CRUNCH: THE FAILURE OF PROJECT MANAGEMENT
IN THE GAMES INDUSTRY

by

Ben Worrall, BA

Dissertation submitted in partial fulfilment for the of Degree of
Master of Science in Programme and Project Management

WMG, University of Warwick

Submitted August, 2012

For completion by Moderators:
QA procedures complete; passed to Exam Board

Signed:                                             Date:
ABSTRACT

Context: During game development, a project may encounter difficulties that require the project to 'crunch'. This involves working extremely long hours for extended periods of time, sometimes for the entire duration of production - hours which are often not compensated financially or otherwise. This has considerable impact on the people who must work such hours, and ultimately on the project itself.

Objective: This paper investigates the phenomenon with a view to describing it completely to facilitate further study, determining the root cause of the issue in order to understand why the problem still exists in the face of modern project management tools and methodologies.

Method: A phenomenological study is conducted based on data from 34 previously completed projects and from 22 anonymous anecdotal accounts detailing encounters with crunch. An aetiological analysis is performed to determine causation and to enable relevant solutions to be applied.

Results: The issue of crunch is shown to be more complex than previously thought. Lack of previous academic study on the subject has led to preconceptions about causation within the games industry, many of which prove to be inaccurate. The phenomenology allows for the creation of clear descriptions of the textural experiences of crunch and their structure, giving a previously unseen insight into how it affects the team and the project. The aetiological study shows ambitious scoping and other project management issues to be the most legitimate causes of crunch.

Conclusion: The phenomenology shows that the impact of crunch on the team is felt physically and psychologically beyond the current project, affecting decision making in future projects. The aetiological study dissociates from crunch widely held beliefs of causation, illuminating deep and complex inter-relationships that prevent contemporary project management from having any effect on crunch beyond mitigation. These results demonstrate the imperative for further study on crunch from the field of project management, and that the standard of project management in the games industry needs to be raised.
Table of Contents

1. INTRODUCTION.................................................................................................................. 8
   1.1 Crunch.......................................................................................................................... 8
   1.2 RQ1 - What is Crunch?................................................................................................. 9
       Aim 1 – Develop a Phenomenologically Sound Description of Crunch......................... 9
   1.3 RQ2 – What Causes Crunch?........................................................................................ 9
       Aim 2 - Determine the Root Cause of Crunch Through Aetiology............................... 9
   1.4 RQ3 – Why is Crunch Still a Problem for Modern Project Management?.................... 10
   1.5 Objectives.................................................................................................................... 10
   1.6 Rationale..................................................................................................................... 11

2. LITERATURE REVIEW......................................................................................................... 13
   2.1 Etymology..................................................................................................................... 13
   2.2 Industrial Research....................................................................................................... 15
   2.3 Game Production.......................................................................................................... 17
   2.4 Contemporary Sources................................................................................................. 20
   2.5 Overview...................................................................................................................... 23

3. METHODOLOGY.................................................................................................................. 26
   3.1 Data Required by Aims................................................................................................. 26
   3.2 Method by Previous Research....................................................................................... 27
       Anecdotal Evidence........................................................................................................ 27
       Case Studies.................................................................................................................. 28
       Interviews...................................................................................................................... 28
       Literature Review......................................................................................................... 28
       Postmortems................................................................................................................ 29
       Surveys.......................................................................................................................... 29
   3.3 Data Set........................................................................................................................ 29
       3.3.1 Postmortems.......................................................................................................... 29
           Assumptions............................................................................................................... 30
           Limitations............................................................................................................... 30
           Expectations............................................................................................................ 30
       3.3.2 Anecdotal Evidence............................................................................................... 31
           Assumptions............................................................................................................... 32
           Limitations............................................................................................................... 32
           Expectations............................................................................................................ 32
   3.4 Coding Schedule for Content Analysis....................................................................... 32
       Description of Crunch.................................................................................................... 32
       Consequences............................................................................................................. 33
       Attitude Towards Crunch............................................................................................ 33
       Causes........................................................................................................................ 34
       Project Management..................................................................................................... 34
   3.5 Phenomenological Study............................................................................................... 35
   3.6 Aetiological Study......................................................................................................... 36
       Strength......................................................................................................................... 36
       Consistency.................................................................................................................... 37
       Specificity...................................................................................................................... 37
       Temporality................................................................................................................... 37
6. DISCUSSION

6.1 What is Crunch?

6.2 What Causes Crunch?

6.3 Why is Crunch Still a Problem for Modern Project Management?

6.4 Crunch Est, Igitur Crunch Existit.
1. INTRODUCTION

"The thing is, very few people ever offer practical advice on what you can do to avoid crunch, outside of 'better scheduling'."

– Paul Ross, CTO Criterion Games

1.1 Crunch

This paper looks at the phenomenon of crunch as it exists in the games industry.

Like its parent, the software industry, game development is largely project-oriented and as such it encounters many of the same issues faced by its parent. There is no shortage of research highlighting the massive failure rate of software projects, and as a means to counter this the games industry has, over several years, developed what is almost a dependency on crunch.

Crunch is essentially a period of intense labour around a milestone where developers can work in excess of one hundred hours a week in order to prevent the game from missing its release date. Were this a seldom, brief occurrence it may not be an issue, but crunch has been seen to run for several months at a time, up to and including the entire development-cycle of some games.

This takes an enormous toll on those working such hours both physically and mentally, as well as putting severe strain on their personal relationships. Commercially, this then feeds back into the quality of the finished game, as while crunch can fit the required man-hours into a shorter time-scale, the quality of work done in that time is dramatically reduced.

This strain causes a high degree of employee migration between companies as they seek to improve their working conditions, which serves to transfer those same working habits and ideals around the industry, perpetuating the practice. This also then causes many to leave the industry altogether, resulting in a huge loss of skills and experience, which is overcome by recruiting young and enthusiastic graduates with few commitments and who are more tolerant of working long hours in order to be a part of an industry they love.

But games are projects, largely run using project management tools and

1 (Ross, 2011)
1. INTRODUCTION

methodologies, and project management is no stranger to fire-fighting around milestones. Software development even has its own branch of methodology which has been used to good effect in the games industry. Studies show however that 56% of reported problems in the industry are related to project management; over-ambitious scope, feature creep, poor resource allocation.

Part of the problem is that crunch seems to be poorly understood. Outside the games industry few know what the term means (academia included), and within, despite it being regarded as a genuine problem and a threat to the industry, it is often seen as the natural way of doing business.

This paper is the first to conduct an in-depth study of crunch as it exists today through description and causation, leading to the understanding of why it is such an issue in the face of modern development and management theory.

1.2 RQ1 - What is Crunch?

The first step is to determine what the current state of knowledge is on the subject, before explicating the phenomenon.

Aim 1 – Develop a Phenomenologically Sound Description of Crunch

In order to aid the games industry and the field of project management in tackling crunch, this paper will define crunch phenomenologically to encourage further study by professional and academic institutions. This includes explication of how crunch manifests itself within a project and how it is viewed and experienced by those who encounter it. Only once we completely understand what crunch is can we begin to look for a root cause.

1.3 RQ2 – What Causes Crunch?

By discovering the most legitimate root cause of crunch in the industry it will be possible to recommend project management tools and methodologies which can be used to solve what is demonstrably a project management problem.

Aim 2 - Determine the Root Cause of Crunch Through Aetiology

In order to effectively make recommendations for tackling crunch we need to be sure that the causes uncovered are genuine root causes and not just incidental. Aetiology is the
1. INTRODUCTION

Study of causation and the determination of the validity of a potential cause, and while its common use is in medicine, the analytical framework can be adapted for use in other circumstances. Causation is important because even once crunch has been explicitly described, that description essentially makes up the symptoms of crunch. Treatment of the symptoms is a mitigation strategy that might make crunch easier to endure but will not prevent its recurrence.

1.4 RQ3 – Why is Crunch Still a Problem for Modern Project Management?

If crunch is a project management issue, there is the question of why project management with its numerous tools and methodologies, some of which are specific to software development, has so far failed to remedy the issue.

If a cause of crunch can be determined and a relevant solution found, it stands to reason that the absence of that solution explains why crunch still exists. The framework of the solution, its purpose, make-up and associated field of practice will give indications of why it was not previously applied. This gives insight into the success of project management application in the industry, perhaps indicating the need for alternative approaches or alteration and adaptation of the original methods to the games industry.

1.5 Objectives

In order to satisfy the above aims, this paper pursues the following:

1. Critically analyse the academic literature surrounding the target industry and subject to determine the current state of knowledge on crunch

2. Formulate an effective research methodology for data collection

3. Describe crunch phenomenologically through the use of horizontalisation and the determination of invariant constituents to create composite textural and structural descriptions

4. Determine the aetiology of crunch using the Bradford-Hill criteria for causation

5. Propose the application of relevant project management devices to remedy

6. Explore how their current absence explains the state of project management in the industry
1. INTRODUCTION

7. Demonstrate the imperative for further study

1.6 Rationale

It is important for this paper to provide the reader with a clear understanding of what crunch is, as without this knowledge it would be unscientific to make recommendations on how to overcome it. This definition needs not only to detail the symptomatic structure of the phenomenon (those elements of it which can be witnessed and experienced by an observer or a participant) but also how it is perceived as a phenomenon by those who experience it.

The literature review shows the current state of knowledge on the subject; which fields have attempted to study it, in what context and to what degree. Accordingly, the review critically analyses both conventional and contemporary sources of literature to develop as complete an understanding as possible.

Combining this broad study with relevant data from a range of sources will give a complete description of crunch as it exists today; a project management anomaly in the games industry. Coupled with this will be data concerning the stated causes for crunch, to which will be applied an inter-coder agreement tested aetiological study to determine the validity of those causes in the face of the new understanding of crunch, and if necessary proposing a new root cause.

This allows for recommendations to be made about the application of project management devices that can be targeted specifically at the newly discovered cause, and to gain insight into the state of project management in the games industry.

This understanding of crunch is extremely important. Crunch has been present in game development for decades, such that it has become a way of life. This can lead to those studying it from within the industry to over-simplify it when defining what they are studying:

“For the purposes of [searching for the root cause], I'm defining "crunch" as "compulsory unpaid overtime"” (Prout, 2010)

Granted Prout's piece is not a scientific paper and his suggestion for root cause is purely based on his opinion, but without understanding completely something like crunch and stating this definition you cannot expect to conclude with a valid root cause, as the definition is too open to interpretation to begin with. This results in different people
reaching different conclusions each time, reducing the likelihood of any of them having a genuine effect on the phenomenon. Outside the industry, the lack of understanding of crunch leads to similar issues:

“Crunch time is a basketball term that describes the last minutes of the game during which both teams fight for victory.” (Petrillo et al, 2009)

While the term 'crunch' may indeed be used in basketball, Petrillo's definition shows how easy it is for study to go astray, and enhances the imperative for a concise phenomenological study. This paper is the first academic study to approach crunch in this manner, defining the anomaly clearly from several perspectives, and is also the first to study the etymological origin of the term.

This detailed and scientific analysis will allow the accurate prescription of relevant solutions to the problem crunch represents to game development, and this paper represents the first time aetiology has been used in a project management context to analyse software development.
Completion of Objective 1\textsuperscript{2} decrees that the current state of knowledge on crunch be determined through examination of the academic literature surrounding it and of contemporary sources as required by study of the games industry.

\subsection{Etymology}

\begin{quote}
"Strange, but I don't remember ever hearing the term "crunch" before joining the software industry."
\end{quote}

\begin{flushright}
– Bill Armintrout, Origin Systems\textsuperscript{3}
\end{flushright}

One of the reasons crunch is largely undefined is that the term is used to describe several different things. By working to gain a deeper understanding of the term and its application we highlight not only the significance of crunch to project management, but also bring to light one of the main reasons it is causing a problem.

Its relevant origin in the context of this work is around 1938 (Harper, 2012), where Winston Churchill used the term to popularise the sense of 'critical moment', which became the genesis of terms such as 'comes to the crunch' and 'crunch time'. In their paper titled 'Optimum Time Compression in Project Scheduling', Lamberson and Hocking (1970) develop an algorithm to reduce the time / cost trade-off in scheduling large projects, the aim being to determine the duration of a project based on completion time for its activities. This then led to the term 'schedule compression', the earliest discovered academic reference to which is the work of Miller and Paré (1986) where they state:

\begin{quote}
"Avoid schedule compression at all costs. Even a small schedule compression usually results in higher cost as a result of poorer quality and increased maintenance".
\end{quote}

Schedule compression is essentially the reduction of the time available to complete a task while maintaining the same level of resources, the risks of which were acknowledged by Abdel-Hamid (1990). The action of compressing a schedule then led to the common description of the type of project to which it was proscribed. In the earliest literary example, Boddie (1987) describes ‘crunch mode’ as:

\begin{flushright}
\textsuperscript{2} Critically analyse the academic literature surrounding the target industry and subject to determine the current state of knowledge on crunch
\textsuperscript{3} (Armintrout, 1993)
\end{flushright}
2. LITERATURE REVIEW

“a schedule that puts you under pressure from the first day of the project.”

At this point in time the computer games industry was gaining its stride, with prominent companies such as Electronic Arts being founded in 1982 (Datamonitor, 2012). Trip Hawkins, EA’s founder, began working for Apple in 1978 before founding EA (Cifaldi & Fleming, 2007), and this move from coder to manager was a common factor in the foundation of many similar companies of the era, Activision (formerly Infogrammes, now Activision - Blizzard), Atari and so on.

While it is almost impossible to pinpoint the migration of terms like ‘crunch’ or ‘schedule compression’ into the games industry, it is fair to assume that people such as Hawkins and his peers would have been familiar with them and they simply passed into common use.

The term Crunch is now commonly used as a noun to describe the period of "firefighting"⁴ around a milestone or deadline (either before or after) where prolonged periods of overtime are necessary to complete the project on schedule – ‘crunch time’. It can also be used as an adjective to describe the act of doing that extra work – ‘crunching’ or to be in ‘crunch mode’.

This is normally found in two varieties:

1. Reactive – the overtime and extra work becomes necessary due to milestones throughout the project being missed or reached late. It is an attempt to fit in more work than would normally be expected in the given amount of time. Usually found at the end of a project, though sometimes encountered earlier in an attempt to get back on schedule. Essentially closer to the definition of schedule compression.

“When the project reached deadlines, most of us spent an extra 2-4 hours a day, but this didn’t last longer than a couple weeks. I have had to do only one all-nighter, and a couple work days ended at 3 am. I consider this getting off easy compared to my boss and the stories he tells of other companies...” (IGDA, 2004)

2. Enforced – The project from the outset entails more work than can realistically be done in the time allotted. Management knows overtime is inevitable, and will impose overtime for extended periods up to and including the entire life of the project. Closer to Boddie’s definition of crunch.

---

⁴ Firefighting is a term used to describe the action of solving crises as they arise in a project; to be constantly putting out fires is a sign of poor planning.
2. LITERATURE REVIEW

“51.7% of respondents say that their management sees crunch as a normal part of doing business in the games industry” (IGDA, 2004)

2.2 Industrial Research

“there has been surprisingly little research performed in the field of business and management aspects of the video game industry” – Mikolaj Dymek, RIT, Sweden\(^5\)

Schedule compression has been an issue in software development for some time (Yourdon, 2004), and attempts at increasing software project scheduling success have varied, and there is a difference of opinion on how this can be achieved.

