



**UNIVERSITY  
OF TURKU**

Turku School of  
Economics

# **Managing multidisciplinary video game development projects and multidisciplinary project teams**

Information Systems Science

Master's thesis

Author:

Lauri Salminen

Supervisor:

MSc Eva Collanus

8.5.2025

Turku

The originality of this thesis has been checked in accordance with the University of Turku quality assurance system using the Turnitin Originality Check service.

Master's thesis

**Subject:** Information Systems Science

**Author:** Lauri Salminen

**Title:** Managing multidisciplinary video game development projects and multidisciplinary project teams

**Supervisor:** MSc Eva Collanus

**Number of pages:** 66 pages + appendices 3 pages

**Date:** 8.5.2025

The game industry and video game projects have grown rapidly both in Finland and globally. Project sizes and complexity have also grown at the same time requiring even more multidisciplinary skills to create games. There has also been little research that focuses on the project management and video game producer perspective on managing those multidisciplinary projects. This research tries to examine how video game producers manage multidisciplinary game development projects and the multidisciplinary project teams developing them by interviewing video game producers.

Prior research on the subject had looked at individual parts of game development and project management like the development process, multidisciplinary development teams, and development methodologies in game development. Research on multidisciplinary teams and knowledge management has also been researched widely while there is little research related to game development from those domains. The theoretical framework for this research was formed by trying to synthesize this earlier research into a framework describing multidisciplinary video game project management and team management.

This research used a qualitative multiple-case study method where five video game producers were interviewed by going through questions related to key themes based on the theoretical framework in a semi-constructive manner. Interview data was recorded, transcribed, and then analysed by using the Gioia method, and conclusions were made by comparing analysed results with prior research.

Results generally support earlier research into multidisciplinary video game project teams and projects. According to the results, stakeholder management was identified as an important part of video game project management not discussed by prior research. Dependency management was also identified as an important part of planning multidisciplinary project tasks alongside scope management of the project. Multidisciplinary teams were seen as an asset by the producers providing valuable insights. Knowledge management on the other hand could help with ensuring commitment to the game's vision and minimizing the amount of tacit knowledge in project teams. The development process was planned to suit various disciplines working on the project by using Agile, Waterfall, or a hybrid methodology. According to the results formal risk management might also be more common than claimed by prior research.

**Key words:** video game project management, multidisciplinary teams, multidisciplinary projects, multidisciplinary project teams, video game producer, managing game development

Pro gradu -tutkielma

**Oppiaine:** Tietojärjestelmätiede

**Tekijä:** Lauri Salminen

**Otsikko:** Monialaisten videopeliprojektien ja monialaisten projektitiimien hallinta

**Ohjaaja:** FM Eva Collanus

**Sivumäärä:** 66 sivua + liitteet 3 sivua

**Päivämäärä:** 8.5.2025

Peliala ja videopeliprojektit ovat kasvaneet sekä Suomessa että globaalisti. Samalla ovat kasvaneet myös projektien koko ja monimutkaisuus vaatien monialaisia taitoja videopelien luomiseksi. Projektinhallinnallista ja pelituottajan näkökulmaan keskittyvää tutkimusta monialaisten projektien hallinnasta on tehty hyvin vähän. Tämän tutkimuksen tavoitteena on selvittää haastattelemalla pelituottajia, miten pelituottajat hallinnoivat monialaisia pelinkehitysohjelmia ja projektien monialaisia kehitystiimejä.

Aiempi tutkimus aiheesta on tarkastellut yksittäisiä pelikehityksen osia ja projektinhallintaa kuten kehitysprosessia, monialaisia kehitystiimejä ja kehitysmetologioita pelikehityksessä. Monialaisia tiimejä ja tiedon hallintaa on myös tutkittu laajasti, vaikka pelialan osalta aihetta käsittelevää tutkimusta ei juuri ole. Tämän tutkimuksen teoreettinen viitekehys on muodostettu yhdistelemällä tätä aiempaa teoriaa, niin että teoreettinen viitekehys kuvaa monialaisten pelialan projektien ja projektitiimien hallintaa.

Tämä tutkimus käytti menetelmänään laadullista monitapaustutkimusta, jossa haastateltiin viittä pelituottajaa useista eri organisaatioista puolistrukturoiduilla haastatteluilla, käyden läpi viitekehysten pohjalta hahmoteltuja teemoja. Haastatteluiden data nauhoitettiin, litteroitiin ja analysoitiin käyttämällä Gioia-menetelmää. Johtopäätökset tehtiin vertailemalla saatuja tuloksia aiempaan tutkimukseen.

Tulokset tukivat yleisesti aiempaa tutkimusta koskien monialaisia pelitiimejä ja projekteja. Tulosten perusteella sidosryhmien hallinta havaittiin tärkeänä osana peliprojektien hallintaa, jota aikaisempi tutkimus ei ollut tunnistanut. Riippuvuuksien hallinta havaittiin myös tärkeänä osana monialaisten työtehtävien suunnittelua projektien laajuuden hallinnan rinnalla. Tuottajan pitivät monialaisia tiimejä hyödyllisinä ja eri osaajien näkemyksiä arvokkaina. Tiedon hallinta taas auttoi projektin visioon sitouttamisessa ja hiljaisen tiedon minimoimisessa projektitiimeissä. Selvisi myös, että kehitysprosessit oli suunniteltu sopivaksi eri alojen näkökulmille käyttäen ketteriä menetelmiä, vesiputousmallia tai hybridimenetelmiä. Tulosten mukaan formaalit riskinhallintamenetelmät saattavat myös olla yleisimpiä, kuin mitä aikaisempi tutkimus väittää.

**Avainsanat:** peliprojektien hallinta, monialaiset tiimit, monialaiset projektit, pelituottaja, pelinkehityksen hallinta

# TABLE OF CONTENTS

<b>1</b>	<b>Introduction</b>	<b>8</b>
	<b>1.1 Background</b>	<b>8</b>
	<b>1.2 Research questions</b>	<b>9</b>
<b>2</b>	<b>Video game project management</b>	<b>10</b>
	<b>2.1 Video game development process</b>	<b>10</b>
	2.1.1 Pre-production phase	12
	2.1.2 Production phase	13
	2.1.3 Post-production phase	14
	<b>2.2 Creating a game experience</b>	<b>15</b>
	<b>2.3 Challenges in video game project management</b>	<b>15</b>
	<b>2.4 Risk management in game development</b>	<b>17</b>
	<b>2.5 Managing creativity in video game projects</b>	<b>18</b>
	<b>2.6 Development methodologies in game development</b>	<b>19</b>
<b>3</b>	<b>Multidisciplinary project management</b>	<b>22</b>
	<b>3.1 Multidisciplinary teams</b>	<b>22</b>
	<b>3.2 Knowledge processes in multidisciplinary project teams</b>	<b>22</b>
	<b>3.3 Multidisciplinary team management practices</b>	<b>23</b>
	<b>3.4 Multidisciplinary projects in small firms</b>	<b>24</b>
	<b>3.5 Multidisciplinary teams in game development</b>	<b>24</b>
<b>4</b>	<b>Theoretical framework for multidisciplinary video game project management</b>	<b>27</b>
	<b>4.1 Multidisciplinary video game project management theory</b>	<b>27</b>
	<b>4.2 Theoretical framework</b>	<b>28</b>
<b>5</b>	<b>Methodology</b>	<b>30</b>
	<b>5.1 Selection of methodology</b>	<b>30</b>
	<b>5.2 Data collection and empirical data</b>	<b>31</b>
	<b>5.3 Data analysis</b>	<b>32</b>
	<b>5.4 Research ethics and evaluation</b>	<b>33</b>
<b>6</b>	<b>Results</b>	<b>35</b>

<b>6.1 Producer role</b>	<b>35</b>
6.1.1 Producer's responsibilities	35
6.1.2 Other type of producer roles	37
<b>6.2 Managing game development</b>	<b>38</b>
6.2.1 Project phases and milestones	38
6.2.2 Development methodologies and software development practices	40
6.2.3 Testing	41
<b>6.3 Scope and dependency management</b>	<b>42</b>
<b>6.4 Managing multidisciplinary game teams</b>	<b>43</b>
6.4.1 Multidisciplinary team composition	45
6.4.2 Building project teams and team formation	46
<b>6.5 Managing creativity</b>	<b>47</b>
6.5.1 Game concept development	48
<b>6.6 Knowledge management and communication</b>	<b>51</b>
6.6.1 Communication in project teams	52
6.6.2 Sharing information in meetings and sessions	53
6.6.3 Documentation	53
6.6.4 External networks	54
<b>6.7 Risk management</b>	<b>55</b>
<b>6.8 Managing stakeholders</b>	<b>57</b>
<b>7 Conclusions</b>	<b>58</b>
7.1 Managing multidisciplinary project teams	59
7.2 Managing game development process and creative process	60
7.3 Managing game projects	62
7.4 Limitations and suggestions for future research	63
7.5 Contributions to practice	63
<b>References</b>	<b>64</b>
<b>Appendices</b>	<b>67</b>
<b>Appendix 1. Data structure</b>	<b>67</b>
<b>Appendix 2. Semi-structured interview questions (in English)</b>	<b>68</b>

## **LIST OF FIGURES**

Figure 1 Game development process phases	11
Figure 2 Framework for multidisciplinary video game project management	29
Figure 3 Multidisciplinary roles of the project team	45
Figure 4 Updated theoretical framework	58

## **LIST OF TABLES**

Table 1 Video game project management anti-patterns	16
Table 2 Producer responsibilities	36
Table 3 Multidisciplinary project teams conclusions	59
Table 4 Development process conclusions	61
Table 5 Project management conclusions	62

# 1 Introduction

## 1.1 Background

The game industry has been growing rapidly both globally and in Finland. A report by PwC about the global entertainment and media outlook released in 2023 estimated that the global size of the gaming industry was already 213 billion dollars in global revenue in the year 2022. It also forecasted that the total size of gaming industry will surpass 300 billion dollars in global revenue by the year 2027. (PwC, 2023.)

The Finnish game industry while still being small on the global scale has also grown from almost nothing to 3,2 billion euros in turnover by the year 2022 according to Finnish game industry umbrella organization Neogames Finland. This is remarkable considering that ten years ago the figure was only 320 million dollars. The industry has multiplied tenfold in size, so it has also become more important to the whole Finnish economy. (The Game Industry Of Finland REPORT 2022.)

However, making video games is not easy but it requires a lot of resources and skill. The projects have huge budgets and for example, the recent Finnish hit game made by Remedy has already been said to be one of the most expensive cultural products ever made in Finland with an estimated budget of 50 million euros according to Helsingin Sanomat (Niemi, 2023). Due to high budgets especially smaller gaming companies rely on few or only one project so managing those projects well is crucial for the companies to succeed.

When it comes to managing those projects the project manager, that is in most cases a video game producer or a development director, has to deal with various challenges like managing multidisciplinary teams and complex development process. Video game teams also have a high developer turnover due to the cyclical nature of the industry. Game producers also must deal with business problems like marketing video games to make sure that the project succeeds financially. Challenges in video game project management are in the end mostly people-related rather than technical challenges. (Politowski et al. 2021.)

Prior research has focused on the challenges of video game development as a whole (Politowski et al. 2021) and from a software engineering perspective (Kanode & Haddad, 2009). The video game production process (Aleem et al. 2016a, Aleem et al, 2016b) and methodology have also been researched previously (Godoy & Barbosa, 2010; Petrilla & Pimenta, 2010). Managing creativity in game production has also been researched but to a lesser extent (Zackariasson & Wilson 2006).

Project management or producer perspective is also something that has been less researched (Marklund et al 2019).

Managing video game projects from a people perspective that considers technical and creative processes of game development while keeping in mind the multidisciplinary nature of the projects and development teams has been researched by a very limited scope (Musil et al. 2010). A wide review examining game development research by Engström et al. (2018) also states that there is a need for research that tries to synthesize results from heterogeneous research related to game development. A lot of the prior research on this field is also relatively old considering how much the industry has grown in the last decade. The purpose of this research is find out how video game project management professionals manage video game projects and multidisciplinary teams.

## **1.2 Research questions**

This research focuses on the production perspective of video game development and aims to find out how video game producers or other video game development project managers can handle both the complex process and multidisciplinary teams. The focus is more people focused rather than technical but things like development processes and methodologies are also very central when it comes to managing people so they should be considered.

RQ1: How do video game producers manage multidisciplinary teams?

RQ2: How video game producers manage multidisciplinary game development projects?

This research aims to answer these research questions by interviewing video game project management professionals (video game producers, development managers, and development directors) from various video game companies. This should give an in-depth perspective about the current practices in the industry as all video game development even in small teams requires multidisciplinary skills and a complex development process that has to adapt to changes during development to achieve desired gameplay experience.

## **2 Video game project management**

### **2.1 Video game development process**

There has been little research that focuses on video game project management holistically. Video game development process and video game projects on the other hand have been researched before even though a big part of research can be already considered somewhat dated given that the industry has grown and matured during the last decade. There is also almost no research focused on the project manager's role in game development even though a big part of prior research on things like game project challenges or game development process is very relevant from a project management perspective.

Marklund et al. (2019) note that it is widely agreed that video game development has clear production processes that can be examined. Things like documentation, project planning and project management are well defined in game projects. However, development methods are applied differently compared to traditional software development. According to the researchers this is mainly caused because of the focus on player experience that must be formed during the process rather than software functionality.

Video game development process also been well defined by research and while there may be some variance in practices or development methodologies Aleem et al. have identified in their two articles (Aleem et al. 2016a; Aleem et al. 2016b) some common and well-defined phases in game development. The first article focuses more directly on the software engineering process life cycle and in it, Aleem et al. (2016b) have conducted a systematic literature review to map out the game development process from a more software development-oriented perspective. The article identifies three key phases: pre-production phase, production phase, and post-production phase. Other studies like Marklund et al. (2019) have also identified that those three phases came up in other studies on the subject. They also identified an ideation phase but that can be seen as part of the pre-production phase as both studies by Aleem et al. (2016a; 2016b) have done.

