AALBORG UNIVERSITET
STUDENTERRAPPORT

PROJECT MANAGEMENT IN DIGITAL VISION-DRIVEN MEDIA CREATION

Mikkel Aagaard Bækgaard – 20145763

Specialerapport

AALBORG UNIVERSITET
STUDENTERRAPPORT

Specialesemester
Interaktive Digitale Medier
Rendsburggade 14
9000 Aalborg
Aalborg University

Titel:

Project Management in Digital Vision-Driven Media Creation.

Projektperiode:

Efterårssemestret 2021

Projektdeltagere:

Mikkel Aagaard Bækgaard – 20145763

Vejleder

Claus Andreas Foss Rosenstand

Sider:

48

Anslag og normal sider:

10664 anslag. 44,4 Normal sider.

Afleveringsdato:

28. september 2021

Abstract:

I dette specialeprojekt, blev der igangsat en undersøgelse I projektstyring ved at fordele udførelsen af denne, i to idealtyper, navngivet Hård projektstyring og Blød projektstyring. Dette sprang ud fra forfatterens egne erfaringer og observationer under hans tid som Project Manager hos DADIU, et uddannelsesforløb i spiludvikling. Ved at tage udgangspunkt I sin empiri derfra, blev de to idealtyper af projektstyring udforsket gennem udvalgte literaturer som alle demonstrerede hvad disse idealtyper bestod af. Gennem sit empiriske perspektiv og udforskningen af litteratur, blev resultaterne derfra samlet i et skema som kortlægger adskillige dimensioner af projektstyring, hvori svaret på hvad det vil svare til indenfor den hårde eller bløde idealtype. Dette skema kortlægger dermed også et sæt af mulige heuristikker som giver svaret på rapportens Research Question, angående brugen af hård og blød projektstyring.

Indhold

Introduction

This master thesis' area of interest is project management in digital vision-driven media creation. Originating from a curiosity on the interplay between the responsibilities a project manager can hold when overseeing a production involving diverse tasks and rationales.

When you are given the task of heading a project and overseeing its completion, there are many factors at play based on the variables that are being worked with.

The type of project being undertaken will describe the tasks involved: software development, event planning, construction, etc. If the project is large enough to require a team, what types of roles and functions are active within the team, and what is the management structure like? What interplay lies between the roles and what sort of communication is used for them? Are there special accommodations to be made due to roles having very specialized focuses, to ensure smoother cooperation? How close are you to the team you are managing, physically, and socially?

Additionally, there are also many decisions to be made before a project is being realized, namely during the product formation phase, that seeks to resolve tasks (when applicable) such as scheduling time, hiring, location scouting, materials, and budgets, among other things based on the project's available resources (money, equipment, staff, talent, locales), needs, and aim.

The management positions in a project all have different responsibilities, that focus on their specific field, but as part of the same project, they all affect and rely on each other's decision-making abilities to hit the mark where it is intended to be, and all the above-mentioned questions are part of objectives that leadership and management positions must consider to reach their goals. As such, the matter of how these objectives are met becomes a matter of methods and means. My initial observation divided these into two separate aspects, focusing on a human and social role versus a methodological one. These can be extended to follow three primary rationales as established by Immanuel Kant (Kant 2005): a theoretical rationale, dealing with logical implications and conclusions, an aesthetic rationale grounded in emotions and experiences, and the normative rationale; working towards what is most practical based on prescriptive actions or norms.

This thought is based on my education regarding project management and subsequently encountered literature e.g. McCarthy, J. (1995), PRINCE2 – Projektledelse med success [Successful Project Management] (2010), and Schwaber K. (2004), combined with the thoughts and experiences I have had while working on several different software-based projects as a university student at the Medialogy Bachelor Programme, and later as the project manager of a team of 14 students (Bækgaard, 2020) creating games during a DADIU

(National Academy of Digital, Interactive Entertainment, DADIU (2021)) course throughout my 9th semester. The DADIU course is a collaboration between several arts- and higher institutions in Denmark, being brought together with the purpose of teaching and providing practical work and guidance into the world of video game development. There I had the role of overseeing and managing the progress and interplay of several disciplines forming into a game, a vision-driven product meant for entertainment purposes, vision-driven referring to the vision of the director behind the product, the man or woman holding the role of being the unifying factor in the development, so the intended experience the product is meant to provide. To point to an earlier point, this role is in theory primarily done through an aesthetic rationale, as the "vision" usually seeks to evoke a certain emotion or experience.

Comparing the experience at DADIU of a team of 14 people, to the smaller project groups of two to six people on regular semesters, where group members still had their expertise, but everyone sat across from each other and contributed to several or all areas of the project, where the products of these projects being practical software with a goal of testing hypotheses, of which a report was written, expanded my insight and perspective into the complexity of uniting the efforts of such media creation.

When entering into this role as project manager, especially with limited to no experience, the matter of your responsibilities and how to perform the best, quickly becomes a concern, as your position as part of the team's leadership means people will look to you for answers, eg: pointing people towards the materials they have available for their work or knowing how to best set them up with a visual guide of their progress. You have to figure out from where you can pull those answers. Since your purpose in the team is to make decisions that will help make sure deliverables are on schedule, your focus is to make sure the team is efficient at their tasks.

So, to make these optimizing decisions, where do you pull the answers? Generally, you pull answers from e.g. the knowledge you possess, and the experiences you have had so you can make an appropriate suggestion based on a value judgement of the situation, but if you don't possess the relevant knowledge to make an informed decision, then the knowledge must come from someone else, and if the closest expert on the topic at hand is the one asking for help, then you must ask them the right questions, so work can maintain its flow with the least amount of time wasted.

As a project manager for a DADIU team, something we were often told about our role from the teachers, was that we were meant to facilitate the solutions for our team, not to create or force the right answer for them. By making sure the team has the tools and opportunity to reach their own best conclusion. A particular metaphor that was used, was the project manager moving from being the team's mom to becoming the team's Gandalf. The intent behind this metaphor was journeying from being the person relied on for the simplest things and telling people how to do everything, which is not the job of the project manager, to simply being there when the dwarves, or rather the team, really needs a guiding hand, like facilitating a process for how to work things out, providing the grand overview of the progress to provide perspective to those working in the details and improve the autonomy of the team.

So far, a couple of things that have been identified to be required of a project manager, lie not only within a realm of knowing when and how things should be done, but also being able to have the insight to recognize one's lack of knowledge, but supplanting it by being able to draw it out from the team you trust to have it. As I was introduced to the idea of the role of project manager at DADIU, I expected it to be filled with schedule making, and following tried and true management methods, but as more and more communication happened with teachers, and the projects moved further along, it became apparent that, in my case, the need for someone to draw forth the tools, and make sure the right people were talking, did more for the team than any schedules and plans for production timelines did; Because plans change due to new ideas, failure to meet expectations or unforeseen setbacks, and being able to adjust to that and reaching a firm compromise that team-leads agree on, will help production continue smoothly towards the project's aim. And that will help the team more than being stressed or disappointed over a missed date for a goal post.

That said, the compromise as mentioned should be firm. This applies to other parts of work the team does as well. Because if part of the team is not delivering, then the problem factor needs to be identified ASAP, and then we reach into the leadership part of project management with human interaction once again, akin to figuring out the best way to facilitate the team's needs. Confronting people with their failings does however present a risk if handled poorly. Be it pride or anxiety, a person possessing either trait (or both) can be difficult to handle when shown their errors, and without due delicacy, the matter of their current place in the team might be put into question. However, letting the issue be, risks it repeating, leaving either option with undesirable risks, but with a preference to the one leaning on growth as a team or individual. Trusting your team to get the job done is vital, if they can't back up that trust with their efforts, a solution must be found, to avoid stagnation or further failures.

This risk illustrates a need for plans to be changeable to accommodate changes to the project, that occur as mentioned earlier due to the occurrence of unwanted situations, or acquisition of knowledge that informs a change of plans (i.e., the project's vision changes as the director felt inspired to include something specific, or the priorities of stakeholders' changing). This makes the planning of a project very difficult, if not impossible, to nail perfectly when done in a linear fashion, which is why recognizing the inevitability of setbacks, and planning for changes can help the managing of seeing a project from start to finish. This approach to software development is specifically named *agile* which is widely used these days, and I have personally become more and more familiar with it throughout project work and education.

The agile approach to software development has a set of four values in the manifesto that was created when the approach was coined, of which one specifically refers to the flexibility in planning mentioned above. (*Manifesto for Agile Software Development* 2002). Two of these values, namely "*Individuals and interactions over processes and tools*'' and "*Customer collaboration over contract negotiation*" deal with the prioritization of aspects about the humans in the project, over the more methodological and practical sides to managing it, arguing that a focus on what I would term *soft aspects* of project management makes for more effective development, compared to the counterparts that I have termed *hard aspects*. To explain it briefly, and in a relatively loose manner, the soft aspects deal with the interpersonal communication issues while the hard aspects deal with issues that can be accounted for and measured. This without discounting the importance of the *hard* management aspects such as specific tools, chosen processes, and methods. These two classifications could also be described as ideal types as per Max Weber's theory of the same name (*Max Weber (Stanford Encyclopedia of Philosophy)*, 2017), which is a method wherein the ideal types are created as a tool for analyzing phenomena by comparing them and learn of their differences and similarities, to develop new hypotheses and in theory, by developing these ideal types one will also learn what lies in-between them.

Much of this lines up with my own experience as described earlier. But a wondering still comes to mind. Where does the line go? This prioritizing of *soft* leadership methods, over *hard* management, is an easy enough ideal to speak of, but to follow an ideology that allows for expression and inspired ideas (aesthetic rationales), when the execution of these goes against technological constraints (theoretical rationale), is not always possible given factors like time and external pressure (normative rationale). On the other hand, being the project manager, your purpose is to account for many normative functions, and putting that aside can also be a difficult thing to realize the correct time for. The leadership and interaction with people, stretch over many functions, but for this paper when it comes to specifics on managing people, it will be delimited to within project teams, meaning the focus will not be on the interactions with customers and suppliers as they are external actors to the project team, and is not something I will claim any kind of experience with.

1.2 Research Question and Design, and Publication Design

In summation of what I have written in the above description of the area of interest, as part of my research design for this project, I state the following research question:

With an empirical point of departure in my vision-driven digital project management experience at DADIU; when to apply hard or soft project management in vision-driven digital projects?

The above research question serves as the primary explanation of what I seek to explore within this project. Below, I detail the work questions that I saw fit to help elaborate and explore my area of interest, through a hermeneutic phenomenological process making use of the work questions to divide the "whole" represented by the research question, into smaller parts to expand upon and help in understanding it.

Work Question 1: What is my empirical experience with vision-driven digital project management at DADIU?

To further establish the foundation for my interest in this topic, I will explain my background and experiences as well as the introspection I had during this time, by looking back at the files and documents created during the process of my DADIU semester, as well as the main report I wrote to finish the semester, to affirm my empirical standpoint for this report. This leads me to take the first work question and divide it into an explanation of DADIU and my experience there, as well as explaining vision-driven digital project management and establish how it differs from "regular" project management.

Work Question 2: What is hard and soft project management?