Callahan and Moretton (1999) attempted to use project management practices to reduce development time, but of the seven hypotheses tested they could really only advise greater stakeholder involvement and more frequent builds. The challenge is that software projects are unique in a variety of ways (Stepanek, 2005) and traditional methodologies can struggle to overcome this. Among these issues, those which relate most to crunch are those dealing with scope and schedule, since these increase development time and can lead to schedule compression.

- Requirements are incomplete
  - Much of game development is a creative process, which can lead to the late addition of features or the late development of tools to facilitate expedient production

- Change is considered easy
  - Comparatively speaking, it is much easier to return to a section of code and make changes than to make changes to the structure of a building

- Change is inevitable
  - Since much of the time it is only possible to know how a piece of code will react to others once it is written and tested, pipelines need to be present to allow bug fixes once testing reveals them

\(^5\) (Dymek, 2007)
2. LITERATURE REVIEW

- Software development is research
  - Tools for creation of a game or its assets are often built by the developers on-the-fly, and the creative process often results in totally new features never before developed and therefore having no precedence.

These factors can easily cause delays or result in the creation of extra work, increasing the need to crunch and making it hard for traditional project management to maintain efficiency. Carmel (1995a) was seeing this a decade earlier when he noticed that PC software developers tended to eschew management practices aimed at reducing development time in favour of ‘an emphasis on “crunch” periods of intense effort’. However, in a separate paper the same year (Carmel, 1995b) he stated:

‘working harder and for longer hours, or ‘speed up’ is generally agreed to be the worst way to reduce time-to-completion’.

Crunch is recognised as being an issue by Petrillo et al (2008) in their survey of problems with game development, though not to the same extent as the broader issues of scope and feature creep. While the list of problems is long and the figures interesting, Petrillo et al are too quick to state that since many of these issues are shared by the broader software development field, solutions successful there will work in the games industry. Although this would seem logical, and the work of Musil et al (2010) tentatively agrees, it has to be noted that games development is software development and so similarities are sure to exist; but in the same way as there are similarities between software projects and construction projects, as software projects are projects.

It would be folly to say that those solutions won’t work until they are tested scientifically, but as Musil notes:

‘...games are different in some aspects and their way of development demands domain specific tailoring. It is therefore not enough to just blindly apply the same proven techniques and best-practices from software engineering, but to [check] if adaptations or complete redesign are necessary in order to serve game developers well.’

One of those critical idiosyncrasies to game development, and Petrillo and Musil agree, is down to the creativity inherent in game development. It accounts for the broad audience which in turn accounts for the crucial seasonal release deadlines. It accounts for the multi-
disciplinary teams which in turn accounts for a number of the communication and managerial issues. It also accounts for the enthusiasm of developers which in turn accounts for the unimpeded feature creep, increasing scope, increasing work and development time, increasing the likelihood of crunch.

It also tangentially accounts for the apparent dearth of strong academic or professional research into game development, and crunch in particular. Indeed, Petrillo's own definition of crunch as 'a basketball term' highlights how little attention the subject has thus far received.

Dymek (2007) illustrates this well in his literature review, showing how research focusses on the softer side of the industry and the games themselves; the merits of play, the literary elements of games, their use as teaching aids, in-game economics and other experiential aspects. His quoting of Aarseth (1997) is very apt:

'Theories of literature have a powerful ability to co-opt new fields and fill theoretical vacuums, and in such a process of colonization, where the “virgin territory” lacks theoretical defense, important perspectives and insights might be lost or at least overlooked. When we invade foreign ground, the least we can do is to try to learn the native language and study the local customs.’

What Aarseth's hyperbole is saying is that if a field of study is not well understood, it is easy for those attempting to study it to be clouded by their own fields of expertise, thus drawing the focus of their study and their conclusions in that same direction. This would seem to be the case with crunch; in order for it to be studied correctly it must first be correctly understood. This paper has already shown that it is a project management issue, and so project management is the most suitable field from which to study it.

2.3 Game Production

“It becomes clear that a lot of the same mistakes are being made over and over again.”

Brandon Sheffield, GDM6

There have been attempts at studying how the creative aspects of game development

6 (Sheffield, 2009a)
affect production and how it can be effectively managed. Zackariasson (2006) discovered that creative functions of a game studio can in fact be managed effectively with very traditional management practices, but it is easy to understand some of the difficulties a project manager may face when encountering these eccentric aspects of production.

Taking, for example, requirement definition; a mainstay of both traditional and software project management. A complex abstract requirement such as ‘fun’ can be extremely difficult to express, even as a non-functional requirement (Draper, 2000). Callele et al (2005) concur that requirement definition in game development can be very difficult, but ultimately recommend:

“the video game industry could learn a great deal from current research and practice in requirements engineering and project management”

Confirmation of this is echoed in a key point raised by Gaume (2006) in that production runs most smoothly when led by a manager who understands that “their task is not to create, but to encourage the emergence of this collective authorship”. Difficulties such as these are recognised, and over the years there have been studies of the problems encountered in games development.

The more comprehensive ones span 7 years, and all show project management or production practices are a serious problem (IGDA, 2004; Tschang, 2005; Petrillo, 2008; Sheffield, 2009a; Potanin, 2010; Musil, 2010; Shirinian, 2011). All bar Tschang also name
2. LITERATURE REVIEW

Crunch as a serious problem – though while Tschang does not mention crunch by name, he records scheduling as the 3rd most prevalent of 39 problems, behind team management and tech issues.

That crunch and project management occur together so frequently in these studies is no coincidence, as it has already been shown that crunch is a project management issue, but as requested by Paul Ross (2011) we do need to look beyond 'better scheduling'.

De Peuter and Dyer-Witheford (2005) discuss crunch when considering the concept of immaterial labour in the games industry, in light of the IGDA Quality of Life survey (IGDA, 2004). They note (as did IGDA) that there are a number of circumstances leading to crunch which involve instilling in the employee a desire to work longer hours. This cajoling often comes directly from production/project management and appears in the form of genuine encouragement or sometimes implied threat. This, when coupled with the knowledge that in areas of the US, Canada and Australia salaried software developers are not entitled to pay for overtime worked, presents a view of crunch beyond simple poor project management. As de Peuter and Dyer-Witheford note in another paper (2006):

“Normalized crunch time therefore points to a very elementary economic fact: it is a good deal, a steal in fact, for game companies.”

There is sometimes encountered a sense that it cannot be possible for management to deliberately disregard the well-being of their employees in this manner, but there is actually research showing what could be its origin. There are many reasons stated for why people are actually willing to work the hours crunch demands (dP and D-W 2005, 2006; IGDA, 2004), and much of it relates to the notion of 'work as play' coupled with youthful exuberance.

Tapia (2004) shows that during the 'dot-com bubble' (roughly 1996-2001) it became the practice of managers in software houses to make use of social myths, discreet suggestion and exploitation of 'time famine' to make people willing to work up to 24 hours / day. These methods included encouraging one-up-man-ship where workers would compete to work the longest hours, encouraging sleep at the workplace by providing amenities, and encouraging play at work. They also employed the use of guilt to make workers feel bad for not staying and working longer periods, the promise of rest or bonuses soon to follow but which may not materialise, and the creation of a culture of 'constant crisis' as a
motivator.

All of these methods are echoed in IGDA (2004) and de Peuter and Dyer-Witheford (2005, 2006) as tools employed within the games industry to get people to crunch.

If we look impartially at the use of crunch to avoid paying compensation for overtime, while legally the precedent exits in the given employment legislation for those areas, the sustainability of such practice must be considered. DeMarco (2002), Spaulding (2009), Boyd and Green (2007) and Legault and Ouellet (2011) address the consequences of putting game workers into the cycle of long hours and little rest that crunch demands, with the latter noting:

“The long work hours pose ethical issues that deserve attention, they affect workers’ health, quality of family life, productivity (due to errors caused by fatigue), income and equity (when they are not paid while necessary and somehow required), under-employment of unemployed persons, employment equity for workers not available for flexible hours if (among other parents and women), transfer the cost of health risks to workers”

All of the above sources also make mention of crunch having become a way of life in the games industry, and this can leave much of the remaining literature focussing on ways to cope with crunch. A notable example is Hight and Novak (2008) in their book Game Project Management, with nothing to offer on the subject beyond how to write off expenses incurred from providing food and drink during crunches.

2.4 Contemporary Sources

“Crunch destroys our health, marriages, personal relationships, and relationships with our kids. The old incentives which were used to retain talent -- promises of future riches and power -- have proven empty for all but a lucky few.”

– Dave Prout, Gamasutra

The above sections represent the current academic thinking on the games industry and the issue of crunch. Unfortunately, the games industry has no academic journal of its own which explains why research to date is so broad and shallow.

7 Translated from French
8 (Prout, 2010)
2. LITERATURE REVIEW

The industry does however have a very active presence on the internet, with sites such as Gamasutra.com (and its sister publication Game Developer Magazine), and Develop Magazine\(^9\) regularly creating articles, features and discussions that serve to contribute to industry knowledge, almost functioning as the industry's knowledge management system. It is these organisations that create the postmortems\(^10\) used by so many of the studies referenced above.

Articles focussing on crunch generally attempt to suggest reasons crunch is most likely to occur and how it can be avoided.

Prout (2010) even makes a move to suggest a root cause, though his assertion that 'getting fun right' is the most important key to success leads him to 'premature production' as the root cause of crunch. It has already been shown by Draper (2000) that the concept of fun is a difficult and problematic point from which to define scope, and Prout's connection between the desire for fun and a rush into premature production is vague, especially as we have already seen that crunch in some form exists outside the games industry where fun is not a requirement at all.

This is further demonstrated by Hyman (2008a) where he discusses serious game\(^11\) development and how it too faces crunch. The reasons stated are associated with lack of communication from a client, leaving development suspended until they respond with requirements, but having the client still insistent on the original deadline. Many of the suggestions presented for avoiding crunch are management related, and where crunch is cited as being less of an issue, strong project management skills are often given as the reason (Remo, 2009; Hyman, 2008b; Martin, 2011a).

There is definite difference of opinion as to whether crunch is good or bad (though 'bad' is definitely the reigning sentiment), and which side you are on seems to be dependent on your approach to management. A great example is shown by Sheffield (2009b) when discussing Epic's Rod Fergusson and 'necessary crunch'. Fergusson's method is to set a release date early, make it inviolable and define scope accordingly. He claims this is a sign that your ambition is greater than your schedule, and so viewing crunch as necessary

---

9 There are many more such organisations within the industry, these are only an example
10 Postmortems are documents created which reflect first-hand on production of a game, addressing what went right and what went wrong.
11 Serious games comprise software environments similar to commercial games, but used for education or as a virtual space for businesses and government organisations (for example).
2. LITERATURE REVIEW

makes it easy to plan for. Such comments were widely scorned however (Cifaldi, 2009), Epic's view being:

“The nine to five work week, I don’t think that fits for our industry,” he added. “And it’s not just because we mismanage or mis-schedule.”

It seems that companies with a view of crunch as a method of production are met in many cases with the scorn of their peers. The cases of EA_Spouse (Hoffman, 2004), Rockstar Spouse (Gamasutra, 2010) and Team Bondi / Rockstar (Martin, 2011; McMillen, 2011) drew very public attention to companies who were forcing protracted periods of intense crunch on their employees, since they felt it was an acceptable method of developing a game.

Their reasons included cutting development costs, immovable release dates, a developer's love for making games, among others. This goes a long way to enforcing the image of crunch as a way of life in the industry. This is viewed from another angle by Fristrom (2004) as he notes that Activision used to provide a laundry service for their developers, as they were working such long hours they could not find the time to do it themselves. It seems a nice gesture on the surface, but it is essentially a way to keep workers at their desks. This was also seen by Dyer-Witheford (2005/6), noting how management would often be keen to provide tents or sleeping bags to workers who were working late.

“Consenting to such hours, he says, 'partly has to do with the fact that they promote, you know, 'Hey, we have a couch here. You can sleep here all night.' . . . You're nineteen'''

Cases such as EA_Spouse were what prompted IGDA's Quality of Life survey (2004) but, notes Hyman (2008c), several years later crunch was still an issue, viewed as 'the elephant in the room'. It is these public cases that bring to the fore the power the larger publishing companies can exert over the smaller development firms, chiefly regarding the setting of unrealistic deadlines to meet marketing demands and the insistence on unethical management techniques such as enforced crunch to meet them (Johns, 2005; IGDA, 2004; D-W, 2005/6). This dictation of annual seasonal release has been seen to affect product diversity and has led to a shrinking number of firms in the industry (Williams, 2002).

The effect of crunch goes beyond the initial experience, as some studios are formed on
the back of developers who had endured all the crunch they could and vowed never to crunch again (Remo, 2009; Hyman, 2008b). Studios such as these often make mention of the utilisation of some form of Agile methodology which has been shown to have significant effect on efficiency both within and outside the games industry where it has been shown to directly reduce overtime (Mann and Maurer, 2005), and this knowledge that perhaps more traditional and rigid methods like Waterfall are less applicable to creative development is becoming steadily more recognised (Smith et al, 2011). It is worth noting that more 'traditional' methodologies such as Critical Chain have also met with success in software development (Meijer, 2003).

2.5 Overview

The above etymological examination shows that crunch has its roots firmly in project management history, evolving into common use through prevalence in software development.

It is a phenomenon which can arise due to project schedule compression at any point in the life of the project, either as a reactionary measure or as a tool in an attempt to increase the amount of work done without necessarily increasing resources. The term can be adapted to describe the work to be done, the act of doing that work, and the time that is spent doing so. Crunch can here be seen to impact the triple constraint. Time is a key factor as crunch is defined in some sense by a factor of time. Resources are affected because schedule compression is often employed to reduce the cost of a project, though the likelihood of over-work causing reduced scope through poorer quality may actually increase costs in the long-term.

Here we can see that while the legacy version of crunch has been tackled in the broader software industry, the modern day manifestation is still not well understood, and that idiosyncrasies in both software and the games industry have proved a barrier to traditional project management techniques. It may even be the case that developers consciously make the decision to crunch rather than employ project management practice.

Both crunch and project management are recognised as being problematic in the games industry on a recurring basis. Managing a multi-disciplinary team where creativity is highly valued has been shown to be difficult, but possible using traditional methods when the manager does not involve themselves too deeply in the creative process, and instead
tries to nurture those whose job it is to be creative. The recurrent nature of these problems suggests there is a lack of education about crunch within the industry, which feeds and is fed by the notion that crunch is a way of life.

While crunch can sometimes arise naturally through a slipped schedule, there is also evidence it can be enforced for reasons including reduction of development costs, and that methods used to do so may be ethically unsound. Evidence shows crunch is detrimental to an employee and also the game in development.