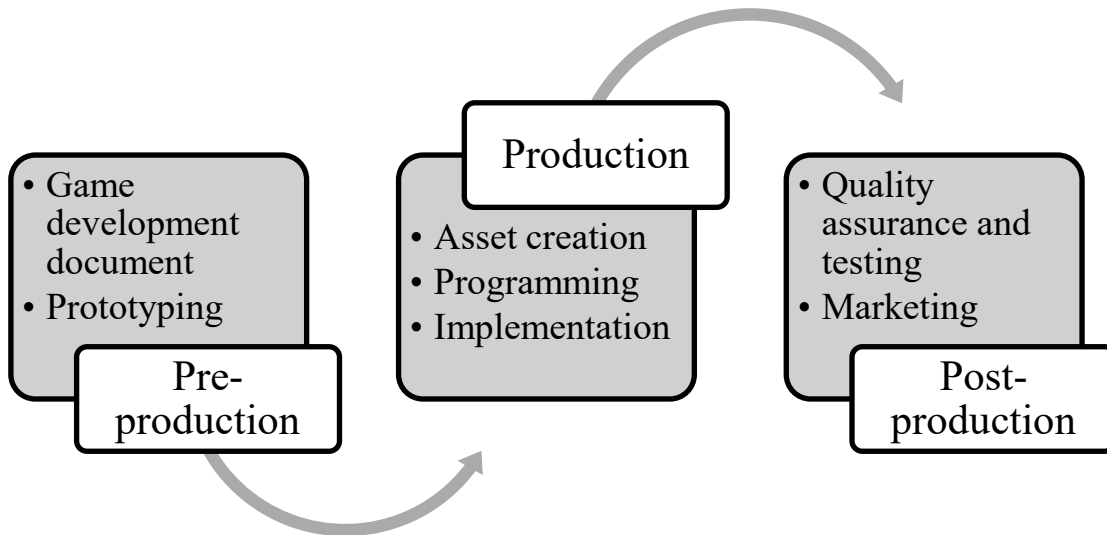


Figure 1 Game development process phases

Pre-production phase focuses on defining the requirements of the project and creating a game design document (GDD) and a prototype based on those requirements. Both are seen as an important deliverables of the pre-production phase. Other kinds of planning like storyboarding and marketing strategy are also part of pre-production. Prototype of the game is meant to test the fundamental mechanic of the game to test out if the idea can create the desired game experience. (Aleem et al. 2016b.)

Production phase contains creating all the assets, levels, scenarios, and programming for the game and implementing them together. Game engine is an important software framework that can be created to support the game, or a ready one can be used to make the development faster. Post-production includes quality assurance and game testing in the form of internal testing or beta testing. Marketing is also a big part of post-production. (Aleem et al. 2016b.)

The other article by Aleem et al. (2016a) focuses on the critical success factors to improve the game development process. According to their research that was conducted on game developers using a questionnaire, some key success factors for the development process were team configuration management, game design document management, game engine development, game test management, and programming practices. Many of these success factors are project management and people-related so they also show that good project management is crucial for a successful game project.

### 2.1.1 Pre-production phase

In the pre-production phase focus is on planning the game project and producing important deliverables like GDD and prototype based on that initial planning. The key purpose of the GDD is to be a coherent description of the different components that form the complete game. Prototype on the other hand is used to test those planned components in practice to make sure that the game produces the desired player experience. Requirement specification is also a key part of pre-production which can be both less formal factors like aesthetics or gameplay or more formal technical specifications. The less researched part of the pre-production phase is risk management which means identifying risks in the game development process and domain. This is usually the responsibility of the game's producer. (Aleem et al. 2016b.)

Marklund et al. (2019) note that while a game design document is seen as an important document and is a commonly named form of documentation in the research literature it seems that game development is much more focused on prototyping than documentation. For example, game design documents are susceptible to change even during the production of the game or based on the prototype. While the game design document is an important document that represents the common creative vision, the researchers found that open knowledge sharing, and informal dialogue are more widely common ways of upholding this common vision. The researchers also note that the project management professional needs to facilitate these kinds of collaborative processes in the game production team, but this aspect has been much less researched. (Marklund et al. 2019.)

Video game projects rarely have specific development requirements that can be more commonly found in traditional software development. Murphy-Hill et al. (2014) state based on their empirical research using interviews and questionnaires that game development professionals felt that functional requirements are hard to define because desired gaming experience is more of an artistic achievement rather than some technical requirement that can be fulfilled in a straightforward way.

Marklund et al. (2019) do however note that especially when it comes to larger AAA studios (a term used for studios creating very high-budget video games) there might be a strict set of technical requirements when it comes to things like software performance, compatibility with different systems and usability. Requirements also play a larger role when things are outsourced, or game companies collaborate. These requirements keep the goal clear and mitigate risks associated with these kinds of external stakeholders. It also seems that requirements become more formal in bigger and more mature video game studios. They also noted that more mature processes and complicated hierarchies are not

inherently positive things and might make continuous creativity and flexibility harder to maintain. (Marklund et al. 2019.)

Murphy-Hill et al. (2014) do not fully contradict this idea, but they do also note that this challenge of defining requirements may lead to poor technical architecture and technical debt. This on the other hand can lead to technical challenges later in the game development process. However, code and tool reuse are less common in the games industry compared to traditional software development according to some researchers (Murphy-Hill et al. 2014) it is not necessarily that beneficial to make strict requirements to minimize technical debt if that means less flexibility and could restrict creativity.

The literature on game development seems to agree that functional and technical requirements have a smaller role than in traditional software development and defining requirements can be hard due to the unpredictable nature of game development. However, to avoid technical challenges later in the process they should be probably considered while trying to make sure that this does not decrease continuous creativity and flexibility in the process.

### 2.1.2 Production phase

Production phase refers to the actual implementation of the game idea and concept. This means things like asset creation of the game where game developers and especially the artists create things like audio, 3D models, and visual assets for the game. Storyboarding on the other hand means developing the game scenarios for the game. This is more related to the designer role. Production phase also includes the programming of the game whether that is scripting to specify game mechanics or more fundamental game engine development. Game engines are the most central software framework in video games, and they offer core functionalities like animation, physics engine, and networking that are needed for the game idea to be built. Game companies either create their game engine or use a pre-developed one. (Aleem et al. 2016b.)

When it comes to other code or tool reuse, Murphy-Hill et al. (2014) have discovered that developers believe that it is less common than in other forms of software development. This is due to there not being as strict performance requirements for games compared to other software development. When performance optimization is done, it is very project specific. However, they also note that game engines are the most important reusable component of game projects.

Tschang (2005) identify two key challenges that came up in the production phase when analysing game project reports. Feature creep that refers to the problem that new features keep getting added

during the production. While these new features might even improve the game this can lead to the project scope continuously expanding leading to further challenges.

Pipelining on the other hand refers to the problem where some aspect of the development can prevent other parts from progressing. For example, animations cannot be done before 3D modelling so if processes are not aligned properly this can cause bottlenecks in the development process. (Tschang, 2005.)

### 2.1.3 Post-production phase

Post-production phase refers to the testing, releasing, and marketing of the game project. Testing is often done during the production and there is usually quality assurance done during the project. Beta testing is a common form of testing video game products where a game is released to test users publicly to identify bugs and see if any desired features are missing. There can also be different kinds of testing like heuristic testing where experts evaluate the game quality or usability testing for the finished game. If the game is for example made for educational or other specific purpose it is important to empirically test that it works as intended for that goal. Video game marketing and publishing can be done by the project team or publisher of the game. (Aleem et al. 2016b.)

Murphy-Hill (2014) points out that while quality assurance and testing play a significant role in game development test automation is less common than in traditional software development. This is because you cannot use unit testing or other common forms of automated software testing when testing gameplay and game experience. Another challenge for automated testing is the non-determinism of video games – things like artificial intelligence, randomness, multithreading, and complex states which create bugs that can only be spotted when testing the gameplay. According to the researchers, this is why user testing has a large role in game development.

Kasurinen and Smolander (2014) have researched game testing by interviewing seven game development teams from different companies and summarized video game testing to have more focus on the soft aspects of testing rather than technical testing. This is well aligned with the observations made by Aleem et al. (2016b) and Murphy-Hill (2014). They noted that even in teams that used their game engine the testing was more focused on game mechanics rather than the technical aspects of the game engine. Project teams also seemed to have good resources for testing due to the emphasis on customer feedback in the games industry. Testing feedback also seemed to influence the end product more heavily compared to different kinds of software products. (Kasurinen & Smolander, 2014.)

## 2.2 Creating a game experience

A common theme that comes up in game development research is the challenge of creating an emergent element of fun or experience for the players. Marklund et al. (2019) note that this requirement for a unique game experience means that while game development may have similar phases and methods as traditional software development (testing, design, ideation, development methodologies) this consideration for something that goes beyond functionality and effectiveness of the finalized product requires an approach unique to game development.

They also discuss how game testing and player experience are related. According to them, game testing is often made in all stages of the game project and that testing is used to shape the game and player experience. According to the researchers, the game development team must make sure the game provides the desired player experience and if that is not the case, it is a problem that must be dealt with immediately. Sometimes that can even mean that the game project is cancelled in the prototype phase. That is why prototyping can be used to decide what games should be expanded into full commercial game products. Game testing can also generate completely new ideas that can play a large role in the final product. While these new ideas can be extremely beneficial to the game, they also note that there is always the risk of feature creep. (Marklund et al. 2019.)

This kind of goal towards a unique desired game experience is also very subjective as noted by the researchers. This means that the requirements of the game projects can vary greatly depending on the type of project. According to them defining requirements can even be impossible before actually starting the game production and the requirements are instead discovered while making the game. This is also something that can be seen as a difference compared to traditional software development. (Marklund et al. 2019.)

## 2.3 Challenges in video game project management

Politowski et al. (2021) have focused on the game industry's problems through an extensive analysis of the grey literature of the field this research was post-mortems of already finished game projects from 1997 to 2019 that report the successes and challenges of the projects. Based on their research business-related problems have increased while management-related problems have decreased. This might be related to the adoption of agile development methods according to the researchers. Technical and design problems have also decreased during the timeframe of the study and design problems most drastically during the last decade.

Team problems have also increased but marketing problems have increased the most compared to the other problem types. This could be related to the larger number of smaller studios emerging that have less marketing knowledge than larger game companies or publishers. The researchers conclude by stating that the root causes of problems are related to people rather than technologies. (Politowski et al. 2021.) This also emphasizes how researching video game project management is important to do from a team and people perspective as well as from a business studies viewpoint.

Ullman et al. (2022) on the other hand focus on game development project management challenges from a different perspective looking at project management anti-patterns in the industry (Table 1). Project management anti-patterns are common but ineffective responses to issues created by elements of software projects like people, processes, and resources. According to the researcher's software engineering project management anti-patterns are already well documented and based on 440 game project postmortem problems they also managed to identify 12 common project management anti-patterns for game development of which four were identified to be unique to game development.

The first one was feature creep – features keep often being added to game projects greatly expanding its scope and while the additions might seem worth the extra effort, they end up distracting from the core features of the game. This anti-pattern problem can be solved by making sure that the game's design requirements and definitions are clear from the start of the project. While it is normal for game projects to be iterative and features to change, allowing for too loose management to cause feature creep might make the whole project very hard to control. (Ullman et al. 2022.)

Table 1 Video game project management anti-patterns

Anti-pattern	Cause	Solution
Feature creep	Extra features added without them enhancing core gameplay	Clear requirements and definitions
Bad feature cuts	Lack of resources and poorly defined priorities	Prototyping and playtesting
Multiple projects	Financial constraints	Adding extra staff to projects or managing time availability
Inadequate tools	No standardized tools for development activities	Considering tools early in the project and either choosing appropriate third party or developing in-house tools

The second anti-pattern is also related to the game's features but kind of an opposite problem. Feature cuts may be done due to lack of resources and too loose priorities definitions. While the first anti-pattern demonstrates that cuts might be necessary badly directed feature cuts caused by a lack of resources might cause features that would have improved the game greatly being cut from the game. Researchers suggest prototyping and playtesting to help define priorities to understand what features should be prioritized and when cuts are made all developers should be included in the discussion and documentation of those cuts. (Ullman et al. 2022.)

The third anti-pattern was working on multiple projects that were often caused by financial constraints. Developing or managing multiple projects made individual developers' or manager's workload much higher causing misaligned project goals and feeling overwhelmed by the amount of work. The two solutions for this problem were either carefully managing the time availability of developers and resources available or if the resources allow adding extra staff to the projects. (Ullman et al. 2022.)

Last game development project management anti-pattern was absent or inadequate game development tools. Team members ended up using outdated or improvised tools rather than there being well-defined and standardized tools for the development activities. This anti-pattern can be avoided by considering tools early in the project and looking for suitable tools fit for the project need from third parties or developing suitable custom in-house tools – which however is often more challenging. The researchers also saw this anti-pattern to be the most unique for game development with other types also being applicable for other types of software. (Ullman et al. 2022.)

Anti-patterns that existed in both software development and game development were underestimated tasks, death march meaning a project requiring unsustainable overwork, understaffed team, underbudgeted project, neglected design, quality assurance/testing issues, false surrogate endpoints meaning focusing on things that seem to add value but actually do not, and lack of technical knowledge and planning. (Ullman et al. 2022.)

## **2.4 Risk management in game development**

Risk management is part of project management and is usually done mostly during the pre-production part of game projects (Aleem et al. 2016b). A study by Schmalz et al. (2014) approaches video game development project management from risk management perspective examining how risks are managed in video game projects. The researchers conducted eight interviews with video game

producers to see how risk management practices are used in the industry. They also wanted to see how the producers perceived different risks in game projects.

They found out that the focus on creativity and art-related goals made game development different compared to other kinds of IT projects. The researchers found three new risk factors that were unique to game development: *audience match* which meant the game was successfully catering to its intended audience, *fun factor* which refers to whether the game is entertaining and enjoyable to play and *extent of originality* that describes how the risks associated with creating something complete original are higher than risks of emulating or expanding a already existing game concept. (Schmalz et al. 2014.)

The researchers also found out that formal risk management was uncommon in game development projects, but informal practices did exist. This was not necessarily a negative thing according to the researchers as the informal practices may suit the rapid and flexible nature of game development. For example, prototyping was used to reduce risks related to the game experience, and emphasis on pre-production phase was used to reduce the overall budget and schedule risk by making the change of possible revisions smaller. The researchers also found out that when a publishing partner was involved in the game project things like budget and schedule were stricter. The researchers also believed that agile practices could help to reduce the risks involved in game development. (Schmalz et al. 2014.)

## **2.5 Managing creativity in video game projects**

Zackariasson (2008) have directly addressed the project management perspective of game development and focused on managing the creative process of the development. Their data was collected from an ethnographic study with one Swedish case company. While the approach is more creativity-focused rather than project management-focused according to its conclusion business-like operations were very prevalent in the game company and there was a need for institutionalized creativity in gaming companies as well as for leadership development. (Zackariasson, 2008.)

Marklund et al. (2019) also look at creativity and ideation as part of the game development projects. They emphasize how ideas are built through interaction between different development groups and testers and how ideas transform during development. They also note the importance of the knowledge base of the development groups. They see that the games industry does not have a strong auteur tradition like the movie industry where one individual has a tight control of the creative process.

According to them the creative part of game development is at its core collaboration between individuals where ideas are shaped during the development process.

## **2.6 Development methodologies in game development**

O'Hagan (2014) et al. have conducted a systematic literature review where they analysed a total of 404 papers to find out how software development processes and development methodologies are used in video game projects. According to them agile practices and agile methods like Scrum, Kanban and Extreme Programming (XP) are very common in game projects and well-suited for different project sizes and different needs. Agile practices also seem to help the game get faster to the market. Many game companies mixed more traditional approaches like waterfall and requirements engineering with agile practices. This was especially common with games that had longer lifespans and requirements that are more stable when it comes to budget and development time.

Marklund et al. (2019) have noted that while many gaming companies claim to use agile methodologies in their development processes, they apply them unorthodoxly when compared to how software development companies usually apply them. Especially artists in gaming teams do not seem to fit into the Agile framework that well – they seem to be more interested in the creative process than the engineering one. This may cause there to be a kind of dual system for game development that may be conflicted in unexpected ways. (Marklund et al. 2019.)