For the second work question, it becomes important to further explain the meaning that I apply to these aspects that I wish to investigate and discuss, beyond the simple examples I provided within the area of interest. To do this I will draw from different literature, such as *"Dynamics of software development", "Agile Project Management with Scrum", "Prince2 - Successful Project Management", "Software project management: a process-driven approach",* that I consider to each highlight and explain elements making up *hard* and *soft* project management. I will also use my own experiences, accounted for during work question 1, to answer what it is that characterizes hard and soft project management, and what points of relation can be established between them. With these, the goal of explaining "what", will come through the use of examples that showcase the understanding I have come to, of why these two different aspects are, how they truly differ, but also in what areas they intersect.

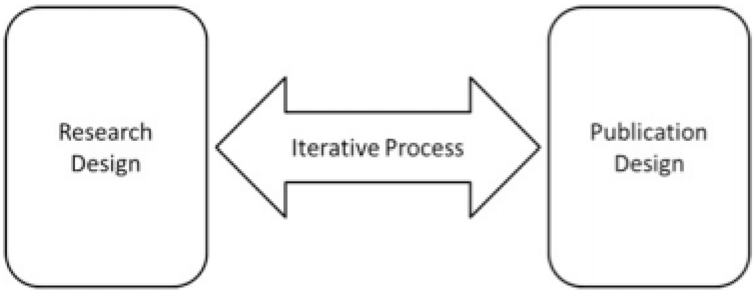
Work Question 3: When to apply hard and soft project management?

The last work question is where I seek to apply what I have experienced and learned, and discuss how the results garnered from the previous work questions would be best applied for practical purposes within a vision-driven digital media production.

The formation of the research design has been an iterative process with at least 15 document versions (hermeneutic), where changes in e.g., a work question would affect the research question and vice versa.

As illustrated in Figure 1 (Mathiassen 2017, p. 2) this pinpoints the interdependence between the research and publication design mirroring the final publication order of work questions, as stated above. The publication design is outlined in Table 1 below:

Figure 1: Designing Engaged Scholarship



Note: By Mathiassen, L. (2017). Designing Engaged Scholarship:

TABLE 1

PUBLICATION DESIGN		
CHAPTER	Question	Heading
1	RQ: With an empirical point of departure in my vision-driven digital project management experience at DADIU; when to apply hard and soft project management in vision-driven digital projects?	Project management in digital Vision-Driven Media Creation
2	WQ 1: What is my empirical experience with vision-driven digital project management at DADIU?	My experience with vision-driven digital project management
3	WQ 2: What is hard and soft management?	Hard and soft management
4	WQ 3: When to apply hard and soft project management?	Applying hard and soft project management
5	RQ: With an empirical point of departure in my vision-driven digital project management experience at DADIU; when to apply hard and soft project management in vision-driven digital projects?	Conclusion

2. My experience with vision-driven digital project management

Work Question: *What is my empirical experience with vision-driven digital project management at DADIU?*

To establish the foundation for my interest in this topic, I will explain my background and experiences as well as the introspection I had during this. This will be done in this chapter by looking back at my time at DADIU, drawing from the files and documents, as well as the report, created during and for that semester. For this purpose, the chapter will be divided into sub-work questions to explore the topics that make the foundation for the answer to the chapter's Work Question at the top. starting with a description of the semester of DADIU I partook in, and how it was formatted during 2019, due to the nature of the specific role as a project manager I had during the time of that semester, which is what inspired undertaking this project. By diving into the role I had, the methods I used, and the lessons I took with me, after participating in this production of a vision-driven nature, with many moving parts, both digital and human, I mean to present the empirical foundation for my viewpoints on the aspects of project management.

As part of all this, I will also clarify the meaning behind the "Vision-Driven" descriptor, to better explain how it is set apart from regular project management, again using my own experiences as the entry point, and especially why it is relevant to the type of work that I have done and am focusing on.

The chapter itself will be structured into the following sub-sections, following a similar structure of the publication design, each focusing on a sub-work question, sequenced to build into the following sections.

Sub-Work Question 1: What is DADIU?

To explain my experiences, providing the information of what DADIU is and consists of is an obligatory first step to provide insight into its relevance to the Work Question. This question will encompass the What, Where, When, and Who of my semester with DADIU. Reviewing the purpose of DADIU, its structure, and the people it included, explaining the roles that were covered in the DADIU team I was part of, and the expert mentors filling an educational aspect of the programme. This is addressed in chapter 2.1 DADIU, which is further divided into the following work topics based on the above:

Sub-Work Question 1.1 My team and project

This topic will detail the people I worked with in DADIU. It will be a description of the team's structure as well as the purpose behind the specific roles within it, and where their responsibilities lie from the perspective of the DADIU programme. A description of the graduation game produced by the team will also be given, for the sake of visibility and ease of reference when it is discussed further along.

Sub-Work Question 1.1a: What does "Vision-Driven" entail?

This was briefly touched upon in 1.1 Introduction while first describing DADIU, due to its presence as a way to distinguish the rationale behind a creation process. It is an important descriptor for both the topic of this chapter and the overall thesis, so to add it to the lexicon an explanation will be provided. This will be done as part of the explanation of the Game Director role in part 2.1.1a Role Descriptions, as the concept and role are intrinsically connected.

Sub-Work Question 1.2: The competence lessons

Here I will touch on the more formally educational parts of the DADIU semester. Describing what, and who, it involved, the methods employed, as well as what I took from it.

Sub-Work Question 2: What was my role on the team?

Here I will go into the specifics of what I did in my capacity as project manager for the DADIU team. Clarifying my overall responsibilities, as well as discussing specific cases during the game developments, where the duty of choice was in my hands, to highlight how and where things could've gone wrong and why I made the decisions I made.

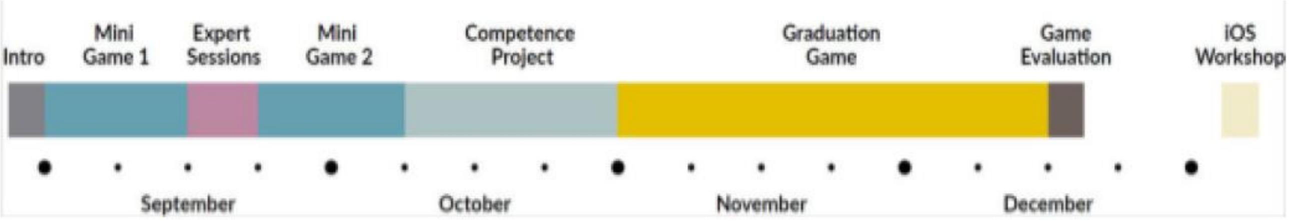
2.1 DADIU

"What is DADIU?" will be the question explored and answered in this section.

The National Academy of Digital Interactive Entertainment [Det Danske Akademi for Digital Interaktiv Unverholdning] (DADIU, u.d.), is a collaborative educational program between several art schools and higher institutions, to gather students from different disciplines, and granting a specialization opportunity within different video game development competencies, based on the student's educational background. The roles and competencies that students will participate as are as following: Art Director, Audio Designer, CG Artist, Game & Level Designer, Game Director, Game Programmer, Project Manager (currently dubbed Producer), QA & UR Manager, and Visual Designer. (DADIU, u.d.a)

The program itself is a semester-long venture, going from the end of August up to and including December, with a program consisting of the creation of two mini-games and one larger graduation game, with two periods of time in between productions, allotted for sessions and courses with educational content catered to the production competencies. The official schedule, as it was in 2019 during my time there, is visualized in figure 2 ("The Semester", 2019) below.

Figure 2: Visualization of the 2019 DADIU timeline



Note: The Semester, from https://www.dadiu.dk/semester during 2019

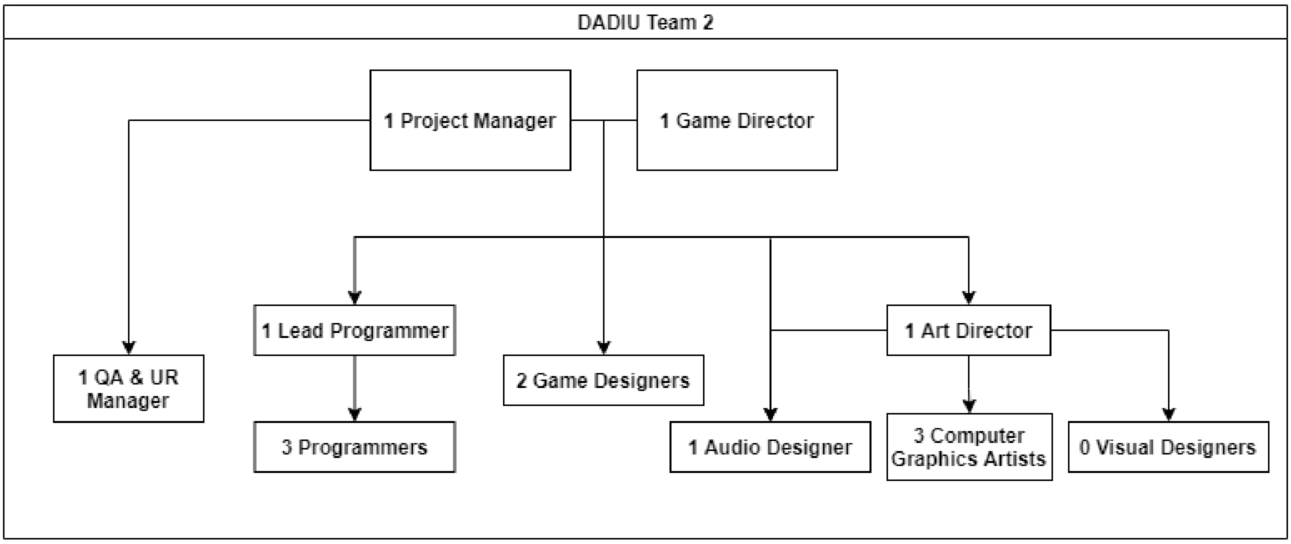
The accepted students are divided into six groups, who are then granted work locations in participating institutions in Aalborg and Copenhagen.

2.1.1 My team and project

In this section, the DADIU team I was part of will be explained as per sub-work question 1.1, explaining the structure we had and the responsibilities of the specific roles, and ending with a description of the game that was made.

The team I became a part of consisted of 14 people (myself included) with different backgrounds, and also were in different stages of their life (Bækgaard, 2020). The configuration of the team's members and structure is visualized in figure 3. The roles were split into three overall categories: Tech (programmers), Design (game designers), and Art (audio-visual assets). The QA/UR manager's role ran parallel to the efforts of all the other categories, and therefore belongs to neither and all. The roles themselves are also set up hierarchically, to indicate the level at which they perform, on a strategic level, and which person they respond to and who are responsible for their output.

Figure 3: Organigram of the DADIU Team



The hierarchical structure of the DADIU Team – assorted into strategical layers, and department category

The effective managerial responsibility of this hierarchy is defined in Rosenstand & Laursen (2013), where the Project Manager and Game Director is taking part in the top management, serving as strategic management and handling the commercial conditions of the project such as customers, external commands, and stakeholders. A condition that was simulated via presentations during and after the game productions for the primary teacher of DADIU who would bring along a guest, who had one or other type of experience with vision-driven media creation. The job of the top management is not to tell people *how* to do their jobs, as those people working with them are presumably more competent at their functions, but to communicate the execution criteria and constraints, so the leads can figure out what to do on an operational level.

