UNIVERSITY OF GOTHENBURG

Crunch time: the Causes and Effects of Overtime in the Games Industry

Bachelor of Science Thesis in Software Engineering and Management

HENRIK EDHOLM
MIKAELA LIDSTRÖM

University of Gothenburg
Chalmers University of Technology
Department of Computer Science and Engineering
Göteborg, Sweden, June 2016

Crunch time: the Causes and Effects of Overtime in the Games Industry

Henrik Edholm
Mikaela Lidström

Examiner: Håkan Burden

University of Gothenburg
Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
Göteborg, Sweden June 2016

Crunch time: the Causes and Effects of Overtime in the Games Industry

Henrik Edholm & Mikaela Lidström
Dept. of Computer Science and Engineering
University of Gothenburg
Gothenburg, Sweden
gusedhhe@student.gu.se, m.s.lidstrom@gmail.com
Supervisor: Jan-Philipp Steghöfer

Abstract—**The games industry is notorious for its intense work ethics with uncompensated overtime and weekends at the office, also known as crunch. By studying postmortems and conducting interviews with employees in the industry we explore the crunch phenomenon. The aim is to discover what effects it has on different aspects of the industry and whether or not agile principles affects this. We have found four types of crunch which all have distinct characteristics and effects the product, employees and schedule in various ways. Crunch is a widespread phenomenon within the games industry that affects the product and the staff in mostly negative ways.**

I. Introduction

Crunch time is a term used in the games industry to describe periods of extreme workload. During this period the employees at games studios often work 12 hours per day for several weeks, or even months. According to Petrillo et al. this primarily occurs before the final product delivery [1]. Crunch has existed within the games industry for a long time and is usually carried out to ensure that the game is released as scheduled [1]. It does not help that the project scope tends to be unrealistic within the scheduled deadlines. Big features will often be added during the development without adjusting the development time accordingly, a practice referred to as "feature creep" [1].

Many believe that crunch time within the games industry is part of the work culture and therefore it can not be changed, leading to employees just accepting it [1],[2]. Recent research claims that crunch time itself can contribute to late deliveries and low quality software, making it a vicious circle [1],[3]. Sleep deprivation can significantly reduce developers ability to make rational design decisions and produce high quality software [1]. Working between 60 and 90 hours per week also has a grave effect on the employee's personal relationships and mental health, leading to many people not wanting to stay within the industry [1],[4]. These reasons should be incentives for the industry to want to change this practice.

Koutonen [5] conducted a study aimed to understand how games companies deploy agile practices and how successful they were. The study showed that many things improve when adopting agile principles, but overwork was still problematic. Keeping the agile practices in mind, in particular keeping a constant development pace, one would think that companies working accordingly would crunch less.

A potential way of avoiding crunch, that many indie game studios have adopted, is to release games iteratively. This could reduce the stress that impending deadlines put on employees, through the use of continuous feature releases until the game studio consider the game completed. A disadvantage of this release approach is that it requires the studio to be quite open with their idea, meaning that competitors could copy the game. This release approach has proved successful, e.g. *Minecraft* was made available to the public in 2011 and was continuously improved during its alpha- and beta stage. Even though the game was officially released in 2014, new features are still getting implemented to this day.

It is still unclear at this point why the game industry maintains this way of working despite evidence of its downsides. Do the benefits of crunch outweigh its downsides? Is it the culture of the industry that influences people to work this way? This research intends to shed light on the phenomenon of crunch and understand why it occurs in the games industry. To do this we have formed the following questions:

- Q1: Is crunch time a widespread phenomenon in the games industry?
- Q2: What are the negative and positive impacts of crunch time?
- Q3: What are the most common reasons for crunch time?
- Q4: Does the games industry's culture affect people's willingness to crunch?
- Q5: How does agile affect the crunch time phenomenon?

By giving an answer to the questions above we believe that both industry and academia will get a better grasp of the effects of crunch.

II. Methodology

This study was split in three phases, in the first phase we explored the current state of the art through literature and collected postmortems; in the second phase we collected data through conducting interviews and reading postmortems; in the third phase we analysed and compared the gathered data.

The first phase of the study was a qualitative exploration of what previous literature has said about the topic. The literature was collected in two separate steps. Firstly we wanted to see how vastly academia had researched the topic, by reviewing scientific papers we got a good grasp about the state of the art. Secondly we collected postmortems written by game studios. A postmortem in the games industry is “a document that summarises the project development experience” [1]. It strongly focuses on reflecting on what went right during the development and what went wrong. It aims to acknowledge the issues of development in a constructive way so that the team can learn and improve for the next project. It is often shared on dedicated websites to bring this knowledge to other companies.

In the second phase we conducted interviews with staff from four different game studios and read through the postmortems. The interview questions were based on the findings from the related literature. For example, we noticed in the related work [1] that the game culture would often influence people to personally crunch without being told to. We made sure to take this into consideration when writing our questions.

Through the interviews we expected to get an insight into why the industry crunches; what impacts crunch have on the employee and the product; how many of the agile principles the organisations fulfils. Based on pilot interviews that took between 30 and 40 minutes, we planned for our interviews to not exceed one hour. The actual interviews with the game studios did however vary a lot, from 15 to 40 minutes. In the postmortems we looked at which issues the game studios mention to have occurred during the development. The main focus was to see if they mention crunch time and what kind of impact it had on the employees, the product and the schedule.

For the third and final phase we have analysed and compared the data gathered from the interviews with the postmortems.

A. Review of postmortems

1) Data sources: Like other researchers in the field [1],[2],[4],[6] we have used the website “Gamasutra” as an information source to further understand the industry.

2) Search strategy: We used a very straightforward approach for finding our postmortems. Gamasutra has a dedicated section just for postmortems (http://www.gamasutra.com/features/postmortem/) which is where we collected them. From this section we selected the first 18 pages of postmortems which were written after the agile manifesto was published.

3) Study selection: Given that *Gamasutra* is a website and that the postmortems are written by those who worked on the product we needed to take validity threats, such as bias, into consideration. We therefore came up with strict inclusion- and exclusion-criteria treating these issues, please refer to Table I. For a study to be selected it had to fulfil **all** inclusion criteria, but if it just contradicted **one** exclusion criteria it was discarded.

I1 describes our focus area. Since we were interested in the challenges of game development, specifically crunch time, postmortems from other types of software development were of no interest to us. This lead to E1 being enforced and the postmortem being discarded. I2 ensured us that we were able to analyse the postmortems. In order to increase the trustworthiness of the postmortems they needed to critically assess what went well and what went wrong, rather than tooting their own horn. This is why I3 exist. A postmortem was deemed as being critical when it mentioned what went wrong during the development and not only what went well. To further strengthen this we have E2 since we did not want to handle hobby projects etc. E3 is based on the release date of the ‘Agile Manifesto’. Before its publication it is not safe to assume that people in the industry had an awareness of the agile mindset.

TABLE I: Postmortem Inclusion & Exclusion Criteria

Inclusion Criteria	
I1	An article that is a postmortem of a game.
I2	The article must be written in English.
I3	The postmortem must critically analyse the development process.
Exclusion Criteria	
E1	An article that is not a postmortem of a game.
E2	A postmortem of a game with a team of less than 5 people.
E3	A postmortem written before 2001.

The table above shows the inclusion and exclusion criteria used to ensure the quality of the postmortems used for this study.

Before applying any inclusion/exclusion criteria we had access to 218 postmortems. After applying exclusion criteria E3 we gathered a total of 180 postmortems. When we added all the postmortems to our spreadsheet we discarded 90 postmortems due to them contradicting the exclusion criteria. After this we read through the remaining 90 postmortems and further discarded 12 due to not fulfilling all of our inclusion criteria. Leaving us with a total of 78 postmortems.