Politowski et al. (2021) also state that the waterfall development model is still largely used in game development even though many agile practices have been adopted in the industry. Petrillo and Pimenta (2010) on the other hand analysed post-mortems made by game production teams to investigate how agile practices have been adopted in the industry. According to their research, many game development teams are informally adopting Agile practices and principles without explicitly stating them. Their findings align with Politowski et al. (2021) about how agile practices can be beneficial to projects.

Koutonen and Leppänen (2013) have conducted a survey on 20 different Finnish gaming companies to find out to what extent has agile been adopted in Finnish games industry. Of those companies nine used Scrum practices, nine used XP and three used Kanban. Scrum practices were the most used agile practices in the surveyed companies when different development phases are considered. Only one company did not use agile methods in any of their development phases.

The survey results indicated that agile methods provided various benefits like iterative and incremental development processes that helped to innovate, design, and test ideas easily. The

development teams reported that agile practices especially helped to find the “fun factor” and desired experience of the game faster. They also felt that features were fast to implement using Agile. However, code quality itself did not necessarily improve according to the survey. The researchers note that continuous integration practices could help with the code quality. While agile was perceived to help with project management and project scoping there were still issues with things like overwork and feature creep. (Koutonen and Leppänen, 2013.)

Zhu and Wang (2019) propose a different model-driven approach to game development. They mention four aspects that make game development complicated:

1. Requirements change during the development process as developers are trying to make the game entertaining and “fun”.
2. Game development also contains a creative process that does not align with the traditional software development process.
3. Multi-disciplinary communication is difficult as different disciplines have their own concepts and language.
4. Tools and workflows evolve constantly and increase demands for the games created.

The first challenge has already been mentioned by several other researchers like Koutonen and Leppänen (2013) and McKenzie et al. (2023) so is an agreed-upon part of game development. The creative process has also been mentioned as an important part of the game development process by several researchers (Musil et al. 2010; Marklund et al. 2019; Zackariasson 2008). Multi-disciplinary communication and knowledge management on the other hand has also been widely researched in both game development (McKenzie et al. 2023) and more widely across different industries (Fong 2003; Vegt and Bunderson, 2005). Evolving tools creating increased demands for the games created on the other hand is a point that has been less discussed. Based on other research it can however be stated that the assumptions that form the basis for the model-driven game development methodology by Zhu and Wang (2019) have been confirmed by prior research widely.

Model-Driven Development (MDD) has been used in different software domains and according to the researchers it is especially suitable for game development because it makes prototyping ideas easier, it reduces the requirement for knowledge and experience required in the development. It also helps to make the communication more efficient. The of MDD is to create domain-specific modelling languages to create the code for software. I could allow the experienced software developers to focus

on enhancing the software infrastructure and modelling language while other team members and less experienced developers could use the modelling language to rapidly create working prototypes of different game ideas. (Zhu & Wang, 2019.)

While the researchers argued stated that the model-based approach has not been widely used in commercial games it could be argued that some commonly used modelling features like visual scripting could be seen as model-driven development. One example of this is the Blueprint visual scripting and visual modelling tool provided by Unreal Engine. The researchers also mention visual scripting as modelling but state that these visual scripting tools do not yet utilize the full benefits that MDD could provide. (Zhu & Wang, 2019.)

### **3 Multidisciplinary project management**

#### **3.1 Multidisciplinary teams**

An influential and highly cited study by Vegt and Bunderson (2005) examines multidisciplinary teams in Dutch oil and gas industry which composed of highly skilled workers like scientists, technicians, and engineers responsible of R&D functions. The research used questionnaires for supervisors and team members to find out about the relationship of team expertise diversity with learning and performance in these multidisciplinary teams.

They have managed to get a complex view of how diversity affects these things in a team and under what circumstances. One key factor was how much the members emotionally identified with the team they belonged to. In their research, the highest performing teams had both moderate level of expertise diversity and the team members identified with the teams while the lowest performing teams also had moderate expertise diversity, but the members did not identify with the team as much. Team identification might also make social diversity (e.g. gender, ethnicity, age, etc.) beneficial for team effectiveness. Too much diversity in expertise however might be detrimental to the team's performance and with even moderate team expertise diversity managers need to foster team identification with measures like supporting the team and allowing it to develop a shared history. (Vegt & Bunderson, 2005.)

Their results also suggested that the expertise diversity provided benefits due to the learning efforts between the team members and the sharing information. They found out that team members using their diverse outside contacts was not the main reason for team diversity causing benefits, but it might be an important part of learning and knowledge sharing in the teams. Their results also indicated that conflict between experts does not lead to benefits, but it was true that teams more focused on learning together were less likely to face conflict within the team. (Vegt & Bunderson, 2005.)

#### **3.2 Knowledge processes in multidisciplinary project teams**

Fong (2003) focuses specifically on the knowledge creation of multidisciplinary project teams by reviewing relevant literature on team processes and knowledge creation and using empirical data from two projects with a multidisciplinary team that was also seen as essential by Vegt and Bunderson (2005). Fong's (2003) research gives a more in-depth view of collective project learning and knowledge sharing in multidisciplinary project teams. Fong (2003) managed to model the knowledge creation process in a multidisciplinary project team that is different from the organisational

knowledge creation process. The model emphasizes that crossing both disciplinary and hierarchical boundaries between the stakeholders of the project and the project team is a necessary process for the four other knowledge learning processes, knowledge generation, knowledge integration, knowledge sharing, and collective project learning, to occur.

Knowledge generation means generating new emergent knowledge in a team through discussion and interaction. Knowledge integration on the other hand refers to integrating the different multidisciplinary perspectives of team members into decision making. Knowledge sharing relates to how team members are more likely to share their unique knowledge than the one that is already common within the team and the benefits provided by the team having a diverse pool of knowledge. (Fong, 2003.) Fong (2003) also mentions external social networks like ex-colleagues and friends as an important part of sharing and generating new information further confirming the point made by Vegt and Bunderson (2005) which stated that it might be an essential part of team learning.

The collective project learning process entails personal learning of team members, learning done as a team, and how knowledge is transferred from previous projects into concurrent ones. Fong (2003) saw that because of this it is a widely accepted strategy to have the team members participate in several repeating projects. This is also supported by the point made by Vegt and Bunderson (2005) that developing a shared history between the team provides benefits. Fong (2003) also saw that collective project learning is at the centre of all other knowledge-creation processes and it allows transferring the benefits of other processes into valuable experience for the whole team.

### **3.3 Multidisciplinary team management practices**

Seidel and Fixson (2013) have suggested a design thinking methods for multidisciplinary teams working with innovation challenges. They defined design approach as a following process: first need finding and defining the design problem, then brainstorming ideas with a formal framework, and finally prototyping and modelling those ideas, then selecting some more final concepts. Based on 14 cases of multidisciplinary teams engaged in design thinking and consisting of novice members they made several observations and suggestions.

First, they stated that how efficiently teams manage to link brainstorming and idea creation to defining the design challenge and prototyping the idea is essential for the team to succeed. Using too much time on brainstorming on the other hand seemed to be a common problem for team efficiency so increasing the number of brainstorming sessions can be a signal for problems. They also propose that

brainstorming can be used to socialize new members to teams especially when members are added during concept development. (Seidel and Fixson, 2013.)

Secondly, they suggest that especially more novice teams should focus more on debating the processes, strategies, and objectives of the team. However, during concept selection when the decisions to develop ideas further are made too much focus on debating practices can become detrimental for the team. Due to this, the researchers suggested that teams should become less reflexive when they progress to the prototyping and concept selection part of the design process. (Seidel and Fixson, 2013.)

### **3.4 Multidisciplinary projects in small firms**

Pan et al. (2022) have researched project management in multidisciplinary research teams based on 1374 R&D projects done in 933 different small firms. They managed to find out that when it comes to small companies making the project's knowledge scope larger containing diverse knowledge makes them less likely to succeed. While video game projects are always multidisciplinary by nature, this might suggest that especially for small game companies limiting the scope of projects and the required knowledge to develop them might be beneficial. However, it is unclear if multidisciplinary research projects and multidisciplinary game development projects are even directly comparable when it comes to project management and knowledge scope of the projects.

They state that at least when it comes to innovation outcomes of these multidisciplinary projects in small companies, managing multiple projects further increases the risk of increasing knowledge scope. This means that multiple projects for one project manager might make their projects less likely to succeed as they must focus on multiple concurrent projects. One way they suggest combatting this is to limit concurrent projects for one project manager to projects that have the same problem domain – meaning that the challenges and knowledge required are similar. Experience from projects of a similar problem domain to the current project seemed to provide benefits overall, as this provides a good foundation for applying existing knowledge to new challenges. (Pan et al. 2022.)

### **3.5 Multidisciplinary teams in game development**

Multidisciplinary teams are a big part of game development as they require skills from various fields. Murphy-Hill et al. (2014) describe this by saying that game development needs both distinct technical specialization and a wide variety of different non-technical skills. So not only does the requirement for technical skills need diverse skills there is also a wide range of complete non-software

development-related skills that are essential to game development. This skill set is only getting broader when new requirements are introduced to games. For example, monetization specialists are needed to make sure that free mobile games can make a profit in some way. (Murphy-Hill et al. 2014.)

Musil et al. (2010) approach game development from a viewpoint that focuses on multidisciplinary team collaboration through flexible software processes. It gives a good outlook on how game teams have artists, designers, and software engineering professionals. However, their research focuses on developing their development process based on Scrum rather than game development project management as a larger theoretical framework. (Musil et al. 2010.)

McKenzie et al. (2023) on the other hand have examined the multidisciplinary team collaboration challenges in video game development. They use a tertiary literature review to analyse 22 pieces of academic and industry literature. Based on this they identified five key issues related to the multidisciplinary nature of game development and game development teams.

Firstly, they see that game development is inherently complex and multidisciplinary. They even go as far as to state that game development is not just multidisciplinary, but transdisciplinary – combining different fields from software development and business innovation to art and interactive media producing a completely new kind of field that focuses on developing unique, transdisciplinary, and complex artifacts – video games. Games also need to be fun to play – so changes can be made very late into development to reach this fun factor. This is why it is hard to efficiently manage game development and multidisciplinary project teams – heterogeneous skills and different types of assets made by the team need to be unified harmoniously all while the project is suspected to uncertainty and changing requirements. (McKenzie et al. 2023.)

The second key issue is how game development is still a relatively new field and it has not developed mature practices and processes suitable specifically for game development. There is not a commonly adopted multidisciplinary development framework for game development or even agreed-upon best practices for either creative production or software engineering. Even some important practices adopted by software engineering teams like testing practices or software development practices are sometimes ignored by game development teams. (McKenzie et al. 2023.)

Management practices are also reported to be fragmented and poorly defined. All these factors contribute to the challenges in video game development projects. The third key issue on the other hand is the mismatch between game industry practices and practices developed by the academy and

other industries. Game development teams and managers have often very little training on systemic approaches, managerial practices, or Agile practices – this causes them to be less willing and likely to adopt these practices. Game development is also a hard field to study due to NDAs (non-disclosure agreements) being common and projects being secret. (McKenzie et al. 2023.)

The fourth issue is the lack of a shared knowledge base within the team or a common vision for the project. This idea is very similar to the team identification mentioned by Vegt and Bunderson (2005) in their knowledge process research. Poor common vision and knowledge base lead to different kinds of misunderstanding and unalignment issues. McKenzie et al. (2023) also mention that there is a need for more research about leadership roles like game producer role that would give findings that help to facilitate unity and collaboration within the team. The fifth issue is also related to this being the lack of shared vocabulary, concepts, or terms within the team. A more coherent shared conceptual model of video game development would help key stakeholders to communicate in a more clear and unified way when making decisions and working together. (McKenzie et al. 2023.)

Marklund et al. (2019) examine the schism between the technological and creative sides of game development and note that while game development seems to be flexible and creativity has been seen as a driving force – more technological or software-oriented developers in the team can be more protective and less flexible about their area of work. For example, less technical managers can be distrusted if they try to make technical decisions without the knowledge that developers believe is required. While this concern and protectiveness has been sometimes proven well founded according to the researchers it can create conflict in the project teams. The researchers concluded that the technical developers sometimes felt that while artistry is required for a complete game product without the software development there is no technological basis on to build the game to start with. (Marklund et al. 2019.)

## **4 Theoretical framework for multidisciplinary video game project management**

### **4.1 Multidisciplinary video game project management theory**

The theoretical framework that is used in this thesis is formulated based on prior research on the topic. Based on the literature review it will incorporate game development and game project management research as well as research on multidisciplinary teams and specifically multidisciplinary game development teams. Project management and software development methodology will also be included considering both Agile practices and more traditional development methodologies.

Managing video game projects is inherently multidisciplinary. Various researchers emphasize that video game teams commonly feature designers, artists, and programmers (Musil et al. 2010) but the required roles and skills are not limited to that as games also require experts in the business domain like monetization specialists (Murphy-Hill et al. 2014) and project management specialists (Aleem et al. 2016b) to handle the business and organizational side of the projects.

This also means that the game development process is always a multidisciplinary process that must incorporate both the software development process (Aleem et al. 2016a; Aleem et al. 2016b; Murphy-Hill et al. 2014) and creative and design (Zackariasson 2008; Marklund et al. 2019) processes of the development. Also finding the desired game experience is always a multidisciplinary effort (Marklund et al. 2019). Multidisciplinary teams also have different kinds of multidisciplinary knowledge processes like knowledge generation, knowledge integration, knowledge sharing, and collective project learning (Fong 2003).

Research also emphasizes how common creative vision and shared knowledge have an important role during the planning of the game (McKenzie et al. 2023). This is well aligned with the other research regarding multidisciplinary teams (Fong 2003; Vegt and Bunderson 2005) and understanding knowledge processes could help with developing a shared knowledge base and vision in game development teams too. Marklund et al. (2019) also further discuss the idea of interaction and knowledge sharing and how this transforms the ideas during development which further suggests that theories on multidisciplinary knowledge processes and management should be taken into account when looking at game development project management.

Game development teams also seem to have incorporated software development methodologies as a core part of their development (O'Hagan et al. 2014; Politowski et al. 2021) and agile practices are

also very common in video game projects (Petrillo and Pimenta 2010; Koutonen and Leppänen 2013) However as Zhu and Wang (2019) have stated the creative activities of game development do not necessarily align with software development methodologies that well.

So, while software development methodologies and Agile practices are adopted by game development teams widely it is unclear how suitable they are when managing game development and its multidisciplinary processes. This means that the project management professional or game producer has to look for a suitable way how to organize game development in a way that helps align the creative process with the software development process. This alignment challenge is for example demonstrated by the pipelining issue mentioned by Tschang (2005) where different production pipelines are misaligned and cause bottlenecks in the development process.

Game development also has a big emphasis on managing the scope of the game project. Feature creep is often mentioned as a central risk for projects in the industry (Ullman et al. 2022; Tschang, 2005). However extensive feature cuts could also lead to challenges (Ullman et al. 2022). That is why adequate scope management is essential to the game project's success. Pan et al. (2022) also emphasizes that managing the project scope is beneficial from a knowledge management perspective.