The arrows as indicated in figure 3, are the structure of how I saw the lines of communication primarily going. The communication structure, in practice, was relatively flat and across the board between leads which will be noted in the role descriptions, but the hierarchy as structured meant that the progress and information flow had a certain priority as to whom it went to, and who held decision-making capabilities over who.

2.1.1a Role descriptions

Before talking more about the team, this section will step by step provide brief explanations of each role the team had in alphabetical order:

Art Director*:* Takes part as lead for the team in defining the concept for the game.

Responsible for the design, visual concept, production, and stylistic consistency of the visual aspects of the game, as well as marketing materials.

Ensures production of concept sketches, storyboards, and color palettes with the art team to define the game's visual concept.

With the lead programmer and art team: ensures the construction concepts and budgets for 2D and 3D elements.

With the Game Director, and Project Manager: responsible for planning and estimating the art team's workload according to time and available resources. (*DADIU,* u.d.A)

Audio Designer*:* Responsible for sounds, ambiance, voice recording, and in-game music.

Ensures production of the game sound to support the gameplay and the aesthetic experience as defined by the lead team.

With the programmers: ensures music and sounds work function properly within the game.

With the Game Director and Project Manager: responsible for planning and estimating the scope of the work according to time and available resources. (*DADIU,* u.d.A)

CG Artist: Responsible for creating the 3D graphical assets and animation assets for the game, under the supervision of the Art Director and Project Manager.

With the programmers: ensures that assets function properly within the game.

With the Art Director and Project Manager: responsible for planning and estimating the scope of the work according to time and available resources. (*DADIU,* u.d.A)

Game & Level Designer*:* Takes part as lead for the team in defining the concept for the game.

Responsible for realizing the concept within the project's scope.

Collaborate with the team to implement the design in the game.

With the Game Director and Project Manager: responsible for planning and estimating the scope of the work according to time and available resources. (*DADIU,* u.d.A)

Game Director*:* Possesses executive powers and responsibilities, but does not make decisions regarding schedule and implementation.

Responsible for defining the game concept, leading the team, and maintaining an overview of the game so the game concept can be realized to the best possible extent.

Part of setting up the testing of the game on the target audience.

Also responsible for communicating the concept and game ideas in a presentation, in collaboration with the team. (*DADIU,* u.d.A)

As mentioned in 1.1, the rationale with which the director makes their decisions for the direction of the game is aesthetic. The aesthetic drive is what powers the vision a director has for the product, what a game director has for the game he wants to make. The vision for the game concept is the spark of creative direction that unifies the work by the technical, design, and art departments in a game production, to realize the director’s vision. This is essentially what makes up the second part of the Game Director description. Having a vision for software projects, in general, is not anything special, that the aesthetic drive for this vision can be of such high importance, compared to other drives is what makes digital vision-driven media projects stand out. (Rosenstand & Laursen, 2013)

Game Programmer: Has technical responsibility for implementing the technical solutions for the game production i.e. builds, pipelines, tests.

Works on the technical structure(s) and system(s) of the game, as well as being the main proponent in optimization and bug fixing.

Specialized roles such as pipeline programmer, tool programmer, and motion matching programmer occur based on the needs of the game. (*DADIU,* u.d.A)

Lead Programmer*:* Takes part as a lead for the team. Alternative title: Tech Lead

Ensures production of designs, prototypes, and pipelines, with the team to define the game’s technical framework.

Responsible for the definition and distribution of tasks to the programmers to fulfill the game design and implementation needs.

In collaboration with the Project Manager, responsible for the planning and assessment of the work scope. (*DADIU,* u.d.A)

Project Manager*:* My role. Possesses executive powers and responsibilities, but is not directly part of decision-making regarding content and aesthetic choices, does however affect it by being responsible for the production's ability to stay on schedule and thereby enforcing production priorities within deadlines.

Part of setting up the testing of the game on the target audience.

Ensure an office setting that reinforces creativity, information, and collaboration within the team. (*DADIU,* u.d.A*)*

QA & UR Manager*:* Is part of the lead team. Ensure a plan for user testing, and reporting findings to the team.

Is part of a larger QA & UR team with the QA from the other teams, and the play-testers they have to recruit.

Responsible for researching the user experience with the game, as a whole, and regarding specific elements to gauge results of product usage and anticipated usage, and reviewing the behaviour and responses against the intended experience. (*DADIU,* u.d.A)

Visual Designer: Responsible for creating the 2D graphical assets and animation assets for the game, under the supervision of the Art Director and Project Manager.

With the programmers: ensures that assets function properly within the game.

In collaboration with the Art Director and Project Manager: responsible for planning and estimating the scope of the work according to time and available resources. (*DADIU,* u.d.A)

The size of this team was already a much larger group, compared to what I had previous experience working with, but it was also technically too few. As shown in figure 3, we ended up not filling up every role in the team roster. With no one assigned to the role of a Visual Designer, the responsibilities of that role were put onto the Art Director, which was a natural fit when it came to the responsibilities, but also a role much more suited for the person in question. The Art Director was a foreign student with a not-always-optimal grasp of the English language, which made him less likely to speak up, but also lead to misunderstandings or time-consuming communications with the three CG Artists who worked in his branch, as well as the Game Director, and myself to some extent. This presumably would not have happened with other members, despite some of them also being foreign. Due to the presence of non-danish nationalities in the team, it was decided to keep the office language to English so everyone could participate, whatever the reason. This was not an issue, as the people on the team were all either students of an international education taught in English or avid video game players meaning the spoken and written English level was generally high.

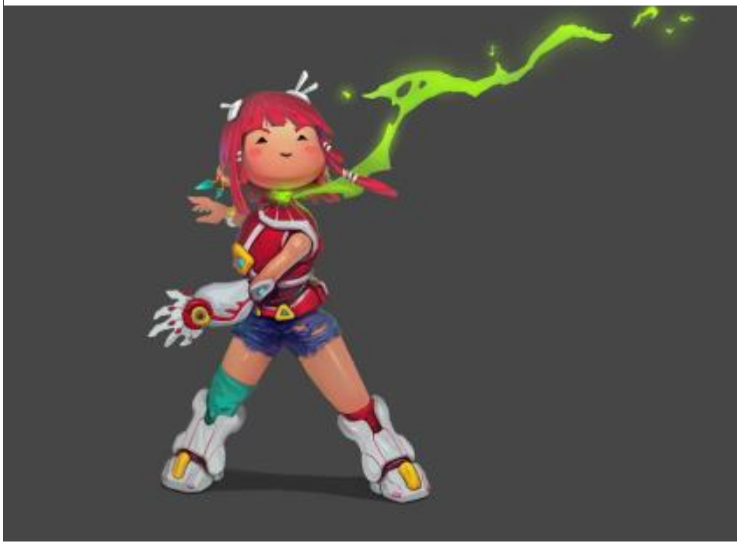
After the completion of the first mini-game, the programmers had to decide amongst themselves who they trusted to be their Lead Programmer, essentially taking the responsibility of managing the tasks of the programmers and being the lead responsible for the information flow between his and other departments, as well as continually updating the pipeline for implementation of tech, assets, tools, etc. As a lead he also participated in the meetings held by the team's leadership, meaning those with director or manager in their role title.

The Game & Level Designers were a two-man team. Per the DADIU guidelines, they were part of the lead team due to their essential role in the game's realization. Practically they operated on the same level as both the lead programmer and the regular programmers, being both responsible for the planning and designing of their tasks as well as the implementation of their ideas (through the tools requested to and provided by the programmers). Being only two it wasn't seen as necessary to designate any of them as a lead game designer, and instead kept the two working in tandem most of the time, while usually inviting one at a time for the lead meetings, for information sharing purposes. This could have caused problems in regards to knowledge-sharing and feelings in regards to delegating trust and authority; in this regard, however, the two were well aligned during the production.

2.1.1b The game and its vision

The graduation game created by the team was a platforming game, made for Android phones, in which the player character, a girl named Zoé (Figure 4), breaks into a gigantic skyscraper standing as an oppressive monolith in the city she lives in. The game's world is a futuristic dystopia, and Zoé seeks the top of the building in the hopes of seeing what lies beyond the closed borders of the city.

Figure 4 Main concept art of Zoé



To achieve this, the player must navigate traps, engineering, and avoid the authorities through levels that go higher and higher into the tower. When Zoé is on the ground she is always moving forward. Using swiping motions for the touch controls, the player must jump, slide, and climb to navigate and avoid the obstacles on the way to the top, with parkour-inspired movements. The game was designed in a way that encourages a state of flow leading to a stylish completion of game levels. (Bækgaard, 2020)

For the creation of this game concept, the game director made use of design pillars meant to be the main essence of the game's concept and execution. The three pillars were **Impact**, **Freedom of Running**, and **Versatile Running**.

Impact meant that actions taken within the game should have a noticeable effect. Whether this is a

graphically noteworthy effect or something that changed the game world, there needed to be a proper weight to what happened on the screen.
Freedom of Running concerned the player's ability to move freely and choose their own way through the game, as well as emulating the feeling you get when you are moving as fast as you can without any burdens.
Versatile Running was an ambition to reach a level of versatility with the basic movement, in the same vein as Mario's jump is used in varied and versatile ways. (Bækgaard, 2020)

These "pillars" can be seen as the guiding principles for the vision the director wanted to achieve with the game. Through this, all decisions made in regards to the game were made so that they could uphold, or at least not contradict and take away from, the vision. Thereby ensuring that the creative forces for the game were unified and vision-driven.

2.1.2 The tutoring/competence lessons

As part of fulfilling the educational part of the programme, the different roles got times where they met professionals within the specific competence assigned to the roles.

The lessons catered towards my role were conducted by people who worked or had worked, as project managers. The lessons themselves were very informal, conducted in a masterclass format, where much of them were at first discussions between the tutor and the project managers of the six DADIU teams, that would evolve into small tasks, with the general theme constituting the lessons being about the role of a project manager, and how to best perform it. It was an opportunity for us to get clarity on our questions and doubts, as well as share our experiences with our teams with each other, and use them as case examples to discuss how to handle them, with the help of an experienced mind to guide us if we erred. Topics we broached included discussing the differences, as well as pros and cons, of waterfall project management versus an agile approach, and an exercise we tried, was to write down all the project phases and responsibilities we held and then ordering them in a way, that made the most chronological sense to us. A problem brought up by most of the teams was people who tended to show up late or outright not showing up at all. So the topic of what reasons the people in question had to do such, and how to best confront a person like that was discussed a good amount. For example, some people simply had responsibilities at home (like children), which made it very difficult to show up on time. The answer suggested to us for this situation was, in short, to use guilt against those with less acceptable motives for tardiness thereby appealing to their sense of morality and their responsibility to the team.