B. Interview strategy

Our study is of an exploratory nature [7], meaning that we aim to find out what is going on in the games industry and whether crunch is deemed a problem or not. Postmortems tend to give a general view of a project. While this is valuable in order to grasp common issues in the industry, we desired a deeper insight to understand the culture. By conducting interviews with staff at game studios we could get a different perspective on the reasons for, and willingness to, crunch. We believed that we could get more elaborate answers by holding interviews in a semi-structured way, i.e. the interviewees were guided through our questions but encouraged to speak freely. To accommodate for this we used open ended questions. To make up for close ended questions we added probing follow-up questions to get more detailed information.

1) Assessing agile and crunch: The questions were divided into two blocks, the first consisting of questions assessing what kind of agile principles the studios follow. This block starts with non-challenging questions, about the interviewees work experience and what their position is at the company etc., so that the interviewees do not feel intimidated. After these warm-up questions we asked the interviewees about their development process.

The principles have been derived from Soundararajan's et al. (2012) **O**bjective, **P**rinciples and **P**ractices (*OPP*) Framework for assessing agile [8]. From these principles we have chosen the following six that we indirectly ask about: *frequent delivery of working software*, *empowering teams of motivated individuals*, *accommodating change*, *continual stakeholder communication and collaboration*, *frequent reflection and improvement* as well as *constant development pace*.

We chose to exclude *technical excellence*, *simplicity* and *striving for customer satisfaction* from the interviews since we determined that the interviewees would potentially not answer them without bias.

we felt like we could not formulate questions which would not put the interview subject in a defensive stance or even have them answer them with bias. For example, to ask if a company strives for customer satisfaction can lead to the subject feeling like they are putting their company in a bad light if they say no, and therefore be more inclined to say yes.

It could even put them in a defensive stance for the rest of the interview.

The second block of questions focused on crunch at the studio as well as the industry in general. With these questions we tried to assess how much the studios crunch and if this crunch takes place over a long period of time. We were interested in what reasons are given for crunch by the management. Our questions also focused on how the game industry's culture affects people's views and willingness to crunch. Therefore we explored if people did overtime on their own accord, and why.

2) Pilot study: To get a better understanding of our questions' quality we conducted three pilot interviews with fellow students. This experience helped us detect issues in question formulation, e.g. a few questions were leading and some provided too little information. We solved these issues by reformulating them and adding probing questions to follow up the ones that provided too little information. The pilot studies took between 30 and 45 minutes which we believed to be a representative time frame for the actual interviews.

3) Subject selection: We gathered interview subjects by A) contacting the game studios that are members of Dataspelsbranschen (http://www.dataspelsbranschen.se/) via email and B) contacting acquaintances in the industry. Five of the 29 potential subjects replied with an interest to participate in our study. We set up an interview with these five subjects, two of which are from the same company.

4) Two types of interviews: One of our subjects did not feel comfortable with answering questions face-to-face. For this subject we reformulated our interview questions so that they fitted a textual response while being equivalent to the responses gathered in the face-to-face interviews. The face-to-face interviews were in the same time frame as the pilot studies. One did however vastly differ and only took 15 minutes. We believe that this was due to the subject feeling uncomfortable with the setting and we would most likely have received more in-depth information if the interview would have been less formal and we had eased the subject in to it. Thanks to our questions having built-in probing we managed to get a general understanding of their studio, however not as in-depth as in the other interviews.

C. Data Extraction

We used similar methods to extract data from both postmortems and interviews.

1) Postmortems: When extracting data from our postmortems we read through them thoroughly and identified issues the game studios had experienced. For issues that were common we created columns representing that particular problem. The common issues identified in the postmortems were: *pre-production issues*, *feature creep/too big scope*, *planning/scheduling issues*, *publisher disagreement or pressure*, *communication issues*, *lack of focus/focusing on the wrong things*, *bugs*, *poor management/management issues*, *financial issues*, *technical issues*, *process issues*, *unfun game* and *staffing issues*.

When a postmortem mentioned that the studio had experienced one of the issues we marked it in our coding sheet under the specific column. We also included columns for if the studio crunches and how this affects employees, product and schedule, if the studio work agile or not as well as the Metacritic score. Metacritic [9] is a website that summarises other game critics reviews to one combined score.

Following is an example of how we extracted data from the postmortems: The studios initial estimate for how long the development would take were off by more than 700%, when the studio adjusted after a few months they still were off by 200% and resorted to crunching. This is a clear indication that the studio had *planning/scheduling issues*. The author mention that the studio constantly put energy on the wrong areas instead of trying to finish the game which shows that they had a *lack of focus/focusing on the wrong things*. As an example they consider the lack of tools one of their biggest issues throughout the development since it made everything more time consuming than necessary. Pilestedt describes how the studio realised in the middle of development that the game was not fun enough, this led to *"wasting several weeks"* making the game fun again [10]. For this reason the studio has fulfilled the *unfun game* column. The author further explain how the studio implemented the agile process Scrum late in development [10]: *"by the time we implemented our SCRUM solution, it was far too late, but it gave us a sense of direction, a sense of progress and a sense of control – something that allowed us to focus on what was really important"*. This indicates that the team worked agile for parts of the development which deem the criteria of the agile column fulfilled. The studio had *financial issues* for a big part of the project which made them borrow money from family and friends. Since the studio was unaware of when the game would be finished the team had a overwhelming feeling of stress and frustration. At a few places the author of the postmortem describes their crunch, e.g. *"We were always having 16 hour/day crunches weeks before milestones that we would, in the end, miss."* [10] and *"We instead took it upon ourselves to work overtime for several* *consecutive weeks to catch up for previous misjudgments and attempt to reach new impossible milestones"* [10] these quotes indicate that the studio did Continuous Crunch (see Section IV-B).

2) Interviews: To extract data from our interviews we created a coding sheet. We split the sheet into two main areas, "Assessing agile" and "Assessing crunch". In the 'Assessing Agile' sheet we used six of the OPP Framework's nine principles, see Section II-B1, as headlines. To be more specific we added sub-headlines, which we derived from the answers of our interviewees, under said headline. We linked the sub-headlines to an identification number in the text so that it could be easier to find the specific place where it was mentioned. The same process was repeated for the 'Assessing Crunch' sheet, but these headlines were derived from our research questions.

Following is an example of how we extracted data from the interviews: When asking one of our subjects from Company C the question of how they handle changes, we got the answer: *"very dynamically. Meaning that we tend to change things sometimes out of very, fairly quick basis"* and *"we're such a small team so we don't need a lot of processes in place to manage change."* This gave us two keywords, "Dynamically" and "Informal", we put these two words as sub-headlines under "Accommodating change" in the coding sheet. Then we marked these two sub-headlines with the given identification numbers. Any other subject who we determined mentioned these keywords would get their own identification number and it would be marked down under the same sub-headline. Later this subject from Company C mentioned: *"The release we're planning right now is sort of, there aren't many things in that release we can drop so if things don't go according to our short term plan we have right now we will probably push the release a little."* Because of this we felt that they fulfilled the "Accommodating change" criterion and marked this headline under the company's column with a "Yes". The same procedure was done for the rest of the assessing agile questions. When coding the assessing crunch answers, one of our interviewees from Company A said this to our question regarding what reasons were given to crunch: *"this week is the last chance to get anything fixed before we submit"* and *"crunch pretty much every single time we have to do a submission."* We put this down in our coding sheet under the headline "Reasons for crunch" as "Deadline". Just like when assessing agile it when another subject mentioned deadlines as a reason we marked it down under the same headline. The same procedure was done in order to answer all of our research questions related to crunch.

D. Validity Threats

The criteria for validity are based on Easterbrook et al. [11].

Construct validity is a threat if the research design is vague and up for interpretation. The research questions variables are all measured by data gathered through studying postmortems and collecting insights from the industry via interviews. The postmortems can however be interpreted differently depending on the reader. When a postmortem for example mentions that the studio lack money for equipment we interpret it as a financial issue, while another researcher could interpret this as a technical issue. To lower the risk of misinterpreting this qualitative information we have cross-read the postmortems and discussed the categories and the data frequently, to ensure that both researchers are on the same page.