## **4.2 Theoretical framework**

The theoretical framework (Figure 2) for this research is built in a way that incorporates this prior research on video game project management and multidisciplinary project management and the combination of the two research areas. It considers how the producer manages both multidisciplinary teams and projects but also the development process.

The first research question of this study is “How do video game producers manage multidisciplinary teams?” and the framework answers this in various project management

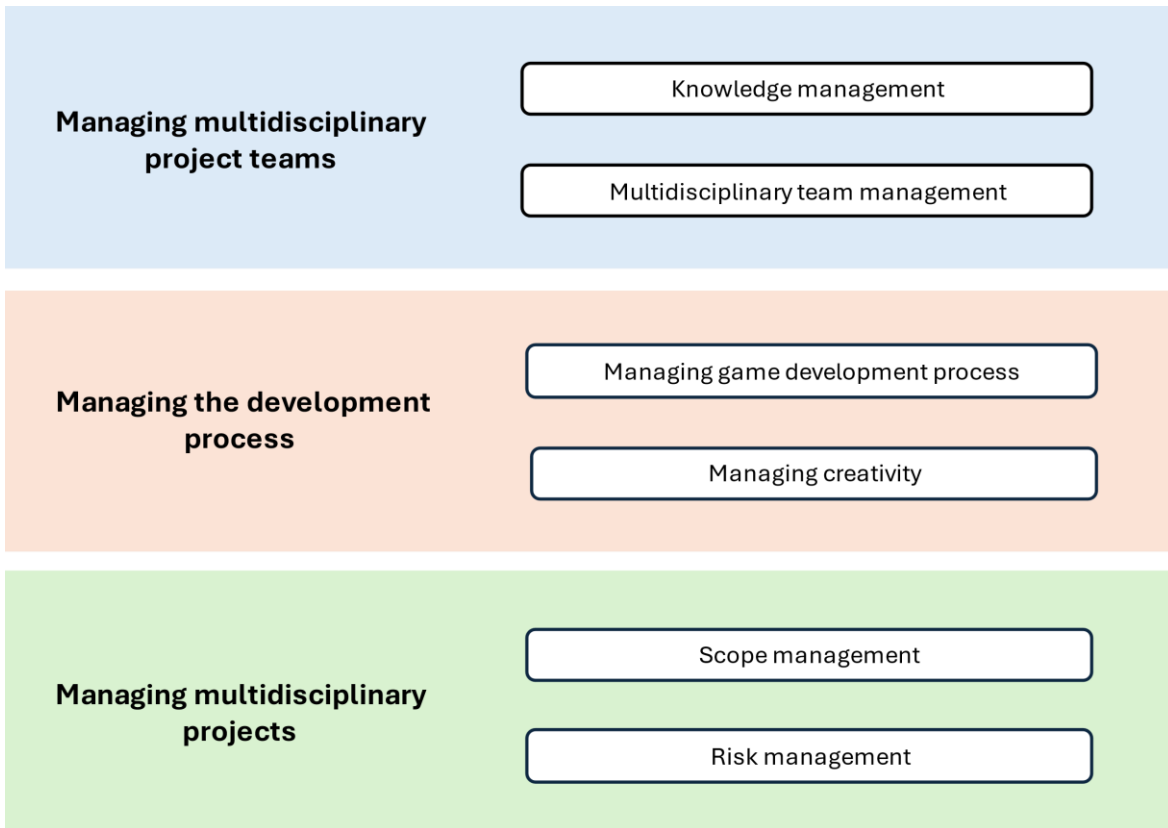


Figure 2 Framework for multidisciplinary video game project management

activities. First are the multidisciplinary team management theories that for example include composing and defining suitable project practices for the multidisciplinary team together with the team members. Then there is knowledge management of multidisciplinary teams that includes managing the knowledge processes and knowledge generated by the team members. Team management is also a key part of managing the game development process that includes both the software development and creative processes which can be organized with different development methodologies.

The second research question: “How do video game producers manage multidisciplinary game development projects?” on the other hand is also answered with more project than people focused management activities like project scope management and risk management. Game development process management is also always managing the game project and setting up the process in a way that benefits the game project in question.

## 5 Methodology

### 5.1 Selection of methodology

This research is qualitative, and it is meant to give a holistic understanding of video game project management rather than specific information about the effectiveness of certain methodologies or practices. Qualitative research is better suited for this kind of research that intends to find out “how” and “why” certain project management practices are done right now in the industry. For this reason, this research will also be done as a multiple case study where the interviewees will be from several different companies. (Eriksson & Kovalainen, 2008.) So, it does not try to give definitive proof of whether some practices are better than others but rather gives an in-depth look at what kind of practices are used and why the project management professionals have decided to use them.

Eisenhardt (1989) describes the strength of building theory from cases is that that the contradictory data from different cases can help to build novel theory. Having to interpret these differences can force the researcher to rethink the theoretical framework used. Two other strengths mentioned by Eisenhardt (1989) are that theory created from multiple cases is closely related to the empirical data meaning it is likely empirically valid, and theory built on cases often provides easily testable hypotheses that can be either supported by further research or disputed.

However, Eisenhardt (1989) also mentions several weaknesses when using multiple case studies. Firstly, a wide empirical set of data can lead to too complex theories that describe the unique cases in a detailed way but do not manage to build a theory with a good overall perspective that identifies the most important relationships from the data. Another weakness is that the rich empirical data from multiple cases can lead to a theory that is detailed but narrow and only describes the cases well rather than the phenomena itself. Eisenhardt (2021) later expands upon their article on multiple case studies and gives additional insights into the method. They note that they would describe multiple case studies as epistemologically and ontologically flexible rather than positivist. They also expand upon designing the research and for example, emphasize the role of previous research in designing the research.

Multiple case studies are especially suitable for researching video game project management from a multidisciplinary perspective as there has been little theory that looks at video game project management from a multidisciplinary perspective. Building theory from multiple cases based on earlier research allows the creating new theory that can be further confirmed and expanded upon or contradicted. Previous research has also been used as basis for designing the research and data

collection as suggested by Eisenhardt (2021). Eisenhardt (1989) also notes that building theory using multiple case research is especially well suited when examining a research subject from a new perspective or there is little previous research on the subject. This supports that multiple-case research is especially appropriate in this research.

## **5.2 Data collection and empirical data**

The qualitative method that was used to gather qualitative data was to conduct interviews of video game project management professionals. Because the interviewees are professionals who already have an in-depth understanding of their field the interviews will be semi-structured to allow further exploring important topics that are being raised during the interview. The interviews had positivist elements when trying to find out facts about practices and methodologies used but the interviews will be mostly constructionist as they try to answer research questions about “how” things are done in video game projects. However, more positivist questions to find out about “what” is being researched were also used in interviews. (Eriksson & Kovalainen, 2008.) All of the interviews were held in Finnish.

All the participants of the interviews were Finnish video game project management professionals who worked with the title of producer or senior producer. The objective was that the project management professionals would be from different kinds of companies so that this research could find out if, for example, mobile game companies have different project management practices than console game companies. Interviews were done by using video calls that were recorded for analysis purposes. The interview questions were formulated based on the theoretical framework and the interview material was also analysed based on that framework and prior research.

The participants had worked in various gaming companies with varying project team sizes. Participants also had experience ranging from free-to-play mobile games to premium console and PC games. Participants also had experience from different sizes of companies from small companies that only had one team’s worth of people to big organizations with several hundred people. All the participants had worked as a producer at least for several years. To ensure the anonymity of participants it will not be further detailed which participant has worked on what kind of projects and companies as this could make identifying the participants or companies they work in easier.

Participants had varying backgrounds. Some participants had educational backgrounds from university and had master’s degrees in information technology, marketing, and information systems science. All the participants who had a university degree noted that the education provided at least

some insight into the producer role. Some of the participants had no relevant educational background when it came to the video game industry but had gained experience through working in different occupations in the gaming industry.

When asked whether participants saw video game projects mostly as software projects or more artistic projects, they mostly agreed that video game projects were entertainment projects that had qualities that made them different from regular software projects. One of the participants noted that user experience was much more essential than compared to for example business software as people played games for the experience rather than some utility.

### **5.3 Data analysis**

The analysis was done by data codification using the Gioia method. The interview material was first openly coded to find code categories that could be combined into larger themes and theory using axial coding and finally selective coding (Eriksson & Kovalainen, 2008). This also forms a data structure for the qualitative data that was used to report the data.

Gioia et al. (2013) give out guidelines for using the Gioia method for building a theory based on analysing empirical data. According to them, data collection should be designed in a constructive manner where new topics can arise during the interviews. This is well aligned with the data collection used in this research as well as with how the interviews are planned on themes that are discussed somewhat openly.

Gioia et al. (2013) also instruct to first code 1st order codes that can be organized into 2nd order theory-centric themes. These 2nd order codes can then be into more general theoretical dimensions. This forms a static data structure that can be used to create a dynamic grounded theory model. Lester et al. (2020) also give out guidelines for analysing empirical data and note that 1st order of coding is important to mainly reduce the whole data set into observations that have some importance to the analysis. According to them, these codes are more descriptive and the 2nd and higher order codes combine these lower order codes into more relevant concepts that form the theory being created in the research.

During this research, the analysis began with first transcribing the interviews so that they could be analysed. Then the transcriptions were coded using 1st order codes identifying relevant statements from them that were then divided into theory-centric themes based on the theoretical framework used. These theory-centric themes as well as the more emergent themes from the interviews like stakeholder management were then divided into 3rd order theoretical dimensions. Based on this static data

structure conclusions of the research were selected comparing it to previous research and presenting key results from the empirical data based on the analysis.

#### **5.4 Research ethics and evaluation**

This research considers the four dimensions of trustworthiness (Eriksson & Kovalainen, 2008):

1. Dependability
2. Transferability
3. Credibility
4. Conformability

Dependability means that the process of research is logical and documented clearly. The research process has been described in detail in this chapter and the data structure is presented in the appendix. Results are also presented with relevant quotes to give more visibility into the empirical data collected and used for the analysis.

Transferability means that the research compares the results with earlier research on the topic. This has been ensured by building the theoretical framework based on earlier research on game development project management and multidisciplinary project management. Conclusions have also been selected by comparing empirical results with previous research and the theoretical framework.

Credibility on the other hand describes whether the observations made from the data are logical. To make sure that conclusions made from the data are credible, data collection was planned based on research on multiple case studies and Gioia method was chosen to analyse the data in rigorous manner.

Conformability means that the findings made in the research could be made other if researched again. Using participants from various companies that had backgrounds in multiple types of game projects was used to make sure that data triangulation considering the selection of participants was as wide as possible so that conformability could be ensured. While the number of interviews made in the research is not particularly wide reducing the validity of the research there were clear similarities found between the interviews confirming the validity of the findings made from the data.

The anonymity and data privacy of participants were protected in various ways. Firstly, all the participants and companies have been kept completely anonymous and have not been shared with

anyone. Data has also been anonymized and any direct company or project names removed during the transcription process. Some quotes provided in the results have also been further anonymized by leaving out details to ensure that participants or companies they work at cannot be identified based on the quotes.

Data has also been gathered and stored properly. Participants were informed beforehand that the interview will be recorded and permission for recording was verbally confirmed before starting the interview during the video calls. Original interview recordings will be deleted after the thesis is accepted while the transcriptions will be stored for 1 year to ensure a retention period after the research.

Artificial intelligence has been used in a minor role when writing the research. Grammarly has been used to check spelling mistakes and ChatGPT has been used to help translate some words or phrases in English. However, any deduction in the research or content hasn't been generated using AI.

## 6 Results

### 6.1 Producer role

Every interview participant was a producer or senior producer. However, the specific tasks assigned to the role varied between participants and companies, in the case where participants had experience with several different video game companies. And while each participant's tasks aligned with the traditional understanding of what video game producers do in some organizations there were different kinds of producers that had their own domains to manage.

#### 6.1.1 Producer's responsibilities

Each participant's tasks were similar and based on the results there were widely agreed-upon responsibilities for video game producers. All the different producer responsibilities are summarized in Table 2. The core responsibilities of a video game producer are summarized by one participant, and it aligns well with the responsibilities of other participants as well.

*"It could be summarized that [the producer] guides and coordinates game development."*  
(Participant 1)

This summarization views project management more from a project and development process-focused perspective as opposed to a more team-focused perspective. Producers are responsible for scheduling development tasks and managing dependencies related to these tasks. They also track the different deadlines within the project. The producer also must know what everyone is working on and who is responsible for tasks or certain parts of the game being developed.

The scope of the game is also something that the producer must manage. Producers also make sure that workloads and tasks have some kind of estimates. Scope management also includes tracking that the project is staying on budget. However, one participant mentioned that while the project has a budget it is something that is given to them from outside of the project teams, and budgeting itself is something that falls within the "game lead" role rather than to the producer.

One participant noted that different projects handle the producer's role differently but noted that the producer facilitates the team's activities, sets, and tracks project goals, and manages stakeholders and outside communication about the project status. All the other participants' responsibilities also aligned well with this summarization. Stakeholder management was something that every participant identified by every participant as something that producers typically handle while the stakeholders varied. The same participant also summarized their role from a more team-focused perspective.

*“You must know and support the team that works [on the game]. I would say that the producer sets the goals, removes barriers, and stays out of the way.” (Participant 2)*

One other participant also emphasized that it is important that the producers stay out of the way as much as possible and try to give them as much freedom as possible within the set boundaries. They also mentioned removing barriers as one important task of the producer. This also means working as a filter for the project team when it comes to meetings and requests and ensuring that the other team members can focus on their tasks.

Table 2 Producer responsibilities

<b><i>Producer responsibilities</i></b>
<i>Coordinating game development</i>
<i>Scheduling development tasks</i>
<i>Managing dependencies</i>
<i>Managing project scope</i>
<i>Tracking project goals</i>
<i>Stakeholder communication</i>
<i>Supporting the project team</i>
<i>Remove development blockers</i>
<i>Knowledge management</i>
<i>Risk management</i>
<i>Development process management</i>
<i>Facilitating creativity</i>
<i>Miscellaneous tasks (for example managing localization)</i>

One participant also explicitly identified knowledge management as a responsibility that falls into the producer’s role. Other participants also mentioned knowledge management activities as their responsibility but did not explicitly reference knowledge management.

Risk management was not directly mentioned as a daily responsibility by the participants. However, risk management activities and handling different risks related to the project were something that the participants were concerned about and when discussing the topic, it is clearly something that falls within the producer’s responsibilities. Some participants had also been responsible for creating risk management-related documents and plans.

Producers also had development and process management responsibilities, and one participant directly mentioned that they also acted as the Scrum Master of the development team managing the daily Agile development workflow. Some participants also managed waterfall methodology workflows if they used waterfall methodology for project management.

Managing creativity was not directly mentioned by the participants when asking about their responsibilities and for example, the design responsibility of the project was seen as the responsibility of the design team. However, when discussing the theme of managing creativity it seemed that the producer could help to facilitate creativity in various ways as well as help with commitment to the project vision and discovering the right game experience.

