

Is Agile Not Agile Enough? A Study on How Agile is Applied and Misapplied in the Video Game Development Industry

Tim McKenzie, Miguel Morales-Trujillo, Stephan Lukosch and Simon Hoermann

University of Canterbury

Christchurch, New Zealand

timothy.mckenzie@pg.canterbury.ac.nz,

{miguel.morales, stephan.lukosch, simon.hoermann}@canterbury.ac.nz

Abstract—Video games have become the most dominant and successful entertainment industry worldwide; however, many video game development (VGD) projects and studios struggle to succeed. At present, there are no commonly accepted VGD best practices or frameworks that can bring together the complex and competing needs of software engineering and creative production. Although studios are reportedly using agile frameworks, the actual extent of application and effectiveness of agile practices in the VGD context is unclear. Therefore, the aim of this study is to empirically determine how and why agile frameworks are applied in VGD. Semi-structured interviews were conducted with eight New Zealand VGD studios. It was found that the agile frameworks, Scrum and Kanban, must often be adapted from their conventional use to meet the needs of different pipelines and delivery milestones within the phases of VGD. However, it seems that not all the needs of VGD can be met by current agile frameworks. Furthermore, inexperience with agile practices often leads to misunderstanding and misimplementing them in ways that seem to contribute to commonly experienced collaboration challenges.

Index Terms—Agile, Scrum, Kanban, Scrumban, Multidisciplinary Team Dynamics, Video Game, Game Development, Software Process Improvement

I. INTRODUCTION

In recent years, globally, video games have quickly become a massive and powerful creative industry that is far surpassing all other entertainment industries such as box office movies and music combined [1] [2] [3]. With a projected 2020 revenue of just under \$160 billion USD, an annual growth rate of 9.3%, and a player base numbering in the billions, the industry is showing no signs of slowing down [4].

But behind the video game development (VGD) industry’s seemingly enormous success, there are hard realities for many developers. At least a decade ago only 20% of all the games that went into production were finished, and only 20% of these that made it into the market turned a meaningful profit [5] [6]. Consequently, the majority of projects fail, as do many studios [7]. VGD is beset with many endemic challenges, particularly in multidisciplinary project leadership [8] [9], team dynamics, and collaboration [10] [11].

The situation in New Zealand (NZ) VGD is comparable with international trends. According to the 2019 NZ game

industry report, this industry made a turnover of \$138 million USD and is growing rapidly with a 39% rate, making it the “fastest growing tech sector and creative industry in NZ” [12]. Yet, according to a panel of industry leaders at the 2019 NZ *PlaybyPlay* conference, project leadership and team dynamics are major concerns for many of NZ studios [13].

These challenges were linked in previous research to VGD being a multidisciplinary ‘marriage’ between the traditional software engineering (SE), creative production, and product design domains [14] [15] – which complicates the management of such heterogeneous teams and the development pipeline [16].

Agile frameworks were introduced in part to address similar challenges in traditional SE, e.g. the Scrum framework is intended to manage complex software projects [17]. Indeed, although agile is designed for SE, game studios around the world are increasingly reported to be adopting agile development practices and frameworks [18] [19] [20] [21]. However, studios can have a false impression about how agile they really are [22], and this “puts into question whether ‘agile’ is actually an apt description of the development model that game projects employ” [23].

At present, there are neither commonly accepted best practices nor mature process models for VGD, which has led studios to use diverse, ‘fragmented’ [14], and improvised development practices which are often misattributed to being ‘agile’. These factors are thought to be contributing significantly to the industry’s multidisciplinary project management and team dynamics problems as well as to high project failure rates [10] [14] [16] [24].

The objective of this study, therefore, is to empirically investigate, through a series of interviews with VGD studios, how and why they use or adapt agile practices and frameworks, such as Scrum and Kanban.

This paper is organized as follows: Section II describes common VGD challenges. Section III explores the related research on VGD. Section IV outlines the study research objective and the semi-structured interview methodology. Section V documents the study’s results as categorized by five themes. Section VI discusses agile framework adaptation, perception,

and related challenges, as well as the study’s limitations. Finally, Section VII presents the conclusion and future work.

II. BACKGROUND

Developing video games is a complex task with the integration of assets and skills from many different fields including engineering, art, game design, music, user experience, film, marketing, data science, and narrative design [14]. This high level of complexity and diversity inherently accentuates challenges in project management and the development process.

For instance, there are issues classical to project management such as: difficulties in maintaining a clear project vision [8] [10], inadequate planning [8] [9] [16], task underestimation [8] [10] [24], scope management issues (e.g. feature creep or feature cutting) [9] [10] [16] [24], and poor scheduling (e.g. unrealistic or missed deadlines) [8] [9] [10] [16] [24].

The lack of structured, documented, or systematized industry development practice [10] [24], is also a concerning trend. Agile frameworks are neither well understood nor consistently adhered to in the industry [25], and consequently, VGD tends to be reactive, ad-hoc [8], or entirely developer-driven [25].

Maintaining effective multidisciplinary team dynamics for senior and junior artists, programmers, designers, and producers has been identified as another common VGD challenge. Team dynamics are a descriptor of how people work and interact together in teams [26], often relating to a team’s ability to be cohesive, share knowledge, and to collaborate and resolve conflict effectively [27].