Internal validity concerns the design of the study. Some interviewees could express concern for their reputation within the industry and also their jobs when agreeing to participate in this study. By offering the interviewees a chance to be anonymous we believe that we got more honest and thorough answers, making the results more accurate to how it is within the industry. We have also designed our interview questions as neutral as possible in order to not put the interviewees in a defensive stance. This was achieved by removing questions regarding three of the OPP Frameworks principles, as mentioned in Section II-B1.

It is likely that some postmortems do not mention that the studio crunched during the development even though this might have been the case. We suggest two possible explanations to this. Firstly crunch is not considered one of the top five things that went right or wrong. Secondly crunch is not seen as something worth mentioning because it's part of the games industry's work culture. This is further discussed in Section V-B. We also believe it possible that more studios from the postmortems, than the ones we have identified, work according to agile principles. This is likely due to two reasons. Firstly the author may not consider the studio's work process as an important part to discuss in a postmortem. Secondly the author may not consider what went right or wrong in their process as a top five priority and therefore it is not mentioned in the postmortem. It can also be difficult to determine if the studio work agile based on the information the author has given in the postmortem. We have been very cautious with determining whether or not a studio work according to agile principles. Studios have been considered agile when the process has been clearly defined or mentioned by name. We have further considered studios to work agile when iterative and incremental development has occurred.

External validity looks at whether the results from the study are generalisable with industry standards. The interviewees consist of two programmers, two artists and one CEO. This makes the insights gathered from the interviews a representative subset from a hierarchical point of view. Since we also gather data from postmortems written by staff at studios from all over the world, in different positions, we believe that our findings are to be considered as generalisable. When reading the postmortems it becomes clear that people with different views have written them, which makes us confident that the data set is varied. Although the aim of a postmortem is to be as transparent of the issues during the development process as possible, there might be certain postmortems that hide sensitive information or downplay issues. We do feel that the authors of the postmortems are honest about their shortcomings in order to help other studios learn from them. We also have strict inclusion- and exclusion criteria where we exclude postmortems that do not seem to critically reflect on their problems.

For *reliability* we look at the likelihood for other researchers to come up with the same results if they were to replicate the study. Our methodology is transparent, therefore we believe that another researcher following our steps would find the same results as us. However, if the researcher would not follow our interview structure and pose questions that are not neutral, they might influence their interviewees to answer differently and therefore get different results. Our interview subjects are all from northern Europe. Because of this it might be an issue reconstructing this study with subjects from other continents, or other parts of Europe. Interviewees might give different answers based on cultural influences.

III. Meet your maker

Following is a description of the different companies we have been in touch with for interviews, all of which are micro, small and medium-sized businesses (SMEs) that reside in Europe. We follow the European Union's (EU) standardised subdivision of SMEs [12]:

- Micro < 10 employees
- Small < 50 employees
- Medium < 250 employees

A. Company A

Company A is a medium sized games company developing for Xbox One, Playstation 4 (PS4) and Personal Computer (PC). From this company we interviewed two programmers who have worked in the industry for four and six years respectively. This company has a tall hierarchy where the management makes most of the decisions with little input from the employees when it comes to the game design. Communication commonly occurs through face-to-face communication within the different teams. Communication between departments is usually conducted through either email or through leads. The company motivates their employees by having yearly appraisals where people who work hard can get a raise. Company A also tend to have the occasional company outings, events, launch parties and holiday parties to keep the spirit up.

By the use of social media, forums and events the studio keeps in touch with its players. To get feedback from the players the studio does playtests and hands-on testing. This has not lead to any new features but tweaks to current ones. The company handles changes that are made to the product early in the project in an ad-hoc fashion, commonly through talking directly to each other. Later on in the project internal change requests will be made through a change log. The change will be discussed internally to ensure that the change is worth the time and effort. If the change will require too much effort, the employees try to come up with a quick solution in order to appease the requester.

The company is strict with its deadlines, if the management believe that the schedule is about to slip they encourage overtime to prevent that from happening. Company A does not tend to schedule time for retrospective meetings so reflections are carried out informally by speaking to one another and are not written down or recorded. Postmortems are usually created at the end of the project and sometimes after big milestones. The developers are self-organising and use JIRA as a tool to keep track of tasks. The employees select tasks from the backlog based on priority and the backlog is prioritised by the managers. The programmers are generalists, meaning that they work with all areas of the code, but are also given ownership over certain areas. This means that, for example, the programmer responsible for weapons will be the one who is notified when something needs to be done within that area.

The studio adheres to three out of the six OPP principles we look for. The criteria for *accommodating change* is not fulfilled since the developers get assigned extra tasks from outside of the plan directly from upper management. Company A does not increase development time in order to accommodate for this extra effort. Due to continuous crunching and no retrospectives the studio fail to fulfil the criteria for both *constant development pace* and *frequent reflection and improvement*.

B. Company B

Company B is a small sized company that develops indie games mainly for PC, PS4, Xbox One and the Wii U, but also for both iOS and Android. From this company we have interviewed a 3D/Environment Artist who have worked in the industry for one and a half year. This company has a flat hierarchy where communication commonly occurs casually through chatting with one another. The staff gets motivated by feeling included and informed about the public's feedback concerning their games.

The company keeps in touch with their players through social media as well as attending conventions. This communication has kept the players aware that issues such as bugs are being taken care off. It has also led to several feature updates. At Company B changes in features regularly happens in the middle of the project which requires people to do some rework and last minute bug fixes, which leads to working overtime in order to finish before the deadline. The studio has previously had plans on introducing retrospective meetings to their process. This has so far not happened.

The studio adheres to three out of the six OPP principles we look for. Company B does not fulfil the criteria for *accommodating change* since the studio does not increase the effort needed to implement new features. Instead the studio work overtime to reach the deadline which is also why the criterion for *constant development pace* is not fulfilled. As mentioned previously Company B does not reflect regularly which is why they do not fulfil the criteria for *frequent reflection and improvement*.

C. Company C

Company C is a micro sized games company developing quiz games for mobile devices, primarily iOS and Android. From this company we interviewed the CEO/Lead Game Designer who have worked in the industry for over 15 years. Because there are only five employees and a couple of freelancers the company has a flat hierarchy where the entire staff is part of the decision making process, although the CEO makes all the decisions together with the CTO. The studio has daily stand-ups in the morning to inform each other of any impediments the employees might have had. Company C motivates their staff by keeping them involved in the decision making process. This is done to ensure that everyone works towards a common goal and to make the employees believe in the company's vision.

Company C does not communicate with players during the development of their products. The studio keeps their play-testing in-house together with a few selected friends and family. After the product's release the studio keep in touch with their players through social media and email to get feedback and react to it accordingly. Company C handles change dynamically and informal; this feels natural to them since they are such a small studio. The studio understand that changes happen and tries to adapt to them as they appear. Company C does not have a set long term plan but rather a vision of where they want to end up. This makes it easier for them as a team to handle changes since there is no fixed plan to disrupt.

Since the studio is publishing themselves they tend to push deadlines if things do not go according to plan. Company C constantly have a build ready and fully functioning, distributing to their test flight several times a week. The company has a philosophy that 'until a feature is fully functioning on a phone it is not finished'. The tasks and deliveries are prioritised together with the entire team and put on a Kanban board. Their process has a higher focus on getting a constant work flow instead of having sprint deliveries every other week. Retrospectives are commonly quite ad-hoc and informal. Company C can not find a natural place for regular retrospectives in their development process since the studio does not have systematic deliveries.

Company C adheres to four out of the six OPP principles that we look for. The studio is self-publishing and has few external stakeholders that they communicate or collaborate with. Company C choose to do most of their play-testing internally and get most of their feedback from colleagues or friends, which is why the studio do not fulfil the criteria for *continual stakeholder communication and collaboration*. The criteria for *frequent reflections and improvement* is not fulfilled either because their reflections and improvements are very ad-hoc and infrequent. Our subject at Company C say this is due to them being such a small company with few employees.

D. Company D

Company D is a small sized company that currently develops an open-world winter sports game for PC and the PS4. From this company we interviewed an Environmental Artist who have worked in the industry for three years. This company has a flat hierarchy where communication commonly occurs in meetings every other week and chatting on a daily basis. Their staff gets motivated by free lunches at the end of each sprint along with freedom to, on the first day of the sprint, develop something fun from outside the sprint.