Producers also typically did the tasks that were left over from the other team. For example, one participant mentioned that they were responsible for the localization of the new game features for different language markets. It seemed typical that these kinds of spare tasks sometimes were handled by the producer when there was no other clear role for the task.

### 6.1.2 Other type of producer roles

While all the participants were either producers or senior producers some of them identified different kinds of producer roles within the gaming industry. One participant's organization had different producers for the core features of the game and another one for the meta-aspects like progression, internal shop, and other miscellaneous features of the game. These two producers typically worked under a lead producer. The participants themselves worked as this kind of lead producer who had other producers working under them.

*“Well roughly these kinds of casual puzzle-games can be divided into the core: which are the levels, level mechanics and things that happen in those levels. So, what happens after you press the play button, a game opens. Some gizmo where you can move things. That is the core-side. And meta is actually everything else that is outside of that. This is how we divide it.” (Participant 4)*

Live service games like free-to-play mobile games also might need LiveOps producer who are responsible for keeping the game running live online as well as managing the different app stores. There might also be a separate Game Lead that is responsible for the more strategic and business side of the games progress when it comes to while the producer is responsible for the day-to-day activities withing the game's development.

If the project team is working with a game publisher there is usually a producer working for a publisher that is responsible for handling the communication and project management from the publisher's side together with the project team's own producer.

## 6.2 Managing game development

Producers managed game development in different phases of the project and could identify common phases of the project which were often seen through different milestones. Scaling the project accordingly was also something that needed to be done when the project progressed. Some participants also had experience maintaining an already-launched video game. The producer was also responsible for coordinating the development tasks with some kind of development methodology.

### 6.2.1 Project phases and milestones

Participants identified the same phases of the game project where the project starts with the idea phase or conception. One participant noted that if the game is created for an outside customer, it might be that the idea is given to the development team from outside. During the conception project is often pitched either internally or to a publisher. Some kind of prototype or proof of concept is also often created. This is important especially if something completely new is being made as noted by one participant. A game development document (GDD) is also typically created during this phase.

*“Pretty standard is conception. It is the phase where you start to think what is it that we are doing, what are the base cornerstones of fantasy on which everything is built on. From that you go to pre-production during which the actual scope will be locked in.”*  
(Participant 2)

After the conception or idea phase comes pre-production where the game's scope is determined, and the production starts with creating tools. One participant mentioned that they usually build some kind of prototype during pre-production. Conception and pre-production seemed to sometimes be seen as one big pre-production phase. Other participants emphasized the importance of pre-production as a whole and said that it ensures a smooth production phase.

*“On the paper it is very pretty, but it never means that in practice you either have enough time or everything is viable to implement code-wise. For that reason, I have regarded pre-production phase as sacred. Where if a good pre-production plan can be made, in the end the ideal situation is that we can just simply get it done [in the production]. This is really iterative work. Things change, everything changes, nothing ever works out the first time, but I think that good planning in the start spares a lot of blood, sweat, and tears --”* (Participant 3)

After pre-production, the production starts using the tools and plans created during the pre-production. Production has key milestones like alpha and beta release that seemed to be standard between different projects and participants.

*“If we note that [the proof of concept] looks good, the next usually comes a kind of first playable version that is an extended version of that POC. It has a little bit more meat around its bones and maybe has further worked out graphical appearance. About how the product could actually look like. And if we can get ahead from these few key gates then there is actually just alpha, beta and then release left. And it is actually these same key gates that I have encountered both in the mobile and PC or console side. There is no difference as such.”* (Participant 5)

After that comes post-launch or post-production. One participant mentioned that how much the game will be maintained depends on the game. Post-launch can also include creating new additional content for the game after its initial release but according to one participant creating that new content can be divided into these same phases during its production. Expectations for the maintenance of the also seemed to differ between PC and console premium games compared to free to play mobile games where things like seasonal content is expected from the consumers.

Some participants also noted that as the project progressed from the pre-production to launch more people were often added to the project to scale its size. Especially prototyping the idea could be done with smaller amount of people. When size of the project increases more specialized people are often needed for the project team.

When it comes to live-service games that needed more analytics like mobile games especially the amount of business and analytics experts increased after the project is launched. One participant also noted that they do not want to globally launch games that scale properly as they try to compete with the biggest games on the mobile game market.

Every participant viewed the project progress through key milestones that were usually set by the producer. Testing was usually included when these milestones were reached, and the project was evaluated at each milestone both internally and in some cases by outside stakeholders like customers. Some participants also referred to these milestones as gates.

*“[the project] is tracked through milestones so before certain dates certain there are certain goals and those are tracked – – So there are kind of test rounds and you get feedback if something needs to be edited or does it work like this – – Typical milestones are prototype, there is a list of things that need to be included. Then there is alpha that has practically all the features and functionalities, but art does not need to be ready. After that comes beta that has all the content and functionality ready and then gold master where the bugs are fixed that were discovered during beta.”* (Participant 1)

One participant also mentioned that these milestones can be combined with development sprints where there is a certain number of sprints before a set milestone. This can help to track the progression before reaching the milestone.

### 6.2.2 Development methodologies and software development practices

Out of the five participants one said they only use waterfall method to organize the development. However, they stated that the projects still had some Agile qualities as implementation always had some iteration. They also noted that waterfall does not mean that plans cannot be changed when there is a need.

One had used waterfall before in the past but used mostly Scrum now. They noted that Scrum works nicely if sprints are respected, and things are not added to the scope during the sprint. They also saw that development methodologies ensure that there is clear organized process where progress can be presented to different stakeholders easily.

Two participants used some form of Agile. One used Scrum but did not adhere to its principles strictly – things like sprint reviews were combined with sprint planning and retros were not held after every sprint. They also said that their project sub-teams were free to organize their work quite freely and for example analytics, art and audio teams usually used just Kanban to manage their workflow instead of Scrum. Kanban was also mentioned to suit smaller and simpler projects well.

The other participant also used Scrum and mentioned that especially when it comes to these Scrum ceremonies things were sometimes dropped. The participant noted that the company leadership did not always see the value in them as they took a lot of time. While the participant felt that Scrum worked especially well with the iterative nature of the game development when finding the right game experience, they did also bring up that as a scheduling tool Scrum does not perform that well.

One said they combine Scrum with Waterfall in way that Agile is used in software development side of the project, but art-content is created according to the waterfall methodology where the project is planned from start to finish. They also mentioned that even the more waterfall-like art-content creation could adapt to changes if needed by adjusting scope or changing targets.

*”Agile Scrum is pretty common, especially on the software development side. Second pretty much the most common is Waterfall methodology, where the project is broken down from start to finish – step by step and then you scope the timeline. It is more common when doing for example art content, characters, assets, guns, and so on, because there it is much easier to determine the end product than for example in programming. So many*

*projects vary and even within the project there can be different approaches depending on the department that is doing the execution.” (Participant 2)*

Several participants also used feature freezes where if tasks were not developed until some deadline they were dropped from the next release. This ensures that there is enough time to test the version being released.

Most participants were not closely involved with the software development practices of the team, and it was handled by a technical lead. However, the projects had clearly defined software development practices like version control. One participant noted that their software development practices were maybe a little bit lighter when compared to the infrastructure department of the organization. One other participant was more closely involved with the software development practices and said they had as much automated testing as possible and used version control also as a project management tool.

### 6.2.3 Testing

While testing has a large role in game development and was mentioned by various participants to management risks it was also a core part of video game project with both more traditional code level testing but also with large user testing being done. Testing was sometimes done by the game’s publisher, sometimes there was a separate testing team in the organization and sometimes the testing specialists were embedded in the project team itself. Testing was seen as important by all of the participants.

One participant saw testing first and foremost as quality assurance but also as a way to provide outside feedback about the project’s status. Other participants saw that testing practices varied a lot between different projects. In their view testing was mainly done to validate that the core gameplay experience worked as intended and then it was extended to the full game with the experience being test on each milestone.

Participants also mention AB testing where some control groups are given two different versions of the game and then their behaviour and experiences are evaluated to determine how the difference between those versions affects the game experience. AB testing seemed to be much more common when it comes to live-service online games rather than single-player PC or console games. AB testing also requires wider analytics capabilities so that data about the player's activity can be gathered.

Participants also saw external video game testing companies as an integral part of the testing landscape as external testing providers had employees specialized in testing and were very much

more analytical as compared to using normal people on testing. These testing companies also had a wide variety of different platforms like different mobile phone versions, different operating system versions, and various gaming consoles that could be used to test the game extensively on various platforms.

One participant also mentioned that different gaming platform providers like console manufacturers Microsoft and Nintendo have strict requirements for games being published on their platform. These can be for example console specific video game controller key bindings. To pass these requirements a separate testing team for this is usually required.

### **6.3 Scope and dependency management**

Producers must manage the scope of the game project. This starts with evaluating the scope during pre-production of the game project. Game scope is closely related to the money and time available for the project. The game scope must also match with the scope of the game idea – is the idea something that can be executed with the time and budget available for the project.

One participant also mentioned that players expectations depending on the game genre must be considered. They emphasized that the player expectations must be matched or exceeded with the skill level of the team available. They mentioned that market research about the player expectations can help with this decision to set appropriate goals for the project.

*“First you have to decide what scope of a game we are making in practice. So, you can think, about what kind of funding is available, and how many years can be spent developing the game. Also, what is the game idea, so what is the minimum time in which it is possible to produce? And then you think if the budget is this, you can get this much more. Or alternatively, if the budget is smaller is it worth it to make it with a smaller number of features and assets? And according to that the scope is decided in an early phase and then you start making it. And then during the project, it is managed. Traditionally if challenges come up the things must be dropped – if schedule or budget cannot be increased.” (Participant 1)*

During the project, the scope is managed and if the scope is revealed to be challenging either scope must be reconsidered or resources – either time or money available must be increased. However, one of the participants remarked that just “throwing people at the problem” usually is not the best way to solve it. One participant also said that reorganizing tasks sometimes bought important time to solve challenges without needing to alter the scope. They however sometimes saw borrowing people inside the same organization to help with the project as a viable solution for challenges with handling the

scope. It also seemed that delays were expected to happen from time to time and were not seen as a big failure.

One participant discussed that the decision also depends on various factors. If the feature that is dropped is not essential for the game experience it can be more easily dropped. They also noted that it is useless to drop features in a place where there is no shortage of work. The scope must be altered in a place where there is a shortage of resources.

Part of the scoping process is making accurate estimates for different project tasks. One of the participants emphasized that you have trust the different disciplines experts when evaluating task workloads.

When it comes to the tasks of the project producers need to also consider the different dependencies of these tasks. Dependencies between the different disciplines were also made clear by the observations for the participants. Designs by game designers must be made before implementation can be made by artists and programmers. Art content must also be synced with the programming either by creating art assets before the implementation or making programmers use some kind of placeholder assets. Sometimes there might also be business specialists like marketing people that needed art designs to proceed with their tasks.

One participant saw dependencies as the biggest risk in game projects. They highlighted the importance of pre-production and said that being cautious when planning tasks and figuring out the dependencies early on usually pays off later in the project. They had recently themselves forgot to add one task to the projects workflow and this caused a delay that also needed communication with platform that was hosting the game.

*“Once again pre-production is important. When you start to do something, you should usually measure two to three times before you start you start [figuratively] cutting. It works much better.” (Participant 3)*

#### **6.4 Managing multidisciplinary game teams**

Participants saw managing and working in a multidisciplinary team primarily as a positive thing even if sometimes there was some conflicts related to multidisciplinary nature of these teams. One participant mentioned that especially in planning the project it very valuable to have people from various disciplines offering their perspective.

The possible conflicts were usually related to differing perspectives or interests. One participant for example mentioned that they had encountered situations were both the artist lead and design lead had

strong opinions that conflicted. The participant mentioned that in these cases it was the producer's responsibility to figure out how this conflict was resolved:

*“Everyone has their own specialty, which they are very skilled at and passionate about. And they approach many problems and matters from that perspective. A critical part of success is that you manage to create a team or you have a team that can also see each other's perspectives and find the compromises there. – – So there will be conflicts and it is a big part of [my] daily work to find the common ground.”* (Participant 2)

One participant also mentioned that miscommunication also played a role in these kinds of conflict situations and according to them, it was usually the main reason for conflicts. However, even they noted that in some cases strong views about the project could also cause conflicts. Many participants also saw the word “conflict” as too harsh and in most cases differing views caused very little challenges in the project teams. Participants mentioned that especially experienced professionals could easily deal with these kinds of situations where views differed in a professional manner.

These conflicts could be solved in various ways. One participant mentioned that the most important thing is to get the parties with differing ways together either physically or in a video call rather than solving it by messaging because this minimizes the risk of miscommunication. One other participant also mentioned that workshops, mediation sessions or daily scrum sessions worked as occasions to solve these kinds of conflicts. Two participants mentioned that they mostly solved conflicts informally through social interaction.

One participant also discussed that considering the individual personalities of team members is even more important than the differences created by different disciplines. They said that for example some people are more likely to go off on tangents and need guidance to help them.

Some people on the other hand are more disappointed when features need to be dropped from the game and need help dealing with that. Some team members also need the producer to help them make decisions. However, according to them, the producer should not be the one to create the decisions but rather summarize ideas and guide the team into making these decisions.

### 6.4.1 Multidisciplinary team composition

All the participants identified the basic roles of art, coding, and design in their project teams. Usually, these different crafts or disciplines were organized into their own sub-teams within the project with all having their own leads that were domain experts in that discipline. Audio could be seen as part of the art team or as its own separate discipline. These roles often form the core project team. Figure 3 illustrates how the roles mentioned in the results form up multidisciplinary team – some specialties themselves requiring multidisciplinary skills.

