subunits (Vinekar et al., 2006, p. 37)

Boehm and Turner (2004) discussed the balance between agility and stability, concluding in the “home grounds” of agile and plan-driven methods. Finally, they summarized the critical factors involved in determining the relative suitability of agile or plan-driven methods in a particular project situation:

	Traditional	Agile
Goals	Predictability, high assurance	Rapid value
Criticality of project	Extreme	Low

	Traditional	Agile
Developers experience	Junior	Senior
Requirements	Stable	Change often
Size	Large products and teams	Small projects and teams
Culture	Clear policies and processes; demands order	Many degrees of freedom; responds to change
Planning and Control	Documented plans, quantitative control	Internalized plans, qualitative control

Table 3: Home grounds for TPM and APM[2]

3.8 CRITICS

Agile methods are not undisputed in science and practice. Seibert (2007) listed the following points of criticism:

- Agile methods only work for small projects, usually possessing less than 10 employees. With increasing project size, the need for communication and documentation also increases. This means that much of the agility is lost.
- Agile projects are riskier. (...) Little methodological support is available for agile collaboration with subcontractors. In the case of employee fluctuation, undocumented knowledge is lost.
- Agile methods do not allow fixed price contracts, so they often encounter resistance in purchasing departments. Nevertheless, contract models are available that can be used in agile projects.
- The agile methods were developed for software projects, which are often conducted in integrated teams and in which practically no production phase exists. However, numerous APM approaches have been employed for a long time. These include, for example, simultaneous and/or concurrent engineering, the target costing or design to cost approach, or the stage-gate model related to the spiral model, as well as other methods

---

[2] Adapted after (Boehm & Turner, 2004, p. 51 -55)

of lean automotive development, as described in Japan by Toyota and in Germany by Porsche.

- Introducing agile practices in traditional corporate structures involves considerable effort. Reservations and resistances must be overcome, and the organization must be transformed into cross-functional teamwork. All this corresponds to a cultural change, which is to be regarded as an independent organizational change project and should be supported by a change management geared towards it.

Vinekar et al. (2006) mentioned that agile methods depend heavily on the customer, who has to identify and prioritize features. The customer also offers feedback regarding the delivered products and triggers or accompanies changes in the course of development. This becomes increasingly difficult when the client's representative possesses goals that differ from other stakeholders.

If the methodology is exclusively agile (i.e., no hybrid model), this introduces a risk of losing the overall view of the entire project cycle. Coordinating several development teams is also difficult in large projects. APM may even be incompatible with existing corporate structures. Furthermore, the communication and coordination effort involved is high. As such, the Agile Manifesto is more a summary of values than a concrete recommendation for action.

3.9 CONCLUSION OF THE DEFINITIONS FOR THIS RESEARCH

For the later simulation, the APM goals, as well as the necessary organizational circumstances, are less important compared to the differences that can be seen in the project’s execution. The most important differences concern the much shorter cycles, in which the customer is provided with a new iteration of the product (or a development step that can be evaluated), and in which the scope and associated priorities are redefined. It is expected that a product, which is assumed for this thesis, can be developed with agile methods. Any reservations or difficulties involved with introducing the APM are regarded as solved.

4 LEAN DEVELOPMENT

4.1 THE HISTORY AND IMPORTANCE OF LEAN PRODUCTION

The original lean concept of "flow" was developed at the beginning of the 20th century, mainly by Fredrick Taylor and Henry Ford (Locher, 2008, p. xiii). Taiichi Ōno, production manager of the Toyota factory in the 1950s, rejected the principles of mass production as not suitable for producing small quantities of high variance. However, this period of change from the seller's to the buyer's market represented the real value for the customer (Erne, n.d.). Toyota took Ford's original flow concepts to the next level, employing a diversified production of small batches. Toyota does not explicitly use the term "lean," but has been applying lean concepts to its production and product development systems (PDSs) for more than fifty years regardless, and it is considered a leader in the application of lean thinking (Locher, 2008). Taiichi Ohno initially applied the concept to the manufacture of car engines in the 1950s, later moving to vehicle assembly (1960s) and the entire supply chain (1970s). It was here that supplier manuals were created and the "secrets" of this lean approach first shared with companies outside Toyota (Hines, Holweg, & Rich, 2004). In their book "The Machine that Changed the World" published in 1990, James Womack and Daniel Jones described the success of the Toyota production system (TPS). In Lean Thinking, published in 1996, Womack and Jones demonstrated that this represents a "fundamentally different way of thinking about processes, systems and organizations as a whole" (Locher, 2008, p. xiii).

Driven by a shortage of resources and intense competition in the Japanese automotive market, new methods were developed. These included the just-in-time production system (JIT), the Kanban method of pull production, respect for employees, and a high degree of (automatic) error checking and problem solving by employees. This lean approach focused on eliminating waste in the value stream, thus offering an alternative model to that of capital-intensive mass production (with its large lot sizes, dedicated equipment and hidden waste) (Hines, Holweg, & Rich, 2004, p. 994). Therefore, the core of the "lean" idea is not primarily becoming slim or slender. The main intention is rather to create value for the customer, or, in negative terms, to avoid anything that is not value-adding to the customer. "Lean" thus means creating values for the customer and avoiding waste as much as possible (Erne, n.d.).

4.2 THE PRINCIPLES OF LEAN PRODUCTION

Three concepts are significant in lean thinking: value, waste and the added value with no waste process, covered by the six lean principles (Oehmen et al., 2012).

Lean describes a systematic manner of maximizing customer value while minimizing waste. The product or service flows through the organization, triggered by the customer requirement (pull principle). Lean’s ultimate goal is perfection, achieved through a continuous pursuit of improvement.

Recognizing and eliminating waste represents lean thinking’s critical capability; all work activities are accordingly divided into the following three categories (after (Womack & Jones, 2003) in (Oppenheim, 2011)):

1. Value-adding activities: Information or material that adds value from the customer’s point of view is transformed. So, the customer must be willing to pay for this. This is performed correctly the first time.

2. Required or necessary activities without added value: The value is not increased from the customer’s point of view, but these activities cannot be eliminated because they are required, for example, by law, contract or for technological reasons.

3. Non-value-added activities: Resources are consumed and no value is created. This represents pure waste (e.g., unneeded reports and e-mails, idle times, post-processable defects, etc.).

Negative formulation of the "lean" idea concerns avoiding any form of waste. Ōno (1988) identified seven sources of waste in production (Erne, n.d.):

1. **Overproduction:** If more is produced than demanded, this leads to excessive stockpiles and excessively high capital-binding capacities, which is wasteful.

2. **Excess Inventory:** Inventories in the production and in the raw material and/or finished parts stocks generate capital costs. These inventories represent a direct consequence of overproduction.

3. **Excess Motion:** Waste is also caused by the non-ergonomic arrangement of tools or workpieces, or by long driveways, which require non-value-adding and thus unnecessary movements.

4. **Transportation:** Transports of all kinds constitute a waste, since this does not create value, instead simply changing the material's position.

5. **Waiting:** Any form of waiting time represents a waste. This is caused by a lack of material, stoppages due to disturbances, or excessive batch sizes on a machine.

6. **Overprocessing:** Excessive and/or unnecessary workflows, such as a deep drilling or redundant testing of the same matter, constitute another waste source.

7. **Defects or Rework:** Defective parts cannot be processed or delivered to the customer in the subsequent process step. The subsequent post-processing or sorting then leads again to higher production costs without adding value.

The "lean" idea can be formulated positively in five basic principles (after Womack, Jones & Roos, 1990 in (Erne, n.d.)):

1. Define value from the customer's point of view

2. Identify the value stream

3. Keep the value stream flowing

4. Let the customer determine the takt of your work

5. Strive for perfection step by step

Oehmen et al. (2012) further introduce a sixth principle:

6. Respect for people

4.3 THE TRANSFER FROM LEAN PRODUCTION TO LEAN DEVELOPMENT: THE MOTIVATING REASONS

For the last few years, most industries have been exposed to enormous cost pressure, leading to significantly reduced time to market. With increasing competition, product launch speed can be decisive.

Liker and Morgan (2006) summarized the positive impact this has generated for Toyota: Through the Toyota PDS, Toyota can develop higher quality vehicles faster, more cost effectively and for a higher profit than the competition. Toyota can manage more vehicle launches annually than most of its competitors, generating a continuous stream of new, high-quality products to satisfy consumer demand. This capability has led to industry-leading profits paired with market leadership, making Toyota the world's largest car manufacturer.

In addition to its economic success, the TPS has also led to significantly accelerated development cycles. In terms of speed to market, Toyota only requires 12 to 15 months from styling freeze to production start, while competitors require 20 to 30 months for a similar task (Liker & Morgan, 2006).

Equally impressive are the quality figures, where Toyota is world leader. Objective data indicates that Toyota stands out regarding the quality of its new products. J.D. Powers conducts an annual survey in which the initial quality in the first 90 days of ownership serves as an indicator. In this survey, Toyota took first place in the decade with 39 vehicles, including an impressive ranking of 10 vehicles in 16 categories in 2005 (Liker & Morgan, 2006).

At Toyota, the cost effects of errors were discovered at an early point in time. An essential aspect of quality is cost of poor quality. The cost of fixing an error grows by a factor of 10 with each development phase. The cost of $1 in the concept phase grows to $10 in the design phase, reaching $100 in the production phase and $1000 if the problem is detected by the customer (Locher, 2008, p. xi).

4.4 TOYOTA’S PRODUCT-PROCESS DEVELOPMENT SYSTEM

The TPS is based on lean principles. This includes customer orientation, continuous improvement and avoidance of waste. The upstream and downstream processes are integrated into a lean value chain (Liker & Morgan, 2006).

The PDS is based on two principles: "Do the thing right" and "Do the right thing." The first leads to operational efficiency and waste reduction, while the second satisfies the needs of all stakeholders. In this way, Toyota’s PDS has demonstrated outstanding performance (Pessôa, Loureiro, & Alves, 2007).

Locher (2008, p. 45) listed the following set of key concepts forming the basis of any lean development system:

- Distinguishing between knowledge reuse and knowledge creation
- Performing development activities concurrently wherever possible
- Distinguishing between “good” iterations and “bad” iterations
- Maintaining a process throughout.

4.5 COMPARISON BETWEEN LEAN PRODUCTION AND LEAN DEVELOPMENT

According to Mund (2014), lean production and lean development differ strongly from each other: Production focuses on converting raw material into an end product through a series of process steps. These process steps are usually both stable and predictable, and each step adds value to the product. In manufacturing, the sequence of tasks and activities is repetitive, and the task time measures in minutes or even seconds. Variability destabilizes the flow and generates wasteful inventory, and thus needs to be reduced or eliminated. The environment for product design, development and engineering is highly complex and poses specific challenges. Unlike manufacturing, PD describes a non-recurring process that focuses on information flow, ideas and knowledge, and data value stream. The PD work result is also information that is intangible and invisible. Work begins with information that is limited, which increases during the development process. In a design process, matters change constantly, with change driving the generation of information.

The main differences between manufacturing and PD identified in the literature are synthesized in following table:

	Manufacturing	Product Development
Work product	Physical objects and products	Information and virtual data
Value stream flow	Linear	Simultaneous and multidirectional
Work character	Repetitive process	Mainly non-repetitive
Variability	Destabilizes flow and creates wasteful inventory – needs to be eliminated	High variability, but necessary and beneficial
Requirements	Known in advance, product must conform to it	Created and modified in the process
Cycle time	Short (minutes, seconds)	Long (weeks, month, years)
Fixed cost	High	Low
Risk taking	Unnecessary	Essential
Capacity utilization	High	Very high (98.5%)
Queues	Visible in the form of inventory, manageable	Invisible, unmeasured and unmanaged
Inventories	Work-in-process inventory	Design-in-process inventory

	Manufacturing	Product Development
Resources	Few manufacturing disciplines	Large group of specialties from diverse technical disciplines

Table 4: Difference between lean manufacturing and lean development (Mund, 2014)

Furthermore, lean development centers on the customer benefit, or the value from its point of view as Oehmen et al. (2012) explained: Value is what the customer considers important and is willing to pay for. In simple jobs, the customer states what is required, and the contractor fulfils this, hopefully to the customer's satisfaction. This works well for simple products, but matters grow far more difficult when developing a new, complex technological system. In large engineering programs (such as government programs), thousands of stakeholders can be represented across numerous communities of users, acquisition participants, general contractors and suppliers along the entire value chain, in addition to other stakeholders, such as politicians, lobbyists, shareholders and banks, and so on. Stakeholders advocate those aspects of value that are important to them, often conflicting with the needs of other stakeholders.

However, it remains rather difficult to define the value from the customer's point of view. As Mascitelli described, "The problem with eliminating waste from projects is that it is hard to separate the waste from value … it is far easier to identify what is not value-added then to specify what is value added" (2002, p. 5). Herein lies the problem of value definition. Complex projects often possess a wide range of stakeholders and numerous different end-users, making a simplistic definition of "whatever the customer will pay for" insufficient. It is often difficult to create a meaningful and actionable statement of value in multi-stakeholder environments. Mascitelli thus distinguished between "strategic value" and "project value."

Instead of trying to define what universally adds value, a number of authors have found it more effective to identify what detracts from value in the form of waste. Several authors ((Oehmen & Rebentisch, 2010), (Morgan & Liker, 2006), (Locher, 2008) and (Oppenheim, 2011)) have adapted Ohno's seven production wastes for engineering programs.

A comparison was subsequently developed by Erne (n.d.):

Production	Project Management
Overproduction	Sloppy requirements management
Excess inventory	Failed or gold-plated outputs
Excess motion	Task-switching
Transportation	Unnecessary interfaces
Waiting	Waiting times
Overprocessing	Project bureaucracy
Defects or rework	Too much recursions and revisions

Table 5: Comparison of wastes in lean production and lean development (Erne, n.d.)

Pessôa et al. (2007) described countermeasures for product development:

Waste	Product Development Perspective
Overproduction	Synchronize the information and resources capability and schedule. Use pull events instead of phase gates.
Inventory	Define clearly what, when and who will perform each task. Execute resource leveling.
Unnecessary movement	Avoid micro planning that may lead to information inconsistency.
Transportation	Define optimized information flows (what, when, to whom and how). Avoid multitasking.
Waiting	Include only dependencies that represent the value flow.
Processing	The project network must include all and only the activities from and to support the value flow. Guarantee the allocation of the right people and materials.

Waste	Product Development Perspective
Defective product	Create a verification and validation plan to check the right value delivery.

Table 6: Countermeasures for waste in lean development (Pessôa et al., 2007)

Erne further applied the positive principles of the "lean" idea to project management (Erne, n.d.):

General	Project Management
• **Value**: Specify precisely the value of your product for your customer	• Clearly define who the customer is and what is important to him.
• **Value stream**: Recognize the value stream	• Do what creates value for the customer—and only that.
• **Flow**: Create a value flow without interruptions	• Focus on one result at a time and avoid second recursions.
• **Pull**: Lasse customers determine the pace of editing	• Deliver only when the customer needs it—but then deliver quickly.
• **Perfection**: Continually improve things	• Gradually and consistently eliminate all forms of waste in the project.

Table 7: Application of the lean principles to project management (Erne, n.d.)

Pessôa et al. (2007) described a method for lean product development planning. In their paper, they derived how, in a so-called “project activity network,” only the necessary information and materials are received by the development team by means of confirmation events. Here, the lean principles of value creation and waste avoidance are applied using the pull principle.

What is interesting here is that they are also moving away from the traditional WBS to a so-called “value breakdown structure” (VBS). This can be compared to the FBS in APM. In addition, the pull principle is also emphasized.

No process alongside the value flow should produce a part, service or information without direct request from subsequent processes. The pull events relate to physical proofs of progress (i.e., models, prototypes, start of production, etc.) and represent significant knowledge-capturing moments. In contrast to stage-gates, where information batches are created, pull events ensure value flow, make quality problems visible and create knowledge (Pessôa et al., 2007).

4.6 CONCLUSIONS FOR THIS RESEARCH

The lean philosophy plays a somewhat different role for later modelling of the various project management approaches. Lean development in the sense described so far does not represent a PM method, but rather a transfer of lean production thoughts, values and principles to the development process. However, some of the principles can be transferred effectively into the simulation. The pull principle in particular, in the form of the one-piece flow, represents a special feature that is well suited for comparison to the TPM.

5 SUCCESS OF A PROJECT

5.1 PROJECT SUCCESS VERSUS PROJECT MANAGEMENT SUCCESS

Any literature on the topic of project success always refers to the so-called "iron triangle." This can be to elaborate on this in greater detail or to illustrate that this is not sufficient for the evaluation alone. The iron triangle consists of the cornerstones cost, time and quality (and sometimes content or performance instead of quality).

Kerzner (2009), for example, mentioned customer satisfaction as a fourth criterion in addition to time, cost and performance: "Time, cost, and performance are the constraints on the project. If the project is to be accomplished for an outside customer, then the project has a fourth constraint: good customer relations" (p. 5).

Compared to this, the IPMA Competence Baseline 3.0 defined success as "the appreciation by the various interested parties of the project outcomes." This definition is more challenging than "to produce the project deliverables within time and budget, which is only part of it" (Gaupin & International Project Management Association, 2006, p. 16).

Similarly, for Lester (2014), the criteria for achieving the project's objectives include on-time completion within the set budget and meeting the quality requirements.

Gardiner and Stewart (2000, p. 251), however, doubted that the measures "on time," "to budget" and "of the required quality" constitute an appropriate or satisfactory measure of project success. They connected costs and time and recommended changing the performance measures to "the project was delivered with the best achievable Net Present Value (NPV) and to the required quality" (p. 252). However, they remain within the definition of project management success if they only connect time and budget.

More generally, the IPMA Baseline formulated this as "To achieve the project, programme or portfolio objectives within the agreed constraints is the overall definition of success" (Gaupin & International Project Management Association, 2006, p. 40).

Lester (2014) expanded the triangle with a fourth component—the security aspect—thus creating a "diamond." He argued that safety represents the most important criterion for, as an example, airplanes, trains or cars. It is more important than cost, time and performance, and must never be compromised. However, this definition also remains within the frame of project management success.

Kerzner (2009, p.60) noted that a project can also be successful if the original specifications for time, costs and quality are not met:

> *"Success is defined as a point on the time, cost, and quality/performance grid. But how many projects, especially those requiring innovation, are accomplished at this point? Very few projects are ever completed without trade-offs or scope changes on time, cost, and quality. Therefore, success could still occur without exactly hitting this singular point."*

The IPMA Competence Baseline also related success to the stakeholders: "Project management success is the appreciation of the project management results by the relevant interested parties" (Gaupin & International Project Management Association, 2006, p. 40).

A difference exists between project success and project management success. Lester reflected the classical example of several literatures:

> *"Success criteria can of course be subjective and depend often on the point of view of the observer. Judged by the conventional criteria of a well-managed project, i.e., costs, time, and performance, the Sydney Opera House failed in all three, as it was vastly over budget, very late in completion, and is considered to be too small for grand opera. Despite this, most people*

> *consider it to be a great piece of architecture and a wonderful landmark for the city of Sydney." (Lester, 2014, p. 30)*

However, the opposite, where a well-managed project delivers a product that nobody wants to buy, can also be easily imagined. Atkinson stated the following to this effect:

> *"Doing something right may result in a project which was implemented on time, within cost and to some quality parameters requested, but which is not used by the customers, not liked by the sponsors and does not seem to provide either improved effectiveness or efficiency for the organization, is this successful project management?" (Atkinson, 1999, p. 338)*

Likewise, Munns and Bjeirmi (2015) distinguished between the success of the project and successful project management. They argued that a project can be successful even if the project management failed, and vice versa.

Baccarini (1999) developed a four-level hierarchy of objectives and measured success according to how well these objectives were met. The first two levels—project goal (long-term goals with regard to the company's strategy) and project purpose (medium-term customer benefit using the project output)—represent why the project was implemented at all. While these two levels remain outside the project team's influence, it is responsible for the next two levels:

- Project output: These describe the immediate, specific and tangible results or deliverables produced by project activities. The outputs explain what the project will produce.
- Project inputs: These resource inputs and activities are required to deliver each product. The activities explain how the project will be performed and are defined … by the WBS, responsibility chart, schedule and budget (Baccarini, 1999, p. 27).

The IPMA Competence Baseline also distinguished between project success and project management success:

> *"Project management success is related to project success; however, it is not the same. For example, it is possible to carry out successful project management work for a project that has to be terminated due to a new strategic direction being taken by the organization... the project is no longer relevant." (Gaupin & International Project Management Association, 2006, p. 40)*

Munns and Bjeirmi (2015) related the scope of success to the project life cycle:

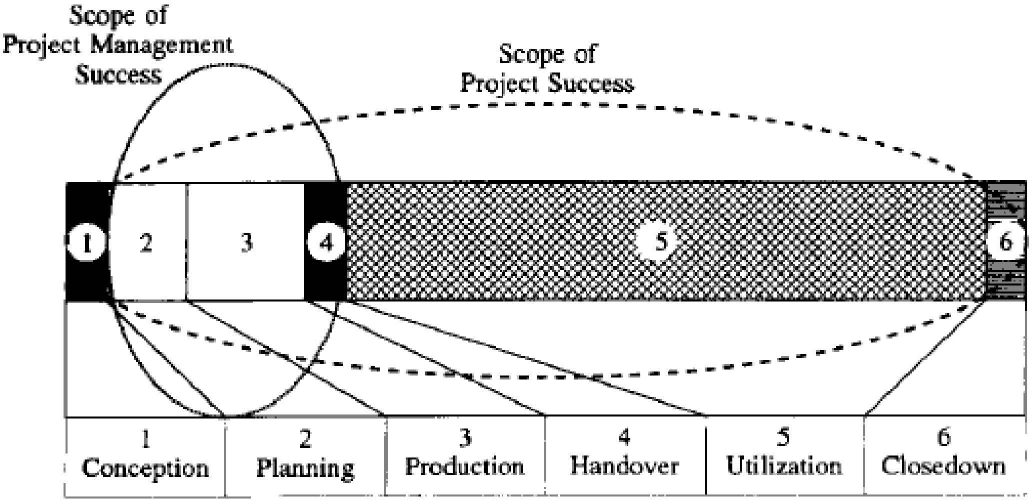


Figure 3: Scope of project management success vs. project success (Munns & Bjeirmi, 2015)

Baccarini (1999, p. 30) defined two distinct components of project success: project management success, which focuses on project process (meaning the successful accomplishment of cost, time and quality objectives), and product success, which concerns the effects of the project's end product. He further mentioned that "project management success can influence the achievement of product success."

Serrador and Turner (2015) analyzed the extent to which project efficiency ("meeting cost, time, and scope goals") correlates with overall project success, concluding that "project efficiency is an important contributor to stakeholder satisfaction and overall project success" (p. 83).

Baccarini (1999) also mentioned that success is affected by time. The product's success can be assessed after the product has been used by the customer, which can be years after the project's completion. On the other hand, successful project management—or the achievement of objectives in terms of costs, time and quality—can be measured directly at the project's conclusion.

Shenhar and Dvir (2007) reinforced the concept of project success linked to efficiency and effectiveness. Adherence to deadlines and budget targets demonstrates that a project has been managed efficiently. However, most projects are part of their organization's strategic management and need to be evaluated against their contribution to business performance (effectiveness) (see also (Osorio, Quelhas, Zotes, Shimoda, & others, 2014))

Shenhar and Dvir (2007) also related their five project success dimensions to time:

Success Dimension	Measures	Time
Project efficiency	Meeting schedule goal Meeting budget goal	End of project
Team satisfaction	Team morale Skill development Team member growth Team member retention	End of project
Impact on the customer	Meeting functional performance Meeting technical specifications Fulfilling customer needs Solving a customer's problem The customer is using the product Customer satisfaction	Month following project
Business success	Commercial success Creating a large market share	Years following project
Preparing for the future	Creating a new market Creating a new product line Developing a new technology	Years following project

Table 8: Measures of project success related to time (after Shenhar & Dvir, 2007, p. 26-32)

De Wit (1988, p. 164) has often been cited for distinguishing between project success and project management success:

> *"Good project management can contribute towards project success but is unlikely to be able to prevent failure. The most appropriate criteria for success are the project objectives. The degree to which these objectives have been met determines the success or failure of a project. The criteria for success of the project management effort tends to be restricted to cost, time and quality/performance. When measuring project success, one must consider the objectives of all stakeholders throughout the project life cycle and at all levels in the management hierarchy."*

Shenhar and Dvir (2007) stated that satisfying customer needs represents a more important project objective than adhering to deadlines or budgets.

Davis (2016) also mentioned the time aspect: Stakeholder perception influences whether the result is perceived as a success, and the success analysis timing could turn the result into a perceived failure.

5.2 CRITICAL SUCCESS FACTORS FOR PROJECT SUCCESS

Concerning the factors contributing to the project's success, Bhuiyan (2011) highlighted the two components of time and meeting customer requirements. It is essential for success to drive product development to market launch as quickly as possible, and to ensure that the product prototype or final design actually meets customer requirements. This requires gathering customer feedback throughout the entire development phase.

Regarding the time factor, Bhuiyan (2011) explained that faster development reduces the influence of a changing environment. This applies to changing customer requirements, new products from competitors and changing markets. This enables achieving better prices, a faster learning effect and the company as a whole gaining a competitive advantage.

In the literature, numerous other factors behind a project's success or failure have been discussed. The most common questions concern choosing the right project, balancing conflicting stakeholder interests and reacting appropriately to changing market conditions. In this thesis, however, the time aspect is primarily dealt with, since this is also important for project management success. As such, this is considered in more detail in the later simulation.

5.3 CRITICAL SUCCESS FACTORS FOR PROJECT MANAGEMENT SUCCESS

Derived from his four-level structure, Baccarini (1999, p. 28) identified three key components of project management success:

- Meeting time, cost and quality objectives (project outputs and inputs)
- Quality of the project management process
- Satisfying the project stakeholders’ needs where they relate to the project management process.

The author further suggested how time, cost and quality success can be measured: While time and costs can be measured as a percentage deviation from the plan, quality is understood as conformance to functional and technical specifications (Baccarini, 1999).

Osorio et al. (2014) related “efficiency” to project management success and identified the following success factors, among others: experience and competence of the project manager and team, clearly defined scope, sufficient resources, realistic time planning and budget, effective risk management and project monitoring.

5.4 STAKEHOLDERS

The IPMA Competence Baseline (Gaupin & International Project Management Association, 2006) stated that the project outcome may be perceived as more successful by some interested parties, but less successful by others.

Given that the success of a project or project management is determined by the assessment or perception of interested parties or stakeholders, a stakeholder analysis needs to be performed.

Baccarini (1999) related project success (product success) and project management success to the stakeholders. Both components of project success—product success and project management success—must correspond to the stakeholders' satisfaction when their interests relate to these components.

The IPMA Competence Baseline (Gaupin & International Project Management Association, 2006, p. 42) further defined "interested parties" as "people or groups who are interested in the performance and/or success of the project, or who are constrained by the project."

Lester (2014, p. 33) used the term "stakeholder," describing stakeholder analysis as listing, classifying and assessing different stakeholders' influence. These can include primary groups (i.e. directly involved in the project), such as the client, project sponsor, project manager, members of the project team, technical and financial services providers, internal or external consultants, material and equipment suppliers, site personnel, contractors and subcontractors, or end users. However, secondary, indirect groups are relevant as well, such as the personnel department, but also regulatory authorities, such as national and local government or licensing and inspecting organizations.

Kerzner (2009) stated that "Stakeholders are individuals or organizations that can be favorably or unfavorably impacted by the project" (p. 6). Stakeholders can also influence the project's progress or direction. Each stakeholder possesses its own goals, so the project manager has to balance these different interests.

Regarding stakeholder perception of success, Davis (2016) mentioned that differences in perception exist between stakeholders, and that some groups share no common success dimension.

5.5 TRADE-OFFS

Given that success is perceived and each project possesses a multitude of stakeholders, each with their own subjective perception of success (Baccarini, 1999), success criteria must be prioritized. Success criteria can collide, however, meaning compromises must often be agreed upon by all parties before the project begins (Baccarini, 1999).

Normally, a project will not be executed as it was planned at the beginning, which can be seen as a problem. Therefore, the IPMA Competence Baseline includes a chapter about problem resolution. The majority of the problems that occur are likely to relate to the project's timeframe, costs, risks or outcomes, or an interaction among all four factors. Problem-solving options may include reducing the scope of project deliverables, extending the timeframe or providing additional resources (Gaupin & International Project Management Association, 2006). However, this decision must be coordinated with and approved by the interested parties.

De Wit (1988, p. 166) elaborated that the view on trade-offs can depend on the project phase:

> *"The emphasis on what is important in a project, changes from one phase of the project to the next. In addition, the cost, time and quality trade-off varies for each phase of the project. 'During the early phase of the project, schedule is of primary importance, while cost takes second place and quality third. Later in the project, cost becomes the controlling interest, with schedule taking a secondary role. After the project has been completed, schedule and cost problems are easily forgotten and quality becomes the key'[3]"*

Kerzner (2009, p. 738), on the other hand, viewed the following priorities as dependent on the life-cycle phase: During the concept, definition and production phases, and at the beginning of the project's operational phase, the compromise priorities are first cost, then time, and lastly performance. During the operational phase, the cost factor becomes more important over time, as does performance, both of which begin to decline. In this stage, the company strives to amortize its investment in the project, and therefore places value on cost control. Performance standards may have been compromised, and the project may fall behind schedule, but management will review the cost data to assess the project's success.

---

[3] (Avots, 1984) cited in (De Wit, 1988)

Shenhar and Dvir (2007) stated that if a person can think of how the project will appear to someone in the future, it will be clearer which aspects are most important.

From the field, Kerzner (2009, p. 529) reported that, while most people are willing to accept that costs may be exceeded, this does not apply to deadlines. This is likely due to how cost overflows are internally resolved, while scheduling issues remain open and customer visible.