We see that within the industry, crunch is a regular topic of discussion in a variety of contexts. Its recognition as an issue goes back over two decades, though it seems to require the attention of the media on large isolated cases to encourage critical thought on the subject, before it fades away again. It is widely regarded as having a negative impact on those who experience it, though its ability to nevertheless move a project towards completion leads to mixed thinking on its effect as a tool, which in turn sees some management viewing it not only as useful, but preferable, and in some cases necessary.

There is a definite sense that management employing crunch deliberately and for extended periods are doing so with an almost total lack of regard for the well-being of those being made to crunch. Development methodologies such as Agile and by extension Scrum have been shown to have positive effect on project efficiency and direct reduction of overtime.

This review shows that study of the games industry has indeed been broad and shallow and there is little peer-reviewed work on which to build a foundation. Since creativity plays such a strong role in game development, there is a tendency for researchers to focus on that, and they do so from a wide variety of fields, but they fear to tread on the more technical ground of development. Those that do focus on development are very quick to dismiss crunch as an annoyance that is a natural part of production, and the majority of works seem content to suggest ways to tolerate it, very much a case of bandaging a wound but not curing it of infection.

This is the first study that actually focusses on crunch from an academic and scientific perspective, and sees it as observable and measurable anomaly, born of project management. With that in mind, here follows a summary of what the literature tells us about crunch:
2. LITERATURE REVIEW

• It is not new; it has existed in some form within project management for over 40 years. It has existed within the games industry for over 20 years.

• It exists in numerous dimensions, and impacts all aspects of the triple constraint.

• It can arise as a reaction to an excess of work, or can be imposed through deliberate schedule compression.

• Idiosyncrasies within game development mean that traditional project management methodologies have been misapplied in overcoming crunch.

• The phenomenon is not well understood outside the games industry.

• Project management and crunch are both seen as problematic in the industry; the cause and effect relationship between the two not always appreciated.

• Not enough importance is attached to knowledge retention, leading to the same issues being reported consistently.

• Crunch has now become 'a way of life' within the industry; a tool to be used and something to be tolerated.

• Crunch is detrimental to those involved directly, often harming family life, health and psychological well-being, and by extension the quality of the game is often reduced.

• Crunch is often unpaid.
The subject of this research is crunch as it exists in the games industry. Since the literature review demonstrates that it not wholly understood, certainly not to the extent that facilitates study, it is necessary to determine how it manifests in projects, how it is experienced by those who encounter it, its impact on the project and what the circumstances are within projects that result in crunch.

The purpose is to bring crunch to the attention of the project management industry in order to bring to bear substantial resources that may allow the games industry to overcome crunch. The process by which this will be achieved involves developing a phenomenologically sound definition of crunch; one that represents the explicit textural and structural characteristics that make up the anomaly and to serve as a foundation on which to build future research. The process also involves using aetiology, the study of causation, to determine the most scientifically feasible cause of crunch in order to more effectively remedy it.

3.1 Data Required by Aims

Before causation can be studied, it is necessary to define crunch to such an extent that the link between cause and effect can be tested.

The kind of data phenomenology requires includes hours worked during crunch and the duration of crunch within the project, and it is important to uncover descriptions of activities performed during crunch. It is also necessary to uncover the attitudes expressed towards crunch by those involved to develop an understanding of potential motives for employing crunch, or allowing it to develop. Similarly, description of the consequences of crunch and its immediate effect on the project and its team contribute to knowledge of its impact on the industry and the field of project management.

Once we can clearly see how crunch manifests, we can begin to uncover its causes. The data required here is the attribution of a causal incident to an instance of crunch, both in specific cases by a participant and in a general sense from an experiential perspective. This shows what people experiencing crunch believe to be the cause of it. Further analysis will then be conducted to assess the validity of those assertions based on the
3. METHODOLOGY

A phenomenological definition of crunch, where necessary suggesting – by the same analytical method - a more probable root cause.

3.2 Method by Previous Research

**Table 1 – Methodologies of Previous Studies**

<table>
<thead>
<tr>
<th>Method</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anecdotes</td>
<td>Draper (2000); Potanin (2010)</td>
</tr>
<tr>
<td>Case Study</td>
<td>Tapia (2004); Mann &amp; Maurer (2005); Zackariasson (2006); Smith (2011); Callele (2005)</td>
</tr>
<tr>
<td>Interviews</td>
<td>De Peuter &amp; Dyer-Witheford (2005/6); Williams (2002); Johns (2005); Dymek (2007)</td>
</tr>
<tr>
<td>Literature Review</td>
<td>Dymek (2007); Potanin (2010); Williams (2002); Draper (2000); Johns (2005)</td>
</tr>
<tr>
<td>Postmortems</td>
<td>Callele (2005); Petrillo (2010); Shirinian (2011); Sheffield (2009)</td>
</tr>
<tr>
<td>Survey</td>
<td>Legault &amp; Ouellet (2011); Musil (2010); Carmel (1995); Callahan (1999); Mann &amp; Maurer (2005)</td>
</tr>
</tbody>
</table>

**Anecdotal Evidence**

Certainly considered one of the least formal methods of research, the use of anecdotal evidence is not to be overlooked. Hornikx (2005) showed that anecdotes provide the least persuasive evidence to researchers, and this is understandable, as it is largely free from control, often having little specific direction, and contains large amounts of bias and opinion.

However, can be very useful if you are looking to record someone's opinion.

Part of the phenomenological study of crunch involves determining the opinion of those who experience it and the literature review above shows that difference of opinion does exist. It can also be useful in determining the direction of bias, as a method of categorising those opinions, and as such is useful to this study.

---

12 Excluded from this list are those references used for etymological study of crunch, non-academic works such as text books, and contemporary sources such as articles.
3. METHODOLOGY

Case Studies

Case studies offer an excellent scientific opportunity to observe and test in a natural environment, and can provide excellent data for study. However, they often need to be conducted over a considerable period of time, and thus can be limited in scope to a small number of experiments. Where this research is concerned, a case study would provide a great chance to observe crunch occurring organically, allow for the prediction of circumstances to follow, and provide accurate time-scales for issues occurring.

The problem is gaining entry to a project at the right time in development to study cause and effect, and there is no guarantee that such an endeavour would even see crunch occur on that project. Furthermore, many game development projects incorporate some intellectual property and can rely heavily on carefully timed exposure to the media, so the likelihood of a studio allowing an external observer entry is reduced. Therefore, while it does present a desirable opportunity, it is impractical for this work.

Interviews

Interviews provide a means to obtain a specific person's view on a given topic, and can provide good data, if you are speaking to the right person about the right topic.

Crunch presents an interesting problem here, as it is a controversial subject. If the study wanted to know how the interviewee would talk about something they did not want to talk about then this may prove useful, but there is no guarantee they would even consider the subject and so the scope of the data then becomes too narrow. The incident with Epic espousing the virtues of crunch in an interview (Sheffield, 2009b; Cifaldi, 2009) met with bad press, demonstrating what is at stake for those who discuss crunch openly.

While phenomenology favours the use of extensive interviews, the difficulty in obtaining audience with those able to provide useful data in this event must also be considered. As a result, formal interviews are considered too impractical for use in this study.

Literature Review

Some studies focussed mainly on a literature review to obtain data, mainly to get a view of the field of study at that point in time. This paper has already done so, but the usefulness of the literature is not yet over. A large range of second-hand data exists in many of the papers discussed in section 2 which has relevance to this work. This data is
3. METHODOLOGY

mainly the expression of durations of crunch and hours worked in surveys conducted by
other researchers, and will be used for an increased sample size for analysis.

Postmortems

The game postmortems are a superb source of data.

They do have their limitations; similar bias as faced by interviews, a single point of
opinion, and no control over the information presented. Even with that in mind, they
represent accounts from people who actually worked on those projects and are in many
cases the accumulated knowledge of project managers, discussing what went right and
what went wrong, and many of them discuss crunch. The postmortems also date back
decades, which allows us to see how crunch has or hasn't changed in that time. Most
importantly, they also discuss why they crunched, which gives valuable data for discerning
a root cause of the phenomenon.

Surveys

Surveys are one of the classic sources of research data, but they are extremely limited
and impractical for this study. They suffer very low return-rates, as was seen by those
studies utilising them, which limits their scope. There is also a vital need to get the
questions themselves exactly right beforehand to get truly accurate data, and again, as
crunch is controversial, there is little guarantee that any such questions would even be
answered. Although they offer the respondent anonymity if desired, the limits presented by
the physical nature of the survey format make it difficult to get data useful for this study.

3.3 Data Set

Objective 2\textsuperscript{13} demands that appropriate data be examined.

3.3.1 Postmortems

Postmortems are the best source of first-hand accounts of game development. They
exist as easily accessible sources of information, and their purpose is to improve the
collective knowledge of the games industry. They are normally published in Game
Developer magazine, and sister-site Gamasutra often publishes them online at a later

\textsuperscript{13} Critically analyse the academic literature surrounding the target industry and subject to determine the
current state of knowledge on crunch
date. This repository of postmortems dates back 25 years and so offers a chance to see how crunch has manifested over this period. Their structure is based on an introduction, then discussion of 5 aspects of development that went right, and 5 that went wrong. This forces the author to be critical of the project in a manner that would be less common in other circumstances.

To select documents for analysis, a lexical search is performed and only documents containing the word 'crunch' are selected. This is important, as the very fact that the phenomenon has a name, a colloquialism attached to it, is a key part of observing how it is perceived in the industry. The data set comprises 34 documents, ranging in length from 1084 words / 47 paragraphs to 6799 words / 276 paragraphs, and covers projects completed between 1998 and 2011.

NOTE: For a full list of sources and various statistics, please see Appendix 1.

Assumptions

It is assumed that the information given in postmortems are generally reliable accounts of what occurred during production of that game. The passionate nature of game development knits with the purpose of postmortems as a knowledge management device and so the deliverance of accurate information furthers that purpose and is more useful to its audience.

Limitations

No control exists over what information is present. What the author chooses to discuss and how they do so is uncontrollable, and any omissions or bias are their prerogative alone. It may be the case that useful data could also be found in documents not specifically mentioning the word 'crunch' and will not be considered for this study. This data only covers projects which have conducted and published a postmortem and so is only representative of them.

Expectations

An interesting range of bias and opinion of crunch is expected, as is a range of data regarding duration of crunch and hours worked. Discussion of project management surrounding crunch is also expected, given its prevalence in other studies, and project
management's perceived difficulty in the industry. A significant amount of data relating to causes of crunch is expected to satisfy Aim 2.

3.3.2 Anecdotal Evidence

Anecdotes represent an open discussion of a subject. The environment in which an anecdote is delivered can affect how open that discussion is, as can the method of solicitation (Dholakia and Zhang, 2004).

This study is seeking anecdotes of the experiences of those who have encountered crunch in some way, analysing not only the specifics of their accounts, but also the manner in which they deliver them, and what information they volunteer to crunch as a stimulus. Since the lack of games industry journals has led to such a strong presence for the industry on the internet, a selection of websites are approached and a request for anecdotes posted.

LinkedIn was chosen due to the large number of games industry professionals taking part in group discussion on the site. The forums of EDGE, TheDailyWTF, VG247, PCGamer, RockPaperShotgun, Gamespot, PennyArcade and IGDA are chosen for the same reason. The website 'Tales from the Trenches' already allows for the anonymous posting of anecdotes from the industry and so these are analysed as well. It is acknowledged that such sources are unconventional but precedence for such sources exists (Dubrovsky et al, 1991), and given the data required by this study, they are valuable (Dholakia and Zhang, 2004).

The thread is opened with a request for 'tales of crunch'. It is clearly declared that the postings will be used for research, and that all contributions will be treated as anonymous. The ethics surrounding information in forums is ultimately open to interpretation, but as noted by Hookway (2008) internet forums are likened to public forums and as such discussions therein are part of the public domain and require no informed consent for their use. Therefore their willing participation in the discussion is justly considered consent. Also, given that the discussion contains no commercially available information its use in research is deemed 'fair use' (Hookway, 2008). Anonymity is guaranteed as forum users do not use their real names or any identifiable information when posting; the exception being LinkedIn, but the content of the post was immediately copied to blank text documents without names attached.
3. METHODOLOGY

The data set comprises 22 documents, ranging in length from 62 words / 1 paragraph (single forum posts) to 904 words / 27 paragraphs (detailed accounts of crunch).

NOTE: For a full list of sources and various statistics, please see Appendix 1.

Assumptions

The assumption is made that the informality of the environment coupled with the guaranteed anonymity will encourage a more natural response, and the open-ended nature of the request will increase the potential scope of the answers.

Limitations

There is no guarantee that these postings will generate any response at all, and if a response is received there is nothing to say that the information will be constructive or true, as users of forums such as those on the gaming websites can have a propensity towards being facetious, and may be deliberately misleading.

Expectations

There is the expectation that data gathered from LinkedIn will be more reliable and constructive than the other sources. Data from the gaming websites may be more emotive, and offer a more negative opinion of crunch and the events surrounding it, increasing bias. Little constructive data regarding project management is expected, replaced by more data contributing to the definition of crunch as required by Aim 1.

3.4 Coding Schedule for Content Analysis

Description of Crunch

This data is used for satisfying Aim 1, and will provide observable, symptomatic data directly required for a root cause analysis using aetiology to satisfy Aim 2, and consists mainly of:

• Duration of crunch
• Hours Worked

This data allows for a causal analysis through examining whether a stated cause could produce the above effects. The source documents are searched for discussion of crunch
3. METHODOLOGY

and cross-referenced with any mention of time-units such as 'hour', or 'month' as well as numerical values. Any description of activities performed during these periods was also coded.

Example from the data:

“With crunch hitting, they added to our schedule AGAIN to 9 AM to midnight Monday through Saturday, with an “anything goes” warning. I ended up working 19 hour days, 7 AM to 2 AM, six days a week for the last six weeks of the project.” \(^{14}\)

**Consequences**

This data provides information on the effect of crunch on the team and the project.

It contributes to the phenomenological study by showing how participants believe crunch impacts upon them and their work. This is necessary for a causal analysis to show the magnitude of the impact of crunch, which helps to determine if a given cause could feasibly have the power to create such an impact. Having found an instance of crunch in a document, further analysis attempts to uncover the subject's perspective on the impact of that crunch on the project, or on the project team.

Example from the data:

“But the worst part of the process was the entire team’s ambivalence about the project. No one was truly excited about the game or where it was heading. We were making it work through sheer effort.” \(^{15}\)

**Attitude Towards Crunch**

This data shows how crunch is described by those who encounter it. It includes both positive and negative references, as well as adjectives used to describe crunch specifically. When studying causation, this helps identify discrepancies between the stated cause and its effect on the team. Instances of crunch are analysed for adjectives used to describe crunch, and any reference to either a positive, neutral or negative viewpoint on the issue.

Example from the data:

\(^{14}\) Anon 8 - TFFT
\(^{15}\) Ratchet and Clank - Postmortem
3. METHODOLOGY

“After a while you get used to it. It’s even easy to some extent, because the extreme pressure forces you to only work on the most important parts of the production while you can safely ignore everything else.” 16

Causes

This data contributes directly to the satisfaction of Aim 2 and is the primary source of data for its analysis. It comprises statements from those involved in a given project identifying what they believe to be the cause of that project's crunch. It also contains speculative data given by members of the industry on the cause of crunch in a broad sense within the industry. Careful reading around instances of crunch uncovers the subject's view on what directly caused the crunch they were experiencing.

