

# The Feature Creep Perception in Game Development

**Exploring the role of feature creep in development methods and employee engagement**

**By: Erik Joachim Neuhofer and Samuel Zelenka af Rolén**

Supervisor: Mikolaj Dymek  
Södertörn University | The School of Natural Sciences, Technology and Environmental Studies  
Bachelor Thesis 30 hp  
Media Technology | HT2020/VT2021  
Computer Games Programme



# Upplevelsen av feature creep inom spelutveckling

**En undersökning om vilken roll feature creep har i utvecklingsmetoder och arbetsengagemang**

## **Abstract**

Game developers often find themselves coming up with ideas along the production period of a game varying in size and may go unnoticed or seem insignificant to the scope of the project but in the long run add up to break deadlines, budget and affect the morale and engagement of developers. In the modern game development industry agile development methods have increased in popularity allowing flexibility in the development process. This agile approach has emerged from traditional software development where waterfall development methods are common practice (Kanode and Haddad, 2009). Through in-depth interviews with developers from Sweden, Finland, and the United Kingdom this study aims to explore how feature creep is perceived by the individual developer and its effect on the day-to-day development. The ambition is to establish whether feature creeping as a phenomenon can be a useful tool for innovation and work culture.

Keywords: Feature creep, scope creep, game development

## **Abstrakt**

Spelutvecklare får kontinuerligt idéer under produktionen som varierar i storlek och kan gå obemärkta eller anses meningslösa för projektets ramar och över tid och kan påverka deadline, budget, moral och engagemang hos spelutvecklare. I dagens spelindustri har agila arbetsmetoder ökat i popularitet som möjliggör flexibla utvecklingsprocesser. De agila arbetssättet har vuxit fram ur traditionell mjukvaruutveckling där vattenfallsmetoder är ofta förekommit (Kanode och Haddad, 2009). Genom ingående intervjuer med utvecklare från Sverige, Finland och Storbritannien undersöker den här studien hur feature creep upplevs av den enskilda spelutvecklaren och dess effekt på det dagliga utvecklandet i spelbranschen med en ambition att se hur fenomenet feature creep kan vara ett användbart verktyg för innovation och arbetskulturen.

Nyckelord: Feature creep, scope creep, spelutveckling

# Table of Contents

<b>Introduction</b>	<b>5</b>
<b>Related Research</b>	<b>7</b>
Production	7
Organization Communication	8
Games as a Service	9
<b>Research Question and Purpose</b>	<b>11</b>
<b>Method</b>	<b>12</b>
Recruitment	12
Data Gathering	13
Data Analysis	14
<b>Results and Analysis</b>	<b>15</b>
Perception	16
Project Management	19
Communication	22
External Influence	24
<b>Discussion</b>	<b>27</b>
<b>Conclusion</b>	<b>30</b>
<b>References</b>	<b>31</b>
<b>Appendix</b>	<b>34</b>

# Introduction

Within software and game development, feature creeping is a known phenomenon that breaks the scope and deadlines numerous times in the production process (Kanode and Haddad, 2009), as well as making the end product over-complicated. Feature creeping in its simplest form could be described as a term for when non-planned features are added or emphasizing existing features in a project with the intent of making it better, but often affects the product negatively (Lee and Lee, 2015; Lee, Woods and Kidwell, 2006). Examples of feature creeping have been well-documented. Norman (1998) gives examples of the phenomenon both within and outside of software development that occurs during the development process of a product, especially but not limited to software development as Norman (1998) bring up products that have been affected negatively by feature creeping such as the personal computer and army knife, which are both outside and within software development. To prevent the negative miscommunications during the development process Hagen (2011) suggests examples and methods used to handle feature creeping to avoid pitfalls and catch the opportunities feature creeping can evoke through prototyping, “the one question” among others to share the vision and communicate the idea within the team.

The cause for the occurrence of feature creep could be divided into three recurring themes found through the related research involving: (1) *Production*, which covers the topic of the process and methods used to implement and evaluate features in the production stage of game development. (2) *Organization Communication* which has an important role as it affects the way features are prioritized and how unplanned features are implemented. It also brings up how developers communicate between each other encouraging new ideas and sharing progress. The communication between coworkers has shown to be a source for feature creep as possible changes to the game are often discussed between the closest team members (3) *Games as a Service (GaaS)* is when a game is released with the possibility to release updates and is becoming a more popular approach as a business model compared to the *Games as a Product (GaaP)* model which was the more popular approach since the launch of the first home consoles selling one game for one price (Vaudour and Heinze, 2020). The centralization of distribution on online platforms has changed the way players acquire games and game developers are no longer dependent on physical copies. The possibility to release unfinished titles on the market allows the developer to further iterate on a title and feature creeping

becomes more manageable with this approach as the scope of the final release becomes flexible as shown by Dubois and Weststar (2021) where it was presented that games in using the GaaS business model support and adapt quickly on input from the consumers.

The literature also discusses the problems in marketing, consumer experience, and project strategy aspects of feature creep. However, feature creep and how it is perceived by the individual game developer remains unexplored. The lack of sources from DiGRA, the prominent international association for game research, suggests that feature creeping in game development is not a highly researched area, even though it is a well-known recurring phenomenon within software development (Kanode and Haddad, 2009). There are positive effects of feature creep mentioned in the literature lifting innovative and creative ideas which can positively differentiate a product in its market (Nowlis and Simonson, 1996; Thompson and Norton, 2011).

To explore the perception of feature creep by game developers and how management handles feature creep through individual developers' ideas and suggestions in-depth interviews with developers. Nine interviews (seven video, one email, one live chat) were conducted with participants from Finland, Sweden, and the United Kingdom where the video interviews lasted a total of 266 minutes. The empirical data was analyzed using a thematic analysis (Braun and Clarke, 2006) to find the common themes.

# Related Research

## Production

Feature creeping appears in the game production phase during the game development process, due to the complexity compared to traditional software development processes (Petrillo and Pimenta, 2010; Blow, 2004), as game projects must handle multimedia assets among different teams and areas such as design, programming, sound, animation, and so on (Petrillo and Pimenta, 2010; Kanode and Haddad, 2009). Research covering feature creep has not arrived at a precise definition about the subject; Neither if the phenomenon is wanted or not (Lee and Lee, 2015).

In game development, many teams have adopted the agile software development (ASD) practice (Petrillo and Pimenta, 2010; Hagen 2011). The workflow is dependent on the developers in comparison to a waterfall software development (WSD) practice often used in industrial design (Petrillo and Pimenta, 2010). The ASD practice is presented as a workflow that can adapt and shift to certain needs along the course of development by Petrillo and Pimenta (2010). In both WSD and ASD methods developers use a single or multiple Game Design Documents (GDD) to document all aspects of the complete game. The GDD has been argued by Blow (2004) as a method that does not work well as it is difficult to foresee the complete design up front. Feature creeping is commonly claimed to be a product of poor project scope from the GDD and a poorly managed production phase (Kanode and Haddad, 2009). The ability to adapt and make changes in short iterations based on the project needs and expectations from developers can increase the risk of feature creep effects along the production, and over time the game will differentiate from the original design (Lee, Woods, and Kidwell, 2006). If feature creeping is not dealt with fast or goes unnoticed, its effect on development usually results in increased costs, waste of resources, loss in performance, more errors, and an increased chance of failure (Kanode and Haddad, 2009). Feature creeping may cause these problems, but studies (Austin, 2001) find that companies that regularly miss deadlines may also produce better long-term results in terms of quality.

