Crunch Time: The Reasons and Effects of Unpaid Overtime in the Games Industry

Henrik Edholm, Mikaela Lidström, Jan-Philipp Steghöfer
Dept. of Computer Science and Engineering
Chalmers | University of Gothenburg
Gothenburg, Sweden
henrik.edholm92@gmail.com, m.s.lidstrom@gmail.com,
jan-philipp.steghofer@gu.se

Håkan Burden
Viktoria Swedish ICT
Gothenburg, Sweden
hakan.burden@viktoria.se

***Abstract*—The games industry is notorious for its intense work ethics with uncompensated overtime and weekends at the office, also known as crunch or crunch time. Since crunch time is so common within the industry, is it possible that the benefits of crunch time outweigh the disadvantages? By studying postmortems and conducting interviews with employees in the industry, we aim to characterise crunch time and discover its effects on the industry. We provide a classification of crunch, i.e., four types of crunch which all have distinct characteristics and affect the product, employees and schedule in various ways. One of the crunch types stands out from the others by only having positive effects on product and schedule. A characteristic that all of the types have in common is an increase in stress levels amongst the employees. We identify a set of reasons for crunch and show that crunch is less pronounced in game studios where prioritisation of features is a regular practice.**

I. Introduction

Crunch time is a term used in the games industry to describe periods of extreme workload. During this period the employees at game studios often work 12 hours per day for several weeks, or even months. 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. According to Petrillo et al. [1] crunch primarily occurs before the final product delivery. 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".

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 cycle [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, causing many people to consider leaving the industry [1], [4]. These reasons should be incentives for the industry to want to change this practice.

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

It is still unclear at this point why the games industry continues to crunch despite evidence of its downsides. Do the benefits of crunch outweigh its disadvantages? Is it the culture of the industry that influences people to work this way? This research intends to shed light on the game development industry's crunch practices and understand how crunch affects different areas of development. To accomplish this, we address the following questions:

- RQ1: What are the most common reasons for crunch time?
- RQ2: What are the reported effects of crunch time on product, people and schedule?
- RQ3: Are there different types of crunch?
- RQ4: Does the games industry's culture affect people's willingness to crunch?
- RQ5: Does fulfilling agile principles have an effect on crunch time?

By giving an answer to the questions above, our contribution a) provides a better grasp of the effects crunch has on the industry; b) gives an introduction to multiple crunch types that can be used to define crunch; c) highlights the reasons given for crunch by the industry; d) explores if the game industry's culture encourages employees to crunch; and e) pinpoints what agile principles have an effect on crunch.

The paper is structured as follows: After this introduction, Section II provides a summary of how we gathered and assessed the data and presents potential validity threats. Section III provides organizational descriptions for the four game studios that we interviewed and assessed. In Section IV we present the results of our data collection and in Section V we discuss these results from the perspective of the research questions above. We finally conclude the paper in Section VI. Related work does not have it's own section, but is rather used continuously throughout the paper.

II. Methodology

This study was split into 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 extracted, analysed and compared the gathered data. The interview guide as well as the coding sheet used is available online[1].

A. Data Collection

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 of the state of the art. We noticed that other researchers in the field [1], [2], [4], [6] had used Gamasutra[2] as an information source. We therefore assessed that the website was a reliable source. Thus secondly, we selected 78 out of the 180 postmortems published on Gamasutra after the release of the agile manifesto. These were all postmortems that fulfilled our inclusion & exclusion criteria: critically reflected on their development process rather than promoting themselves; were written by studios with more than five employees, to reduce the chance of assessing the development of hobby projects; were written after the agile manifesto was published, since we can not assume that people in the industry had an awareness of the agile mindset before this date. 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 and what went wrong during the development. It aims to acknowledge the issues of development in a constructive way so that the team can learn and improve for the next project. Postmortems are often shared on dedicated websites, such as Gamasutra, to disseminate knowledge within the industry.