VGD teams often have difficulties in cross-discipline communication [8] [10] [16] [24], and there are barriers to team cohesion and collaboration for developers with diverse skill sets, perspectives, priorities, and work styles [23]. There are also complexities ensuring work dependencies between developers do not delay the integration of multimedia assets in the production pipeline [16] [24]. Non-technical managers can make uninformed engineering decisions [25], which can frustrate the team. These problems often lead to tribalism, which is a source of potential team conflict and limits knowledge sharing [22]. This in turn may cause shortages in specialty skills and domain-specific knowledge [11] [25].

Many studios have poor or sometimes exploitative work cultures due to a lack of workforce diversity [28] [29] or due to an implicit or explicit expectation that staff work mandatory overtime (also known as ‘crunch’) [24] [30] [31]. Game developers commonly desire industry unionisation [29] [32], and a better work/life balance [11], as many experience mental health problems or burnout [28]. All these problems can result in high employee turnover which incurs the overhead cost of onboarding new team members [9], and so further exacerbates team dynamics difficulties.

Considering the grey literature, Politowski et al. [33], analyzed over 200 informal game industry project reports (also known as postmortems) to identify key VGD problems. The analysis showed that people rather than technologies were the root cause of most development challenges, and that reports of team dynamics problems were steadily increasing over the past

decade. Their summary of other related work in the field also concluded that management challenges are the single greatest problem in VGD, although the reporting of such challenges was in decline in recent times. Significantly, the authors note that “*one factor that helped decrease management problems might be the adoption of agile methods*” [33].

III. RELATED WORK

A literature review was carried out to identify primary studies in agile VGD and prior work through an organic backwards and forwards search of existing systematic literature reviews (SLR)s in the domain.

One of the more significant works on the field was a SLR conducted by Ruonala [34] to cover what was known on the topic up until 2016. Ruonala’s SLR identified 23 empirical studies and examined how widely and to what degree agile is used in VGD, what benefits agile offers, and what agile adoption challenges there may be.

From multiple studies the use of agile practices and frameworks such as Scrum in VGD were indeed widely reported [18] [19] [20] [21] [25] [35] [36], but the use of Kanban was barely mentioned in the literature [19]. However, the implementation of these frameworks varied considerably across studios, and even across development projects within a studio. For some projects, the entire development process can be agile whilst for others only some of the development phases are, and still others simply have an agile veneer over traditional power structures.

The myriad of inconsistent reports about the use of agile frameworks like Scrum lead Ruonala to question the objectivity of the perceptions of game studios on how ‘agile’ they really are, “*it is uncertain whether self-reporting on the development process is actually reliable. It is nevertheless clear that many companies have at least an interest in agile development*” [34].

Ruonala surmised that when agile practices have been applied properly, it does benefit multidisciplinary project management and team dynamics in VGD in two significant ways. First, an iterative and incremental development process of design, rapid feature prototyping, playtesting, and user experience feedback improves game quality and reduces the risks of feature creep and schedule overruns. Secondly, full adoption of Scrum, collocation, cross-functional teams, and hiring open-minded staff with good social and conflict resolution skills were common means to mitigate communication and accountability problems [34].

In addition, Ruonala discovered three challenges that must be overcome for agile adoption in VGD to be effective in fully realizing its reported benefits. First, all stakeholders must be fully committed to agile and respect the process. For instance, even though Scrum encourages self-management, artists and animators often resist any form of perceived control mechanism put in place over their work, including Scrum [36]. There can also be sudden and unexpected demands from external stakeholders such as publishers or parent companies, or from internal managers. These often result in hampered

communication, conflicting messages, disrupted work, interference with the team’s autonomy, and feature creep [34].

Secondly, it is also necessary for a change in power dynamics from hierarchical structures to flatter ones instead. Management must be willing to relinquish some of their responsibility and autonomy to the development team, and likewise, the development team needs to be willing to take on additional decision-making responsibilities. Without this, agile can in fact be harmful to development if the team cannot effectively handle the responsibility of self-management [34].

The third challenge is a common lack in agile experience or training, which a number of struggling studios and/or managers exhibited. Ruonala’s SLR notes that, in contrast, the successful companies “*had adopted agile across the company, and employees had received training in the method adopted*” [34].

Innovating on existing agile SE frameworks, such as Scrum, or creating entirely novel VGD-specific process models or frameworks to address the known challenges in the industry and reconcile the competing creative and technical tensions of VGD necessitates a deeper, more granular analysis of the existing industry development methods. This was the main conclusion of several successive SLRs, which all sought to identify and describe SE practices used in VGD, including O’Hagan et al. [37] in 2014, Aleem et al. [14] in 2016, Ruonala [34] in 2017, Engström et al. [15] in 2018, Engström [38] again in 2019, and Marklund et al. [23] in 2019. These papers encourage further empirically-based studies [15] [37] to identify areas where VGD can be improved [14], to determine what good practices can be adopted [14] or adapted [34], to further understand how agile is actually used in VGD [23] [34], and to understand how creative production and SE are managed from a dual perspective [34] [38].

Marklund et al. [23] also found that studios often misunderstand or mischaracterize their unstructured and unsystematic practices as ‘agile’, and that studios’ common belief on what VGD is doesn’t necessarily match what they actually do. Therefore, the authors stress the need for further research to clarify what VGD practices genuinely add value.

