OULUN YLIOPISTO
UNIVERSITY of OULU

A Systematic Mapping Study of Crunch Time in Video Game Development

University of Oulu
Faculty of Information Technology and Electrical Engineering / Degree Programme of Information Processing Science
Master's Thesis
Jonne Niemelä
5.6.2021

Abstract

Crunch time means heavy overtime work, which usually happens before deadlines when development is behind schedule. It is common in game development and causes stress and social harm to developers. The purpose of this thesis was to study how crunch time is discussed in scientific literature by finding the trends and gaps using a systematic mapping study research method.

The main research question was: How is crunch time in video game development discussed in scientific literature? Three assisting research questions were used to help with answering the main research question: How are the causes for crunch time in video game development discussed in the literature? How are the effects for crunch time in video game development discussed in the literature? How are the solutions for crunch time in video game development discussed in the literature?

A search string was defined along with inclusion and exclusion criteria in a way that the resulting papers could best be used to answer the research questions. A total of 36 relevant, primary studies were included for this study after conducting the search on Google Scholar and applying the inclusion and exclusion criteria. The papers were categorised based on their contributions to the causes, effects and solutions related to crunch time. These papers were used as the source material for this research.

It was found that the number of studies is increasing and the most common contribution of these studies is presenting the causes of crunch time. The most common causes presented were cultural, planning and process, and structural causes. Health and social effects were the most common effects. Process changes and no-crunch policies were the most common types of solutions to crunch time.

The main contribution of this study was presenting an overview of how crunch time is discussed in scientific literature. It can be used by researchers to determine what kind of research might be necessary. The study could also be useful for game developers and managers in deciding what actions to take to avoid crunch time.

This study found that there are gaps in the literature regarding solutions to crunch time, but often the causes seem to imply solutions. This study suggested that more research into feature creep and structural changes to the game development companies could be useful for learning how to reduce crunch time.

Keywords
video game development, crunch

Supervisor
Ph.D. University Lecturer Raija Halonen

Foreword

I would like to thank Adjunct Professor Raija Halonen for her useful advice and support. I would also like to thank Ph.D. Mikko Rajanen for his feedback.

Jonne Niemelä

Oulu, June 5th, 2021

Contents

1. Introduction

“Crunch time” is used to refer to long periods of overtime work that can last for weeks or months. It usually happens before a deadline for project delivery. (Petrillo et al., 2009.) These periods happen in game development more often than in other software engineering companies (Gershenfeld et al., 2003, as cited in Petrillo et al., 2009).

The purpose of this study was to analyse how the causes, effects and solutions of crunch time in game development is discussed in literature. The motivation was to help game developers find the answers for avoiding the crunch and also to help other researchers in deciding what type of research related to crunch time might still be necessary.

Game development projects often face situations where the developers realise that the release date or another deadline is coming up sooner than the work is ready. To avoid delays, game developers frequently end up in a crunch situation, where they have to work long hours to finish a project before a deadline. Crunching causes stress and social problems for game developers and their families. Crunching for over two weeks seems to decrease the game quality, but it allows game companies to release their product on time. Developers often lose their passion for developing video games because of crunch and end up leaving the industry. (Edholm et al., 2017.)

Crunching is commonly seen as a problem in scheduling (Weststar & Kumar, 2020). However, the problem is often that features are added during development without adjusting the schedule (Musil et al., 2010; Edholm et al., 2017). Crunching often happens voluntarily by the developers (Weststar & Kumar, 2020). Game developers can feel pressure to crunch for systematic or social reasons. (Peticca-Harris et al., 2015.) Developers are also often very passionate about the game they are developing and choose to crunch (Cote & Harris, 2021; Edholm et al., 2017).

This study was done as a systematic mapping study to present an overview of primary studies related to crunch time in game development. This paper also presents the most common causes, effects and solutions in one mapping study for the benefit of game developers, managers and quality assurance workers, and also identifies research areas that are lacking and should possibly be research further.

The main research question for this study was:

How is crunch time in video game development discussed in scientific literature?

The following questions were used to support the main research question:

How are the causes for crunch time in video game development discussed in the literature?

How are the effects for crunch time in video game development discussed in the literature?

How are the solutions for crunch time in video game development discussed in the literature?

Each of these questions imply the identification of gaps and trends in the literature regarding the question. Answering the sub-questions made it possible to present an overview of how the topic of crunch was discussed in scientific literature.

The structure of this thesis is as follows. Chapter 2 is the literature review, where prior scientific knowledge on the topic is presented. In Chapter 3, the research method is presented and justified through a review on the literature regarding systematic mapping studies. In Chapter 4, the results of the study data collection are presented and analysed. Chapter 5 is for the discussion on the results of the study and the results are discussed in relation to the prior literature. Chapter 6 has the conclusions of the study.

2. Related work

In this chapter, the prior research is presented. The sub-chapters are split into general information about crunch time, the effects of crunch time, and the reasons that lead to crunch time.

2.1 Crunch time

During crunch time, employees have to work extremely long days for weeks or months. Crunches are meant for meeting a deadline while avoiding the game release being bad or buggy. (Edholm et al., 2017.) Crunching also happens in other software engineering companies, but it is more common in game development (Gershenfeld et al., 2003, as cited in Petrillo et al., 2009).

Crunch time is not only a problem in software and game development, but according to Robinson (2005) it has been mostly stopped in other industries because it is not productive. Overtime hours have been demonstrated in construction projects to increase productivity in the short term, with the “short term” meaning at most eight weeks. Having 60-hour work weeks for longer than that seems to drop productivity to levels where it would have been better to use 40-hour work weeks from the beginning. The workers’ production also temporarily falls to a lower than normal level after returning to normal working hours from a crunch time. (Robinson, 2005.)

According to Petrillo et al. (2009), working days of over 12 hours with 6 or 7 work days in a week, are common during crunch time. In 2019, most employees reported working between 50 to 69 hours per week during crunch time and 13% reported working for over 70 hours per week in the Developer Satisfaction Survey 2019. Thirty four percent of employees working overtime or in crunch time did not receive additional compensation. From those who did, only eight percent were paid for the overtime and the rest received perks such as meals or time off in the future. (Weststar et al., 2019.)

Crunches often happen before a scheduled release but it can sometimes even last for the majority of the development project. Sometimes developers do small crunches in the middle of the game development project to reach certain milestones. Sometimes studios believe they are not crunching, but end up spending time working late anyway. Crunch seems to be very common, although the amount varies by company. Small game development studios seem to crunch more often than micro- and medium-sized studios. Companies that use agile methods seem to crunch less than those that don’t. (Edholm et al., 2017.) Musil et al. (2010) found crunch time to be the most frequent in Massively Multiplayer Online game development and least frequent in mobile game development.

Borg et al. (2019) considered game jams to be a period of voluntary crunch time, where the development time is very limited. The developers in game jams sometimes intentionally reduce the scope of the project so that the game has a higher quality at release and it can reduce the intensity of the necessary crunching. They recognised that, while game jams can propose solutions to crunching, they can also further normalise the problematic culture of crunching. They suggested that game jam organisers could raise awareness of the problems with crunching.

According to Weststar et al. (2019), based on the Developer Satisfaction Survey 2019, the amount of crunch time is decreasing and the expectation to crunch is decreasing also. In the 2014 survey according to Edwards et al. (2014), 19% of respondents had not crunched in two years and 45% had at least two crunch periods within two years. In 2017, 51% of respondents said that their work involved crunch time. In 2019, this number was only 41%. Extended hours that the respondents did not consider crunching was reported by 44% in 2017 and 35% in 2019. In 2019, 36% of respondents reported having been in crunch time more than twice in two years. (Weststar et al., 2019.) Kerr (2019) noted that most of the respondents in the Developer Satisfaction Survey from 2016, who responded that they crunch, work in large companies, despite those companies possibly having better processes.

2.2 Effects of crunch time

Game developers often end up burnt out from crunching and it affects their relationships outside work. Many developers end up hating their work because of crunching. Crunching can have a negative effect on the game itself because of the increased stress. Having a vacation after a crunch seems to only reduce stress temporarily because people associate work with stress without considering recovery daily. Long worktimes seem to eventually lower the team’s work morale as well. (Edholm et al., 2017.)

Crunching might limit the diversity of workers in the game development industry because it is harder for people with families to accept the difficult work-life balance during crunches compared to recent graduates. (Borg et al., 2019). Weststar and Legault (2018) argued that video game industry practices create a bigger barrier of entry to women because the long hours and project-based work have been documented to pose bigger challenges to women. Long hours and required mobility tend to be more accepted by young and unattached males.

The main positive thing about crunching seems to be that the games can be released on time. Some developers see crunching as a positive thing because the developers may feel that they are doing their best as a team. (Edholm et al., 2017.) Gershenfeld et al. (2003), as cited in Petrillo et al. (2009), suggested that crunching can be good for ambitious people without families, although they also say that people work more efficiently with eight-hour work days.

According Edholm et al. (2017), crunches under two weeks can increase the quality of the game, but the crunch time still causes stress and frustration. All the other types of crunch had several problems with both the quality of the game and actually releasing on time. Based on their interviews, it was common for overworked and tired workers to create more bugs even when bug fixing. In all crunches longer than two weeks, the product quality was overall worse, but they managed to add more features and generally managed to release on time. Targeted mini crunches allowed two game companies in the study to release the game a month early.

2.3 Reasons for crunching

This chapter is split into sub-chapters based on the different types of reasons for crunching found in the literature.

2.3.1 Management and planning reasons

Unclear scope, feature creep and deadlines were seen as the three main reasons for crunch by Edholm et al. (2017), where they interviewed game developers from four game development studios. One of their interviewees noted that without deadlines there wouldn’t be any crunch at all and this implies that crunching is a planning and management problem. According to the Developer Satisfaction Survey 2019 by Weststar and Kumar (2020), 62% of respondents considered crunch to be a failure of scheduling. Unclear expectations were seen as a reason for 30% of the respondents. Not having enough people on time was a reason given by 30% of respondents and the lack of experience of the managers was a reason given by 23% of respondents.