Company D communicate with their players through social media and the Steam community (Steam is a popular distribution platform for PC that include forums where developers can communicate directly to the players). This communication has led to new features getting implemented. Changes in features and requirements are handled differently depending on priority. If it is something game-breaking that the studio need urgently they will move other things out from the sprint.

Company D release when they have a set of features that the employees feel happy with. Since the studio rarely have any hard deadlines the employees can work freely. It is only when one of their partners have an event that the studio has more of a regular deadline. Company D use Scrum as their development process, and follow it quite strictly. Throughout the two week sprints the teams get their daily tasks from JIRA, which have been estimated in hours. The staff collect their own tasks at the beginning of their workday. When the sprint is finished the employees have a retrospective to reflect on what went well and what went bad for the last couple of weeks.

Company D adheres to five out of the six OPP principles that we look for. Since the studio has crunched twice in the last two years they do not fulfil the criteria of a *constant development pace*. Even though this is not frequent and that during this period some days have been regular hours the employees have had 14 to 16 hour days for up to months at a time.

IV. Results

In this section we will present the results that we gathered by analysing the interviews and postmortems.

A. Occurrence of crunch

According to our interviewees crunch is common within the industry. They all believe it to be a widespread phenomenon. One of the subjects from Company A describes crunch as being expected of you as an employee. Our interviewee at Company B thinks that crunch is something everyone in the games industry does. *"I know there's studios that say they don't but I think they're probably lying"* says one of the interviewees at Company A, he believes that all studios crunch to some extent.

Data from the postmortems indicates that crunch has been prominent within the games industry from the early 00's to the current date as seen in Figure 1. We can see that the data from 2001 and 2009 have the exact same correlation between postmortems read and number of crunches found. This suggests that the reports on crunch do not follow any trend. It is therefore not possible to say whether crunch time have become less relevant with time, even though the amount of crunch mentioned is falling from 2011 and no crunches were found 2013 or 2014. Interview data do however support that crunch is still occurring at this time.

As mentioned in Section II-A1 we have used a total of 78 postmortems for this study. 35 of which (45%) describe having crunched. From the postmortems we can see that the size of

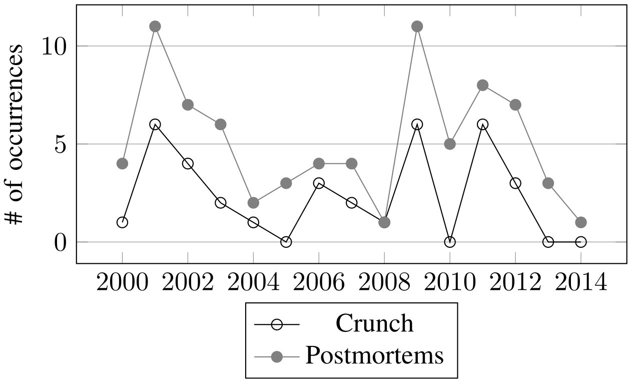


Fig. 1: Crunch occurrence per year (2000-2014)

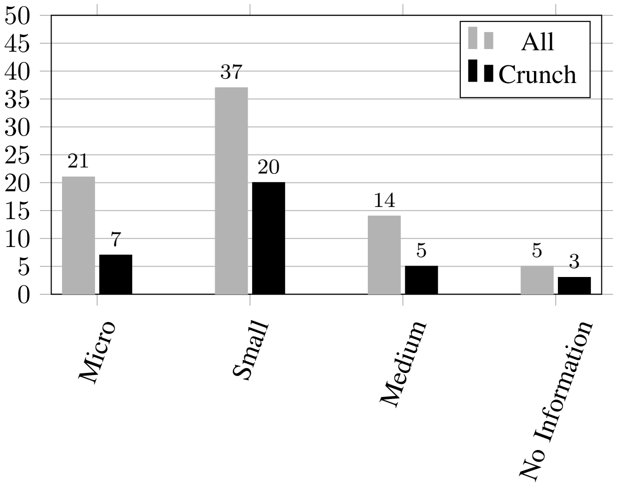


Fig. 2: Crunch occurrences based on game studio team sizes from the postmortems

a studio affect its likelihood to crunch. Small studios seem to be more prone to crunch (54% crunch) than both micro (33%) and medium (36%) sized studios. This can be seen in Figure 2 where the grey column represent the total number of postmortems per size and the black column represent how many postmortems mention crunch.

B. Four types of crunch

We have found that there are different types of crunch and what sets them apart is how long the period of crunch last, how often and when in development it occurs. We have categorised them as *Mini Crunches*, *Continuous Crunch*, *Delusional Crunch* and *Final Crunch*. There have also been cases where there has been too little information about the crunch for it to be definable. These cases have been categorised as indefinable. The various types of crunch can be defined as:

- *Continuous Crunch* — Crunch that goes on throughout either the whole project or large parts of the project until the game is complete. Often carried out because of unreasonable scheduling created early in the project that forces employees to work extra hard in order to get the game shipped according to schedule.
- *Mini Crunches* — Multiple crunches throughout the project that do not last longer than a fortnight each. Often conducted on developers own accord to ensure that their features does not get cut from milestone deliveries.
- *Final Crunch* — One big crunch for the last weeks, or months, of the project before the final deadline. It is an attempt to finish the game on time so that the studio do not have to push the deadline or risk releasing a bad or bug riddled game.
- *Delusional Crunch* — In the one case of delusional crunch that we found, the author of the postmortem explicitly said that the studio did not crunch to finish the game. However, later the author explain that the studio had to work overtime, late nights and weekends, because their financial situation was dire.

C. Reasons for crunch

The data gathered from the interviews shows that there is not just one reason for crunch. What seems to be the common factor in the interviews is that crunch is not explicitly forced upon the employees, but rather something that the employees would do because they want to. One of the subjects from Company A explains that *"I don't really want to finish and walk away from something when I could have done a better job."* Both subjects from Company A and Company D describe that there is some implicit pressure from management to crunch. The employees at the companies will for example be asked by management to finish their assigned features *"do your tasks, get your work done. As long as you get your work done I don't care if you're going home at 4, as long as you're getting everything sorted out that's fine"* as described by our interviewee from Company A.

Among the reasons given by the interviewees for crunch, deadline is the primary one. All of them mention it several times throughout the interviews. The subject at Company C says that: *"the only reason for crunch in a way is that you have a deadline that you need to meet"*. He explains that: *"if every project would be 'the game is done when it's done' you don't really need to crunch"*.

Managerial issues are something that most of the interviewees see as a reason. Ineffective techniques, poor time management and planning issues are problems argued to be the basis for crunch. One of the subjects from Company A describes this by saying: *"Poor time management and the inability to say no and plan. So poor time management, not on the part of the project managers but more on the part of people who make the decisions not having the idea of project management and saying 'You know what, I want to add this thing but that causes me to remove things'. That's the normal thing I think that you'd say 'I want to add a new feature to the game but I've already fully booked everyone up until the launch date. So I need to get rid of something in order to fit this in'. Instead it kind of feels as though it is expected, it is built in to the schedule that there will be periods of crunch where people are doing 50 to 60 hours allowing them to get maybe*