Example from the data:

“All, instead of removing features from the game the producer and designer actually added new features as late as the last week in August. Oh ya, and the deadline got moved up a week because the distributor made a mistake on the retail shipping date.” 17

Project Management

Data relating to various aspects of project management was also gathered, primarily for analysis of its significance relating to discovered causes and effects of crunch. It is also useful to see how discussion with crunch overlaps discussion of project management.

This data was further categorised by:

- Requirements and Planning
- Team Management
- Estimation
- Scope / Feature Creep
- Scheduling

These headings were chosen as in the literature review they were consistently identified as being issues in game development.

16 The Path - Postmortem
17 Anon 22 - Email
3. METHODOLOGY

Example from the data:

“Unfortunately, we didn't have a plan. At least not a plan that had any reasonable way of tracking how we were doing, where we were, or how much we had left. All that existed was a timeline on the whiteboard with numbered weeks associated with levels and features. If a level slipped past the week to which it was assigned, we would just consider it “good enough” -- even though it was missing crucial gameplay features.”

Note – For examples of coded documents, please see Appendix 2 – Sample Coding

3.5 Phenomenological Study

As per Objective 3\textsuperscript{19}, data is explicated using a modified Van Kaam method as described by Moustakas (1994). 151 relevant quotes are identified that represent a moment of the participant's experience of crunch.

These are then subject to reduction through testing against 2 criteria:

- Is the moment necessary and sufficient for understanding crunch?
- Can it be abstracted and labelled?

These constitute the horizons of the experience; moments all of equal value to understanding the phenomenon. Those that fail are discarded as being unrepresentative of the experience. Further reduction follows by eliminating horizons containing vague language or repetition. Remaining are 89 instances across all sources corresponding to 33 distinct Invariable Constituents. These are grouped into 7 thematic clusters, all of which can be seen on the Ishikawa diagram in Appendix 4 – Effects of Crunch.

These are then used to develop a Composite Textural Description of crunch; a descriptive integration of the textural constituents and themes that outlines what the experience of crunch represents to the participants. This is then subjected to Imaginative Variation, a process by which alternate meanings and perspectives are pursued, such as relationship to time or inter-personal perception. From this is developed a Composite Structural Description of crunch; a descriptive representation of what structures of the

\textsuperscript{18} Magicka - Postmortem
\textsuperscript{19} Describe crunch phenomenologically through the use of horizontalisation and the determination of invariant constituents to create composite textural and structural descriptions
3. METHODOLOGY

experience evoke the textural qualities. These together comprise a synthesis of the meanings and essences of crunch.

3.6 Aetiological Study

To complete Objective 4\textsuperscript{20}, the causes identified as per section 3.4.4 are subject to similar process as described in 3.5. 56 causative statements are identified across all documents, varying from two or three words to several detailed sentences. These are then grouped into 9 thematic clusters representing a consensus across several sources of the likely causes for crunches they experienced or that they believe apply to the industry generally. These thematic clusters are used to generate Composite Textural–Structural Descriptions describing the circumstances of all clustered causes.

The causal descriptions are used as a basis for analysis using the Bradford Hill criteria (Bradford Hill, 1965); a set of nine criteria by which one can determine the potential of the link between cause and effect. While initially used in medicine and diagnosis, they can be applied to other fields too. Bradford Hill was keen to emphasise that these are not to be seen as boxes to be ticked, rather that they be used in conjunction with the available knowledge on the subject to make better informed decisions regarding causality.

The criteria in the context of this study are as follows:

\textit{Strength}

How strong is the association between the suspected cause and crunch, in light of the number of cases where the cause is \textit{not} present and given the total number of different suspected causes?

Strength is measured against those source documents where there is sufficient information provided to constitute an actual account of crunch and where a cause can be determined as invariant, of which there are 39. In order to determine strength, we need to determine what would constitute an equal association. The lowest factor of 39 is 3, and there are 9 potential causes under investigation.

Therefore:

\textsuperscript{20} Determine the aetiology of crunch using the Bradford-Hill method of causation
3. METHODOLOGY

\[
\frac{39}{9} = 4.33 = 4.33:39 = 1.44:13
\]

It can be generalised that a cause with a ratio of less than 1.44:13 has a weaker association than one of the others.

**Consistency**

How consistently is the cause associated with crunch each time an association is investigated?

Given that this is the first study of crunch specifically, consistency will be measured against the studies conducted by Petrillo (2009), Sheffield, (2009a) and Shirinian (2011) as they all studied issues or problems in game development, and IGDA (2004) also summarised some of the causes given for crunch in their survey. This will highlight how consistently a given cause is seen as problematic.

**Specificity**

Is there anything which links the suspected cause specifically to crunch?

This looks at whether a cause could be a factor in other problems, or just crunch. It has to be considered in advance that there are 9 suspected causes for crunch in this investigation which already rules out specificity of one particular cause, though it could be possible for a cause to have specificity of magnitude, where it is more likely to result in crunch than any other outcome.

**Temporality**

Can the cause be seen to explicitly precede crunch, and not in fact also be caused by it?

Given that a single studio may run several successive projects, it could be the case that the crunch from a previous project can have a lasting residual effect on the next, which would negate grounds for causation by temporality.

**Gradient of Exposure**

If the magnitude of the cause is altered, is there a direct impact on the effect?
3. METHODOLOGY

This examines how increasing or decreasing the scale of causative circumstances effects the potential scale of crunch. The work done during crunch can vary, as can its duration, so the ability of the suspected cause to alter such variables is grounds for causation by gradient.

**Plausibility**

Does it make sense that the suspected cause leads to crunch?

Where there is a lack of a direct link between a cause and crunch, this examines whether the two could conceivably be related, given what we know about them both. It is important that Hill cautions plausibility is affected by available knowledge at the time. The new knowledge that this paper represents on crunch (*not* a basket-ball term, rooted in project management, more than simply unpaid overtime) means that some things not seen as plausible before may now be considered so, the converse applying as well.

**Coherence**

Does the association conflict with what is already known about the suspected cause?

This forces us to consider causation even in the absence of evidence from the other criteria. For example, simply because a cause is not common or even plausible, if it does not contradict what we do know about crunch then it cannot necessarily be ruled out.

**Experiment**

Can the results be repeated in controlled conditions?

Ideally the above could be tested in a controlled environment where variables could be altered to verify their association with crunch, a criterion owing its existence to aetiology's medical genesis. Essentially this could be satisfied through observation of a game in development; the crucial test in this case is more about whether the suspected cause could in fact be a variable in an experiment.

That a cause can be subject to experiment is important in the context of this study, as it represents opportunities for the project management industry to begin addressing the issue of crunch in a practical project management context.
3. METHODOLOGY

Analogy

Are there similar circumstances in unrelated cases to support the link?

This asks whether something similar has happened elsewhere that supports the link. It is not always the case that this will be so, but that does not give grounds for dismissal. It's main use here is in the absence of a cause or grounds for causation, but since this investigation has 9 causes, analogy is largely unnecessary.

Note – For inter-coder reliability figures, please see Appendix 3 – Inter-Coder Reliability.
4. RESULTS & ANALYSIS 1 - PHENOMENOLOGY

4.1 Description of Crunch

Here can be seen the data gathered relating to the observable symptomatic features of crunch.

Consecutive Hours Worked

*Fig. 2 – Coding of Consecutive Hours Worked*

<table>
<thead>
<tr>
<th>Range Statistic</th>
<th>Mean Statistic</th>
<th>Std. Error Std. Statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>40.00</td>
<td>17.0179</td>
<td>1.79238</td>
</tr>
</tbody>
</table>

This information shows that hours worked during crunch can vary considerably. While the majority of shifts tend to be between 12 and 14 hours, extremes where people have worked for as much as two whole days and nights without sleep push the mean average to 17 hours.
4. RESULTS & ANALYSIS 1 - PHENOMENOLOGY

**Hours Worked in a Week**

Weekly hours during crunch can vary even more, and people do seem to be working throughout the spectrum. The average is 81 hours, with more people working shifts on the higher side of that figure. The extreme in this case is 120 hours; the equivalent of working three 9-5 jobs in one week. *Fig. 3 – Coding of Hours Worked per Week*

![Bar chart showing hours worked per week]

*Table 3 – Hours Worked per Week*

<table>
<thead>
<tr>
<th>Range</th>
<th>Mean</th>
<th>Std. Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistic</td>
<td>81.5673</td>
<td>3.89846</td>
</tr>
<tr>
<td>Std. Error</td>
<td>19.87832</td>
<td></td>
</tr>
</tbody>
</table>

**Duration of Crunch**

The duration of crunch within a project can vary from a few days repeated at various milestones to over a year of constant crunching. A mean duration of 5 months is a considerable time, especially given the above figures.

**4.1.1 Considerations**

When the stated time-frames were a range such as '3 or 4 months' or '11 – 12 hours' a mid-point was taken to be used for analysis, and this will have altered the averages and standard deviations slightly. There were various instances of a time-frame being vague.
4. RESULTS & ANALYSIS 1 - PHENOMENOLOGY

and descriptive such as 'a few months' or simply 'endless', and while interesting they cannot be accurately analysed and so were omitted from these results.

![Fig. 4 – Duration of Crunch (months)](image)

**Table 4 – Duration of Crunch (months)**

<table>
<thead>
<tr>
<th>Range</th>
<th>Mean</th>
<th>Std. Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.5</td>
<td>5.1318</td>
<td>0.73877</td>
</tr>
</tbody>
</table>

4.1.2 Analysis

During crunch people spend a large portion of their waking lives working, and for some the time spent awake seems to be at the limit of human endurance. The work week for game developers is on average twice as long as the standard 9-5. The mean values together would suggest a 5 day week at around 16 hours a day, but the modal figures for daily hours suggest a more marathon 13.5 hours 6 days a week. When combined with the stated durations of crunch, the average being 6 months, that represents a huge amount of labour for one employee to commit to.

Taking for example California (where many game studios are based) software programmers paid above $42 per hour are not entitled to overtime pay (Truesdell, 2012).
4. RESULTS & ANALYSIS 1 - PHENOMENOLOGY

Therefore a 40 hour week at that minimum would earn $1680 for the employee, or considered another way, costs the company $1680 in wages. Of course, if that same programmer works the mean of 81 hours, he is only entitled to the same amount, which actually works out at a net loss of $42. Conversely, the studio has made a saving of $1722. Over a period of months or years, this obviously represents a considerable saving for a company, and goes a long way to suggesting why crunch is still around today.

*Fig. 5 – Crunch as a Proportion of Dev Time (months)*

The durations alone are interesting figures, but when compared with total development time (where able) they become even more so, as seen in *Fig. 5* and *Fig. 6*. These charts give a tentative understanding of how crunch has come to manifest in these projects.

BioShock would appear to have experienced the minimum impact of crunch. 3 months while not the smallest amount is well under the mean average, and in a dev-time of 36
months this accounts for around 8.3%. When the figure is this small, it could be that crunch was reactionary around a milestone or deadline; a burst of increased activity to repair a slipped schedule, or perhaps to fix something which has broken.

Several projects crunched for 60% or more. Figures like these suggest more than simple firefighting, as if that were the case it would mean that half the project involved making mistakes that needed the other half to rectify.

The obvious extreme is My Life as King, which spent its entire 13 month dev-time in crunch. This suggests either an imposed crunch to meet an unrealistic deadline, or simply bad management throughout the project. The mean average is 36% of dev-time spent in crunch, which has obviously been affected by My Life as King, but even without that project included the mean is still 31% which is ultimately a large proportion of your project spent working extremely long hours.

Again though, for a company this represents a huge saving in labour costs. At the figures above, a 12 month project that crunched 36% of the time would save around $31500 per employee.
4.2 Composite Textural Description

The view is held that crunch is complex; an anomaly that merits its own attention and is demonstrably different from the act of working long periods of overtime.

“If your company regularly does “crunches” to the point where it’s an actual thing that you call “crunch time”, then you have a problem and you need to fix it.”

The opinion of crunch is overwhelmingly negative due to its impact on the team and on the project itself. It is widely regarded to be something that needs to be avoided; a sentiment expressed form the perspective of those who intend to avoid it in future projects, those who are attempting to avoid it in their current project, and those who have failed to avoid it in the past.

“The only thing we were certain of was that we wanted to avoid another crunch like we had just experienced.”

This is because it is seen as problematic, both at the project level and in the games industry as a whole. It is viewed as symptomatic of deeper concerns and there is frustration at its continued existence. There is reference to modern methodologies like Agile being able to reduce the need to crunch and enable teams to avoid it in the future.

A recurring theme is resentment, where team members resent the situation that makes them crunch, resent having to do the work crunch requires, and end up resenting their jobs and the industry. They are left feeling unwanted and undervalued, often causing team members to leave the team immediately following crunch, and this contributes to the large skill migration from the industry.

Physical and mental well-being are the main concerns of those who crunch, who describe prolonged periods of intense stress where they are pushed beyond their mental limits, suffering breakdowns where they lose control of their tempers, mental stability and will to continue with the project.

“I had many nights where I would tell my wife that I was done, that I didn't want to make the game any more, that it wasn't worth it, and that I would gladly bow out and take the loss just to go back to my normal life. She would “talk me off the roof,” I'd go to sleep.

21 Anon 5 – TDWTF forum
22 Diablo 2 - Postmortem
4. RESULTS & ANALYSIS 1 - PHENOMENOLOGY

wake up five hours later, and repeat the same day again.”

Fatigue and physical exhaustion are common complaints, which are a direct result of the long hours discussed in section 4.1 and the associated circumstances of sleep deprivation, interrupted body clocks and lack of proper diet. There is the feeling that once crunch has begun it consumes the life of the participant, leading to the exclusion of personal relationships which further contributes to stress and increases job dissatisfaction.

It is often discussed on a scale from bad to worst, where there is no actual talk of crunch itself being good (Fig. 7), only grudging recognition that it can have positive impact on the project.

“..this should NOT mean it's normal or acceptable, nor should we consider it a casualty of the job.”

The positive impact crunch can have on a project is mainly derived from its ability to instil in some team members a greater focus, allowing them to concentrate on specific aspects of production to a greater degree, and this drive can lead to increased productivity. This productivity can in turn have desirable results like being able to include features that would otherwise have been left out, or to some participants the opportunity to finish the game at all. In these situations there can be great determination in the team and spirits can be high as they pull together for a common cause.

“The ominous pressure combined with our love of the project forced us to focus, cut, and refine the game in bursts of activity.”

There is recognition of a counter-productive effect of this focus though, where it can lead to the neglect of other areas of the project that would benefit from attention, which can result in other features being cut due to them being incomplete or left unstable when crunch has finished. It can affect decision making ability which can lead to the introduction of more mistakes, reducing the efficiency of the team and breaking its dynamic and structure resulting in poor communication.