Also in 2019, McKenzie et al. [22] investigated the state of NZ VGD practice and juxtaposed the perceived level of adherence of 12 game studios to agile frameworks against their actual level of adherence. They found that all the game studios overestimated their actual level of adherence to Scrum and Kanban and many do not use some key Scrum practices (e.g. retrospectives) at all. They also noted that the very ways in which studios deviate away from Scrum may in fact be contributing to the team dynamics and project leadership challenges studios face. So, echoing Ruonala’s call, the authors concluded that more work needs to be done to understand why studios adapt or ignore certain agile practices to confirm this observation [22].

In 2020, Engström [39] performed an extensive review of the research on VGD to date. A key finding was that the challenge of project management and planning was much more intense in VGD than it is for SE due to VGD’s non-

functional requirements that are subjective to user experience (e.g. what the player finds ‘fun’). Consequently, the methods SE uses to address these problems cannot so readily be applied in VGD. Creating a shared vision and effective dynamic for multidisciplinary teams requires game producers to be proficient at both understanding and communicating technical and non-technical perspectives. How producers do this, is as yet unexplored [39].

In summary, the literature suggests that there is a connection between the complex and heterogeneous nature of VGD and the difficulty of managing such projects and their development teams. Agile practices and frameworks are intended to meet such requirements for flexibility, adaptability, complexity management, and high team collaboration [17] [40]. It raises the question then, as to why VGD studio practice often deviates away from the agile practices that ought to be eminently suitable for solving the challenges of their craft.

This study will help determine what are the actual or perceived limitations of ‘conventional’ agile practices in VGD, as well as how agile frameworks may be misunderstood or misapplied and so will contribute to known research gaps [15] [22] [23] [34] [37]. Secondly, the study will be a precursor for exploring a potential VGD-specific agile/creative production framework or set of best practices to address the common challenges experienced in VGD that many researchers have been calling for [10] [14] [16]. Finally, the findings will also equip tertiary education providers to improve how they teach agile VGD to address the knowledge and experience gaps that are known to be in the industry [11] [25].

IV. METHODOLOGY

A. Research Objective

The purpose of this study is to achieve a better understanding of how and why SE agile practices and frameworks are used and/or adapted to meet the unique needs of the VGD process. This is done by producing a more detailed view of agile VGD practices itself that has been empirically derived from semi-structured interviews with NZ VGD studios. Hence, this motivated the following research question (RQ):

What are the reasons for how and why studios adhere (or do not adhere) to agile frameworks such as Scrum or Kanban in their own development context?

B. Interviews Design and Validation

Interviews were chosen as the primary research instrument because they seemed to be the most expedient empirical means to collect qualitative data on studios’ perceived use and adaptation of agile practices. The unit of analysis was one senior game developer in a leadership position per studio. The interviews were semi-structured with six initial questions to determine the game project’s context and chosen framework, then three questions identifying the use and adaptation of a studio’s chosen development framework. The interview questions were as follows.

Project Context:

- 1) How many full-time development staff did you employ on your most representative project?
- 2) Was your most representative game an internal or external project?
- 3) What development methodology you used on your most representative project?
- 4) Is this the normal methodology you use when your studio works on game projects?
- 5) Do you use this same methodology during pre-production, production, and post-production on the same project?
- 6) Why is that?

Development Framework Adaptation:

- 1) For what reasons did you choose this methodology you use over other ones?
- 2) How have you adapted this methodology for the context of your most representative project?
- 3) What were the reasons for doing so?

The participant was subsequently given lists describing the essential elements of the three agile frameworks reportedly used by NZ studios. These being Scrum, Kanban, and Feature Driven Development – as determined in [22]. Each list described the practices associated with one of the agile frameworks. The participant would be asked to go through each agile framework their studio reportedly used and identify which elements their studio applied and describe how they implemented the element and/or the reasons for why they adapted it. For instance, if a studio said they used Scrum, a list describing each of the Scrum team roles, events, and artifacts would be given to the participant for comment.

Similarly to Koutonen and Leppänen [19], interview questions were worded in the VGD parlance and nomenclature, and were constructed for consistent structure, clarity, and unambiguity.

While the interviews were scripted to ensure that there was sufficient coverage of the RQ, flexibility was given to include variation in the conversation so that a wider range of factors surrounding the central topics of interest could also be captured. In order to ensure a rigorous interview design process and minimize potential researcher bias, the interview script was validated and improved iteratively through reviews with senior (non VGD) software engineers. A professional Scrum coach was consulted to identify and describe each key Scrum practice. The study was approved by the University of Canterbury Human Ethics Committee (reference number: HEC2019/05/LR).

C. Study Recruitment and Inclusion Criteria

The interviews were advertised through email, one-on-one meetings with studio representatives at industry events (e.g. game developer meetups and NZ PlaybyPlay), and via online industry networking channels (e.g. Slack and Discord). The diversity of recruitment techniques was intended to maximize the awareness of and interest in the study in the hopes of

achieving a high response rate for population sampling and so improve external validity. Eight studios responded to these advertisements.