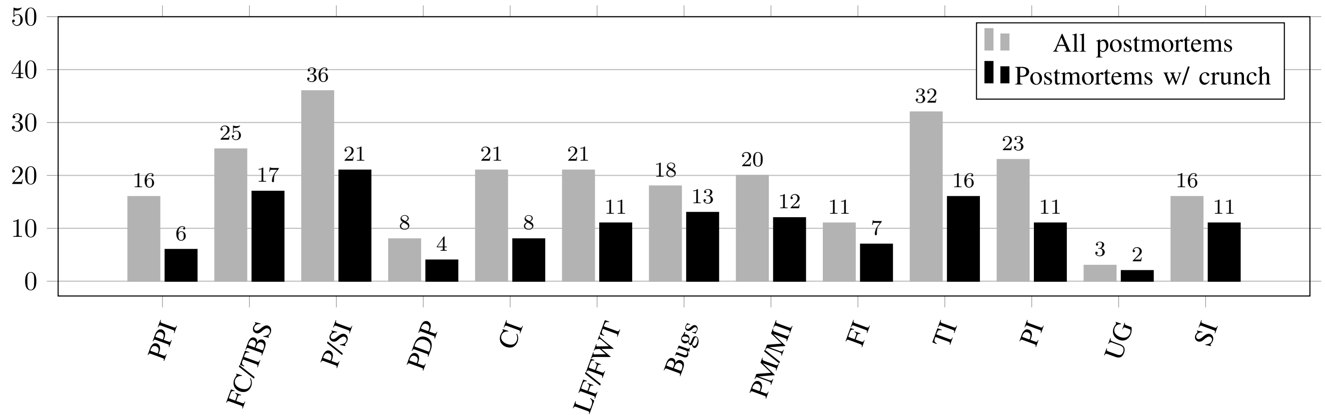


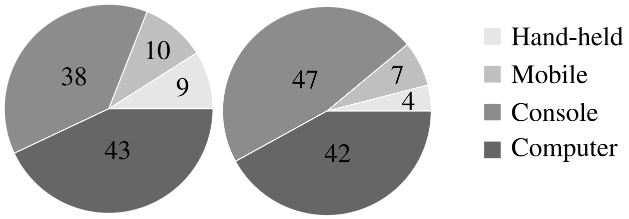
Fig. 3: Common issues within the industry based on postmortems. From left: *pre-production issues* (**PPI**), *feature creep/too big scope* (**FC/TBS**), *planning/scheduling issues* (**P/SI**), *publisher disagreement or pressure* (**PDP**), *communication issues* (**CI**), *lack of focus/focusing on the wrong things* (**LF/FWT**), *bugs*, *poor management/management issues* (**PM/MI**), *financial issues* (**FI**), *technical issues* (**TI**), *process issues* (**PI**), *unfun game* (**UG**) and *staffing issues* (**SI**)

an extra man-day, two or three man-days out of everyone per week or per month."

A reason raised by a few of the interviewees is feature creep. Our subject from Company B mentions this as a big contributor for crunching: *"I think that it is quite common that you add features too late in the process which will add to a lot of redoing, bug fixing and so on."* Having an unclear or too big scope at the beginning of the project is also mentioned to be a reason for crunching, as mentioned by our subject from Company D: *"I think sometimes it's just like the scope isn't defined properly."*

It is important to note that a lot of crunching is done on staff's own accord due to pride in their own work and being in the coding flow. Colleagues staying late can be a big influence in staying late yourself, in order to not let the group down. A subject from Company A describes this: *"the leads stay late and that can influence other people staying late as part of you know.. It's a group mentality thing"*, which shows that feeling part of the group and wanting to prove yourself as a productive member can lead to overtime. Another driver for working overtime on your own accord is pure passion, as our interviewee from Company C describes it: *"as creatives and passionate about this game and this is our legacy we want to be proud of it, we can do better. Let's use the last two-three months to just maximise everything, every effort into it to make it the best game it can ever be."*

Other issues mentioned less frequently throughout the interviews are lack of staff, financial issues, bugs and the game being underwhelming. Our subject from Company C mentions three of these issues by saying that: *"many times I think all the reasons are coming together but one is that your game is crap, you have a deadline and you know you can't delay past that deadline, you just need to get it working so you're in a sort of critical crisis mode. We need to fix the bugs, we need to get this game in a shipping state otherwise the company will shut down or whatever big risk is on the rising."*



(a) Distribution among all of the postmortems

(b) Distribution postmortems with crunch

Fig. 4: Platform distribution

We have found that there is a correlation between which platform a company develops for and whether or not the studio crunch. It is however unknown to us if this is due to technical or non-technical factors. The *Platform distribution* chart in Figure 4a shows how the platforms were distributed amongst the postmortems. The platform that has been developed for most is computers with 43%, console is at 38% whilst the mobile platform is at 10% and hand-held consoles is at 9%. The *Crunch per platform* char in Figure 4b shows the distribution of crunch per platform. Out of the 35 postmortems that mentioned crunch 4% developed for hand-held devices, 7% developed for mobile, 42% developed for computers and 47% developed for consoles. Even though there are fewer postmortems where games have been developed for consoles than computers, studios developing for consoles seem to be more inclined to crunch.

D. Issues in the industry

The postmortems have mentioned various issues which we have grouped together to form categories, as previously mentioned in Section II-C1. Figure 3 shows how many times an issue has occurred in the postmortems. Please refer to Section II-C1 for an explanation of the abbreviations.

As seen in Figure 3 the most common issues are *Planning/scheduling issues* (**P/SI**) and *Technical issues* (**TI**) while *Publisher disagreement or pressure* (**PDP**) and *Unfun game* (**UG**) are rarely mentioned. When it comes to issues for postmortems where crunch has been experienced, the most common issues are *Planning/scheduling issues* (**P/SI**) and *Feature creep/Too big scope* (**FC/TBS**).

E. Effects of crunch

From the postmortems and interviews it is clear that crunch has multiple effects on a project, primarily on the employees, the product and the schedule. As mentioned in section IV-B we have found four distinct types of crunch. We have assigned a type to all postmortems that were determined to crunch. The distribution of the types can be seen in Figure 5.

For the one case of *Delusional Crunch* the postmortem does not provide enough information about the effects of crunch on either the product, the staff or the schedule.

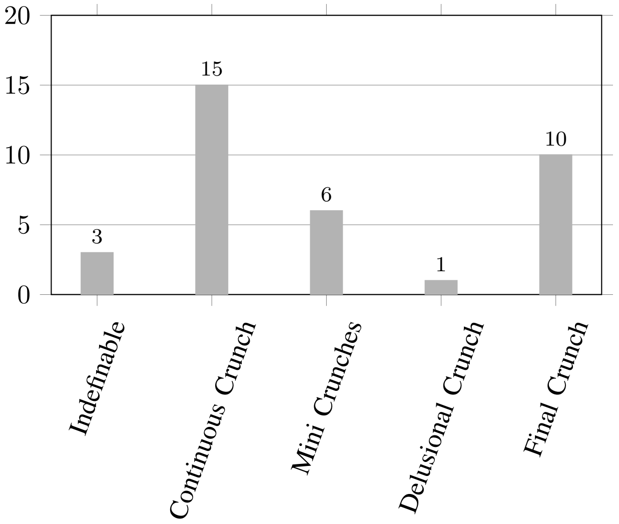


Fig. 5: Distribution of the various crunch types from the postmortems

According to the interviewees the positive impacts of crunch is that you get a lot of work done in a short amount of time. Our subject at Company C thinks crunch can be a good way of getting everyone focused on making the game as good as possible at the end of the project and that it can give *"this whole feeling of: together as a team we're doing the very best we can"*. All interviewees do however mention that the workforce gets burnt out and that it becomes difficult to have a good work-life balance. One of the subjects of Company A says that it can affect peoples relationships in the way that *"people's partners get very angry because suddenly their husbands or their wife isn't coming home at 5 o'clock, they're coming home at 8 and they're too tired to do anything."* He goes on saying that it also takes a physical and mental toll on you. The other subject at Company A says that *"it makes people hate what they do"*. According to our interviewee at Company B, crunch has a negative impact on software quality. She says that *"you get sloppy because of the stress"*.

1) Effects of Continuous Crunch: Continuous Crunch has led to non-requested features being implemented. Such features have been developers own pet-features but also unplanned features added late in development [13],[14]. There has also been implications on the quality of the product due to the crunch and the team having to make decisions without getting enough time to reflect on them [15]. In a few cases there have been shortcuts made in order to save time, which in the end has led to more bugs [16]. Overworked and tired employees creating more bugs while bug fixing is also prominent for this type of crunch [17],[18]. This has lead to employees having no work-life balance and a low quality of life [19],[20]. They have been feeling extremely overworked, pressured, stressed, frustrated and exhausted [10],[16],[17],[21],[22]. The teams morale has taken a toll while their bodies, relationships and spiritual well-being have been neglected [14],[15],[22],[23]. It has also led to the employees forgetting about the passion they once had for the game [18]. In a few cases the game has been late by up to two months and in a few cases the game was released on time, up to three weeks early.