During game production, feature creeping is inevitable since games at its core must be entertaining, which creates a dilemma among game designers when new unplanned features must be implemented (Lee, Woods and Kidwell, 2006; Kanode and Haddad, 2009). This can

vary during different parts of the project. Kanode and Haddad (2009) discuss that prototyping and iterating are probably inescapable, due to the indefinable nature of fun. This could present an argument that pre-production is reliant on feature creep to shape the game right from the start and to find the “fun-component”.

One more cause for feature creep can come from the individual developer’s passion for exhibiting excellence in game development, as discussed by Lee and Lee (2015). They bring up further that in connection to a “more is better” belief, developers might want to show off their knowledge and talent, so they add and emphasize them in their designs. Although, allowing or accounting for some feature creeping in a project might prove to be beneficial in terms of innovative approaches to problems (Lee and Lee, 2015) that might differentiate the product from competitors. Feature creeping can be minimized through project management strategies, observation of the suitability of a feature as well as thorough documentation (e.g GDD); Suggesting that by critically reflecting and evaluating if its real value versus the overall complexity, the fitness of the feature can be established (Lee, Woods and Kidwell, 2006; Kanode and Haddad, 2009). To meet the deadlines, developers take shortcuts to finish, which affects the quality of the product (Austin, 2001).

## **Organization Communication**

Through the different production phases in game development communication is important as the team size often varies throughout production (Hagen, 2011). Blomkvist, Persson, and Åberg (2015) have discussed how lack of communication within the organization can separate the developers' understanding of the consumer persona, which will in turn, have an effect on the usability of the product or service developed. It has been confirmed that internal communication results in higher employee engagement (Verčič and Vokić, 2017). However high engagement from employees increases the risk of implementing unplanned features according to Hagen (2011).

Thominet (2018) claims that modern game development applies open development strategies which involve opening up the development process to the consumer through *access*, *feedback*, and *transparency*. *Access* refers to the process of giving the consumer access to the product in the development stage. *Feedback* is when the consumer gets to communicate their thoughts about the product directly to the development team, indirectly letting the audience provide the



GDD (Thominet, 2018; Kanode and Haddad, 2009). Finally, *Transparency* refers to the communication process from the development team to the public which is a big part in communicating the vision of the project (Thominet, 2018). This communication process puts the consumer in a position where the community has a role in affecting the game design (Thominet, 2018). The ASD methods allow the consumer input channel in the development process, however, the vast amount of input from the consumer side puts the developers in a position where the decision must be made if wanted features fit in the design vision, Indirectly putting the consumer as a part of the development team (Thominet, 2018).

The communication methods have been divided by Hagen (2011) into verbal and non-verbal ways to express the design vision that goes beyond the GDD. The different communication methods could show the importance of communication to prevent the negative effects of feature creeping (Hagen, 2011).

Hagen's (2011) study covering how game developers communicate design ideas has shown that along the production ideas come from different parts of the production team containing designers, artists, engineers, and others. From interviews with the lead designer and senior producer at DICE (a prominent Swedish game development company) in 2011, it is clear that this creative approach of unexpected features from different teams in development is a welcomed phenomenon and is highly valued in the game industry.

## **Games as a Service**

Feature creep is a product of iteration where unplanned features over time find their way into the scope of the project. Therefore, the GaaS business model is becoming more important where a game title may be released in an unfinished state allowing the developer to keep iterating on new ideas even after launch (Dubois and Weststar, 2021). Which differentiate from the previously popular GaaP model where a shipped title is complete and sold as one price; one product under a set scope and deadline (Vaudour and Heinze, 2019). As well as online distribution channels are making the GaaS business model the dominant way of distribution allowing updates, downloadable content, and in-app purchases on distribution channels such as Google Play, App Store, Steam, Epic Games Store, Origin, Playstation Store, Microsoft Store and Uplay to name a few (Dubois and Weststar, 2021; Guiney and Xu, 2019). The GaaS model allows engagement from the consumers in the development process

and helps with maintaining a relationship with the consumer over time and creating perpetual ongoing revenue and value (Guiney and Xu, 2019). Vaudour and Heinze (2020) stress the point of GaaS as a way for companies to release unfinished products to the market allowing longer development lifecycles and the fact that it is cheaper to release updates than new games. Today the ease of online centralized distribution allows developers to transition from having to develop multiple sequels to having fewer titles to maintain over longer life-cycles. The extended life cycles allow a gradual complexity over time by small iterations of a feature which makes it hard for designers to detect feature creep. Lee, Woods and Kidwell (2006) encourage a measurement of fitness for features to make sure that a feature does not become dominant in a product in an environment where developers constantly adjust accordingly to input from marketing, sales, engineering, management and competitors to “improve” the product. Complexity in a product is not necessarily bad as studies (Carpenter, Glazer, and Nakamoto, 1994; Nowlis and Simonson, 1996; Thompson and Norton, 2011) have shown that the advantage in feature creeping is that complexity can promote a product as premium and support the value of the game. That is why extensive testing of a game is crucial for several reasons, one of them is that it may lead to new ideas, not showing in the early stages of development (Berg Marklund et al., 2019). Each wave of games evolves toward increasing technical complexities to give players a never-seen-before experience, leading to attempts of several technical feats (Blow, 2004). This carries risks for game developers that impacts the “designers dilemmas” of how the features will feel for the player if it will be worth the effort, cost, and how far to push it to be ahead of the competition (Lee, Woods and Kidwell, 2006; Blow, 2004; Thompson and Norton, 2011; Kanode and Haddad, 2009). As GaaS pushes towards an indefinite timeline, attempting implementation of new technology can be harmful for the game as the added complexity can affect the usability which is a well documented negative effect in design work (Norman, 1998; Nielsen, 1993).

# Research Question and Purpose

As shown by Norman (1998) feature creeping is not a new phenomenon and can be seen in most product designs. However, what puts game development in a unique position is where it blends between traditional software development and the entertainment business. Where the game developer must take in consideration cooperation with multiple roles such as sound engineers, story writers, gameplay design, producers (Kanode and Haddad, 2009).

Considering the multi-media differentiation between the industries and the more explored area of feature creep in software development gives an opportunity to research an industry that shares many similarities. Therefore asking,

- (RQ1) how feature creep is perceived by game developers,

is relevant to understand how negative aspects of feature creep could be reduced in an industry driven by multimedia developers. The aim is to understand feature creeping through the individual game developer experiences in their work and what effect it has on them and the projects they work or worked on.

This study also explores current project management methods in order to see how project management handles feature creeping, lifting the question of,

- (RQ2) how project management strategies for feature creep prevention looks like.