Adopted from Koutonen and Leppänen [19], the only inclusion criteria for participation in the study was that a game development studio employed at least five full-time staff. The reasoning being was that such small independent studios were likely to apply ad-hoc ways of developing rather than agile methods or professional SE practices. It is acknowledged that the majority (62%) of game development studios only employ between one and five members [11]. However, it was expected that studios numbering at least five members would provide more useful insights in the challenge of managing multidisciplinary teams.

The methods to improve the external validity of the study were similar to those used by Musil et al. [18], in that target participants were those who would have deep understanding of the development processes of their studios – such as producers, team leads, or project managers. Eight individual participants representing eight eligible studios were interviewed.

D. Interviews Protocol

The interviews were conducted in person at the relevant studio when possible, or else via video conferencing. The interviewee was greeted, the purpose and protocol of the interview was explained, and an opportunity for asking questions was given. An information sheet and consent form reminding the participants of their rights and informing them of the conditions and ethical considerations of the study was given to review and sign.

The interviews were audio recorded (and video recorded in the case of video conferencing) using two different recording tools as a failsafe, written notes were taken by the researcher as well. Any summary notes taken during the interview were immediately available to the interviewee so they could check their answers. The interviews typically lasted between 40 to 60 minutes.

Selective transcription using a voice typing feature was used for speed and efficiency, and was sufficiently accurate for the purposes of analysis. For concision, speech tics were removed from the transcripts. The answers were sometimes edited where clarity on conveying key ideas was needed, or when the researcher asked a confirmatory question or made a comment, and such editions were denoted with brackets ‘[]’.

Data anonymization was used, where the transcriptions were sanitized by removing or generalizing references to specific individuals, companies and products. All these editions were done carefully and conservatively so that the original intent and meaning of the answers was not altered.

Once the transcription was complete it was offered to the interviewee for review for up to a week to make corrections or clarifications to their answers. This gave opportunity for the interviewee to check that their views and thoughts were faithfully preserved given the editions made by the transcriber.

E. Interviews Analysis

NVivo 12 was used for the coding of the transcriptions and for generating visualizations of the data and Microsoft Excel was also used to graph some of the data.

The General Inductive Approach (GIA) [41] was applied for the qualitative analysis of the interviews. Built from existing models such as grounded theory, discourse analysis, and phenomenology, the GIA is a highly regarded and popularly used methodology. In brief, the GIA process is as follows:

- 1) The raw data is read closely, interpreted and summarized in categories of recurring concepts (known as coding).
- 2) Common themes are identified from the categories.
- 3) The themes are used to construct an experiential and/or process domain model directly linked to the RQ [41].

The GIA advocates several means to validate the qualitative model to demonstrate trustworthiness, credibility, and dependability of the analysis. Hence, code consistency checks, as outlined in the GIA, were performed by the research team in order to evince these traits. For instance, the data analysis process was validated by another researcher who carried out independent parallel coding and generated the same or similar concepts and themes. These results were compared with those themes generated by the primary-researcher for discrepancies leading to discussions within the research team on whether to revise or refine the themes and concepts once consensus was reached. Two stakeholder checks of the generated themes and interpretations were carried out with several of the industry participants to verify the correctness of the interpretations [41]. Again, the primary researcher used the feedback from the participants to make any needed revisions.

V. RESULTS

The wide range of insights the interviews produced were coalesced into five key themes as they related to the RQ. These generally describe how and/or why agile frameworks are used and adapted in VGD, along with the implications these adaptations can have for the VGD team and process. Other non-agile development frameworks and methodologies were excluded from this analysis as none of the participants mentioned using them. A summary of these themes now follows.

A. Choosing Agile Frameworks for VGD

In line with Murphy-Hill et al. [25], interviewed studios reported to choose agile frameworks for their flexibility in handling the highly changeable and experimental VGD process. Many studios were partial to agile techniques as many of their staff were already familiar with these from previous employment in other SE domains.

The reported studio experiences suggest that agile methods empower developers to have more influence over the design direction of a game, thereby increasing their investment. However, without proper direction, these factors can easily result in uncontrolled feature creep, development dead ends, and pivots.

Another reason agile frameworks are used is because they are suitable for the two main deliverable pipelines (or workflows): Scrum for game mechanics/features and Kanban for asset creation. Because of its structured yet flexible short-term delivery and stakeholder feedback cycle, Scrum is ideal for producing prototypes, feature tools, core game mechanics, and additional post-release feature content periodically.

Due to its continuous release cycle, minimal overhead, and ease-of-pivoting attributes, studios also may use a process resembling Kanban. Kanban is well fit for both individual team members or discipline-based groups (such as artists) delivering art and other assets, which must be done in production line stages (e.g. concept → story boarding → modeling → texturing → ...).

Kanban is also used for fixing game defects as it allows developers to react to sudden requirements changes or priorities. Similar to Kasurinen et al.'s findings [20], the high number of inter-dependencies in the VGD asset pipeline and overhead that Scrum brings makes it unsuitable for this kind of work. As one studio explains its essential to get the right framework for the right context: *“Agile: apply the right tool in the right situation at the right time... To deliver the core feature set it made sense to use Scrum, because is exploratory work and anything can change at any time. But the art pipeline and the development of levels is pretty predictable and steady, and so it makes more sense to use Kanban as you work your way top-down the list of items”*.