2) Effects of Mini Crunches: The Mini Crunches led to an increased quality of the game [24],[25], ensured that features were not discarded [26] and that the game got some extra playtesting [27]. The employees felt overtasked, stressed and frustrated while the game got done in time [24],[26],[27],[28]. In one case it even got done a month ahead of the deadline [29].

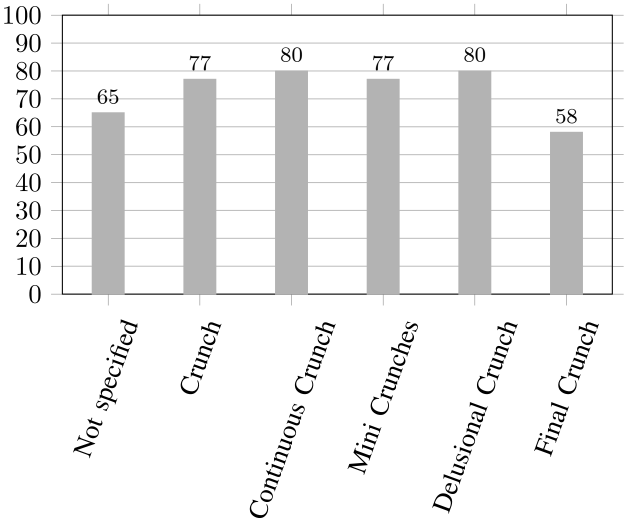


Fig. 6: Metacritic score for the games mentioned in the postmortems

3) Effects of Final Crunch: The projects where there has been Final Crunch have had mixed effects on the product. From improved frame rate [30], some extra features [31] and good master builds at release [32] to bug fixing generating more bugs [33] as well as compromised performance and quality [34],[35]. The staff felt pressured [36] and stressed [30],[35]. They got sleep deprived [31], exhausted and burnt out by all-nighters and weekends at the office [34],[37]. There have been a few cases where this has led to missed deadlines by up to five months [31],[35].

4) Effects of Delusional Crunch: In the one case of Delusional Crunch [38] there has been no clear indication of how their crunch impacted the product, employees or schedule.

5) Outcome of crunch: When looking at the Metacritic scores, see Figure 6, of games whose postmortem mention crunch with those who do not, we can see that the ones that crunched received a higher average score (77) than the ones who did not mention crunch (65). We can also see that the games where Continuous- or Delusional Crunch had occurred received the highest score of all the types, i.e. 80. Final Crunch did however have a significantly lower score than the other types, with an average of 58.

F. Agile

The studios fulfil the criteria of Sondararajan's OPP Framework to a varying degree. Table II shows that two of the studios, namely Company A & B, fulfil the same three criteria. Company D fulfils most of the criteria (5/6) while Company C manages to fulfil four.

TABLE II: Agile principle fulfilment

OPP Principle	**Company A**	**Company B**	**Company C**	**Company D**
Frequent delivery	X	X	X	X
Empowering teams	X	X	X	X
Accommodating change			X	X
Continual stakeholder participation	X	X		X
Frequent reflection				X
Constant development pace			X	

Frequent delivery of working software — is fulfilled by all four companies. Most of the studios have achieved this by frequent internal releases. Company D do however stand out since they continuously release new content to their players.

Empowering teams of motivated individuals — is also fulfilled by all four companies. This is achieved by sharing knowledge among the team(s), encouraging teams to self-organise and by keeping teams motivated through involvement and group activities. As described by our interviewee at Company C: *“having a chance of being involved I think motivates people”*. He continues with: *“not just involving them in the decisions but also I very much believe in keeping the team informed, so informing them on what is happening with the company”*.

Accommodating change — is fulfilled by two of the four companies. All four companies allow change in features to be a natural part of their development. What sets Company D and Company C apart from the rest is that they re-prioritise, drop other features or push deadlines in order to make room for the change. This is mentioned by our interviewee at Company C: *“if things don't go according to our short term plan we have right now we will probably push the release a little”*.

Continual stakeholder communication and collaboration — is fulfilled by three of the four companies. This has been achieved by keeping in touch with stakeholders, e.g. the players and publishers, using means such as social media, live demonstrations of the game as well as meetings to receive feedback and improve.

Frequent reflection and improvement — is only fulfilled by Company D. The studio achieved it by having retrospective meetings every other week to find out what worked and did not work for the past two weeks, aiming *“to work out kind of what causes are hindered in the future and trying to stop them from happening again.”*

Constant development pace — is only fulfilled by Company C. By barely having any overtime, apart from a few days a year (*“it's very rare that I, that anyone works more than their time, 8 hours full-time per day. Maybe a few days per year”*) and a mindset of reducing the possibility of crunch (*“So typically our philosophy is to, is that everyone is doing their best every day and the game will be done when it's done”*) they have achieved the criteria.

From the 78 postmortems we found that 14 described that the studios worked agile and 8 of those mentioned having crunched. This means that 18% of the postmortems describe agile projects and 57% of these mentioned crunch. In comparison postmortems that do not mention agile crunch in 42% of the cases. This suggests that agile may affect crunch.

V. Discussion

In this section we will discuss the results found in Section IV in an effort to try and answer our research questions.

A. Culture affecting people's willingness

Crunch has been a part of the games industry for a long time and much points towards it being a cultural thing. As described by de Peuter & Dyer [39] *“Those in long-term relationships, those who have children or want to start a family, or those who simply don't want to reduce the time of life to time spent at work, are ostensibly excluded from the game sector, or will find it tremendously difficult to commit to the ludicrous hours that can be expected of them. Enduring excessive hours without complaint is tied to the game industry's ‘hard work ethic”’*.

The culture of the games industry becomes even more clear when reading the work of Kerr [2]; *“While certain characteristics are shared with other media industries, including the sense that work can be fun, other characteristics, like the longer term contracts, acceptance of crunch time, lack of workforce diversity and ongoing loss of experienced staff may be more specific or at the very least more pronounced in this sector.”* This is further supported with the postmortems where on several occasions it is mentioned that crunch is expected. In some cases authors have spoken fondly about it. We see from the interviews that this influences them to crunch themselves. Often no one is actually telling them to work overtime, the employees crunch because it is expected of them and because everyone else is doing it. It is mentioned that guilt plays a big role in crunching, you do not want to let your team down.

B. A widespread phenomenon

The interview subjects indicate that crunch is something that all studios do, even though they might not acknowledge it. Out of the four studios we investigated through interviews 75% crunched. We believe this to be a percentage closer to reality than what is shown from the postmortems, where crunch is mentioned 45% of the time. We believe that this could be due to two things; Firstly the postmortems tend to have five things that went well and five that went wrong, meaning that crunch could be of lower priority and therefore not end up in the postmortem; Secondly we think it possible that the authors of the postmortem do not see crunch as a problem at all, but rather something that is required when making games.

C. Reasons for crunch

We have found that there are many reasons for crunch based on the interviews. As mentioned in Section IV there is mostly an implicit pressure to crunch rather than an explicit pressure from management, many crunch on their own accord because of pride, guilt or passion.

We have found that the three most prominent reasons for crunch are too big- or unclear scope, feature creep and deadlines. Deadlines are however the most frequently mentioned reason of the three. As argued by our subject at Company C: *"there would be no crunch if we wouldn't have a deadline."* He continues to reason that: *"Deadline implies a plan, so it is a planning problem."* Issues related to poor management and planning is frequently mentioned by most of the interviewees. This suggests that deadlines are highly coupled with other factors, such as planning and management issues. If avoiding crunch is an aim, it is important to set realistic deadlines.