5.6 CONCLUSIONS FOR THIS RESEARCH

This research investigates various approaches to project management. Therefore, an evaluation of the success of the project management itself would suffice for the first step. So, the classical "iron triangle" of time, budget and quality needs to be included within the evaluation. At the point in time when the product is finally developed and handed over to manufacturing, the comparison of the different project management approaches comes to a natural end, because at this point, the (R&D) project is completed. Baccarini (1999) wrote that projects end when the result is delivered to the customer, constituting the end of project management. Cooke-Davies (2002) stated that profits are not generated or realized by the project manager and project team; they require measures of operational management.

The later simulation represents a typical development project. As such, it ends with the project's completion. As described above, at this point in time, information is only available on the classic criteria of time, costs and quality.

However, Collins' comment that this is not meaningful, particularly for projects managed in an agile manner, should be noted. Specifically, Collins (2014, p. 525) stated that "the definitions of success are part of the problem." If success is measured by the achievement of the original specifications, measuring agile projects that aim to accommodate changes in objectives to achieve maximum business value for the customer will certainly be difficult. The measurements should be based on what the customer sees as valuable, and that this is constantly updated.

Serrador and Pinto (2015, p. 1040) analyzed how organizations' agile use affects two dimensions of project success: efficiency and overall stakeholder satisfaction against organizational goals. Their findings "suggest that Agile methods do have a positive impact on both dimensions of project success."

The distinction between project success and project management success, on one hand, and the typical end of a development project at the point in time when the product is handed over to the responsibility of manufacturing, on the other hand, apparently contradiction each other. This can be resolved using the hierarchy of project success and project management success (Baccarini (1999, p. 29): "Project management success is subordinate to product success."). Project management success, as a measurement of efficiency, supports the project success, as a measurement of effectiveness.

For the later simulation, project management success is measured using the output values for time and costs and the determined quality key figures. In contrast, the different handling of change requests, as well as the faster implementation of changed priorities in the APM, is mapped in the model as an input quantity, and thus cannot be evaluated as a result.

To compare the project management approaches, only the factors need to be evaluated, which are different. So, all the factors handling aspects outside the direct impact of the PM approach can be ignored (e.g., choosing the right project, management commitment to the project, etc.).

Additionally, it needs to be determined which factors are directly influenced by applying the principles of lean development, as well as how this differs between the PM approaches.

6 HYPOTHESIS

6.1 DERIVATION FROM LITERATURE / FORMULATION OF THE RESEARCH QUESTION

The preceding chapters discussed the state of research on TPM and APM, lean development, and project and project management success. All topics have already been covered extensively in the literature. This thesis thus strives to establish their relationship and compare the different project management approaches. In addition, the influencing parameters on the project were also investigated, and decision criteria for project success were determined. This leads to the research question of which project management approach offers greater prospects for a successfully accomplished project, and how do the project's framework conditions affect the result. The following hypotheses were developed based on this question.

6.2 HYPOTHESES

The previous chapters elaborated on the objectives of both the lean development approach and the APM in detail, as well as why these concepts promise greater project success. Both approaches not only aim at the success of the project or product, but should also make project management more efficient. This leads to the assumption that the classical success criteria for project management should also demonstrate an advantage over TPM. The first group of hypotheses derives from this:

I. The APM approach leads to more successful project management compared to TPM.
    A. APM leads to lower cost (meaning lower effort) of the project
    B. APM reduces project time

   C. APM leads to less rework (unnecessary iterations)

Of course, the question then arises as to whether this is generally the case, or whether certain project characteristics make one PM approach superior to another. The literature has discussed in detail that the APM approach is particularly suitable for projects beginning with greater uncertainties or where the environment is subject to more frequent or stronger changes.

In the second group of hypotheses, it is therefore assumed that APM's advantage should present itself in these projects especially, as well as in the question of project management success, as follows:

II. The more challenging the project's framework conditions are, the more the APM approach presents its advantages.
   A. Agile projects are less sensitive to increase in requirements in terms of effort.
   B. Agile projects are less sensitive to increase in requirements in terms of required project duration.
   C. Agile projects are less sensitive to increase in requirements in terms of required iterations of individual tasks.

All these hypotheses together relate to the research question of which project management approach promises the greatest success and under which conditions this applies. The simulation described in the following chapter therefore compares the different PM approaches and examines the effects of different project characteristics.

7 SIMULATION

7.1 THE USE OF SIMULATIONS IN SCIENTIFIC RESEARCH

Simulation offers an effective tool for investigating complex systems, allowing process flows with several parallel paths, overlaps and iterations to be represented. However, a simulation can only represent reality in a simplified manner, and the assumptions made must be reasonable. The model can be checked for whether actual values from practice are processed and the results are compared with reality.

Adler, Mandelbaum, Nguyen and Schwerer (1995) used simulation techniques to investigate development processes. Specifically, they utilized a discrete event simulation (as does this work) for performance analysis and identifying bottlenecks. Bassil (2012) investigated software development following a waterfall approach using simulation. Cho and Eppinger (2005) utilized a simulation model determining probabilities for lead time distribution using a design structure matrix. Furthermore, rework probabilities were also determined using a design structure matrix (Yassine, 2007).

All this work has illustrated that simulation represents a powerful tool for analyzing complex processes such as product development.

7.2 FRAMEWORK CONDITIONS OF THE PROJECT

Chapter 2.2 "Theory on New Product Development (NPD)" explained the framework conditions for a development project. The following parameters have been selected for the simulation:

- Newness

- Difficulty
- Complexity

Urgency was not chosen as a boundary condition, as the time used is a result of the simulation.

In addition, the development team's maturity was taken into account, as the literature often requires an experienced development team for the agile approach.

The following mentions a typical project, with "project" referring to that which serves the development of a complex technical product. This could include, for example, a car navigation system or a liquid chromatography system, as this reflects the author's personal experience. Such projects are not limited to software, but also consist of mechanical and electronic subprojects.

7.3 DESCRIPTION OF THE SIMULATION

7.3.1 General description of the simulation

The following sections are partly based on a publication of the author regarding the transfer of the PM approaches into a simulation. This was published in advance in (Engelhardt, 2019).

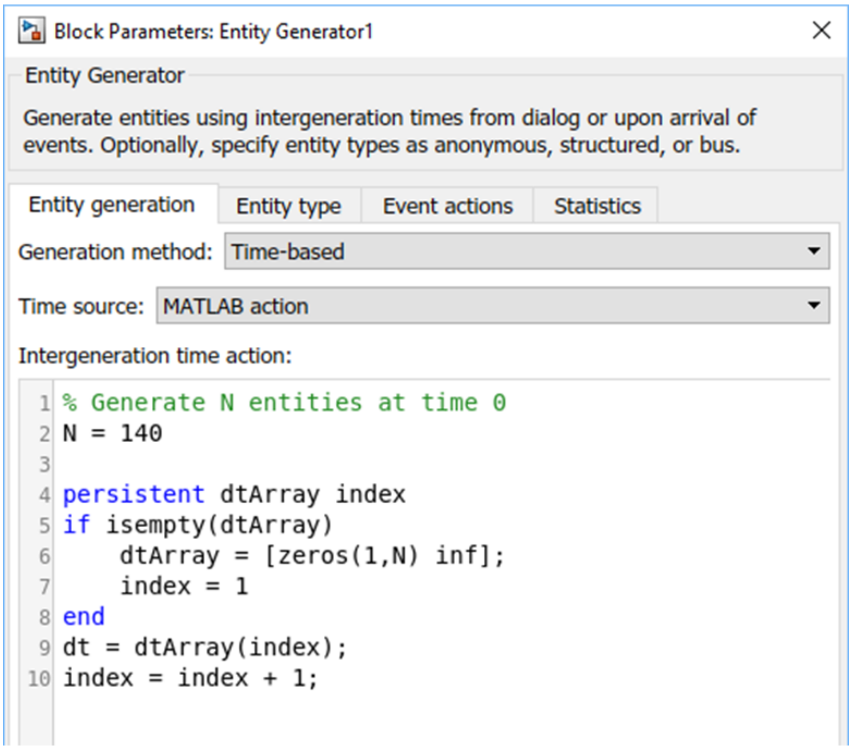
The program package Matlab/Simulink® Version R2017B was employed for the simulation. The module SimEvents® enables simulating discrete events, which were considered more suitable for modelling than continuous processes.

The complete model can be provided upon request. The detailed description of the model, all parameters and block properties, as well as the functions and calculations used, is available as an HTML and pdf file (>500 pages). Therefore, only exemplary screenshots are presented in the text for better comprehensibility.

SimEvents works with so-called "entities," which run through the simulation. The entity equivalent in project management is a defined development task, or a defined requirement or feature. In TPM, the specific tasks are determined at a sufficiently granular level of the WBS. In APM, the project is structured from the user's point of view, such as in terms of requirements or features. For comparability of the different approaches, transferability of both approaches is assumed. This results from the consideration that the total effort determined in the initial estimation should be the same, as should the associated uncertainties and resulting errors. Finally, APM tasks are also processed in order to implement the requirements. The term "entity" thus constitutes an abstraction, which can be applied in its meaning as a "task" for both approaches.

Different parameters with random values are assigned to each entity. These correspond to the framework conditions determined in chapter "2.2 Theory on New Product Development (NPD)" as possible influencing factors on the approaches' suitability, and are thus named in the hypotheses. Further parameters (e.g., for effort) allow a statistically more secure consideration due to their random values. Finally, certain auxiliary parameters still serve the respective models' functionality.

To be able to compare the different projects, it is necessary to work with the same input variables (e.g., effort or priority of the individual tasks) and values of the framework conditions. Therefore, a defined number of tasks is generated in the so-called "entity generator" (see Figure 4) in the next step, combined with initial random values of the parameters (called "attributes" in SimEvents) (see Figure 5 and Figure 6) and then copied for the different models. This is illustrated in Figure 7.



Generate entities using intergeneration times from dialog or upon arrival of events. Optionally, specify entity types as anonymous, structured, or bus.

Generation method: Time-based

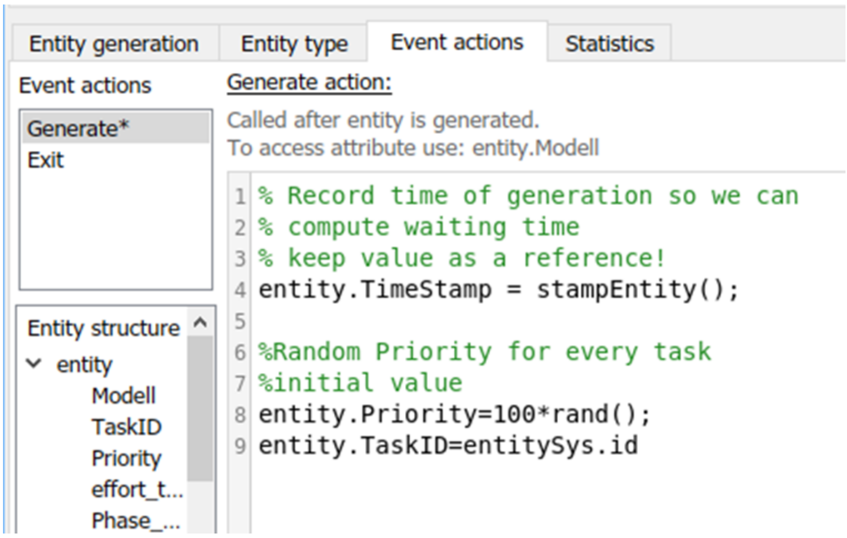
Time source: MATLAB action

Intergeneration time action:

```
% Generate N entities at time 0
N = 140

persistent dtArray index
if isempty(dtArray)
    dtArray = [zeros(1,N) inf];
    index = 1
end
dt = dtArray(index);
index = index + 1;
```

Figure 4: Generation of a defined number of tasks



Generate action:

Called after entity is generated.
To access attribute use: entity.Modell

```
% Record time of generation so we can
% compute waiting time
% keep value as a reference!
entity.TimeStamp = stampEntity();

%Random Priority for every task
%initial value
entity.Priority=100*rand();
entity.TaskID=entitySys.id
```

Figure 5: Initial identification with time stamp and randomized prioritization

Store entities in a queue. The block can queue entities based on arrival order or priority. The entity at the head of the queue departs when the downstream block is ready to accept it. You can specify the queue capacity.

Optionally, configure this block to receive entities from a multicast source with the same tag.

Called after entity has entered this block.
To access attribute use: entity.Modell

```

%entity.Newness - use for calculation of iterations;
%0 = everything is to 100 = everything is totally new
entity.Newness=100*rand();

%entity.Difficulty - use for calculation of quality issues (see Iter_Q)
%0 = very easy to 100 = very difficult
entity.Difficulty=100*rand();

%entity.Complexity - use for calculation of iterations (number and size)
%0 = very easy bis 100 = very difficult
entity.Complexity=100*rand();

%entity.Maturity - knowledge of the resource working on that task (reciprocal representation for easier calculation)
%use for calculation of quality issues
%0 = High professional bis 100 = total beginner
entity.Maturity=100*rand();

```

Figure 6: Exemplary initial parameterization in attribute manipulator

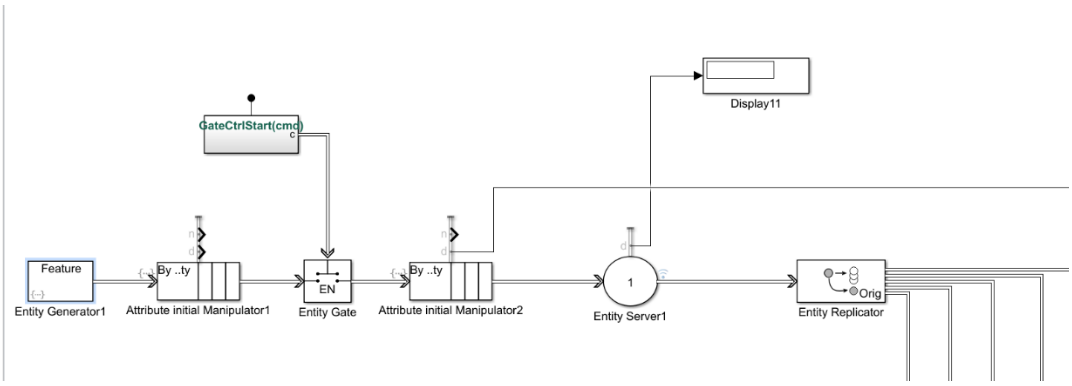


Figure 7: Generation, parameterization and replication of tasks

In the simulation model, five paths (i.e., sub-models) are mapped, corresponding to the different approaches in different characteristics (see Figure 8). These paths are called models one through five in the simulation. After branching, each task receives an attribute with the respective value for the corresponding model. This enables collecting data for all tasks and easily relating this to the respective sub-model.

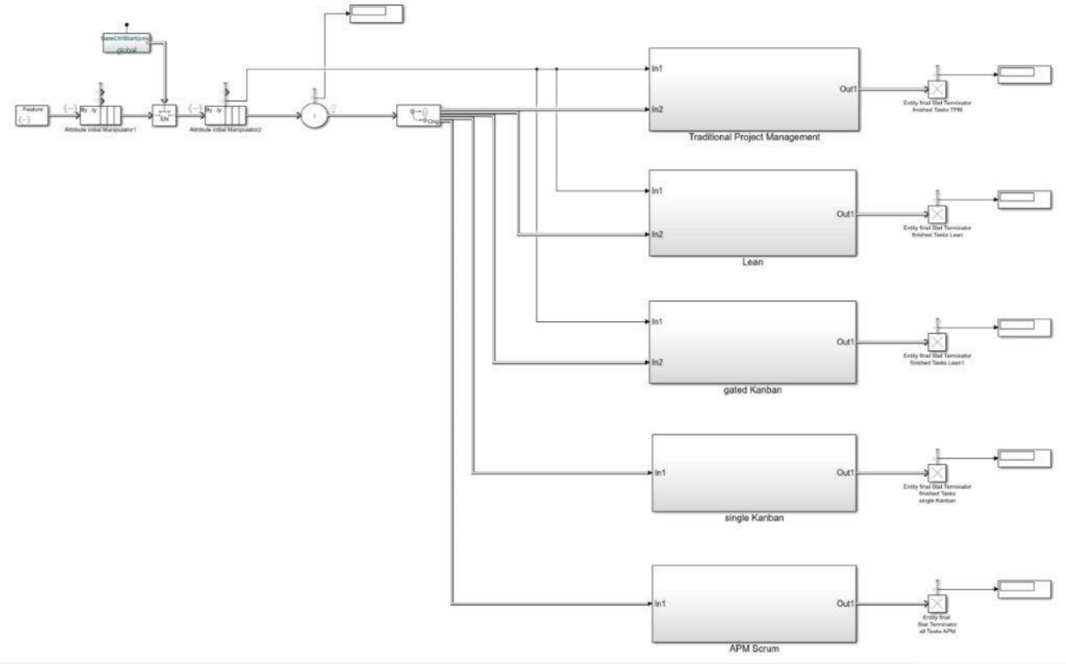


Figure 8: Parallel paths of the respective models after replication of the entities

The following models are presented:

Path/ Model	Name	Description
1	Traditional Project Management	Model of a traditionally structured project with several phases. The tasks are processed sequentially. Only when one development step has been completed is the next started.
2	Lean	Similar to the model one, “TPM," the model is structured in phases. The difference from model one is that, within the phase, the tasks are processed in the one-piece flow and are only collected for the review gate at the end of the respective phase.

Path/ Model	Name	Description
3	Gated Kanban	In the Kanban model, the number of tasks per development step is limited. The gated model collects the tasks at the end of the phase for the review, comparable to model two, "Lean."
4	Single Kanban	The model is structured within the phases exactly like the model three, "Gated Kanban." In contrast, however, the processed tasks are not collected for a review gate, but are evaluated immediately after processing.
5	Agile Project Management (Scrum)	This model represents the Scrum approach. The respective phases are not determined by the number of tasks, and the time per phase is exactly defined and is not changed.

Table 9: Description of the sub-models of the PM approaches

While model one can be clearly assigned to the TPM and model five to the APM, given its representation of the Scrum method, models two, three and four reflect both agile methods and principles of lean development.

All paths utilize their own resource pool, including resources for developers, prototype builders and testers (see Figure 9). All corresponding resource pools possess the same content (resource amount), as well as the same settings (parameters for resource usability). This ensures comparability of the models and prevents mutual interference.

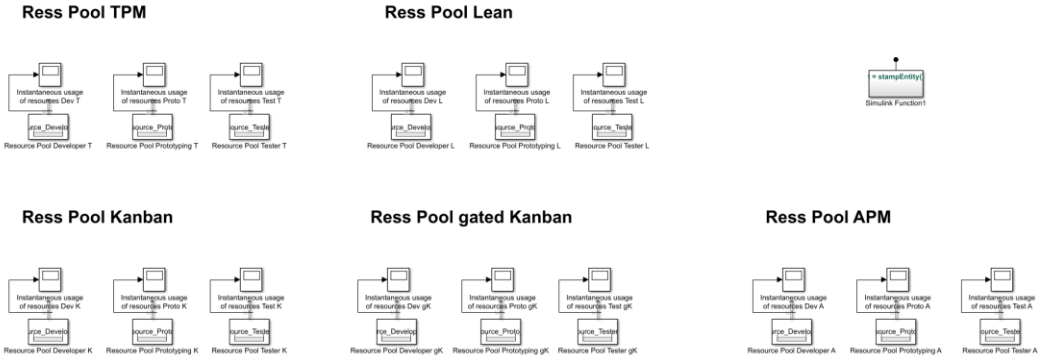


Figure 9: Resource pools for every sub-model

7.3.2 Structure of the development process and mapping in the simulation

7.3.2.1 General structure of the development process

For comparability of the different approaches, the development processing had to be presented on a more abstract level. It did not appear expedient to depict a concrete development project, since the results may not be easily transferred or generalized. In addition, a highly detailed structure defined at the beginning would be contrary to the APM’s agility principle. Therefore, the following generalization was made, which can be found in all approaches, both in the consideration of individual tasks and individual requirements as well as in the general processing of a development phase (see Figure 10).

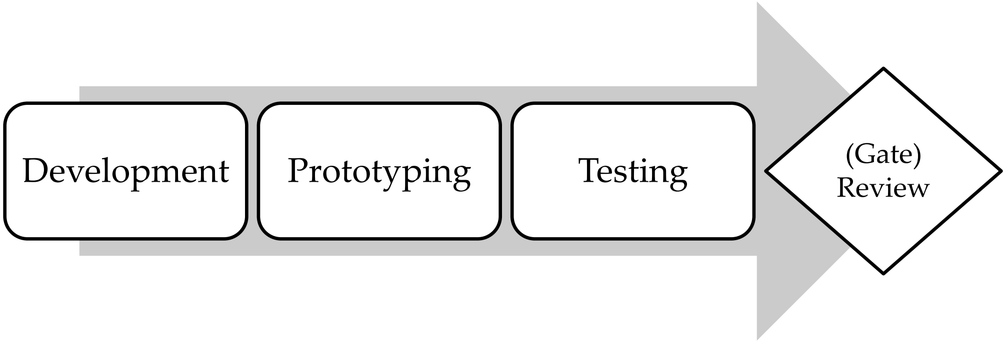


Figure 10: Generic sequence of the development process (own illustration)

In all models, equal consideration was given to the fact that errors can be found during testing that do not yet allow the development result to be passed on to the review. For this reason, an (internal) rework loop was generally built into the model in each phase, which also consists of development, prototyping and testing. The necessity, as well as the effort required, are derived from the parameters determined at the beginning and adapted in the course of the project. Details can be found in the explanation of the parameters provided in "7.3.4.2 Effort for development/prototyping/testing and rework."

Figure 11 presents an example of the procedure in the TPM (section for development, which also applies to prototyping and testing), which is used comparably in all models, apart from approach-specific special features (in this case, the entry gate, in which the tasks are collected). The resource allocation can also be seen here.

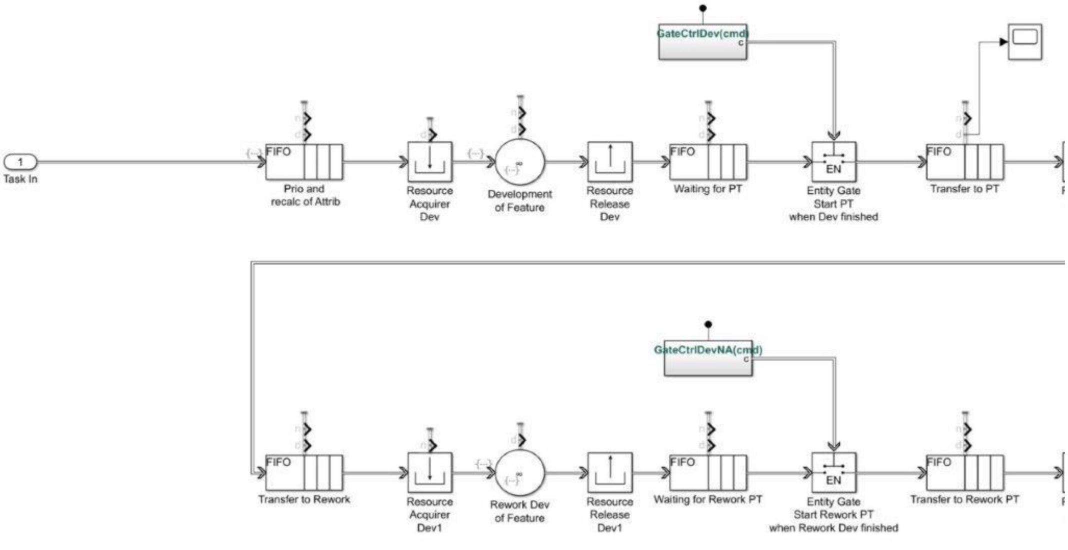


Figure 11: Illustration of the development and rework development steps

7.3.2.2 *Review gates*

Each task had to be reviewed after the end of each phase (whether a long development phase in TPM, a Scrum-sprint in APM, or in extreme cases of the

Single Kanban model, even directly after processing and providing the task). This procedure is the same in all approaches and is presented in Figure 12.

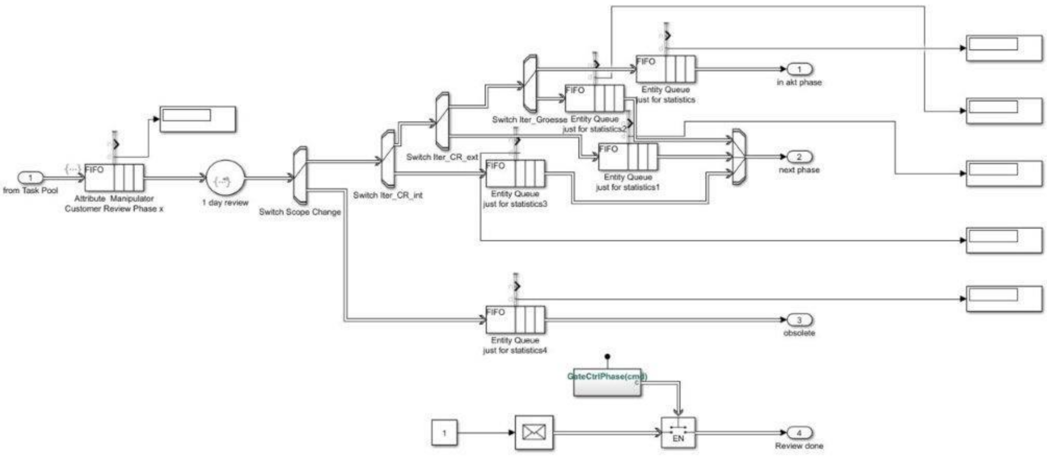


Figure 12: Gate review

The basic consideration here is that the review will decide whether a task

- has been implemented to a sufficient extent and quality, and can therefore be regarded as "done,"
- whether it must be processed again in the next phase due to defects, or
- whether the result cannot be used in its present form (“obsolete”).

This decision is represented in the model, in which the criteria for the individual decisions (i.e., the respective switches) are taken from the calculations resulting from the framework conditions derived in “2.2 Theory on New Product Development (NPD)” (and the corresponding entity parameters). The exact description of which framework condition (i.e. which parameter) and with which calculation rule the decision criterion is influenced can be found in “7.3.4.7 Reject rates – Calculation of the values for iterations or obsolete tasks and determination of the threshold for the acceptance of a task."

The sequence regarding a single phase with subsequent review and corresponding further processing of the evaluated tasks can be found in Figure 13.

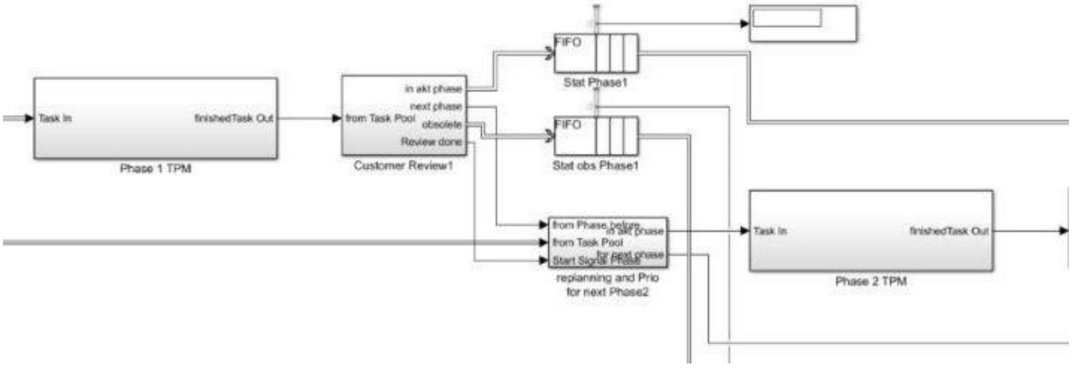
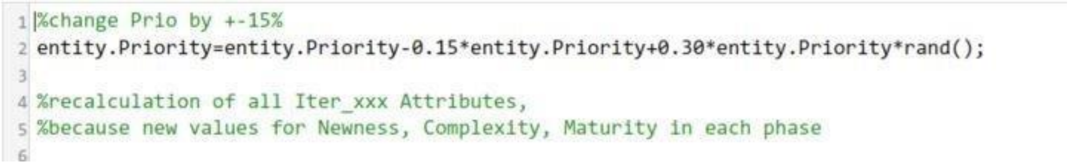


Figure 13: Development phase with subsequent review and dependent further processing

7.3.2.3 *Changing the framework conditions during the course of the project – Changing the simulation parameters*

Framework conditions change over the course of a project. For instance, uncertainties decrease as the project duration progresses. Knowledge about the project, its environment and its technical nature also increases. As a result, quality-related rework or obsolete tasks are becoming increasingly rare. Changes in the framework conditions or knowledge gain (learning effects) can also lead to a change in task or requirement priorities. This has been taken into account in the model. After each review, the tasks to be processed again, as well as the tasks not yet started, run in a block "Re-planning and Prio for next phase" (see Figure 13 again). In this block, the parameters for newness, complexity and maturity are adjusted accordingly—namely, they are improved such that the probability of further iteration decreases. At the same time, the initial effort estimates and priorities are changed using random numbers (see Figure 14 and Figure 15).



```
%change Prio by +-15%
entity.Priority=entity.Priority-0.15*entity.Priority+0.30*entity.Priority*rand();

%recalculation of all Iter_xxx Attributes,
%because new values for Newness, Complexity, Maturity in each phase

```

Figure 14: Randomized change of priorities

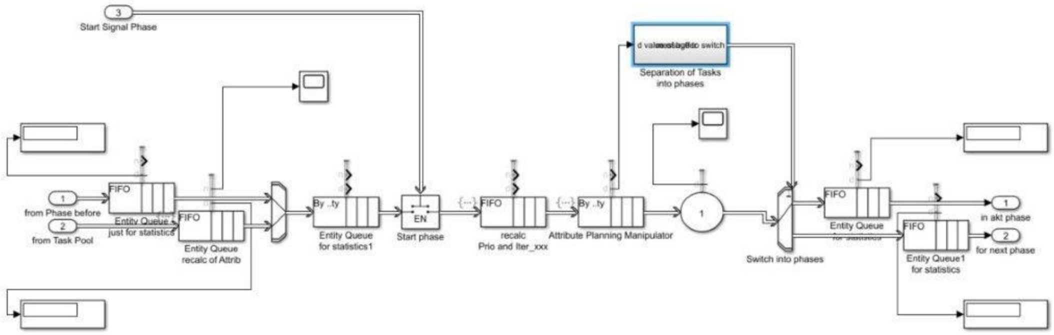


Figure 15: Adjustment of the parameters of the tasks (attributes of the entity)

This revaluation is conducted in the same way in all model paths. It is listed in the model description in the respective block.