Something slightly surprising, but in line with how the project manager role had otherwise been presented to DADIU, was that no tutors had much emphasis on effective scheduling or time management through specific software or methods, which was explicitly stated by one of the tutors to represent his own work approach, but mostly the right way to accommodate the team in whatever needs were present, and how to set up contingencies with risk assessments. Besides the administrative duties, the focus was very much on how to be a proper leader for the team, with techniques regarding discussion, facilitation, and discipline.

A takeaway from this: the importance of the ability to evaluate a given situation and how to respond to it, is not a simple task given the many variables that can be present within different scenarios and the nature of the problems. This is also why it is very difficult to provide a one-fits-all solution or simple fixes in general to situations that may have occurred because of more than one occluded reason, that cannot be easily amended without some transparency from affected parties.

Being a person that thinks themself to possess common sense (most of the time), the lack of teaching for methods of scheduling and planning was slightly disappointing, but the focus on teamplay and approaches to the more socially oriented problems was no less interesting.

2.2 My role

What was my role on the team?

For this section, I will go into detail about the responsibilities I had, as defined briefly in 2.1.1a, and describe how I went about fulfilling my role and supporting my team. Additionally, there were also specific cases where the choices I made with executive responsibility had a larger potential impact on the team, which I will discuss.

2.2.1 Functions

As part of my management role, I had several jobs to keep track of at all times concerning the team. These included facilitating the team’s communication, providing them with an overview of the production status, enabling and ensuring inter-team meetings, scheduling the production to best fit the deadlines set by DADIU and the team’s capability, and procuring physical assets or locations from the university for the production.

Facilitation

Producing games with different dedicated areas of expertise, between 14 individuals, mean that a lot of different things are happening at the same time. And a team of this size means that you cannot have an in-depth conversation with all team members every day, thus the team should to some point be selforganized, and central project management is a necessity. All the things that happen are meant to come together into one great unity, but productions being what they are, means everything rarely lines up perfectly on time. Meanwhile, setbacks also happen and these can affect other parts of the production, such as the project's version control software experiencing severe downtime, or faulty implementations causing unwanted results within the game.

To make sure that the team was as knowledgeable about the status of each other, we implemented the Scrum method of daily stand-up**s** as part of the morning ritual (*What Is a Daily Scrum?*, n.d.), where the entire team at 09:15 would get off their chairs, and every single person would make a short report of the status of their current project. This way they were able to relay this information to the other side of the office (something the programmers gained a lot from) and inform what their next intended steps were as well as any they needed help from. Before production began, we had been introduced to the concept of making "SWAT teams"; meant to use individuals from different roles and put them in smaller teams to focus on the production of specific tasks. This method; however, was never intentionally made full use of within this team, as the daily stand-ups were great at making people prepare each other for the day's tasks, on top of the members in the team just being good at just *walking* over to other team members when they needed them for whatever reason (Bækgaard, 2020), an issue that seemed silly to think about, considering the relatively small size of the production area, but as I learned from the other project managers, it was an issue some of the other DADIU teams had. Part of ensuring this information flow between team departments included the seating arrangement. The arrangement largely corresponded with the way the organigram shows (figure 3), putting people next to the ones they were most frequently likely to communicate with, and seating me and the game director in the middle of the area, so we were easy to access for everyone, and equally, we could easily oversee the action around us.

Meetings

For the sake of the production overview, meetings were a regular occurrence. The primary type was the lead meetings in which all designated "leads" took part, to detail progress and steps, as well as more overarching needs that were not mentioned by individuals during the daily standups. Irregular meetings between individuals, about specific tasks, were also agreed upon at times, and while not personally present, I also kept a notice as to whether they had happened, and reminded participants if they were missing their appointments.

Scheduling

During the first mini-game production, making the production schedule was incredibly simple, as it was basically provided to us by DADIU, with the essential tasks for starting the production itself, and then the stages of the game's production also plotted on the calendar. The mini-game productions only lasted seven and a half days (deliverable hand-ins were at noon), not counting days allotted for pre-production, so the production phases were very abbreviated. For the second mini-game, the situation was similar, but we didn't have to create the pipeline from the bottom, as we had stolen some time for ourselves in the previous week, where DADIU had nothing scheduled for us, to sort the next mini-game's pre-production. During that week I had project management tutoring for half the time, so I provided a simple day-to-day plan of what events and tasks to expect during the time where I was also gone. A method I used again during the second mini-game production, dividing the production into three main parts: alpha (feature complete), beta (content complete), and post (retrospectives and presentations). This day-to-day schedule was posted in the office and had space allocated for potential changes, although few were made due to the preparation in the previous week, and the compact timeline. The same method did not work for me, for the graduation game; due to its much larger size and the complexity of the tasks involved, which necessitated a much larger time allotment. The method was retooled into a general weekly overview of the phases involved and the expected milestones at the given weeks. This was presented as a guiding suggestion for the production, with an invitation for feedback so it could be adjusted if any department saw an unreasonable deadline.

For personal timekeeping, I made a Gantt chart with Google Sheets to track the progress, dependencies, ETA, delays, and completion of tasks (Bækgaard, 2020). Dividing the chart into the project phases and categorizing the tasks by their department responsibility (design, CG, etc.) or complex tasks with multiple steps, and assigning the name of the person responsible for each task. The tasks of the programmers were numerous, of varying size, and only piled on as production went further along, and as the handling of these was generally more closely monitored by their lead, I noted their tasks in much more general umbrella terms for what they were doing. Tasks that were dependent on the completion of previous steps were illustrated as such.

Arranging and documenting Post-Mortems

As part of every post-production, I sat the team down and asked them to bring up the negatives and positives of the production, asking them to be reflective of the methods, actions, and results we acquired throughout the process. This way we had a log of the greater do's-and-don't's we had experienced and wanted to bring with us to the next production, and as lessons learned from the final production. This post-mortem was also a mandatory responsibility that was used for presentations

Obtaining production assets

As the project manager, I had producer responsibilities, and also being a resident student of Aalborg

University, I was the point of contact with the faculty and handled material requests for the sake of the production. Examples include printing materials, regular and poster-sized, acquiring locales within the faculty for recording purposes, borrowing physical materials, and props for recording purposes, and a television monitor for presentations held in the office.

Rewarding the team

To keep the morale, I proposed and organized rewards for the team in the way of food. With the programmer's propensity for drinking energy drinks, the pant on the cans funded cakes for the team on two occasions. At the end of the last two productions, I polled and organized restaurant visits with the entire team, to celebrate a successful production.

2.2.2 Actions

In this section, I will go into detail about specific cases and the actions I took for the sake of the team, to ensure its progress towards a successful production.

Setting up visual aids and communication servers

To facilitate an overview structure of the production for the team, a few options were put into use. DADIU made use of the website Podio to communicate and handle the obligatory deliverables. No one liked that website and I was easily the one who spent the most time there. Already being familiar with an alternative, I suggested and established a Trello group to make use of Kanban planning for the different departments. There was enthusiasm for this at first, but the use of this died fast, mainly due to its insubstantial presence in the office, and people not wanting to visit a website each time they did something. The QA manager; however, made great use of this, both internally and externally, as the QAs across the six teams had their own communication and task management boards which they used to help each other with bug tracking and updating. The same board was also used internally on the office television as a live bug tracker for the programmers to be wary of. The same method was later in the graduation game, used diligently by the programmers with a physical Kanban board.

For internal communications that weren't strictly work-related and to cut down on across-the-office talking, a Slack server for the team was established. This way the rest of the team also had an easy way to access or create digital records of conversations and plans. Slack was also used to make a channel for the requests the team had regarding production materials as described earlier.

Conflict resolution

Unlike what I heard of the other teams, mine did not have members slowing production due to unannounced lateness or truancy. Other issues did however regrettably occur. The most notable of these having the same source: the Lead Programmer. The first issue was brought to my attention by the programmers collectively wanting to speak with me. I brought them to a nearby room, closed off from the office. They told me that they were unhappy with the way their lead programmer was doing his job. Without going into further detail about his specific wrongdoings, I listened to their grievances and promised to talk to him about them. Before the time I had planned that I shared the information with the Game Director, who then offered to handle the situation. While I slightly regret not specifically being the one to fix the issue, the director did manage to smooth things over between the programmers and made the lead refocus his efforts on what he was meant to.

The second issue came as a consequence of the Lead Programmer being very opinionated around certain topics. As an example, he would spoil conversations that were had in jest about topics that had some political charge to them, by arguing against the topic, eventually leading to snarky responses of "I don't care" or "of course you don't". To prevent this bad mood in the workspace, during a team meeting I politely and firmly requested that political talks and topics be kept out of the office, as they sparked tensions (that the Lead Programmer was arguably bringing onto himself). I didn't, and still don't, particularly like imposing bans on speech that isn't necessarily meant to be disruptive, but when the alternative was further souring the mood during work time (and potentially off-hours relationships) I thought it was the wisest decision to make. Later in the production, the Lead Programmer moved seating arrangements next to me, for logistics purposes, so I became sort of an outlet for the topics that weren't strictly disallowed, but he knew weren't appropriate given the ban already imposed because of him, which was much to the amusement of the programmer that the lead most butted heads with.

Cutting content

To the team's utmost credit, they were all very good at reaching conclusions regarding what content they could feasibly produce within the remaining time, given the schedule and deadlines provided to them. This did not always hold of course. Near the end of the graduation game's production, it became a race for the inclusion of assets and designs not yet implemented in the game. The Game Director of course fought, as he should, for the inclusion of things he wanted to fulfill his vision for the game. At this point, we were well near the end of the production which meant that the pressure on the technical department was the highest. Knowing this, and that the programmers needed as much time as they could get for the final stretch, I would argue in favour of the technical department, whenever scoping assessments were made during meetings. It also helped that the technical department was already regularly dealing with implementation issues, but also had the most realistic overview of the game's status.

3. Hard and Soft Management

Work Question: *What is hard and soft project management?*

Here comes the matter of exploring the meaning of the aspects behind project management, hard and soft, that I want to investigate and discuss, more so than simple examples given during the introduction. To manage this, I will make use of different items of literature (see below), that all relate to the process of project management and the creation of software. The literature is selected by how they each highlight and explain elements that I think consist of *hard* and/or *soft* project management. To this effort, I will also point to my experience at DADIU and the examples of management methods used or spoken of in chapter 2, to find where they fit within the hard and soft ideal types, and see what points of relation can be established between them (See Introduction) with the help of the selected literature, as well as find where the methods might impact vision-driven productions.

With this, the goal of explaining "what", will come through the use of examples the understanding that I have arrived at, of why the *hard* and *soft* aspects are present, how they are fundamentally different, but also what areas that they can intersect.

This chapter will address the two following sub-work questions in parallel.

Sub-Work Question 1: What characterizes "hard" project management?

Sub-Work Question 2: What characterizes "soft" project management?

The reason these questions are addressed in parallel, rather than separate like the work questions in the previous chapter, is due to readability and the chosen method of topic exploration, ergo both questions will be explored through literature examples, that contain methods and methodology that speaks for (or against) the usage of both hard or soft project management. I will also further reiterate the explanation of the aforementioned from the introduction.