The issues brought up in the postmortems can be linked to the reasons given for crunch by the interviewees. For example the biggest problem that is described by the postmortems that mentioned crunch is *planning or scheduling issues*. This correlates with the deadline-, management- and planning reasons mentioned in the interviews. The second biggest issue in the postmortems is *feature creep or too big scope*, which is the second most mentioned issue in the interviews as well. This indicates that these issues still exist within the industry today and are as prominent as they were a decade ago. The purpose of the postmortems is to help other studios to not repeat the same mistakes but to learn from them. It has been mentioned as a reoccurring theme within the postmortems that they do not heed the advice of others, e.g.: *"When Arrowhead was founded, we had a lot of good will from experienced developers throughout the Swedish game development industry who wanted to spill the beans on how to make the best game possible, and save us from the biggest pitfalls a new studio can fall into. We failed miserably at heeding their advice. It was almost as if we were told about the exact position of all the mines in a minefield and we still, like some sort of imbeciles, were compelled to step on them"* [10].

D. Agile affecting crunch

The agile manifesto was published in the early 00's which is when game studios formally started applying the agile methodologies. This does however not seem to have reduced the amount of crunch the studios do. As seen in Figure 1 it has been following the same pattern of peaks and valleys for the past decade.

Even though agile processes has been on the rise in popularity since it was introduced and its principles are seemingly appropriate for this type of industry, the crunch still persists. In the postmortems that mentioned that agile was used crunch occurred more often than the ones that did not mention agile. This might indicate that the games industry is either bad at applying agile methods or agile has no effect on mitigating crunch. We believe it to be the former. As shown in the interview data, studios that apply more agile principles crunch to a lesser extent. Company D and Company C crunch to a minuscule extent and as seen in Table II they fulfil the majority of the criteria. Company D and Company C have one criterion in common that they alone fulfil, i.e. accommodating change, which can be associated with feature creep. This is one of the biggest reasons for crunch that we have found in interviews and second most mentioned issue in the postmortems. This suggests that if studios accommodate for change they would lower the impact feature creep has on development.

E. Effects of crunch

1) Effects on product: As the results reported in Section IV-E show crunch has numerous impacts on a project, both good and bad. In terms of effects on the product, some had extra features implemented while others had an increase of bugs and a suffering quality. The most negative impacts on the product occurred in the cases where Continuous Crunch was conducted. We believe this is because overworked and burnt out employees have a tendency of making more mistakes and thus create more bugs and reduce the quality. Mini Crunches however only had positive impacts on the product. This could be attributed to that there is less pressure on the employees when they only need to work overtime for short iterations. This means that they get more rest in between crunches. This has been noted by Olson & Swenson as well: *"While overtime may not be a problem if it occurs infrequently, it can be a serious problem when it becomes the mode of operating"* [40].

2) Effects on schedule: We can see clear signs that crunch affects the schedule in a positive way, letting the studios release the game on time. The type of crunch that has the most positive impact on schedules are Mini Crunches. We believe that Mini Crunches are used to hold the schedule and catch up before it is too late. This belief is supported by two of the postmortems that released their game one month early and had utilised Mini Crunches during their development

[27],[29]. They explain that they kept a tight schedule by keeping the team involved, meeting often to re-estimate tasks as well as updating and maintaining the schedule. These factors, including conducting targeted Mini Crunches, could all be contributing to meeting the deadline. In contrast Final Crunch is the type that most frequently leads to an increased time-to-market than initially estimated. We believe this is because Final Crunch rarely has a realistic schedule, hence the late realisation of being behind.

3) Effects on employees: If we instead look at what impact crunch has on employees we observe just a few positive effects. These effects are explained by our subject at Company C as a sense of belonging to the group and working towards the same goal. Besides this it is clear that crunch has mostly negative impacts on employees, no matter the type. As described in Section IV-E, crunch has been found to make people hate what they do, work until exhaustion leading to a burnt out workforce and lowers the team morale. It has a major impact on your personal relationships as mentioned in the literature [1],[4],[39]. This is further explained by one of our interview subjects at Company A who says that it becomes difficult to maintain a relationship while working within the industry because of crunch time. It is mentioned to have an impact on the employees temper and health, where it gets to the point of employees neglecting their bodies in order to get a few extra hours of work every day. All these effects are frequently mentioned in both interviews and postmortems, but the most reoccurring impact is nonetheless stress.

When done in short intervals stress is unlikely to harm your health. If *"recurrent, prolonged, or very intense"* [40] it may however cause long term effects, both mental and physical.

Employees in the gaming industry sometimes receive vacation days after a project is complete as compensation for immense overtime. According to Olson & Swenson [40] this does not necessarily reduce stress, since the employees associate their workplace with stress. It will quickly return once they are back at work. It is therefore suggested that in order to keep the staffs health and performance at good levels they need to get time for recovery on a daily basis.

F. Outcome of crunch

Programmers who are exposed to overtime are more prone to make errors and become fatigued. Research by Akula & Cusick [3] show that stress lead to poor quality software and that avoiding stress will lead to increased productivity. If this is the case, how come studios that crunch produce games that receive better reviews? We can see a few explanations for this. Just because the postmortems have not mentioned crunch, does not necessarily mean that the studio did not crunch and therefore our data may be misleading. Another possible explanation is that studios that have Mini- or Continuous Crunch may do so in order to stay on schedule rather than falling behind, meaning that error handling is done iteratively during development so the final delivery is handled as smoothly as possible. This could mean that a lot of the initial bugs and problems with the game is handled early in the development. As found by Olson & Swenson: *"error removal is one of the most time-consuming phases of the product development lifecycle"* [40]. In contrast, games where there has been Final Crunch have received a significantly lower score than any other type of crunch and where there have not been any mentioning of crunch. We believe this is because they don't produce a realistic schedule in the pre-production stage. This leads the studios to rush the error handling in the end, creating a Final Crunch.

VI. Conclusion

It is safe to say that crunch is a widespread phenomenon within the games industry, but exactly how prevalent is hard to tell. Data gathered from interviews and postmortems suggest that at least 45% of game studios crunch. Based on the culture within the industry we believe this number to be larger in reality. Creative passion sets the tone for the industry, the well-being of the product goes before your personal welfare. Since people have a personal investment in the product they create, they blame themselves if it ends up badly. These factors make people willing to crunch, not only on their own accord but also begrudgingly.

Depending on your viewpoint crunch can be seen as either good or bad. But the industry in general seems to view it as a necessary evil. Common positive elements of crunch are meeting deadlines and adding more features to the game, while stress and the resulting negative impacts on personal health and product quality are the most prominent downsides of crunch.

Most commonly crunch occurs due to unrealistic schedules and feature creep. These are both issues that should be dealt with by adopting agile best practices. This is however not something we have noticed. Most companies which have adopted agile methods still crunched. This is likely due to not implementing the agile principles correctly, allowing the culture to dictate development pace and accommodating for change.

We recommend the games industry to introduce more realistic schedules when they plan games. If features must be added we suggest to accommodate them by removing something of less importance. If neither of these options are possible and crunching is the only option, we urge for Mini Crunches of no more than two weeks at a time. This type of crunch has the least negative impact on all areas, i.e. product, schedule and employees. Moreover we believe that the industry should strive to retain staff in order to maintain knowledge and thus reduce the risk of making the same mistake twice.

VII. Future research

It seems unlikely that crunch is the best way of releasing games according to schedule. We would therefore like to see research that focuses on finding new best practices for creating games. Maybe the principles and practices of agile development could be integrated to the games industry in a streamlined fashion. Or why not come up with something revolutionary?

Figure 3 suggests that crunch is more of an organisational issue rather than a technical one. In this study we have however not looked for this specifically and can therefore not draw any conclusions. We would like to see research being done in order to get confirmation if this might be the case, so that future efforts in mitigating crunch are focused on the right areas.

Acknowledgement

We would like to thank our supervisor Jan-Philipp Steghöfer for his encouragement, continuous feedback and guidance in making this paper the best it could be. We would also like to thank our interview subjects who helped us gain a deeper understanding of the games industry.

References

[1] F. Petrillo, M. Pimenta, F. Trindade, and C. Dietrich, “What went wrong? a survey of problems in game development,” *Computers in Entertainment (CIE)*, vol. 7, no. 1, p. 13, 2009.