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

Through the interviews we expected to get an insight into why the industry crunches, what impact crunch has on the employee and the product, and 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 only 15 minutes up to 40. In the postmortems we looked at which issues the game studios mentioned 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.

B. Data Extraction

For the third and final phase two of the researchers extracted the data gathered from the postmortems and the interviews which we later analysed and compared. The analysis and comparison can be seen in Sections IV and V.

When extracting data from the postmortems we read through them thoroughly and identified issues relevant to crunch that the game studios had experienced. The issues were then grouped thematically and labelled as *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*.

After the interviews were conducted we transcribed them word-by-word in order to not alter the point that the interviewee was trying to make. Then we extracted data from our transcripts and created a coding sheet to get a better overview of the information found. The extracts were coded, grouped thematically and each theme given a new code. The extracted codes were emergent and iteratively defined to arrive at the final set of codes.

C. Assessing agile

In order to understand the root cause of why the interviewed game studios crunch, we looked into the company's development process. We wanted to see if there can be any correlation between the level of agility and the amount of crunch. We have used the **O**bjective, **P**rinciples and **P**ractices (*OPP*) Framework by Soundararajan's et al. [7] to assess how much the companies adhere to the agile principles. From the principles given by Soundararajan et al. we have chosen the following six that we indirectly asked about during our interviews: *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 as a preemptive precaution since we determined that the interviewees might get into a defensive stance and would potentially provide biased answers if asked about these issues. It is unlikely that an employee of a technology company would say 'no' when asked if their employer *aims for technical excellence* or *strives for customer satisfaction*. We believe that the implicit pressure to always promote the organization you work for would cause the interviewees to provide us with biased answers, which in turn would invalidate parts of our findings.

D. Validity Threats

The criteria for validity are based on Easterbrook et al. [8].

Construct validity is a threat if the research design is vague and up for interpretation. The research questions metrics

[1] https://gubox.box.com/v/2017-ICSE-SEIP-Crunch-Time

[2] http://www.gamasutra.com

e.g., the frequency of reasons in RQ1, 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 lacks 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 data frequently, to ensure that the two researchers responsible for analysing the data 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. We have also designed our interview questions to be as neutral as possible in order to not put the interviewees in a defensive stance. This was achieved by pre-emptively excluding questions regarding three of the OPP Frameworks principles (*technical excellence*, *simplicity* and *striving for customer satisfaction*).

It is likely that some postmortems do not mention whether or not the studio crunched during the development even though this might have been the case. We suggest two possible explanations to this. Firstly, crunch may not be considered one of the top five things that went right or wrong. Secondly, crunch is not seen as something worth mentioning because it is part of the games industry's work culture. This is further discussed in Section V-A.

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 standpoint. 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. Since all of our interviewees are from northern Europe there is a risk that the study would yield different results if a researcher were to interview subjects from parts of the world with a different work culture. 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 [9]:

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

We believe the companies to be a good subset because they are so diverse. Company A, B and D release for the big platforms, i.e., Xbox One, Playstation 4 (PS4) and Personal Computer (PC), whilst Company C releases games for iOS and Android. Not only does this mean that they release in different ways, e.g., hard copies and through online platforms, but also produce different types of products. Furthermore, the companies corporate cultures vastly differ.

A. Company A

Company A is a medium-sized games company developing for Xbox One, PS4 and 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 managers. The company motivates their employees by having yearly appraisals where people who work hard can get a raise. Company A also tends to have occasional company outings, events, launch 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 fulfil the request.