Therefore a study about how developers look at feature creeping would contribute to understanding the opportunities and pitfalls with the feature creeping phenomenon and how it can be used in production to enhance quality and innovation in games. Further this study aims to explore,

- (RQ3) how feature creep affects the game developers' engagement?

# Method

To answer the research question of how feature creeping is perceived by game developers the study utilized in-depth interviews (Cote and Raz, 2015) for data gathering and analyzed the empirical data using the thematic analysis-method (Braun and Clarke, 2006). Individual and semi-structured interviews allow the participant to reflect thoroughly about their answers and provide valuable answers to the study. This format allows us to maintain control over the conversation, ensuring the participant feels involved in the research as well as induces numerous views and answers (Cote and Raz, 2006). In addition to RQ 1-3, this study will also look at (1) if developers are aware of feature creep. As employee engagement is one of the main reasons the phenomenon occurs according to Hagen (2011). Therefore it is important to understand whether the developer is aware of it. If (2) the developer is aware that feature creeping occurs, is there a way to utilize the phenomenon in the development process? Since Lee and Lee's (2015) study of feature creep presented both positive and negative effects of feature creeping there is a possibility that feature creeping could be maintained in a controlled manner. It is therefore important to settle (3) whether the perception of feature creeping is consistent among game developers to ensure the phenomenon is understood and handled in everyone's favor since feature creeping breaks scope and deadlines in projects (Kanode and Haddad, 2009).

The study also allows for email interviews for those who prefer it. Known limitations stated by Burns (2010) involve issues as (1) the target group may include participants with limited literacy or lack of internet, (2) the potential for participation drop out is larger than live interviews. (3) The interviewer has to develop a research etiquette to engage the participant over multiple rounds of conversation. (4) Finally, it has been a concern that responses may be faked, unlawful, or less socially acceptable.

## Recruitment

Participation invites were sent to 99 game companies all over the world. Where 10 declined, 74 did not respond, 15 showed interest and nine interviews (one live chat, one email, and seven video interviews) were completed. The participants were employees with at least one year of experience within the game industry and have been part of the design, asset creation, implementation, or planning of a game project from the invited companies.

The email after the first contact followed Cote and Raz's (2006) template starting with overviewing the purpose of the study, as well as ensuring confidentiality towards the participants; Ensuring they are aware that their name, project info, other confidential or personal information will not be included or be anonymized if a statement is relevant to the study as well as informing that the interview may be stopped at any point. The participants were offered a copy of the study in its final state as an appreciation for participation.

## **Data Gathering**

The study consisted of 9 participants, with a variety of different roles including director, gameplay programmer, head of design, producer, and sound designer. The interviews were primarily held as video interviews, text-based interviews were given for those who preferred it, due to the risk of Covid-19 as well as convenience for both parties. The interviews start with a few warm-up questions to create a relation, build rapport and release tension with the participant. Followed by more substantive questions which involve open-ended questions provoking in-depth answers and allow the interview to be more fluent with follow-up questions. Finally, the interview includes questions about demographics to characterize and ensure the participant is in a position to provide meaningful empirical data, heavily based on field experience. Email interviews require multiple rounds of exchange of conversation allowing iteration and follow up questions.

The length of the video interviews varied between 24 to 55 minutes for a total of 266 minutes. The length of the email and chat interviews matched the transcribed versions of the video material.

The questions used in the interview are based upon our research questions as well as what was found in recent research; Where the intent of the question is to open up the discussion on how each of these themes are used in the real world. A sample of substantive questions used in the interview to cover these topics are:

- How are new features implemented in a project?
- How is the project vision shared within the company?
- Have any of your projects been affected by feature creeping? If so, in what way?

Demographic questions were included to establish the validity of the participant and their background as reliable sources. Even so, questions covering their personal experiences with game development.

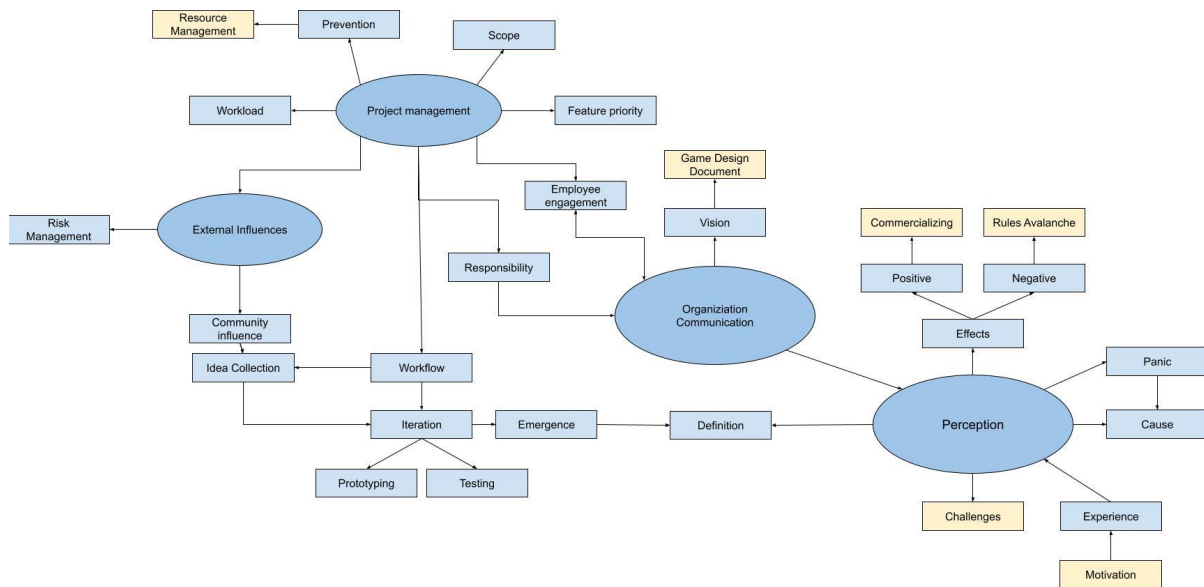
Email interviews were separately compiled into text documents with fitting format. Video interviews conducted have been transcribed into text documents using the same format. The interviews were conducted in Swedish and English. Quotes in Swedish have been translated and the original quotes can be found in the appendix.

## **Data Analysis**

The data analysis was made through an inductive thematic analysis (Braun and Clarke, 2006) where no pre-planned themes were set to make connections with the data. The themes and aspects presented are the product of the analysis of the results through a series of thematic steps. (1) the interviews were transcribed and initial thoughts were noted as an overview. (2) The transcribed interviews allowed coding quotes and grouping them together in a sheet to easily overview the codes. (3) The themes were formed through the grouping of the codes by finding common groups of codes that relate to each other. (4) Once the themes were defined initial conclusions were noted and a thematic map was created to visualize the connections between the codes and the themes. (5) finally the report was produced by presenting and explaining the data, analysing, connecting and discussing the data as well as comparing it to related research.

# Results and Analysis

The data gathered was analysed through an inductive thematic analysis (Braun and Clarke, 2006). The data was coded and divided into four themes: perception, project management, communication, and external influences, visualized in a thematic map (see Fig.1).



(Fig.1. Thematic map.)