B. Agile Framework Adaptation

How and why studios adapt agile frameworks is dependent on each studio's individual context – size, project type, organizational structure, etc. However, there are common business, organizational, and ideological themes of influence. Most studios follow a periodic milestone release structure, and Scrum and Kanban practices are applied or optimized for specific kinds of deliverables to be produced to fit this structure.

For Scrum, only two of the eight studios implemented all of the Scrum events, artifacts and practices. Stakeholders and leadership members may only attend significant project milestone reviews (rather than Sprint Reviews) where the direction of the game and project risks are discussed, and features are added or cut as necessary.

Sprint Reviews, on the other hand, may just involve the developers informally sitting around a TV playing the game build as it stands. Daily Scrums were almost universally employed in the Scrum manner, but there was a danger for artists and developers to gravitate to their own disciplines and form silos.

Studios differed on whether Sprint Planning should involve the entire team or just the producers. Retrospectives were controversial. They are used by studios that value them, and not by studios who do not. Definitions of Done are not used by many studios at all. As one studio relates about their Scrum adaptation: *“Our development methodology is a mixture of a bunch of things... However, we started with Scrum and have morphed it into our own thing”*.

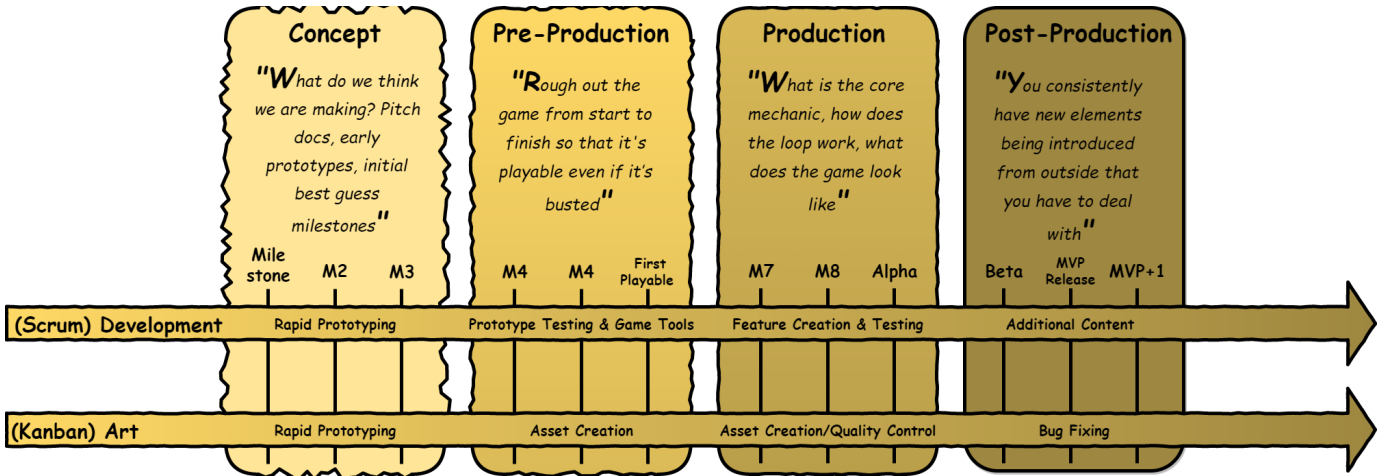


Fig. 1. Agile VGD process model with phases and delivery pipelines/workflows. Milestones shown are examples only.

Kanban is often stripped of several of its cadences, performance metrics, and work-in-progress limits because studios do not perceive them to be useful for the purposes of producing assets. For some studios, efficiency is less important than results so that assets or features may be dropped if they are not completed in a certain period. Further, similarly to Koutonen and Leppänen's findings [19], and those of Ruonala [34], not many studios use Kanban practices adequately due to lack of knowledge – as one project manager put it, *"Me and perhaps only three other people in the studio even know what these [Kanban terms] mean"*.

C. Game Development Roles

The interviews revealed that a significant area of difference between traditional Scrum and game development is around team roles and composition. VGD introduces two new hybrid roles. The producer fulfills some of the PO's responsibilities by planning sprints and milestones, grooming the backlog, delegating tasks to the team, and enforcing leadership decisions. They also have some, but not all, Scrum Master (SM) responsibilities such as facilitation and mentoring. Small studios often could not afford to have a full time SM. The team lead also surrogates some of the SM functions similarly to the producer.

Studios showed the greatest diversity and confusion in defining who had the Product Owner (PO) role (if anyone), but most often the PO was a subset of the project manager's or director's responsibilities. However, producers are also often thought of as varying degrees of project managers. Consequently, the project leadership role can become muddled in this configuration of the PO, SM, and team lead.

Team sizes varied between studios but only two studios had sizes exceeding the Scrum recommendations of 10 or fewer people [17]. Also, the studio's description of their teams reinforces the well-established findings of other research [14] [37] that development teams tend to be heterogeneous in nature with a greater need for specialized skills, which makes cross-functionality harder to maintain for smaller teams. It is also

more difficult for a team to be self-managing when there is a dependency on the PO to communicate a clear game vision, on producers to manage the backlog, and on the team leads to ensure the vision's delivery.