[2] A. Kerr, “The culture of gamework,” pp. 225–236, 2011.

[3] B. Akula and J. Cusick, “Impact of overtime and stress on software quality,” in *4th International Symposium on Management, Engineering, and Informatics (MEI 2008), Orlando, Florida, USA*, 2008.

[4] N. Dyer-Witheford and G. S. de Peuter, “" ea spouse" and the crisis of video game labour: Enjoyment, exclusion, exploitation, and exodus,” *Canadian Journal of communication*, vol. 31, no. 3, 2006.

[5] J. Koutonen and M. Leppänen, *Agile Processes in Software Engineering and Extreme Programming: 14th International Conference, XP 2013, Vienna, Austria, June 3-7, 2013. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, ch. How Are Agile Methods and Practices Deployed in Video Game Development? A Survey into Finnish Game Studios, pp. 135–149.

[6] M. Washburn Jr, P. Sathiyanarayanan, M. Nagappan, T. Zimmermann, and C. Bird, “What went right and what went wrong: an analysis of 155 postmortems from game development,” in *Proceedings of the 38th International Conference on Software Engineering Companion*. ACM, 2016, pp. 280–289.

[7] P. Runeson and M. Höst, “Guidelines for conducting and reporting case study research in software engineering,” *Empirical software engineering*, vol. 14, no. 2, pp. 131–164, 2009.

[8] S. Soundararajan, J. D. Arthur, and O. Balci, “A methodology for assessing agile software development methods,” in *Agile Conference (AGILE), 2012*. IEEE, 2012, pp. 51–54.

[9] Metacritic. (1999). [Online]. Available: http://www.metacritic.com/

[10] J. Pilestedt. (2011) Postmortem: Arrowhead game studios’ magicka. [Online]. Available: http://www.gamasutra.com/view/feature/134840/postmortem_arrowhead_game_.php

[11] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, “Selecting empirical methods for software engineering research,” in *Guide to advanced empirical software engineering*. Springer, 2008, pp. 285–311.

[12] U. Loecher, “Small and medium-sized enterprises-delimitation and the european definition in the area of industrial business,” *European Business Review*, vol. 12, no. 5, pp. 261–264, 2000.

[13] J. Fristrom. (2013) Postmortem: Treyarch’s 2002 hit, spider-man. [Online]. Available: http://www.gamasutra.com/view/feature/194852/postmortem_treyarchs_2002_hit_.php

[14] B. Kijanka. (2002) Postmortem: Gas powered games’ dungeon siege. [Online]. Available: http://www.gamasutra.com/view/feature/131321/postmortem_gas_powered_games_.php

[15] M. Samyn and A. Harvey. (2010) Postmortem: Tale of tales’ the path. [Online]. Available: http://www.gamasutra.com/view/feature/134274/postmortem_tale_of_tales_the_path.php

[16] L. Hyvärinen and J. Kinnunen. (2010) Postmortem: Frozenbyte’s trine. [Online]. Available: http://www.gamasutra.com/view/feature/134198/postmortem_frozenbytes_trine.php

[17] P. Molyneux. (2001) Postmortem: Lionhead studios’ black & white. [Online]. Available: http://www.gamasutra.com/view/feature/131476/postmortem_lionhead_studios_.php

[18] P. Wilson. (2009) Postmortem: Realtime worlds’ crackdown. [Online]. Available: http://www.gamasutra.com/view/feature/132515/postmortem_realtime_worlds_.php

[19] J. Garvin, J. Ross, F. Gilbert, and C. Reese. (2012) Postmortem: Sony bend studio’s uncharted: Golden abyss. [Online]. Available: http://www.gamasutra.com/view/feature/181082/postmortem_sony_bend_studios_.php

[20] N. S. Entertainment. (2007) Postmortem: Naked sky entertainment’s roboblitz. [Online]. Available: http://www.gamasutra.com/view/feature/130154/postmortem_naked_sky_.php

[21] T. Train and B. Reynolds. (2003) Postmortem: Big huge games’ rise of nations. [Online]. Available: http://www.gamasutra.com/view/feature/131242/postmortem_big_huge_games_rise_.php

[22] A. Lindberg. (2012) Postmortem: Avalanche studios’ renegade ops. [Online]. Available: http://www.gamasutra.com/view/feature/171897/postmortem_avalanche_studios_.php

[23] I. Kuittinen and A. Raula. (2012) Postmortem: Housemarque’s outland. [Online]. Available: http://www.gamasutra.com/view/feature/169282/postmortem_housemarques_outland.php

[24] L. Currie. (2007) Postmortem: Blue fang’s zoo tycoon 2: Marine mania. [Online]. Available: http://www.gamasutra.com/view/feature/130165/postmortem_blue_fangs_zoo_tycoon_.php

[25] W. D. Hao. (2003) Postmortem: Tom clancy’s splinter cell. [Online]. Available: http://www.gamasutra.com/view/feature/131239/postmortem_tom_clancys_splinter_.php

[26] M. Fridley. (2013) Postmortem: Kingdoms of amalur: Reckoning. [Online]. Available: http://www.gamasutra.com/view/feature/197269/postmortem_kingdoms_of_amalur_.php

[27] D. Wu. (2002) Postmortem: Pseudo interactive’s cel damage. [Online]. Available: http://www.gamasutra.com/view/feature/131416/postmortem_pseudo_interactives_.php

[28] C. Bray and B. Fraser. (2012) Postmortem: Stardock entertainment and ironclad games’ sins of a solar empire: Rebellion. [Online]. Available: http://www.gamasutra.com/view/feature/179030/postmortem_stardock_entertainment_.php

[29] R. Bernstein. (2002) Postmortem: Frog city’s trade empires. [Online]. Available: http://www.gamasutra.com/view/feature/131422/postmortem_frog_citys_trade_.php

[30] R. Muzyka. (2001) Baldur’s gate ii: The anatomy of a sequel. [Online]. Available: http://www.gamasutra.com/view/feature/131493/baldurs_gate_ii_the_anatomy_of_a_.php

[31] J. Koning and F. Akker. (2009) Postmortem: Ronimo games’ swords & soldiers. [Online]. Available: http://www.gamasutra.com/view/feature/132618/postmortem_ronimo_games_swords__.php

[32] P. Miechowski. (2013) Postmortem: 11 bit studios’ anomaly warzone earth. [Online]. Available: http://www.gamasutra.com/view/feature/185535/postmortem_11_bit_studios_.php

[33] K. Sheller. (2011) Postmortem: High voltage software’s conduit 2. [Online]. Available: http://www.gamasutra.com/view/feature/134786/postmortem_high_voltage_.php

[34] J. Russell. (2009) Postmortem: 8monkey’s darkest of days. [Online]. Available: http://www.gamasutra.com/view/feature/132589/postmortem_8monkeys_darkest_of_.php

[35] T. Engel. (2002) Postmortem: Factor 5’s star wars rogue leader: Rogue squadron ii. [Online]. Available: http://www.gamasutra.com/view/feature/131402/postmortem_factor_5s_star_wars_.php

[36] T. Strzelczyk, G. Brol, and K. Komisarek. (2013) Postmortem: Vivid games’ real boxing. [Online]. Available: http://www.gamasutra.com/view/feature/185085/postmortem_vivid_games_real_.php

[37] M. Spanel and O. Spanel. (2001) Postmortem: Bohemia interactive studios’ operation flashpoint. [Online]. Available: http://www.gamasutra.com/view/feature/131427/postmortem_bohemia_interactive_.php

[38] C. Cleveland. (2013) Postmortem: Unknown worlds entertainment’s natural selection 2. [Online]. Available: http://www.gamasutra.com/view/feature/187299/postmortem_unknown_worlds_.php

[39] G. De Peuter and N. Dyer-Witheford, “A playful multitude? mobilising and counter-mobilising immaterial game labour,” *fibreculture*, vol. 5, no. 1, 2005.

[40] B. Olson and D. Swenson, “Overtime effects on project team effectiveness,” in *The Midwest Instruction and Computing Symposium*, 2011.