Ideas and examples will be drawn forth from the chosen literature of *"Prince2 - Successful Project Management", "Software project management: a process-driven approach"*. The themes of these books approach project management, from a very goal-oriented perspective of the purpose behind project management, as well as covering methods within this perspective to achieve the goal, with the first being a very top-down-oriented methodology and the other a comprehensive handbook. The other literature in question is *"Dynamics of software development"* and *"Agile Project Management with Scrum"*. These books, of course also have the intention for the project manager to find the best way to reach the goal and providing methods to do so, but the background for the methods acts on a very different philosophy towards project management than the previous pair of books. The main difference between the philosophies is that the first methodology is phase-oriented and the second is activity-oriented. In the phase-oriented philosophy, a project can be planned from the beginning to the end before it starts where project content and project time are coupled in a plan with dependencies. In the activity-oriented philosophy, a project cannot be planned from the beginning to the end before it starts; thus, project content and project time are not coupled in a plan with dependencies, but tasks are listed and prioritized. To this end, the activity-oriented methodology is more abstract than the phase-oriented, because project-content and project-time are coupled into empirical phases during the activity-oriented process.

The application of this topic is specifically intended within vision-driven media creation, and the chosen literature in this chapter is admittedly not specifically aimed towards this. However, they are all aimed at or can be used within, software production and the project manager's role being a normative one, they all contain relevant instructions. Simultaneously, due to the vision-driven topic, I will be referring to examples from my time at DADIU and specific points from chapter 2 throughout the process, as part of the overarching discussion of the points I draw forth, throughout the chapter.

3.1 Characterization of hard and soft project management

During the introduction, I made a brief description of how I define the two management aspects. The distinction I present sets hard management as focused on using specific processes and methods, that are used to deal with problems that can be measured and accounted for reliably. I would add to this, that hard project management is the employment of measured top-down organized approaches, meaning to implement the decisions that are intended to ensure the accomplishment of the project's targeted goals.

Soft project management I very shortly described as dealing with the social and interpersonal communication issues that are probable within team-based product development. To elaborate further, it is taking a more bottom-up approach towards the management of the team. Focusing more on ensuring the needs of the team and providing a sense of ownership of the project to the team, to motivate and increase the team's innovation capacity thereby assuring a higher quality end-product, working under the assumption that the team possesses the larger innovation potential than the leadership.

The paradox of vision-driven digital media creation is the top-down need of the director's strictly unifying vision of the product while wanting the potential of innovation surrounding the creation of the product to be as high as possible, needing a strong bottom-up foundation, wherein the challenge of the project manager lies for this type of project.

3.2 Project Management literature

This will be the section where the four chosen pieces of literature will be looked through and discussed while referring to events from DADIU. The books have different structures and goals affecting what is ultimately relevant and interesting for this report's exploratory purpose, and because of this, there will be differences in how they are approached in their segments. After all four are reviewed, their findings will be summarized where the examined content is used to answer the sub-questions for this chapter.

3.2.1 "PRINCE2 – Successful Project Management"

PRINCE2 – Successful Project Management (2010) is a manual for project managers, aiming to be a generalized source of knowledge to teach the ins and outs, of all the things that a project is comprised of and how the project manager best maneuvers their related responsibilities, and providing a structure for these both for inside the team and the communication that goes outside it (stakeholders, steering committee). By its admission, it will not teach leadership skills. Conceding it is very important for project management, it says it is impossible to formalize it into a specific method, due to the variances in the approach used by successful leaders. Instead, it refers the reader to use other leadership models and courses to fulfill this need (PRINCE2, 2010). The name, PRINCE, is an abbreviation for PRojects IN Controlled Environments.

The book's chapters are divided into the explanation of the PRINCE2 principles, themes, and processes. The principles being the pillars a project must uphold to be considered a PRINCE2 project. I do not claim that my DADIU semester games could be considered PRINCE2 projects, but PRINCE2's applicability still makes it possible to draw parallels from some of the principles, which I will go through, and follow up with further general discussion of the method within digital vision-driven media creation.

Continued Business Justification; a project must make sense from a business standpoint, the reason for the project's existence must be sustainable throughout the project's lifetime i.e. have a clear Return on Investment otherwise it will be terminated, the details and justification for these are called the Business Case. Therefore the validity of the project's business justification should be documented and approved regularly (PRINCE2, 2010, p. 11).

Learn from Experience; Projects are a unique venture, and there are new things to make or find during them. These unknowns present a risk, which urges a team to learn as much as possible from similar projects, as the experience of lessons already learned by others, can speed up the progress and/or help to avoid risks of the project. Therefore team members are urged to, and per PRINCE2 held responsible for, seeking and sharing valuable lessons, rather than waiting for their provision. All experiences achieved from and during the project should be made part of reports and reviews so that further completion can be improved upon than it otherwise would be, likewise, the experiences should be made available for further passing on when the project is completed (PRINCE2, 2010, p. 12).

With the DADIU semester being educational, parallels for this can be drawn towards the actual lessons all members had for their discipline and the experts that were available for asking questions. As well as the experiences we took with us from the post-mortems after each production.

Defined Roles and Responsibilities; The right people are needed for the proper completion of a project, and if participants don't know who is responsible for what or what is expecting of them, then no planning or management will save the project. This pillar of PRINCE2 also emphasizes that a project has 3 primary stakeholders (business sponsors, users, and suppliers), that the project's organization must engage with, and must be represented in the project board by people taking the roles of Executive (essentially responsible for the business case and Continued Business Justification), Senior User (ensures the project is progressing with the user's needs in mind), and Senior Supplier (ensures the project has access to the necessary people and resources and represents the interests of those supplying them) (PRINCE2, 2010, p. 12).

This pillar is where PRINCE2 makes it clear it is intended for projects in proper business organizations, as it is impossible to unify this pillar with a student program like DADIU without overwhelming changes to its management. Roles and responsibilities are of course part of DADIU and vision-driven media creation, as laid out in the 2.1.1a Role Descriptions section (and any other project for that matter), but the role defining is limited to the people in the management hierarchy and the project board, which cannot be applied to DADIU due to the administrative set-up of the program, and the "staff" limitations of the student teams, not being set up to support PRINCE2. The project board responsibilities would have to lie on the administration from the DADIU staff, with the project manager being the only intended link of communication to the team (Figure 5), which would not work in practice due to the Director's role and close relationship with the DADIU teachers at the Film School.

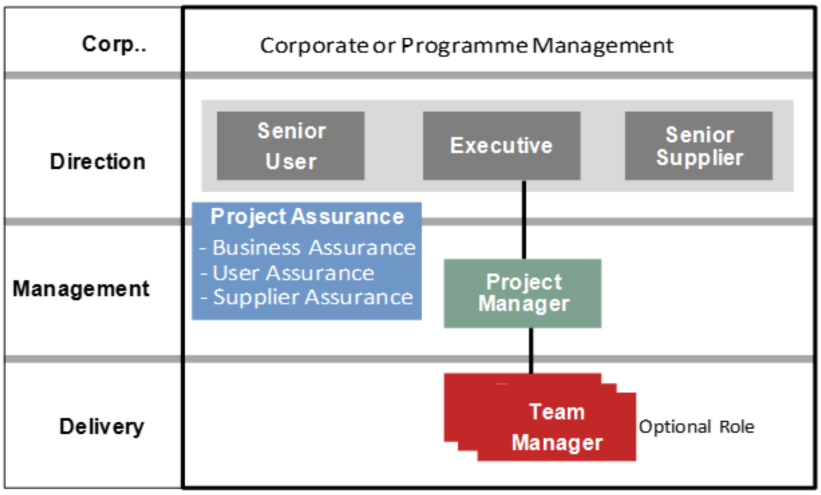


Figure 5 The Project Board. PRINCE2 wiki https://prince2.wiki/img/project-board.png Note: Project Manager and Team Manager can be the same person, hence the "optional role".

Manage by Stages; A PRINCE2 project is planned, monitored, and controlled in phases. Splitting a project into phases makes it possible for management to review the phases for the business case, and adapting the project's progress to the business need, risks, and complexities that may occur. Planning can only work on a detail level that can be handled and predicted, ergo a detailed plan over several months will quickly become unprecise. PRINCE2 uses stages as they allow the project to be divided into more manageable segments, and uses a high-level/low-detail plan for the overall project and high-detail plans for the individual stages. For a PRINCE2 project, there are at least two management stages: The Initiation Stage and a Management Stage (PRINCE2, 2010, p. 13).

This is an approach that resembles what I ended up doing myself when the Graduation game started up. A note to make here; in the case of vision-driven media creation, such as games, unless you have allocated time within the phases for how the Director's vision might change (which is planning for a deliberate change of plans that may or may not happen), and even then; the proposed changes would theoretically be part of a report for the project board to approve or deny, taking agency away from the Director and his power to influence the product's aesthetic qualities, and thereby denying ultimate ownership of the product's vision.

Manage by Exception; This is the term used to describe an agreed-upon level of tolerance for the project factors (time, cost, scope, quality, risk, benefit) before the project manager has to consult with the project board. It's a problem management method to qualify the scale of a problem and whether it is to be dealt with by the project manager or if the problem has become big enough to become an Exception so the board has to intervene (PRINCE2, 2010, p. 13).

All the project factors are things to keep an eye on in vision-driven media production, but since we didn't make use of a project board, the responsibility of managing what was a problem versus an Exception (to use the PRINCE2 terms), and how to fix it was a discussion between me and those the problem was related to. If the issue was at an Exception level it would likely be a matter of what could be cut while retaining most of the Director's vision, due to the strict time scale of the productions.

Focus on Products; The product should be clearly defined so that stakeholders don't have differing ideas about what is being made, and to prevent meetings, delays, new requirements, and what the expected level of quality is, emphasizing quality rather than delivery. (PRINCE2, 2010, p. 14).

For the vision-driven production made at DADIU, this was handled from two sides. DADIU set some basic requirements for the games themselves i.e. The game must run on an Android phone, the game must have menu texts, and narrative texts, be toggleable between Danish and English. Meanwhile, the matter of what we were making, was of special importance to both DADIU *and the team*. Here it was the Director's job to make sure that his vision for the games was properly communicated, both for the sake of team coherence, but also for the educational part of "selling" the product that was our game.

Tailor to Suit the Project Environment; The method should be adapted to what fits the project's need. This means that themes and processes can be altered to fit the project, depending on its scale, available people, and complexity of the project. Following PRINCE2 to the letter for a small project might needlessly complicate the level of complexity needed from an administration level, and will hamper the project's efficacy (PRINCE2, 2010, p. 14).

While this is a neat virtue to claim for a management methodology, I would wager that there are simply types of projects that cannot practically fully "fit" into the intended PRINCE2 scheme, as was discussed out under *Defined Roles and Responsibilities,* due to the size of the project's organization. The number of management layers put forth in the Project Board figure indicates a certain size, and while it does specify what factors to consider when tailoring the project (PRINCE2, 2010, p. 14), all of these hierarchical bureaucracy methods make very little practical sense, if you are an independent team of three people creating a game together to use an extreme example.