The company is strict with its deadlines. If management believes 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 without any transcription. Postmortems are usually created at the end of the project and sometimes after big milestones. The developers are self-organising and use JIRA[3], a project management tool, to keep track of tasks. The employees select tasks from

[3] https://www.atlassian.com/software/jira

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 fails 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. We have interviewed a 3D/Environment Artist who has worked in the industry for one and a half years. 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 happen 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 employees 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. We interviewed the company's CEO/Lead Game Designer who has 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 stays in touch with their players through social media and email to get feedback and react to it accordingly. Company C handles change dynamically and informally. This feels natural to them since they are such a small studio. The studio understands 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 has a build ready and fully functioning, distributing it 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 than 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 chooses to do most of their play-testing internally and get most of their feedback from colleagues or friends, which is why the studio does not fulfil the criteria for *continual stakeholder communication and collaboration*. The criteria for *frequent reflections and improvement* are not fulfilled either because their reflections and improvements are very ad-hoc and infrequent. Our subject at Company C says 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 both PC and PS4. From this company we interviewed an Environmental Artist who has worked in the industry for three years. Company D 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 communicates with their players through social media and the Steam community (Steam is a popular distribution platform for PC that includes forums where developers can communicate directly with their 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 needs urgently, other things will be moved out of the sprint.

Company D releases when they have a set of features that the employees feel happy with. Since the studio rarely has any hard deadlines the employees can work freely. It is only when one of their partners has an event that the studio has more of a regular deadline. Company D uses Scrum as its development process, and follows it quite strictly. Throughout the two week sprints, the teams get their daily tasks from JIRA. The tasks are estimated in man 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 a frequent event and though some days have regular hours during this period, the employees worked for 14 to 16 hours a day for up to months at a time.

IV. Looting the data

In this section we will present the results that we gathered by analysing the interviews and postmortems. The data for the research questions are presented in this section and their answers can be found in Section V. We do however answer RQ3 in Section IV-B.

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. *"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. Our interviewee at Company B thinks that crunch is something everyone in the games industry does.

Data from the postmortems indicates that crunch has been prominent within the games industry from the early 2000's to the current date as seen in Figure 1. We can see that the data from 2001 and 2009 shows the exact same ratio between postmortems read and number of crunches found. From this data alone it is not possible to say whether crunch time has become less relevant since the agile manifesto was published. Even though there are no cases of crunch in the postmortems for the last two years, interview data support that crunch is still commonly occurring.

We have used a total of 78 postmortems for this study, 35 of which (45%) mention crunch time. From the postmortems we can see that the size of a studio affects its likelihood to crunch. Small studios seem to be more prone to crunch (54% crunch) than both micro- (33%) and medium-sized (36%) studios. This can be seen in Figure 2 where the grey column represents the total number of postmortems per size and the black column represent how many postmortems mention crunch.

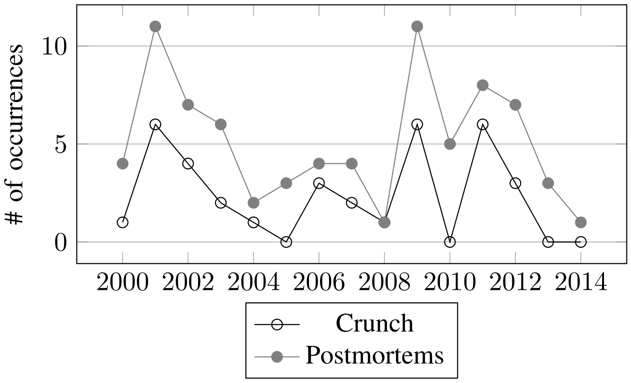

Fig. 1. Crunch occurrence per year (2000-2014)