According to Legault and Weststar (2015), crunch is often a result of a failure in risk management. Developers lose a lot of time to technical problems that weren't expected during planning and the difficulties of many tasks are often underestimated. The project-based structure that is common in video game development does not generally allow for moving deadlines due to unexpected problems. It is often difficult to find out who to blame for failures in schedule, which might lead to thinking that crunching is just something to expect. More experienced developers tend to blame the management rather than taking crunch for granted. Some developers blame their own lack of experience, but recognise that they could avoid the crunch by planning better.

Edholm et al. (2017) concluded crunch time to be an organisational issue more than technical. In fact, technical issues have been decreasing for game developers (Politowski, 2021). However, switching technologies during development can cause unexpected technical challenges, resulting in crunch time (Cote & Harris 2021). Musil et al. (2010) considered crunch time and scope creep to result from workflow and integration issues.

2.3.2 Scope creep

Scope creep (or feature creep) means adding features to video games during development without adjusting the development time (Edholm et al., 2017). Kanode and Haddad (2009) considered some scope creep to be necessary because the intention is usually to make the game more fun, which is important for video games. However, it can cause problems with time management if it's not managed properly. Politowski et al. (2021) found that feature creep has been decreasing, but is still common.

Musil et al. (2010) found a correlation between crunch time and scope creep. They suggest that scope creep might be a result of bad requirements elicitation and lacking prototyping. According to Legault and Weststar (2015), it is uncommon that there is a change in contract terms even if the customer's requests change. This means that the scope can increase during development due to changing customer requirements without allowing the game development studio to adjust the schedule.

2.3.3 Social and cultural reasons

There seems to be some level of acceptance and expectation for crunching in the video games industry (Kerr, 2011). Often crunching is done because a developer is in the right mood for working. This can cause a group mentality, where other developers feel that they should also stay late for work. Developers are also often passionate about making the game as good as they possibly can before release, so they prioritise the quality of the game over their own welfare. However, crunching can cause developers to lose their passion. (Edholm et al., 2017.) Kasurinen et al. (2017) found crunch time to be a minor concern among video game developers.

According to the Developer Satisfaction Survey 2019 by Weststar and Kumar (2020), between 2015 and 2019, crunch had become less expected as a normal part of video game development, but 42% of respondents still expected to crunch. Crunching was most expected by quality assurance workers (59%) and least by managers (37%). Out of the respondents who were developers, 44% said yes to crunch being expected as a normal part of the job. Freelancer or contract workers were most likely to consider crunching as expected (56%), while self-employed respondents were the least likely (30%). Out of employed respondents, 42% thought crunch time was expected. Thirty percent of respondents saw crunch occurring because of people doing it voluntarily.

Extreme working conditions are commonly seen as justified and required to release successful and innovative games on time. One justification is that there are many other jobs with bad working conditions and long hours. (Peticca-Harris et al., 2015.) According to Weststar (2012), crunch is generally perceived by game development workers as a temporary thing and not completely the fault of the management. Crunch time seems to be taken as a “fact of life” in the game industry despite game developers quitting frequently. Game developers usually accept having to work crunch time temporarily for a project or two, but they leave the company afterwards. (Gershenfeld et al., 2003, as cited in Petrillo et al., 2009.)

Independent developers might justify their habit of crunching by the ownership of the company and the game they are developing because they gain the benefits from the potential success of the game. However, it still has the same harms on personal life from crunching as working for a large studio. (Keogh, 2021.) Developers usually recognise crunching as a problem, but try to reduce it instead of eliminating it completely. A nostalgia for hobbyist development keeps many professional game developers from processes and formal business structures that could help avoid crunch. Developers often see corporate policies as harmful to creativity. (Cote & Harris, 2021.)

Peticca-Harris et al. (2015) suggested that neo-normative control mechanisms result in extreme work practices such as the crunch time to meet deadlines. These same mechanisms also make it difficult for the developers to see them as extreme. According to the paper there doesn’t seem to be much resistance to working conditions in the video game industry, except for the developers quitting completely. People personally close to the developers give some resistance to working conditions. They suggest that project-based work causes game developers to work long hours for the sake of their portfolios and reputations, which help with their employability.

Cote and Harris (2021b) found that developers commonly see game development as unmanageable. That perception often causes game developers to not plan sufficiently and crunching becomes expected. Another reason for underplanning is that developers expect a heavy influence from publishers so they feel that they aren’t in full control of the process anyway. They also generally prioritise meeting the publisher’s expectations over avoiding crunch. The paper also found that it was common for employees to voluntarily work late nights or overnight and that was seen as the employee being a passionate developer. According to Borg et al. (2019), there seems to be a stigma of missing internal deadlines between game developers, which pushes developers to crunching. Because of this, crunching does not always happen at the end of a project or near the release date.

According to Chung and Kwon (2020), game developers tend to not be very interested in collective action against their employers because the cultural ideas of individualism, professionalism and meritocracy are common in game development and the workers see themselves as collectively creating games with their employers. This view is supported by Legault and Weststar (2015), who stated that the meritocratic nature of game developers conflicts with the egalitarian ideology behind unions. The system of hiring video game developers is also largely based on recent merit, which means that the meritocratic culture is supported by the system, creating a systematic obstacle for collective action.

There seems to be a difference between what the developers expect from working at a game company and what the reality is and the grievances and frustrations build up slowly. In South Korea, the employers were able to require long work hours without paying extra for overtime work, which resulted in works days of over 12 hours being common. The workers are often required to deliver an update frequently, which means that crunching can sometimes become the norm. The game developers seemed to enjoy their work more when the working hours were reasonable and the management was horizontal, but before unionising the employers ended up pushing more working hours. (Chung & Kwon, 2020.)

Weststar (2012) argued that bad working conditions alone aren't enough to motivate game development workers to take action towards change because the values that lead to crunch are legitimised in the community and industry. Taking collective action could be more likely if the management were perceived to be violating the values of the community. According to Legault and Weststar (2015), there is a high coordination cost to collective action, meaning that it takes more effort initially to take action collectively rather than as individuals. Developers often feel that they can influence the organisation individually. According to the Developer Satisfaction Survey 2019 by Weststar and Kumar (2020), six percent of game developers said that they are union members. However, most of the respondents said that they would vote in favour of a union. The support was higher for unions across industries instead of by work place.

2.4 Solutions to crunching

Edholm et al. (2017), suggested using the best practices of agile development to avoid unrealistic schedules and feature creep. However, in their research even companies using agile methods experienced crunch time, possibly because of bad implementation of the practices and the culture deciding the development pace. According to Keith (2010), the agile practice of early quality assurance and fixing bugs as soon as they are found helps with avoiding a bug fixing crunch. Musil et al. (2010) suggested that a process model that considers the nature of game development (multi-disciplinary, focus on the product and non-functional requirements) could help with the changes in workload. They also state that game development would generally benefit from best practices in software engineering, despite the differences compared to traditional software engineering.

One solution to feature creep is to remove less important features whenever a more important feature has to be added if possible (Edholm et al., 2017). However, Kanode and Haddad (2009) considered it important to define the requirements during pre-production through the use of prototyping. A well-defined scope reduces the need for feature creep, and if the schedule takes the potential for feature creep into consideration, scheduling problems and the resulting crunch can be avoided. Prototyping during pre-production can help with defining the requirements and the scope of the game, while discovering what makes the game fun through experimentation. This was supported by Musil et al. (2010), who directly connected requirements and lack of prototyping to scope creep.

Risk management is useful for avoiding crunch time because certain game development problems are very common and crunch is used as the last resort to avoid missing a deadline (Legault & Weststar, 2015). Kanode and Haddad (2009) suggested considering feature creep in risk management because some level of it seems inevitable. A postmortem by 11 Bit studios mentioned that they added extra time for every milestone to avoid falling behind on schedule. This was because missing milestones was common and they wanted to avoid crunch time. (Politowski et al., 2016.).

Keeping crunch time to under two weeks was found to be the least harmful by Edholm et al. (2017) and actually had some benefits to the quality at the cost of stress. They suggested to aim for crunching for at most two weeks if it is necessary at all. Cote and Harris (2021a) also recognised that a short, planned crunch can work, but they also noted, however, that even a “good”, scheduled crunch can end up lasting longer than intended.

Politowski et al. (2021) suggested that senior developers should document their work to make it easier for other developers to estimate time needed for completing a task. They also recommend long pre-production phases for research purposes, which could help with making better time estimations, and as a result, avoiding crunch time.

Weststar (2012) suggested that the game development community could accept the idea that bad working conditions forced by the executives is harmful to the goal of making a great game. This would direct the passion of game developers to taking collective action towards improving their working conditions instead of accepting crunch. Social media discussions allow game developers and others to raise awareness of bad working conditions in the video game industry. Sharing experiences could affect the culture and acceptance of crunch time along with other harmful practices in game development. (Peticca-Harris et al., 2015.) A solution to the social issue of crunch could be instituting pre-defined working hours. Because developers often feel personally motivated to work late, they may cause social pressure for other developers to do the same. Making it a rule that working is only allowed on certain hours could thus reduce voluntary crunching. (Cote & Harris, 2021.)

3. Research Method

This chapter describes the research method. The research question is presented along with the explanation of how the research method is applied to this study.

3.1 The purpose of a systematic mapping study

Systematic mapping studies are meant for researching what the current situation is in primary studies related to a research area. It is done through classifying different studies based on topic and how they are published. The frequencies of these studies are then counted to create an overview of the field and it can be used to answer research questions related to the field of study. Generally, the goal is to find what kind of research is missing from a research area and to recognise trends. (Petersen, et al., 2008.)