PRINCE2's pillars and priorities are sensible in their applicability and reinforce a focus on knowledge sharing, quality assurance, and ensuring a project's commercial viability. The method is very much agnostic towards elements that don't pertain to the organizational aspects of managing a project, and barely touches upon the activities that happen on the delivery level (figure X), and beyond what their responsibilities and competencies are (PRINCE2, 2010, p. 279), gives little in the way of instruction for how the team leader ought to perform their role on the floor, which PRINCE2 does admit to (PRINCE2, 2010, p. 7). While it does state that the project and team manager can be the same person, the initial distinction between the two sets the precedent for the project manager to be the person the team manager talks to when he can’t or isn’t allowed to address an issue (Manage by Exception) which the project manager then, has to report to the project board for them to make a decision.

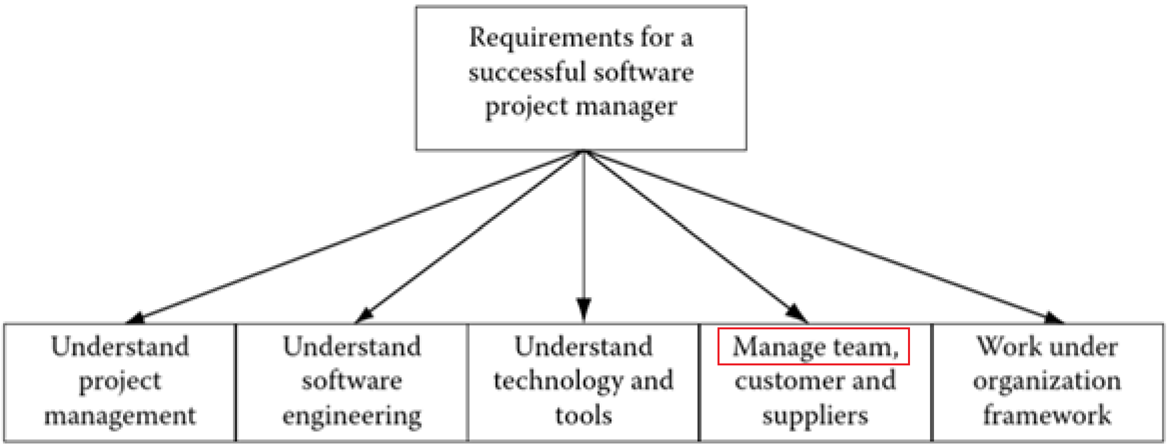
3.2.2 “Software project management: a process-driven approach”

This book (Ahmed, 2011) is written with the express focus on how to manage software-creation projects and details many parts, large and small, that comprise software projects and how to best manage them. As such it is written in larger parts covering several broad areas, with the large categories specifically covering: Project management, software life-cycle management, process improvement and selection, people management, and technology management, as the essentials for software project management. The volume of knowledge being shared and discussed for these topics has been aggregated from various sources and is disseminated within the book at the level the author deems necessary for software project management for a specific reason. In the preface for the book, the author mentions that he specifically wrote the book, so it “would cover the entire syllabus of software project management.” (Ahmed, 2011, p. xix) as the books he found available at the time never covered more than half the syllabus required for students.

Much of the book’s structure lies in the discussion of small topics within the larger topic, and many concepts within these topics are conveyed with figures to explain them; such as how to use an ATM to explain what a process is (Ahmed, 2011, p. 10), or what the predictability for processes within manufacturing versus within projects (Ahmed, 2011, p. 102). Another apt example is provided in figure 6 below showing all the important skills that are required of a software project manager, which the book covers, and of which “Manage team” in the fourth box is of specific interest to this report’s topic, and while the other boxes all contain relevant topics, only the highlighted topic will be examined. Due to its purpose as an educational resource, it also has chapters that start by outlining topics and ends with relevant exercises and review questions.

It is not a strict methodology book. As it aims to give a comprehensive knowledge foundation. The differences between standard waterfall and iterative project models, and how this affects processes, are also regularly discussed within chapters.

Figure 6 "Requirements to be a successful software project manager".


Requirements for a successful software project manager
Understand project management
Understand software engineering
Understand technology and tools
Manage team, customer and suppliers
Work under organization framework

Note: Adopted from Ahmed, 2011, p. 12. With own added highlight

Team Management

Regarding Team Management, Ahmed (2011) makes it the point of ensuring the team's performance. What can the project manager do, to get better performance from the team? The techniques for achieving this include performance-linked awards, mentoring, and skills training. These options are listed as viable answers to some of the common problems within team management which respectively are identified as attrition, unavailability of IT talents filling specialist and integrator roles, and lack of training (Ahmed, 2011, p. 265). Another important facet of any kind of people management, whether it be internal or external, is communication and how to handle that correctly given the project type.

As team members are the project manager's main resource for delivering the goods, they should identify the members who are high and low performers. This can be difficult to assess in team-based projects and assignments; to this end, proper tools and techniques to evaluate individuals must be adopted, both so they can be rewarded for a job well done, on top of identifying the ones who need counseling on how to improve their performance

On the matter of attrition and productivity, Ahmed (2011) puts the two factors into the context of project environments of high and low process-orientation, environments which can be translated into places making use of top-down and hard management or bottom-up and soft management strategies respectively. With high process-orientation, productivity is high and predictable, but tasks are repetitive and solutions are prescribed, so because of the stifling of creativity, tasks can become monotonous, and therefore team members lose motivation for work, become less productive, and attrition rises. In low process-orientation environments, the people are freer to apply their creativity to their work and are thus more motivated, but the lack of structure and process flows into less predictable output, which translates into being less favorable from a business perspective (Ahmed, 2011, p. 270). Project managers should therefore be able to identify signs of attrition and provide a good work environment that gives employees a worthwhile reason to stay, so they remain satisfied beyond the salary they earn which should otherwise sustain their basic human needs.

In DADIU all the processes included in the realization of the vision required specific skill-sets, and due to the relatively small project size, the tasks involved were never the same for long, and in most cases required creative problem solving from the ones involved, whether they were working it alone or with someone else, so strict protocol and method enforcement could stifle their motivation, and hurt their productivity as you are taking away a creative individual's ability to apply their talent and take ownership of their work. This does to some extent apply even to those working in more theoretical and logically founded disciplines, like the programmers. While their rationales are not primarily aesthetically driven, they are still individuals wanting to apply themselves, so the need for creative freedom towards problem-solving still applies. This needs to be balanced against the reality that is deadlines and tight schedules, requiring some level of oversight processes, so a task doesn't end in a progress dead-end or suffer from feature creep. At DADIU the resolution to this was typically through bargaining of what could acceptably be cut from the production to maintain the schedule and director's vision.

With regards to interpersonal conflicts on the team, they are unsurprisingly considered bad for the project, and the project manager should be able to see the troubling signs early to proactively take steps to resolve conflicts (Ahmed, 2011, p. 278). In essence: identify issues and have the social skills and know-how to apply the conflict escalator to appease and satisfy the conflicting parties.

Two other factors important in team management are managing knowledge and communication. When projects are developed within an organization, knowledge is aggregated by the members working there, and over time this knowledgebase resides more and more within those members. If a particularly knowledgeable team member leaves an organization, then their knowledge is lost leaving future (or current!) projects without that benefit. An example of this could be how to talk to a specific customer to leave them satisfied, or a programmer of a very specific algorithm that drives a vital part of a software. Unless a proper repository for this knowledge is established and properly updated (aka managed) so that all vital information about projects is accessible to future endeavors and team newcomers.

Management of communication encompasses the structuring and communication of specifications and requirements from customers, ensuring they can be found, and agreeing upon meanings and definitions of terminology within the project team to a coherent project language, so misunderstandings are kept to a minimum.

Having a DADIU curriculum requirement list, rather than customers, and a team with different disciplines led to the requirement of the above. The requirements for the projects' at DADIU would change with each game production, so making sure the team knew to fulfill them was essential, while the director had to make his game vision fit with them in mind, and direct accordingly. Beyond the communication of requirements and asking for elaboration on them, the internal communication didn't so much lack on the general vernacular as the team, as mentioned before in the 2.1.1 My Team sub-chapter, all had some background interest in video games, and spoke English. Only the much more specialized terms within disciplines required elaboration for those that had to work with it, but conflicting double-meanings were not an issue with the disciplines present as they possessed unique terms.

As a resource for project managers, this book provides a thorough and wide-reaching source of knowledge that covers all the bases for managing software projects. This wide-reach does however mean that there are some specific sub-categorical subjects, that are contained within very brief sections, meaning they aren't elaborated on, which is both good and bad since brevity is welcome in a book this dense, but the smaller detail level can lead to imbalances in related topics which results in readers having the kind of questions or interpretation that would usually be cleared up with discussions, which arguably lends itself well to the book's educational format, but if it is used outside the lecture hall or classroom then significant value is lost.

3.2.3 "Dynamics of Software Development"

This book can at the simplest, be summed up as a list of thoroughly elaborated collection of rules, all based on the experiences and dynamics (hence the title) observed during software development projects, that the author (McCarthy, J. 1995) has put onto paper, to serve as helpful ideas and rules of thumb from start to finish of production and shipping of software products. As pointed out in the foreword by Denis Gilbert, McCarthy's partner, the book is about developing great software, and great software teams. It is also not a book that takes an academic approach to its subject matter, but from "The School of Experience, the School of Hard Knocks, the School of We-Never-Wanna-Lose-Another-Software-Review-Again" (McCarthy, J. 1995. p. xi). The book is structured into the four parts of Opening Moves, The Middle Game, Ship Mode including The Launch correlating to things the manager needs to be wary of from the start, middle, and ends of a software production cycle, and what meaning lies within these times of software development. Additionally, it includes an appendix discussing the importance and hows of hiring and keeping good people. These parts have important topic chapters, these are where the rules come into play as subchapters, which is where the value of this book lies. The book was recommended to the leads in the DADIU teams, to read before starting the semester. For this purpose, I will present and discuss the rules that in hindsight resonated with me for my time at DADIU, and that I consider most relevant in answering the research questions, thus again excluding the rules more oriented towards the business and customer side of development. While they are important, they are not considered significant to the research question.

Very fittingly this section can start with rule **#1 Establish a shared vision.** (McCarthy, J. 1995. p. 13). Conceding that it seems obvious and absurd to have to put forth this idea, he also says that establishing the shared vision can be one of the most difficult feats. There is a slight clash of terminology here, as the operative word “vision” is not only about the aesthetic rationale as so far referred to with “Vision-Driven” in this report. The unification of vision for what the product is supposed to become is still part of this, but more than that it is about the unification of the team’s vision for the entirety of the production itself and achieving a sense of purpose in the team, and the start of achieving this lies in a leader’s “empathetic perception of the psychological state of his or her team”. The crux of this is that the ability to establish an empathetic bond between team members and the team and its leader(s) will result in unity, which in turn will support the leader's ability to support the vision. The Director and Project Manager, in DADIU and vision-driven media creation, both sit on a large part of the responsibility for creating this vision. While speaking in strict terms, the Director has to lock the vision for the product’s purpose and expression, and the Project Manager the product strategy and team resources, they both must stand as leaders for the team and establish that pivotal vision, to ensure success.