23 Super Meat Boy - Postmortem
24 Ticket to Ride Pocket - Postmortem
25 Octo Dad - Postmortem
4. RESULTS & ANALYSIS 1 - PHENOMENOLOGY

**Fig. 7 – Word Cloud of Adjectives Used to Describe Crunch**

“..worst of all, it's ultimately bad for the project, because there's no time for reflection when making decisions.”

Work done during crunch is sometimes discarded afterwards as the act of being in crunch reduces the quality of the work done to the point where it can't be used. It is sometimes the case where there simply isn't work for everyone in the team to be doing but they are made to crunch anyway. It is also possible for crunch to be completely ineffectual rendering it unnecessary to begin with. There is mention of it having either no positive effect on the project or absolutely no impact at all.

“The irony is that this overtime didn't help anyway -- the project didn't move any faster or go out any sooner”

Participants will sometimes acknowledge that they are responsible for crunching due to their own mistakes during production, but many feel crunch to be imposed upon them, sometimes unethically or illegally. They feel they are offered incentives which will never materialise on the other side of crunching, which breeds mistrust in management and publishers who compress schedules and move forward deadlines.

26 The Path - Postmortem
27 Gabriel Knight 3 - Postmortem
4.RESULTS & ANALYSIS 1 - PHENOMENOLOGY

There is complaint of being made to feel guilty for taking days off or not working the same hours as others in the team, which can further reduce team cohesion and effectiveness. This is unwelcome as participants feel they are going out of their way during crunch engaging in activities they would not normally engage in at that point in time so to have their efforts wasted reduces morale further, and having those people in the office with team members who do need to crunch is a distraction and only serves to reduce their morale as well.

These problems are exacerbated by the impression that crunch is a way of life in the games industry. Participants feel that crunch has become almost a methodology in itself, and has become the go-to method for solving problems in production. It is felt to be omnipresent throughout the industry and has been for some time.

4.3 Composite Structural Description

Crunch is viewed in such a negative light because of the direct impact it has on the person experiencing it.

Crunch is defined in some sense by a measure of time, be it the amount of time crunch causes a person to work, or the broader duration of the crunch period itself and this is felt on the participant's short and long term perception of time. In the short term the long shifts and lack of days off mean that crunch takes up all of their waking time. There is no time to plan for simple respites that involve taking care of themselves, such as shopping for food, since they spend all their time at work and this causes one day to blur into the next so that soon crunch is all there is. This couples with the lack of proper sleep cycles and day / night cycles to leave the participant feeling disoriented which increases physical and mental discomfort.

In the long term the knowledge that crunch is unlikely to end until the project does and that the duration of crunch could be in a factor of months make it appear as though the end will never come.

This compound impact of crunch on the self unsurprisingly leaves participants with little to say about crunch as an entity that isn't negative. Where they do discuss its positive features they focus on its effect on the project, because they love the project. The desire to make games is extremely high in the industry and the satisfaction gained from doing so
can be enough to see people through even the toughest crunches, all for the good of the project. This drive and dedication is highly sought after in project management because when a team is so keen on the project they need little corrective management beyond fixing issues in production.

Team members will seek solace in the companionship of one-another which develops camaraderie and they ride this through crunch to its conclusion for the sake of the project, even at a cost to their own well-being. This group mentality facilitates the assignment of blame to themselves for the crunch being necessary, through inner reflection in a more positive light.

This enhances their negative expression when the crunch proves ineffectual as the dedication they showed to the project has yielded nought, distancing them from the team and the industry through a sense of abandonment. This same effect leads to feelings of betrayal and mistrust of management who are seen to use crunch as a tool with no regard for the impact on the team.

Note – For an Ishikawa diagram of the effect of crunch see Appendix 4 – Effects of Crunch
Here can be seen the analysis of the stated causes of crunch, through application of the Bradford Hill criteria.

### 5.1 Reduction of Team Size

This cause accounts for 3 coded constituents, being 5.56% of the total.

The project team is reduced in size at a point after work has commenced. This can involve team members being moved to different projects the studio is working on, ironically to reduce the risk of crunch for the other team. It can also occur where team members are fired, either to cut costs for the studio, or because the project itself has been reduced in scope and there is no longer any work for them to do. It may also be that team members leave of their own accord to pursue work elsewhere.

“Circumstances arose that caused the two programmers to leave Ritual, and the level designers were assigned to the Sin team. This caused F.A.K.K. 2 production to slow down during Sin’s final crunch, from March to October 1998.”

#### Analysis

- Strength of association

Out of 39 accounts given of crunch, 3 cite having their team reduced, which is a ratio of 1:13. This suggests a low strength of association in the face of the other potential causes and the proportion of crunches where staff reduction was not an issue.

- Consistency of association

Petrillo identifies 'loss of professionals' during production as a problem in the industry, but the other three studies make no mention of it. This suggests that the cause is not consistently observed or identified as an issue.

- Specificity of association

Reduction in the size of the team could contribute to many issues, in any given industry, and there is nothing to suggest it is more likely to cause crunch than any of those, so there

---

28 Heavy Metal: F.A.K.K. 2 - Postmortem
is no ground for specificity.

- Temporality of cause and effect

  Given that crunch can create a high degree of employee dissatisfaction as well as ill health and low morale, it is entirely possible that this residual effect from crunch on a previous project could cause team members to leave the current project once their dissatisfaction reaches critical mass. This means it cannot be seen as grounds for temporality.

- Gradient of exposure

  The more the team is reduced, the more work the remaining team has to do per person in the same amount of time, which is equivalent to reducing the amount of time available to do a given amount of work and as such is definitive of crunch, giving it strong grounds for causation by gradient of exposure.

  It could be, however, that there was not sufficient work to justify the original size of the team, and therefore the reduction would not impact the workload directly. The loss of those members of the team could conceivably cause a blow to morale, but there is no guarantee that this would then lead to crunch.

- Plausibility of association

  Given what we know of crunch from section 4, it is entirely plausible that a reduction in team size could lead to a project entering crunch, though the stronger link through gradient renders this less important.

- Coherence

  The link between having the team reduced in size and crunch does not contradict anything known of either.

- Experiment

  It is certainly possible to remove team members from a game in development to observe the effect it has on the project and if it leads to crunch.

**Summary**

It makes sense that reducing the size of the team during production could lead to
crunch, and the amount by which it is reduced could affect the scale of the crunch, but it appears that it simply is not a common enough occurrence to be a major factor, and the strong case for prior crunches resulting in team members leaving means it cannot be considered a root cause.

5.2 Late Change of Direction

This cause accounts for 3 coded constituents, being 5.56% of the total.

A conscious decision is made late in production to make significant changes to the game. This can be the result of outside influence on the project, feedback from focus groups, or the opportunity to release on additional platforms with little advance warning. It can mean a significant increase in the amount of work that needs to be done, and pressure to complete it can be intense as to miss the new release date can be disastrous. This differs from feature creep in that instead of a gradual build-up of extras there is a distinct, significant change at a discernible point late in production.

“We made a conscious decision to delay SpellCraft's launch (and possible support from Apple) because we felt the game just wasn't ready for prime time. We were divided internally about the decision to delay, but ultimately decided that we only had one chance to launch, and that we were invested in SpellCraft for the long haul, which meant that we couldn't go to market without additional UI polishing and gameplay balancing.”

Analysis

• Strength of association

Out of 39 accounts given of crunch, 3 cite significant changes late in production, which is a ratio of 1:13. This suggests a low strength of association in the face of the other potential causes and the proportion of crunches where late changes were not an issue.

• Consistency of association

Shirinian notes that around half the postmortems he studied reported late changes or additions but it is in the context of feature creep rather than a significant change of direction. None of the other studies mention this issue.

29 Spellcraft: School of Magic - Postmortem
5. RESULTS & ANALYSIS 2 – AETIOLOGY

• Specificity of association

It could be argued that there exists specificity of magnitude, as unless an extension is secured, crunch would be the most likely outcome.

• Temporality of cause and effect

There is no evidence that crunch from a previous project would result in a team making significant changes late in production on the next, and it is highly unlikely (though not impossible) that a team would make such a decision while already crunching, so there is grounds for temporality.

• Gradient of exposure

Both increasing the scale of the change and the time into production it was introduced would affect the likelihood of crunching and the severity of the crunch. Similarly, reducing both factors would lessen the need to crunch altogether, so there is strong grounds for causality by gradient.

• Plausibility of association

It makes sense that a sudden and dramatic change to the project late in development could result in the team crunching to complete the work, unless an extension was granted.

• Coherence

The link between significant changes late in production and crunch does not contradict anything known of either.

• Experiment

It would certainly be possible in observed production to present the team with instruction to make the game ready for release on an additional platform. The platform chosen and the point at which the decision is made could be varied.

Summary

Unless given an extension, it is reasonable to expect that a significant change late in production could result in crunch. The increased amount of work required coupled with the possibility of new deadlines at short notice is essentially the recipe for crunch, the severity of which would depend on the scale of the change and the point in development at which it
5. RESULTS & ANALYSIS 2 – AETIOLOGY

occurred. It may not be as likely to happen as other suspected causes, but when it does, crunch is more likely.

5.3 Over-Burdening

This cause accounts for 4 coded constituents, being 7.41% of the total.

A team or an individual is tasked with completing more work than can be completed, and that they would not normally do. This can be through late realisation of the amount of work involved in an undertaking, especially discovered by younger studios when balancing game development with running a business. It can involve the offloading of excess work onto individuals from a different team, in addition to their own work. It can also occur out of desperation or a sense of responsibility where an individual willingly shoulders a heavy workload in an attempt to make a difference to the project without realising that they may not actually be able to do it.

“About two-thirds of the way through the project, we realized just how much work it takes to run a company as well as make a game -- especially if you’re making your first game. The newness of everything compounded upon itself and overburdened each of us. Just to stay on top of everything, we found that we had to work more hours.”

Analysis

• Strength of association

Out of 39 accounts given of crunch, 3 cite over-burdening, which is a ratio of 1:13. This suggests a low strength of association in the face of the other potential causes and the proportion of crunches where over-burdening was not an issue. It is worth noting though that it occurred twice in Bit Trip Beat, for two separate reasons (running a new company and “do-do-do mode”).

• Consistency of association

Shirinian mentions over-work, but in the context of making someone work too much, which is not the same as someone having too much work to do. IGDA mentions that the large amount of young and enthusiastic, but inexperienced, entrants to the industry are more likely to take on more work than they can handle, though it is not a direct link to

30 Bit Trip Beat - Postmortem
crunch. The other studies do not mention this issue.

• Specificity of association

Although crunch can be seen as representative of an over-burdened project, it would not necessarily be a specific outcome of being over-burdened. There may be an argument that crunching is a more likely outcome than others, but as it stands there is no major grounds for specificity.

• Temporality of cause and effect

Crunch could cause team members to become over-burdened and to over-burden themselves so there is not grounds for temporality.

• Gradient of exposure

Increasing the burden on an individual or team is more likely to make them reach a critical breaking point than to increase the chance of crunch. Crunch is more likely to occur shortly after the realisation of being over-burdened in order to relieve it.

• Plausibility of association

It makes sense that a person or team may crunch to relieve the burden they are under, if done in relatively short order.

• Coherence

It does not make sense that they would crunch after being over-burdened for a significant amount of time, as the increased strain of crunch would likely cause them to snap.

• Experiment

It would be difficult to replicate this in experiment due to the risk of undue harm to the subject. The point at which different people or teams reach the limit of how much work they can do before breaking would vary, increasing the risk of over-working them rather than giving them too much work to do.

Summary

Even though it does make sense that over-burdening would lead to crunch, there isn't
enough grounds for causation by aetiology. Having too much work to do is to an extent symptomatic of being in crunch, so the line between cause and effect is too indistinct.

5.4 Feature Creep

This cause accounts for 4 coded constituents, being 7.41% of the total.

Features are added to the game throughout production, often with a lack of appreciation for the impact this may have on several different processes, and the amount of work involved in making those features work within the game as a whole. Features may be requested following genuine belief that they will make the game better, pressure from a stakeholder such as the publisher to make it more saleable, or simply because the person making the request has authority and thinks adding it will be no big deal.

“...letting just about every feature request get added to the list was a major reason why we had to go into crunch mode in order to finish everything by our July deadline.”

Analysis

- Strength of association

Out of 39 accounts given of crunch, 3 cite feature creep, which is a ratio of 1:13. This suggests a low strength of association in the face of the other potential causes and the proportion of crunches where feature creep was not an issue. It was also mentioned in Anon 4, where it is presented as a cause for crunch within the industry generally, without referencing a specific case of crunch.

- Consistency of association

Feature creep is stated as problematic by Shirinian, Sheffield and Petrillo. It is not mentioned by IGDA in this context.

- Specificity of association

Crunch is not a specific reaction to feature creep, as it is not simply the fact that features are being added would cause crunch, but the type of features and the rate at which they were added. This strong lack of specificity would seriously suggest lack of grounds for causation.

31 Heavy Metal: F.A.K.K 2 - Postmortem
5. RESULTS & ANALYSIS 2 – AETIOLOGY

- Temporality of cause and effect

  Ever increasing scope could precede crunch, and it is unlikely (though not impossible) that feature creep would continue once in crunch.

- Gradient of exposure

  Increasing the number of features added would likely increase the chance of a prolonged crunch to work through them, and increasing the rate at which they are added could increase the chance of several smaller crunches to work through them in short bursts. This gives strong grounds for causation by gradient of exposure.

- Plausibility of association

  It is entirely plausible that constantly adding features to a game throughout production could lead to a period of crunch to incorporate those features, integrate them with other aspects of design and development, and to test for and fix bugs introduced as a result.

- Coherence

  The act of adding features to a game cannot be said to cause crunch, it is the circumstances surrounding the act that may result in crunch.

- Experiment

  Variability is high in both the type of features that could be introduced and the rate at which this happens to observe their impact on the project and its chance of crunching.

Summary

Feature creep is widely cited as an issue in the games industry and in the broader software industry, and it certainly is plausible that it could lead to crunch. However, the action of adding features to a game during production is not in itself adequate grounds for causation where crunch is concerned. More specific is the type of features added, and the rate at which feature creep occurs. More data is required in this area before causation could be asserted; fortunately experimentation would be elementary.
5. RESULTS & ANALYSIS 2 – AETIOLOGY

5.5  The Absence of a (good) Project Lead

This cause accounts for 4 coded constituents, being 7.41% of the total.

There is no one person with clear responsibility for decision making, direction and accountability to stakeholders. Alternatively, the person in that role is not doing their job satisfactorily or effectively. This can result from not hiring a lead through lack of funding or lack of knowledge about what effect the role has on a project, or that the lead has an inadequate understanding of project management and its role in production.

“There were 12 months of development time in between, during which we had little to no management on the project. There were members of the team that took on this role, but only in a limited capacity, as they had tons of other work to do as well.”

Analysis

• Strength of association

Out of 39 accounts given of crunch, 2 cite the producer / project manager, which is a ratio of 0.67:13. This suggests a very low strength of association in the face of the other potential causes and the proportion of crunches where project leadership was not an issue. It was also mentioned in Anon 4 and Anon 5 as being causes of crunch generally within the industry.