7.3.3 Description of the implementation of the different approaches

The transfer of different PM approaches into corresponding models and their processing in the simulation were described in detail by Engelhardt (2019). Here, it was revealed that the PM approaches' specific properties can be transferred into a simulation, and that these can produce meaningful results.

7.3.3.1 Model 1 – Traditional project management

At the beginning of each phase, a number of tasks to be processed in this phase are defined (block "Separation of Tasks into Phases"). The tasks with the highest priority at this time are selected for this purpose. During the phase, a sequential procedure is employed. Each sub-step is run through by all tasks; the next sub-step only starts after all scheduled tasks have been processed. This is simulated by so-called "entity gates" (see Figure 16 for an example of the transition from development to prototyping).

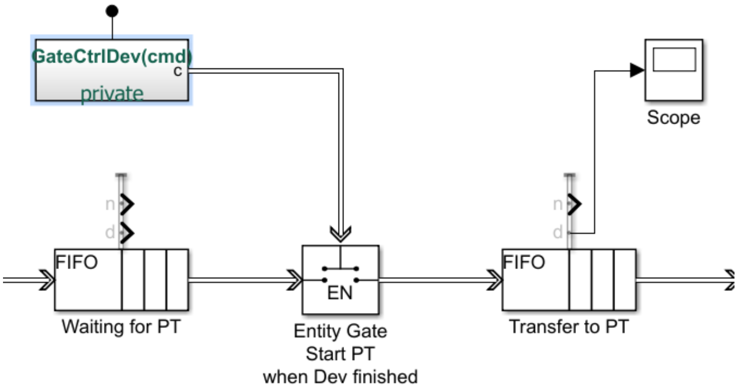


Figure 16: Gate for processing a sub-step in batch

At the end of each phase, the tasks are collected to review all the tasks processed in that phase. At the same time, the parameters for newness, complexity and maturity are adjusted (see Figure 17).

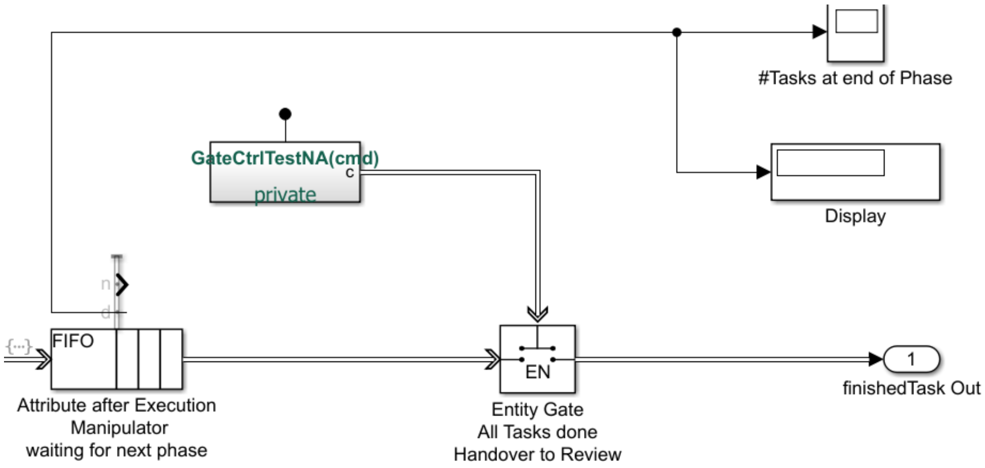


Figure 17: End of a phase in TPM

7.3.3.2 *Model 2 – Lean*

The special feature of the "lean" model is the presentation of the one-piece flow concept. A new task is taken over for processing when the resource is free again, this "pulls" the next task (implementation of the pull principle). Since the respective resources perform both initial processing and rework, both are taken into account in the model for the allocation of resources. A new task is therefore only started if the resource is not tied to rework (see Figure 18)

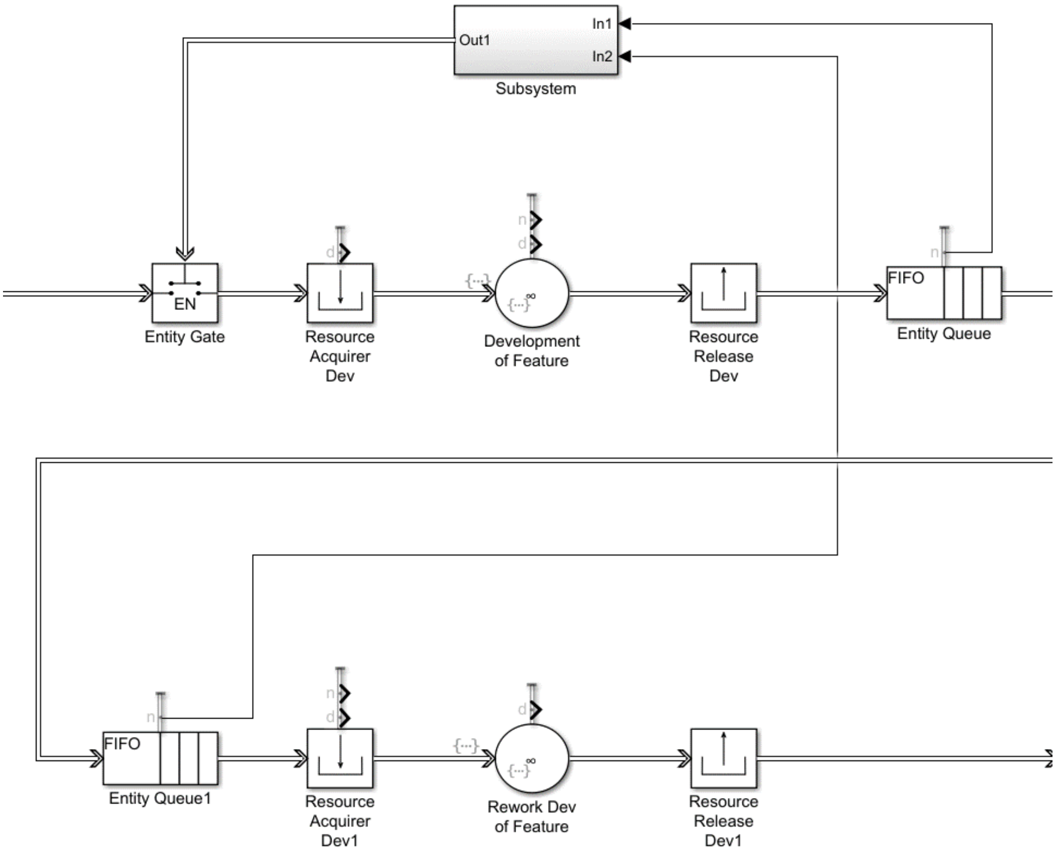


Figure 18: Implementation of the one-piece flow and pull principle in the lean model using the example of development

It is assumed in the model that the overall project nevertheless runs in phases (in particular with gate reviews) that correspond to those in TPM. Therefore, the tasks are collected for the review at the end of the phase. The planning of the tasks for the respective phase is modeled directly to the TPM.

7.3.3.3 Model 3 – Gated Kanban

The "Gated Kanban" model also assumes a phase model with review gates at the end of each phase. Here, too, a set number of tasks is defined at the beginning of the phase depending on their priority, which is to be processed. The difference from the two previous models concerns the implementation of the Kanban principle of a limited WIP. This means that only a defined number of tasks can be processed simultaneously. The gate at the beginning of the process opens only if the WIP limit is not reached (see Figure 19).

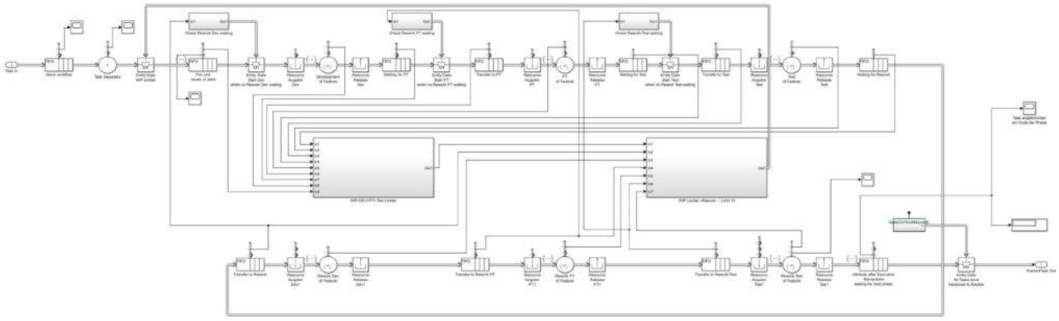


Figure 19: Implementation of the WIP limit in the Kanban model (detailed illustration of a phase)

7.3.3.4 *Model 4 – Single Kanban*

The single Kanban model takes the idea of the WIP limit one step further. The only thing left is to ensure that the WIP limit is not exceeded, but no more tasks are collected for the review at the end of a phase. As such, this is no longer a phase model with gates; each completed task is instead immediately reviewed and either evaluated as completed or fed back for processing (see Figure 20).

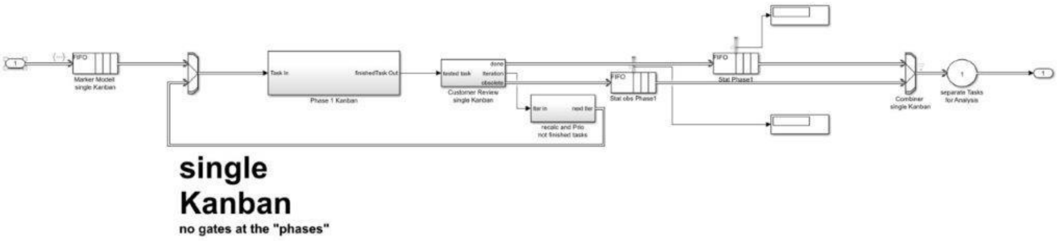


Figure 20: Implementation of the Kanban principle without gates at the end of the phase

In its rigor of implementation, this is certainly rare or never to be found in practice. However, since the APM literature does not specifically address the question of project phases and gates or else refers to hybrid models, this is helpful as a comparison.

7.3.3.5 Model 5 – Agile project management / Scrum

The Scrum approach was selected for modeling the APM. Its special feature involves proceeding in much shorter, precisely defined iteration steps than TPM. The duration of the individual phases is defined, and tasks that were not completed during this time are returned to the task pool (backlog). At the beginning of a new iteration, all tasks in the backlog are scheduled according to their priority for the new sprint. The duration remains fixed, representing the decisive difference from the other models. In APM, reviews occur at the end of each phase (here, sprint), evaluating the processing of the tasks (done, rework or change request, or obsolete).

The model in its overall view thus resembles the model "single Kanban" (see Figure 21).

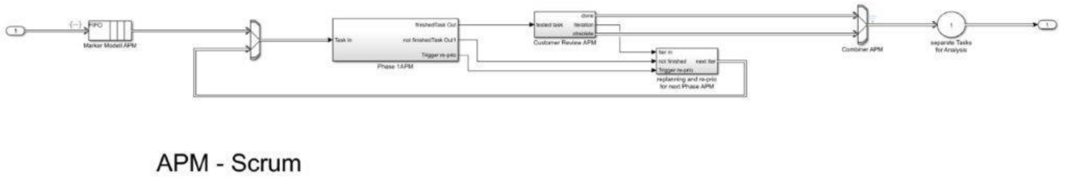


Figure 21: Overview of the APM-Scrum model

Within a phase (i.e., a sprint), it is ensured that the rework of a task is processed first in order to be allowed to set a task to "done." Afterwards, before each new task is started, it is checked whether the planned time for development, prototyping and testing remains less than the remaining time of the sprint (Check Dev+PT+Test in Iter), and thus whether completion within the iteration is possible. If not, it will not be started and goes directly to the backlog.

The time end is triggered by a timer (see Figure 22).

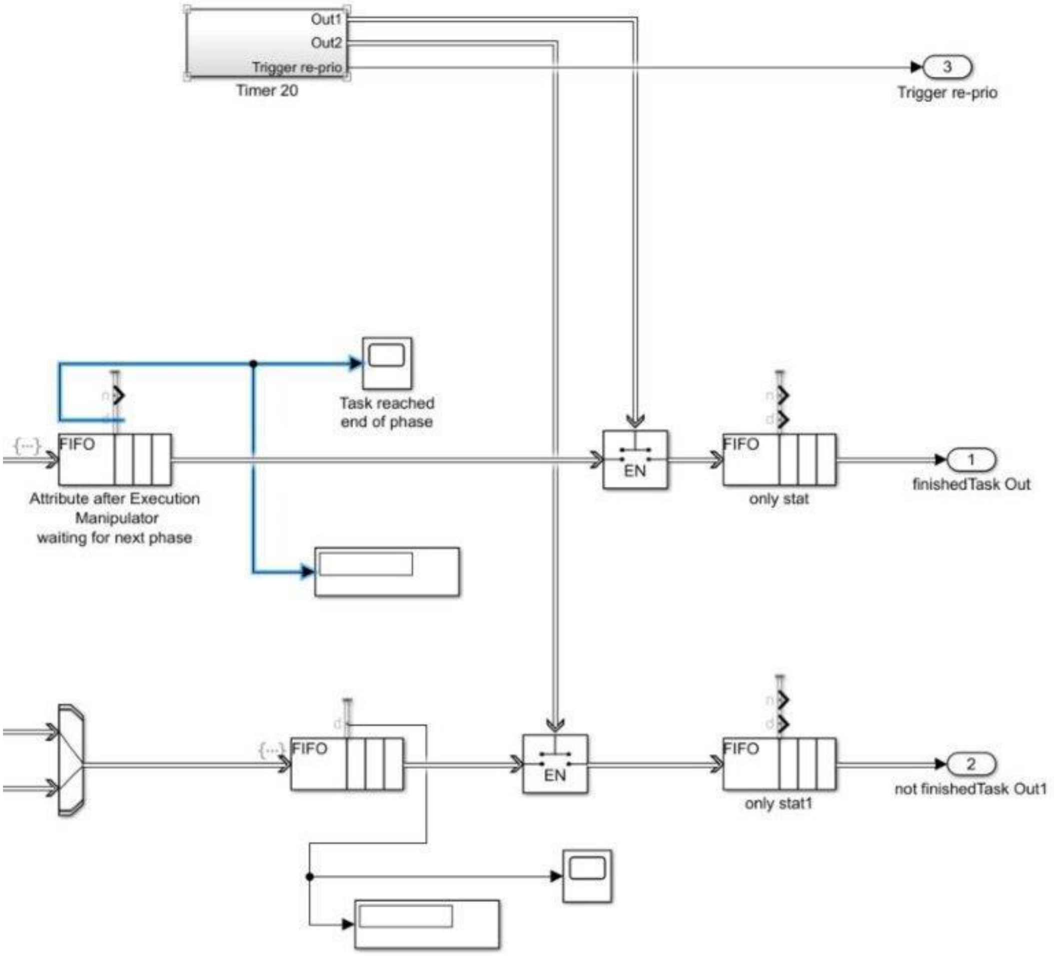


Figure 22: Timer for the end of a sprint (phase)

After the set time has elapsed, all completed tasks are placed into review, while the rest return to the backlog (see Figure 23).

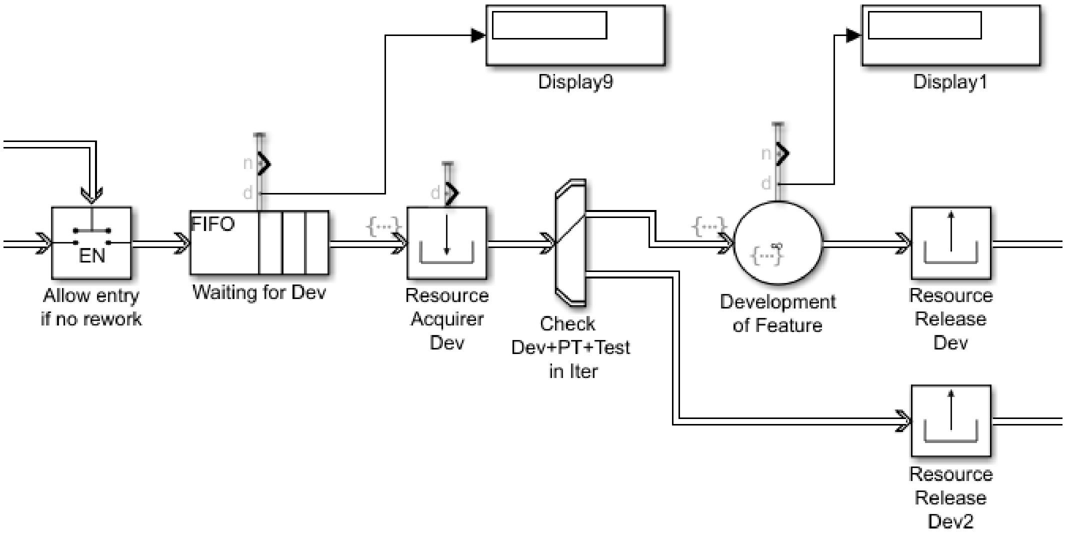


Figure 23: Modeling a sprint (section for development)

7.3.4 Assumptions for the models

7.3.4.1 Size of the resource pool

The number of resources for development, prototyping and testing was chosen so that the focus remains on development (five employees); prototyping (two employees) and testing (three employees) thus possess fewer resources. This corresponds to the usual distribution in development projects. The sum of 10 employees fits the desired team size in Scrum; in the traditional project environment, smaller to medium-sized projects can be handled.

The respective number of resources greater than one also prevents artificial bottlenecks caused by a single employee who is unavoidably necessary but may be tied up in a longer task.

When comparing the different approaches, it is important to work with the same values, but the concrete size of the resource pool is not relevant.

7.3.4.2 Effort for development/prototyping/testing and rework

The effort required to process a task depends heavily on the project's structure. The finer the project's subdivision, the lower the effort for the individual elements is. For the simulation, the concrete value is only of importance insofar as it must be the same in all models. This is ensured by the common generation and later replication of the tasks into the paths. The following consideration was made for the size: In APM or Scrum, sprints lasting two or four weeks are usual. In order to be able to plan meaningfully, a correspondingly smaller duration of the individual tasks results. A range granularity from single days up to one week is practicable; this can also be meaningfully used in practice in the TPM. For this model, an average duration of one day each for development, prototyping and testing is employed. Initial efforts are planned, which average five man-days for development, two man-days for prototyping and three man-days for testing. If these efforts are divided by the number of available resources per area, the average duration equals one day each (after five days, five developers completed five tasks, or completed one task each day together, constituting the same result). The initial planned value for the development effort varies between four and six (per random function), and the prototyping and testing values are changed in the same ratio (2/5 or 3/5 of the development effort). The effort for possible rework derives from the value "entity.Iter_Q," representing the probability of errors in the development process. For a detailed derivation of this value, see "7.3.4.7.2 Iter_Q – Quality problem, rework within the phase." This amounts to a maximum of 50% of the original effort.

7.3.4.3 Changing the initial values for effort during the course of the project

In the simulation, it was taken into account that the actual effort can and will differ from the initial planned effort. The greater the uncertainty at the beginning of the project, and the longer the period between original planning and actual implementation, the greater this deviation will be. The "newness" parameter is used as the determining factor for this planning uncertainty, meaning it is assumed that the less is known about a task or requirement, the greater the uncertainties, and thus possible deviations from the plan. In the calculation, a random factor was included that can result in a change of +-30% when recalculated (with maximum uncertainty, i.e., newness = 100%). This recalculation is performed for each task in each phase and can lead to significant deviations from the initial value until the end of the project.

7.3.4.4 Initial priority and its change during the course of the project

The priority determines the order in which the tasks are processed. It is assumed that prioritization of tasks occurs depending on the uncertainties or knowledge available at this point in time.

A simplification of the model compared to reality is that all tasks or features are considered independent from each other. As such, no logical order has to be kept for content reasons; only the tasks' priority determines the order. Since the priority in this simulation is rather an abstract term, and is understood as an agreement of all stakeholders regarding the desired order, it is assumed that content-related constraints have also been incorporated into the prioritization. The prioritization must have taken into account the interests of all stakeholders, including the consideration of urgency versus importance.

In the entity generator, initial priorities between 0 and 100 are randomly assigned. These priorities may change over the course of the project. Since these changed priorities usually result from influences outside the project, the priority is changed in the simulation using random factors. The calculation takes place in each phase before the tasks for processing are re-sorted according to the new prioritization. The priority changes by up to +-15%, a value that was chosen to be so large so that tasks that are already processed, but decided in the review as change/rework, may not be found in the next phase, as they have been downgraded accordingly. In the APM model, the priority changes after each sprint by up to +-5%, since the iterations are much shorter in time, and therefore, the external influences on the prioritization are smaller.

7.3.4.5 Number of tasks of the total project

Experience has indicated that projects lasting a total of two to three years, and usually incorporating four project phases of approximately half a year each, are common. The assumed average processing time for the simulation is thus approximately four days (three days for development + prototyping + testing, plus up to 1.5 days for rework). In order to be sufficiently statistically evaluable, 140 tasks are generated in the simulation at the beginning. This results in a project duration of 560 working days for a perfect project without any rework or obsolete tasks, corresponding to slightly more than two years. Here, too, the exact value is not relevant as long as it applies equally to all models. In order to better compare the simulations, it is also assumed that, for an obsolete task ("Not needed anymore"), a new task with comparable scope and parameters is introduced ("For this, we need this now."). Thus, at the end of the simulation, 140 tasks always arrive in each model under all given conditions, making the evaluation easier and more comparable.

7.3.4.6 Number of tasks in a phase TPM/lean/gated Kanban

In the models of the TPM, lean and gated Kanban approaches, a defined number of tasks are scheduled and processed in the respective phase. Starting from the usual four phases in a development project ("A-, B-, C- and D-sample" or concept, lab prototype, production prototype, ramp up) and the 140 tasks determined in the previous chapter for the entire project, a total of 40 tasks per phase appears suitable. This means that, despite possible iterations, all tasks can be completed within the planned time frame, provided that the project conditions allow this.

7.3.4.7 Reject rates – Calculation of the values for iterations or obsolete tasks and determination of the threshold for the acceptance of a task

7.3.4.7.1 General considerations on iterations

In the review, various criteria are provided for assessing the task as not completed, and thus feeding it back into the development process. The causes for such can come from within the organization (e.g., internal change request) or from the customer (e.g., external change request). The reasons are based on the influencing factors of newness, difficulty, complexity and maturity, as described in “2.2 Theory on New Product Development (NPD)." In the simulation, the parameter “urgency” (of the overall project) was omitted, since the tasks’ processing speed results from the respective simulation, and not the input parameter. The urgency of individual tasks compared to others is mapped in its effect via the priority, as described in "7.3.4.4 Initial priority and its change during the course of the project." The respective, possibly combined effect of the parameters on the iteration criteria is described in the following subchapters. The threshold values for the decision of whether a calculated value for a criterion is considered "done" or "not done" were chosen in such a way that, in a usual project (all values of the input parameters are equally distributed) usual reject rates (as been experienced in practice) occur.

7.3.4.7.2 Iter_Q – Quality problem, rework within the phase

The Iter_Q attribute evaluates the errors discovered and corrected within the phase—namely, purely technical implementation errors that occur during development or prototyping and can be corrected by minor rework. Its value determines the effort required for the phase-internal rework. The value Iter_Q is determined from the average of difficulty and maturity. This is based on the assumption that the probability and severity of these errors grows with increasing difficulty or decreasing experience of the developers. The value is between 0 and 100, with an average of approximately 50. This value is halved to determine the rework effort, so that rework is between 0 and 50% of the initial effort, with an average of approximately 25%.

7.3.4.7.3 Iter_CR_int – Change request for internal reasons

The attribute Iter_CR_int stands for a change request resulting from an internally caused change requirement. This means that something has been developed, prototyped and successfully technically tested, but the review indicates that the task cannot be accepted in this manner. This may be due to changed or misunderstood requirements from other departments, or the integration of multiple components may require a fundamental overhaul. Such interface problems tend to grow with increasing complexity of the development; therefore, the parameter "complexity" is used to determine the attribute's value. Its value is randomly distributed between 0 and 100, while the threshold value is 85. As such, if the complexity value is evenly distributed, approximately 15% of the tasks are rated as not accepted and must be fed back into one of the next phases or iterations. A more detailed consideration of this and the following threshold values is provided in chapter "7.3.4.9 Sensitivity analysis." Such an iteration is usually undesirable, because the project is extended unplanned and, in most cases, the customer will not be willing to pay for additional internal costs.

7.3.4.7.4 Iter_CR_ext – Change request for external reasons (customer requests)

In contrast to the previous attribute, Iter_CR_ext stands for a change request resulting from an externally caused change requirement (that is, a change requested by the customer). This is also only discovered in the review, meaning a requirement was implemented technically correctly, but is still not completely usable. This is either due to the requirement being imprecisely defined or misunderstood, or because the project environment has changed since the order was placed, thus requiring this change. The parameter "newness" is employed to determine the attribute’s value. This constitutes a direct effect of uncertainties resulting from the novelty for either the client (the customer) or the contractor (the developer). In addition, its value lies evenly randomly distributed between 0 and 100 with a selected threshold value 85, leading to an iteration of approximately 15% of the tasks. Such iterations may be financially less critical from a project perspective, since the customer will bear part of the costs, and—even expressly desired in APM—the effects on effort and project duration are nevertheless given and undesirable.

7.3.4.7.5 Iteration size (“Iter_Groesse”) – Combination of several parameters

It is also possible that not a single parameter alone is decisive for a required iteration, but instead, a combination of several influences leads to a problem that must be solved. The attribute Iter_Groesse is accordingly introduced for this simulation. This attribute combines the parameters complexity, difficulty and maturity, while the newness parameter is considered again in the next criterion. For the attribute, the mathematical average of the three parameters is determined and defined as the threshold value 65. It should be noted that only tasks for which complexity’s individual value remains below 85 are considered here, since these have already been filtered out in the CR_Iter_int criterion.

7.3.4.7.6 ScopeChange – Replacement of obsolete tasks

From a project perspective, the most undesirable case occurs when something has been developed that is no longer needed. In this circumstance, a task or requirement becomes obsolete. For this simulation, it is assumed that this can occur when significant uncertainties and considerable complexity occur at the same time, such as when the newness and complexity parameters together possess high values. Here, a threshold of 70 is employed for the average of both parameters, leading to a rather small number of obsolete tasks (see “7.3.4.9 Sensitivity analysis”). In order to not make a project appear shorter by mistake if these obsolete tasks are simply omitted, it is assumed that a comparable new task is introduced and included in the next iteration. However, the number of obsolete tasks is evaluated separately as a result of the simulation.

7.3.4.8 Changing the newness, complexity and maturity parameters

It is assumed that the framework conditions will change over the course of the project. Learning effects and decreasing uncertainty should significantly improve these parameters towards the end of the project. Therefore, correction factors that take this development into account were incorporated into the simulations. It is also assumed that not only do the values of the actually processed tasks improve, but the values of unprocessed tasks also profit from these effects, albeit to a lesser extent. A discussion of the concrete values can be found in the next chapter, “Sensitivity analysis."

Only the (objective) difficulty is assumed to remain constant, since the subjectively perceived difficulties are represented by the maturity (i.e., the developer’s knowledge and experience).

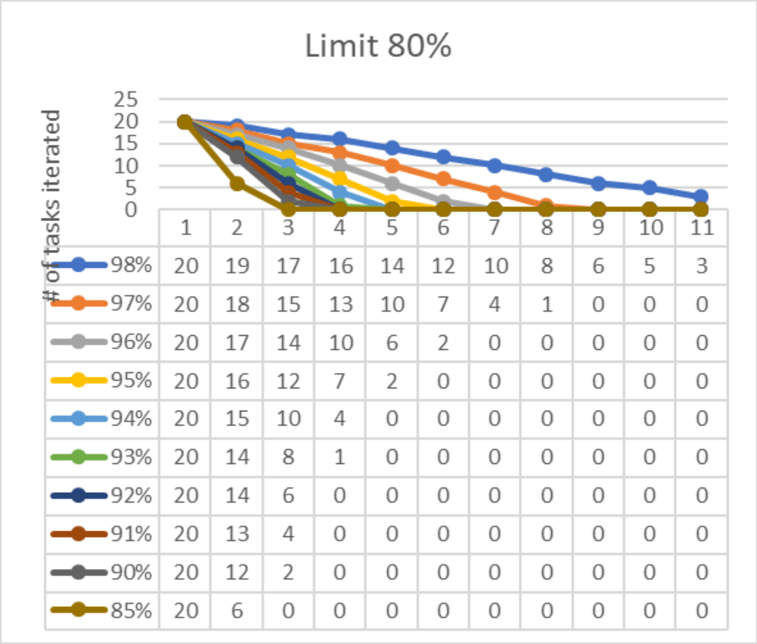
For newness, it is assumed that most uncertainties should be eliminated by the end of the project. A large reduction per iteration is therefore assumed for both the processed tasks (15%) and unprocessed tasks (5%).

A lower learning curve is assumed for both complexity and maturity; here, 7% improvement is assumed for processed tasks and 3% for unprocessed tasks.