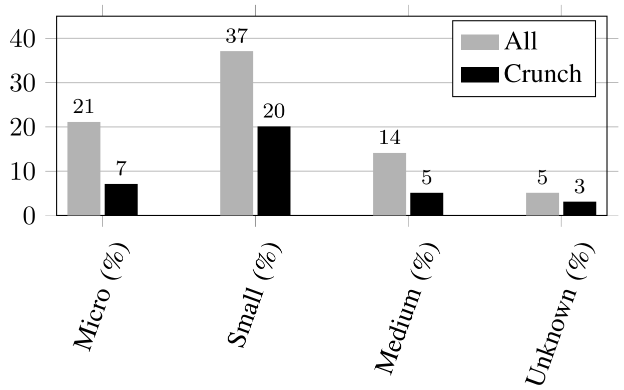

Fig. 2. Crunch occurrences based on game studio team sizes from the postmortems

B. Four types of crunch

We have found that there are different types of crunch What sets them apart is how long the period of crunch lasts and how often and when in development it occurs. We have categorised the occurrences of crunch time as *Continuous Crunch*, *Final Crunch*, *Mini Crunches* and *Delusional Crunch*. There have also been three cases with too little information about the crunch for it to be definable. These have been disregarded. The various types of crunch are defined as follows (number of occurrences in parantheses):

Continuous Crunch (15) — Crunch that goes on throughout the majority 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.

Final Crunch (10) — One big crunch for the last couple of weeks of the project before the final deadline. It is an attempt to finish the game on time so that the studio does not have to push the deadline or risk releasing a bad or bug-riddled game.

Mini Crunches (6) — 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 do not get cut from milestone deliveries.

Delusional Crunch (1) — When the studio truly believes that they are not crunching, yet the staff works overtime and spends late nights at the office. This type of crunch is not based on a time aspect, but rather demonstrates the industry culture, and can therefore exist in combination with one of the other types.

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 choose to do themselves. However, both subjects from Company A and 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 at any cost: *"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 four, 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… 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… 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 an extra man-day, two or three man-days out of everyone per week or per month."*

A reason raised by two 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 excessive scope at the beginning of the project is also mentioned to be a reason for crunching.

It is important to note that a lot of crunching is done on the 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"* This 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 [sic] 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."*

TABLE I
AGILE PRINCIPLE FULFILMENT

OPP	**Company**			
Principle	**A**	**B**	**C**	**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	

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."*

D. Issues in the industry

The postmortems have mentioned various issues which we have grouped together to form categories of common issues. These issues are not explicitly reasons for crunch. We can however see a correlation between the issues presented in the postmortems and the reasons for crunch mentioned in the interviews, as seen in Section IV-C. The correlation between these will be further explained in Section V-B.

Figure 3 shows how many times an issue has occurred in the postmortems. 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/Excessive scope* (**FC/ES**).

E. Agile principles affecting crunch

The studios interviewed fulfil the criteria of Soundararajan's OPP Framework to a varying degree. Table I 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.

The agile principle that seems to be the most prominent when it comes to mitigating crunch is Accommodating change. This principle is fulfilled by two out of the four companies. All four companies allow change in features to be a natural part of their development. What sets Company C and Company D apart from the rest is that they re-prioritise, drop other features or push deadlines in order to make room for the change. Handling change and addition of new features this