It is interesting to note that the two postmortems cited the lack of someone in the role as the problem, whereas the two anonymous accounts saw the problem as someone doing the job, but badly.

• Consistency of association

The lack of a project manager or producer is not explicitly mentioned as problematic or related to crunch in any of the other studies.

• Specificity of association

This cause could account for any number of problems in any project-driven industry, so there is no grounds for causation by specificity.

32 Heavy Metal: F.A.K.K. 2 - Postmortem
5.RESULTS & ANALYSIS 2 – AETIOLOGY

• Temporality of cause and effect

The notion of crunch as a way of life in the industry would suggest that time spent working in projects that crunched could breed project managers or producers who felt that crunch was a viable tool to use in production, which would account for their lack of proficiency, as they could make up for their short-comings by sending the project into crunch. While this is grounds for an interesting debate on which occurred first, it is not grounds for causality by temporality.

• Gradient of exposure

While increasing the capability or ineptitude of the lead might affect the magnitude of crunch, it would be very difficult to quantify and there is no direct evidence to support the link.

• Plausibility of association

Given that crunch has its genesis in project management theory, it is quite plausible that an ineffective lead, or lack of a lead altogether, could lead to crunch by proxy of allowing the project management issues that affect development to go unabated.

• Coherence

The link between a bad / absent project lead and crunch does not contradict what is known of either.

• Experiment

It would be possible to test production with and without a producer to test their relationship to crunch, but testing different levels of ability would be difficult.

Summary

It may be plausible that a bad or absent project lead could cause what has been demonstrated to be a project management problem, but aetiologically it cannot be stated with confidence that it is a cause of crunch.
5.6 Release / Ship Date

This cause accounts for 5 coded constituents, being 9.26% of the total.

The shipping date is selected based on when the publisher wants the game in the hands of the public, not on when the game can feasibly be completed. Public pressure and the enormous spending on advertising, especially around seasonal sales, make the dates immovable. When the dates are brought forward, effectively compressing the production schedule, it can apply serious pressure to the team.

“I worked for an MMO studio that was gearing up for its public launch. We weren't ready, but the funding was ending so the ship date was unmovable. The entire studio went on crunch mode which required working seven days a week, at least 14 hours a day.”

Analysis

• Strength of association

Out of 39 accounts given of crunch, 2 cite the ship date, which is a ratio of 0.67:13. This suggests a very low strength of association in the face of the other potential causes and the proportion of crunches where shipping was not an issue. It was also mentioned in Anon 4, 5 and 6 where they cite it as a problem in the industry generally.

• Consistency of association

Deadlines are highlighted by IGDA as being problematic, but there is no specific mention of it in the other studies.

• Specificity of association

Lack of flexibility in the shipping date does not specifically result in crunch, as it would probably have been inflexible for the duration of production. Moving the deadline forward does not specifically require the team to crunch, but depending on how much this compresses the schedule it could make crunch the more likely outcome, so there may grounds on specificity of magnitude.

However, as with feature creep, it is not the fact that the deadline is inflexible, or even that it is brought forward that would cause crunch, but rather it is the state the project is in when this occurs that is important. This strongly suggests lack of grounds for causation.

33 Anon 2 - LinkedIn
5. RESULTS & ANALYSIS 2 – AETIOLOGY

• Temporality of cause and effect

Crunch could follow the shipping date being brought forward, and it would not be likely that a project being in crunch would cause the shipping date to be brought forward, so this gives credence to the suspected cause.

• Gradient of exposure

Increasing the amount by which the date is brought forward increases schedule compression so could indeed increase the severity of crunch. Increasing the flexibility of the date could possibly allow for extension, reducing the amount of crunch needed, so there is grounds for gradient.

• Plausibility

It does make sense that altering the release deadline would impact the amount of crunch needed to complete a project.

• Coherence

The flexibility of the ship date or the act of moving cannot on their own cause crunch, but may increase the chance of a project already in trouble at that point doing so.

• Experiment

The variables associated with how much the date was changed would be easy to test, but since it is the condition of the project that is more important, it would be difficult to test the combinations to any real accuracy.

**Summary**

While the association is plausible, and the amount by which the date is moved can have an impact, it is the health of the project that is really the deciding factor as to whether a project crunches or not, so there is not enough validity to the claim that the shipping date causes crunch.

**5.7 Ambitious Scope**

This cause accounts for 5 coded constituents, being 9.26% of the total.

The scope of the project far exceeds the capabilities of the resources available. This
largely results from over-ambition, and can manifest as a huge uncontrolled feature list that the team knows it will not be able to complete by the deadline, or from commitment to a particular feature which then continues to increase in complexity.

“Partially, [the delays] were the result of the ambitious scale of the project: because the plot of Clancy’s evolving novel was driving our level design, we’d committed ourselves to creating sixteen completely unique spaces — a huge art load.”

**Analysis**

- **Strength of association**

  Out of 39 accounts given of crunch, 4 cite ambitious scope, which is a ratio of 1.33:13. This suggests a slightly low strength of association in the face of the other potential causes and the proportion of crunches where scope was not an issue.

- **Consistency of association**

  Unrealistic or ambitious scope is observed by Shirinian, Sheffield and Petrillo as being problematic.

- **Specificity of association**

  While crunch and scope are not specific to one-another, there may exist specificity of magnitude as crunch may be the more likely outcome of a truly over-scoped project. The main alternative would be to cut features and reduce the scope, but this may not always be viable.

- **Temporality of cause and effect**

  Over ambitious scope would precede a crunch, and it is doubtful that being in crunch would directly cause scope to increase.

- **Gradient of exposure**

  Increasing the scope of the project creates more work to do than a given time allows with the available resources, a key aspect of crunch, and so increasing the scope more would mean more crunch was needed to work through it all.

---

34 Rainbow 6 - Postmortem
5. RESULTS & ANALYSIS 2 – AETIOLOGY

- Plausibility of association

It does make sense that increasing the amount of work you are giving the team to do in a the same amount of time could lead to crunch.

- Coherence

The link between ambitious scope and crunch does not contradict what is known of either.

- Experiment

Experiment through altering the scope of a team's project would be simple.

Summary

Over-ambitious scope satisfies almost all the criteria for causation, providing a strong case for it being a cause of crunch. Its strength of association, while a low figure, is twice as high as that of ship date or project leadership, and a third higher than the others listed above. It is also regarded as an issue by other researchers. The association makes sense, the amount by which scope increases could affect the magnitude of the crunch, and this could be tested in controlled conditions.

5.8 Team Management

This cause accounts for 7 coded instances, being 12.96% of the total.

The management of the team is such that it cannot complete the project without difficulty. This is a result of overly cautious hiring leaving the team too small therefore increasing their workloads. Entering production with an inexperienced team increases lead times through reduced efficiency. Failure to communicate effectively with the team clouds their vision and increases the likelihood of encountering the other issues above.

“Despite all of this, we still didn't hire aggressively enough. Many years making small, tightly focused games had ingrained an efficiency bias and "smaller is better" mentality that was hard to shake. We were afraid of the additional management required to hire more people, and it resulted in a longer harder crunch for all of us.”

35 Rock Band - Postmortem
5.RESULTS & ANALYSIS 2 – AETIOLOGY

Analysis

• Strength of association

Out of 39 accounts given of crunch, 6 cite team management, which is a ratio of 2:13. This suggests a high strength of association in the face of the other potential causes and the proportion of crunches where the team was not an issue.

• Consistency of association

Petrillo cites team building and communication, Sheffield cites hiring and Shirinian cites team management as being problems in the industry.

• Specificity of association

Team management is not specific to crunch or to the games industry so there is no grounds for specificity.

• Temporality of cause and effect

Poor team management early in development could result in a team that is not equipped to meet the demands of late-stage production pressures and so has to crunch. However, the nature of crunch's impact on the team will generate team management issues so there is no grounds for temporality.

• Gradient of exposure

How well something is managed can be hard to assign to a gradient, especially when that something is a team. Being a better manager or a worse manager would be expected to yield better or worse results, but it would be difficult to gauge and there is no direct link between the extent of management ability and crunch.

• Plausibility of association

Knowing how crunch affects a team's performance makes it very understandable that mismanagement of a team could reduce their morale and their efficiency, which increases the likelihood of them crunching.

• Coherence

The link between team management and crunch does not contradict what is known of either.
5. RESULTS & ANALYSIS 2 – AETIOLOGY

- Experiment

It would be possible to experiment with teams of different sizes, and to some extent differing capabilities to measure their effect on the project and its relationship to crunch, though accurate data may be hard to gather in some cases.

**Summary**

It has been shown that a more cohesive team with a more positive attitude can be less susceptible to the rigours of crunch and even avoid it through dedication to their work. Strength of association is high and it can be seen that team management is thought of as an issue in the industry, but its direct causal link to crunch is not clearly defined and so caution must be maintained in asserting causation.

### 5.9 Time Management

This cause accounts for 19 coded constituents, being 35.19% of the total.

There is a fundamental lack of understanding of the amount of time it takes to complete tasks in a project. This means an accurate schedule cannot be developed or maintained, priorities cannot be assigned, and the actual state of the project difficult to determine.

The person or team responsible for generating the task under-estimates the amount of time needed, which inevitably makes the project schedule too short and too tight. It can lead to teams accepting contracts they believe they can complete by the stated deadline because they assume they can complete all the necessary work in time. Ultimately it can leave leads and teams with no idea how complete various tasks on the project are, how complete the game is, and how much time they need to finish it. This means that the schedule cannot be updated accurately, tasks prioritised nor information disseminated throughout the team and stakeholders.

“Worse yet, we had no idea of how much we had left incomplete as we went through each level, to the point that we could only guess that there was no way we’d be finished by the established milestones, and we were far too proud and afraid to bring this issue forward”

“Our engineers never had an accurate development schedule -- the schedules we had

---

36 Magicka - Postmortem
5. RESULTS & ANALYSIS 2 – AETIOLOGY

were so obviously wrong that everybody on the team knew there was no way to meet them. Our leads often lied to management about progress, tasks, and estimates, and I believe this was because they were in over their heads and weren't responding well to the stress. Consequently, upper management thought the project was going to be stable and ready to ship long before it actually was, and we faced prolonged crunch times to deliver promised functionality.”

Analysis

• Strength of association

Out of 39 accounts given of crunch, 15 cite team management, which is a ratio of 5:13. This suggests a very high strength of association in the face of the other potential causes and the proportion of crunches where time management was not an issue.

• Consistency of association

Petrillo cites delays and optimistic schedules and Shirinian cites estimation as problematic. IGDA notes that estimation of tasks and scheduling were commonly offered as reasons for crunching. Estimation and scheduling are common complaints in software development generally.

• Specificity of association

Time management is not specific to game development or crunch, however given that so much of crunch as a phenomenon involves some factor of time, there may exist specificity of magnitude.

• Temporality of cause and effect

Poor time management can certainly precede crunch, and it is unlikely (but not impossible) that crunch would lead to poor time management. An exception may be similar to that in section 5.5, where way of life may mean continued exposure to crunch breeds poor time management given the knowledge that crunching can be employed to compensate for it.

• Gradient of exposure

As with team management, determining a gradient for poor time management could be

37 Gabriel Knight 3 - Postmortem
difficult. There is no evidence in the data that really bad time management made crunch more or less a factor, and teams with smaller scale issues still crunched.

- Plausibility of association

  It does make sense that poor time management can lead to crunch, as it pushes tasks further and further towards the deadline as the whole project slips behind schedule.

- Coherence

  The link between time management and crunch does not contradict what is known of either.

- Experiment

  It may be possible to experiment with different scheduling methodologies or software, or different padding of estimates, but it may not be easy to get useful data.

Summary

Strength of association is overwhelmingly high, 2.5x higher than even its closest counterpart. It is also 5x higher than feature creep and 7.46x higher than shipping date, both of which seem highly plausible and are widely held to be causative. It must be borne in mind that grounds for causation in several of the other criteria is tenuous and dependent on other factors. Time management is indeed a strong candidate for causation, if not absolute, and out of all the causes may represent the greatest threat to project success.

5.10 Composite Summary

Four of the above issues present relatively strong cases for causation of crunch.

Team management is commonly seen as an issue, which is not a surprise given the multi-disciplinarity of game development teams. A direct causative impact may not be clear, but the high plausibility coupled with team management's ability to nurture the environment in which people find themselves when crunching gives it credence.

A late change of direction, although an uncommon event, can very easily lead to crunch through the sudden increase in workload late in the project's life. The projects citing this as a cause did so almost accepting crunch as a reality when they made the decision to make the change; a realisation that the change doubtless meant crunch was necessary.
5. RESULTS & ANALYSIS 2 – AETIOLOGY

Time management has a very high strength of association with crunch which again is unsurprising given the role time plays in crunch, and although as with team management the direct causal link is dependent on other factors, this is because time management can actually be seen as the cause of several other issues in development.

Ambitious scope has the strongest case, with strength of association, high plausibility and direct links to what is understood of the way in which crunch manifests in a project.

Note – For an Ishikawa diagram of the strongest cases for causation, see Appendix 5 – Causes of Crunch
6. DISCUSSION

6.1 What is Crunch?

There is a considerable gap in the knowledge of existing literature when it comes to crunch, largely because there has been no in-depth study of the subject. When it is discussed it is from myriad different experiences and so equally as many perspectives. This is what leads people to simplify it, expressing it as unpaid overtime or a rush to meet a deadline.

This study has shown that crunch does indeed constitute overtime, lots of it, and when unpaid this represents a strong argument in its favour from the point of view of a company's finances. It also shows that the relationship between crunch, employee and project runs extremely deep, generating strong emotional and experiential ties between the three that can bond or break a person's association to the game being made and the industry. This relationship goes beyond a single project and can mould a person's perception of how they develop games, the effect of one crunch having lasting effects that transfer to other projects even beyond the same company. The effects of crunch on the team are overwhelmingly negative, impacting health, morale, psychological well-being and ultimately the ability of people to successfully make games.

The tangible impact on the project is less often vocalised. It can stoke the passion people have for making games to the point where sheer determination sees them through crunch to complete the game. Being pragmatic, crunch as a tool does usually get the game finished, and that alone may be justification enough to employ it. When partnered with the possibility that it can save thousands of dollars per employee per year it makes an extremely strong business case; the ability to complete the game in less time for less money with most if not all of the features the public demands.

It raises some very interesting questions, not least of which is the issue of whether the effect on the team matters at all. If they are salaried and working in territories that decree they be exempt from compensation for overtime, not paying them for crunch is perfectly legal. They cannot be forced to stay the long hours crunch entails, so there is nothing to stop them leaving to go and get some sleep or spend time with their families.

What keeps them there is the prospect of career advancement or reward which may or
may not materialise. It is the sense of guilt, either self-imposed or thrust upon them, at
leaving behind their colleagues who are staying to crunch for whatever reason. It is the
desire to see the game finished, which goes beyond the sense of satisfaction at a job
completed but comes from game development as a creative endeavour and the desire to
see the game take shape and ultimately made real in front of them.