7.3.4.9 Sensitivity analysis

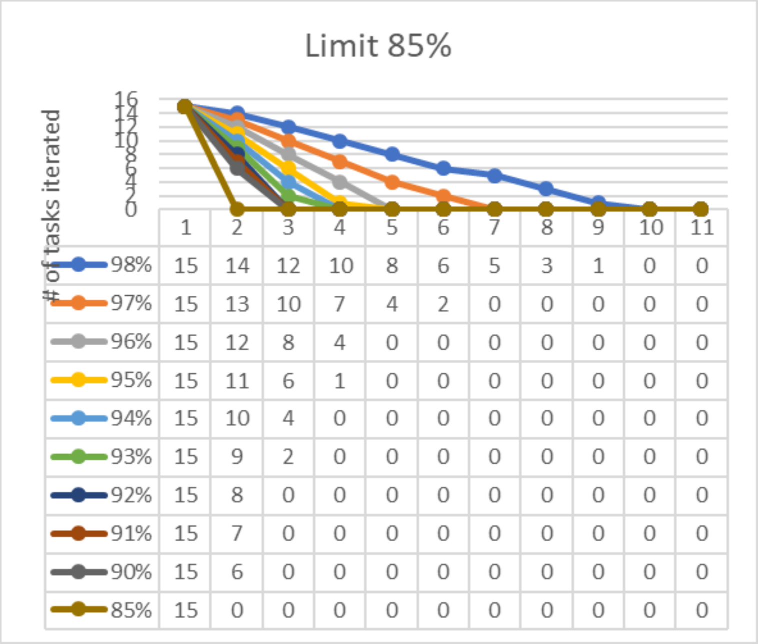
For the simulation, it is primarily important that the different approaches are comparable. This is ensured by how the iterations are calculated according to the same criteria and are based on the same parameters. Nevertheless, the iterations' frequency should correspond to the experience gained in practice.

For a task that has already been repeated, the probability of a new iteration is reduced. Here, too, the same values were used in all approaches. Nevertheless, the effects of different limits and learning curves were considered. Figure 24 through Figure 26 present the number of tasks per iteration frequency (for 100 tasks) for the limits 80%, 85% and 90% for the respective learning curves (e.g., 98% = reduction of the corresponding attribute's value, and thus of the distance to the threshold value, by 2%; other values correspondingly). As an example, it can be seen that, with a learning curve of 93% (i.e., knowledge gain 7%), at the 85% limit, 15 of 100 tasks have to be processed again, only nine tasks a second time, only two tasks a third time, and no task four times or more.



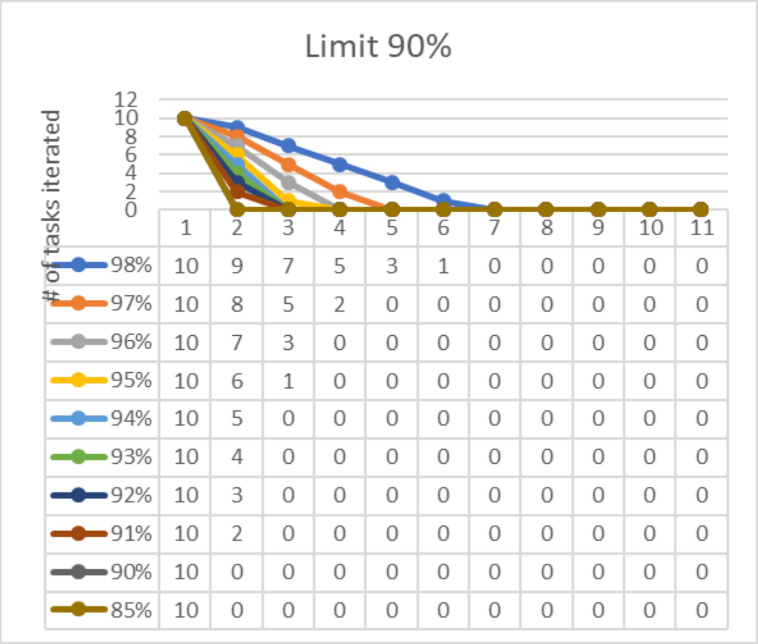
	1	2	3	4	5	6	7	8	9	10	11
98%	20	19	17	16	14	12	10	8	6	5	3
97%	20	18	15	13	10	7	4	1	0	0	0
96%	20	17	14	10	6	2	0	0	0	0	0
95%	20	16	12	7	2	0	0	0	0	0	0
94%	20	15	10	4	0	0	0	0	0	0	0
93%	20	14	8	1	0	0	0	0	0	0	0
92%	20	14	6	0	0	0	0	0	0	0	0
91%	20	13	4	0	0	0	0	0	0	0	0
90%	20	12	2	0	0	0	0	0	0	0	0
85%	20	6	0	0	0	0	0	0	0	0	0

Figure 24: Representation of necessary iterations as a function of the learning curve at a threshold value of 80%



	1	2	3	4	5	6	7	8	9	10	11
98%	15	14	12	10	8	6	5	3	1	0	0
97%	15	13	10	7	4	2	0	0	0	0	0
96%	15	12	8	4	0	0	0	0	0	0	0
95%	15	11	6	1	0	0	0	0	0	0	0
94%	15	10	4	0	0	0	0	0	0	0	0
93%	15	9	2	0	0	0	0	0	0	0	0
92%	15	8	0	0	0	0	0	0	0	0	0
91%	15	7	0	0	0	0	0	0	0	0	0
90%	15	6	0	0	0	0	0	0	0	0	0
85%	15	0	0	0	0	0	0	0	0	0	0

Figure 25: Representation of necessary iterations as a function of the learning curve at a threshold value of 85%



	1	2	3	4	5	6	7	8	9	10	11
98%	10	9	7	5	3	1	0	0	0	0	0
97%	10	8	5	2	0	0	0	0	0	0	0
96%	10	7	3	0	0	0	0	0	0	0	0
95%	10	6	1	0	0	0	0	0	0	0	0
94%	10	5	0	0	0	0	0	0	0	0	0
93%	10	4	0	0	0	0	0	0	0	0	0
92%	10	3	0	0	0	0	0	0	0	0	0
91%	10	2	0	0	0	0	0	0	0	0	0
90%	10	0	0	0	0	0	0	0	0	0	0
85%	10	0	0	0	0	0	0	0	0	0	0

Figure 26: Representation of necessary iterations as a function of the learning curve at a threshold value of 90%

It can be seen that the criterion Iter_CR_int (change request for internal reasons), which is based on the "complexity" parameter, leads to the following multiple iterations for the selected learning curves of 7% of processed and 3% of not processed tasks at the set threshold of 85%: for tasks in progress, 15 require at least one iteration; nine require at least two iterations; and two demand three iterations. For tasks that are not processed, the probability decreases with each project phase to 13/10/7/4/2 of 100 tasks.

The criterion Iter_CR_ext (change request for external reasons) is based on the parameter "newness," which changes faster, as described in chapter 7.3.4.8. With the set threshold value of 85% and an improvement of 15% for processed tasks, a task is iterated a maximum of once; for unprocessed tasks with an improvement of 5% per phase, even after 1/2/3 phases, only 11/6/1% of the tasks are affected by a possible iteration.

A comparable value can be found with Cho and Eppinger (2005), whose simulation reduced the rework probability by 50% in each iteration. In their study, however, no distinction is made between processed and unprocessed tasks, and the learning curve is also referred to in the first iteration only.

After conducting the simulation, it was examined how the iteration frequency actually behaves in relation to the threshold values in the respective models under different framework conditions.

7.3.5 Influence of the framework conditions on the simulation result

7.3.5.1 Permutations of parameters for various projects and their representation in simulation runs

As described in “7.2 Framework conditions of the project," the parameters newness, difficulty, complexity and maturity are employed to map different framework conditions of the projects. Since their influence on the project result in the respective approaches comprises the subject of the investigation, different characteristics of these parameters were simulated. The following values were accordingly defined:

- Low expression (LOW): random values between 0 and 50
- Mean expression (MID): random values between 0 and 100
- High expression (HIGH): random values between 50 and 100

The iteration criterion based on each of these parameters will therefore not be applied in the LOW expression, since the threshold value 85 is not exceeded. Conversely, this occurs in the HIGH expression with twice the probability compared to the MID expression.

Various simulation runs were conducted for the investigation, in which the parameters are available in different forms. These are referred to as runs Vxx. Since three variants exist for each of the four parameters, theoretically, 3^4 = 81 runs should have been executed and evaluated. In order to optimize the analysis effort, however, only runs possessing extreme values were examined, meaning all parameters possessing the values LOW or HIGH, and with either one parameter LOW and three parameters HIGH or three parameters LOW and one parameter HIGH. In addition, run v00 was defined with all parameters as the comparative value in the MID form. An overview of the different runs' parameter values can be found in Table 10.

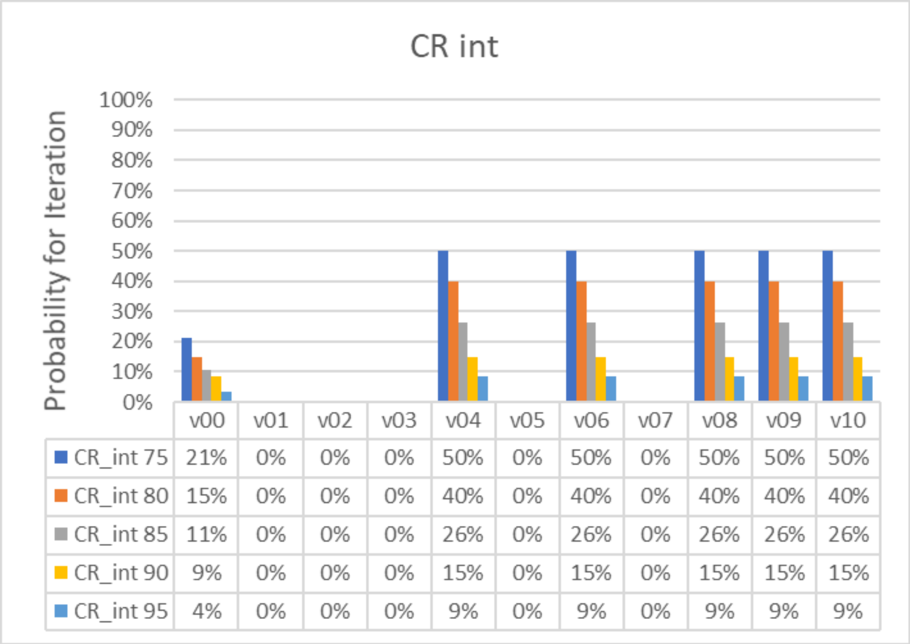
Run	Newness	Difficulty	Complexity	Maturity
V00	MID	MID	MID	MID
V01	LOW	LOW	LOW	LOW
V02	HIGH	LOW	LOW	LOW
V03	LOW	HIGH	LOW	LOW
V04	LOW	LOW	HIGH	LOW
V05	LOW	LOW	LOW	HIGH
V06	HIGH	HIGH	HIGH	LOW
V07	HIGH	HIGH	LOW	HIGH
V08	HIGH	LOW	HIGH	HIGH
V09	LOW	HIGH	HIGH	HIGH
V10	HIGH	HIGH	HIGH	HIGH

Table 10: Overview of the parameter characteristics of the various runs

7.3.5.2 *Sensitivity analysis depending on the framework conditions*

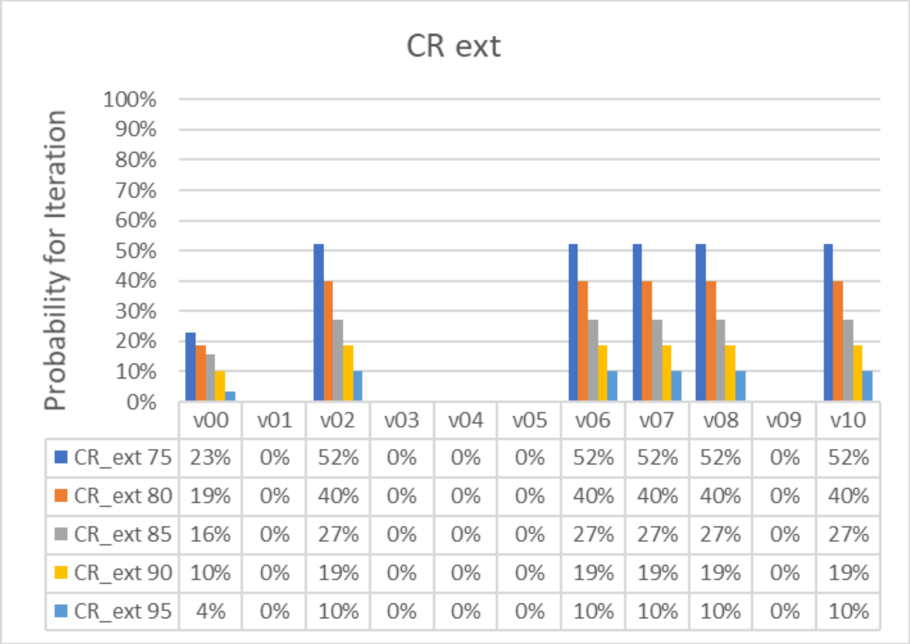
For this consideration, the tasks are analyzed before being split into the different approach models; this is called "Model 0." It appears that the actual probabilities for an iteration at run v00 (uniform distribution of all parameters in MID form) do not significantly differ from the expected values. Deviations result from the not too high number of tasks of 140 and their randomly assigned parameter values.

The evaluation is performed separately for each iteration criterion (see Figure 27 through Figure 30). The percentage corresponds to the proportion of tasks that would not be evaluated as "done" in the first review, meaning they would be put into a new iteration. The selected value for the simulation is the mean value; for comparison, two values above and below were used.



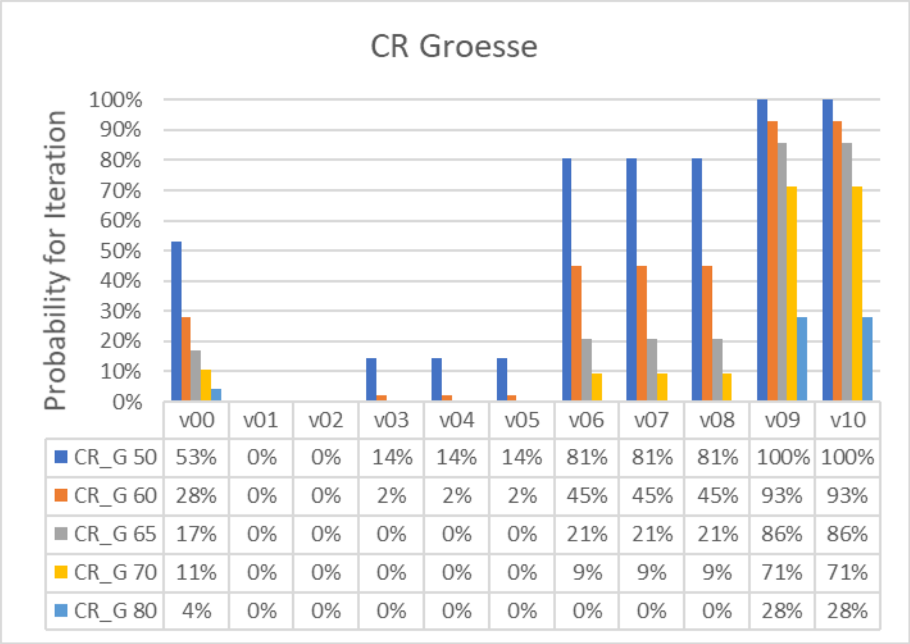
	v00	v01	v02	v03	v04	v05	v06	v07	v08	v09	v10
■ CR_int 75	21%	0%	0%	0%	50%	0%	50%	0%	50%	50%	50%
■ CR_int 80	15%	0%	0%	0%	40%	0%	40%	0%	40%	40%	40%
■ CR_int 85	11%	0%	0%	0%	26%	0%	26%	0%	26%	26%	26%
■ CR_int 90	9%	0%	0%	0%	15%	0%	15%	0%	15%	15%	15%
■ CR_int 95	4%	0%	0%	0%	9%	0%	9%	0%	9%	9%	9%

Figure 27: Sensitivity for Iter_CR_int



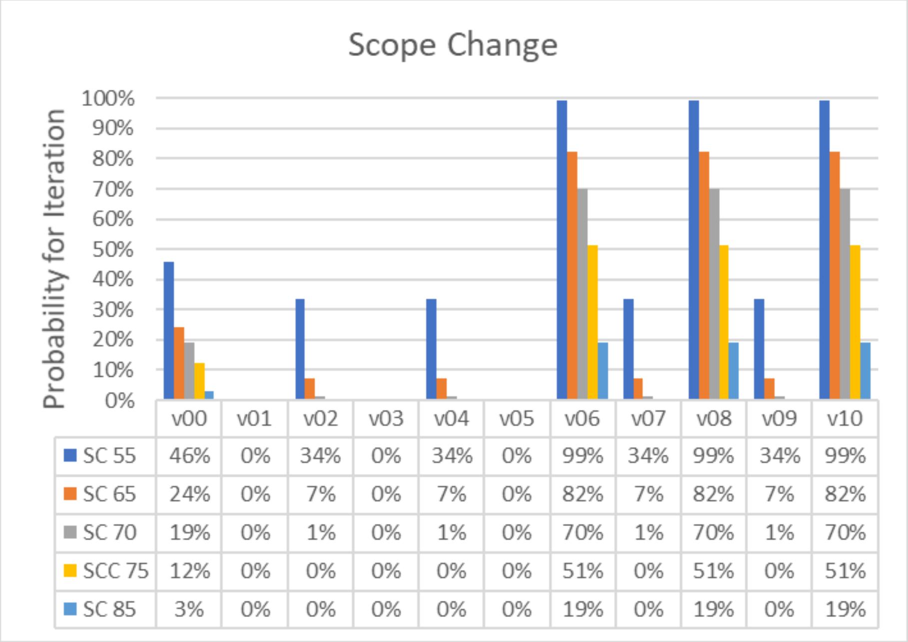
	v00	v01	v02	v03	v04	v05	v06	v07	v08	v09	v10
CR_ext 75	23%	0%	52%	0%	0%	0%	52%	52%	52%	0%	52%
CR_ext 80	19%	0%	40%	0%	0%	0%	40%	40%	40%	0%	40%
CR_ext 85	16%	0%	27%	0%	0%	0%	27%	27%	27%	0%	27%
CR_ext 90	10%	0%	19%	0%	0%	0%	19%	19%	19%	0%	19%
CR_ext 95	4%	0%	10%	0%	0%	0%	10%	10%	10%	0%	10%

Figure 28: Sensitivity for Iter_CR_ext



	v00	v01	v02	v03	v04	v05	v06	v07	v08	v09	v10
CR_G 50	53%	0%	0%	14%	14%	14%	81%	81%	81%	100%	100%
CR_G 60	28%	0%	0%	2%	2%	2%	45%	45%	45%	93%	93%
CR_G 65	17%	0%	0%	0%	0%	0%	21%	21%	21%	86%	86%
CR_G 70	11%	0%	0%	0%	0%	0%	9%	9%	9%	71%	71%
CR_G 80	4%	0%	0%	0%	0%	0%	0%	0%	0%	28%	28%

Figure 29: Sensitivity for Iter_CR_Groesse



	v00	v01	v02	v03	v04	v05	v06	v07	v08	v09	v10
■ SC 55	46%	0%	34%	0%	34%	0%	99%	34%	99%	34%	99%
■ SC 65	24%	0%	7%	0%	7%	0%	82%	7%	82%	7%	82%
■ SC 70	19%	0%	1%	0%	1%	0%	70%	1%	70%	1%	70%
■ SCC 75	12%	0%	0%	0%	0%	0%	51%	0%	51%	0%	51%
■ SC 85	3%	0%	0%	0%	0%	0%	19%	0%	19%	0%	19%

Figure 30: Sensitivity for scope change

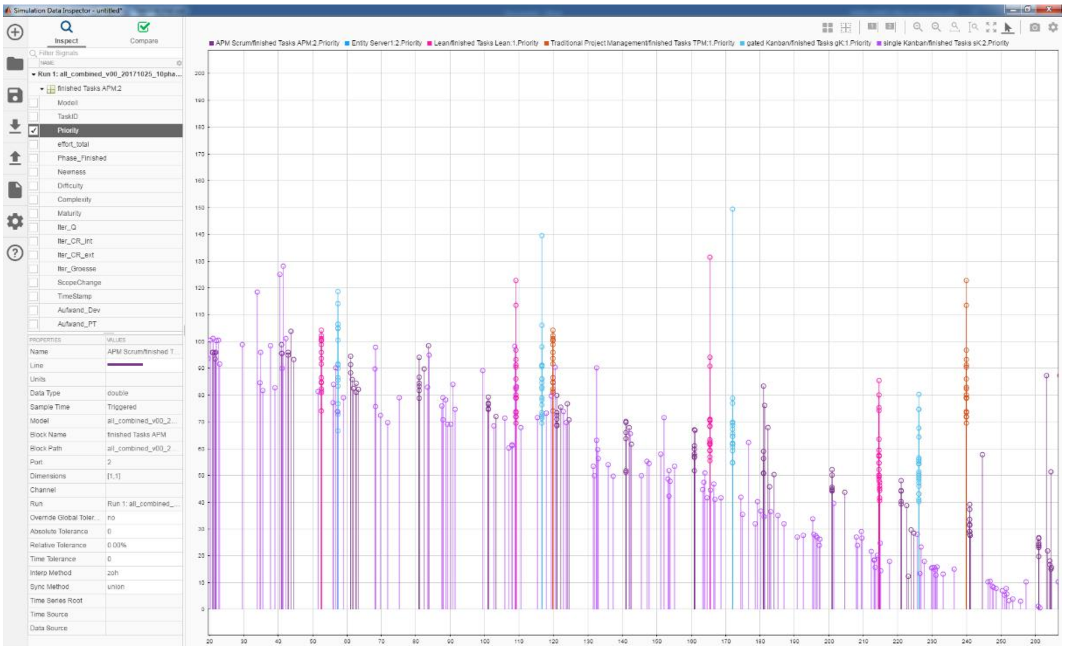
The demonstrated probabilities describe the value of each model's first phase. These probabilities decrease with each phase, as described in “7.3.4.8 Changing the newness, complexity and maturity parameters," so in the simulation, only a small portion of the tasks is affected by the high probabilities.

It can be seen that with the criteria Iter_CR_int and Iter_CR_ext the exact threshold value is not important for comparability of the different framework conditions (i.e., different "runs"; see “7.3.5.1 Permutations of parameters for various projects and their representation in simulation runs”); the effect of the parameters' different values behaves similarly in principle. For the CR_Iter_Groesse and ScopeChange criteria, the selected thresholds exhibit rather high values in their more extreme form of the runs, but this represents a middle course between the possibly large effect (which is the same for all models) and the perceptible effect in the other runs.

7.3.6 Collection of data

In the simulation, the data of each individual task was logged. For comparison, the attributes' values after initial assignment were recorded for each task ("Model 0"). At the output of each path of the corresponding models one through five, time stamps and attribute values were also recorded. The task ID assigned at the beginning also allows each individual task's effects in the respective approaches to be compared.

These values can be evaluated graphically, as presented in Figure 31. If required, the display can be zoomed in at will in order to examine individual values more closely.



PROPERTIES	VALUES
Name	APM Scrum/finished T...
Line	▬
Units	
Data Type	double
Sample Time	Triggered
Model	all_combined_v00_2...
Block Name	finished Tasks APM
Block Path	all_combined_v00_2...
Port	2
Dimensions	[1,1]
Channel	
Run	Run 1: all_combined_...
Override Global Toler...	no
Absolute Tolerance	0
Relative Tolerance	0.00%
Time Tolerance	0
Interp Method	zoh
Sync Method	union
Time Series Root	
Time Source	
Data Source	

Figure 31: Graphical evaluation of the output signals of a run using the priority parameter as an example

Due to the large amount of data (140 tasks in five models + model zero as comparison value for the start with 14 parameters each to be evaluated, and this in 11 runs (v00 to v10); i.e., 140*6*14*11=129.360 single values), the individual values were exported to a spreadsheet, and from there, transferred to a statistics program for evaluation.

7.4 SIMULATION RESULT

7.4.1 Cumulative view

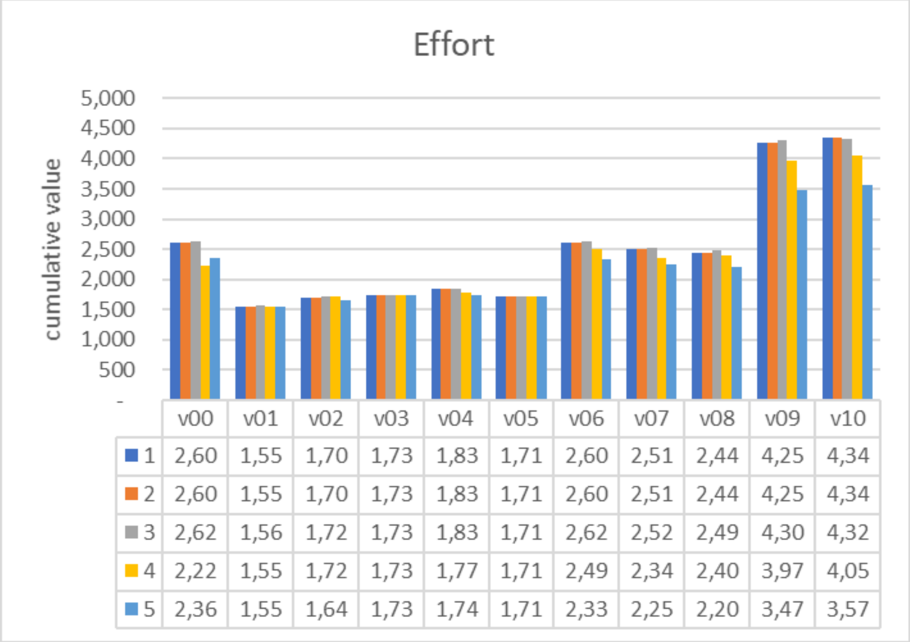
The initial evaluation examined how the different framework conditions (reflected in the runs) affect the number of project phases and the number of tasks evaluated as obsolete. For the simulation, it was assumed that the project is actually completed only after all tasks have been processed and is not aborted beforehand or terminated as "good enough." The number of phases is only meaningful for the models that follow an approach with planned gates, meaning models one through three (TPM, lean and gated Kanban), so the required iterations for the individual tasks are examined in greater detail in "7.4.2.2 Statistical investigation of the dependence of the results on the input parameters." As expected, "more difficult" framework conditions (i.e., several parameters in the HIGH form) lead to more obsolete tasks and more iterations, and thus more required project phases. Nevertheless, differences between the models can already be seen at this point (see Table 11).

Run	**Characterization of the attributes newness/difficulty/ complexity/maturity**	**Number of modelled phases**	**Number of obsolete tasks in model 1/2/3/4/5**
V00	Mid/Mid/Mid/Mid	9	14/14/13/11/9
V01	Low/Low/Low/Low	5	0/0/0/0/0
V02	High/Low/Low/Low	6	1/1/1/1/0
V03	Low/High/Low/Low	5	0/0/0/0/0
V04	Low/Low/High/Low	6	1/1/1/1/0
V05	Low/Low/Low/High	5	0/0/0/0/0
V06	High/High/High/Low	7	52/52/51/52/34
V07	High/High/Low/High	7	1/1/1/1/0
V08	High/Low/High/High	7	53/53/54/52/33
V09	Low/High/High/High	10	1/1/1/1/0
V10	High/High/High/High	10	47/47/44/52/29

Table 11: Number of phases and obsolete tasks dependent on run and model

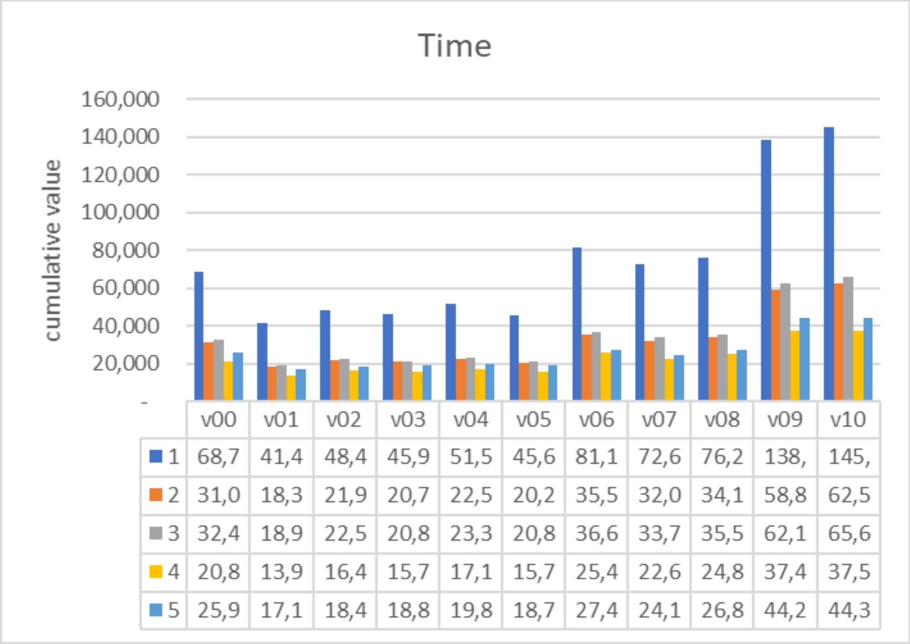
It is noticeable that in model five (APM), fewer tasks are evaluated as obsolete than in the other models with similar values. This offers the first indication of the influence of learning effects, which are faster in the agile model due to the short cycles.