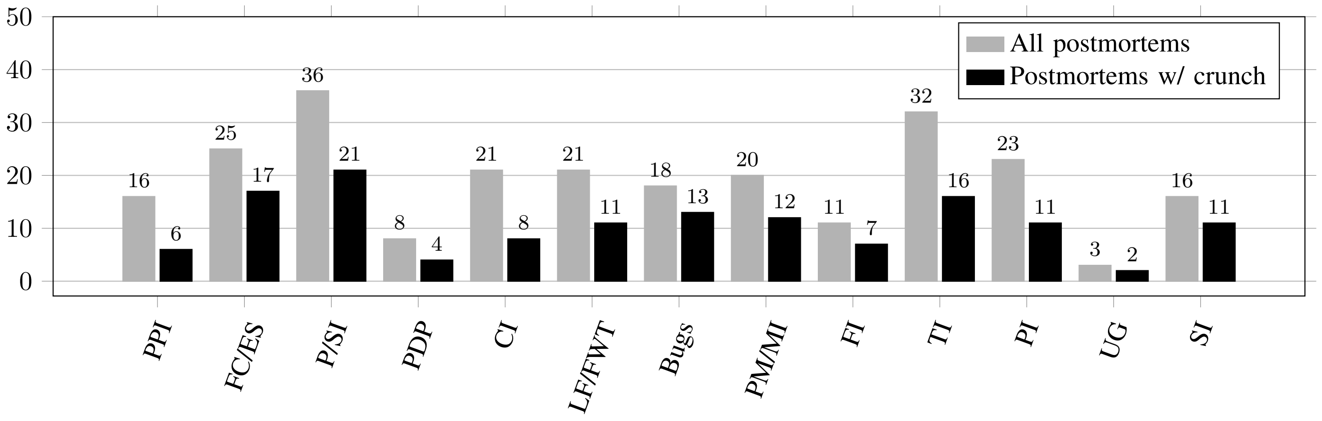

Fig. 3. Common issues within the industry based on postmortems. From left: *pre-production issues* (**PPI**), *feature creep/excessive scope* (**FC/ES**), *planning/scheduling issues* (**P/SI**), *publisher disagreement or pressure* (**PDP**), *communication issues* (**CI**), *lack of focus/focusing 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**)

way means that the studios reduce feature creep, which in turn lead to less scheduling issues. As seen in Section IV-D these are the most common issues that the game studios from the postmortems that crunch have.

V. Solving the Final Puzzle

In this section we will discuss the results found in section IV in an effort to try and answer our research questions. In Section V-A we answer RQ4, in Section V-B we answer RQ1 and in Section V-C we answer RQ2 and RQ5.

A. Overview of the crunch culture

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 [10] *"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 by 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 employees to crunch on their own volition. Often no one is explicitly telling them to work overtime but employees crunch nevertheless 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.

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.

B. 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. As outlined above, many crunch on their own accord because of pride, guilt or passion.

We have found that the three most prominent reasons for crunch are excessive 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. . . Deadline implies a plan, so it is a planning problem."* Issues related to poor management and planning are 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 excessive 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 goodwill 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"* [11].

C. Effects of crunch

This section starts by providing an overview of the effects of crunch. A more thorough analysis is then presented, first from the perspective of the identified crunch types and then from the perspective of product, people as well as schedule. The data is insufficient to provide quantifiable measures of the effects, but there are evident qualitative trends.

1) Overview of the effects: 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.

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.

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"*.

2) 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 [12], [13]. 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 [14]. In a few cases there have been shortcuts made in order to save time, which in the end has led to more bugs [15]. Overworked and tired employees creating more bugs while bug fixing is also prominent for this type of crunch [16], [17]. This has lead to employees having a skewed work-life balance and a low quality of life [18], [19]. They have been feeling extremely overworked, pressured, stressed, frustrated and exhausted [11], [15], [16], [20], [21]. The teams morale has taken a toll while their bodies, relationships, and spiritual well-being have been neglected [13], [14], [21], [22]. It has also caused employees to forget the passion they once had for the game [17]. In a few cases the game was delayed by up to two months and in others the game was released on time, up to three weeks early.

3) Effects of Mini Crunches: The Mini Crunches led to an increased quality of the game [23], [24], ensured that features were not discarded [25], and that the game got some extra playtesting [26]. The employees felt overtasked, stressed, and frustrated while the game was finished on time [23], [25], [26], [27]. In one case the game was even released a month ahead of the deadline [28].

4) Effects of Final Crunch: The projects with Final Crunch have had mixed effects on the product. Reported outcomes range from improved frame rate [29], some extra features [30], and good master builds at release [31] to bug fixing generating more bugs [32] as well as compromised performance and quality [33], [34]. The staff felt pressured [35] and stressed [29], [34]. Employees were sleep deprived [30], exhausted and burnt out by all-nighters and weekends at the office [33], [36]. In two of the cases Final crunch failed to save the project from pushing the deadline [30], [34]. Instead the projects ran late by up to five months.