Systematic mapping studies are useful for areas of study that are broad or lacking in evidence. Sometimes, a systematic mapping study can reveal that a systematic literature review could be appropriate for a research area. The results of a mapping study can be used to decide what kind of research to do in the future. (Kitchenham & Charters, 2007.)

This research method was chosen because crunch time as a research area seemed to not have a lot of evidence, but enough to recognise the trends and gaps using this method. This method seemed to be a good way to investigate how the literature presents crunch time at the moment and what kind of research could and should still be done.

3.2 Systematic mapping studies compared to systematic literature reviews

The similarity between a systematic mapping study and a systematic literature review is in the search of primary studies and the selection methods. The difference is in how the data is analysed. A systematic literature study uses the found evidence to answer specific questions about a topic, while a systematic mapping study focuses on the trends. (Petersen, et al., 2015.) A systematic mapping study generally requires less effort compared to a systematic literature study while presenting a good overview of the evidence related to the research area. It can highlight a lack of evidence in certain problems areas. (Kitchenham & Charters, 2007.)

Systematic mapping studies use less focused search terms than systematic literature studies to get a broader coverage of the primary studies. The data extraction only has to answer the broad research questions and the main purpose of the data extraction is to classify the papers. The distribution of the studies based on classification can be presented visually. (Kitchenham & Charters, 2007.)

A systematic mapping study was chosen over a systematic literature review for this study because it appeared, based on the prior research, that there are some gaps in research into crunch time in video game development. It seemed to be more useful to study the topic broadly for this study, instead of answering very specific questions. A systematic mapping study can still present an overview of the evidence as stated by Kitchenham & Charters (2007), so it can still have some of the benefits of a systematic literature review.

3.3 The systematic mapping study process

Bailey et al. (2007) explained that mapping studies focus on the following stages of a literature review: identification of research, selection of primary studies and study quality assessment. Their mapping study included only the most recent paper of ones that report on the same primary study, but if there were several studies in one paper, those were included separately. They excluded studies that were lacking in the reporting of findings. They defined the classifications and calculated how many times each type of study appeared in the literature. The classifications and frequencies were presented visually as tables. They recognised which publication types, research topics and research methods appeared most commonly in the literature.

According to Petersen et al. (2008), the research questions should be defined based on the goal of the study. They suggested that the search for studies can be done either through specific searches or manually. The criteria for inclusion and exclusion should be designed to answer the research questions. Keywords can be taken from the abstract to understand the contribution of the study, but also from the introduction or conclusions. These keywords can then be used to form classifications and the frequencies of each category can be analysed. (Petersen et al., 2008).

According to Petersen et al. (2015), quality assessment is not necessary in a systematic mapping study, but some systematic mapping studies have it. The full text of a study can be read if inclusion or exclusion is not clear. Categories can be created by recognising and merging the keywords or concepts found in the selected studies, which can be based on the abstract alone if it is of sufficient quality. (Petersen et al., 2015.)

Snowball sampling can be used in addition to the systematic search to find more studies (Petersen et al., 2015). Backward snowballing means that the researcher goes through the list of references in selected papers to find more relevant papers that did not show up in the search. Forward snowballing means finding papers that use the selected papers as references. Snowballing should be done only after the inclusion is certain to avoid having to remove papers later. (Wohlin, 2014.)

3.4 Research question

The main research question for this study was:

How is crunch time in video game development discussed in scientific literature?

The research question was supported by answering three assisting research questions:

> *How are the causes for crunch time in video game development discussed in the literature?*
>
> *How are the effects for crunch time in video game development discussed in the literature?*
>
> *How are the solutions for crunch time in video game development discussed in the literature?*

Each of these questions imply the identification of gaps and trends in the literature regarding the question. Answering the sub-questions made it possible to present an overview of how the topic of crunch was discussed in scientific literature.

3.5 The search

Google Scholar was used for the search because of the appropriate amount of results it provided. Also, both Scopus and Web Of Science only brought up results that were already found on Google Scholar. Yasin et al. (2020) showed that only very few studies appear in other databases that do not also appear in Google Scholar. It has also been used as the only search engine in published studies, although it was recognised as a problem that the results change with time (Wohlin, 2014).

The search string for this study was chosen to be broad to find as much information as possible on the research topic. The search was not limited by publication year and citations were excluded from the search.

The search string for Google Scholar was as follows:

“game development” AND “crunch”.

The word “crunch” was included as a necessary keyword to make sure that the study relates the recognised problems or solutions discussed to crunch. This means that possibly relevant studies were excluded from the search, but the purpose of this study was to specifically research how crunch is discussed in the literature. Alternative wordings for “game development” either did not change the results much or included too many studies that were not relevant.

3.6 Study selection

The initial search brought up 961 results on Google Scholar. Inclusion and exclusion criteria were defined to select the papers that are appropriate for the systematic mapping study.

The following inclusion criteria was used for this study:

1. Is in English.
2. Is relevant to crunch in video game development.
3. Is a primary study.

The following exclusion criteria was used for this study:

1. Is a duplicate.
2. Is not available to read.
3. Doesn’t answer any of the supporting research questions.
4. Crunch is only mentioned in passing.

Only English studies were included to avoid wrong results based on a mistranslation, but also because the search terms being in English would have likely excluded many of the results in other languages. Only primary studies were included because secondary studies would have skewed the results by using the same data that was already potentially gathered from another study. “Is not available to read” meant that the papers had to be accessible through the university grant that was used for this research.

The papers were examined one-by-one and the inclusion was done according to these criteria based on the title, abstract or full text depending on when it became clear if it should be included or excluded. Student papers and grey literature were included in the results because only 17 papers were published in scientific journals or conferences. After applying the inclusion and exclusion criteria, 36 papers were identified.

3.7 Keywording, data extraction and mapping

The following data was extracted from each paper, and they were saved using a spreadsheet software:

1. Year
2. Publication type
3. Causes (keywords)
4. Effects (keywords)
5. Solutions (keywords)
6. Author(s)
7. Year

Keywords relevant to the causes, effects and solutions of crunch time were identified. The keywords were based on concepts and did not necessarily appear in the same exact form in each paper. The purpose of the keywords was to use them for grouping papers based on their contributions. The keywords are presented in simplified format in Appendix B and a more raw format is presented in Appendix C. The other data is presented in Appendix A. The identified keywords were used to form classifications, which are presented in Chapter 4.

4. Results

In this chapter the results of the systematic mapping study are presented. At first, general information about the studies is presented and the sub-chapters are presented based on the sub-questions of the research.

4.1 Crunch time studies

A total of 36 studies that fit the inclusion and exclusion criteria were identified.

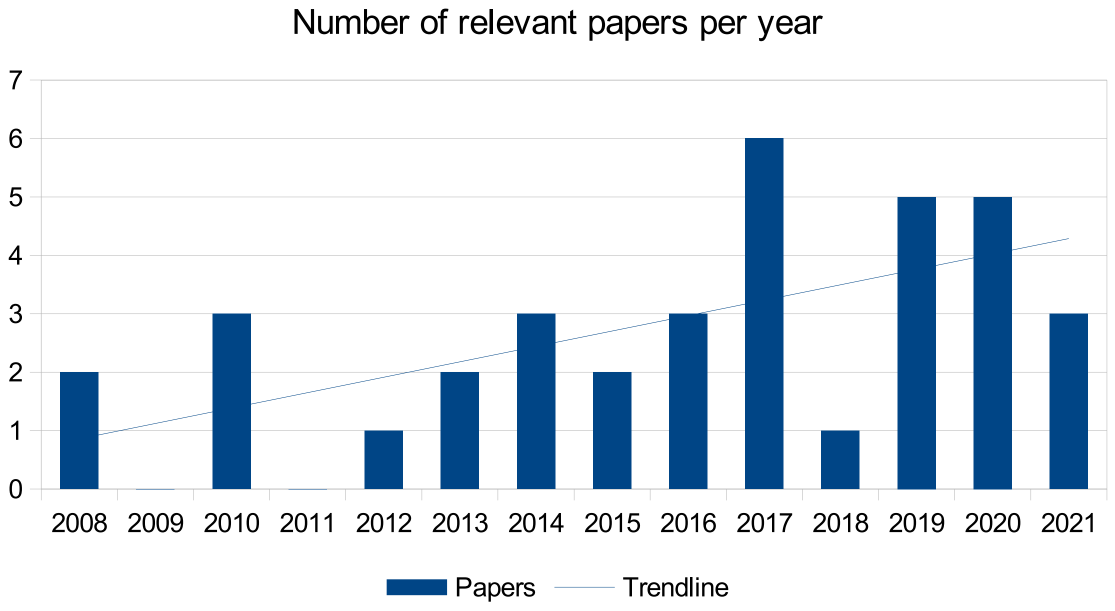

Figure 1. Number of relevant papers per year.

Figure 1 shows that there have been more studies related to crunch time in recent years compared to previous years and the trend in the number of papers is upward. The first paper related to crunch time was released in 2008 and the newest studies are from 2021. There were no papers related to crunch time published in 2009 and 2011. The most papers in a year was in 2017, when 6 papers were published. There was only one paper from 2018, despite the previous and following years having many more.