In the next step, the tasks of each approach were considered in their sum, with these values being compared separately for each run. The total time, total effort and number of iterations were accordingly determined (see the following Figure 32 through Figure 34. This step prepares for examining hypotheses I.A to I.C, which are further detailed in “7.4.2.2 Statistical investigation of the dependence of the results on the input parameters."



	v00	v01	v02	v03	v04	v05	v06	v07	v08	v09	v10
■ 1	2,60	1,55	1,70	1,73	1,83	1,71	2,60	2,51	2,44	4,25	4,34
■ 2	2,60	1,55	1,70	1,73	1,83	1,71	2,60	2,51	2,44	4,25	4,34
■ 3	2,62	1,56	1,72	1,73	1,83	1,71	2,62	2,52	2,49	4,30	4,32
■ 4	2,22	1,55	1,72	1,73	1,77	1,71	2,49	2,34	2,40	3,97	4,05
■ 5	2,36	1,55	1,64	1,73	1,74	1,71	2,33	2,25	2,20	3,47	3,57

Figure 32: Cumulative values for the effort dependent on run and model



	v00	v01	v02	v03	v04	v05	v06	v07	v08	v09	v10
■ 1	68,7	41,4	48,4	45,9	51,5	45,6	81,1	72,6	76,2	138,	145,
■ 2	31,0	18,3	21,9	20,7	22,5	20,2	35,5	32,0	34,1	58,8	62,5
■ 3	32,4	18,9	22,5	20,8	23,3	20,8	36,6	33,7	35,5	62,1	65,6
■ 4	20,8	13,9	16,4	15,7	17,1	15,7	25,4	22,6	24,8	37,4	37,5
■ 5	25,9	17,1	18,4	18,8	19,8	18,7	27,4	24,1	26,8	44,2	44,3

Figure 33: Cumulative values for the time dependent on run and model

	v00	v01	v02	v03	v04	v05	v06	v07	v08	v09	v10
■ 1	207	140	154	140	165	140	210	186	202	318	323
■ 2	207	140	154	140	165	140	210	186	202	318	323
■ 3	208	140	154	140	165	140	210	184	203	322	318
■ 4	179	140	156	140	159	140	203	173	199	298	303
■ 5	188	140	147	140	156	140	188	167	181	260	273

Figure 34: Cumulative values for the iterations dependent on run and model

As an example, for run v00 (i.e., equally distributed values of all parameters), the time curves of cumulated effort and cumulated priority were also determined (see Figure 35 and Figure 36).

It can be seen that, above all, the traditional procedure model requires considerably more time for task execution. This is due to the sequential processing of the individual steps. This result is not surprising, because it was modeled accordingly in the simulation. However, in order to exclude the influence of the specific modelling as such, the results are normalized later (i.e., the differences justified in the model's technical design are eliminated).

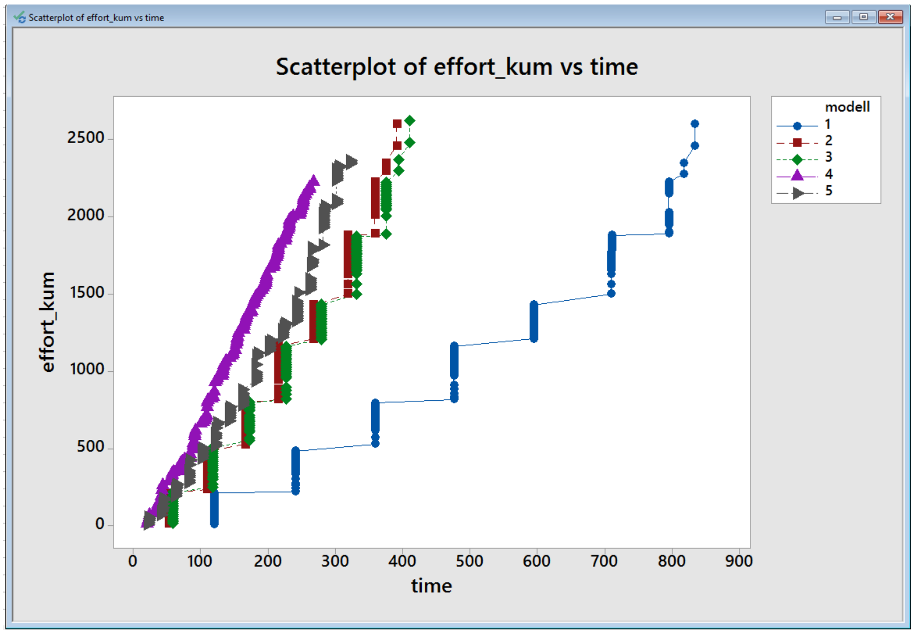


Figure 35: Cumulated effort per model dependent on time

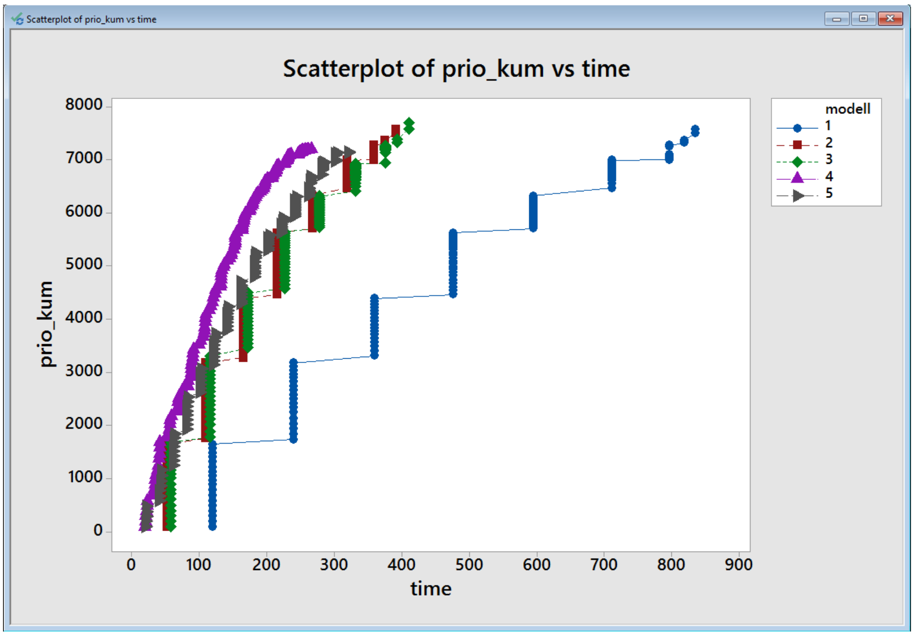


Figure 36: Cumulated priority per model dependent on time

For a general impression of the influencing factors, a main effects screener was run over all data. Differences exist in the influence on effort, time and iterations. According to the expectations, a strong influence of the run is given (i.e., the characteristic of the framework parameters). With regard to the various PM approaches, however, it can be seen that there is a strong influence on time, but a much smaller influence on effort and iterations.

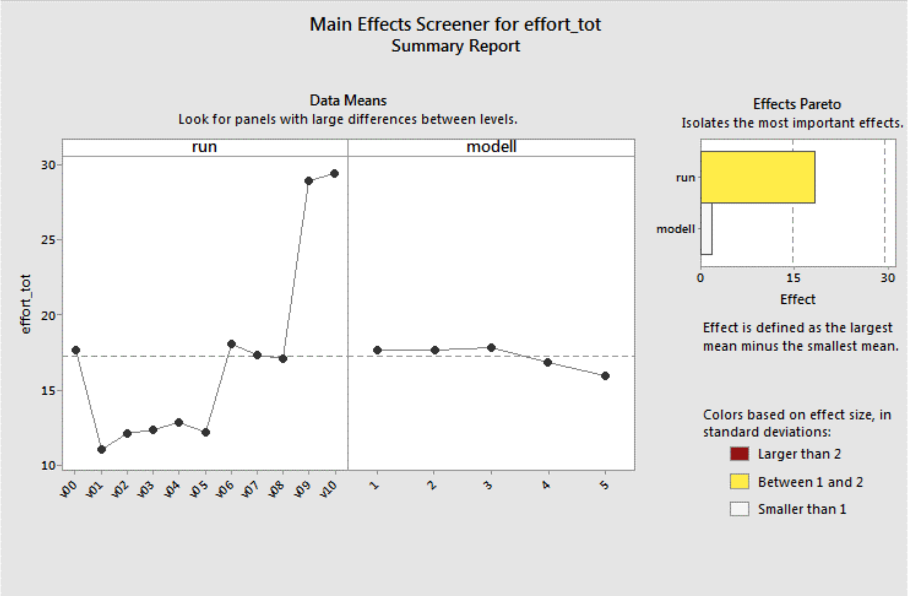


Figure 37: Main effects screener for effort

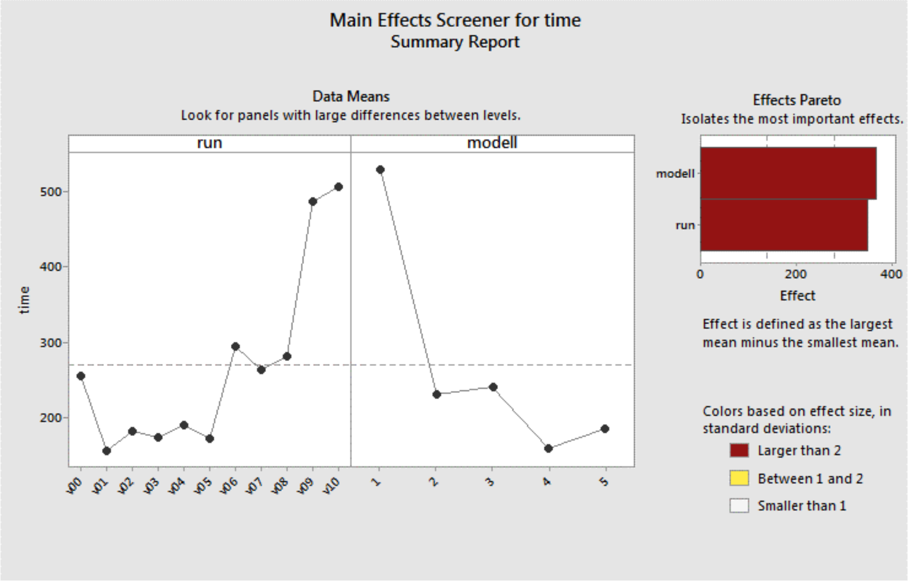


Figure 38: Main effects screener for time

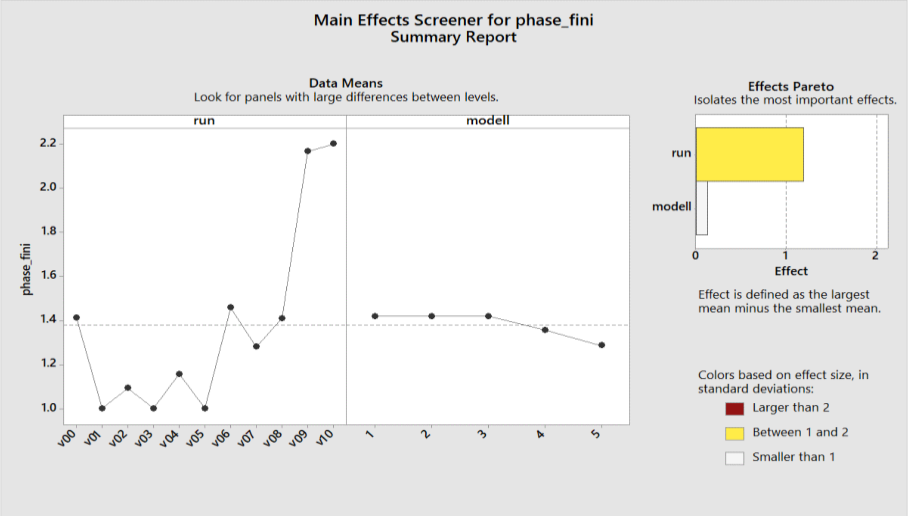


Figure 39: Main effects screener for iterations

7.4.2 Statistical analysis

7.4.2.1 General influence of the models on the result

First, it was examined whether the respective models exhibit differences with regard to the simulation results for the values for effort, time and required iterations. Run V00, which possesses equally distributed values for the framework conditions, was analyzed for this purpose. The total values were determined for the effort of all tasks (Effort_total), for the total duration until project completion (time) and for the required iterations (i.e.) the total number of phases of all tasks (phase_finished)).

With regard to effort, it can be seen that the differences between PM approaches are recognizable, but not statistically significant (see Figure 40 and Figure 41). The same applies to the required iterations (see Figure 44 and Figure 45). The evaluation of required time duration results in statistically significant differences, especially for model one (TPM), but also between the other models (see Figure 42 and Figure 43).

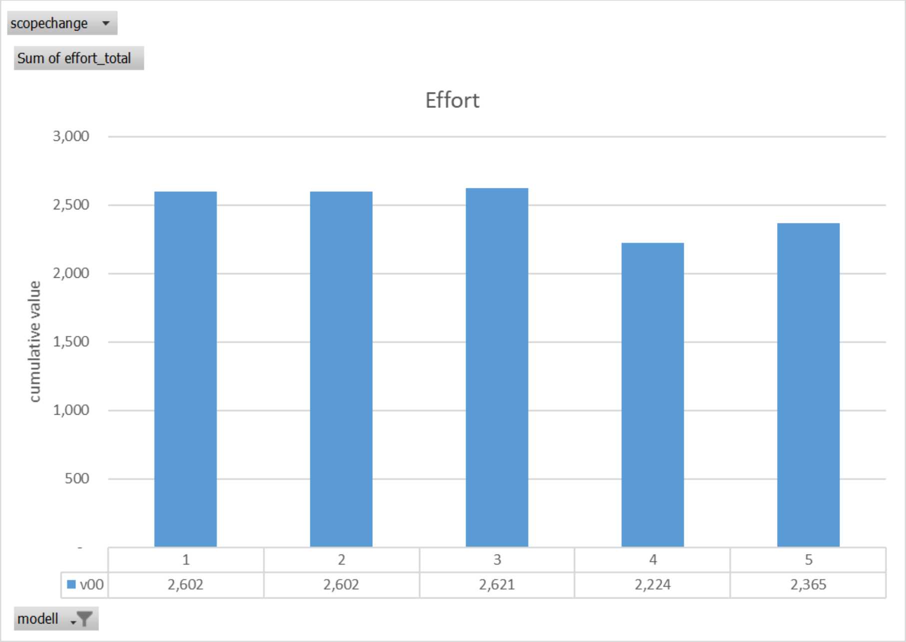
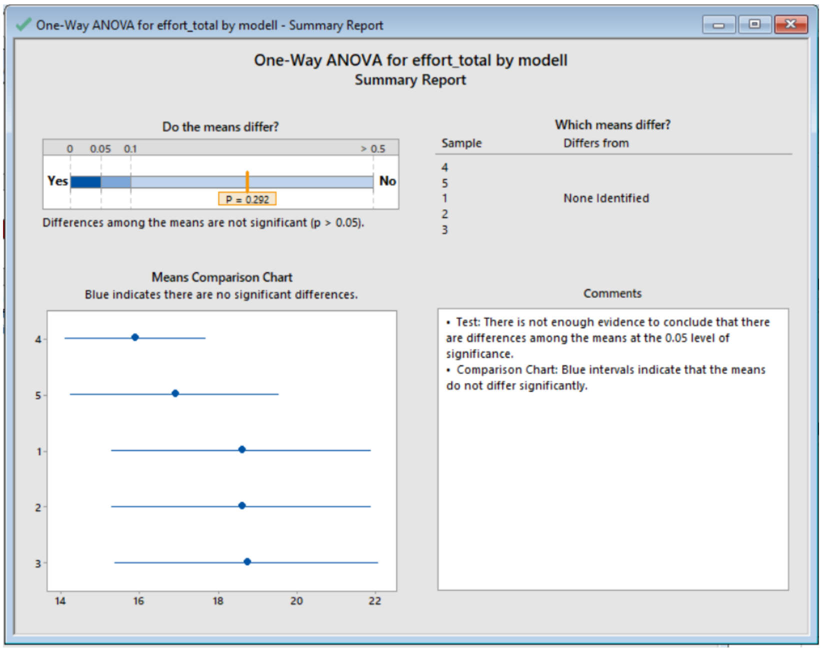


Figure 40: Total effort for the different models in run V00



Differences among the means are not significant (p > 0.05).

Blue indicates there are no significant differences.

- Test: There is not enough evidence to conclude that there are differences among the means at the 0.05 level of significance.
- Comparison Chart: Blue intervals indicate that the means do not differ significantly.

Figure 41: One-way Anova for effort by model in run V00

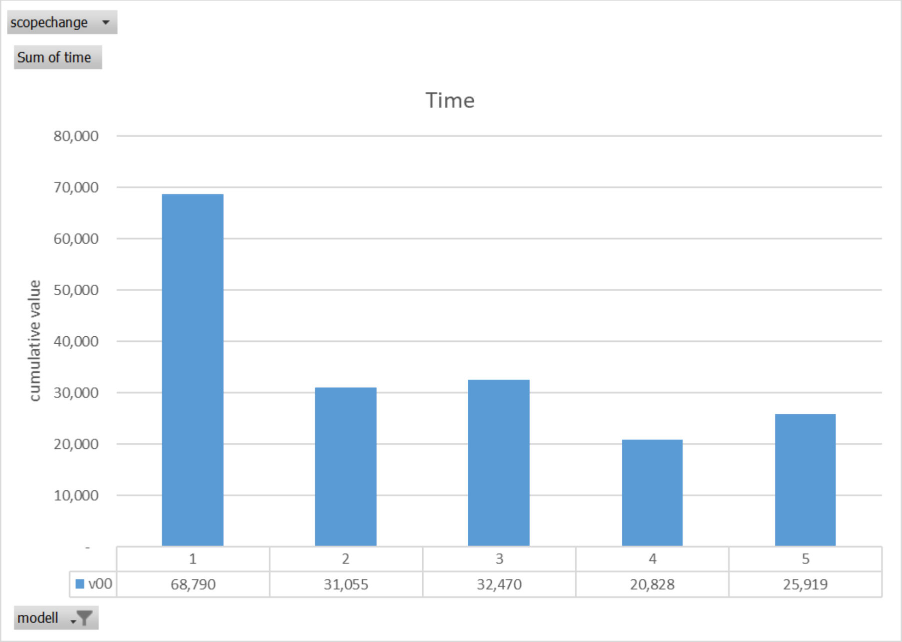
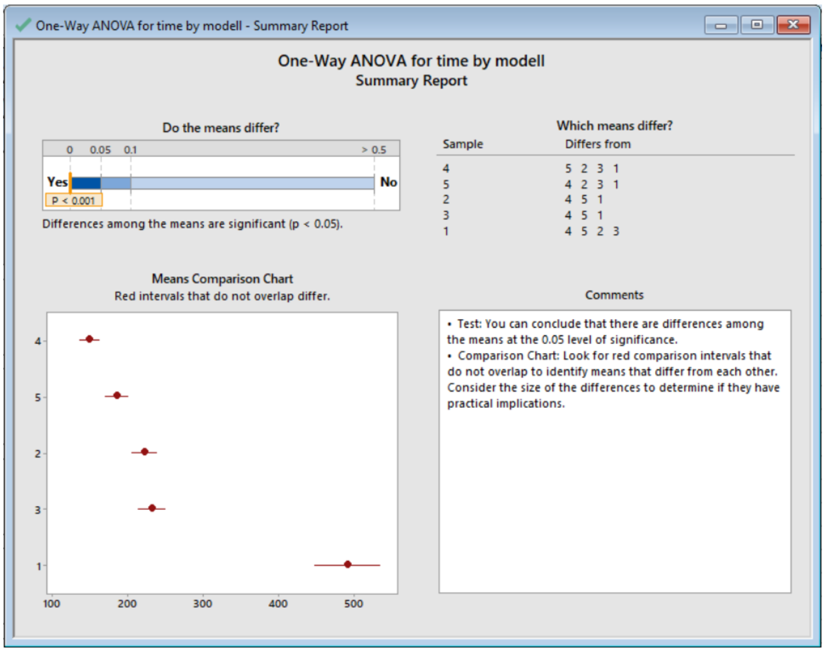


Figure 42: Total time for the different models in run V00



One-Way ANOVA for time by modell
Summary Report

Do the means differ?

Differences among the means are significant (p < 0.05).

Which means differ?

Sample	Differs from
4	5 2 3 1
5	4 2 3 1
2	4 5 1
3	4 5 1
1	4 5 2 3

Means Comparison Chart
Red intervals that do not overlap differ.

Comments

- Test: You can conclude that there are differences among the means at the 0.05 level of significance.
- Comparison Chart: Look for red comparison intervals that do not overlap to identify means that differ from each other. Consider the size of the differences to determine if they have practical implications.

Figure 43: One-way Anova for time by model in run V00

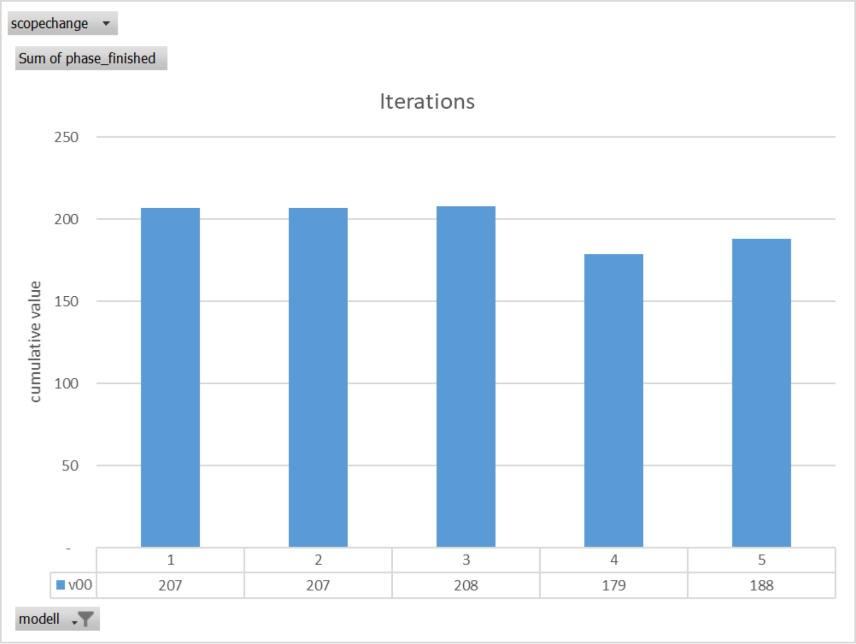
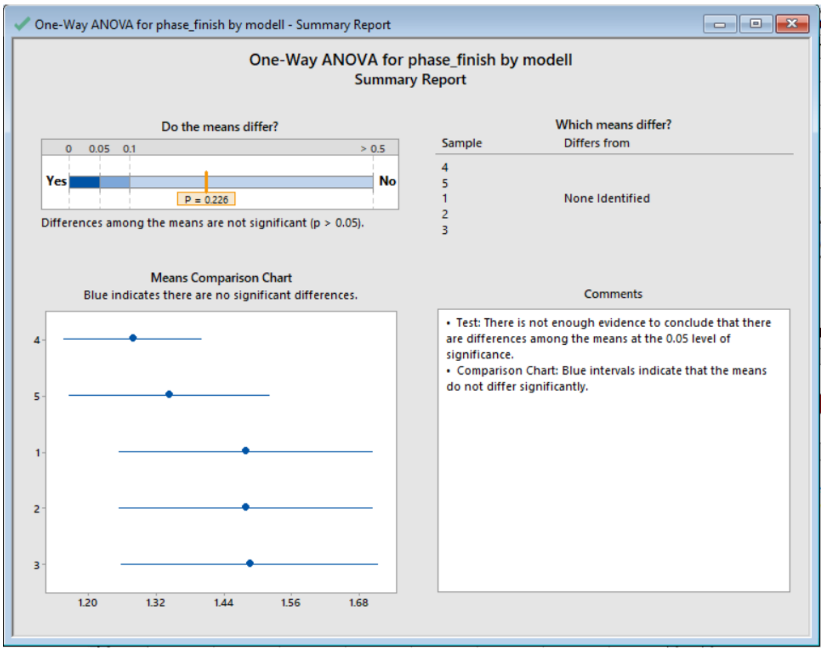


Figure 44: Total iterations for the different models in run V00



One-Way ANOVA for phase_finish by modell
Summary Report

Do the means differ?

Differences among the means are not significant (p > 0.05).

Which means differ?

Sample	Differs from
4 5 1 2 3	None Identified

Means Comparison Chart
Blue indicates there are no significant differences.

Comments

- Test: There is not enough evidence to conclude that there are differences among the means at the 0.05 level of significance.
- Comparison Chart: Blue intervals indicate that the means do not differ significantly.

Figure 45: One-way Anova for iterations by model in run V00

7.4.2.2 Statistical investigation of the dependence of the results on the input parameters

7.4.2.2.1 Low-value input parameters (subset LOW)

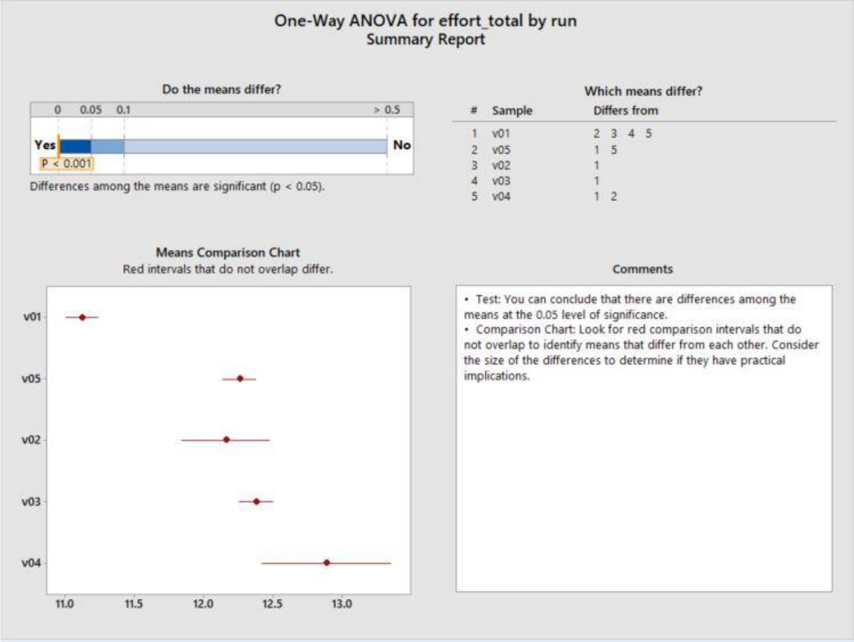
The simulation was programmed so that the input parameters could be changed independently of each other. As such, it was possible to vary the values for newness, difficulty, complexity and maturity arbitrarily, although in practice, correlations are probable. However, this makes statistical analysis easier, because the influencing factors can be considered separately. The statistics in chapters 7.4.2.2.1 through 7.4.2.2.3 aim to examine hypotheses II.A through II.C.

In order to avoid having to compare each run individually with each other, two groups were formed: one group with low input parameter values, and one group with correspondingly high values.

In group one, analyzed in this chapter, run V01, which possesses low values (LOW) for all parameters, offers the comparison for the runs V02 through V05, in each of which one of the parameters newness, difficulty, complexity or maturity possesses a high value (characteristic HIGH).

Group two is examined in chapter 7.4.2.2.2. There, run V10, which presents all attributes as high, offers the comparison for runs V06 through V09, in each of which one attribute is characterized as low.

First, a one-way Anova statistic was performed to identify any significant differences between the runs. This was performed in terms of effort, time and iterations. As a result, significant differences were found for the effort and the required time. Regarding the iterations, runs V00, V03 and V05 exhibit no iterations, but runs V02 and V04 differ significantly (see Figure 46 through Figure 48).



One-Way ANOVA for effort_total by run
Summary Report

Do the means differ?

Differences among the means are significant ($p < 0.05$).

Which means differ?

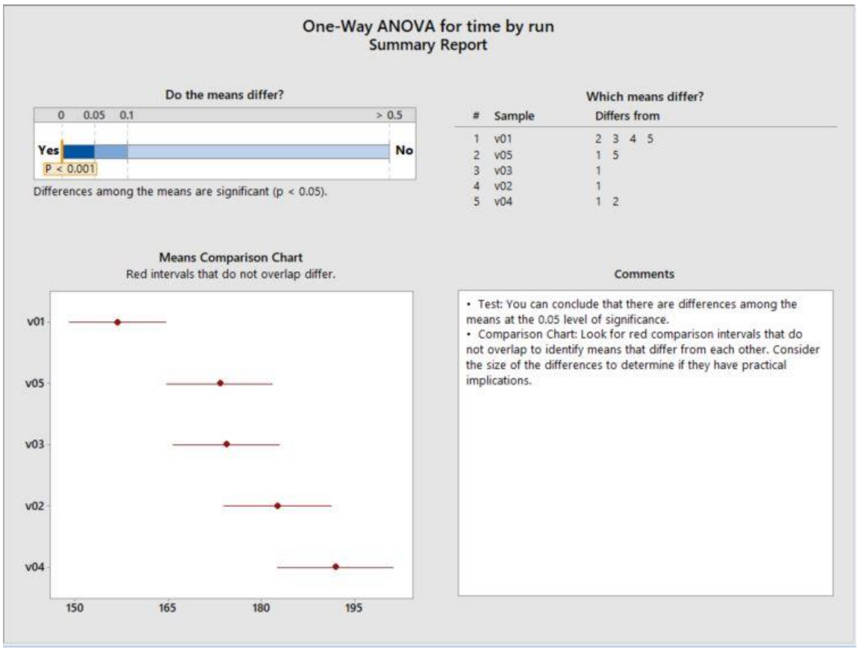
#	Sample	Differs from
1	v01	2 3 4 5
2	v05	1 5
3	v02	1
4	v03	1
5	v04	1 2

Means Comparison Chart
Red intervals that do not overlap differ.

Comments

- Test: You can conclude that there are differences among the means at the 0.05 level of significance.
- Comparison Chart: Look for red comparison intervals that do not overlap to identify means that differ from each other. Consider the size of the differences to determine if they have practical implications.

Figure 46: One-way Anova for effort by run in subset LOW



One-Way ANOVA for time by run
Summary Report

Do the means differ?