5) Effects of Delusional Crunch: In the one case of Delusional Crunch [37] there has been no clear indication of how their crunch impacted the product, employees or schedule.

6) Effects on product: As the results reported in Section V-C show, crunch has both good and bad impacts on a project. In terms of effects on the product, some companies managed to implement extra features while others saw an increase of bugs and suffering quality. The most negative impacts on the product occurred in the cases where Continuous Crunch was conducted. We believe this is due to the fact that overworked and burnt out employees have a tendency of making more mistakes and thus create more bugs and reduce the quality. This was also noted by Akula & Cusick [3] whose research shows that stress lead to poor quality software and that avoiding stress will lead to increased productivity. Mini Crunches however only had positive impacts on the product. This could be attributed to the fact 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"* [38].

7) 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 stay on 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 [26],[28]. 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 when a project has an unrealistic schedule, Final crunch is often conducted in order to catch up with the deadline. This often leads to an increased time-to-market which we think can be attributed to the late realisation of being behind.

8) 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 V-C, crunch has been found to make people hate what they do, work until exhaustion leading to a burnt out workforce and lowering the team morale. It has a major impact on your personal relationships as mentioned in the literature [1],[4],[10]. 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"* [38] 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 overtime. According to Olson & Swenson [38] this does not necessarily reduce stress. Since the employees associate their workplace with stress they will quickly regain the same stress levels once they get back to work. It is therefore suggested that recovery needs to happen daily in order to keep the staff's health and performance at good levels.

9) 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 had an impact on the amount of crunch the studios do. As seen in Figure 1 the ratios between postmortems read and the occurrence of crunch has been almost identical for the past decade.

Even though agile processes have been rising in popularity since they were introduced and agile principles are seemingly appropriate for this type of industry, crunch time still persists. As shown in the interview data, studios that apply more agile principles crunch to a lesser extent. Company C and Company D crunch to a minuscule extent and as seen in Table I they fulfil the majority of the criteria. Company C and Company D 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 change they would lower the impact feature creep has on development.

VI. Conclusion

It is safe to say that crunch is a widespread phenomenon within the games industry, but exactly how prevalent it is cannot be said with certainty. Data gathered from interviews and postmortems suggests that at least a sizeable portion of game studios crunch. Based on the culture within the industry we believe that a majority of game studios applies the practice. Creative passion sets the tone for the industry and the well-being of the product is prioritised over employee 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 not accommodating for change. Such problems can also be witnessed in software companies outside the gaming industry [39].

We recommend that the games industry 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.

Figure 3 suggests that crunch is more of an organisational issue than a technical one. In this study we have pinpointed the organisational issues that are likely to lead to crunch. We would like to see research that confirms that these issues are really reasons for crunch. This research could aid future efforts of crunch mitigation by setting the focus on the right areas. 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, in particular accommodating for change and constant development pace, could be integrated to the games industry in a streamlined fashion.

We see a quantitative study focusing on measuring the negative impacts of crunch as the next step to pinpoint where effort should be directed in order to minimize crunch.

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] 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.

[8] 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.

[9] 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.

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

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

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

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

[14] 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

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

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

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

[18] 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

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

[20] 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

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

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

[23] 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

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

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

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

[27] 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

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

[29] 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

[30] 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

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

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

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

[34] C. Corry. (2001) Postmortem: Lucas arts' star wars starfighter. [Online]. Available: http://www.gamasutra.com/view/feature/131456/postmortem_lucas_arts_star_wars_.php

[35] 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

[36] 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

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

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

[39] J. Babb, R. Hoda, and J. Nørbjerg, "Barriers to Learning in Agile Software Development Projects," in *Agile Processes in Software Engineering and Extreme Programming*, ser. Lecture Notes in Business Information Processing, H. Baumeister and B. Weber, Eds., vol. 149. Springer Berlin Heidelberg, 2013, pp. 1–15.