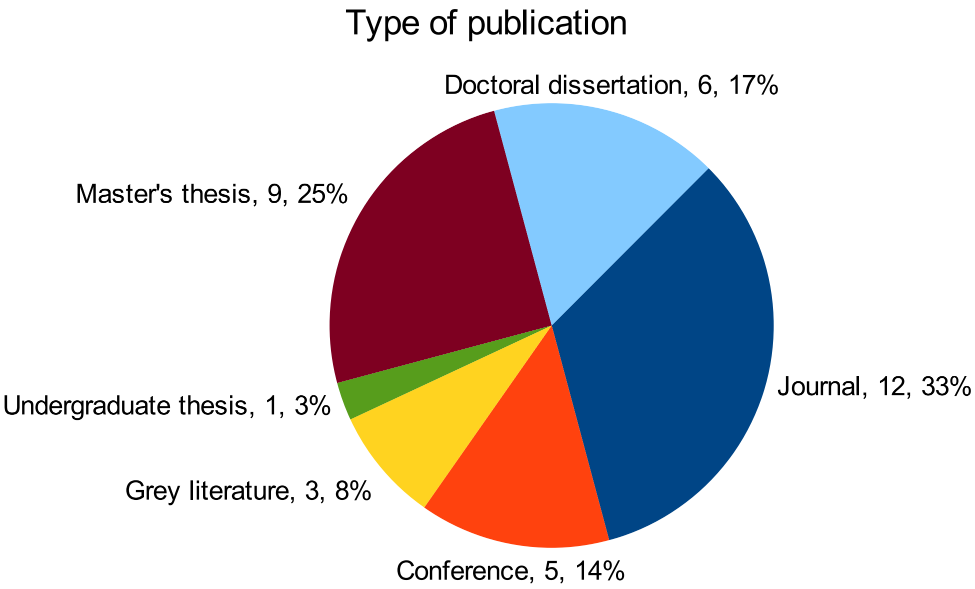

Figure 2. Distribution of papers by type of publication.

From Figure 2 it can be seen that slightly fewer than half of the papers (47%) were published in a conference (14%) or a journal (33%). The total number of scientific, published research was only 17 papers, which is why student papers and grey literature were included in this systematic mapping study.

Almost half (44%) of the papers were student papers, including master’s theses (25%), doctoral dissertations (17%) and one undergraduate thesis (3%). This shows that there is a relatively high interest towards crunch time in students. The studies in Figure 2 that are presented as “grey literature” (8%) were all IGDA (International Game Developers Association) reports based on surveys. These reports fit the inclusion and exclusion criteria, but were not published in scientific publications.

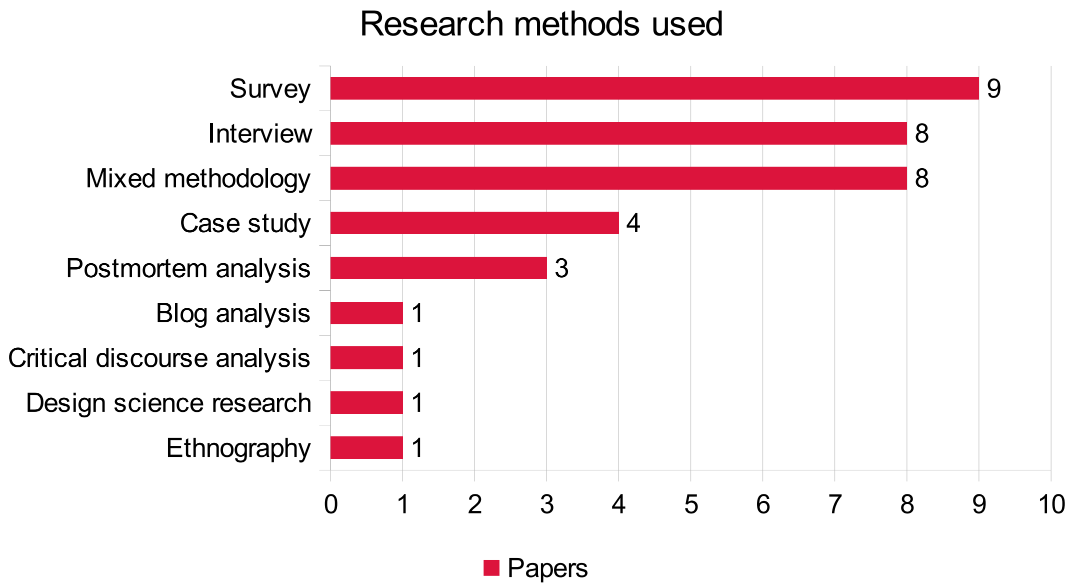

Figure 3. Number of papers by research method.

Figure 3 shows the research methods used. Survey (25%) was the most common research method followed by interview (22%) and mixed methodology (22%). Mixed methodology studies were generally studies that used both interviews and surveys, but also other methods. Case studies (11%) and postmortem analyses (8%) were relatively common. Blog analysis, critical discourse analysis, design science research and ethnography each appeared once (3%) in the relevant studies.

From the research methods it can be seen that most of the contributions (recognised causes, effects and solutions) are based on self-reports from developers, management or from the researchers observing development teams. None of the research actually measured if a solution to crunch time had an effect on the amount of crunch time hours worked, although one paper (PA23) recognised that the solution could reduce it.

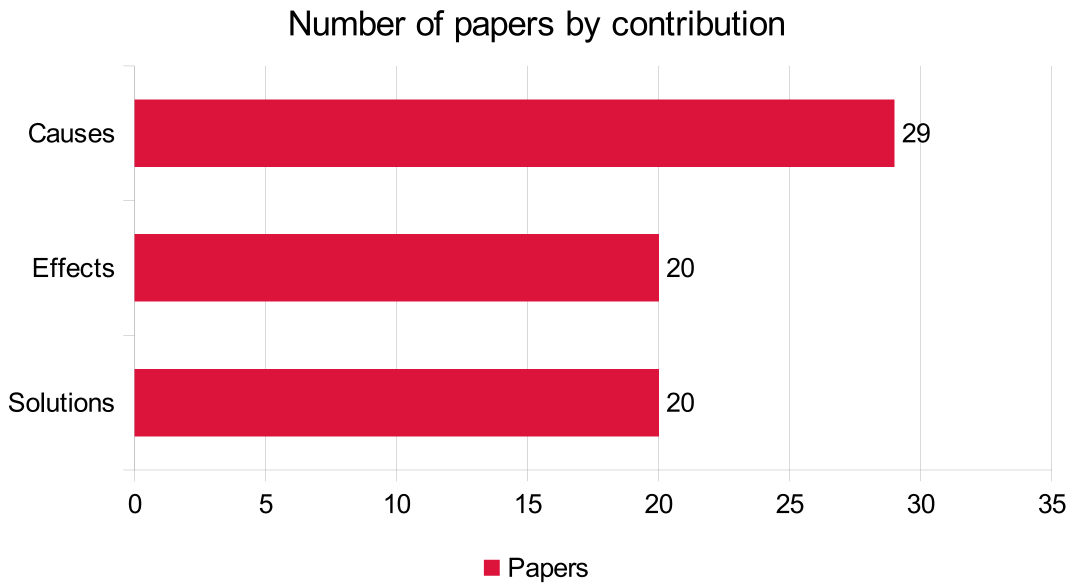

Figure 4. Number of papers by each type of contribution.

Figure 4 shows the number of papers by the contribution. Contribution means recognising a cause, effect or solution related to crunch time. Causes appear in 81% of the papers and it is the most common contribution. Effects of crunch time are presented in 56% of all papers. Equally common contribution with presenting effects is presenting solutions (56%). There is a lot of overlap because most of the papers have several contributions in relation to causes, effects and solutions.

Table 1. Types of paper by contribution.

Type	Papers (ID)	Frequency
Recognises a cause of crunch	PA1, PA3, PA4, PA5, PA7, PA8, PA9, PA10, PA11, PA12, PA13, PA14, PA16, PA18, PA19, PA20, PA22, PA24, PA25, PA26, PA27, PA28, PA29, PA31, PA32, PA33, PA34, PA35, PA36	29 (81%)
Recognises an effect of crunch time	PA1, PA3, PA4, PA5 , PA14, PA15, PA19, PA21, PA22, PA25, PA26, PA27, PA28, PA30, PA31, PA32, PA33, PA34, PA35	20 (56%)
Recognises a solution for crunch time	PA1, PA2, PA3, PA4, PA5, PA6, PA7, PA8, PA9, PA16, PA17, PA20, PA23, PA24, PA25, PA26, PA29, PA33, PA34, PA36	20 (56%)

Table 1 shows the contributions in table format and lists the paper ID's. Information regarding the corresponding papers for each ID can be found in Appendix A.

4.2 Causes of crunch time

The causes for crunch time are presented in this chapter first as a figure and then as a table showing each paper ID. These causes are classifications based on the more specific keywords presented in Appendix B.

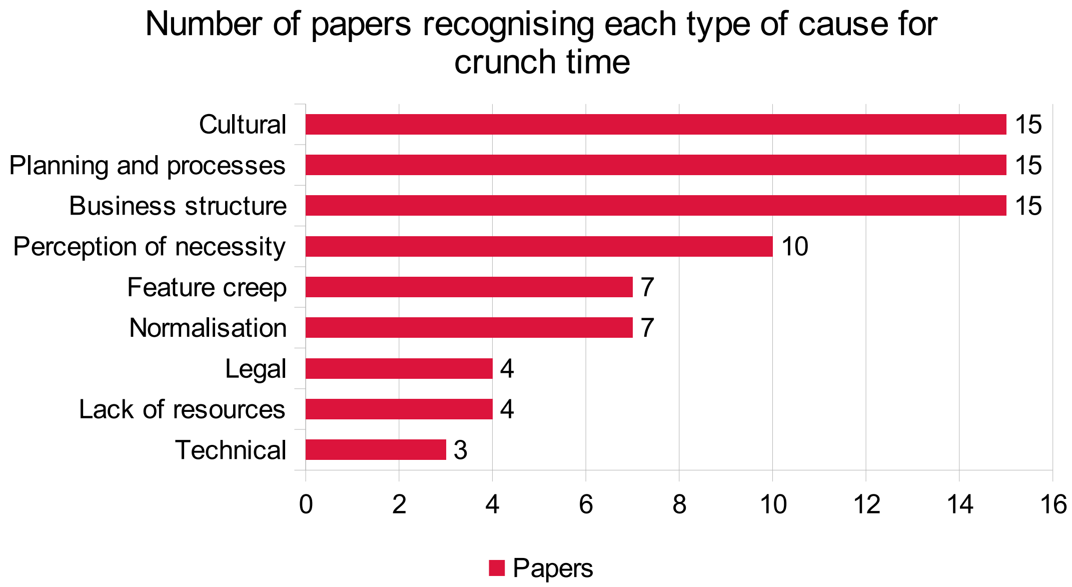

Figure 5. Number of papers recognising each type of cause for crunch time.