To describe the four themes found through the thematic analysis (1) *Perception* revolves around common personal experience with feature creeping from the developer perspective involving thoughts about the phenomenon, personal definition, as well as causes for feature creep occurrence. (2) *Project Management* looks into how feature creeping is perceived from a project management point of view, how it affects resource management, scope, workload and how bad aspects of feature creeping can be prevented through management. (3) *Organization Communication* is about how communication internally within the company impacts development, how the vision is shared within the development team, how feature creep is brought up during development. Communication has also shown to be an important role for employee engagement during development. (4) *External Influences* such as consumers, publishers, competitors or new technologies as a theme showed the impact of how these sources affected the management and perception of feature creep in development.

## Perception

The participants showed a perceived consensus that feature creep entails re-designing and iterating on already established elements adding complexity and a growing project scope due to expectations or overambition.

*“It is when a developer goes outside the scope of what you have like planned and discussed what a new feature should be for example. That you take the matter in your own hands and move forward with something and make it broader and bigger then [it] initially was going to be. I would say that feature creep really is when you do something and it affects several in the team.” - P2*  
(Quote.1, Translated, see original in appendix)

*“I would say it’s a phenomenon where people start to re-design stuff that is already, kind of set but when there starts to be a little bit of [a] panic mode and [when] the project is closer to [the] end, it’s common to start thinking like ‘ah, this is not good enough, this is not good enough, we need to re-do this, that and that’, and then also start to come up with a lot of new ideas suddenly.” - P3*  
(Quote.2)

*“Simply adding unplanned features to a game specifically at the detriment of other aspects, like schedule or work time)” - P5 (Quote.3)*

*“Feature creep is kind of when you have a feature designed in mind and say ‘we want to implement this particular thing’ and then because of miscommunication or over ambition from one or the other, then it leads to an exception that there is more than have been specified to be added basically.” - P6 (Quote.4, Translated, see original in appendix)*

*“...for me feature creep is more about when the scope of the game grows beyond what is already planned.” - P8 (Quote.5, Translated, see original in appendix)*

These perceived definitions of feature creep follow along the lines of related research (Lee and Lee, 2015; Lee, Woods, and Kidwell, 2006; Norman, 1998; Thompson and Norton, 2011) suggesting feature creep as a phenomenon adding or emphasizing the current features of a product. One participant (P3) perceived feature creep as a form of emergence which is similar to the definition but on an abstract level not pointing towards a specific feature but an



idea that feature creeping is part of the iteration process and is not something that can explicitly be specified as part of one feature.

The participants brought up multiple views on the cause of feature creep. Some express it as misinterpretation of tasks, others say it is the stress towards the end of a production cycle or that it originates from external forces pushing ideas on the developers to implement.

There can also be that technical limitations require unplanned implementations of features to solve a problem, inexperience in the field, and usertesting can surface unexpected issues requiring new features passionate developers want to improve the game with new ideas. This has been concluded by P1, but also expressed by the other participants.

*“It is a big and difficult subject, but here are some thoughts. There are many variants of feature creep I would say. What comes first to mind is when you add new features to improve the game, someone has a vision that grows or similar. It seems more common in less experienced teams. The other is more about how a certain existing mechanic seems to not work, maybe it’s not turning out to be fun, maybe it takes too much time to develop, maybe an unforeseen technical limitation. Then it is more about having to add new features in order to save existing features, or completely replace something you thought would be good.”* - P1 (Quote.6, Translated, see original in appendix)

The participants' answers point towards the common idea of feature creeping by the literature and the name itself that the phenomenon creeps up on the team. In the majority (8 out of 9) of the cases feature creeping was perceived as a negative phenomenon but over time 7 out of 9 described feature creep with a more positive tone.

*“I would say that feature creep really is when you do something and it affects several in the team without... you can add stuff without it being feature creep, but only if it only affects you and your own time that you feel ‘I can do this and finish it’*

[...]

*When you first say the words feature creep you think ‘yeah, it’s something negative’ but to fully stop feature creep is kind of like stopping... artistry too.”* - P2 (Quote.7, Translated, see original in appendix)

*“Mhm... that’s hard... I think that feature creep like that specific term isn’t something that I have a clear relation to... I usually focus more on ‘scope creep’ I would like to call it*

*[...]*

*Yeah, I would say that... it definitely happens... I think we are getting better and better at it*

*[...]*

*so I think you have a point in that, that it [the project] is so small and agile so it is not really, like that feature creep that becomes something positive is sort of accounted as part of the process already.” - P8 (Quote.8, Translated, see original in appendix )*

Where the one interview standing out had a positive tone to feature creep throughout the entire interview. Where it was seen as an essential part of game development. Which goes along with Kanode and Haddad’s (2009) conclusion where iteration and unplanned features are inescapable in game development to find the “fun”.

*“There he [Salen and Zimmerman, 2003] talks about emergence and games emergent nature, and that would be what I think of when I think of feature creeping*

*[...]*

*[fun] is nothing you know because sometimes the most certain could be... this do people always think is fun. And in some game context it becomes boring or broken. Something like that is what I think about emergence” - P4 (Quote.9, Translated, see original in appendix)*

The participants in the study had a scattered view on what engages them in a project. The views touched on themes as educational, problem solving, financial, product appreciation, progression, consumer appreciation, professionalism and games cultural impact. Where progression was the only aspect that could be related to this study. Progression is about the iterative process and how the emergence of features on a day to day basis can be a motivating factor for employee engagement. The remaining aspects are outside of the scope of this study.

*“I’m a very result driven person. So on a day-to-day basis it’s just to see new things, that things are happening, that motivates me very much.” - P8*

(Quote.10, Translated, see original in appendix)

## **Project Management**

This study has seen cases of both ASD and WSD methodologies for development. Majority of participants show tendencies to start projects with WSD in terms of idea collection, requirements, design meetings and documentation of what and sometimes how an element will be implemented.

*“There are of course like some iteration going on and we have [a] very like... flat hierarchy and those kind of like agile stuff, but I still see game development as pretty much waterfall.” - P3 (Quote.11)*

This then changes during development to ASD methods being adopted in day-to-day development where the flexibility of faster implementation that leads to more testing, evaluation and iteration of features is appreciated. The ASD method also helps in the instances where, for some reason, feature priorities change and focus is needed elsewhere.

*“that is why scrum has become so generally accepted or used within games because there you have this very good balance between that there is some sort of vision statement and then there is a very flexible process how you approach the development of it.” - P4 (Quote.12, Translated, see original in appendix)*

The reason for the mix of development methodologies is because of the involvement of producers, investors and publishers who want to know where the budget is going and that there is a plan for development to ensure success. Two participants (P2 and P9) had similar thoughts about how WSD can be a more suitable development method while working with these external parties to be able to deliver a promised product.