Yet paradoxically, VGD developers enjoy greater creative freedom to explore and experiment on features than what one may expect of traditional agile SE teams who must keep to a requirements brief. Thus, similarly to Hodgson and Briand's findings [36], there is an underlying tension between the need to afford developers sufficient flexibility and freedom of expression, but from within the constraints of the game vision and hierarchical direction: *"The PO acts as a vision holder and communicates with the stakeholders but relies on the team to build the Sprint Backlog, particularly the producers and the leads. However, [the PO] signs off on the backlog so the PO has the final say"*.

D. Phases of Game Development

The data gleaned from the interviews were used to produce an empirical agile VGD process workflow. Confirming the observations of Aleem et al. [42], Ruonala [34], Marklund et al. [23], and Clinton Keith [43], the interviewed studios described what can generally be understood as four distinct game development phases (or stages), see Figure 1.

These phases constrain what type of deliverable needs to be produced, and so determines which agile framework is used and how strictly it is applied. Some studios reported that the phases themselves can be broken up into deliverable milestones such as first playable, alpha, beta, and minimum viable product (MVP). To maximize their effectiveness, the exact use of Scrum and Kanban differs for each phase which will now be discussed.

Concept. Also known as the 'pitch' phase, this is for discovering and experimenting on key 'fun' game mechanics through rapid prototyping. Good market analysis, minimal process overhead, and getting quick feedback is essential in this phase as the project vision is fuzzy and may change drastically. There are many unknowns and constant flux around

the core game vision and the requirements and scope of the project. This means Scrum’s overhead provides less value and so necessitates that light-weight versions of it are implemented very loosely in shorter sprints with less defined goals (e.g. just to ‘find the fun’).

Alternatively, one studio mentioned they used Kanban in this phase. A very general project roadmap with key milestones and risks is also formulated once a picture of the game prototype can also be formed.

Pre-Production. For validating and iterating on game prototypes, building the pipelines to create the final content, and trialing new technology to find out what kind of game is going to be built, and how.

Production. With the game vision more well-defined, game assets are created using Kanban and the core game is built and tested properly to release milestones using Scrum.

Post-Production. Has final testing, release and marketing of the game, as well as post-game maintenance and possible feature updates. Now that the game vision and development process is more fixed, Scrum application looks ‘conventional’ by this point: *“We find that as a project gets more and more mature, and the team knows what they are doing, and the game is more well defined, and towards post-production... the more the processes begin to resemble traditional Scrum”*.

E. Agile Use and Adaptation Issues and Challenges

Studios related many different issues around agile development but had common pain points around agile adoption bias, lack of agile training, agile (mis)perception, Scrum (mis)implementations, and team communication issues.

ScrumBut [44], the deviation from or only partial implementation of the Scrum framework, is common among studios. That is, studios pick and choose certain Scrum practices for preconceived reasons, and this seems to be because agile can be misunderstood or misimplemented. An example of this can be seen in Table I which records studio agile practices in relation to the team dynamics issues they face.

Critics, who had had bad experiences or misimplementations of Agile, often had to be shown the value of Scrum practices as well as the philosophy behind such practices in order to adopt them. A studio’s perception of agile and its effectiveness seems to play a key role in how they implement or adapt an agile framework.

There can be misgivings and adoption bias against Scrum reminiscent to the ‘*Change is Bad*’ fallacy [43]. As Cohn notes that in traditional SE, the love of the status quo and fear of the unknown can lead to ‘agile phobias’ [45]. One studio said: *“Whenever there is a big change people resist a lot... It’s a form of confirmation bias in that they don’t understand you are presenting a way to the same outcome but for less pain”*.

Similar to Politowski et al.’s findings [21], a common lack of training or mentoring in agile practices leads to developers’ misunderstanding how aspects of Scrum work and so have a wrong impression of the framework as a whole. For example, one studio believed that ‘cross functional’ is a quality of individual developers when it actually applies to the whole

team collectively [17]: *“A traditional Scrum team is cross-functional and any ticket can be completed by anybody – this does not work in game development at all, as the discipline is way too specialized”*.

As a consequence of misunderstanding agile, Scrum can be misimplemented – the common ‘Cargo Cult Scrum’/‘Mechanical Scrum’ antipattern whereby studios apply Scrum practices by the book rigidly without understanding or appreciating the intention or agile principles behind each practice [43]: *“Where we are most failing in the adoption of Scrum is that team members apply it mechanically without coming to grips with the principles and values of the process, in other words, professional Scrum. They don’t understand what agile is meant to be”*.

Both management and the team may not be fully committed to or respect Scrum. This can lead to disruptive subversion similar to those articulated by Hodgson and Briand [36] and Murphy-Hill et al. [25]: *“They [management] were not respecting the [Scrum] process and trying to bypass it. And this set a bad example for the rest of the team”*.

This in turn may lead to misassociations with Scrum where success is attributed to other factors and problems are falsely attributed to Scrum, such as Cohn’s ‘waterfallacies’ – which is a mistaken belief or idea about Scrum or agile due to working for too long on waterfall projects [45]: *“I had one of the POs come to me and say ‘the team hates agile...’. But when they described the situation what they described was waterfall. What they were identifying was that there were old habits from the past still inherent in your process, not that agile itself is your problem”*.