Differences among the means are significant ($p < 0.05$).

Which means differ?

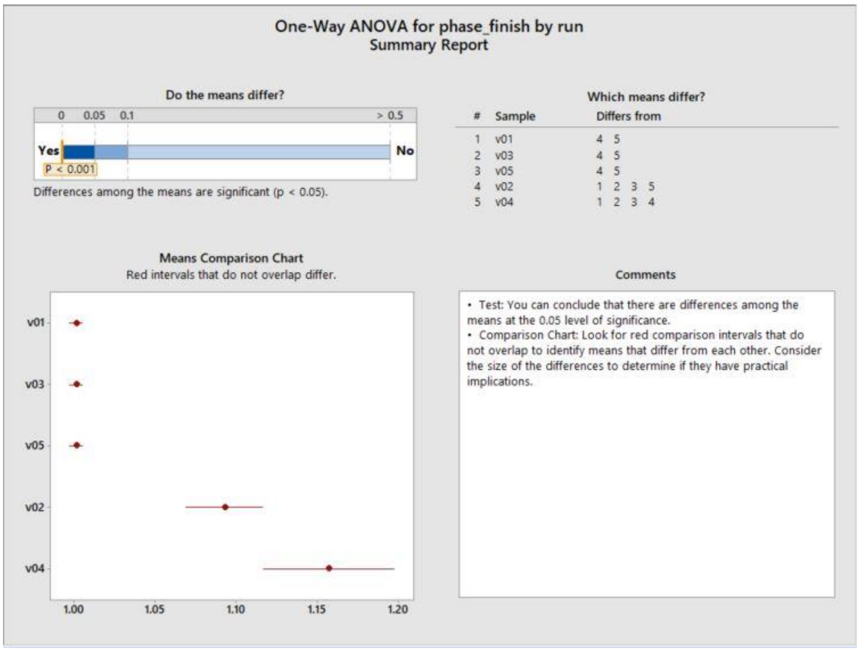
#	Sample	Differs from
1	v01	2 3 4 5
2	v05	1 5
3	v03	1
4	v02	1
5	v04	1 2

Means Comparison Chart
Red intervals that do not overlap differ.

Comments

- Test: You can conclude that there are differences among the means at the 0.05 level of significance.
- Comparison Chart: Look for red comparison intervals that do not overlap to identify means that differ from each other. Consider the size of the differences to determine if they have practical implications.

Figure 47: One-way Anova for time by run in subset LOW



One-Way ANOVA for phase_finish by run
Summary Report

Do the means differ?

Differences among the means are significant (p < 0.05).

Which means differ?

#	Sample	Differs from
1	v01	4 5
2	v03	4 5
3	v05	4 5
4	v02	1 2 3 5
5	v04	1 2 3 4

Means Comparison Chart
Red intervals that do not overlap differ.

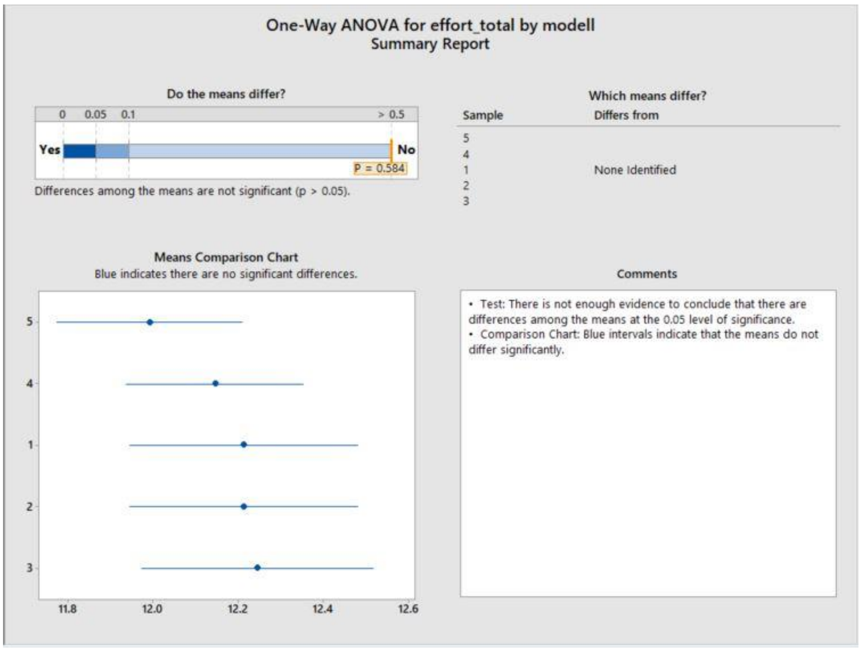
Comments

- Test: You can conclude that there are differences among the means at the 0.05 level of significance.
- Comparison Chart: Look for red comparison intervals that do not overlap to identify means that differ from each other. Consider the size of the differences to determine if they have practical implications.

Figure 48: One-way Anova for iterations by run in subset LOW

After it could be demonstrated that the framework parameters influence the simulation result, it was subsequently examined whether these influences exert different effects on the different PM approaches. One-way Anova statistics were also created for this purpose, this time depending on the models.

Here, a statistically significant difference occurs between the models only with regard to the time required, whereas effort and required iterations exhibit no statistically significant difference.



One-Way ANOVA for effort_total by modell
Summary Report

Do the means differ?

Differences among the means are not significant (p > 0.05).

Which means differ?

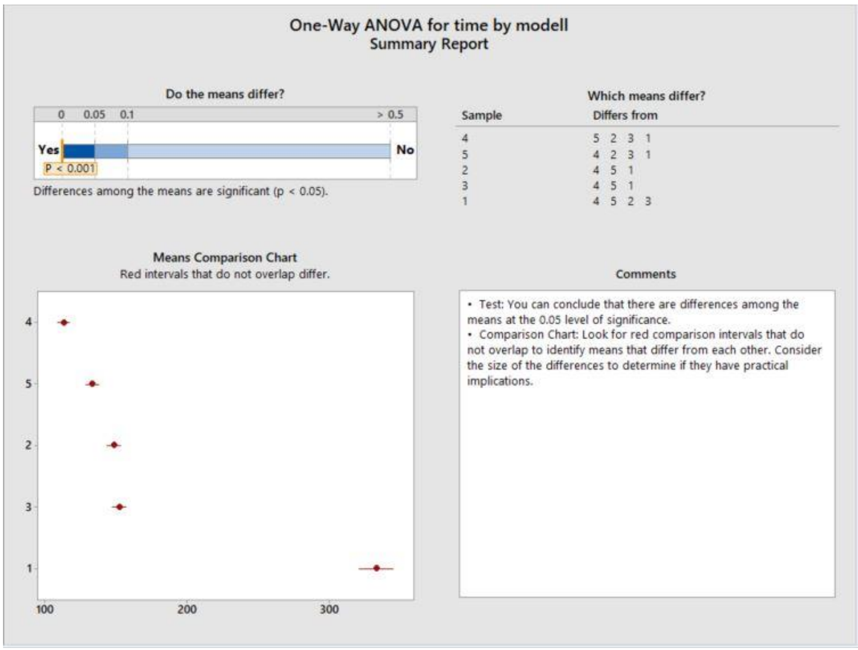
Sample	Differs from
5	
4	
1	None Identified
2	
3	

Means Comparison Chart
Blue indicates there are no significant differences.

Comments

- Test: There is not enough evidence to conclude that there are differences among the means at the 0.05 level of significance.
- Comparison Chart: Blue intervals indicate that the means do not differ significantly.

Figure 49: One-way Anova for effort by model in subset LOW



One-Way ANOVA for time by modell
Summary Report

Do the means differ?

Differences among the means are significant (p < 0.05).

Which means differ?

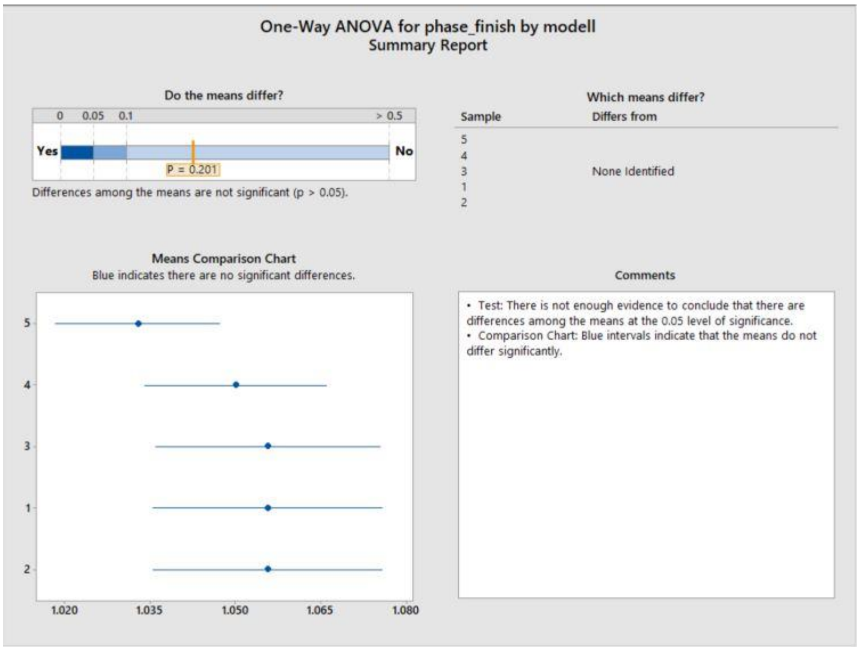
Sample	Differs from
4	5 2 3 1
5	4 2 3 1
2	4 5 1
3	4 5 1
1	4 5 2 3

Means Comparison Chart
Red intervals that do not overlap differ.

Comments

- Test: You can conclude that there are differences among the means at the 0.05 level of significance.
- Comparison Chart: Look for red comparison intervals that do not overlap to identify means that differ from each other. Consider the size of the differences to determine if they have practical implications.

Figure 50: One-way Anova for time by model in subset LOW



Differences among the means are not significant (p > 0.05).

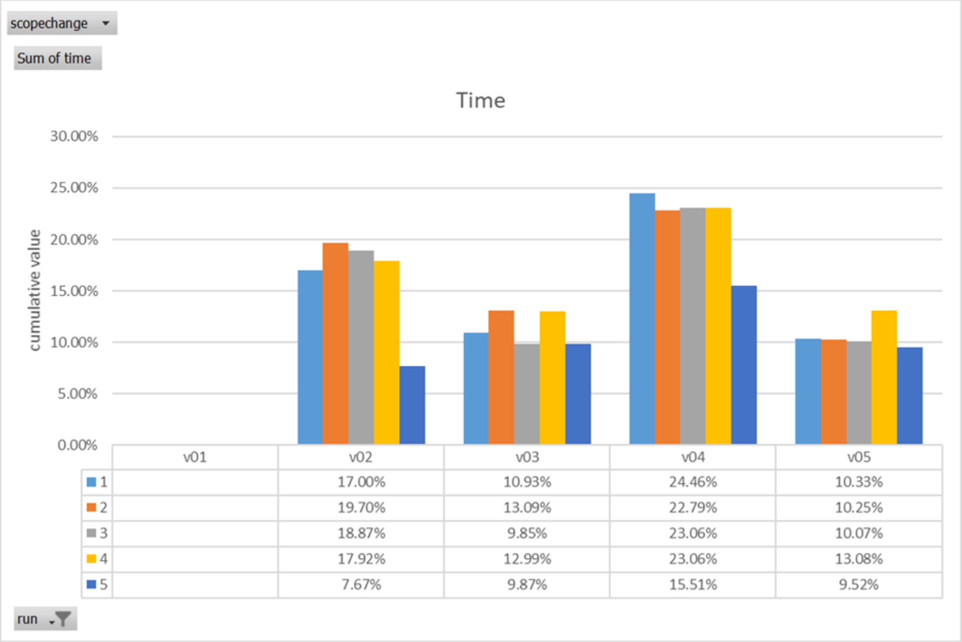
- Test: There is not enough evidence to conclude that there are differences among the means at the 0.05 level of significance.
- Comparison Chart: Blue intervals indicate that the means do not differ significantly.

Figure 51: One-way Anova for iterations by model in subset LOW

In the next step, it was determined how the results for effort, time and iterations are displayed depending on the runs and the models, if the values are normalized. As base value, run V01 was chosen, in which all parameters correspond to the characteristic LOW; the values of the other runs were displayed as percentage deviation.

Since the mean of the absolute values between the individual models already do not significantly differ with regard to effort and iterations, it only makes sense more closely examine the time required.

It is noticeable that the relative changes do not considerably differ between the models (see Figure 52). This can also be seen in the representation of the interaction plot (Figure 53). The values for model one are much higher, but the lines for the individual runs remain rather parallel.



	v01	v02	v03	v04	v05
1		17.00%	10.93%	24.46%	10.33%
2		19.70%	13.09%	22.79%	10.25%
3		18.87%	9.85%	23.06%	10.07%
4		17.92%	12.99%	23.06%	13.08%
5		7.67%	9.87%	15.51%	9.52%

Figure 52: Relative differences in time depending on model and run for subset LOW

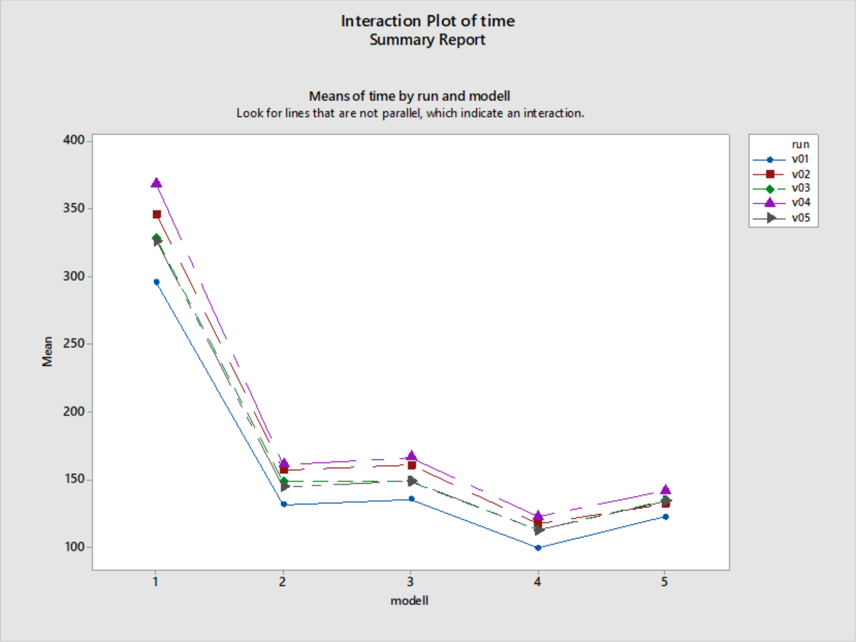


Figure 53: Mean values of time per run depending on the model for subset LOW

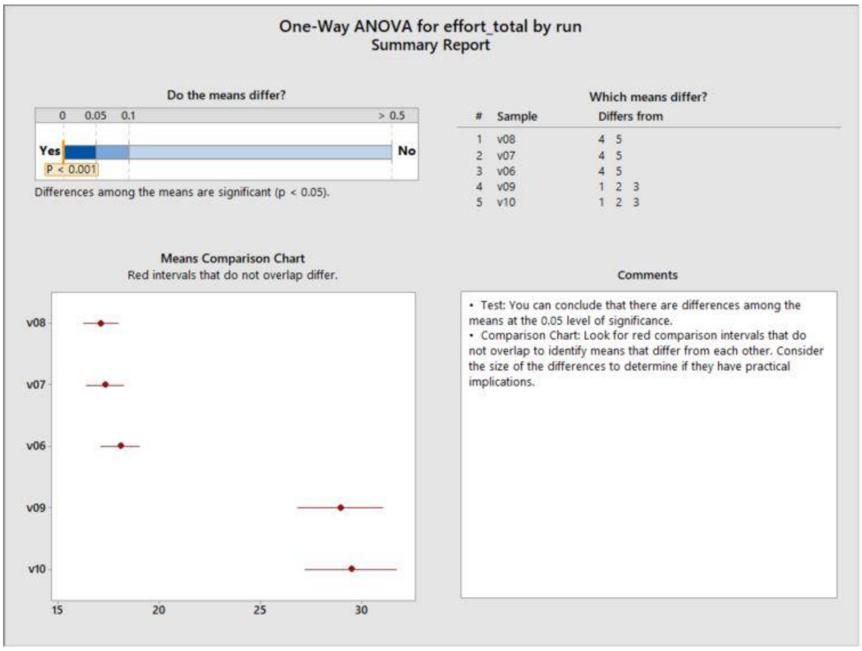
From this, it can be concluded that for projects with low requirements (i.e., low values for newness, difficulty, complexity and maturity), the choice of PM approach exerts no significant influence on effort and required iterations. With regard to the time required, model one (TPM) remains noticeably slower, and the framework conditions also influence the project duration. However, these effects do not reinforce each other.

7.4.2.2.2 High-value input parameters (subset HIGH)

According to the subset LOW, in which a run with all parameters of the characteristic LOW was utilized as comparison value, the subset HIGH takes the run V10, in which all parameters correspond to the characteristic HIGH, as basis. This run is compared to runs V06 through V09, in which only one of the parameters of newness, difficulty, complexity and maturity correspond to the characteristic LOW.

In subset High, a one-way Anova statistic was also performed first to check the significance of the differences between the runs. The statistics illustrate that significant differences exist in the mean values between the runs for time, but also for effort and iterations (see Figure 54 through Figure 56).

Subsequently, the analysis for significance was conducted depending on the models (see Figure 57 through Figure 59). In the HIGH subset, the one-way Anova statistics illustrated significant differences in mean values for effort, time and iterations between the corresponding PM approaches so that additional analysis can be performed.



One-Way ANOVA for effort_total by run
Summary Report

Do the means differ?

Differences among the means are significant (p < 0.05).

Which means differ?

#	Sample	Differs from
1	v08	4 5
2	v07	4 5
3	v06	4 5
4	v09	1 2 3
5	v10	1 2 3

Means Comparison Chart
Red intervals that do not overlap differ.

Comments

- Test: You can conclude that there are differences among the means at the 0.05 level of significance.
- Comparison Chart: Look for red comparison intervals that do not overlap to identify means that differ from each other. Consider the size of the differences to determine if they have practical implications.

Figure 54: One-way Anova for effort by run in subset HIGH

One-Way ANOVA for time by run
Summary Report

Do the means differ?

Differences among the means are significant (p < 0.05).

Which means differ?

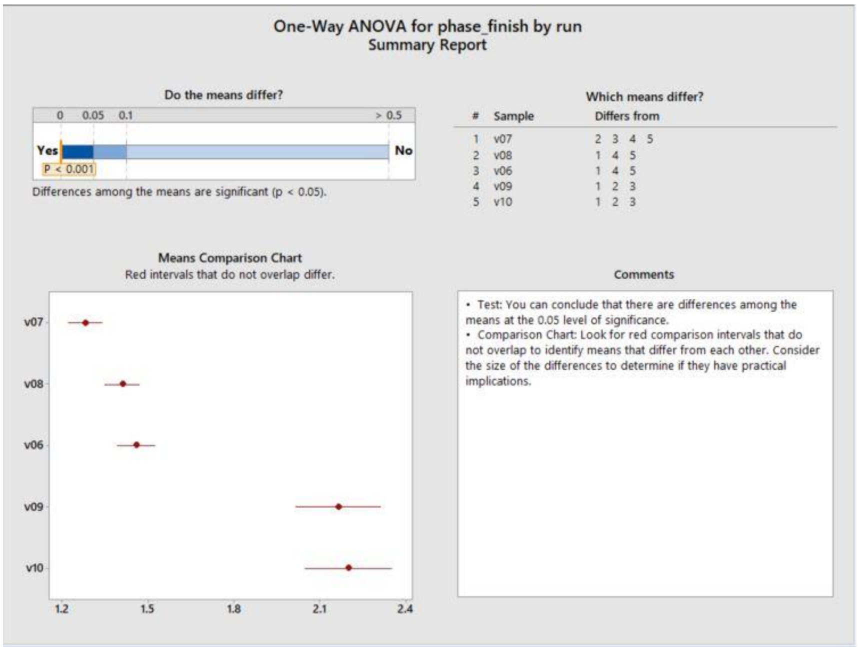
#	Sample	Differs from
1	v07	3 4 5
2	v08	4 5
3	v06	1 4 5
4	v09	1 2 3
5	v10	1 2 3

Means Comparison Chart
Red intervals that do not overlap differ.

Comments

- Test: You can conclude that there are differences among the means at the 0.05 level of significance.
- Comparison Chart: Look for red comparison intervals that do not overlap to identify means that differ from each other. Consider the size of the differences to determine if they have practical implications.

Figure 55: One-way Anova for time by run in subset HIGH



One-Way ANOVA for phase_finish by run
Summary Report

Do the means differ?

Differences among the means are significant (p < 0.05).

Which means differ?

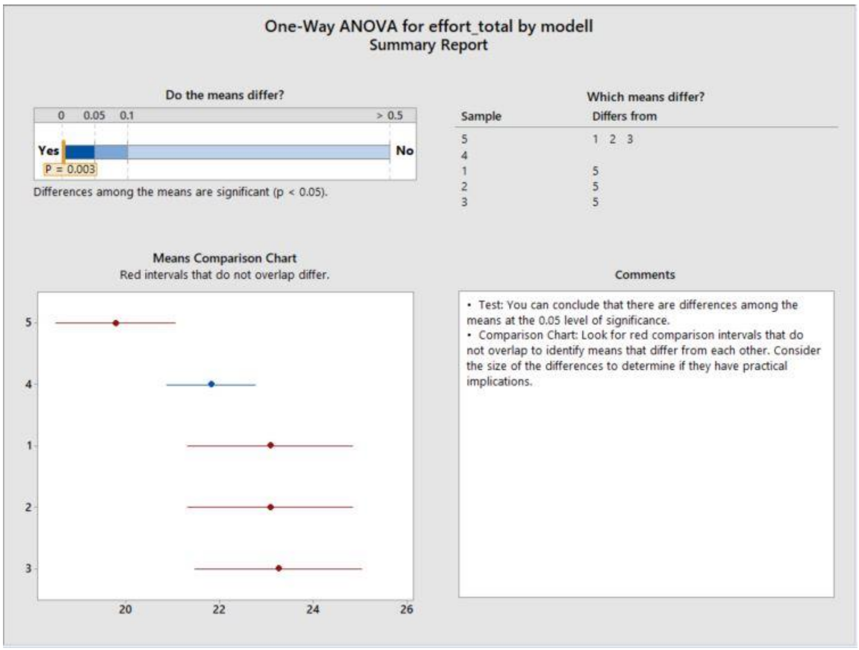
#	Sample	Differs from
1	v07	2 3 4 5
2	v08	1 4 5
3	v06	1 4 5
4	v09	1 2 3
5	v10	1 2 3

Means Comparison Chart
Red intervals that do not overlap differ.

Comments

- Test: You can conclude that there are differences among the means at the 0.05 level of significance.
- Comparison Chart: Look for red comparison intervals that do not overlap to identify means that differ from each other. Consider the size of the differences to determine if they have practical implications.

Figure 56: One-way Anova for iterations by run in subset HIGH



One-Way ANOVA for effort_total by modell
Summary Report

Do the means differ?

Differences among the means are significant (p < 0.05).

Which means differ?

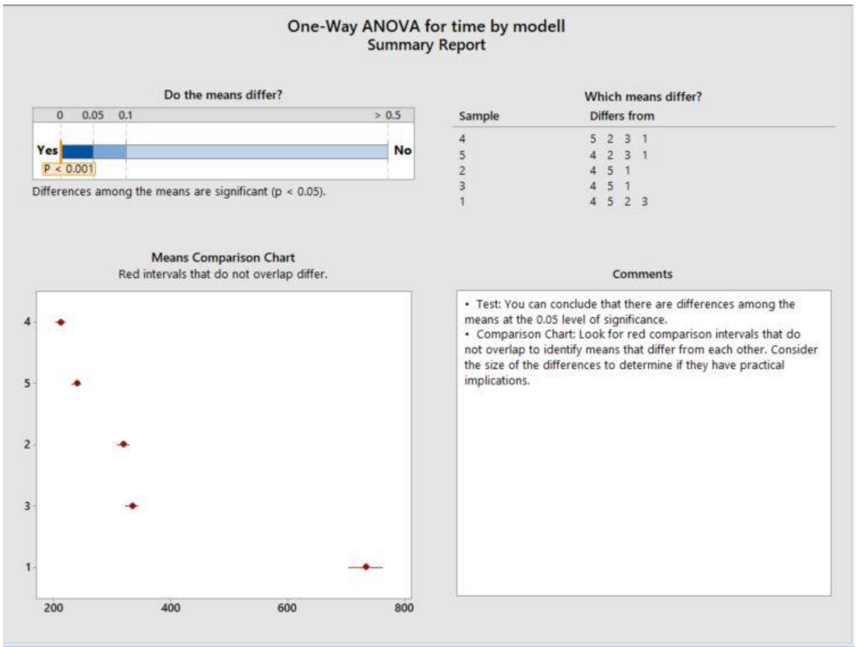
Sample	Differs from
5	1 2 3
4	
1	5
2	5
3	5

Means Comparison Chart
Red intervals that do not overlap differ.

Comments

- Test: You can conclude that there are differences among the means at the 0.05 level of significance.
- Comparison Chart: Look for red comparison intervals that do not overlap to identify means that differ from each other. Consider the size of the differences to determine if they have practical implications.

Figure 57: One-way Anova for effort by model in subset HIGH



One-Way ANOVA for time by modell
Summary Report

Do the means differ?

P < 0.001

Differences among the means are significant (p < 0.05).

Which means differ?

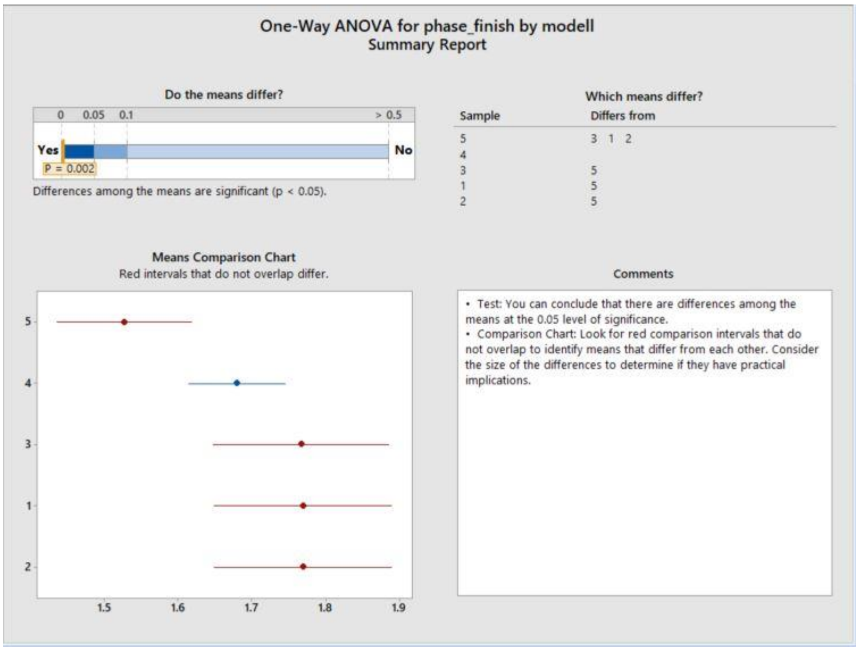
Sample	Differs from
4	5 2 3 1
5	4 2 3 1
2	4 5 1
3	4 5 1
1	4 5 2 3

Means Comparison Chart
Red intervals that do not overlap differ.

Comments

- Test: You can conclude that there are differences among the means at the 0.05 level of significance.
- Comparison Chart: Look for red comparison intervals that do not overlap to identify means that differ from each other. Consider the size of the differences to determine if they have practical implications.

Figure 58: One-way Anova for time by model in subset HIGH



One-Way ANOVA for phase_finish by modell
Summary Report

Do the means differ?

P = 0.002

Differences among the means are significant (p < 0.05).

Which means differ?

Sample	Differs from
5	3 1 2
4	
3	5
1	5
2	5

Means Comparison Chart
Red intervals that do not overlap differ.

Comments

- Test: You can conclude that there are differences among the means at the 0.05 level of significance.
- Comparison Chart: Look for red comparison intervals that do not overlap to identify means that differ from each other. Consider the size of the differences to determine if they have practical implications.

Figure 59: One-way Anova for iterations by model in subset HIGH

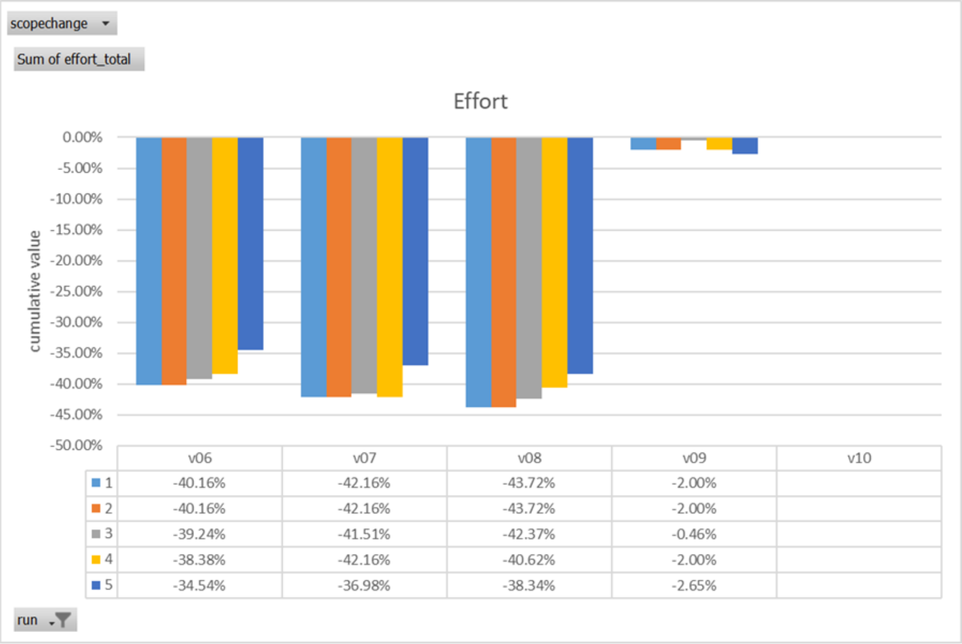
The next step involved investigating how the results for effort, time and iterations are represented as a function of the runs and the models when the values are normalized. As base value, run V10 was chosen, where all parameters correspond to the characteristic HIGH; the values of the other runs were displayed as percentage deviation.