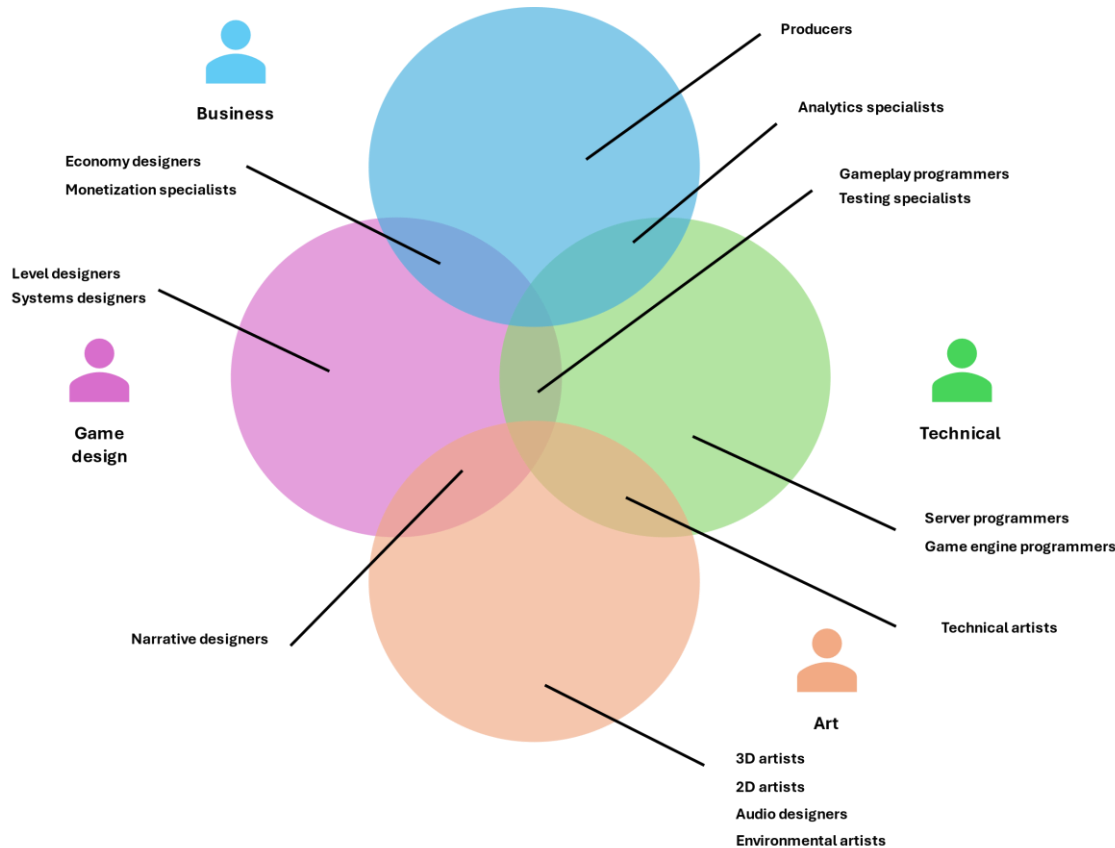


Figure 3 Multidisciplinary roles of the project team

Each discipline also had its own specialization inside that discipline. For example, programmers can be specialized in roles like server programming gameplay programming, or some other specific technical area of the game. Designers on the other hand specialize in designing some areas of the game like level designers design the level in the game and roles like narrative designers focus on shaping the narrative of the game. Artists can also specialize in similar fashion in things like 2D or 3D design or environmental design.

In smaller teams, one person usually had a wider role whereas in bigger projects and organizations there were more specialized roles. For example, one participant noted that they did not have separate

3D artists, and their artists handled all aspects of creating 3D assets for the game doing both the modelling and creating textures for those models.

There are also roles like tech artists that are themselves inherently multidisciplinary as tech artists need to have both programming skills and artistic skills to create different kinds of special effects for the game using the game engine. There were also unique roles like an economy designer. The designer is not focused on designing the gameplay itself but rather the inner economy of the game where the player has different items and in-game currency that can be spent and bought.

Testing specialists could be part of the project team but in most cases, they were mentioned either inside as a support team in the organization but outside of the project team or as something that was outsourced to a separate testing company or the publisher of the game.

Business and analytics specialists were rarer but in some cases, they were included in project teams. One participant mentioned that they had monetization specialists that worked with their designers to figure out how the game will be monetarized.

#### 6.4.2 Building project teams and team formation

Forming the project team is an important part of the project and suboptimal team composition was seen as a big risk for the project. If interests or expertise did not match the game project in question it usually caused big challenges in the project. One participant mentioned that for example, people who did not like free-to-play games were usually a bad fit for a project that is planned as a free-to-play game. Another participant also mentioned that even inside one discipline like programming their expertise within that discipline varies a lot and some people and the same programmers are not usually suited for programming the game user interface (UI) as the ones programming the graphics of the game.

*“It's not often you have a team like that readily available that is a perfect fit for that project, whatever it may be. Either there are simply too few people, or there are people who just happen to have a lot of same interests. – – Then there are for example programmers who die of boredom if they must do UI. That is a tough gig to throw to that person, but we must have an UI, or we do not have a game at all.” (Participant 5)*

One aspect that also affects team formation is what kind of people are available to work on the project. One participant noted that in general there was an overabundance of junior workers and shortage of shortage of experienced specialists. According to them the overall skill requirements for even juniors were quite high.

When it comes to talent that was especially in demand for video game project teams technical artists were mentioned by to participants. One noted that skilled technical artists were hard to hire as it needed both advanced graphic coding skills and artistic skills. It also seemed that specialized experts like game designers that were particularly good in designing systems within the game or skilled 2D artists were in demand and hard to find.

Specialists like analysts and server programmers that could easily switch occupation from gaming industry to other industries utilizing the same skillset were also mentioned when discussing in demand talent. According to the participants these kind of specialists were harder to keep in the organization because when trying to keep them they had to compete with companies working on various industries not just in the games industry.

Part of building project teams is also handling training new workers. One participant observed that even more experienced workers need time to orient to new teams and their practices. Most participants said that they train new people organically by trying to give them easier tasks that help them familiarize themselves with the project first and through that experience they can be given more challenging tasks. Participants also mentioned more formal onboarding practices like assigning a mentor to new workers.

## 6.5 Managing creativity

Participants had various views about managing creativity during the game projects. One participant described how the constraints of creativity change during the project. During conception and pre-production there is more room for creativity whereas in production when the vision of the game is locked creativity happens more in the details of the game.

*“That obviously varies quite a lot depending on the phase of the project. It affects how wide the sandbox edges are. During conception they are very wide whereas in production we already know that this is what we have to do. And in that case the creativity happens more in the details.”* (Participant 2)

They also emphasized that it is important to give as much space as possible for creativity, producer must give clear constraints like timelines and then clear barriers like pointless meetings to help the focus. According to them building teams that can independently execute and create solutions within these boundaries is also important.

Another participant also acknowledged that giving project team members independence to solve challenges after making sure the team is committed to the project is how to best ensure creativity.

The producer should only step in when it's clear the problem cannot be solved without some additional input or decision from the producer. They also underlined that it is important that the team members feel that they can be open with their ideas and those are valued even if sometimes those ideas are not used. In that way, the project uses as much creativity as possible from the team members in the form of their ideas.

One participant noted that it can also be possible to include programmers in more creative and design side of projects depending on the case. For example, sometimes it can be better to let some kind of system be designed by the programmers because they often understand the technical options available for designing that system. Participants used animal movement in the game world's nature as an example of this.

However, the same participant noted that programmers do not usually appreciate half-designed feature tasks that need more design due to the unfinished design work by the designers. But they did note that in smaller teams like small mobile game teams' people often design new features and divide responsibilities in the implementation informally even during coffee breaks and other informal communication.

### 6.5.1 Game concept development

All participants identified an ideation phase or game concept development stage when observing game projects. Sometimes these ideation phases also appeared later in project when developing new content or new features so while ideation phase typically happen during conception or pre-production smaller ideation phases can happen later too.

One participant mentioned that they have the whole project team included in the initial brainstorming when a new project starts. After that, the responsibility shifts to the design team. That is followed by determining the scope with the design team and different discipline leads. For example, the idea must be realistic to implement technically. They also mentioned that after the brainstorming producer tried to interfere as little as possible with the creative side.

*"Whenever a project starts or a new idea is initiated, we always have a team brainstorming session where everyone can share their ideas. After that, the writing responsibility shifts completely to the design team, and they are given the space to work on it. Then, together with the leads of design and other areas, we look at what is possible within the scope and what makes sense. Especially at this point, it's the developers' expertise that determines what is feasible and what makes sense, and so on. So, things start moving forward from there, and from a production perspective, we try to interfere as little as possible with the artistic side." (Participant 1)*

After that initial ideation phase presenting new ideas for the game is possible in an internal message channel. These ideas are collected for the future but are not included in the active development of the game, except if they are essential to include. One other participant described that there are sometimes situations where global gaming trends make it economically almost essential to include some kind of new content that was not originally planned in the game. They noted that sometimes the economic benefit might even mean that some gameplay features will get less attention.

One participant described that workshops are a common way to facilitate creativity. They are a way to get different specialists to sit together, make ideas and solve certain problems. Other participant had been sceptical about workshopping but talked positively about one brand partnership brainstorming workshop they had attended.

The same participant had experience of more formal concept development and of more informal creative phase where the game project team just “negotiates, fights and vibes around the same table”. They said they prefer this kind of democratic approach to the creative process. However, they also noted that despite of this more democratic approach, discipline leads did usually have more say in what is included in the game. They also described this collective brainstorming as fun.

One participant had less experience about the game concept development stage as they had joined their current game project later in the development. They did note that the game concept development was often figured out between the game lead and game designers.

When they previously joined the projects as a producer the concept had already existed. After that, the creative process of game content and levels was mainly done by game designers and level designers. One other participant also noted that sometimes game lead and creative directors had the responsibility of developing the game concept while other times they only provided a loose high-level concept for the game. They also brought up that there are situations where the game is being developed for a customer who already has decided on the concept.

The participant that had joined the project after the concept development was already done did however have experience of collective brainstorming sessions when it came to additional content to the game. In these sessions, project team members came up with new ideas and voting was used to decide the best ones. The producer also knew that game designers and level designers used these kinds of brainstorming sessions when coming up with new level mechanics for the game.

They also had some sort of brainstorming when coming up with plans for regular future content for the game where a game designer, product manager, user research specialist, and a community

manager were present. After that, the idea is iterated with concept art made by artists. The participant also brought up that artificial intelligence could be used to quickly visualize new concepts and test out ideas. This AI art however was not necessarily used in the final implementation.

During the game project, the game's vision must be both created and managed. According to the participants the ownership of the game's vision was rarely held by the producer – usually, some sort of creative director or game lead had responsibility for the vision and had the last say on what is included in the game. The vision was usually documented in a GDD. However, when discussing the game's vision, it was clear that the producer was able to influence how well people are committed to the vision or at least it seemed possible.

One participant noted that it was important that the ownership of which part of the game and the roles were clear. According to them in some projects, this could be a core team that votes on the decisions, but experience of the certain genre or subject matter of the game should be taken into consideration when determining the ownership. The decision-makers should be chosen strategically and with the project's goals in mind.

Sharing the game's vision with the project team clearly and often enough also came up to help uphold commitment to the vision. One participant noted that it is critical that every discipline's lead understand the vision clearly and know what their disciplines contribution to that vision is.

*“Game's concept and vision must be presented clearly and shared often enough. The most critical thing in the implementation is that all the team leads and others understand very clearly what it is we are aiming for and what their specific area of expertise contributes to that vision. For instance, how the artists bring that certain feeling or visual to the experience and how it supports the gameplay loop that the designers create.”*  
(Participant 2)

They also had a principle called “disagree and commit” that spelled out that it was acceptable to disagree with ideas presented but after the decision is made and team has a consensus even those who had voiced disagreement committed to that decision. According to the participant especially team members with experience and maturity could do this without any issues. Other participants also noted that usually team members could be professional even when they had some disagreements with the vision or decision being made.

One participant told that in very rare cases during their career there might have been some artists or designers “with unnecessary big egos” who had problems with committing to decisions they did not get to make. However, they did note that nowadays this was extremely rare, and those kinds of personalities rarely could function in bigger organizations developing video games.

Another participant also noted that usually team members were very professional and had no problem committing to the game's vision. The only challenge they could think of was if there were some very strong opinions about the type of game being made – they said that for example, some people might have very strong negative opinions about free-to-play games. Those kinds of people are often a bad fit for free-to-play game projects. One participant even noted that they themselves are much more comfortable working with premium games rather than free-to-play games as they did not like their often addictive qualities.

Two producers also discussed how the game project usually changes shape during the game project. The other one pointed out that handling this change is difficult and time-consuming area of their work. Because even if the game changes from the original vision in some ways it still must be something that the project teams and stakeholders can stand behind and it must still match original business idea of the game. They also noted that this change is easier to handle with a smaller team than a big one.

The other producer that discussed the vision change during the project noticed this in a situation where the game was in a long post-launch maintenance phase and the game's lifecycle had been so long that even the people working on it had changed. This led to questioning some parts of the original game vision and even the art style had evolved from the original concepts. The producer saw this as ultimately a positive evolution of the game.

When asked about managing the game experience it seemed that this was seen as something that was more as the design team's responsibility. However, one producer did note that getting all the planned features in the game was in their case the producer's and the technical lead's responsibility.

Many participants emphasized the role of testing in ensuring the intended game experience. The project milestones often acted as a control point for the game experience too. One participant also said that finding the "fun" in that is needed for the intended game experience was something that cannot be planned beforehand, but it has to be tested and then iterated. That is why this participant also saw Scrum as a good project management methodology as it works well with inherently iterative nature of game development.

## **6.6 Knowledge management and communication**

Only participant themselves named knowledge management as something that is part of their responsibilities. However, all the participants observed different ways to share knowledge and

communicate in video game development projects. Documentation also came up as an important way to collect and share information in game projects.

One participant discussed how learning is seen very much as an individual's responsibility and people are expected to familiarise themselves with information available. They also mentioned that while experience does not necessarily help people to understand how people specialized in different disciplines than their own do their work, it helps to understand the overall project constraints and practices. One other participant also mentioned how experience usually allows people to collaborate with the team better..

### 6.6.1 Communication in project teams

All the participants used some kind of digital communication platform as the basis for project communication. Especially communication platform Slack was referenced as something that the producers used in day-to-day communication and was directly mentioned by four of the five participants. Microsoft Teams was also mentioned as a way to make calls and hold digital meetings between team members.

One participant brought up that the fact that Slack has channels where everyone can see the messages, as opposed to private messages, as especially beneficial. According to them this kept decision-making visible to everyone and emphasized how they made sure that no decisions were made without the producers and creative director seeing them.

One participant used Slack but said that sometimes it is hard to find information from the platform. They also prefer a more pyramid-like communicational structure if the organization is so big that there starts to be too much communication to follow for individual team members. In those cases, it's better if some communication happens only between different discipline leads and managerial roles.

*"At one point the idea of a flat organization was very popular and it was very appealing and functional but, in my opinion, humans are pack animals: you're probably only interested in the fates or well-being of at most 100 other people. You do not have the energy to pay attention when everyone gets lost in the noise." (Participant 3)*

One other participant also observed how when they were part of a smaller-scale project team, they were able to be part of every communication channel on Slack but when the scale is bigger it might be challenging to make everyone be able to follow even the communication inside their own discipline's area. One of the participants also noted that communication is also important to ensure smooth work between the different disciplines working on the project.

## 6.6.2 Sharing information in meetings and sessions

The most common way to share information between the project teams were different kind of meetings. Participants mentioned different kind of weekly and daily meetings where status of the project or its components were checked and tasks coordinated. However, any daily meetings were for usually just for quick syncing of project tasks.

One participant also mentioned that when they start working on a new feature, they have a separate feature kick-off to which the people working on that feature are included. Participants also mentioned different kind of demo sessions where new features were presented to the project team and discussed.

One participant also mentioned that they have a studio wide business weekly where different gaming projects share information between each other. For example, AB-test results are sometimes applicable between different games within the same genre. Any information that is useful for the project team will be later shared with them too.

*“And because we make the same genre of games in every production, even though they are different products there are a lot of similarities. We can share AB-test results: what kind of features have worked and what kind have not.” (Participant 4)*

Different kind of retro perspectives and postmortems were also mentioned as a way to share information and especially to learn from work already done. One participant mentioned that especially different challenges that had occurred could be discussed and then fixed by making process changes based on that discussion.