VI. DISCUSSION

A. Agile Framework Adaptation in VGD

The most important factors in VGD that influence which agile framework is used and how it is adapted is the type of milestone deliverable, the phase of development, and the type of pipeline. Specialized VGD roles differ from traditional SE and so Scrum roles are not often applied in the same way.

Both iterations of Scrum and Scrum events must be adapted to fit inside a structure of repeating milestones of ideation, design, implementation, (play)testing, feedback, and refinement along fixed phases of development to ensure it delivers value in the right context.

Multiple Scrum sprints are tied to deliver specific milestones, where each milestone are often between one to three months long. A sprint can occasionally be a milestone as in the case in post-production where additional features are being introduced periodically to the game after it has been shipped. Hence, it could be argued that this ‘structured agile’ approach is reminiscent of Water-Scrum-Fall – a hybrid of Scrum operating in a waterfall superstructure [46].

Conversely, in the early concept and pre-production phases, the process to produce quick prototypes for playtesting, feedback, and iteration seem to resemble rapid application development (RAD) [47] far more than it does Scrum.

TABLE I
AGILE PRACTICES IN RELATION TO TEAM DYNAMIC ISSUES FACED. ARROWS INDICATE A TRANSITION OVER TIME

Studio	Way of Working	Team size	Retrospectives?	Action Items?	Scrum Master?	Team Dynamics Issues
1	Agile→ScrumBut	20 with up to 30 extra outsourced	No	No	Producer	Communication, Knowledge Siloing
2	Agile	7→15	No	No	Dev lead	Communication, Accountability, Shared Game Vision, Agile Experience
3	Agile/Kanban	8	No	No	PO/Dev lead	Crunch (in the past)
4	Agile	9	Yes	No	Producer	Communication, Accountability, Process Commitment
5	Scrumban	9	Yes	No	No	Communication (solved by collocation and chat systems)
6	Scrum	5 to 8	Yes	Yes	Yes	Process Commitment, Shared Understanding, Knowledge Siloing
7	ScrumBut	45 broken into teams of 3 to 10	Yes	Yes	Producer	Team diversity, Shared Game Vision
8	Scrum→Scrumban	28 broken into teams of 5 to 9	Yes	Yes	Yes	Shared Understanding, Knowledge Siloing, Process Commitment

The results also suggest that VGD often follows a workflow that reassembles a two-track delivery system – an asset pipeline/workflow using Kanban-like approaches and a feature or mechanic pipeline/workflow using Scrum-like approaches. As noted by Clinton Keith [43], these pipelines may or may not progress through the four development phases in the same cadence – phases may overlap or even revert [34], and the art pipeline often finishes earlier than the feature pipeline. as one studio said, “*We used to use pure Scrum on our programming and Kanban with a little bit of waterfall for our art team due to that our art is being made on a more conceptual basis while of course programming is more of a deliverable-based object. But we found that usually meshing our teams together worked better*”.

This finding underscores the complex, heterogeneous nature of game development and the potential for the associated communication or coordination problems that the two-track pipeline delivery structure may create – particularly for workflow synchronization. Consequently, neither Scrum nor Kanban alone can entirely mitigate this complexity, it seems both agile frameworks must be used in tandem.

One solution for the two-track pipeline problem for live games (i.e. post-release or games-as-a-service) that some studios use, is a development process with elements drawn from both Scrum and Kanban in what could be approximated as *Scrumban* [48]. Kanban’s continuous release structure (e.g. for immediate bug fixing) operates as a second workflow alongside the Scrum iterative structure (e.g. for new content release). The two pipelines are thus bridged, and this facilitates multidisciplinary collaboration. One studio said, “*Scrumban is probably the most efficient and effective way of running a live game*”.

Hence, the use of Scrumban contributes to filling the knowledge gap noted by Engström [38] [39] around how the differences between creative and engineering ways of working are managed effectively. Similar to how Marklund et al. [23] concluded, VGD can perhaps be seen as a necessary amalgamation of diverse development models such as Scrum, Kanban, RAD, and waterfall, and this only *appears* ad-hoc to

the untrained eye.

B. Studio Perceptions of Agile

The limited and contextualized implementations of Scrum and Kanban are somewhat in contrast to the industry’s overall perception of their adherence to them as highlighted in [22]. A studio may maintain a particular view of Scrum or Kanban regardless of how far they have adapted the frameworks away from their intended purpose.

This finding also supports the conclusions presented by McKenzie et al. [22], in that if a studio blames agile for their development issues, sometimes it may be less that the framework is at fault than an agile framework or practice is being used outside its intended function: a case of applying a good framework in the wrong context.

These perception issues highlight that for many game studios adopting agile properly is often a work culture and company values problem. Crucially, those studios who had the closest Scrum-aligned implementation exhibited the clearest agile mindset and were the most positive about Scrum, calling it a “*flexible framework*”. In contrast, those studios that deviated away from core Scrum practices were less positive about Scrum. Contradicting the other studios, they said that “*textbook Scrum is just simply not agile enough*” as it is a “*set in stone process*” and they did not want to burden developers with “*unnecessary overhead*”. Yet tellingly, these studios also stated that they had communication and/or leadership issues.