Since the mean of the absolute values differ considerably between the individual models in this subset with regard to effort, time and iterations, all simulation outputs are analyzed further.

Both bar charts (Figure 60, Figure 62 and Figure 64) and interaction plots (Figure 61, Figure 63 and Figure 65) were selected for the graphical representation. Because the comparative value of run V10 is a project with the highest conceivable requirements (all parameters in characteristic HIGH), the values for runs V06 through V09 are relatively smaller, so the percentage deviation is negative.

In the bar graphs, it is noticeable that the relative deviations between runs are much greater than between the models. However, the interaction plots exhibit no parallelism. Rather, it can be seen that the simulation results for effort, time and iterations remain closer for model four (gated Kanban), and even more for model five (APM/Scrum).

This can be interpreted in such a way that both the characteristics of the parameters (represented in the runs) and the respective PM approach (represented in the models) influence the results. However, in the case of the HIGH characteristic, these mutually reinforce each other in such a way that in models four and five, the higher requirements do not come into play as strongly as in models one, two and three.



	v06	v07	v08	v09	v10
1	-40.16%	-42.16%	-43.72%	-2.00%	
2	-40.16%	-42.16%	-43.72%	-2.00%	
3	-39.24%	-41.51%	-42.37%	-0.46%	
4	-38.38%	-42.16%	-40.62%	-2.00%	
5	-34.54%	-36.98%	-38.34%	-2.65%	

Figure 60: Relative differences in effort depending on model and run for subset HIGH

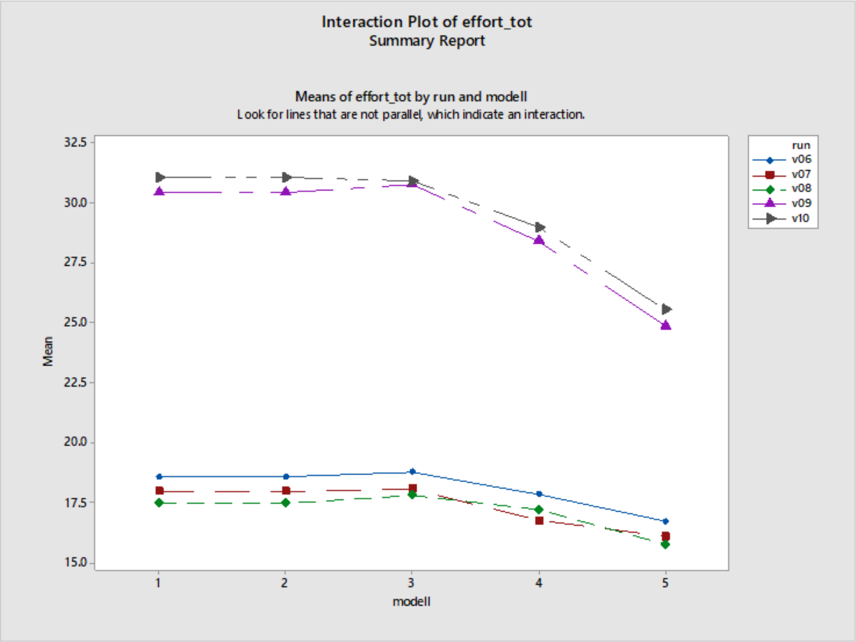
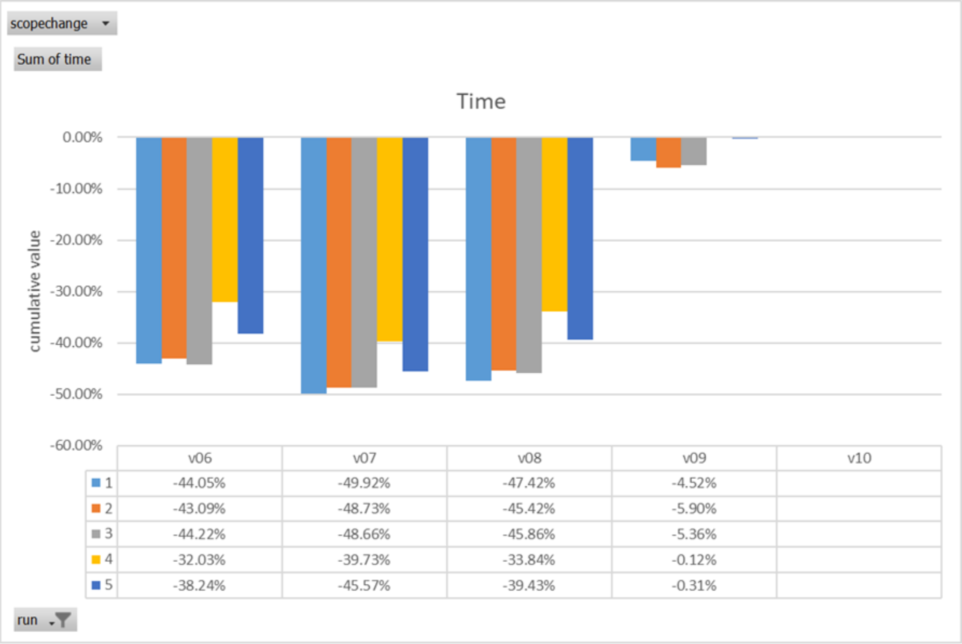


Figure 61: Mean values of effort per run depending on the model for subset HIGH



	v06	v07	v08	v09	v10
1	-44.05%	-49.92%	-47.42%	-4.52%	
2	-43.09%	-48.73%	-45.42%	-5.90%	
3	-44.22%	-48.66%	-45.86%	-5.36%	
4	-32.03%	-39.73%	-33.84%	-0.12%	
5	-38.24%	-45.57%	-39.43%	-0.31%	

Figure 62: Relative differences in time depending on model and run for subset HIGH

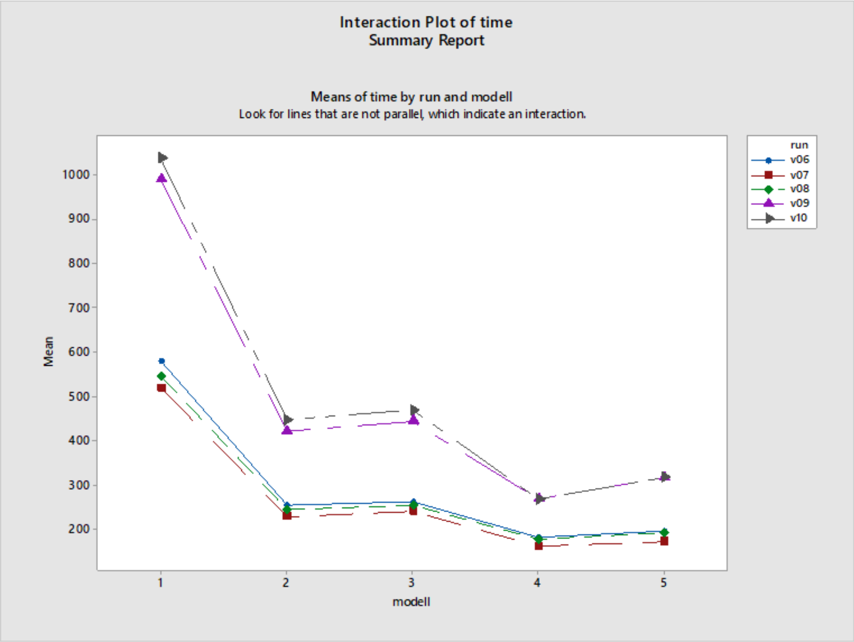
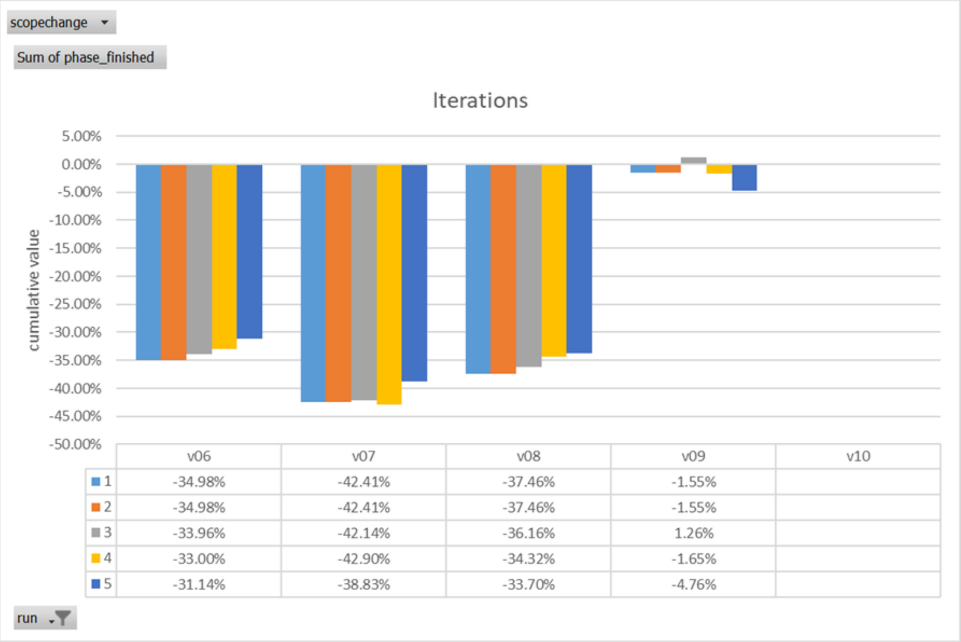


Figure 63: Mean values of time per run depending on the model for subset HIGH



	v06	v07	v08	v09	v10
1	-34.98%	-42.41%	-37.46%	-1.55%	
2	-34.98%	-42.41%	-37.46%	-1.55%	
3	-33.96%	-42.14%	-36.16%	1.26%	
4	-33.00%	-42.90%	-34.32%	-1.65%	
5	-31.14%	-38.83%	-33.70%	-4.76%	

Figure 64: Relative differences in iterations depending on model and run for subset HIGH

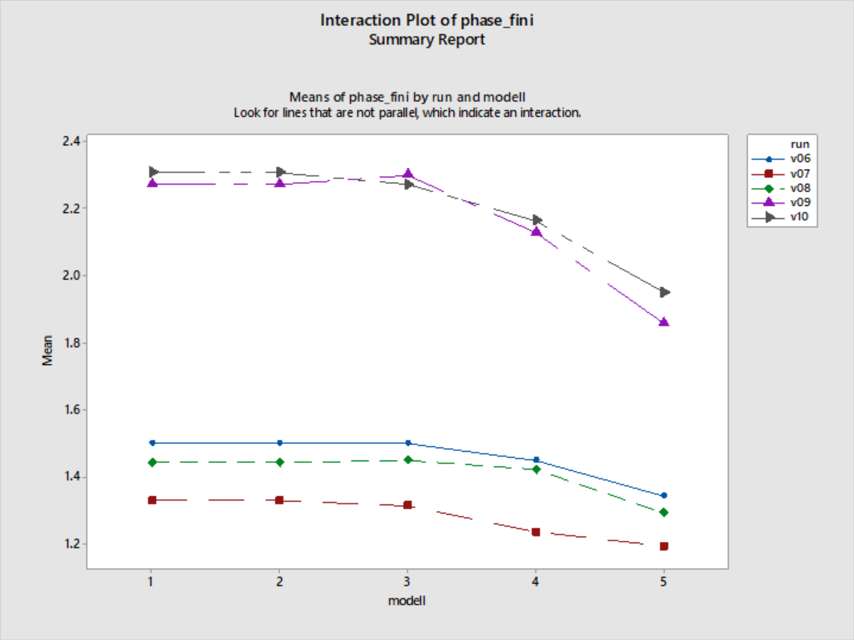


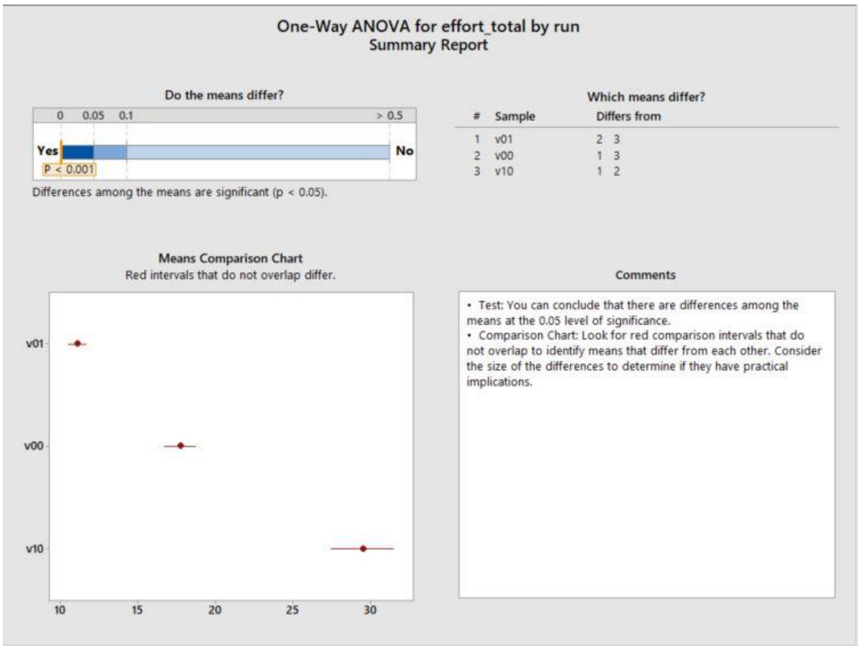
Figure 65: Mean values of iterations per run depending on the model for subset HIGH

7.4.2.2.3 Comparison of input parameters LOW vs MID vs HIGH (subset MID)

The previous two chapters examined how the different input parameters affect the simulation outcomes independently of each other. The current section now examines the relationship between their general characteristics and the different models with regard to the results. For this purpose, a subset is formed that compares the runs V01 (all parameters LOW), V00 (equally distributed values of the parameters) and V10 (all parameters HIGH) against each other.

This chapter also employs one-way Anova statistics to examine whether the mean values of the runs and of the models' outputs significantly differ. The output values are then compared in a normalized manner, and the relationship between run and model is plotted using an interaction plot.

First, the effect of the input parameters (reflected in the runs) on the result is analyzed (see Figure 66 through Figure 68).



One-Way ANOVA for effort_total by run
Summary Report

Do the means differ?

Differences among the means are significant (p < 0.05).

Which means differ?

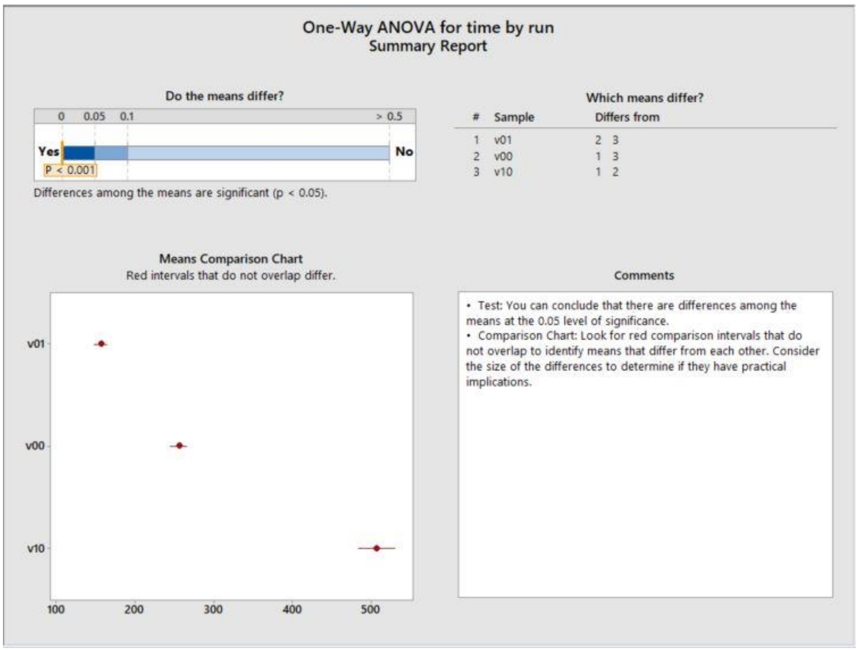
#	Sample	Differs from
1	v01	2 3
2	v00	1 3
3	v10	1 2

Means Comparison Chart
Red intervals that do not overlap differ.

Comments

- Test: You can conclude that there are differences among the means at the 0.05 level of significance.
- Comparison Chart: Look for red comparison intervals that do not overlap to identify means that differ from each other. Consider the size of the differences to determine if they have practical implications.

Figure 66: One-way Anova for effort by run in subset MID



One-Way ANOVA for time by run
Summary Report

Do the means differ?

Differences among the means are significant (p < 0.05).

Which means differ?

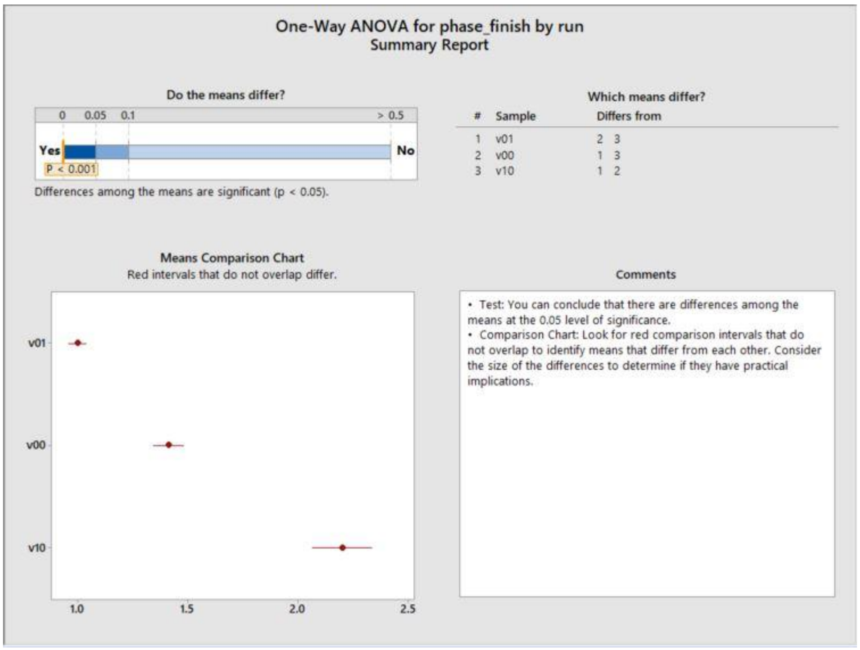
#	Sample	Differs from
1	v01	2 3
2	v00	1 3
3	v10	1 2

Means Comparison Chart
Red intervals that do not overlap differ.

Comments

- Test: You can conclude that there are differences among the means at the 0.05 level of significance.
- Comparison Chart: Look for red comparison intervals that do not overlap to identify means that differ from each other. Consider the size of the differences to determine if they have practical implications.

Figure 67: One-way Anova for time by run in subset MID



One-Way ANOVA for phase_finish by run
Summary Report

Do the means differ?

Differences among the means are significant (p < 0.05).

Which means differ?

#	Sample	Differs from
1	v01	2 3
2	v00	1 3
3	v10	1 2

Means Comparison Chart
Red intervals that do not overlap differ.

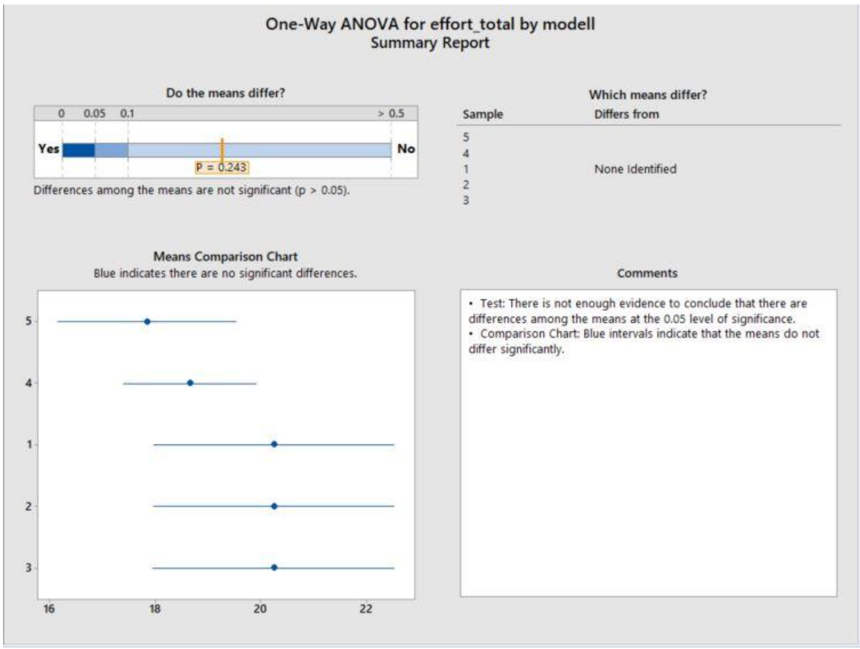
Comments

- Test: You can conclude that there are differences among the means at the 0.05 level of significance.
- Comparison Chart: Look for red comparison intervals that do not overlap to identify means that differ from each other. Consider the size of the differences to determine if they have practical implications.

Figure 68: One-way Anova for iterations by run in subset MID

It is recognizable that the output values of the simulation for effort, time and iterations differ significantly. As expected, the input parameters' effect on the result is rather large, especially if all parameters occur in their extreme form.

In the next step, the models are compared against each other. Significance occurs only for time, and not for effort or iterations.



One-Way ANOVA for effort_total by modell
Summary Report

Do the means differ?

Differences among the means are not significant (p > 0.05).

Which means differ?

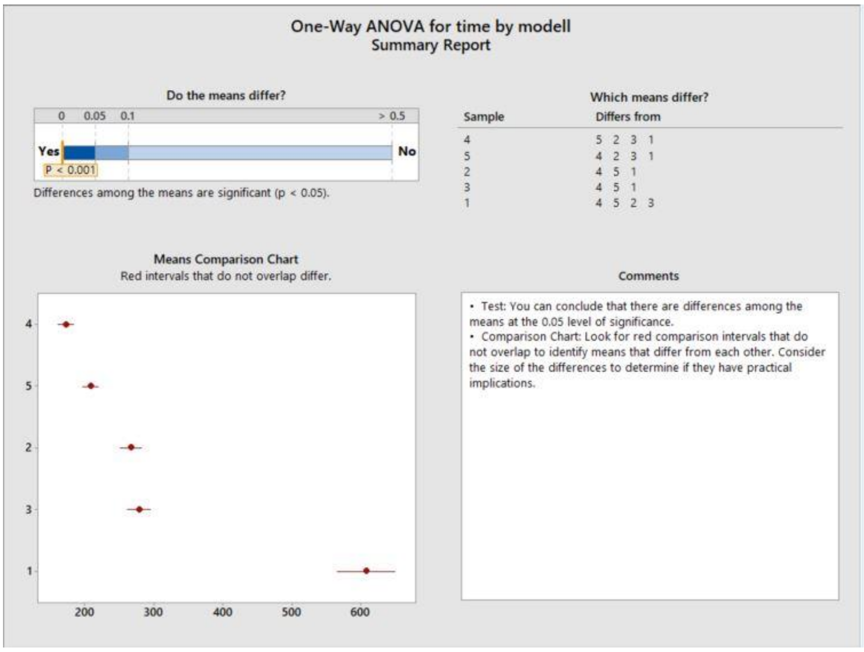
Sample	Differs from
5	
4	
1	None Identified
2	
3	

Means Comparison Chart
Blue indicates there are no significant differences.

Comments

- Test: There is not enough evidence to conclude that there are differences among the means at the 0.05 level of significance.
- Comparison Chart: Blue intervals indicate that the means do not differ significantly.

Figure 69: One-way Anova for effort by model in subset MID



One-Way ANOVA for time by modell
Summary Report

Do the means differ?

Differences among the means are significant (p < 0.05).

Which means differ?

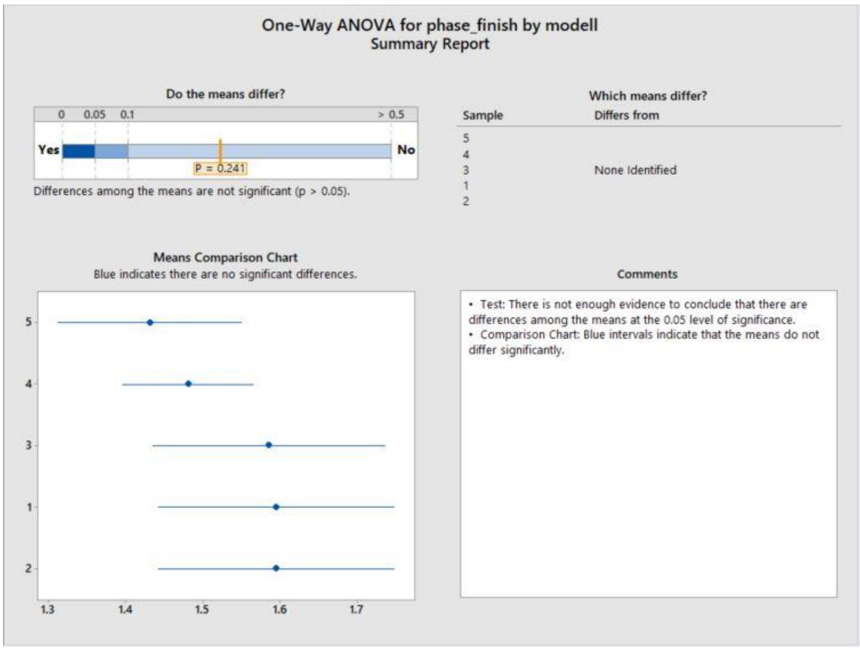
Sample	Differs from
4	5 2 3 1
5	4 2 3 1
2	4 5 1
3	4 5 1
1	4 5 2 3

Means Comparison Chart
Red intervals that do not overlap differ.

Comments

- Test: You can conclude that there are differences among the means at the 0.05 level of significance.
- Comparison Chart: Look for red comparison intervals that do not overlap to identify means that differ from each other. Consider the size of the differences to determine if they have practical implications.

Figure 70: One-way Anova for time by model in subset MID



One-Way ANOVA for phase_finish by modell
Summary Report

Do the means differ?

Differences among the means are not significant (p > 0.05).

Which means differ?

Sample	Differs from
5	
4	
3	None Identified
1	
2	

Means Comparison Chart
Blue indicates there are no significant differences.

Comments

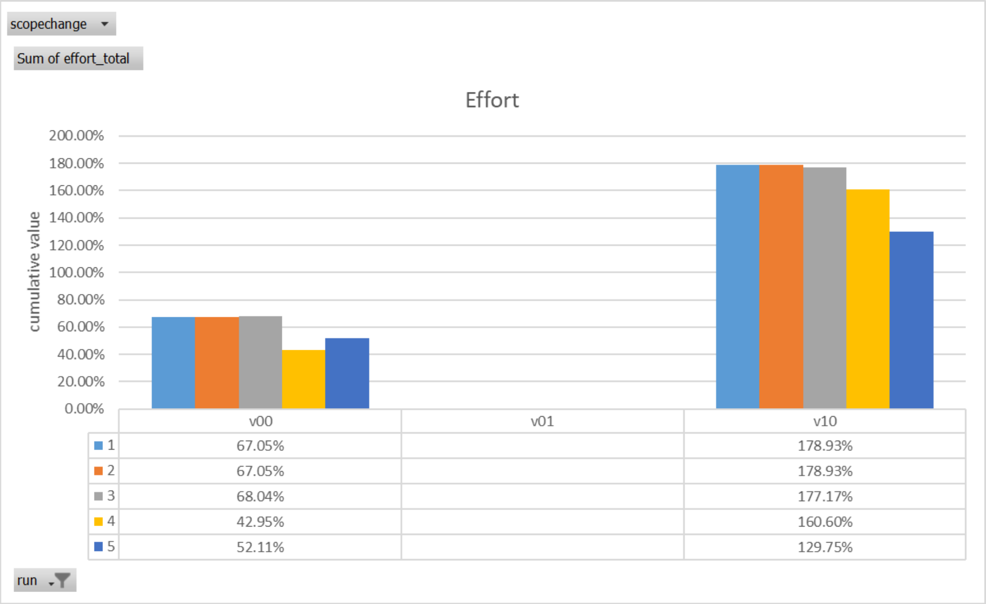
- Test: There is not enough evidence to conclude that there are differences among the means at the 0.05 level of significance.
- Comparison Chart: Blue intervals indicate that the means do not differ significantly.

Figure 71: One-way Anova for iterations by model in subset MID

When comparing the extreme values, the results are also displayed relative to each other (i.e., normalized) (see Figure 72 through Figure 77). Here, run V01 serves as the basis, since it possesses the smallest values, making the increase in the other runs more recognizable.

Again, the relative values for the output parameters as well as the interaction plots are presented.