From Figure 5 it can be seen that cultural, planning and structural causes are very common in the literature (42% of all papers for each). Cultural causes are causes that increase both voluntary and peer-pressured crunching, usually because of passion for game development or taking pride in working long hours. Planning and process-related causes include underplanning, unclear vision and unfitting or misapplied process models. Business structure- related causes are causes arising from the structure of the business, for example a hierarchical system or the project-based structure of game development.

Perception of necessity is common (28% of all papers) in the literature, however it is not always clear in the studies if it implies that crunch is actually necessary or if it merely seems that way. A common cause for crunch time noted in the literature is that companies exploit the passion and culture in game developers and use it to normalise extreme work hours (19% of all papers). However, in the literature this culture and passion also causes developers to crunch voluntarily even if they're not necessarily forced or pressured to do so. Feature creep appeared as a cause for crunch time in 19% of all papers.

Technical issues don't appear as causes for crunch time very often (8% of all papers). Other uncommon causes were legal reasons that make crunch more likely (11% of all papers) and lack of resources, such as insufficient staffing or expertise (11% of all papers).

Table 2. Recognised causes of crunch time in the literature.

Topic	Papers (ID)	Frequency	Description
Cultural	PA1, PA4, PA5, PA12, PA13, PA15, PA16, PA19, PA25, PA27, PA28, PA31, PA32, PA34, PA36	15 (42%)	Recognises cultural issues that result in voluntary crunching.
Planning and processes	PA7, PA9, PA10, PA18, PA19, PA20, PA22, PA24, PA26, PA27, PA32, PA33, PA34, PA35, PA36	15 (42%)	Recognises planning- and process model-related issues as a cause for crunch.
Business structure	PA1, PA3, PA4, PA5, PA7, PA10, PA13, PA14, PA18, PA22, PA25, PA27, PA31, PA32, PA33	15 (42%)	Recognises the structure of the business as a cause for crunch time.
Perception of necessity	PA1, PA3, PA 8, PA10, PA14, PA18, PA19, PA24, PA31, PA35	10 (28%)	Recognises that crunch happens because it is considered necessary.
Normalisation of crunch and exploitation of passion	PA1, PA3, PA13, PA14, PA27, PA29, PA34	7 (19%)	Recognises tactics used by management to pressure workers into crunching.
Feature creep	PA4, PA7, PA9, PA10, PA18, PA20, PA33	7 (19%)	Recognises feature creep as a cause for crunch.
Lack of resources	PA10, PA18, PA20, PA22	4 (11%)	Recognises causes related to a lack of resources (understaffing and lack of skills or experience).
Legal	PA5, PA11, PA14, PA16	4 (11%)	Recognises legal reasons that allow or incentivise crunch.
Technical	PA1, PA22, PA27	3 (8%)	Recognises (unexpected) technical issues that result in crunch.

Table 2 shows the same classifications in a table format along with the paper ID's and explanations. The corresponding paper for each ID are listed in Appendix A.

4.3 Effects of crunch time

The effects of crunch time presented in the literature are shown here. These are based on the keywords presented in Appendix B.

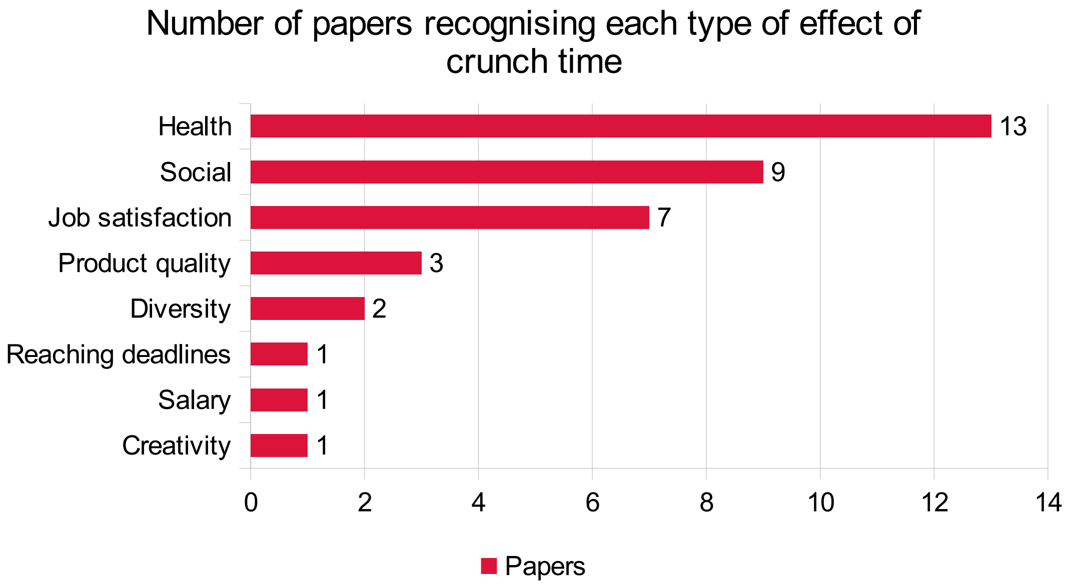

Figure 6. Number of papers recognising each type of cause for crunch time.

In Figure 6 it can be seen that health effects appear the most often in the literature (36% of all papers). Health effects include both physical and mental effects (mainly stress). Social effects, including family conflict were the second most common effect of crunch time in the literature (25%). Another common effect in the literature was job satisfaction (19%), which includes both the enjoyment of work and grievances towards the employing company.

Less common were effects on product quality (8%) and diversity (6%). In all cases where product quality was found to be an effect, the effect was negative. Diversity as an effect meant fewer female developers in one paper, and in another one it meant that people with families had a harder time finding a job in big companies (due to expectation to crunch).

Three effects appeared only in one paper each (3%). Those were reaching deadlines, salary and creativity. The effect on reaching deadlines was positive. Salary meant that the effect on an individual workers pay was positive. The effect on creativity was negative, but the paper recognising it did not clarify if it affected the product quality.

Table 3. Recognised effects of crunch time in the literature.

Topic	**Papers**	**Frequency**	**Description**
Health	PA3, PA4, PA11, PA14, PA15, PA19, PA25, PA26, PA30, PA31, PA33, PA35, PA36	13 (36%)	Recognises the health (physical and mental) effects of crunch.
Social	PA3, PA11, PA15, PA22, PA25, PA30, PA31, PA32, PA34	9 (25%)	Recognises the social effects of crunch.
Job satisfaction	PA1, PA4, PA5, PA11, PA21, PA27, PA31	7 (19%)	Recognises grievances toward company and enjoyment of work from crunch time.
Product quality	PA4, PA11, PA33	3 (8%)	Recognises the effects of crunch on the product quality.
Diversity	PA28, PA34	2 (6%)	Recognises the effect of crunch on diversity of workers in game development.
Reaching deadlines	PA4	1 (3%)	Recognises crunch having an effect on reaching deadlines.
Salary	PA31	1 (3%)	Recognises the effect of crunch on the individual worker's salary.
Creativity	PA35	1 (3%)	Recognises the effect of crunch on creativity.

Table 3 shows the same classifications along with the paper ID's and explanations. The corresponding paper can be seen in Appendix A.

4.4 Solutions for crunch time

The solutions found in the literature are presented here. The classifications are based on the keywords presented in Appendix B.

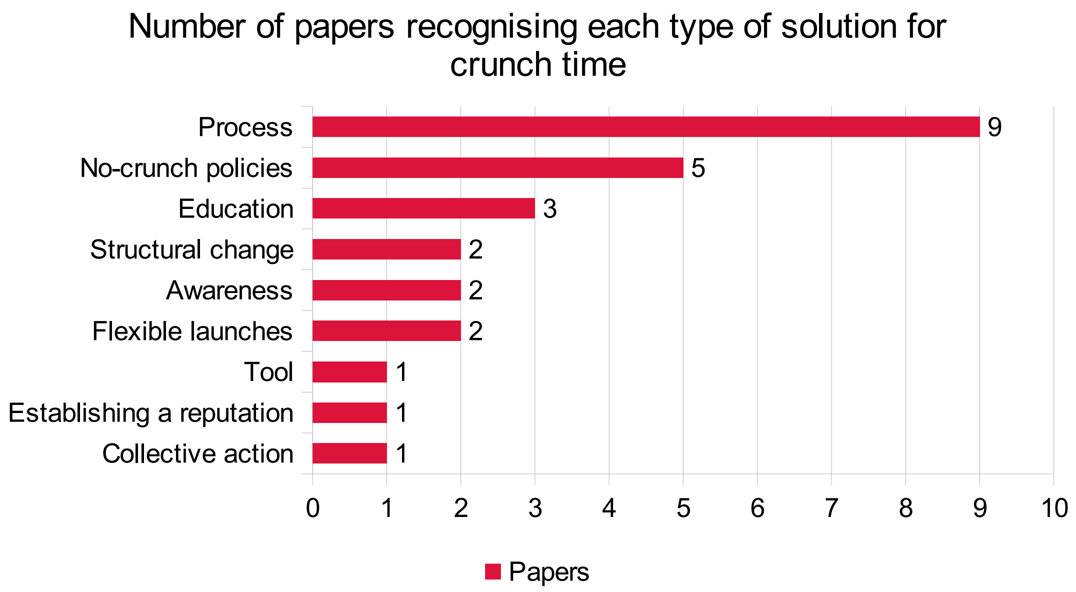

Figure 7. Number of papers recognising each type of solution for crunch time.

Figure 7 shows that process changes are the most common suggestions to appear in the literature (25% of all papers). These include entire process model changes and adding or improving certain practices, such as agile practices, prototyping and risk management. No-crunch policies were presented as solutions in 14% of all papers. These include papers that do not elaborate on the policies, but suggest that it is possible to simply not crunch.