*“...and I can imagine that there will be conflicts with an agile approach when you must hand to a publisher for example exactly what it is you are doing. I think it collides with using an agile... but we try to make something fun here and like that but we will give you money then I want to know exactly what you are going to do and it is a lot harder to do that in an agile process than [a] waterfall [process].” - P2 (Quote.13, Translated, see original in appendix)*

*“We found that with work-for-hire... waterfall - we dabbled in agile practices - tended to work best because publishers wanted to be very specific about what they were receiving so we could be paid, but I don’t think that there’s any one methodology that like, I think you’ve got to find what’s right for you.” - P9*

(Quote.14)

Estimating the size of a task is a common cause of feature creeping where the developer overestimates their ability to perform a task and underestimates the time it takes to do the given task. As discussed by Austin (2001) developers are likely to cut deadlines to prevent the feeling of being less talented than their coworkers. Which explains the perceived under- and overestimation where the developer wants to prove their talent by hitting deadlines of ambitious tasks. To accommodate the poor sense of estimation by developers Austin (2001) suggests that planning and deadlines should be set separately limiting the comfort zone of having plenty of time as well as allowing a loose end goal.

*“We tend to either overestimate and underestimate all the time. and by that we overestimate our own ability and we underestimate how much things take to build.” - P7 (Quote.15)*

The participants brought up suggestions that feature creep is difficult to spot and manage early on, spiraling into what has been described as panic or crunch in the interviews. This confirms Lee, Woods and Kidwell’s (2006) mentioning how difficult it is to detect feature creep due to the small iterations.

*“because first of all you need to identify proper feature creeping to be able to prevent it. the reason I am saying it is difficult to prevent it.” - P7 (Quote.16)*

However P8 brings up an interesting point about the small iterations throughout development talking about how game development demand iteration and the perceived view is not that adding new features is the problem but not planning for iteration time is what causes negative effects of feature creep.

*“the question is kind of, where.. where does planned iteration time become feature creep? Because I think, those times things we’ve done have been delayed, that hasn’t been a result of us wanting to add more features rather than a result of us underestimating how much time we needed to iterate on a feature that was already planned.” - P8 (Quote.17, Translated, see original in appendix)*

By planning iteration the development process opens up for emergence which is a term used to describe complex patterns that occur through interplay between simple structures. In the case of game development fun is a complex pattern that can be identified through testing iterations of features and game mechanics. Emergence is therefore a necessary evil in game development where the developers are aware of the fact that these iterations take time but are necessary to find the “fun” in the game. Participants expressed an uncertainty between feature creeping and planned iteration time where it has been described as a feature creep party by one of the participants.

*“You have an idea. But you try to pick it apart and try to find where the idea becomes fun. Like you say where it becomes fun. It becomes a big feature creep party.”* - P2 (Quote.18, Translated, see original in appendix)

*“feature creeping almost becomes equated to that the game is not what you thought it would be from the beginning. And that the game... if you now think games... I don't think that it's possible... I think it's almost impossible to think from the start like 'now I think the game is over here... this is how it will look'.”* - P4 (Quote.19, Translated, see original in appendix)

The fun component in game development is what makes game development different from traditional software development where functionality in many cases is not the priority but an abstract concept of fun.

*“The problem that we face that is different from a lot of other mediums is that we rely heavily on iteration. and unlike other software that also has iteration, we at the heart of it is the fun right?”* - P7 (Quote.20)

To find the “fun” in game development it is required to go through multiple iterations and prototypes. Apart from the software engineering challenges presented in game development other divisions from the development team will participate through internal influences (Kanode and Haddad, 2009). An example of feature creeping from the sound department occurred during one of the interviews where it ended up affecting the game budget.

*“That we want to create the most authentic game as possible so a lot of the audio we record is live audio from actually blowing some stuff up or stuff like that. so like one of the “stuff” that they wanted to record is that they wanted to... so when you're playing the game those are actual zero [airplane] sounds. That was [an] incredible, amazing amount of passion. very expensive.*

*[Laughter] that's one of those things where we go "...We can do that or we can [do] something else." but that wouldn't fit the game." - P7 (Quote.21)*

This example could also indicate that the company brand and mission statement has an important role in prioritizing features and that it is hard to calculate the cost versus gain from a single decision. What can affect the budget negatively but in the end return as newsworthy and allow the game to be successful.

One aspect that the study did not foresee was the rise of another concept as some participants expressed the term "scope creep". The notion here is that, surely one single feature could go out of hand and stray away from original designs, but that the scope of the project due to the addition of features, tools or coming with solutions to problems that do not need a solution.

*"like when we plan our projects we leave a lot of room, a lot of space in the schedule. for iteration. Ehm... for me feature creep it is more about... when the scope of the game grows outside of what is already planned" - P8 (Quote.22, Translated, see original in appendix)*

When analysing the data gathered it appears as if the perception of feature creep has more things in common with what would be scope creep where feature complexity and usability was not perceived as an effect of feature creep but focused on schedules and budget.

## **Communication**

Communication is one of the most essential aspects of game development. Everybody has to know what is going on and get in the same mindset in order for everyone to have the same vision. Having a level of communication is easier in smaller projects and teams, as projects and teams expand, the maintaining of communication and the clarity of it is one of the biggest challenges, expressed by one participant.

*"but if I was to summarize the main things that don't work is that maintaining communication and clarity of communication is probably one of the key challenges that we face. and the bigger scale the project [is] the harder it is to maintain that level of communication." - P7 (Quote.23)*

An unveiled perception through this study is how communication has evolved over time from text based means of communication such as GDD's to more visual methods. As mentioned in the literature by Hagen (2011) examples of simple and concise descriptions such as the

‘elevator pitch’ or another alternative that surfaced during the interviews ‘parking lot pitch’. Prototyping seems to be the most common way of both communicating but also testing game ideas and is the most common way of communicating the design vision. It came as a surprise that moodboards and concept art were not common methods which were common in Hagen’s study along with music, moving images. However these methods may not have surfaced due to limited focus on vision sharing during the interviews.

It is possible to over-communicate where describing each and every aspect of a game can take away precious time from development through meetings and reading documentation. In contrast to over-communication the design vision can be under communicated leaving more time for development and increases productivity. However, under-communicating the design vision can lead to features having to be removed or reworked. It is perceived by the participants as a tug of war between over- and under-communication and that it is depending on the size of the studio.

As of the Covid-19 outbreak the ways of communication were different as all interviewed studios were working from home. This change of workspace has forced the online meeting environment onto employees. The positive and negative effects are inline with Lippe and Lippényi (2020) research where their extensive research concluded that productivity was increased on an individual level due to lack of downtown moving between meetings, however manager positions seemed to struggle more to follow the development due to lack of social aspects that is accomplished outside scheduled meetings. By isolating social aspects to professional official meetings a lot of the spontaneous ideas disappear as coworkers can not easily discuss new findings and ideas. However P6 expressed that working from home experienced less problems regarding feature creep as routines looked different on-site versus from home.

*“I honestly think that it has been less feature creeping now that we’ve been home but I also think it’s because that we’ve established better routines for it now too...”* - P6 (Quote.24, Translated, see original in appendix)

Working from home brings up communication difficulties as face to face interactions are not as common making the developer more isolated from the project. By removing the face to face interaction one participant suggested miscommunication as a cause if feature creep where the phenomenon is not an active decision from any part of the team to include a feature

in a certain way, but miscommunication steering the direction of a feature that in turn would affect workload, deadlines, or scope.