The result is comparable to the subset HIGH result. The interaction plots indicate that the input parameters exert less of an influence for sub-models four and five (single Kanban and APM/Scrum).



	v00	v01	v10
■ 1	67.05%		178.93%
■ 2	67.05%		178.93%
■ 3	68.04%		177.17%
■ 4	42.95%		160.60%
■ 5	52.11%		129.75%

Figure 72: Relative differences in effort depending on model and run for subset MID

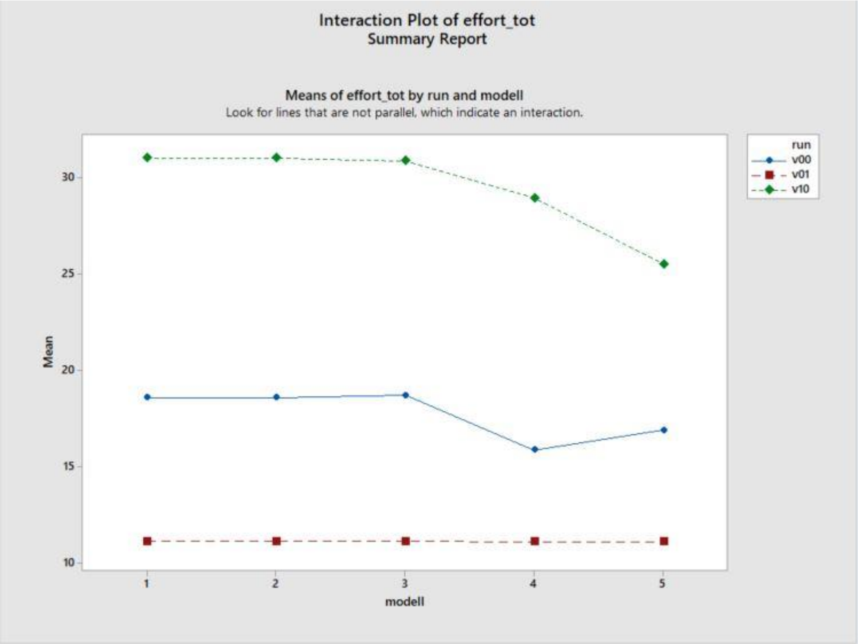
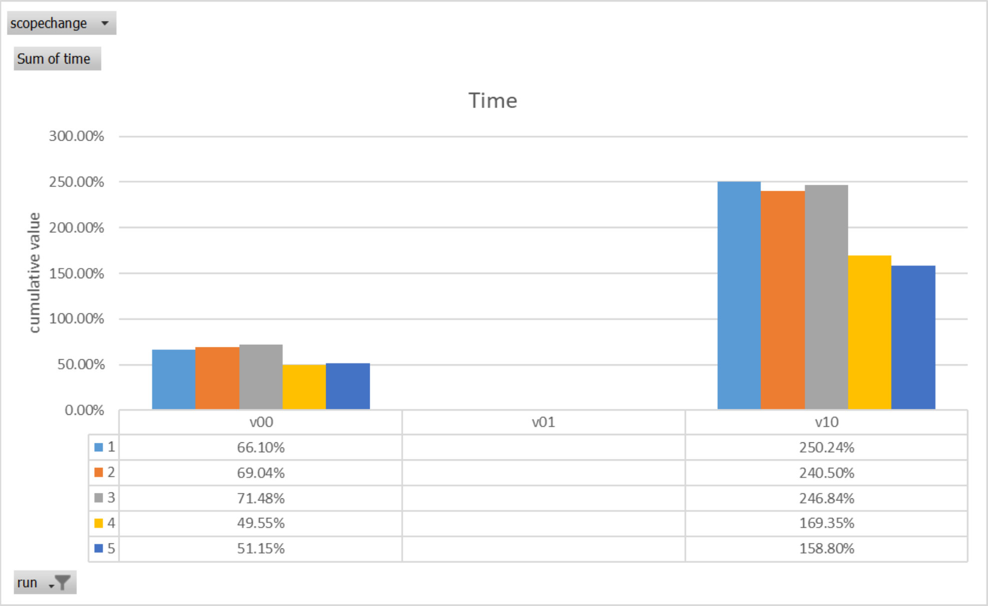


Figure 73: Mean values of effort per run depending on the model for subset MID



	v00	v01	v10
1	66.10%		250.24%
2	69.04%		240.50%
3	71.48%		246.84%
4	49.55%		169.35%
5	51.15%		158.80%

Figure 74: Relative differences in time depending on model and run for subset MID

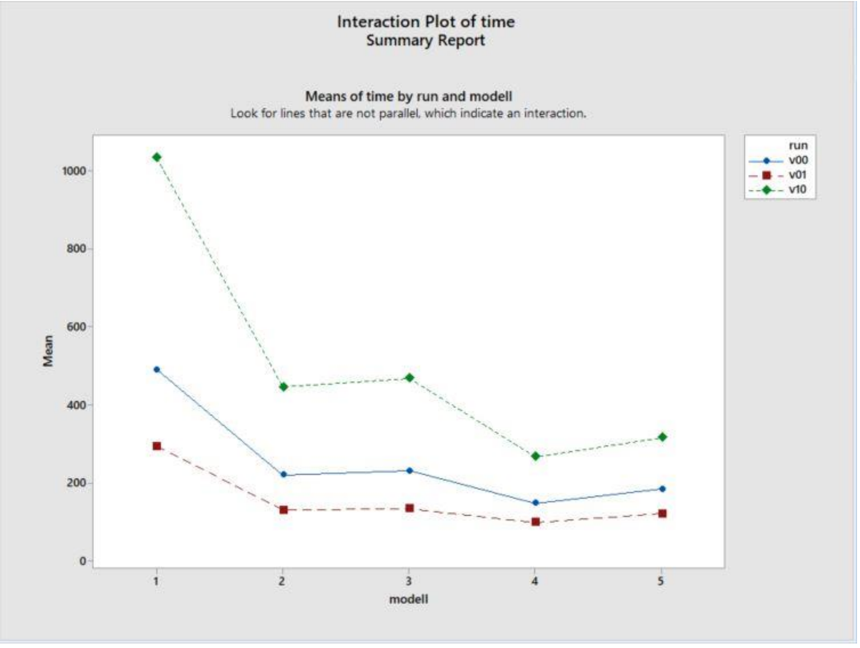
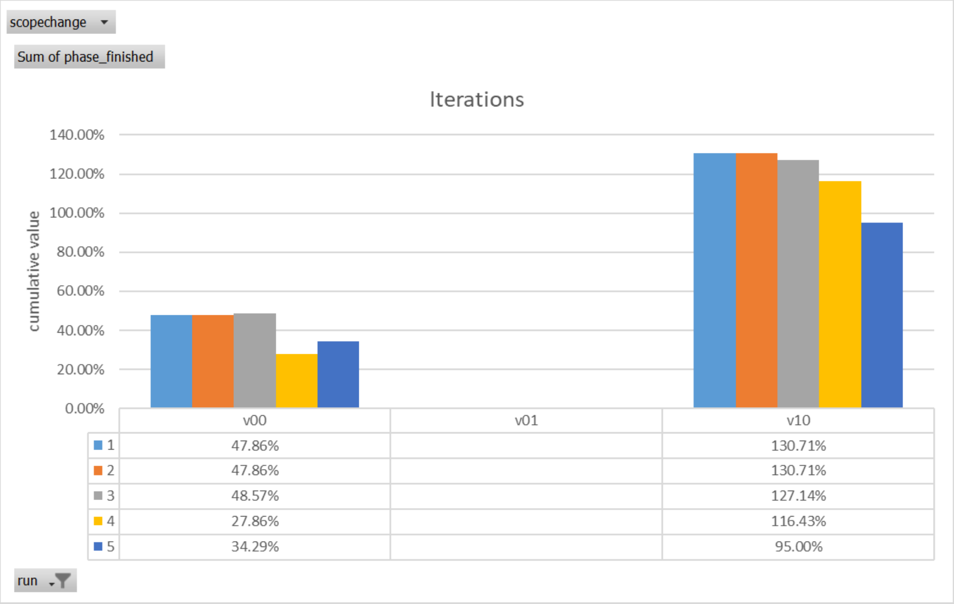


Figure 75: Mean values of time per run depending on the model for subset MID



	v00	v01	v10
🟦 1	47.86%		130.71%
🟧 2	47.86%		130.71%
⬜ 3	48.57%		127.14%
🟨 4	27.86%		116.43%
🟦 5	34.29%		95.00%

Figure 76: Relative differences in iterations depending on model and run for subset MID

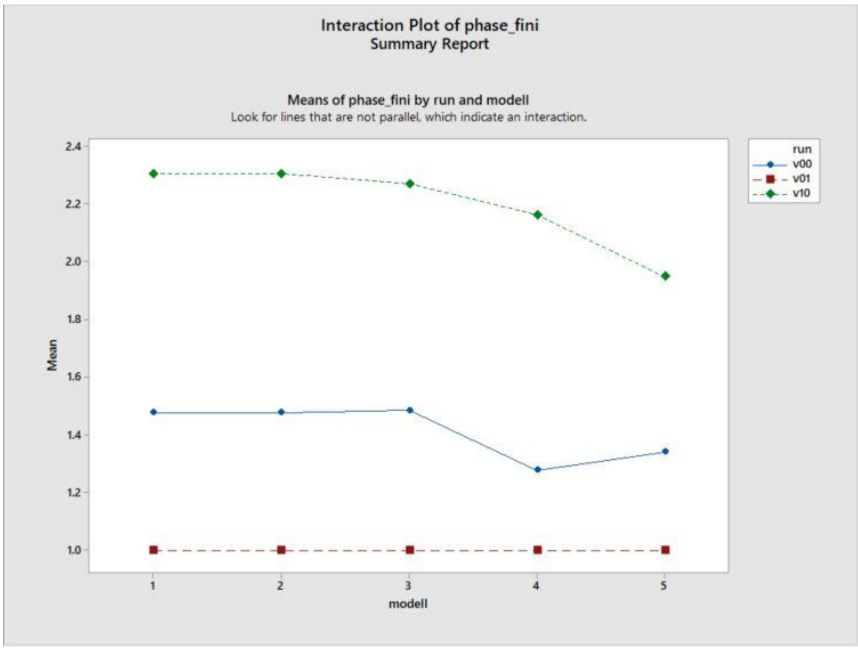


Figure 77: Mean values of iterations per run depending on the model for subset MID

8 INTERPRETATION AND DISCUSSION OF THE RESULTS

8.1 GENERAL INTERPRETATION

The statistical evaluations demonstrate significant differences of the output values depending on the project management approach (according to the simulation models) and the input parameters (in combination according to the runs). The respective results are related to the hypotheses in the following chapter. In general, more significant effects of the input parameters compared to the models' influence can be recognized. Even if this corresponds to experience from everyday project work, it should not be interpreted as a quantified result. The input parameters were artificially abstracted, whereby the direct effects as well as the learning effects regarding these parameters are derived from practice, but nevertheless cannot serve to determine absolute results. The sensitivity analysis (see "7.3.4.9 Sensitivity analysis") has revealed that the reviews' calculations, which led to the simulation results based on the input parameters, generate plausible effects. However, these results can only be compared relatively to each other. Thus, the parameters' effects cannot be directly compared with the models' effects, but a statement can be made as to whether these effects influence each other. This is nevertheless sufficient to answer the hypotheses.

8.2 REFERENCE TO THE HYPOTHESES

Hypothesis group I states that the APM approach is more successful than the traditional approach. Subtheses A through C detail this statement with regard to the success criteria of cost (effort), time and quality (rework, iterations).

To compare the different PM approaches, it makes sense to use run V00, as this represents an average project. Then, the average of all runs is considered.

With regard to hypothesis I.A, no significant differences are found between the models in run V00 (see Figure 41: One-way Anova for effort by model in run V00). On average over all runs, model five (APM/Scrum) exhibits lower efforts (see Figure 73: Mean values of effort per run depending on the model for subset MID). This results from the influence of the runs with the characterization HIGH, where this effect was statistically significant.

Therefore, hypothesis I.A could not be fully confirmed.

The shorter project duration of the APM compared to the TPM mentioned in hypothesis I.B could be seen in all runs as significant differences (see Figure 43: One-way Anova for time by model in run V00 and Figure 75: Mean values of time per run depending on the model for subset MID for the average).

Therefore, hypothesis I.B is confirmed.

Concerning the quality aspect (required rework and iterations), no significant difference occurs in run V00 (see Figure 45: One-way Anova for iterations by model in run V00), but in the average of all runs, there is (see Figure 77: Mean values of iterations per run depending on the model for subset MID). Here, too, the HIGH characterization runs exert an effect.

Therefore, hypothesis I.C could not be fully confirmed.

Hypothesis group II describes the influence of the project properties (i.e., the framework conditions reflected in the input parameters) on the first result. Subtheses A through C state that APM is less susceptible to the influence of less favorable project requirements (i.e., more challenging projects in terms of newness, difficulty, complexity and maturity) regarding the project outcome in terms of effort, time and quality.

For the investigation of hypothesis group II, the MID subset (i.e., the comparison of the simplest, the average and the most challenging case) and the statistics based on the runs are utilized.

In the MID subset, statistically significant differences were found with regard to all three criteria that demonstrate the parameters' influence. Nevertheless, only the result for the output variable time was statistically meaningful when comparing the models.

As such, the results are further compared between the subsets LOW (low project requirements) and HIGH (challenging projects). This demonstrated that, in projects with high requirements, differences exist between the models that did not exist in those with low requirements. This can also be clearly seen from the interaction plots of both the MID subset and all runs, where the lines for effort and iterations are much narrower in model five than in model one. Overall, this leads to the conclusion that the APM approach is less sensitive to the increase in project requirements, or more beneficial in more challenging projects.

Therefore, hypotheses II.A, II.B and II.C could be confirmed.

8.3 DISCUSSION

When comparing the different PM approaches, TPM and APM were always explicitly mentioned. However, the research question also concerned which of these approaches better reflects the lean principles. In particular, models two (lean), three (gated Kanban) and four (single Kanban) should be considered. Model two can rather be assigned to the TPM (long project phases), since it only takes up the lean idea by processing the tasks within the project phase (pull principle and one-piece flow). Models three and four, which transfer the Kanban principle from the lean method to the APM, aid in answering this question. Model three, with its review gates at the end of each (long) project phase, represents more of a hybrid model, while model four, with its abandonment of these gates, clearly assumes the agile idea of rapid delivery of results.

It can also be seen in the graphical evaluation of the simulation results that model one (pure TPM) rather stands for itself. Models two and three (which take up lean principles, but are to be assigned to the TPM or the hybrid approach due to the gates at the end of the long project phases) form a group. Models four and five, which more consistently reflect the APM approach, represent the third group.

The observation that, for projects with higher requirements, the APM approach demonstrates advantages regarding the criteria of effort, time and quality is also due to the interaction between these criteria. If poor quality leads to more rework or iterations, this directly affects the effort to be made and automatically leads to an extended project duration.

The significantly longer project runtimes (even for simple projects) in the traditional approach result from the underlying process design and its representation in the simulation. If tasks, which are processed in batch, have to wait for the whole completion of all other tasks, or even worse, if the resources of a category (e.g., prototyping) remain idle until the previous step (in this case, development) is completely processed, then these resources are not used for this project to a large extent. Of course, in practice, the staff would not sit idly by, but instead most likely work on other projects. At the level of a multi-project landscape, therefore, several projects in the TPM have to be guided through the organization at the same time. This corresponds on a large scale to batch production on a small scale: All in all, the same is performed, but the lead time for the individual project is considerably extended, so the project results (e.g., a newly developed product) are not available (i.e., could be marketed) as early as they could be if this project had been fully focused on.

Both considerations, including the APM's required focus on the project on the one hand (i.e., the associated faster provision of project results), and on the other hand, the faster learning effects resulting from the shorter-cycle delivery with its positive effects on the project, suggest that APM possesses an advantage over TPM—the more challenging a project is, the more benefits APM provides.

8.4 CONCLUSION

It was demonstrated that the different project management approaches—both the TPM and APM, as well as the principles of lean development —can be transferred into a simulation. By means of this simulation, the PM approaches could then be compared. As a result, the classical criteria of PM success—costs, time and quality—were evaluated. In so doing, the agile approach was identified as superior to the traditional approach, especially for the time required. With effort (costs) and additional iterations (quality), however, the differences are not so significant. Only for more challenging projects was the agile approach more promising.

This is in line with the literature's views that APM should especially be used in an environment where the product requirements remain rather unclear at the beginning of the project, or else change more frequently during the course of the project.

8.5 LIMITATION

To evaluate the different approaches, the so-called "iron triangle" was employed. Although these success criteria are often presented as being insufficient in the literature, this investigation is limited to evaluating costs, time and quality.

The simulation ends with the development's completion. Thus, only the information available at this time can be evaluated (i.e., only the project management success can be determined).

An important motivation of the APM concerns the desire to develop a product that better meets the actual needs of the (end) customer. In order to determine the project's success, later usage or sales figures would have to be included in the analysis. Another possibility would have been to evaluate individual tasks' importance or significance from the customer's point of view and to determine whether "more" is delivered here, or "faster". In the simulation, a task or feature's importance for the customer was mapped via the "priority" parameter, but this was to determine the task order in the simulation. Thus, the priority was already used as an input or control variable, and thus could no longer be analyzed as a result.

8.6 FUTURE RESEARCH

It would certainly be interesting to remove this limitation and investigate whether the APM is really developing a "better" product. For example, parameters could be introduced for customer benefit and related sales probabilities. The increasing cost of changes could also be taken into account as the project progresses.

In this thesis, only one single project was considered. Naturally, no resource conflicts occurred between different projects, which are often encountered in practice. Further research could thus investigate a multi-project landscape with the typical bottlenecks in shared resources.

9 LITERATURE

Adler, P. S., Mandelbaum, A., Nguyen, V., & Schwerer, E. (1995). From Project to Process Management: An Empirically-Based Framework for Analyzing Product Development Time. *Management Science*, *41*(3), 458–484. https://doi.org/10.1287/mnsc.41.3.458

Anderson, D., Augustine, S., Avery, C., Cockburn, A., Cohn, M., Highsmith, J., & Wysocki, R. K. (2005). Declaration of Interdependence. Retrieved March 6, 2017, from Declaration of Interdependence website: https://pmdoi.org/index.html

Atkinson, R. (1999). Project management: Cost, time and quality, two best guesses and a phenomenon, it's time to accept other success criteria. *International Journal of Project Management*, *17*(6), 337–342.

Avots, I. (1984). Informations systems for matrix organizations. In *Cleveland David I (Ed.) Matrix Management systems handbook* (pp. 535–537). New York: Van Nostrand Reinhold.

Baccarini, D. (1999). The Logical Framework Method for Defining Project Success. *Project Management Journal*, *30*(4), 25–32.

Bassil, Y. (2012). A simulation model for the waterfall software development life cycle. *ArXiv Preprint ArXiv:1205.6904*. Retrieved from https://arxiv.org/abs/1205.6904

Bhuiyan, N. (2011). A Framework for successful new product development. *Journal of Industrial Engineering and Management*, *4*(4). https://doi.org/10.3926/jiem.334

Binder, J., Aillaud, L. I., & Schilli, L. (2014). The Project Management Cocktail Model: An Approach for Balancing Agile and ISO 21500. *Procedia - Social and Behavioral Sciences*, *119*, 182–191. https://doi.org/10.1016/j.sbspro.2014.03.022

Boehm, B. W., & Turner, R. (2004). *Balancing agility and discipline: A guide for the perplexed*. Boston: Addison-Wesley.

Boehm, B. W., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE Software*, *22*(5), 30–39.

Cho, S.-H., & Eppinger, S. D. (2005). A Simulation-Based Process Model for Managing Complex Design Projects. *IEEE Transactions on Engineering Management*, *52*(3), 316–328. https://doi.org/10.1109/TEM.2005.850722

Clancy, T. (1995). The Standish Group Report. *Chaos Report*. Retrieved from https://www4.in.tum.de/lehre/vorlesungen/vse/WS2004/1995_Standish_Chaos.pdf

Cockburn, A. (n.d.). Process: The Fourth Dimension (Tricking the Iron Triangle). Retrieved January 10, 2006, from Humans and Technology website: http://alistair.cockburn.us/crystal/articles/ptfd/processthefourthdimension.htm

Collins, G. (2014). Agile Project Management. In E. I. A. Lester, *Project Management, Planning and Control: Managing Engineering, Construction, and Manufacturing Projects to PMI, APM, and BSI Standards* (6th edition, pp. 523–538). Retrieved from doi.org/10.1016/B978-0-08-098324-0.15001-2

Cooke-Davies, T. (2002). The "real" success factors on projects. *International Journal of Project Management*, *20*(3), 185–190. https://doi.org/10.1016/S0263-7863(01)00067-9

Cooper, R. G., & Sommer, A. F. (2016). Agile-Stage-Gate: New idea-to-launch method for manufactured new products is faster, more responsive.

Industrial Marketing Management, *59*, 167–180. https://doi.org/10.1016/j.indmarman.2016.10.006

Davis, K. (2016). A method to measure success dimensions relating to individual stakeholder groups. *International Journal of Project Management*, *34*(3), 480–493. https://doi.org/10.1016/j.ijproman.2015.12.009

De Wit, A. (1988). Measurement of project success. *International Journal of Project Management*, *6*(3), 164–170. https://doi.org/10.1016/0263-7863(88)90043-9

DIN 69901-5:2009-01 Project management - 🐦"Project management systems "Part 5: Concepts. (2009, January). Retrieved from https://dx.doi.org/10.31030/1498911

Dr. Angermeier, G. (2016, September). Projektmanagement kompakt. Retrieved December 30, 2016, from Www.projektmagazin.de website: https://www.projektmagazin.de/projektmanagement-kompakt

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, *50*(9–10), 833–859. https://doi.org/10.1016/j.infsof.2008.01.006

Engelhardt, N. (2019). Comparison of Agile and Traditional Project Management: Simulation of Process Models. *Acta Academica Karviniensia*, *19*(2), 15–27. https://doi.org/DOI: 10.25142/aak.2019.011

Erne, R. (n.d.). Lean Project Management. Retrieved February 11, 2016, from GPM Infocenter - 🌀"Lean Project Management website: http://www.gpm-infocenter.de/uploads/PMMethoden/Lean%20Project%20Management.pdf

Fernandez, D. J., & Fernandez, J. D. (2008). Agile Project Management 🌀"Agilism versus Traditional Approaches. *The Journal of Computer Information Systems*, (Winter), 10–17. https://doi.org/10.1080/08874417.2009.11646044

Gardiner, P. D., & Stewart, K. (2000). Revisiting the golden triangle of cost, time and quality: The role of NPV in project control, success and failure. *International Journal of Project Management*, *18*(4), 251–256.

Garel, G. (2013). A history of project management models: From pre-models to the standard models. *International Journal of Project Management*, *31*(5), 663–669. https://doi.org/10.1016/j.ijproman.2012.12.011

Gaupin, G., & International Project Management Association (Eds.). (2006). *ICB: IPMA competence baseline ; Version 3.0*. Nijkerk: IPMA.

Griffiths, M. (2004). *Using Agile Alongside the PMBOK*. Presented at the PMI Global Congress Proceedings, Anaheim, California.

Harris, M. (2007). *Agile and the Declaration of Interdepencence 🐦"A new approach to Process Improvement Projects"* Retrieved from http://www.compaid.com/caiinternet/ezine/harris-agile.pdf

Highsmith, J. (2001). The Agile Manifesto. Retrieved from www.agilemanifesto.org

Hines, P., Holweg, M., & Rich, N. (2004). Learning to evolve: A review of contemporary lean thinking. *International Journal of Operations & Production Management*, *24*(10), 994–1011. https://doi.org/10.1108/01443570410558049

Kerzner, H. (2009). *Project management: A systems approach to planning, scheduling, and controlling* (10th ed). Hoboken, N.J: John Wiley & Sons.

Koskela, L., & Howell, G. (2002, June). *The underlying theory of project management is obsolete*. Retrieved from http://usir.salford.ac.uk/9400/

Lester, A. (2014). *Project Management, Planning and Control: Managing Engineering, Construction, and Manufacturing Projects to PMI, APM, and BSI Standards* (6th edition). Elsevier.

Leutze, J. (2014). *Eine neue Definition agilen Projektmanagements*. BoD.

Lewis, J. P. (2002). *Fundamentals of project management: Developing core competencies to help outperform the competition* (2nd ed). New York: AMACOM.

Liker, J. K., & Morgan, J. M. (2006). The Toyota way in services: The case of lean product development. *The Academy of Management Perspectives*, *20*(2), 5–20.

Locher, D. A. (2008). *Value Stream Mapping for Lean Development: A How-to Guide for Streamlining Time to Market*. New York: CRC Press.

Mascitelli, R. (2002). *Building a Project Driven Enterprise: How to Slash Waste and Boost Profits through Lean Project Management*. Northridge, CA: Technologies Perspectives.

Morgan, J. M., & Liker, J. K. (2006). *The Toyota product development system: Integrating people, process and technology*. Boca Raton, FL: CRC Press (formerly Productivity Press).

Mund, K. (2014). From lean manufacturing to lean development. *Transfer Inovacii*, (29), 12–15.

Munns, A., & Bjeirmi, B. F. (2015). The role of project management in achieving project success. *International Journal of Project Management*, *14 No 2*(1996), 81–87.

Murthy, D. N. P., Rausand, M., & Østeras, T. (2008). New Product Development. *Product Reliability: Specification and Performance*, 15–36.

Oehmen, J., Oppenheim, B. W., Secor, D., Norman, E., Rebentisch, E., Sopko, J. A., & others. (2012). *The guide to lean enablers for managing engineering programs*. Retrieved from http://dspace.mit.edu/handle/1721.1/70495

Oehmen, J., & Rebentisch, E. (2010). *Waste in lean product development*. MIT Whitepaper Series, Boston.

Oppenheim, B. W. (2011). *Lean for Systems Engineering with Lean Enablers for Systems Engineering.* Hoboken, N.J: Wiley & Sons, Inc.

Osorio, P. C. F., Quelhas, O. L., Zotes, L. P., Shimoda, E., & others. (2014). Critical Success Factors in Project Management: An Exploratory Study of an Energy Company in Brazil. *Global Journal of Management And Business Research*, *14*(10). Retrieved from http://www.journalofbusiness.org/index.php/GJMBR/article/view/1483

Owen, R., Koskela, L. J., Henrich, G., & Codinhoto, R. (2006). Is agile project management applicable to construction? In Sacks, R. & Bertelsen, S. (Eds.), *Proceedings of the 14th Annual Conference of the International Group for Lean Construction* (pp. 51–66). Retrieved from http://usir.salford.ac.uk/9369/?utm_source=twitterfeed&utm_medium=twitter

Parsaei, H. R., & Sullivan, W. G. (1993). *Concurrent Engineering 🐦"Contemporary Issues and Modern Design Tools* (1st ed.). London: Chapman & Hall.

Pessôa, M. V. P., Loureiro, G., & Alves, J. M. (2007). An approach to lean product development planning. In *Complex systems concurrent engineering* (pp. 229–237). Retrieved from http://link.springer.com/chapter/10.1007/978-1-84628-976-7_26

Pons, D. (2008). Project management for new product development. *Project Management Journal*, *39*(2), 82–97. https://doi.org/10.1002/pmj.20052

PRINCE2 🌀"Official Website. (2009). Retrieved from https://www.axelos.com/best-practice-solutions/prince2

Project Management Institute (Ed.). (2013). *A guide to the project management body of knowledge (PMBOK guide)* (Fifth edition). Newtown Square, Pennsylvania: Project Management Institute, Inc.

Salameh, H. (2014). What, When, Why, and How? A Comparison between Agile Project Management and Traditional Project Management Methods. *International Journal of Business and Management Review*, *2*(5), 52–74.

Seibert, S. (2007, January). Agiles Projektmanagement. *Projektmanagement Aktuell*. Retrieved from www.wiso-net.de

Serrador, P., & Pinto, J. K. (2015). Does Agile work? — A quantitative analysis of agile project success. *International Journal of Project Management*, *33*(5), 1040–1051. https://doi.org/10.1016/j.ijproman.2015.01.006

Serrador, P., & Turner, R. (2015). The Relationship between Project Success and Project Efficiency. *Project Management Journal*, *46*(1), 30–39. https://doi.org/10.1002/pmj.21468

Shenhar, A., & Dvir, D. (2007). *Reinventing project management: The diamond approach to successful growth and innovation*. Boston, Mass: Harvard Business School Press.

Sliger, M. (2006). *A Project Manager's Survival Guide to Going Agile*. Retrieved from https://cs.anu.edu.au/courses/comp3120/public_docs/Sliger_a_project_manager-s_survival_guide_to_going_agile.pdf

Sliger, M., & Consulting, S. (2008). Agile Project Management and the PMBOK® Guide. *PMI Global Congress Proceedings*, 7. Denver, Colorado, USA.

Stare, A. (2014). Agile Project Management in Product Development Projects. *Procedia - Social and Behavioral Sciences*, *119*, 295–304. https://doi.org/10.1016/j.sbspro.2014.03.034

Stepanek, G. (2005). *Software project secrets: Why software projects fail ; [resolving the conflicts between software development and project management]*. Berkeley: Apress.

Tolbert, M. (2012). Part Five of AgileProject Management vs. PMBOK® Guide and EVM "Revolution or Evolution" PMI Washington DC. Retrieved from https://www.pmiwdc.org/article/mark-tolbert/part-five-agileproject-management-vs-pmbok%C2%AE-guide-and-evm-%E2%80%9Crevolution-or

Vinekar, V., Slinkman, C. W., & Nerur, S. (2006). Can agile and traditional systems development approaches coexist? An ambidextrous view. *Information Systems Management*, *23*(3), 31–42.

Womack, J. P., & Jones, D. T. (2003). *Lean thinking: Banish waste and create wealth in your corporation* (2nd ed.). New York: Simon & Schuster.

Wysocki, R. K. (2006). *Effective software project management*. Indianapolis, IN: Wiley Pub.

Yassine, A. A. (2007). Investigating product development process reliability and robustness using simulation. *Journal of Engineering Design*, *18*(6), 545–561. https://doi.org/10.1080/09544820601011690