The less common solutions were education (8%), structural change (6%), raising awareness (6%) and flexible launches (6%). Solutions related to education included every suggestion related to improving skills that could help with avoiding crunch time. A software tool suggestion, establishing a reputation and taking collective action appeared in one study each (3%).

Table 4. Recognised solutions for crunch time in the literature.

Topic	Papers	Frequency	Description
Process	PA1, PA2, PA3, PA6, PA7, PA9, PA24, PA33, PA36	9 (25%)	Suggests a process change as a solution.
No-crunch policies	PA16, PA17, PA20, PA25, PA26	5 (14%)	Presents solutions that forbid crunching and forces planning everything else around it.
Education	PA2, PA25, PA33	3 (8%)	Suggests improving skills and knowledge.
Awareness	PA2, PA3	2 (6%)	Suggests raising awareness as a solution.
Structural change	PA1, PA9	2 (6%)	Suggests a change to corporate or development team structure.
Flexible launches	PA8, PA29	2 (6%)	Suggests that flexible launch windows could make crunching less necessary.
Tool	PA23	1 (3%)	Suggests a (software) tool as a solution to avoid crunch.
Establishing a reputation	PA34	1 (3%)	Suggests that a worker can decline crunch without negative effects on career by establishing a good reputation first.
Collective action	PA5	1 (3%)	Suggests collective action and/or unionisation.

Table 4 shows the solutions in a table format along with the explanations and paper ID’s. The papers corresponding each ID are listed in Appendix A.

5. Discussion

In this chapter, the results of the study are discussed in relation to the main research question and the three assisting research questions, and compared against the prior literature. The approach was based on the assisting research questions, which are presented and answered in this chapter along with the main research question. This chapter is ordered in such a way that it at first discusses the assisting research questions, then the main research question, and finally, the implications and limitations of the study are discussed. The papers referenced in this chapter with the paper ID are presented in Appendix B.

5.1 Discussion based on assisting research questions

The first assisting research question was focused on the causes presented for crunch time:

> *How are the causes for crunch time in video game development discussed in the literature?*

The results in Chapter 4.2 showed that cultural and planning reasons were among the most common causes presented in the literature for crunch time. These reasons were expected to be common based on the prior literature. Equally common, as seen in Chapter 4.2, were structural causes, such as game development being generally project-based and hierarchical. Nearly as common was the perception of necessity, which meant that it seemed to game developers like there was no alternative to crunch time and it was taken as a fact of life.

Cultural issues include the culture of passion, meaning that developers seem to be motivated to work long hours to make the game as good as possible. This could be counterproductive as the other results of this study seem to indicate that crunching can have a negative effect on the quality of the game as seen in Chapter 4.3. Other cultural issues were taking pride in long hours and feeling the pressure to crunch when other members of the team do the same. Similarly, culture among game developers made it less likely for developers to take collective action against employers (Legault and Weststar, 2015; Chung and Kwon, 2020).

Chapter 4.2 also shows that technical issues weren’t a very common reason for crunch time. The ones identified, such as switching platforms mid-development causing delays, could even be considered a problem of planning or management. This is in line with the prior literature, which did not consider technical issues to be a big reason for crunch time (Edholm et al., 2017) .

Feature creep was presented as a cause for crunch time in fewer papers than expected based on the prior literature, which made it appear to be one of the biggest causes for crunch time (Musil et al., 2010; Edholm et al., 2017). This might imply that more research about the effect of feature creep on crunch time could be necessary and useful. Feature creep in the literature had it’s own causes, which often overlap with crunch, while being considered a cause for crunch itself. A literature review or a systematic mapping study for feature creep separately could be useful regardless of it being connected to crunch.

The second assisting research question was about the effects presented for crunch time:

> *How are the effects for crunch time in video game development discussed in the literature?*

Health and social effects were the the most common effects from crunch time as can be seen from Chapter 4.3. This was expected based on the prior literature. Both physical and mental health effects were included in the same category, but mental effects were more common than physical. Stress and family conflict were common effects as expected based on prior literature. These were included in health effects and social effects, respectively.

A relatively common effect, as seen in Chapter 4.3, was reduced job satisfaction, which might partly explain why game developers seem to leave the industry relatively early. A few papers also recognised that the product quality of the game is negatively impacted by crunching (PA4, PA11, PA33), which could be an unexpected effect for some. This could be a result of the losses of passion (Edholm et al., 2017) and creativity (PA35), which were noted in the literature. Only one paper (PA4) recognised a positive effect of crunch time being that it helps developers reach their deadlines, but this could be an obvious point that most studies would not bother to mention, because it is often the purpose of crunch time (Edholm et al., 2017).

The third assisting research question was concerned with the solutions presented for crunch time:

> *How are the solutions for crunch time in video game development discussed in the literature?*

As presented in Chapter 4.3, process changes were the most common types of solutions for crunch time. Many of those were simply implementing best practices from software engineering, but also adding prototyping or creating a game development-specific process model. The prior literature similarly suggested best practices from software engineering (Musil et al., 2010; Edholm et al., 2017) and agile (Keith, 2010; Edholm et al., 2017).

Several papers also suggested that crunch could be avoided simply by making it a rule not to crunch, which would imply planning other things around that rule. This was mentioned as a solution in the prior literature as something that was used by a game development company, but it was not not presented as a suggestion (Politowski et al., 2016).

5.2 Discussion based on the main research question

The main research question was as follows:

> *How is crunch time in video game development discussed in scientific literature?*

In this study, 36 studies from between 2008 and 2021 (before May) were examined. As can be seen in Chapter 4.1, of the 36 papers, only 47% were published, which could be taken as a need for more peer-reviewed, published, scientific research. The number of studies related to crunch time in video game development are increasing as can be seen in Figure 1 in Chapter 4.1, but the prior literature showed that the amount of crunching has been decreasing (Weststar et al., 2019). This could imply that the increase in studies has helped with recognising crunch as a problem and that the studies may have helped with reducing crunch time. Recognising causes was the most common contribution, but most papers included more than one type of contribution.

As shown in Chapter 4.1, surveys and interviews were the most common research methods along with mixed methodology studies, which use several methods. There were no experimental studies in the literature for the solutions apart from one paper (PA23), which only suggests that the solution could reduce the need for crunch. Instead, the studies suggested possible solutions and presented solutions used by developers who themselves found the solutions effective. More experimental studies could be useful, for example an experiment on set work-times along with a flexible launch window could provide useful results.

This study refrained from making assumptions about proposed solutions when the original paper did not do so. The same is true for the other direction when a solution could imply a cause for crunch time. For example, the lack of collective action was shown as a cause for crunch time in two papers (PA5, PA27), but only one of them (PA5) suggested it as a solution. A similar issue was that several papers mentioned strict deadlines as a cause (PA10, PA14, PA22, PA27), but only two papers mentioned that a flexible schedule could be a solution to avoiding crunch time (PA8, PA29). This might be because the studies assume that there is a good business reason for the strict deadlines, so changing the schedules to be flexible may not be considered to be a reasonable suggestion despite recognising strict deadlines as a cause for crunch time.

Several papers (PA1, PA3, PA5, PA10, PA13, PA14, PA18) recognised the project-based structure of game development as a cause for crunch time, but there were no suggestions for alternative business structures in the literature. A study for alternative structures could be useful. A common structural issue was that crunch time would be caused by a hierarchical structure, which likely implies that a flat hierarchy in a company would be less likely to result in crunch time.

Only two papers within the inclusion and exclusion criteria directly stated that crunch time reduces the diversity among game developers (PA28, PA34), but in the prior literature it was recognised that long work hours have a stronger effect based on factors such as gender and that could create a barrier of entry (Weststar and Legault, 2018). This could be a limitation of this study due to it specifically looking for the term “crunch”. Family conflict and other social issues arising from crunch time might also make it less likely for people with families to want to work in video games (Borg et al., 2019), but this was not a common conclusion in the literature.

Some of the classifications (recognised causes, effects or solutions) are more specific than others, but they didn’t fit the other categories. Many of the results of this study could be affected based on how the contributions are categorised. As an example, the category “job satisfaction” included both enjoyment in the work and how satisfied the worker is with the company they are employed at. Both of them have could have slightly different implications, but they are similar enough in relation to the worker’s well-being that I decided to merge them.

5.3 Implications

The main contribution of this study for researchers is presenting the current state of studies related to crunch time. It can be suggested, based on this research, that more studies should be done regarding feature creep and its effect on crunch time. Another suggestion is conducting studies on companies using uncommon company structures that are not project-based or hierarchical to see if they really result in less crunch time.

Game developers and managers can mainly use these results to recognize the causes for crunch time, which could imply some solutions in themselves. The solutions found in the literature were lacking, as seen in Chapter 4.4, but the main suggestions of adopting the best practices of software engineering (PA4, PA6, PA7) and agile development (PA2, PA4), and implementing no-crunch policies (PA16, PA17, PA20, PA25, PA26) could be effective.

5.4 Limitations

A potential limitation for this study is that there could be studies that fit the inclusion and exclusion criteria in other databases that were not used for this study. However, the decision to only use Google Scholar was mostly justified in Chapter 3 because it seems to present almost all results anyway. Some other databases that were looked at had no results that were not already in Google Scholar results. There was also a limitation in accessing scientific papers as some results were not fully accessible through University of Oulu. Limiting the language to English left out some papers that were written in other languages.

Study quality of the papers was not assessed. It was noted by Petersen et al. (2015) that it is common in published studies not to perform the quality assessment at all. In this study, there were only a total of 36 papers and only about half of them were peer-reviewed, published research, which could mean that possibly a lot of the included studies weren’t very high quality.