The data shows though that crunch doesn't always get the game finished, and
sometimes crunch has no actual positive impact on the project at all. This leaves the team
feeling used and disenfranchised, and all the physical and mental strain they endured has
no positive balance and so overcomes them, leaving their view of future projects jaded,
triggering the onset of stress even sooner once the initial signs of crunch are detected.

This is the most disconcerting impact on the project, as a team in this condition does
not work efficiently, their cohesion as a team is reduced, communication stutters and
dissent spreads. This leads to delays, and the project finds itself once again in a position
where it must crunch to avoid breaking the game apart.

6.2 What Causes Crunch?

The study of causation yielded some interesting results.

As said by Paul Ross in this paper's opening quote (Ross, 2011), 'better scheduling' is
not the answer. Time management, incorporating not only scheduling but estimation and
even to some extent knowledge management was considered by most in the data to be
the cause of crunch. There was the temptation to break that category down into smaller
thematic clusters but they were all so tightly interconnected that to do so would have
suggested that the relationship did not exist.

Scheduling alone is of course important, but if the estimates given to develop the plan
or work breakdown structure are wrong the schedule will be flawed. It is possible to be
shrewd assigning float to tasks to account for this, but if the estimates are bad because
your team doesn't communicate you have a deeper problem and scheduling is not going to
fix it. This suggests why time management and team management have caveats attached
to their declaration of causation, as there are strong cases for both but their intricacies
cloud a direct, tangible link to crunch.

Big surprises came in the form of feature creep and shipping date. Both are widely
considered problematic and are highly plausible candidates for causation, but laid bare as they were they do not present a sufficiently strong case. One of the advantages of the Bradford Hill criteria is that they force you to consider a cause purely on its own characteristics, in isolation, and to be thoroughly critical of the link between it and the effects that are witnessed.

The act of adding features to a game over the course of production is not in itself grounds for causing crunch, nor is the act of bringing forward or making immovable the shipping date. Both depend too heavily on the state of the project and how it is being managed.

The issue of a late change in direction is interesting too, as it is similar to the issue of being over-scoped. The impact of the major change late in production is often a sudden increase in the scope of the game, possibly a new feature set or release on an additional platform, which over-scopes the game in light of the remaining production time available. The fact that it does not occur often did not significantly affect its candidacy for causation, as with the decision to make the change came consideration or a realisation that the project would have to go into crunch to accommodate it.

That is also one of the factors in making over-scoping the most legitimate cause; you can over-scope a game even before production begins, to the point where you will have no choice but to crunch to complete it. Although over-scoping can share the same dependencies on other factors as some of the other causes, a direct link to crunching exists when considered in isolation and it does not have to be dependent on the state of the project beforehand.

It will be shown however, that this is still not the root of the problem, and why completion of Objective 5\textsuperscript{38} is not possible.

6.3 Why is Crunch Still a Problem for Modern Project Management?

The original purpose of this paper was to find out what crunch was and why it was a problem so that a recommendation could be made as to a project management tool or methodology that could be applied to solve the problem. The reason it was still giving the industry trouble after so long was that it wasn't fully understood; how it manifested or what

\textsuperscript{38}Propose the application of relevant project management devices to remedy
6. DISCUSSION

caused it, so no-one knew where to begin fixing it. This study described and defined crunch, then used that to determine its cause, ambitious scope being the most legitimate.

It must follow then that whichever tool or methodology can be applied to solve the problem of over-scoping should also solve crunch, and the absence of that tool should explain why crunch is still a problem today.

A recent paper by Bjarnason et al (2012) tackles the issue of over-scoping in market-driven software development. They discovered during a case study that implementation of agile development practices noticeably made the effects of over-scoping 'more manageable'. Therein lies the real issue. Making something more manageable is a mitigation strategy, arguably no different to employing younger developers with fewer family ties so they don't mind being at work and crunching, or providing laundry services and tents at your desk so you don't need to go home at all.

This paper has no intention of recommending another mitigation strategy for crunch, because it doesn't solve anything.

If we look then at what was said to cause over-scoping:

### Table 5 – Causes of Over-Scoping

<table>
<thead>
<tr>
<th>Constituents from this Study</th>
<th>Bjarnason et al, 2012</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Excessive uncontrolled feature list</td>
<td>Continuous requirements inflow via multiple channels</td>
</tr>
<tr>
<td>2. Exceeds resource capability</td>
<td>No overview of software resource availability</td>
</tr>
<tr>
<td>3. Over-ambition</td>
<td>Low development team involvement / Requirements not agreed with dev team</td>
</tr>
<tr>
<td>4. Commitment to overly complex mechanics</td>
<td>Detailed requirements specification upfront</td>
</tr>
</tbody>
</table>

1. Essentially feature creep. In Bjarnason it is over a longer period of time, increasing scope as the project progresses. In this study it is mainly associated with the inflation of the original plan in a short period of time with unchecked suggestions.

2. The requests / suggestions are made with the assumption that it will not be a problem to add them to the project, totally unaware of the limitations present in
resources available, whether it be manpower, tools, time or other variables.

3. The unchecked ambition and poor estimation comes from lack of communication with those who are ultimately responsible for implementing those features.

4. The commitment to a rigid plan, specification or game-play feature which cannot later be altered and which is resource intensive.

Here we see that within the issue of over-scoping also lie feature creep, team management, time management and the absence of a good project lead. Poor team management leads to the lack of communication with the relevant teams, leading to poor time management as the features are not based on sensible estimates made by those responsible for implementing them, leading to over-ambition and feature creep, none of which is being held in check by a project lead who should have one eye on the feature requests and the other on the entire project, its resources and it deadlines.

Valve don't even have managers, let alone project management (Valve, 2012). There is less need for them to do so as they have taken 'time' out of the triple constraint. Valve games are released 'when they're ready' so time doesn't constrain their development in the same way. They can focus more attention on resources by nurturing the creativity and job satisfaction of their employees and so over-scoping fails to be a problem. There's never any need to crunch because there's no temporal end to the project until it is finished, and the project cannot become over-scoped if you can always take extra time to incorporate all the features.

At the other end of the spectrum is a company like Electronic Arts. EA publish a large number of franchises, and release dates are highly market-driven so 'time' is extremely constrained. Over-scoping is less of an issue to an extent because each successive title only needs marginal improvement upon the last; a graphical upgrade, occasional new features, but in the example of their sports franchises, updating the game for the current year's sports season. This reduces the reliance on creativity as an asset.

Unfortunately, there is still the issue that even where time is extremely short, sometimes there is nothing you can do about it. It fits the old project management joke of a pregnant woman giving birth in nine months being able to do it in one month if there are nine women. In software development, and in many other project-driven industries, some processes take a fixed amount of time no matter how many people or computers you have
working on it. If it takes 18 months to make a new football game but they want it on shelves in 12, there is going to be 6 months of crunch. Perhaps more.

Fortunately, if creativity is not an important requirement you can burn out your developers with crunch and still recruit a fresh batch next year. This is what Erin Hoffman (Hoffman, 2004) wrote about in her EA_Spouse blog, the article that brought crunch and quality of life in the games industry to the fore, and prompted IGDA’s white paper (IGDA, 2004).

It is therefore possible to have no management and no crunch, and very well defined management and lots of crunch.

As mentioned above, a benefit of the Bradford Hill criteria is the ability to look at a cause in isolation and judge it thus, yet this can also be seen here as a hindrance. The aetiological study in section 5 assesses each cause as if it were acting upon a controlled project, where the various constituents of the cause itself could be measured against what is known of crunch to determine causation, yet each of the causes uncovered by this study is intricately related to the others in some way. Even those that were not seen as directly being causative of crunch such as feature creep or shipping date can, when in combination with one or more of the others, lead the project into such a state as to be unable to avoid crunch.

The application of a project management tool or methodology to any of these issues has little chance of solving that issue itself, let alone solving the issue of crunch, since they cannot explicitly be said to occur in isolation. To quote one of Bjarnason’s interviewees:

“We still have over-scoping in all projects. But, it is more controlled now and easier...”

It is still the case that all nine causes in this study were stated to have caused a crunch at some stage, so to consider all nine causes each potentially being themselves caused by one or more of the others, it is no wonder that crunch is still a problem today.

6.4 Crunch Est, Igitur Crunch Existit

In section 2.1 it was shown through etymology that crunch has its genesis in project management, and in 2.3 it was shown that project management accounts for a significant proportion of issues in game development. It is no surprise then that the three most

39 Interviewee Sg, Bjarnason et al, 2012
commonly cited causes of crunch make up the three sides of the triple constraint; Time (time management), Resources (team management) and Scope (ambitious scope).

What is surprising is just how deep the association between crunch and project management actually runs.

Figure 8 shows the proximity of the coding of project management references to the invariant constituents that make up the descriptions of crunch. 25% of the invariant constituents, those essential moments that describe the experience, emotion, rationale of crunch, are in the same paragraph as discussion of project management. Another 25% are found one paragraph away, and 10% two paragraphs away. Half of all discussion describing crunch is almost in the same breath as discussion of project management.

That is because crunch is project management in the games industry.

It is a spectre present behind each critical project management decision, representing knowledge that if this tool fails, or that methodology is misapplied, at least we can always crunch to fix the problem. Nobody ever really wants to do it, and those who do crunch end up hating it and vowing never to crunch again, but as long as crunch exists as a last resort, that most times genuinely does get the game finished, it will always exist.
6.DISCUSSION

The problem with completing Objective 6\textsuperscript{40} is that crunch and project management are so deeply entwined that they are almost propping one-another up; remove project management altogether and crunch will just run away with itself. Commit yourself to removing crunch and you suddenly find yourself having to learn a lot about project management very quickly.

The key lies in removing crunch as an option. At no point in development should a lead consider crunch over sensible alternatives. What those alternatives are will depend on the situation and the state of the project, and the lead needs to be aware that what seems the most obvious issue at the time is not necessarily what is causing the problems. It begs the question of how many of the teams saying that feature creep was the cause of crunch were simply unaware that they may have had deeper issues to do with estimation and communication within the team.

This study shows that there is a need to increase the standard of project management knowledge in the industry if it is to have any hope of removing its dependency on crunch for solving its problems, which completes Objective 7\textsuperscript{41}. It is exactly as Musil et al (2010) said:

\begin{quote}
\textit{“It is therefore not enough to just blindly apply the same proven techniques and best-practices […] but to countercheck if adaptations or complete redesign are necessary in order to serve game developers well.”}
\end{quote}

6.5.1 Recruitment

Recruitment in the games industry is typically from within, where the easiest route is as a tester or other low level QA position, or maybe into a developer role with a relevant degree. It is preferred that people earn their dues in the trenches (de Peuter and Dyer-Witheford, 2006) and it is understandable that studios want their members to have experience in the games industry at several levels before leading a team.

This results in leads who are well versed in the technical aspects of development, and project management skills learned ‘on the job’. It does not allow for new thinking in the field of project management to enter the industry, and it means that teams who crunch breed

\begin{footnotesize}
\textsuperscript{40} Explore how their current absence explains the state of PM in the industry
\textsuperscript{41} Demonstrate the imperative for further study
\end{footnotesize}
team leaders who use crunch. They may resolve never to put a team through what they experienced before, but if you remove crunch from the equation you need to learn a lot about project management, and without that knowledge you run out of alternatives sooner and crunch becomes a viable option once more. It takes the formation of a new company to make that kind of commitment (Remo, 2009; Hyman, 2008b).

Jobs at project management / production level require significant experience in the games industry, several years in many cases. Looking at the first few vacancies listed on the job boards at GamesIndustry.biz reveals the following requirements:

“Gaming industry experience is a must. Experience in a similar Producer/Project Management role for online browser game(s) is preferable.”

“Previous experience on Multiplayer games, at least one title released. 5+ years of experience in the games industry.”

“Several years of work experience in a leading position within the Games industry.”

The opinion seems to be in several cases that acceptable project management experience can be learned while making games, but an experienced project manager can't learn how to manage making games without first having made games.

Recruitment at this level needs to be opened up beyond the games industry. Since this is the first in-depth study of crunch the field of project management is not currently familiar with crunching, and so it would not be considered an option.

6.5.2 Training

Training ought to be extended to the entire team where possible. If they are given increased awareness of the relationships that exist between departments, disciplines, teams, team members and so on it will improve the efficiency of the resources within the project. It could reduce the careless addition of features as everyone is more aware of who is responsible for implementing them and what they actually require, improving communication overall. This applies to leads too, as it improves their decision making abilities and the likelihood that they will just say no.

“Most games that run late or require excessive 'crunch time' are victims of feature creep. This is solely the fault of the producer, who must be objective and, even at the risk

42 http://www.gamesindustry.biz/jobs/by-category/production
of bruising a few egos, stand up and say 'stop'.” (Hight and Novak, 2008)

This can be achieved through the use of experiential project management simulations or 'business games'. These are training exercises simulating various aspects of an operation where the team must usually work together in a realistic way to achieve a shared objective. They often encourage creativity in problem solving by being open-ended in how the goal is achieved and so would work very well in training people from creative industries (Tanner et al, 2012).

It makes sense to use games to train people who love making games, and the opportunity to be creative while actively learning project management techniques, even if it is only gaining an appreciation of what a team-mate does, will increase buy-in of project management and make the entire team more efficient and effective.

6.5.3 Knowledge Management

To truly improve the standard of project management in game development, the industry needs a dedicated peer-reviewed journal. A repository for academic knowledge on the subjects of production, management, methodologies, best practice, and tools specific to their application in the games industry will make research such as this much easier for those that follow.

This study aims at opening discourse on the subject of crunch as a project management phenomenon, and the knowledge generated from academic and professional study of the issue ought to be available to those who are affected by it on a vocational basis.

If studies on the games industry were to be collected into one or a series of journals for that specific purpose as opposed to being kept among the fields from which they are conducted (literature, psychology, media etc) the level of knowledge in the industry would increase at a greater rate and issues such as crunch not left decades to take hold.
Crunch is complex. Far from being as simple as unpaid overtime or frenzied activity near a deadline, it has been revealed by this study as a phenomenon so intricately a part of project management that further research must be conducted if either PM or the games industry is to fully understand it.

By Objective:

1. The literature review demonstrated a stark lack of knowledge of crunch, the understanding of which was hampered by the broad but shallow research thus far conducted on the games industry, and for the first time showed the origin of the phenomenon in project management.

2. In spite of the lack of an academic body of knowledge, there exist contemporary sources such as game postmortems and anecdotes detailing encounters with crunch, from which valuable data was gleaned.

3. Phenomenology showed that crunch extends into the lives of those who encounter it beyond wages and long hours, affecting them physically and psychologically so that decisions they make on projects years later are affected by the experience; the first time such a study was conducted.

4. Aetiology showed causation of crunch to be extremely complicated. Commonly held beliefs about causative circumstances were shown to be unfounded, and even once the most legitimate cause of crunch was determined, there were uncovered intricacies to the phenomenon that contribute to understanding why crunch is still an issue today.

5. Consequently, these intricacies preclude the application of a single tool or methodology as a solution.

6. The absence not only of a solution but the ability to apply one shows that the standard of project management in the games industry needs to be raised, and suggestions were made on how to do so.