*“So I would want to claim that it means that there is a miscommunication somehow between the person who designed something and the one whos implementing something, in what is expected to include.” - P6 (Quote.25, Translated, see original in appendix)*

Another aspect also worth mentioning is how the engagement among the developers relate to feature creep. Comparing Verčič and Vokić (2017) study where only 13% of the employees were engaged in their workplace this study found that 88% (8 out of 9) of the participants in this study’s sample felt engaged in their workplace. As mentioned by Lee and Lee (2015) the interviews in this study further show signs that developers who are passionate and driven tend to come up with more ideas and increase the complexity and feature list of a given project. This study shows indication that developers who are passionate are prone to be blind to their own contribution to feature creep supported by Hagens’ (2011) study. Developers in the game industry are different from many other industries in the way that employees within game development are passionate about games on both a personal level and professionally, expressed by P7.

*“When you're that close to what you're doing it opens up the room for two things. first of all you might be blind...you have blind spots...and you might be blind to your own passion making you do things you'd regret. like feature creeping.” - P7 (Quote.26)*

*“we are a very passion driven industry so like everybody that works on games really, really like games. ” - P7 (Quote.27)*

## **External Influence**

The causes for feature creep can be divided into two types; internal and external feature creeping. The internal type suggests that the project emerges from within the development team. While the external type is influenced by publishers, community, hardware restrictions, or the current market.

*“Then sometimes it happens that things outside the projects itself change, new [platforms], technologies and trends show up that makes you feel like you are*



*forced to change the game, which can often make you replace or add features.”*

- P1 (Quote.28, Translated, see original in appendix)

*“on one project we actually had a system in place in which we used a mediation company between the developer and publisher, and agreed in contract that any changes had to go through them which everyone was really pleased with, but as I remember it the publisher just tried to ignore it and push changes on us anyway.”* - P5 (Quote.29)

This type of event as P5 told can sometimes be detrimental for projects and is unfortunately a reality where games never see the light of day.

*“I’ve seen this.. publisher asking for big changes.. so more time being added (though not enough) - then publisher cancelling [the] project.”* - P5 (Quote.30)

Although outside sources can disrupt internal workflows or morale, they can also be beneficial in some regards. A few of the participants that work at independent game studios, where they themselves decide what projects to make and how without e.g. publisher interference use the community they have when deciding new projects. They do this in different ways, either by collecting ideas sent to them or by prototyping in-house concepts, sharing them and gathering the responses.

*“[...], the next step is that we’ve chosen an idea, is to verify it by developing a prototype of the core experience, like really focus on what is unique with this idea and then take 2,3,4 weeks to implement it.”* - P4 (Quote.31, Translated, see original in appendix)

This seems to be the favorable approach for smaller game studios. Whether a game is intended to follow the GaaS or GaaP model, the game must be tested to make sure it is wanted and establish potential buyers before greenlighting a project. Which P1 also thinks is an important step to make sure it is possible.

*“It’s very different depending on the project, but often you implement some very rough prototype version of the functionalities to assess if the idea is practically good.”* - P1 (Quote.32, Translated, see original in appendix)

In both cases P8 and P1 show that small studios might have limited resources, so the profitability and interest is therefore tested beforehand. If it is not a makeable prototype, then it will not be profitable to develop, according to P8 and P1.

External influence is perceived as a cause for feature creep as external parties expect certain features making the scope bigger than initially planned and developers must be able to adapt quickly to accommodate those ideas and suggestions. Similar to what Lee and Lee (2015) found in students' work that the expectations from the teacher determined if a student would produce on a quantity or a quality level. From the interviews conducted it was commonly acknowledged that the feature priority was heavily dependent on the external influences expectations and that negotiations over features or functionality was in the external influences advantage.

## Discussion

The study conducted interviews with game professionals and sought out to further understand the feature creep phenomenon in the game industry that previous research is more found in traditional software development and other industries. Although game development share a lot of software engineering principles and can even be more complex due to the multi-media handling which includes e.g. 3d rendering, animation, sound, code and more, into a format that expressed by P7 (see Quote.20), in its essence also has to be fun (Kanode and Haddad, 2009), have been somewhat neglected as if there is a notion that game developers would not experience it the same way.

The participants' perception of feature creeping showed a general consensus of the phenomenon that it is when the established elements undergo an unplanned design stage again. This is manifested either with discussions of ideas to add with the coworkers or that the developer took the matter themselves and implemented it how they saw fit. The former one is the more common to unfold, as the latter one speaks more towards lack of experience (see Quote.6 and Quote.33)

*“Otherwise I think it’s more common for more junior employees to do the feature creeping so more experienced ones have like have seen enough many game development cycles...”. - P3 (Quote.33)*

By communicating feature creeping amongst coworkers it could be possible to get a shared vision and fully grasp the positive and negative effects. This in turn would lead to conversation starters in the workplace and build upon the friendly environment as well as providing new ideas and solutions to a day to day problem by having a shared understanding on why feature creeping could have positive effects on the project as well.

There are multiple causes for feature creep including the number of team members, project management strategies and team involvement to name a few. What can be seen is that clear communication is essential to minimize risks (Blomkvist, Persson, and Åberg, 2015). This study has seen that game studios with fewer employees have it easier to involve everyone in the process, therefore letting the developers share their ideas and opinions. By the time the design is completed, everybody’s thoughts have been considered and it can be accepted that it is the conclusion of all ideas.

Due to the nature of games' emergence nature and unknowing course of direction as we have seen in this study it makes sense to see an ASD method being used to accommodate for the free artistry required in this entertainment medium. However, parts of the workflow are still close to WSD as there are plenty of useful patterns in the different fields of development that can be pre-planned as mentioned by one of the participants (P7, see Quote.34 in appendix). As discussed by Petrillo and Pimenta (2010) the informal nature of ASD methods such as Scrum and extreme programming can easily be implemented and combined with traditional WSD methods to maintain structure and agility in the development. It is therefore not easy to say that a company is strictly using ASD or WSD methods to develop games in the modern industry. What can be said is that the transitioning from WSD to ASD has allowed the good parts of feature creeping to emerge and from the more experienced participants in this study it is suggested that the ASD process is more engaging than a WSD approach where it is perceived as more limited environment where the developer may not have much input in the direction of the project.

A mix between WSD and ASD methods has surfaced in the industry and is a key component to bring out the good parts of feature creeping such as finding the 'fun', employee engagement, and brand value through complexity. The mixed methods also limit bad aspects being missed deadlines, poor usability, and going over budget. The small iterations put the developer in a good set of pressure where the developer must complete a small task within an acceptable time frame (Austin, 2001). Where suggestions and ideas can be added to the backlog at any time due the fast-paced nature of game trends (Vaudour and Heinze, 2020) and management roles regularly groom the backlog to fit the scope and budget of the project allowing new and old ideas to be included.