6. Conclusions

This thesis presented a systematic mapping study about crunch time in video game development. The purpose was to give an overview of the studies related to crunch time and what kind of contributions they have so that it can point other researcher in the right direction in regard to making more studies about crunch time. It could also be useful for workers in game development, who recognise crunch as a problem and would like to avoid it.

Only papers that presented a contribution about the causes, effects and solutions related to crunch time were included. It was found that the number of studies related to crunch time are increasing, while the prior literature showed that the amount of crunch time is decreasing as seen in Chapter 4.1. Almost half of the relevant papers were student papers.

The most common contributions in the literature were causes, as seen in Chapter 4.1, but most papers included several types of contributions. Cultural, planning and processes, and structural (business) causes were the most common causes for crunch time as seen in Chapter 4.2. Health effects were the most common effects from crunch time, as seen in Chapter 4.3, but social effects (conflicts) were also prominent. Process changes and no-crunch policies were the most common types of solutions to crunch time as seen in Chapter 4.4.

More research regarding feature creep could be necessary because it did not seem to be studied as much as the prior literature implied it to be a cause for crunch time. Studies for alternative business structures in game development companies could be useful as project-based and hierarchical structures were a very common reason for crunch time, as seen in Chapter 4.2, but the solutions were missing in the literature. As explained in Chapter 5.3, the implications for game developers and managers were to consider implementing best practices from software engineering and to consider explicit no-crunch policies.

References

Bailey, J., Budgen, D., Turner, M., Kitchenham, B., Brereton, P., & Linkman, S. (2007, September). Evidence relating to Object-Oriented software design: A survey. In *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)* (pp. 482-484). IEEE.

Borg, M., Garousi, V., Mahmoud, A., Olsson, T., & Stålberg, O. (2019). Video game development in a rush: A survey of the global game jam participants. *IEEE Transactions on Games, 12*(3), pp. 246-259.

Chung, S. W., & Kwon, H. (2020). Tackling the crunch mode: the rise of an enterprise union in South Korea's game industry. *Employee Relations: The International Journal*.

Cote, A. C., & Harris, B. C. (2021a). The cruel optimism of "good crunch": How game industry discourses perpetuate unsustainable labor practices. *New Media & Society,* 14614448211014213.

Cote, A. C., & Harris, B. C. (2021b). 'Weekends became something other people did': Understanding and intervening in the habitus of video game crunch. *Convergence*, 27(1), 161-176.

Edholm, H., Lidström, M., Steghöfer, J. P., & Burden, H. (2017, May). Crunch time: The reasons and effects of unpaid overtime in the games industry. In *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)* (pp. 43-52). IEEE.

Edwards, K., Weststar, J., Meloni, W., Pearce, C., & Legault, M. J. (2014). Developer satisfaction survey 2014. Summary report. https://igda-website.s3.us-east-2.amazonaws.com/wp-content/uploads/2019/04/21173808/IGDA_DSS_2014-Summary_Report1.pdf

Kanode, C. M., & Haddad, H. M. (2009, April). Software engineering challenges in game development. In *2009 Sixth International Conference on Information Technology: New Generations* (pp. 260-265). IEEE.

Kasurinen, J., Palacin-Silva, M., & Vanhala, E. (2017, May). What concerns game developers? a study on game development processes, sustainability and metrics. In *2017 IEEE/ACM 8th Workshop on Emerging Trends in Software Metrics (WETSoM)* (pp. 15-21). IEEE.

Keith, C. (2010). *Agile game development with Scrum.* Pearson Education.

Keogh, B. (2021). The cultural field of video game production in Australia. *Games and Culture, 16*(1), 116-135.

Kerr, A. (2011). The culture of gamework. http://mural.maynoothuniversity.ie/2901/1/Kerr_culture_of_gamework_2010.pdf

Kerr, A. (2019). 31. Game Production Logics at Work: Convergence and Divergence. *Making Media: production, Practices, and Professions*, 413.

Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Version 2.3. https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf

Legault, M. J., & Weststar, J. (2015). The capacity for mobilization in project-based cultural work: A case of the video game industry. *Canadian Journal of Communication, 40*(2), 203-221.

Musil, J., Schweda, A., Winkler, D., & Biffl, S. (2010). A survey on the state of the practice in video game software development. In *Technical report, Technical report, QSE-IF S-10/04*. TU Wien.

Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008, June). Systematic mapping studies in software engineering. In *12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12* (pp. 1-10).

Petersen, K., Vakkalanka, S., & Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64, 1-18.

Peticca-Harris, A., Weststar, J., & McKenna, S. (2015). The perils of project-based work: Attempting resistance to extreme work practices in video game development. *Organization*, 22(4), pp. 570-587.

Petrillo, F., Pimenta, M., Trindade, F., & Dietrich, C. (2009). What went wrong? A survey of problems in game development. *Computers in Entertainment (CIE),* 7(1), 1-22.

Politowski, C., Fontoura, L., Petrillo, F., & Guéhéneuc, Y. G. (2016). Are the old days gone? A survey on actual software engineering processes in video game industry. In *Proceedings of the 5th International Workshop on Games and Software Engineering* (pp. 22-28).

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*, 134, 106538.

Robinson, E. (2005). Why crunch mode doesn't work: 6 lessons. https://ericdodds.com/wp-content/uploads/2019/07/88558-igda-articles-evan-robinson-why-crunch-mode-doesnt-work-6-lessons.pdf

Weststar, J. (2012). Occupational community-opportunity or threat to collective action among video game developers. *LERA for Libraries*.

Weststar, J., Kwan, E., & Kumar, S. (2019). Developer Satisfaction Survey 2019. Summary report. https://igda-website.s3.us-east-2.amazonaws.com/wp-content/uploads/2020/01/29093706/IGDA-DSS-2019_Summary-Report_Nov-20-2019.pdf

Weststar, J., & Kumar, S. (2020). Developer Satisfaction Survey 2019. Industry Trends and Future Outlook Report. https://s3-us-east-2.amazonaws.com/igda-website/wp-content/uploads/2020/11/25095744/IGDA-DSS-2019-Industry-Trends-Report_111820.pdf

Weststar, J., & Legault, M. J. (2018). Women's Experiences on the Path to a Career in Game Development. In *Feminism in Play* (pp. 105-123). Palgrave Macmillan, Cham.

Wohlin, C., Runeson, P., Neto, P. A. D. M. S., Engström, E., do Carmo Machado, I., & De Almeida, E. S. (2013). On the reliability of mapping studies in software engineering. *Journal of Systems and Software, 86*(10), 2594-2610.

Wohlin, C. (2014, May). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering* (pp. 1-10).

Yasin, A., Fatima, R., Wen, L., Afzal, W., Ahar, M., & Torkar, R. (2020). On using grey literature and Google Scholar in systematic literature reviews in software engineering. *IEEE Access*, 8, 36226-36243.

Appendix A. List of studies

PA	Authors	Year	Title	Publication type	Research method
1	Cote, A. C., & Harris, B. C.	2021	‘Weekends became something other people did’: Understanding and intervening in the habitus of video game crunch.	Journal	Critical discourse analysis
2	Borg, M., Garousi, V., Mahmoud, A., Olsson, T., & Stålberg, O.	2019	Video game development in a rush: A survey of the global game jam participants.	Journal	Survey
3	Peticca-Harris, A., Weststar, J., & McKenna, S.	2015	The perils of project-based work: Attempting resistance to extreme work practices in video game development.	Journal	Blog analysis
4	Edholm, H., Lidström, M., Steghöfer, J. P., & Burden, H.	2017	Crunch time: The reasons and effects of unpaid overtime in the games industry.	Conference	Interviews
5	Edholm, H., Lidström, M., Steghöfer, J. P., & Burden, H.	2020	Tackling the crunch mode: the rise of an enterprise union in South Korea's game industry.	Journal	Case study
6	Petrillo, F., Pimenta, M., Trindade, F., & Dietrich, C.	2008	Houston, we have a problem... a survey of actual problems in computer games development.	Conference	Survey
7	Musil, J., Schweda, A., Winkler, D., & Biffl, S.	2010	A survey on the state of the practice in video game software development.	Journal	Survey
8	Lehtonen, M., Lu, C., Nummenmaa, T., & Peltonen, J.	2019	Adoption of Requirements Engineering Methods in Game Development: A Literature and Postmortem Analysis.	Conference	Postmortem analysis
9	Politowski, C., Petrillo, F., Ullmann, G. C., & Guéhéneuc, Y. G.	2021	Game industry problems: An extensive analysis of the gray literature.	Journal	Postmortem analysis
10	Legault, M. J., & Weststar, J.	2015	The capacity for mobilization in project-based cultural work: A case of the video game industry.	Journal	Mixed methodology
11	Kim, C., & Lee, S.	2020	Fragmented industrial structure and fragmented resistance in Korea’s digital game industry.	Journal	Interviews
12	Legault, M. J., &	2010	Is the very notion of "representation" relevant for the	Conference	Mixed

	Weststar, J.		regulation game of video game developers?		methodology
13	Legault, M. J., & Weststar, J.	2017	Videogame developers among 'extreme workers': Are death marches over?	Journal	Mixed methodology
14	Cockayne, D.	2020	Learning to labor in high-technology: experiences of overwork in university internships at digital media firms in North America.	Journal	Interviews
15	Weststar, J., O'Meara, V., Gosse, C., & Legault, M. J.	2017	Diversity among videogame developers, 2004-2015.	Journal	Mixed methodology
16	Sotamaa, O.	2021	Studying Game Development Cultures.	Journal	Interviews
17	Sapsed, J., Mateos-Garcia, J., & Grantham, A.	2008	Upsetting and Settling-Down Genres: The role of organisation practice in the reproduction of cultural products	Conference	Interviews
18	Edwards, K., Weststar, J., Meloni, W., Pearce, C., & Legault, M. J.	2014	Developer satisfaction survey 2014. Summary report.	Grey literature	Survey
19	Legault, M. J., & Weststar, J.	2012	Quality of Life in the Game Industry. Report of the Quality of Life survey 2009.	Grey literature	Survey
20	Weststar, J., & Legault, M. J.	2014	Developer Satisfaction Survey 2014. Employment Report.	Grey literature	Survey
21	Larsson, C.	2018	THE ART OF CRUNCH-A quantitative study on the effects of a high-pressured period during video game production on organizational commitment.	Master's thesis	Survey
22	Suorsa, N.	2017	Independent game development–Developers' reflections in postmortems	Master's thesis	Postmortem analysis
23	Hooper, S.	2017	Automated Testing and Validation of Computer Graphics Implementations for Cross-Platform Game Development	Master's thesis	Design science research
24	Archontakis, I.	2019	Agile development in the video game industry: Examining the effects of iteration and methods of limiting it.	Master's thesis	Mixed methodology
25	Roininen, T.	2013	The quality of life in the Finnish game industry.	Master's thesis	Survey