Other participant mentioned that these postmortems were very important especially if the project had failed or never released. They also mentioned that especially different processes could be improved based on the information shared on these retrospectives and postmortems. Postmortems could also be made into a document that could be later used to learn from past projects.

## 6.6.3 Documentation

Documentation was seen as an important way to share information in project teams. One participant mentioned that they have shared a digital document bank where for example every feature is designed and technically implemented and how they can be accessed in the game engine. Other participants also mentioned shared documentation banks like Confluence and Google Drive.

This was seen as critical information by that participant. They also mentioned that unified language is very important when it comes to documentation so every person working on the project understands

the terms used. Terms should also be clearly defined and according to the participant achieving this unified language can take a lot of time.

The same participant also mentioned other examples for documentation like retro perspective and post-mortem documents that have clear highlights for the most important content in the document. This can according to them help retain information in the organization even if some people leave the organization. They also highlight that people must be encouraged and be given enough time to familiarize themselves with the documents.

One of the participants was more critical about documentation and pointed out that in the worst-case scenario badly made documentation can be completely pointless. They did see a shared digital documentation space as a useful way to share information but mentioned that if documentation is too heavy it ends up being very hard to maintain an unnecessary overhead for the project.

They did point out that especially more technical documentation in the code is always necessary. They also mentioned that documentation needs to be light enough and have a clear purpose. As an example, they saw project roadmaps as an important way to keep everyone track on the project schedule and feature documentation provided important knowledge for people later needing to work on something related to that feature.

However, they observed that even with good documentation there is always tacit knowledge that cannot be stored in a document or shared with the team but minimizing the amount of this tacit knowledge is important.

One participant also discussed that a project management system like Jira also works as documentation when both present and already completed project tasks are described and their status tracked. According to the participants this is a very helpful way to share information in the day-to-day work.

Other examples mentioned for documentation were testing documents, production configuration documents, LiveOps schedules, and concept art. Tools for documentation mentioned in addition to the documentation banks were Miro, Figma, and Word which could be used to document graphical assets. One participant also mentioned that they store some documentation on Slack.

#### 6.6.4 External networks

When discussing external network participants had varying views about their role and importance. Many participants mentioned that the gaming industry has a relatively open culture that encourages

learning from others and sharing experiences. Finnish game industry was also seen as a positive environment.

*“The gaming industry is unique in that a lot of things are shared between companies, and there are conferences like GDC, where companies literally share how they succeeded in certain areas.”* (Participant 2)

One participant however did note that usually people working on games had NDA's that limited how openly especially ongoing projects could be discussed. However, even they mentioned that past project successes even for example regarding technical architecture of the game were often openly shared. They did note that external networks usually provided more general inspiration and viewpoints rather than exact solutions for challenges.

Other participants also echoed these views but mentioned that sometimes directly helpful information could be gathered using contacts outside the organization like solutions for technical problems and recommendations for tools used on projects. One participant also noted that sometimes people working outside of the project but inside the organization could be valuable sources of information outside of the actual project team. They said that a bigger company usually means that you have a bigger knowledge pool that can be freely used.

Another participant also mentioned that while sometimes things were shared more openly, they were usually quite protective of their own ideas. Rather than using outside contacts information was gathered by doing competitor analysis. Experience gathered from working with different game companies was also seen as a benefit when trying to recruit new candidates.

The role of game conferences was also discussed, and many participants directly mentioned the Game Developers Conference (GDC). Most of the participants saw that conferences had little importance on the actual game development work but did mention that for small indie developers and people looking for employment opportunities conferences could be very important. One participant who had experience working on a smaller-scale game project and they mentioned that conferences were able to give them valuable new contacts. One other participant also mentioned that conferences were usually the place to openly share information about past successes in the game projects.

## **6.7 Risk management**

Participants identified both external and internal risks for the success of the project and had various ways to manage these risks. One participant mentioned that especially good time buffers in the schedule help to deal with risks. They also mentioned that there are always risks that cannot be

completely avoided like the risks of key people leaving the team. Especially very specialized team members leaving the team could cause notable challenges to the project.

One external risk they saw was ensuring that the game had funding. Especially failed project could mean that finding funding for succeeding project can be very difficult. One other participant also managed that these kinds of external stakeholders must be managed as the project cannot proceed without their commitment to it.

Other participants mentioned that especially if the project were bigger and more complex the number of specialists would grow and could increase the probability of this kind of key person risk. According to them bigger and more complex projects also increase the chances of miscommunication and unclear vision. They saw that knowledge management and vision management were needed to mitigate especially these risks. They also mentioned that because gaming is a fast-moving industry where trends come and go ensuring that the game meets players' expectations is especially important. They mentioned that regular play testing and using the project milestones to review the game's status was very important.

One of the participants saw dependencies as the most important risk when it came to projects and according to them these dependencies must be analysed as thoroughly as possible during the pre-production phase of the project. However, they did mention that having specialists of various disciplines helps to reduce risks as this specialized knowledge provides valuable information when planning out the project.

*“When people know what needs to be done, they can usually also explain better why something cannot be done. If I’ve only negotiated with, for example, the coding team or the art team, then suddenly someone drops a bunch of features or whatever for the code to do that they’ve never even heard of. The more knowledgeable people you have in the mix, the better the end result usually is” (Participant 3)*

Other risks mentioned by the participants were feature creep where the game scope grows and suboptimal team composition with wrong people being assigned to tasks that did not suit their expertise or interests. Technical risks related to the IT infrastructure were seen as something that could be easily mitigated by things like backups and at most caused small delays.

One participant had a documented risk management plan, and one other participant used a fishbone diagram to map out risks and their probability. According to the participant who had used the diagram visualizing and organizing information in some way always helps and this is also true when it comes

to risk management. This kind of risk management document also helps to communicate the risks to stakeholders and acts as proof that risks have been considered properly.

## 6.8 Managing stakeholders

Managing different stakeholders was a producer's responsibility that was discussed by most participants. One participant noted that there is always someone funding the project who needs visibility about the progress of the project. They also noted that these stakeholders whether they were the organization's leadership, customer, publisher, or brand partnership need to be committed to the project for it to succeed. Things like project scope, budget, and schedule need to be agreed upon. Another participant also noted that if for example the publisher and the project teams start to drift away from each other working on the project may become very difficult.

Game publishers were a common stakeholder, and they usually had their own producer working on the publisher side communicating with the projects team's producer. Publishers usually take some of the project responsibilities to themselves like community management, localization, marketing, and distribution.

*"So, traditional publisher services include community management, testing, localization, marketing, and then distribution management. Of course, if the game is just on Steam, it's pretty straightforward. But those have traditionally been the publisher's responsibilities, along with financing the development. Then, of course, one risk in this business model because usually publishers funding payments are based on milestones."*  
(Participant 1)

Testing was sometimes also the responsibility of the publisher. While sometimes discipline leads, or other team members were needed in discussions with these stakeholders' the producer was at the very least the first link of communication and facilitator when it comes to communication with the stakeholders.

One producer also had experience working with stakeholders that had significant cultural differences that made collaboration more challenging. While there were more practical challenges like language barriers and time-zone differences, the participant also mentioned that differences in how hierarchical the organization was also made things more challenging.

## 7 Conclusions

The games industry has grown rapidly, and project sizes have increased in size and complexity as the skills required by game projects are only expanding. Prior research on video game project management has focused more on individual parts of the video game project rather than the producer's perspective (Marklund et al 2019). There is also a need to synthesize results from heterogeneous research related to game development (Engström et al. 2018).

Design teams consist of designers, artists, and programmers (Musil et al. 2010) but also experts in the business domain (Murphy-Hill et al. 2014) and project management specialists (Aleem et al. 2016b). The project also has coexisting software development process (Aleem et al. 2016a; O'Hagan et al. 2014; Politowski et al. 2021) and creative (Zackariasson 2008) processes. Archiving the desired gameplay experience (Marklund et al. 2019) and common vision (McKenzie et al. 2023) needs the collaboration of all the disciplines in the team. Multidisciplinary teams also have unique knowledge processes (Fong 2003; Marklund et al. 2019; Vegt and Bunderson 2005) that need knowledge management. They also need scope management so that the project avoids scoping risks like feature creep (Ullman et al. 2022; Tschang 2005) and risk management.

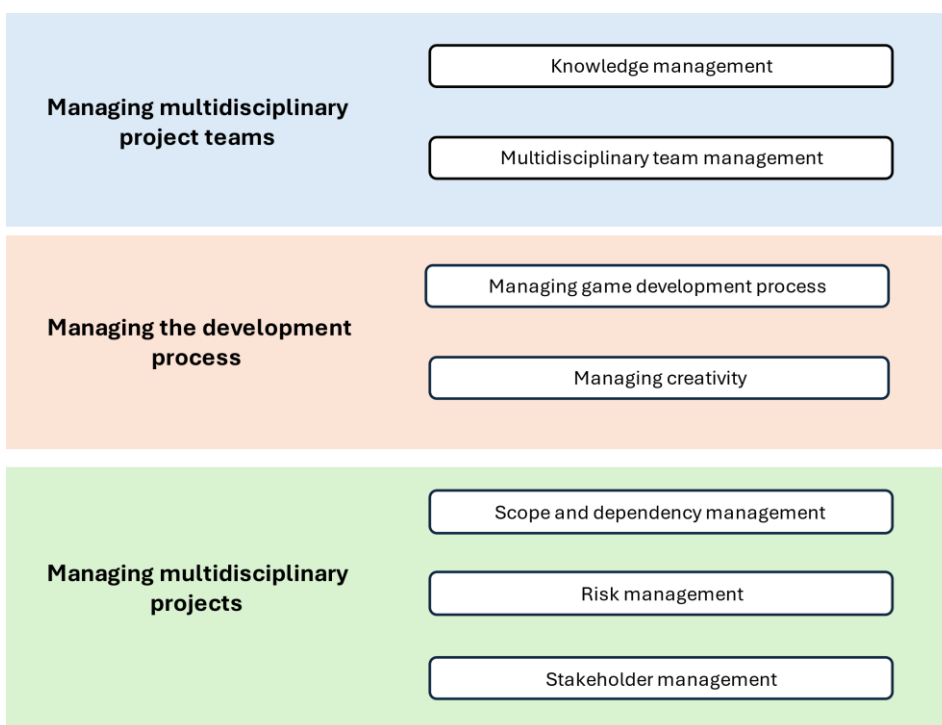


Figure 4 Updated theoretical framework

Theoretical framework has been upgraded with the results of this research (Figure 4). Stakeholder management was identified as an important producer responsibility that had not been discussed in

earlier literature. Dependency management has also been added alongside with scope management as both have to do with scoping and organizing areas and tasks of the project.

## 7.1 Managing multidisciplinary project teams

Table 3 Multidisciplinary project teams conclusions

Topic	Previous research	Results
Multidisciplinary team management	Murphy-Hill et al. (2014): - Games need distinct technical specialization and wide variety of non-technical skills. - Skillset only get broader as game become more complex.	- Supported the need for multidisciplinary roles, increasing complexity. - Especially free-to-play and live-service games with long maintenance periods require even more specialists.
	Musil et al. (2010): - Presents the main disciplines in game teams: artists, designers and software engineering.	- Supported the division into art, design and software development. Although business roles were also prominent in the results.
	Aleem et al. (2016a): - Team configuration is a critical success factor for game projects.	- Supported by the results; seen as a key risk for the project.
Knowledge management	Vegt & Bunderson, (2005): - Project experience made the teams less likely to face conflict as participants that more experienced members had an easier time accepting changes.	- Supported by the results. More experienced members had an easier time accepting, for example, feature drops from the project.
	McKenzie et al. (2023): - Lack of shared knowledge base is one of the issues of multidisciplinary teams.	- Results answer this as documentation could help to achieve unified language for the project team.
	Fong (2003): - External networks is a possible source of knowledge.	- Not fully agreed upon in the results but at least party supported.

The results supported the earlier research (Table 3) into game project teams (Aleem et al. 2016a; Murphy-Hill et al. 2014, Musil et al. 2010) and knowledge management (McKenzie et al. 2023; Vegt & Bunderson, 2005) but also provided new insights and role of external networks (Fong, 2003) was not fully agreed upon.

According to the results, participants saw working in a multidisciplinary team mainly as positive. The multidisciplinary perspectives gave valuable insights that could also help to mitigate risks during the project. Any conflicts between disciplines or individuals could be easily solved through mediation sessions or more informal communication. It was also noted that often these conflicts were more due to personality differences and miscommunication rather than the different perspectives of the disciplines.

When building the teams both interests and expertise of team members should align with the needs of the project to ensure the success of the project. Producers also need to consider that specialists with unique or specialized expertise can be harder to recruit. Training new team members also needs the producer's support with practices like mentoring and giving them suitable training tasks.

Managing the communication and knowledge in the project team turned out to be an important part of the producers' responsibilities. It was also clearly connected to helping the team to connect with the game's vision and miscommunication was seen as a key risk for game projects. Participants also mentioned that remote or hybrid work meant that the producers needed to ensure adequate communication in digital channels.

Documentation was seen as an important way to share and store information during the projects. Documentation was also seen as a tool to transfer knowledge from previous projects to the next ones in the form of post-mortem documents. The goal of documentation was to minimize the amount of tacit knowledge while it seemed a certain amount of tacit knowledge was seen as inevitable.

## **7.2 Managing game development process and creative process**

Results supported the phases identified by earlier research (Aleem et al. 2016a; Aleem et al. 2016b; Marklund et al. 2019). and previous research of development methodology (Koutonen and Leppänen. 2013; O'Hagan et al. 2014; Politowski et al. 2021; Zhu and Wang (2019). New insights were made about multidisciplinary hybrid methodologies.

One common way to manage the development process that was not discussed by previous research was how all the participants used milestones or gates as important points in the process where the game was tested, and certain goals had to be met. These milestones were also usually synced up with the development process, for example if Scrum sprints could be used to see how many sprints there are before certain milestones.

One participant used Agile for other development but waterfall for art development which aligns especially well with Marklund et al. (2019) that Agile does not necessarily fit art development and this may cause dual systems. However, in this case, this was intentional, and systems were not conflicted which was something that Marklund et al. (2019) suspected might happen. According to the results as a scheduling tool, Agile does not work that well.

The relationship between knowledge management and game vision was clear in the results. Participants suggested that it was important to share the project vision often and make sure that

especially the discipline leads understood how their discipline work was needed to achieve that vision. Overall, there was no problem with the commitment of team members to the game vision. The results also emphasized the importance of giving space for creativity. According to the participants, the producer should mainly try to remove obstacles and give clear constraints for the project. It was also seen as beneficial to have some kind of brainstorming or ideation sessions for the project and sometimes for additional features where all the different project disciplines were represented.