Participants showed how GaaS is a model that can take away pressure from the developers as all features do not have to be implemented by launch. The flexibility in the deadlines allow the developer to produce usability and user centered features through user testing over time which is a sign that the game may increase its value as Austin (2001) discussed how companies who miss deadlines often end up with higher quality products. Usability was not on the other hand explicitly expressed by the majority of the participants. Participants mentioned aspects of the importance of the consumer feedback which indicates the importance of usability as games in its essence have to be fun and easy to use.

GaaS is not used as a method to prevent feature creep but rather the opposite; allowing it to be a bigger part of development where missed deadlines in other methods may seem like sin. In the attempt to create and maintain relationships with the consumer and having an indefinite timeframe for completion improvements and additional features only raise the value of the game as stated by Dubois and Weststar (2021) on launch of a game using the GaaS model it is only the beginning of the journey while in the GaaP model it is the end of the production cycle. GaaS is a way to extract positive aspects from feature creep to allow artistry and find innovative ideas while also implementing and monetizing a service to maintain a successful business. The freedom of ASD used in GaaS methods attracts passionate and creative individuals allowing each developer to participate and have an impact on the game. Participants in this study are aware that feature creep occurs but due to the ASD methods used feature creep is not as threatening as discussed in related research where regular discussions within the organization gather ideas and suggestions for later in the development cycle which is in a GaaS project indefinite. However, participants who have experience with GaaP projects claimed that feature creep may have been the cause for cancelled or delayed projects (see Quote.30).

# Conclusion

The results of this study found signs that the perception of feature creeping at its core is similar among game developers. As a phenomenon that is not easy to control and that there is no one known way to handle the phenomenon and only extract the good effects that comes with it. Feature creeping is a phenomenon that is perceived as an inescapable phenomenon for game development to find the fun component during development which has worked its way into the workflow in the ASD methods used in game development as planned iteration time. Good features emerge from iteration and communication thus making feature creeping a necessary phenomenon with side effects such as long working hours, unhealthy schedules, absence from social circles and in some cases financial struggles. Employee engagement showed no obvious correlation to feature creep in this study when participants were asked, a variety of ideas on what engaged them in projects were given. Where only progression; mentioned by one participant could be seen to have a relation to feature creep where seeing new ideas being implemented on a daily basis was a driving factor.

Future research may also want to investigate how the negative aspects of the feature creep phenomenon can be approached to maintain a positive day-to-day experience for the individual employee while maintaining efficiency in the development process. Feature creep is today initially perceived as a negative phenomenon and this study suggests that if communicated and approached appropriately feature creeping can lead to innovative solutions and it seems that the ASD methods increased popularity is showing a trend where the agility of a production is positive in both the end product and employee engagement. However, it opens up the question whether a high employee engagement in a project can be harmful and would therefore motivate further studies how development methods approach the negative unhealthy and moral expectancy of having employees risking their physical and mental health for the cause of an unachievable perfection in game development.

## References

- Austin, R.D., 2001. The Effects of Time Pressure on Quality in Software Development: An Agency Model. *Information Systems Research*, 12(2), pp.195–207.  
<https://doi.org/10.1287/isre.12.2.195.9699>.
- Berg Marklund, B., Engström, H., Hellkvist, M. and Backlund, P., 2019. What Empirically Based Research Tells Us About Game Development. *The Computer Games Journal*, 8(3–4), pp.179–198. <https://doi.org/10.1007/s40869-019-00085-1>.
- Blomkvist, J.K., Persson, J. and Åberg, J., 2015. Communication through Boundary Objects in Distributed Agile Teams. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*. [online] the 33rd Annual ACM Conference. Seoul, Republic of Korea: ACM Press.pp.1875–1884. <https://doi.org/10.1145/2702123.2702366>.
- Blow, J., 2004. Game Development: Harder Than You Think: Ten or twenty years ago it was all fun and games. Now it's blood, sweat, and code. *Queue*, 1(10), pp.28–37.  
<https://doi.org/10.1145/971564.971590>.
- Braun, V. and Clarke, V., 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), pp.77–101. <https://doi.org/10.1191/1478088706qp063oa>.
- Carpenter, G.S., Glazer, R. and Nakamoto, K., 1994. Meaningful Brands from Meaningless Differentiation: The Dependence on Irrelevant Attributes. *Journal of Marketing Research*, 31(3), p.339. <https://doi.org/10.2307/3152221>.
- Cote, A. and Raz, J.G., 2015. In-depth interviews for game research. In: P. Lankoski and S. Björk, eds. *Game research methods*. Pittsburgh: ETC Press, pp. 93–116.  
<http://press.etc.cmu.edu/index.php/product/game-research-methods/>.
- Dubois, L.-E. and Weststar, J., 2021. Games-as-a-service: Conflicted identities on the new front-line of video game development. *New Media & Society*, p.146144482199581.  
<https://doi.org/10.1177/1461444821995815>.
- Hagen, U., 2011. Designing for player experience: How professional game developers communicate design visions. *Journal of Gaming & Virtual Worlds*, 3(3), pp.259–275.

[https://doi.org/10.1386/jgvw.3.3.259\\_1](https://doi.org/10.1386/jgvw.3.3.259_1).

Kanode, C.M. and Haddad, H.M., 2009. Software Engineering Challenges in Game Development. In: *2009 Sixth International Conference on Information Technology: New Generations*. [online] 2009 Sixth International Conference on Information Technology: New Generations. Las Vegas, NV, USA: IEEE.pp.260–265. <https://doi.org/10.1109/ITNG.2009.74>.

Lee, B. and Lee, W., 2015. Feature Creep in Design Students' Works: Why and How it Happens in Student Design Processes. *The Design Journal*, 18(3), pp.345–365. <https://doi.org/10.1080/14606925.2015.1059603>.

Lee, D.-S., Woods, D.D. and Kidwell, D., 2006. Escape from Designers' Dilemma on Creeping Featurism. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50(24), pp.2562–2566. <https://doi.org/10.1177/154193120605002410>.

Lippe, T. and Lippényi, Z., 2020. Co-workers working from home and individual and team performance. *New Technology, Work and Employment*, 35(1), pp.60–79. <https://doi.org/10.1111/ntwe.12153>.

Nielsen, J., 1993. *Usability engineering*. San Francisco: Morgan Kaufmann.

Norman, D.A., 1998. *The Invisible Computer: Why Good Products Can Fail, the Personal Computer is so Complex, and Information Appliances Are the Solution*. Cambridge, Mass: MIT Press.

Nowlis, S.M. and Simonson, I., 1996. The Effect of New Product Features on Brand Choice. *Journal of Marketing Research*, 33(1), p.36. <https://doi.org/10.2307/3152011>.

Petrillo, F. and Pimenta, M., 2010. Is agility out there?: agile practices in game development. In: *Proceedings of the 28th ACM International Conference on Design of Communication - SIGDOC '10*. [online] the 28th ACM International Conference. São Carlos, São Paulo, Brazil: ACM Press.p.9. <https://doi.org/10.1145/1878450.1878453>.

Salen, K. and Zimmerman, E., 2003. *Rules of Play: Game Design Fundamentals*. Cambridge, Mass: MIT Press.