#2 Get their heads into the game. (McCarthy, J. 1995. p. 22) means acknowledging that everyone on the team has thoughts and ideas, and if the team consists of intelligent and resourceful individuals (which it of course does) then chances are these ideas are beneficial to the team, especially in a creative environment. It is all about encouraging thought and discussion, added with how sharing and having ideas acted upon is affirming. If team members won’t discuss their thoughts openly with each other it could be a sign of them believing they simply aren’t supposed to, and that creative behaviour is frowned upon. (McCarthy, J. 1995. p. 24)

Closely related to that is rule **#4 Don’t flip the bozo bit.** (McCarthy, J. 1995. p. 30). Making software requires intellects to work together to produce it, and produce it properly. And to get intellects to work they need to think, and to the question “What’s the hardest thing about software development?” McCarthy answered: “Getting people to think.” (McCarthy, J. 1995. p. 30), even though many people will claim to want to, they default to the easier option of “flipping the bozo bit”: turning themselves into someone who at best won’t screw things up for everyone else, deciding they don’t know what they’re doing and in a powerless position. This is of course unacceptable for a functional team, as all having a dead-weight does is drag the team. Flipping the bit is also done between people, and likewise unwanted, and stems from communication problems leading to this flip. McCarthy provides two more examples; which can be summarized as the unwillingness to listen to critique due to a bruised ego flipping the bit on the critic, and the person who keeps giving unsolicited ideas and feedback being branded a bozo. As is also pointed out, if a leader flips the bozo bit on someone, the ones under the leader's influence will do the same. It is therefore important to examine your part of the communication that threatens your want to brand someone a bozo, no matter which end of it you are. (McCarthy, J. 1995. p. 32). Mixing an early and later point on this, relating it to my own DADIU experience, declaring yourself a bozo as a project lead, as I came close to doing (Bækgaard, M. A. 2020), is beyond dangerous for the team, even if they are highly self-sufficient as mine was, there were still many a managerial task that they naturally didn't have eyes on. At the same time, due to the sometimes adversarial nature of the partnership between the Director and Project Manager in the fight between features against remaining time, this rule becomes important to maintain, as the two leaders must steer the team together, not separately, for the visions to remain intact.

#21 Minimize Dependencies. (McCarthy, J. 1995. p. 83): Defined as "any necessary thing, not under the team's control." The fewer dependencies at the start of the project the better off it will be, and while one task might be done within the scheduled time, another might not. The addition of new dependencies during the production should be scrutinized to the utmost, and the reliability of the providers of this potential dependency should be researched. Due to the mixing of disciplines, aka different "teams", within a vision-driven production such as a game, many dependencies exist, as a matter of fact, just based on all the different types of assets that need to mesh together in the right ways for the vision to be fulfilled.

#28 Remember the triangle: Features, Resources, Time. (McCarthy, J. 1995. p. 96): Those three elements are what you have as a development manager, so when the schedule slips you normally only have four solutions in general terms: add time, subtract features, add resources, or do some combination of the three (McCarthy, J. 1995. p. 97). For DADIU our allotted time was, as mentioned, set in stone, ergo the adjustments of the triangle was always a matter of discussing the features wanted (and cutting when needed) and allocating the right people for the job, or finding a resource that could improve performance, to accommodate for the limited time resource. You could however find more time, by resorting to working outside the agreed-upon hours, but then the work goes into "crunch" territory which was disallowed by DADIU and is the usual cause of burnout within the industry when chasing the deadline, not that they had any real power to stop those insisting on doing it by their own volition, as was the case in my team (Bækgaard, M. A. 2020).

#34 Use ZD milestones (McCarthy, J. 1995. p. 115): A software product is a virtual construct and is primarily, only real in the minds of the team working on building it. To force it into "reality" it must reach a state where it can be presented in a usable or representative manner, to help the team gain insight into their actual progress, and where there are issues. ZD stands for zero defect and is a way of determining a quality level for the product in question, for the specific milestones. It is not meant to signify the product has no bugs or missing functionality and is more defined by deliverables for the milestone, meaning work does not continue to the next milestone until all work for the current one is done (McCarthy, J. 1995. p. 115-116). In the case of DADIU, it would neatly align with the way we defined the pre-production-alpha-beta-release milestones, where each stage was a hard limit for what needs to be done before the next stage could properly begin. The point McCarthy makes about the product not being quite real is very salient and with the unavoidable bugs present in software and game development, compiling the team effort into a "whole picture" as it were, really puts the state of the product into perspective against the director's, and team's, vision. For Q&A purposes we did however shift to a model of daily builds for our games, which also became a helpful way of marking our daily progress.

The rules all provide insight into the experiences had by the author, some of which are supported by anecdotes of the projects he has worked on, and can therefore serve as heuristics for the reader to follow. Many of the heuristics do have some overlap with each other in practice, but that is due to them having follow-up rules that expand on, or relate to, the preceding rule, and some of them (either within themselves or separate ones) might even seem contradictory, but that all belies the experience-based nature of these rules to show that they are not meant to be exclusive from each other. As an example, the ZD Milestone is the start of a section of heuristics building upon the purpose and proper usage of milestones in software development. Rather, they are naturally imperfect, as they are coloured by an imperfect perspective and are meant to be malleable and applicable within the situations they are intended.

3.2.4 "Agile Project Management with Scrum"

This is the book about Scrum written by one of the original sources and creators of the method Ken Schwaber (2004), which is another book describing a method and the included skills to possess, to manage a team to properly deliver software products. It goes through the what and how of the Scrum method as a whole, and does it by explaining the methodology and tools used within it, and by providing several anecdotal stories from many different companies and projects, that give some insights into the application of Scrum. For the review of this book, it will be a look into what defines the methodology of Scrum, and what philosophy lies behind it. As mentioned back in section 2.2.1 my team at DADIU did make use of some

Scrum methods, so this will serve as further exploration into how we deviated, and whether more strict usage would have been a benefit or a detriment.

The roles involved in a Scrum project, as far as the methodology is concerned, number only up to three: The Product Owner, the Team, and the Scrum-Master. The Scrum-master is the project manager and the person who provides leadership, guidance and coaching to the team. He or she is to ensure that the project participants follow Scrum processes and practices, and implementing it so that it fits within the organization's culture. The Product Owner is the one who is responsible for the upholding of stakeholder interests and creating project requirements and release plans. The requirements being located within the Product Backlog, which is a prioritization tool for the project to organize and ensure progress on the most important parts of the product at a given time (Schwaber, K. 2004. p. 6-7). The Team is the collection of developers and experts who develop the functionality of the product. All management responsibilities within the project are divided amongst these three roles, as the Team members themselves are also self-managing, organizing, and works across the different team functions, as it is their responsibility to take the agenda set by the product backlog and build towards the goal set within the iteration. Scrum works within incremental iterations, wherein each iteration lies the work towards an “increment of functionality” that could potentially be shippable for the product.

The so-called rules that a Scrum-master must help facilitate for a project to be a Scrum project are the Sprint Planning Meeting, Daily Scrum Meeting, Sprint, Sprint Review Meeting, Sprint Retrospective Meeting (Schwaber, K. 2004. p. 133-139). Something of note is that while many of the rules have specifics, it is specifically mentioned that changes to the rules are an option, but they should come from the Team, not management (being the Scrum-master/Product Owner or management above the project team), and these changes should only be considered if the Scrum-master thinks those involved understand the processes of Scrum enough to know the impact of the changes so the Scrum mindset is not lost (Schwaber, K. 2004. p. 133).

Sprint Planning Meeting

As it is written in Schwaber (2004) the Sprint Planning Meeting, is time-boxed to be two 4 hour segments, one for selecting items off the Product Backlog, as prepared by the Product Owner beforehand, the other is for the Team to discuss how they will plan to perform the incremental development of the selected items, and how far they think they are capable of doing, based on the requirements, their available technology, and the assessment of their collective aptitude. This second part must result in the Sprint Backlog, which must at the very least, give a complete enough picture of what the team has mutually agreed to commit to through the sprint (Schwaber, K. 2004. p. 134). For the more informal DADIU course, we did not make use of this method, and I could not see it done with my team specifically, especially with all the unknowns within the projects, and the fidgety go-getter energy of the students, an entire day of planning sounds like a good way to demotivate participants when considering the relatively short time frame of the semester-long programme, compared to other game development project which can take several years, where an entire day of planning can have great value.

Daily Scrum Meeting

15 minutes at the start of the day, all Team members (no matter how many) attend a meeting at the same place every day, where they are asked "What have you done since the last meeting with the project?", "What will you do for the project until the next meeting?", and "What impedes you from being as effective as possible?". Only the person reporting on these questions should be talking. If anything reported pertains to another Team member, meetings between interested parties can be arranged for immediately after the Daily Scrum (Schwaber, K. 2004. p. 135). As mentioned back in 2.2.1 Functions, I made use of this method with my DADIU team. To avoid repetition, refer to 2.2.1 Functions, in the second paragraph under the first section named Facilitation for personal experience under DADIU with the method.

Sprint

The sprint is the specific number of days set aside for the project team, to work undisturbed by outside influences, but they can reach out for help. The standard set by Schwaber (2004) is 30 calendar days. This is the time the Team has to build on the items selected from the Product Backlog and bring them to a state where they could be shippable. If there are items on the Backlog that the Team thinks they can't or can do more of during the Sprint, they can consult the Product Owner on how to prioritize additions/removals. If the value or viability of the sprint itself is lost, due to lack of resources, or failing to meet targets, then the Scrum-master can terminate the sprint early and start procedures for a new Sprint plan meeting. During this time the only administrative duties the Team has, are attending the Daily Scrum meeting and making sure the Sprint Backlog is updated (Schwaber, K. 2004. p. 136-137). Having tried many times in different teams and project types to make use of a continuously updated backlog, I can attest that it is something that requires a certain discipline to uphold, and one that failed for most members of my DADIU team. Which is a shame because it is a great tool to provide progress transparency to not only other team members but to you, the project manager, and the director, during the workday moving beyond what is being reported at a Daily Scrum meeting.

Sprint Review Meeting

The purpose of the review meeting is to present functionality that reached a “Done” stage to the Product Owner and stakeholders, if something is not done in the agreed-upon sense, as in ready to be implemented or shipped, then it is not part of the presentation. Stakeholders can give feedback and identify new wishes for functionality that can be requested for the Product Backlog (Schwaber, K. 2004. p. 137-138). This is a meeting wherein the results of a work period, i.e. sprint, is shown, which in a vision-driven context also makes for a good tool to mark the progress towards a Director’s vision if they are in a position or organization-structure that does not accommodate a more frequent way of updating their knowledge of how each different branch of development progresses.

Sprint Retrospective Meeting

This is an internal meeting, where only the Team and Scrum-master are obligated to attend. During the meeting members of the team are to answer the questions of “What went well during the last Sprint?” and “What could be improved in the next Sprint?”. The Scrum-master’s function here is mostly as a note-taker and facilitator, as the Team should know itself best, and find ways to better make Scrum work for them. The solutions or suggestions agreed upon by the team should be implemented into the next sprint because if a Retrospective adds nothing to the team, it only adds frustration (Schwaber, K. 2004. p. 138-139). The same method of asking team members “what went right/wrong?” was part of our Post-Mortem procedure at DADIU (2.2.1 Functions). It helps greatly at letting the team air their thoughts as to how the production has proceeded and puts a focus onto where they, or you as the project manager, can try to improve, or add, things for the better.