7. This paper therefore represents a clear imperative for further study into project
7. CONCLUSION & FURTHER WORK

management in game development, and crunch in particular.

Implications

This paper has taken the first steps towards filling the gap in academic literature on crunch and on the study of game development, and it is hoped that it will serve as a foothold from which much-needed research can be launched. The true issue behind crunch shows that the field of project management has a great deal to offer an over-looked industry that is heavily project-driven, and the complexities of crunch present opportunities for other fields to conduct research too.

The findings of this paper contradict assertions made in the existing literature regarding the nature and onset of crunch, while reinforcing others that call for different approaches to be taken and modifications made to superjacent methods. They show that the tendency toward mitigation instead of correction is not sustainable, for as long as crunch exists as an option it will never be eradicated, and that even though a pragmatic argument exists for making use of crunch, it cannot be held up beneath the weight of the overwhelming negative effects on the team and on the project.

The findings here and those which will follow from the research of others will hopefully contribute to improving the quality of life for game developers and increasing the efficiency and effectiveness of development itself through the influx of new skills, ideas and experience from outside the industry, resulting in improved job satisfaction, skill retention and collective knowledge leading to the production of better games.

Limitations

This study only looked at postmortems where crunch was explicitly mentioned, and so does not take into account games encountering issues that shared some characteristics but were not called crunch. All of the games covered by the postmortems were completed and released and so show nothing of the relationship (if any) between crunch and project failure.

Not all postmortems contained instances of each type of data, meaning durations, hours worked, experiences etc were scattered across them all inconsistently. This prevented analysis of the phenomena against other metrics, as while each document included a release date for the game, platform developed for and team size, the data was so...
7. CONCLUSION & FURTHER WORK

scattered that the gaps between them meant they could not be used to draw accurate conclusions.

Use of extensive interviews in place of anecdotal evidence may have yielded better data to be analysed using phenomenology, through focussed questions drawing out more definitive horizons of experience. It would also have allowed improved verification of the horizons and invariant constituents with the subject to ensure that the reduction process did not reduce the actual meaning.

The Bradford Hill criteria have not previously been used in this context so their adaptation to software project management was open to interpretation, even with the good inter-coder agreement score. The aetiological analysis does not take into account the interdependencies the causes exhibit, resulting in dozens of possible permutations which could not themselves be tested.

Recommendations

1. To address the limits of this work, it is recommended that more varied sources are covered. It would be interesting to see how crunch had developed over time, how it was affected by team size or platform, and what its relationship may be to project failure rates. It would also be prudent to include a larger number of projects from recent years, and to include ones where crunch is not explicitly mentioned to determine if the same problems occur and how they avoid crunching. It would be very useful if the permutations of the various causes could also be studied to determine their various impacts on the project and the implications for crunch.

2. From the discussion section of this paper, it is recommended that this research be applied to raising the standard of project management in the games industry. If developers dislike the rigidity of PM or just find management impedes their creativity then a compromise needs to be found. Methods need to be discovered that increase team buy-in so that they understand the decisions a lead has to make and the responsibilities other team-members have in relation to their own. Most game developers play games when not working on them so the use of games in training them should be explored.

3. Where discourse is concerned, simply put, there needs to be more. The field of
7. CONCLUSION & FURTHER WORK

game development needs to recognise more the importance of project management in how it operates, and project management needs to appreciate game development for being separate from software development in interesting and unusual ways. The same tools that work for one may not for the other, at least without modification and specialisation. Producers and project managers in the games industry need an arsenal of options available so that crunch is no longer considered one of them.

The most important lesson this research has to teach, and which therefore merits repeating:

_As long as crunch exists as a last resort, that in most cases genuinely does get the game finished, it will always exist._
References


References


References


<table>
<thead>
<tr>
<th>TITLE – DEV</th>
<th>YEAR</th>
<th>PLATFORM</th>
<th>PUBLISHER</th>
<th>TEAM SIZE</th>
<th>DEV TIME mths</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age of Empires 2: AoK – Ensemble</td>
<td>1999</td>
<td>PC</td>
<td>Microsoft</td>
<td>40</td>
<td>24</td>
</tr>
<tr>
<td>Bioshock – 2K</td>
<td>2007</td>
<td>XBOX360 / PC</td>
<td>2K</td>
<td>84</td>
<td>36</td>
</tr>
<tr>
<td>Bit Trip Beat – Gaijin Games</td>
<td>2009</td>
<td>WiiWare</td>
<td>Akys</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Brutal Legend – Double Fine</td>
<td>2009</td>
<td>XBOX360 / PS3</td>
<td>EA</td>
<td>74</td>
<td>54</td>
</tr>
<tr>
<td>Conduit 2- High Voltage</td>
<td>2011</td>
<td>Wii</td>
<td>Sega</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crackdown – Realtime Worlds</td>
<td>2007</td>
<td>XBOX360</td>
<td>Microsoft</td>
<td>71</td>
<td>48</td>
</tr>
<tr>
<td>Diablo 2 – Blizzard</td>
<td>2000</td>
<td>PC</td>
<td>Blizzard</td>
<td>40</td>
<td>36</td>
</tr>
<tr>
<td>Dungeon Siege – Gas Powered Games</td>
<td>2002</td>
<td>PC</td>
<td>Microsoft</td>
<td>27</td>
<td>44</td>
</tr>
<tr>
<td>Far Cry 2 – Ubisoft</td>
<td>2008</td>
<td>PC / XBOX360 / PS3</td>
<td>Ubisoft</td>
<td>265</td>
<td>43</td>
</tr>
<tr>
<td>Final Fantasy Crystal Chronicles:My Life as King – Square Enix</td>
<td>2008</td>
<td>WiiWare</td>
<td>Square Enix</td>
<td>19</td>
<td>13</td>
</tr>
<tr>
<td>Gabriel Knight 3 – Sierra</td>
<td>1999</td>
<td>PC</td>
<td>Activision</td>
<td>45</td>
<td>36</td>
</tr>
<tr>
<td>Gunstringer – Twisted Pixel</td>
<td>2011</td>
<td>XBOX360</td>
<td>Microsoft</td>
<td>20</td>
<td>12</td>
</tr>
<tr>
<td>Heavy Gear 2 – Activision</td>
<td>1999</td>
<td>PC</td>
<td>Activision</td>
<td>20</td>
<td>19</td>
</tr>
<tr>
<td>Heavy Metal FAKK 2 – Ritual</td>
<td>2000</td>
<td>PC</td>
<td>Gathering of Developers</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>Hydrothunder Hurricane – Vector Unit</td>
<td>2010</td>
<td>XBLA</td>
<td>Microsoft</td>
<td>7</td>
<td>14</td>
</tr>
<tr>
<td>Igneous – Digipen</td>
<td>2009</td>
<td>PC</td>
<td>Going Down in Flames</td>
<td>6</td>
<td>16</td>
</tr>
<tr>
<td>Little Big Planet – Media Molecule</td>
<td>2008</td>
<td>PS3</td>
<td>Sony</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>Magika – Arrowhead</td>
<td>2011</td>
<td>PC</td>
<td>Paradox</td>
<td>12</td>
<td>24</td>
</tr>
<tr>
<td>Octo Dad – DePaul Game Experience</td>
<td>2011</td>
<td>PC</td>
<td>Self</td>
<td>19</td>
<td>6</td>
</tr>
<tr>
<td>Outland – Housemarque</td>
<td>2011</td>
<td>XBLA / PSN</td>
<td>Ubisoft</td>
<td>12</td>
<td>22</td>
</tr>
<tr>
<td>Rainbow 6 – Red Storm</td>
<td>1998</td>
<td>PC</td>
<td>Self</td>
<td>16</td>
<td>21</td>
</tr>
<tr>
<td>Roboblitz – Naked Sky</td>
<td>2006</td>
<td>PC / XBLA</td>
<td>Self / Microsoft</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>Rock Band – Harmonix</td>
<td>2007</td>
<td>XBOX360 / PS3</td>
<td>MTV / EA</td>
<td>100</td>
<td>24</td>
</tr>
<tr>
<td>Sin – Ritual</td>
<td>1998</td>
<td>PC</td>
<td>Activation</td>
<td>17</td>
<td>20</td>
</tr>
<tr>
<td>Skittleball – Magnin and Associates</td>
<td>2010</td>
<td>iOS</td>
<td>Self</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Spiderman: Shattered Dimensions – Beenox</td>
<td>2010</td>
<td>XBOX360 / PS3 / PC / Wii</td>
<td>Activision</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>Spllosion Man – Twisted Pixel</td>
<td>2009</td>
<td>XBLA</td>
<td>Self</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>Stubbs the Zombie – Wideload Games</td>
<td>2005</td>
<td>XBOX / PC</td>
<td>Aspyr</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>Super Meat Boy – Team Meat</td>
<td>2010</td>
<td>PC / XBLA</td>
<td>Self / Microsoft</td>
<td>3</td>
<td>18</td>
</tr>
<tr>
<td>System Shock 2 – Looking Glass / Irrational</td>
<td>1999</td>
<td>PC</td>
<td>EA</td>
<td>30</td>
<td>18</td>
</tr>
<tr>
<td>The Path – Tale of Tales</td>
<td>2009</td>
<td>PC / Mac</td>
<td>Self</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>Ticket 2 Ride: Pocket – Days of Wonder</td>
<td>2011</td>
<td>IPhone / iPod Touch</td>
<td>Self</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Time shift – Saber Interactive</td>
<td>2007</td>
<td>PC / XBOX360 / PS3</td>
<td>Sierra</td>
<td>142</td>
<td>48</td>
</tr>
</tbody>
</table>

### Anon Source

<table>
<thead>
<tr>
<th>Designations</th>
</tr>
</thead>
<tbody>
<tr>
<td>LinkedIn</td>
</tr>
<tr>
<td>VideoGames 24/7</td>
</tr>
<tr>
<td>The Daily WTF</td>
</tr>
<tr>
<td>Tales From the Trenches</td>
</tr>
<tr>
<td>Email</td>
</tr>
</tbody>
</table>

Anon 1, 2
Anon 3
Anon 4, 5, 6, 7
Anon 8 – 21
Anon 22
Appendix 2 – Sample Coding

The worst crunch I ever encountered was in shipping a game based on a popular television show. It was a real-time hybrid simulation/RPG with our brand new state engine (still in development, patch). One critical task was pathfinding, which broke almost everything else downstream. If an NPC needed to find the PC to issue a quest that had a tight duration, and that NPC could not navigate out of the room to find the PC, the quest wasn’t issued, or it was issued so late that the PC could not complete it in time, and it all looked like scripter issues. But it was all pathfinding.

Pathfinding was one week away, according to the programmer assigned to it. It stayed that way for almost three months. That programmer then went on a scheduled well in advance vacation. I tried to get management to hire a coding friend of mine with pathfinding experience to finish it during the vacation, but then that programmer would have been offended, and he was best friends with the lead programmers, and both would have left, according to the producer.

About six months after pathfinding was starting to be a problem, we completely rebuilt the geometry to make for more room, to work around the pathfinding and collision detection issues. Crunch continued. We had to discard lots of story design because time needed for a different environment was spent re-rendering the main environment.

Four months later, the lead designer left the team, semi-voluntarily (he’d found another job). Pathfinding was still a major issue. Shortly thereafter, after about six months of crunch, all the programmers quit. Without them, the game would not ship. Desperate management hired them back as contractors to ship the game; I don’t know the full terms, but I gathered it was a huge raise.

Almost a year after pathfinding and collision detection started to be an issue, we finally could finish the game repeatedly. It only took two months to polish and finalize the game at that point. Crunch had lasted nine months. QA, anticipated being hired for no more than six months, and hoped for only three months, had to be paid for almost nine months. The company managed to survive. The game was a flop, since by the time it came out the star of the game, the guy on the box cover, had left the TV show.

During the crunch I drove 100 miles round trip on many a Saturday to show solidarity even though I, as a designer, had little I could do. There were occasional bursts of extremely hard work as we had to stitch the story line back together based on already recorded dialog and removed game levels or features, but most weeks in crunch I did not have much more than 40 hours of useful work to do, let alone the 50-60 I was putting in. For programmers and QA, I’m sure the crunch was much worse.

My takeaways:

1) If somebody will be offended because you take a task away from them, and the offense is bad enough to cause multiple employees to
quit, rip the band-aid off as early as possible.

2) Be aware of systems bugs that present themselves as scripting bugs.

3) Don’t make people crunch except when they have something to contribute.

4) Give mission critical tasks to the lead once the estimated time to completion has been exceeded by 100% of the original estimates. Challenge the lead to find other tasks that can be given to his buddy/underling. Don’t let a task take 1200% longer than the time estimated while still having one programmer mainly responsible for it.

5) Amenities provided during crunch by management went a long way to easing resentment of the crunch, at least for me. Management was trying to ship a game without making every employee quit.

6) Nine months of crunch for non-exempt employees is lots of overtime and extra money. Nine months for exempt employees is extra travel expenses leading to resentment.

7) Don’t make everybody crunch all the time except when there is useful work to do. It bears repeating.
### Appendix 3 – Inter-Coder Reliability

<table>
<thead>
<tr>
<th>Variable (cols)</th>
<th>Percent Agreement</th>
<th>Scott's PI</th>
<th>Cohen's Kappa</th>
<th>Krippendorff's Alpha (nominal)</th>
<th>N Agreements</th>
<th>N Disagreements</th>
<th>N Cases</th>
<th>N Decisions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable 1</td>
<td>75%</td>
<td>0.467</td>
<td>0.467</td>
<td>0.5</td>
<td>6</td>
<td>2</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Variable 2</td>
<td>87.5%</td>
<td>0.709</td>
<td>0.714</td>
<td>0.727</td>
<td>7</td>
<td>1</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Variable 3</td>
<td>75%</td>
<td>0.333</td>
<td>0.385</td>
<td>0.375</td>
<td>6</td>
<td>2</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Variable 4</td>
<td>100%</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Variable 5</td>
<td>75%</td>
<td>0.5</td>
<td>0.529</td>
<td>0.531</td>
<td>6</td>
<td>2</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Variable 6</td>
<td>87.5%</td>
<td>0.746</td>
<td>0.75</td>
<td>0.762</td>
<td>7</td>
<td>1</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Variable 7</td>
<td>75%</td>
<td>0.333</td>
<td>0.385</td>
<td>0.375</td>
<td>6</td>
<td>2</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Variable 8</td>
<td>87.5%</td>
<td>0.709</td>
<td>0.714</td>
<td>0.727</td>
<td>7</td>
<td>1</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Variable 9</td>
<td>87.5%</td>
<td>0.709</td>
<td>0.714</td>
<td>0.727</td>
<td>7</td>
<td>1</td>
<td>8</td>
<td>16</td>
</tr>
</tbody>
</table>

Each variable accounts for a cause of crunch as determined through the process of extracting invariant constituents. Satisfaction or non-satisfaction of the BH criteria was reduced to a nominal value, and agreement with a 2nd coder was tested. It is important to remember that BH is not to be used as a list to be checked off, therefore this is a measure of how accurate the causes were as constituents to be suitable for analysis and not a measure of their causality.