Thompson, D.V. and Norton, M.I., 2011. The Social Utility of Feature Creep. *Journal of Marketing Research*, 48(3), pp.555–565. <https://doi.org/10.1509/jmkr.48.3.555>.



Tkalac Verčić, A. and Pološki Vokić, N., 2017. Engaging employees through internal communication. *Public Relations Review*, 43(5), pp.885–893.  
<https://doi.org/10.1016/j.pubrev.2017.04.005>.

# Appendix

## Original quotes that has been translated

*“Det är när en utvecklare går utanför ramen av vad man har liksom planerat och diskuterat att en ny feature ska va till exempel. att man på ett eget bevåg tar och går vidare med någonting och gör det bredare och större än vad [det] initiiellt skulle vara. Jag skulle säga att feature creep är verkligen så när man gör någonting så och det påverkar flera i teamet...” - P2 (Quote.1)*

*“Feature creep de e väl liksom när du har feature designad i åtanke och säg ‘vi vill implementera den här specifika saken’ och sen på grund av misskommunikation eller på grund av överambition hos ena eller andra parten så leder det till en förväntning att det mer än det som specificerats som ska läggas in basically” - P6 (Quote.4)*

*“För mig är feature creep, handlar mer om.. om när scopet av spelet växer utanför det som är.. redan är planerat” - P8 (Quote.5)*

*“Det är ett stort och svårt ämne, men här är några tankar. Finns flera olika varianter av feature creep skulle jag säga. Det man kanske först tänker på är att man lägger till nya features för att man vill förbättra spelet, nån har en vision som växer eller liknande. Den verkar vanligare i mindre erfarna team. Den andra handlar mer om att någon existerande mekanik visar sig inte fungera, kanske inte blir roligt, kanske visar sig ta för mycket tid att utveckla, kanske någon oförutsedd teknisk begränsning. Då handlar det mer om att man måste lägga in nya features för att kunna rädda existerande features, eller kanske helt ersätta nått som man trodde skulle blir bra” - P1 (Quote.6)*

*“Jag skulle säga att feature creep är verkligen så när man gör någonting så och det påverkar flera i teamet utan att... man kan ju lägga på saker utan att det är feature creep. men att det bara påverkar dig själv och din egen tid att du känner ‘jag kan göra det här och det blir färdigt’[...]*

*först när man säger ordet feature creep då tänker man ‘ja det är någonting negativt’ men att helt hindra feature creep är ju lite som att hindra... artisteri också.” - P2 (Quote.7)*

*“Mm [paus] den e svår [paus] jag tror att feature creep som asså just specifikt den termen är inte nån’ting som jag har en, ett tydligt förhållningssätt, alltså.. jag brukar mer fokusera på ‘scope creep’ vill jag kalla det istället [...]*

*[...] Aa, det skulle jag säga [paus] det händer definitivt [paus] jag tror vi blir bättre och bättre på det [...]*

*[...] så jag tror du har en poäng i det att det är så litet och agilt så det inte riktigt, alltså att den där feature creep som blir nåt positivt är liksom inräknat som en del av processen redan.” - P8 (Quote.8)*

*“Där pratar han [Salen and Zimmerman, 2003] om emergens och spelens emergenta natur. och det är väl det jag tänker på när man tänker på feature creeping [...]*

*[...] det är ingenting som man vet för ibland kan det mest säkra...det här det här tycker folk alltid är kul. och så i något spelsammanhang så blir det tråkigt eller brutet så. någonting sån't tänker jag iallafall om emergens.” - P4*

*(Quote.9)*

*“Asså jag är en väldigt resultatdriven person. Så på en sån här ‘day-to-day’ så är det bara åh se nya grejer, att jag ser att det händer saker, det motiverar mig, att bara att spelet rör sig, asså att, nåt är annorlunda från igår eh motiverar mig väldigt mycket.” - P8 (Quote.10)*

*“det är därför som scrum har blivit så allmänt vedertaget eller använt inom spel därför att där har man den här väldigt bra balans mellan att det finns någon sorts vision statement och sen finns det en väldigt flexibel process för hur man angriper utveckling av det här.” - P4 (Quote.12)*

*“...och jag kan tänka mig att det blir konflikter med en agil approach när du måste lämna till en utgivare till exempel exakt vad det är du ska göra. Där tror jag att det krockar mer med att ha ett agilt...men vi försöker skapa någonting kul här och sådär men vi ska ge dig pengar då vill jag veta exakt vad ni ska göra för någonting och det är mycket svårare att göra det i en agil process mot [en] vattenfalls [process].” - P2 (Quote.13)*

*“För frågan är liksom, vart.. vart går planerad iterations tid in över till feature creep. För jag tror att dom gångerna som saker vi har gjort blivit försenade, det har inte vart ett resultat av att vi vill lägga till fler features utan snarare ett resultat av att vi har underskattat hur mycket tid vi behövde för å iterera på en feature som var planerad” - P8 (Quote.17)*

*“Du har en idé. men du försöker plocka isär den och försöker hitta var den här idéen, var någonstans den här idéen blir kul. som ni säger var den blir rolig. det är ju bara en stor feature creep fest.” - P2 (Quote.18)*

*“feature creeping blir nästan likställt med att spelet inte är vad man tänkt sig från början. och att spelet...om man nu tänker sig att spel... Jag tror inte att det går... Jag tror det är nästan omöjligt att tänka från början såhär. ‘Nu tänker jag att spelet här borta...så här kommer det se ut tänker jag’.” - P4 (Quote.19)*

*“asså när vi planerar upp våra projekt så lämnar vi mycket utrymme i, mycket luft i schemat för tid, för iteration. Ehm... och för mig är feature creep, handlar mer om... om när scopet av spelet växer utanför det som är.. redan är planerat för...” - P8 (Quote.22)*

*“Asså jag tror helt ärligt att det mindre feature creeping nu när vi varit hemma men jag tror också att det är på grund av vi har satt upp bättre rutiner för det nu också...” - P6 (Quote.24)*

*“så skulle jag vilja påstå att det innebär att finns en misskommunikation på nåt vänster mellan den som har designat nåt och den som implementerar nån’ting, i vad som förväntas ska finnas med.” - P6 (Quote.25)*

*“Sen händer det ibland att saker utanför själva projektet förändras, nya [plattformar], tekniker och trender dyker upp som gör att man känner sig tvingad att ändra spelet, vilket ofta kan göra att man måste byta ut eller lägga till features.” - P1 (Quote.28)*

*“[...], nästa steg är att vi har valt en idé, är att verifiera den genom att utveckla en prototyp av core-upplevelsen, asså verkligen fokusera på vad är det som är unikt med den här idén och ut sen ta 2,3,4 veckor för att implementera det” - P8 (Quote.31)*

*“Väldigt olika beroende på projekt, men ofta att man implementerar nån väldigt grov prototypversion av funktioner för bedöma ifall idén blir bra i praktiken.” - P1 (Quote.32)*

*“and it carries it through and through. where's in like a larger AAA production you often find a variant in which like. we are agile in development so there's a lot of scrum happening. but at the same time there are big parts of our pipeline that just have to be waterfall.” - P7 (Quote.34)*

# Thematic map