Much of the concept of Scrum lies in the belief in the transparency of the project, both to those within the project team and those outside it. At the same time, it also stresses the importance of letting the Team work on their job as agreed upon, to eliminate meddling or piling more and more work onto the developers, by other branches of the organization. This protected autonomy also stresses the importance of them being responsible for themselves, but it is borne out of the knowledge and experience that software productions are inherently very very complex creations to undertake, and the addition of people doesn’t make this any easier (Schwaber, K. 2004. p. 5), so to make the management effort simpler for the project manager, much of the management responsibility is spread out among more people. This method of self-management also feeds into the philosophy that the team as a whole carries more potential for innovation capacity, than what can be found in their leadership (cf. section 3.1). Something that is not mentioned at all for this methodology within Schwaber (2004), is knowledge retention. It goes unsaid, but I can only assume that holding onto knowledge about projects to learn from, is completely foisted upon the Team to handle, since you as a Scrum-master holds no responsibility to do so. Admittedly the focus of the book is solely on the virtues and applications of Scrum, but when it tries to control or limit complexity, it doesn't offer any hint at what can be provided by would-be-known projects beyond "the team can get help", or incentivizing documentation of an ongoing project for future reference. Within a vision-driven production in general, the Scrum-master, Product Owner, and Team *could* fit onto the Project Manager, Director, and… well the team of developers. Within this framework, the Director becomes more responsible for what can be possibly produced for a Sprint, but for the emergence of his vision, this control might not necessarily be a detriment, as he and the Team will still have to discuss specifics, and the Team gets to pick, while the Director gets to prioritize what is important for him (and potential stakeholders). All the while, the Project Manager can focus energy on being a leader and a facilitator to support the Team, and keep them within the Scrum guidelines. However, from a regular project management perspective, the role and responsibilities of the product owner would also be within their purview of skills, and therefore the two roles would be held by two different project managers, which is nothing crazy albeit not supporting the usual team setup at DADIU, leaving the vital Director role outside of the designated "upper management" where he has to be to be positioned to align the vision and goals of the different disciplines. Connected to this; Scrum as a method was not made with multidisciplinary teams in mind, like what is required of digital media creations, and therefore has the potential shortcomings in regards to the way the role applications should work on top of the leadership duo of Project Manager and Director.

From the chosen literature, further examples of what characterizes the dichotomy of hard and soft project management have been described. These have been expanded upon by discussing them within a vision-driven project perspective, usually within the confines of the DADIU programme. The results of this will be used in the next chapter to build the answer to the work question contained within.

4. Applying hard and soft project management

Work Question: *When to apply hard and soft project management?*

To gather the primary lessons from the presented sources and experience, and provide answers to the sub-work question "*When to apply hard and soft management?",* the key elements that I have found to make up these two ideal types, from the literature in chapter 3 are summarized in Table 2 further below. This table will present these key elements in paired dichotomies, under the categories of Hard Management and Soft Management, to show how the facets of hard and soft management differ from each other on topics that mirror each other in purpose or focus. Explanations of what dimension these key elements represent, and why they fall within hard or soft is provided before the table itself. The table itself can thereby be read like a table of heuristics that can provide an answer as to what sort of management focus might be applied given the situation represented in the specific dimension.

As it has been laid out by the contents of the previous chapters, creating software is a complex affair no matter the team and project size, and considering the vision-driven digital projects makes use of the same field of developers (as well as others that are just as important), it would follow that within vision-driven digital project management, the usage of Soft Management is a condition to its execution, due to the nature of the aesthetic driver and working with subjects that are complex to start with, and adds development branches that increase the complexity of the project itself. That however does not in any way exclude usage of hard management, because when you as a project manager need to balance the factors of Time, Resources, and Features, there needs to be a way to align the project to reach the goals within those factors. Especially when a deadline is approaching. Routine tasks and specified processes, whether they be on a management or operational level, are also still part of vision-driven projects.

Management orientation: Motivation-management – Vision-management

A priority for management laid upon them, in regards to the team and its members. Within hard management, the focus lies in ensuring the team follows the decided upon processes under the assumption that this will yield the wanted results. Due to this focus on processes within hard management, it becomes important to provide motivational support to team members, to prevent attrition and turnover rates. For soft it is assumed that personal investment afforded by creative freedom will prevent attrition on its own. Instead, it becomes a focus to make sure that the team stays on target, regarding the project's vision, ensuring a shared vision amongst the team members, so their efforts can be better united, as a project's vision is frequently calibrated, across potential different team branches working on the same project.

Management style: Supervisor – Supporter

A secondary role of the project manager within the type of management. The supervisor being there to make sure people do their work and know what to do, being the link to upper management. In contrast to a supporter role, where the project manager affords themselves a more personal investment in their team members, by facilitating their personal development on top of their management duties.

Project structure: Waterfall – Agile

The structure of the project. With waterfall being the linear progression path built upon dependencies and set processes. And agile being iterative through sprints where each sprint will have a new focus depending on new inclusions or priorities. Hard management wants the project to follow a linear approach, and soft wants to allow for iterations, ergo the barriers set for the management of time and content within the project, have different restrictions set upon them e.g. hard management wanting time and content to be set upon project start, and soft allowing for restructuring of the time and content within iterations.

Process philosophy: Phase-oriented – Activity-oriented

The type of project work is allotted within specific prescribed phases of project time within the hard management ideal, but soft management is more based on prioritizing activities and tasks adjusted to the needs at the given time. It is also this philosophy that provides the base on which milestones are organized around.

Organizational approach: Top-down – Bottom-up

The way managerial decision power is prioritized for the project team, with hard management putting it at the top-down level with an upper management that's not part of the project team, and bottom-down giving much more power to the team itself. The reasoning for top-down management lies in ensuring the understanding and follow-up on decisions made on a high level so workers know to move in the same direction, so to speak. Conversely, bottom-up giving power to the team also allows them to provide much more perspective and knowledge during times of decision making, making use of a larger knowledge base rather than what is available within management.

I**nherent capacity: Production capacity– Innovation capacity**

For hard management, the prioritization of the team's output lies in their capability to produce to keep within allotted time and resources, as it works within contexts of high predictability and routine work. Whereas soft lies in encouraging the team's innovative capabilities by allowing creativity to be a driving force, in creating solutions to complex situations and problems where uncertainty is high. It is the focus of the team's problem-solving capability in the face of risk.

Meeting frequency: Few meetings – Many meetings

Describing the purpose of meetings and their structure. In hard management, the meetings come when phases are over and the project status must be reviewed with big decisions following, whereas soft management uses many smaller meetings to encourage information flow within the project team. This follows within the same philosophy that lies beneath top-down and bottom-up management approaches respectively.

Skill focus: Knowledge management – Competence management

Within the highly process-oriented hard management work environment comes a focus on logging knowledge obtained for, during, or because of a project. This way it can be used for future projects and refine processes, or be used as a foundation to build upon. This focus is not inherent within soft management. Instead, it lies in managing the people in the team and facilitating the growth of their skills and competencies, to further encourage their self-reliance.

Table 2 "Key elements of the Hard and Soft Management Ideal Types".

Dimension	Hard Management	Soft Management
Management Orientation	Motivation-management Process-oriented	Vision-management Vision-oriented
Management Style	Supervisor	Supporter
Project Structure	Waterfall (dependencies) Project time, and content, are firmly defined at the starting point.	Agile (sprints) Project time, and content, are defined throughout the project.
Process philosophy	Phase-oriented	Activity-oriented
Organizational approach	Top-down Hierarchical organization structure (Bureaucracy)	Bottom-up Team organized structure (Organic)
Inherent capacity	Production capacity	Innovation capacity
Meeting frequency	Few meetings	Many meetings
Skill focus	Knowledge Management (Learn from experience)	Competence management (Personal and skill development)

Conclusion

The area of interest for this project has been Project Management in a Digital Vision Driven Media Creation context, and with that, I established the Research Question: With an empirical point of departure in my vision-driven digital project management experience at DADIU; when to apply hard and soft project management in vision-driven digital projects?

Through three larger work questions, which were further explored by dividing them into sub-work questions, I examined different areas of project management. I have laid out the background experience and knowledge I possessed going into this research. Using that as my base, I explored select examples of literature on project management within software development, providing insight into the methods used by project managers, and putting these into the perspective of my experience at DADIU to explore their place in a vision-driven context. Finally, I took this knowledge and to answer my Research Question, created a table for which the key elements of hard and soft management were paired up in dichotomies, belonging to specific dimensions of project management that can be used as a heuristic guide, on what sort of management is most relevant for a given situation within the dimension.

Further research

With the findings within this paper's research, and establishing of the Hard and Soft ideal types within project management, a foundation for further research into the codifying of the requirements of a project management methodology specifically aimed towards Vision-Driven media creation has been made. The focus of this would be towards the inclusion of the Aesthetic vision as part of the formal project management process, giving the director role and tasks, a part that is not explicitly coded into project management methodologies, like the ones explored in this paper.

Bibliography

Ahmed, A. (2011). *Software Project Management: A Process-Driven Approach* (1st ed.). Auerbach Publications. https://ebookcentral.proquest.com/lib/aalborguniv-ebooks/reader.action?docID=826941

Bækgaard, M. A. (2020). *DADIU Projektrapport*. Aalborg Universitet.

DADIU. (2021). Retrieved February 01, 2020, from https://www.dadiu.dk/

DADIU(u.d.). Retrieved February 01, 2021, from http://www.dadiu.dk/#newsinfo

DADIU(u.d.A). Retrieved April 26, 2021, from https://www.dadiu.dk/game-competences

Manifesto for Agile Software Development. (2002). Retrieved December 02, 2020, from https://agilemanifesto.org/

Mathiassen, L. (2017). Designing Engaged Scholarship: From Real-World Problems to Research Publications. *Engaged Management ReView*, *1*(1), 1. https://doi.org/10.28953/2375-8643.1000

Max Weber (Stanford Encyclopedia of Philosophy). (2017, November 27). Standford Encyclopedia of Philosophy. https://plato.stanford.edu/entries/weber/#IdeTyp

McCarthy, J. (1995). *Dynamics of Software Development*. Microsoft Press. https://archive.org/details/dynamicsofsoftwa00mcca/mode/2up

PRINCE2 - Projektledelse med succes. (2010). London: TSO, The Stationery Office.

Rosenstand, C. A. F., & Laursen, P. K. (2013). Managing Functional Power: In Vision Driven

Digital Media Creation. *Akademisk Kvarter* , *6* , 81-94.

Schwaber K. (2004) *Agile Project Management with Scrum.* Microsoft Press. https://archive.org/details/agileprojectmana0000schw/mode/2up

The Semester. (2019). Retrieved January 13, 2020, from http://www.dadiu.dk/semester.

What is a Daily Scrum? (n.d.). Scrum.Org. Retrieved August 17, 2021, from https://www.scrum.org/resources/what-is-a-daily-scrum