Table 4 Development process conclusions

Topic	Previous research	Results
Development process management	Aleem et al. (2016a), Aleem et al. (2016b): - Game development phases. - Engine development, game test management, and programming practices as success factors.	- Project phases supported by the results: common phases identified by the participants. - Results do not discuss these specialized management areas as they fall more into the discipline leads rather than the producer.
	Marklund et al. (2019): - Conception as a separate phase - Too strict requirements do not fit flexible game development and can hinder creativity.	- Was also seen as separate phase by some participants - Even if Waterfall method was used there was need for flexibility and iteration.
	Murphy-Hill et al. (2014): - Poorly defined requirements lead to later challenges like technical debt.	- One participant emphasized the role of pre-production for decreasing risks. Especially key dependencies of the project should be analysed during this phase.
	O'Hagan et al. (2014), Politowski et al. (2021): - Agile and Scrum are common in game development, but Waterfall method still used alongside agile.	- Supported by the results and hybrid approaches also identified.
	Koutonen and Leppänen (2013) - Agile methods provided various benefits like iterative and incremental development process that helped to innovate, design and test ideas easily.	- Supported by the results, Agile methods were especially compatible with the iterative nature of game development as well as discovering the right gaming experience and "fun factor" of the game.
	Zhu and Wang (2019): - Model-Driven Development (MDD) could provide benefits but not widely adopted in the industry.	- MDD was not mentioned in the results. This supports the observation that it is not commonly adopted in the industry as a development method.
	Creative process management	Zackariasson, 2008: - Game companies operate with business-like operation.
Marklund et al. (2019): - Common knowledge base across disciplines needed.		- Supported by the results.

### 7.3 Managing game projects

Compared to earlier research (Schmalz et al. 2014) more formal risk management appeared to be common than expected. Otherwise, results aligned well with earlier research on risk and scope management, expanding the existing research with additional insights (Table 5).

Risk management could be done formally using risk management techniques like the Fishbone Diagram. Having documented risks can also help producers to communicate with different stakeholders. One of the internal risks mentioned in the results was key person risk, especially with very specialized team members. Bigger and more complex projects increased the probability of this risk according to one of the participants while knowledge management could be used to help mitigate this risk. Dependencies, IT infrastructure risks, and suboptimal team composition were also mentioned as an internal risk of the project. Funding was seen as a key external risk that needed to be mitigated using stakeholder management.

Table 5 Project management conclusions

Topic	Previous research	Results
Risk management	Aleem et al. (2016b): - Planning during pre-production reduces risks	- Supported by the results
	Schmalz et al. (2014): - Formal risk management practices are uncommon in game development	- Results somewhat contradict this. Two out five participants of this research had some form of formal risk management.
Scope and dependency management	Politowski et al. (2021), Ullman et al. (2022), Tschang (2005): - Feature creep as a possible risk or challenge	- Answered in the results with scope management
	Tschang (2005): - Production pipeline misalignment as a challenge	- Results answer this with scope management

According to the results, the producer must also ensure that the scope matches player expectations of the game's genre or idea. Feature cuts must also be intentional and made in places where resources are lacking rather than places where the cuts do not help with the schedule as there is no shortage of resources. The project's scale also usually increases throughout the project phases which must be handled by the producer.

Stakeholder management was not mentioned by previous research but was brought up as an important project management task in the results. Stakeholders like the upper management, publishers or customers usually determined the game's budget and funding and their commitment to the project

must be ensured by the producers by communication and meeting expectations that are agreed between the project team and the stakeholder.

#### **7.4 Limitations and suggestions for future research**

This research focused on the producer role perspective of managing multidisciplinary projects and this could mean that lacking the perspective of other team members like artists, designers, and programmers does not give a complete picture of the intricacies of video game projects and how they are and should be managed. This research was also done by interviewing Finnish video game project management professionals so their approach to project management may be different according to their culture.

Further research could broaden the perspective of this research by also gathering data from other roles in project teams as well as observing their work during projects to get more detailed information about how multidisciplinary project management works in practice. Research done in for example Japanese video game companies that often have their unique culture as well as longstanding and successful game development industry differing from Western gaming companies could give a different perspective on managing multidisciplinary video game projects.

#### **7.5 Contributions to practice**

Video game projects offer an interesting type of multidisciplinary project that combines software development, art, and business interests uniquely. This research offers a holistic perspective of video game project management that examines managing these projects from a multidisciplinary perspective. It can benefit project management professionals in both the video game industry and in general. While this research focuses on game development projects some insights probably apply to different kinds of projects like software projects or entertainment production projects too.

Knowledge management can help collaboration and create a unified multidisciplinary team. Project management professionals also need to facilitate creativity by giving different specialists the independence they need. Development methodologies should support multidisciplinary project work either by using Agile, Waterfall, or some kind of hybrid methodologies that consider the perspective of different disciplines. Risk documentation or visualization techniques also help to manage risks associated with the projects. Producers must also ensure the commitment of stakeholders to funding the project and having a well-defined development methodology, clear milestones, and documentation can make communicating with them easier.

## References

- Alan Wake II on yksi Suomen kalleimmista kulttuurituotteista, ja siitä voi tulla Remedyllle kultakaivos (2023) Helsingin Sanomat 27.10.2023 <<https://www.hs.fi/talous/art-2000009952209.html>>, retrieved 14.11.2023.
- Aleem, S. – Capretz, L. F. – Ahmed, F. (2016a) Critical Success Factors to Improve the Game Development Process from a Developer’s Perspective. *Journal of Computer Science and Technology*, Vol. 31(5), 925-948.
- Aleem, S. – Capretz, L. F. – Ahmed, F. (2016b) Game development software engineering process life cycle: a systematic review. *Journal of Software Engineering Research and Development*, Vol. 4(6), 1-30.
- Eisenhardt, K. M. (1989). Building theories from case study research. *Academy of management review*, Vol 14(4), 532-550.
- Eisenhardt, K. M. (2021). What is the Eisenhardt Method, really?. *Strategic Organization*, Vol 19(1), 147-160.
- Engström, H. – Marklund, B. B. – Backlund, P. – Toftedahl, M. (2018). Game development from a software and creative product perspective: A quantitative literature review approach. *Entertainment Computing*, Vol. 27, 10-22.
- Eriksson, P. – Kovalainen, A. (2008). *Qualitative methods in business research*. SAGE Publications Ltd, <https://doi.org/10.4135/9780857028044>
- Fong, P. S. (2003). Knowledge creation in multidisciplinary project teams: an empirical study of the processes and their dynamic interrelationships. *International journal of project management*, Vol. 21(7), 479-486.
- Gioia, D. A., Corley, K. G., & Hamilton, A. L. (2013). Seeking Qualitative Rigor in Inductive Research: Notes on the Gioia Methodology. *Organizational Research Methods*, Vol. 16(1), 15–31.
- Godoy, A. – Barbosa, E. F. (2010) Game-Scrum: An approach to agile game development. In *Proceedings of SBGames (2010)*, 292-295.
- Kanode, C. M. – Haddad, H. M. (2009). Software engineering challenges in game development. In *2009 Sixth International Conference on Information Technology: New Generations* (pp. 260-265). IEEE.
- Kasurinen, J. – Smolander, K. (2014). What do game developers test in their products?. In *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (pp. 1-10).

- Koutonen, J., & Leppänen, M. (2013). How are agile methods and practices deployed in video game development? A survey into Finnish game studios. In *Agile Processes in Software Engineering and Extreme Programming: 14th International Conference, XP 2013, Vienna, Austria, June 3-7, 2013. Proceedings 14* (pp. 135-149). Springer Berlin Heidelberg.
- Lester, J. N. – Cho, Y. – Lochmiller, C. R. (2020). Learning to do qualitative data analysis: A starting point. *Human resource development review*, Vol 19(1), 94-106.
- Marklund, B. – Engström, H. – Hellkvist, M. – Backlund, P. (2019) What Empirically Based Research Tells Us About Game Development. *The Computer Games Journal*, Vol 8, 179-198.
- McKenzie, Tim - Hoerrman, Simon - Lukosch, Stephan - Morales, Miguel. (2023) The Tower of Babel Game Developers Built: A Tertiary Literature Review into Why Multidisciplinary Teams Struggle to Collaborate in the Games Industry. Available at SSRN: <https://ssrn.com/abstract=4669233> or <http://dx.doi.org/10.2139/ssrn.4669233>, retrieved 3.10.2024
- Musil, J. – Schweda, A. – Winkler, D. – Biffel, S. (2010) Improving video game development: Facilitating heterogeneous team collaboration through flexible software processes. In *Proceedings of the 17th European Systems and Software Process Improvement and Innovation Conference (EuroSPI 2010)*, 83-94.
- Murphy-Hill, E. – Zimmermann, T. – Nagappan, N. (2014). Cowboys, ankle sprains, and keepers of quality: How is video game development different from software development?. In *Proceedings of the 36th International Conference on Software Engineering* (pp. 1-11).
- Pan, M. – Chandrasekaran, A. – Hill, J. – Rungtusanatham, M. (2022). Multidisciplinary R&D project success in small firms: The role of multiproject status and project management experience. *Production and Operations Management*, Vol 31(7), 2806-2821.
- Osborne O'Hagan, A. – Coleman, G. – O'Connor, R. V. (2014). Software development processes for games: A systematic literature review. In *Systems, Software and Services Process Improvement: 21st European Conference, EuroSPI 2014, Luxembourg, June 25-27, 2014. Proceedings 21* (pp. 182-193). Springer Berlin Heidelberg.
- Petrillo, F. – Pimenta, M. (2010) Is agility out there? Agile practices in game development. *Teoksessa: Proceedings of the 28th ACM International Conference on Design of Communication*, 9-15.
- Perspectives and insights: Global Entertainment and Media Outlook 2023–2027.  
PricewaterhouseCoopers International Limited

<<https://www.pwc.com/gx/en/industries/tmt/media/outlook/insights-and-perspectives.html>>, retrieved 14.11.2023.

- Politowski, C. – Petrillo, F. – Ullmann, G. C. – Guéhéneuc, Y. G. (2021). Game industry problems: An extensive analysis of the gray literature. *Information and Software Technology*, Vol 134, 106538.
- Seidel, V. P. – Fixson, S. K. (2013). Adopting design thinking in novice multidisciplinary teams: The application and limits of design methods and reflexive practices. *Journal of Product Innovation Management*, Vol 30, 19-33.
- Schmalz, M. – Finn, A. – Taylor, H. (2014). Risk management in video game development projects. In 2014 47th Hawaii International Conference on System Sciences (pp. 4325-4334). IEEE.
- The Game Industry Of Finland REPORT 2022. Neogames <<https://neogames.fi/finnish-game-industry-report-2022/>>, retrieved 14.11.2023.
- Tschang, F. T. (2005) Videogames as interactive experiential products and their manner of development. *International Journal of Innovation Management*, Vol. 9(01), 103-131.
- Ullmann, G. C. – Politowski, C. – Guéhéneuc, Y. G. – Petrillo, F. – Montandon, J. E. (2022). Video game project management anti-patterns. In Proceedings of the 6th International ICSE Workshop on Games and Software Engineering: Engineering Fun, Inspiration, and Motivation (pp. 9-15).
- Van Der Vegt, G. S. – Bunderson, J. S. (2005). Learning and performance in multidisciplinary teams: The importance of collective team identification. *Academy of management Journal*, Vol. 48(3), 532-547.
- Zackariasson, P. – Martin, W. – Wilson, T. L. (2006) Management of Creativity in Video Game Development: A Case Study. *Services Marketing Quarterly*, Vol. 27(4), 73- 97.
- Zhu, M. – Wang, A. I. (2019). Model-driven game development: A literature review. *ACM Computing Surveys (CSUR)*, Vol. 52(6), 1-32.

## Appendices

### Appendix 1. Data structure

3 <sup>rd</sup> order code	2 <sup>nd</sup> order code	1 <sup>st</sup> order codes
<b>Managing multidisciplinary project teams</b>	Knowledge management	Documentation, Remote work, Learning, Sharing knowledge, Communication in teams
	Managing multidisciplinary teams	Team configuration, benefits of multidisciplinary teams, conflicts in multidisciplinary teams, workload management, coordinating the teams, skill requirements in game teams, organizational structure, onboarding new team members
<b>Managing the development process</b>	Managing the game development process	Agile development, Waterfall method, Hybrid methodologies, Software development methodologies, Project phases, Game maintenance, Project management practices, Project scaling, Milestones
	Managing creative process	Ideation phase and brainstorming, Game visions management, Game experience management, Managing creativity
<b>Managing multidisciplinary projects</b>	Scope management	Scope management, Dependency management, Budgeting
	Risk management	Formal risk management practices, Risk mitigation, Testing
	Stakeholder management	Stakeholder communication, Stakeholder management practices, Publisher's role, Cultural differences
<b>Background</b>	Producer role	Work title, Different type of producers, Producer's responsibilities
	Former experience	Education, Former occupations, Platform experience, Project size experience, Previous projects
	Nature of game projects	Game projects as entertainment, Game projects as software

## **Appendix 2. Semi-structured interview questions (in English)**

Interview questions divided into themes with main questions with optional questions to help guide the conversation if needed.

### **Interview structure:**

#### **General background questions:**

- Tell me about your background. How did you end up in project management role in the game industry and what are your responsibilities?
  - What kind of education have you received for this role?
  - What kind of game projects have you been involved in, and what size companies have you worked for in the game industry?
  - What title(s) have you worked under?

#### **Managing multidisciplinary teams:**

- What kinds of specialists have made up the teams you've been a part of?
  - How do you think that has worked?
  - What kind of specialists do you think such teams should consist of?
  - Do you feel there is a need for any role or expertise that is typically missing from project teams?
  - How do you think diversity within the team affects the team's performance?
  - How do you perceive your own role within the project?
- Do you think professionals from different fields have different priorities, and does this lead to conflicts?
  - How are these conflicts resolved?

#### **Game projects and Scope management:**

- What kinds of production phases have there been in the projects you've worked on?
  - What do those phases include?
  - How is the scope of a project defined and managed?
  - How has the progress of projects been managed and monitored?

#### **Creative process:**

- How has creativity been managed in the projects you've been part of?
  - How have you ensured that the desired player experience was achieved in the project?

- How are different team members involved in ideation within the team?
- How was a shared vision of the project's goals created within the team?

**Development methods:**

- What kinds of development methodologies or frameworks have you used in game projects?
  - Do you, for example, use agile development methods?
  - What principles guide software development in the teams you've been part of?
  - How have different multidisciplinary tasks, such as programming and creation of various visual assets, been integrated?
  - How do you avoid production bottlenecks where progress is blocked because a previous phase has not been completed?
  - How is an even distribution of workload ensured?
- What do you think the role of testing is in game projects?

**Knowledge management:**

- How was information shared within the team, especially between different roles like artists and programmers?
  - How do you think shared learning in projects can be ensured?
  - How can knowledge and experience be transferred from one project to the next?
  - How do team members' external contacts or networks influence access to external knowledge?

**Risk management:**

- How do you try to manage risks in game projects?
  - Do you feel that the multidisciplinary nature of projects affects the risks?
  - How do you avoid feature creep?
  - What do you see as the biggest external and internal risks for teams and projects?

**In conclusion:**

- Did anything come to mind during the interview—regarding for example the multidisciplinary nature of game projects or project management—that you would like to mention?