26	Romine, M. L.	2016	Fractured Imaginaries-an Ethnography of Game Design	Doctoral Dissertation	Ethnography
27	LaLonde, M.	2020	Behind the Screens: Understanding the Social Structures of the Video Game Industry.	Master's thesis	Interviews
28	Ruiz, B., & Nadia, V.	2019	VIDEO GAME CREATION: Inhibitors and Enablers in Female Inclusion.	Master's thesis	Interviews
29	Vanderhoef, J.	2016	An industry of indies: the new cultural economy of digital game production	Master's thesis	Mixed methodology
30	Mendes, J. C. D. B. M.	2020	Burnout and job demand and resources among Game Developers.	Master's thesis	Survey
31	Bulut, E.	2014	Creativity and its discontents: a case study of precarious playbour in the video game industry	Doctoral Dissertation	Case study
32	Johnson, R. S.	2010	The digital Illusio: gender, work and culture in digital game production.	Doctoral Dissertation	Mixed methodology
33	Westerdahl, M.	2019	Challenges in video game development-What does Agile management have to do with it?	Undergraduate thesis	Mixed methodology
34	Newbery, M.	2013	Gender and the Games Industry: The Experiences of Female Game Workers	Doctoral Dissertation	Interviews
35	Josefsson, I.	2017	Navigating Creative Work.	Doctoral Dissertation	Interviews
36	Lipkin, N. D.	2016	Agents at work: decision making capacity and creative labor in network society	Doctoral Dissertation	Case study

Appendix B. List of keywords with papers containing them

Keyword	**Type**	**Papers**	**Number of papers**
Passion / Culture / Voluntary crunching	Cause	PA1, PA4, PA5, PA12, PA13, PA15, PA19, PA25, PA28, PA31, PA34	11
Perceived necessity	Cause	PA1, PA3, PA8, PA10, PA14, PA18, PA19, PA24, PA31, PA35	10
Stress	Effect	PA4, PA14, PA24, PA26, PA33, PA35	6
Frustration	Effect	PA4, PA5	2
Agile	Solution	PA2, PA4	2
Agile	Cause	PA27	1
Feature creep	Cause	PA4, PA7, PA9, PA10, PA18, PA20, PA33	7
Worse quality	Effect	PA4, PA11, PA33	3
Scope management	Solution	PA2, PA9	2
Bad scope management	Cause	PA9, PA10, PA20	3
SE best practices	Solution	PA4, PA6, PA7	3
Misapplication of SE processes	Cause	PA7, PA33	2
Switching technologies mid-development	Cause	PA1, PA27	2
Project-based structure	Cause	PA1, PA3, PA5, PA10, PA13, PA14, PA18	7
Unrealistic schedule	Cause	PA4, PA10, PA25	3
Prototyping	Solution	PA9	1
Lack of prototyping	Cause	PA22	1
Improving communication	Solution	PA2, PA33, PA36	3
Bad communication	Cause	PA9	1
Raising awareness	Solution	PA2, PA3	2

Bad documentation	Cause	PA9	1
Family conflict	Effect	PA3, PA15, PA25, PA31, PA32, PA34	6
Bad risk management	Cause	PA10	1
Improving risk management	Solution	PA33	1
Unclear vision	Cause	PA10, PA18, PA20	3
Lack of experience	Cause	PA10, PA18, PA22	3
Ends justify the means	Cause	PA1, PA12, PA14	3
Feeling of being exploited	Effect	PA27	1
Burnout	Effect	PA19, PA30, PA31, PA33	4
Normalisation of crunch	Cause	PA1, PA27, PA35	3
Strict deadlines	Cause	PA10, PA14, PA22, PA27	4
Lack of collective action	Cause	PA5, PA27	2
Unionisation / collective action	Solution	PA5	1
Overstaffing	Cause	PA9	1
Understaffing	Cause	PA18, PA20	2

Appendix C. List of studies with raw keywords

PA	Causes	Effects	Solutions
1	Perceived necessity, normalisation, ends-justify-the-means, unmanageable creative industry, anti-corporate ethos; passion and perfectionism, project-based structure, culture of passion and secrecy, unexpected inspiration, switching technologies, underplanning due to publisher demands, crunch preferred to not meeting publisher expectations		Changes over time, set work hours, corporate structures
2			Iterative brainstorming, continuous integration, minimum viable product, scope management, version control, stand-up meetings, regular communication, internal playtesting, dynamic and proactive planning, familiarity with agile, raising awareness in game jams
3	Project-based structure, perceived necessity, neo-normative control mechanism	health issues, family conflict	Raising awareness on social media
4	unrealistic schedule, feature creep, culture overrides best practices, passion, organisational more than technical	Increased stress, increased frustration, meeting deadlines, more features, worse product quality	Regular prioritisation of features, agile best practices
5	passion, project-based work, lack of collective action, reputation based hiring, wage system allows unpaid overtime	Increased grievances, increased frustration	unionisation

6			SE best practices
7	Feature creep, inappropriate application of software engineering processes		SE best practices, VGD-specific process models
8	Perceived inevitability		Loose scheduling
9	Poor scope management, feature creep, overstaffing → bad communication, unclear vision		Pair programming and code reviews, scope management, balance expertise levels, small teams, small scopes, protect from external interference, documenting work for time estimation, long pre-production, prototyping, playtesting, creative control for developers
10	Bad risk anticipation, scheduling, underestimated difficulty, project-based structure, strict deadlines, poor management, changing customer requests (scope creep), perceived necessity, inexperience		
11	Wage system allows unpaid overtime	mental and physical pain, breakdown of work-life balance, reduced quality of product and life, work becomes pain instead of fun	
12	Ends-justify-the-means, meritocracy		
13	pressure from management, project-based organisation and career progress system, social pressure		
14	Project-based structure, “organised around deadlines” (necessity	stress	

	justifies overwork), cultivation of feelings of indebtedness and obligation, wages create an incentive to work more, meals and meal-tickets as incentives		
15		negative effect on family and personal relationships, emotional and physical health	
16	formal regulations and working cultures might affect no-crunch policies (or lack of)		No-crunch policies (following work time regulations)
17			No-crunch policy
18	Perceived necessity (by 24% of developers), scheduling, feature creep, unclear expectations, insufficient staffing, inexperienced management, project-based structure		
19	Failure in scheduling, crunch planned into the schedule (so it is inevitable), culture (taking pride in long hours), Perceived necessity (by 34.7% of developers)	burnout	
20	poor scheduling, feature creep, unclear expectations, understaffing, time management, scope, resourcing		“doing everything to avoid it”
21		lower affective commitment to company	
22	Errors found late, not enough time left for testing and polishing, poor documentation, lack of prototyping, building a company while developing, lay-offs, platform changes, deadlines = crunch, poor scheduling (optimistic estimates, no experience in estimating time required for phases)	“Toll on personal life”	

23			Automated testing
24	Too much iteration → crunch, perceived necessity		Reduce iteration, interact with customers, trust established gameplay mechanics, rely on franchise
25	Optimistic schedule, culture, just another way of working	stress, family conflict	better management skills, formal policies
26	Mismanagement, poor planning, perceived as worth the sacrifices	stress	just stop crunching
27	switching technologies, deadlines = crunch, agile and scrum (bad implementation? Paper doesn't elaborate), normalisation tactics (company tries to make crunch a "voluntary" effort toward a common goal), pressure to crunch to get a full-time job, feeling of privilege to work on games, collective action is difficult, scare tactics against organising	feeling of being exploited	
28	individualistic behavior (culture, toxic masculinity)	Inhibits female inclusion	
29	Demand to be in the office, narrative of passion and self-determination		flexible launch windows
30		Increased time demand, increased mental demand, increased concentration demand, increased emotional demand, lower levels of ethical resources (disrespect of equity and justice), burnout, loss	

		of insight and experience, sacrificing personal relations	
31	“fraternity house”, desire to make the best game, “youthful camaraderie”, culture, pride in work, necessary evil, passion, requirements by parent company	mental and physical burnout, family conflict, working and leisure blur, demoralisation, loss of passion, physical pain, loss of social life, degradation of fun, better pay	
32	structural, part of the process, masculinity, long hours are honorable	increased affective labor for spouses and family members	
33	hierarchical structure, strong profit drive, incomplete agile methods, feature creep (as a result of the previously mentioned reasons)	stress, burnout, worse product quality	Fully implementing agile principles, prioritising risk analysis more, implementing measures for the risks, improving interpersonal relationships and “soft skills” for better communication of concerns
34	Culture (competitiveness about hours worked/commitment, peer pressure), companies/managers exploit passion, pressure to work long hours to get respect and advance in career	worse work-life balance, discourages from starting families, having a family might prevent a worker from getting a job in bigger game	establishing a reputation first can help a person to work normal hours without penalties

		companies	
35	Insufficient or ineffective pre-production, normalisation of crunch, perceived necessity	stress, lower creativity	
36	Bad planning, bad scoping, culture of secrecy		avoid over-scoping by making participants work on the most important things first (prioritisation), conversations about practice (communication, better approximation of hours required for tasks)