Mechanical (mis)application of Scrum practices generates resistance particularly from senior developers, and consequently leads to the aforementioned agile perception problems. One project manager thought the problem often came down to the team’s (miss)expectations of a Scrum event, rather than the event itself being flawed. Therefore, developers often need the value of an agile practice or Scrum event explained to them and/or experience the value for themselves before they buy into it.

A studio even said they adopted agile purely out of peer pressure – a finding which affirms both Ruonala’s [34] and Marklund et al.’s [23] suggestion that many studios adopt

certain practices out of popularity or tradition rather than thinking through the inherent value they bring. So, it seems that studios who embrace “professional Scrum” with an agile mindset seem to get more value from the framework precisely because they understand the principles behind each Scrum event sufficiently to ensure such events are value-driven.

Conversely, those studios who shy away from such Scrum practices because they view them as flawed may experience more development antipatterns. This supports McKenzie et al.’s [22] inference that a studio’s level of agile adoption is correlated with their attitude towards it, and this seems to be a circular relationship. Therefore, there is a non-trivial relationship to how agile is perceived relative to how a game studio adapts the agile framework, and whether this perception and/or adaptation leads to common developmental issues.

C. Agile Adaptation and VGD Challenges

Multidisciplinary Team Dynamics. Ruonala [34] noted that for one VGD studio, communication problems were solved when the team adopted Scrum and maximized the benefits of the XP best practice of collocation [49] through an open office seating arrangement. Similarly in this study, one interviewed studio mentioned using a ‘pod’ seating arrangement whereby a team of artists, designers, and programmers sat together in a circle in the office and this helped their communication problems.

McKenzie et al. [22] observed that those studios with team compositions and sizes larger than the Scrum guide recommends may be having communication and coordination issues, and it was found from the interviews that this indeed was the case. As can be seen in the interview data shown in Table I, studios who followed the Scrum definitions for team sizes, retrospectives, and SMs reported no communication issues, whereas those studios not using these practices had the most apparent communication and accountability problems.

Moreover, the interview results suggest that Scrum antipatterns such as ‘Business as Usual’ and ‘PO without authority’, identified by Eloranta et al. [44], which occur regularly in SE also appear to be present in VGD. Notably, even the most agile-minded and Scrum aligned studios nevertheless found it difficult to maintain shared understanding among the disciplines, break down knowledge silos, and keep the team committed to the agile processes they had put in place. It is noteworthy that, while following SE agile practices correctly helps address some issues, agile frameworks alone are not enough to meet all the multidisciplinary collaboration challenges of VGD.

Project Management. McKenzie et al. [22] noted that game studios had a diverse understanding and use of the PO’s role in the Scrum team, and this may be causing leadership problems. From the interviews of this study, those studios who had the closest mapping of project roles to the Scrum definitions of POs and SMs had no leadership issues. Conversely, studios who did not have a PO role, or had it spread over multiple people (an antipattern the Scrum guide forbids [17]), articulated they had various kinds of leadership issues.

Another observation is that producers often decompose the Product Backlog items into work tasks and often delegate these out too – a responsibility the Scrum guide should only be done by the developers themselves [17]. This will have an impact on a developer’s sense of ownership for their work, their understanding of the game vision, and their ability to self-manage.

One studio noted that game developers often assume that the team shares their individual thinking process and that there is collective understanding of the game vision when it is not the case. Consequently, it can be difficult to convince developers to come to Sprint Reviews as they see them as a waste of time. So, it would seem that without proper direction from a clearly defined PO or SM, and without opportunities to check their assumptions (e.g. during Sprint Reviews or Retrospectives), then communication and coordination issues, which lead to team conflict and agile antipatterns, are inevitable for many studios.

D. Summary

While not entirely problem free, the most agile minded and agile-aligned studios had noticeable absences of some VGD management and team dynamics challenges. Ruonala [34] speculated that “*some companies may rush to adopt agile methods as a ‘trendy’ thing to do, without proper training, and subsequently run into problems*”. The findings from this study confirm that this can often be exactly the case, and so one of the root causes of these problems appears to be a lack of training and knowledge in agile SE practices within the VGD industry.

To say, however, that a studio not practicing certain agile practices (e.g. collocation or retrospectives) is causing their communication issues, would be too simplistic. Certainly, such a situation would not be helped by not using practices meant to foster team openness, trust, and improve communication. Hence, it is fairly safe to assume that not following key agile practices, or following such practices rigidly without understanding their value, are indeed contributing factors to the problems that some game studios are facing.

This strengthens the claims of the likes of Musil et al. [18], Petrillo and Pimenta [35], and Politowski et al. [33] that SE agile best practices can be used to solve some VGD workflow or management problems – but only when properly understood *and* properly applied. As Aleem et al. [14] aptly put it, “*the game industry can follow the good and proven practices of traditional SE, but only a clear understanding of these practices can enhance the complex game development engineering process*”.

E. Study Limitations

As noted by Murphy-Hill et al. [25], interviews are essentially qualitative, and represent the subjective opinions of developers. Moreover, the actual development process of the studio’s interviewed was not observed, hence the findings are based on a studio’s reported practices, not necessarily their actual ones. However, investigating the developer’s perceptions

and impressions of their practices was precisely one of the study's primary objectives, so its construct validity should be preserved. Determining the agility of a studio requires all aspects of the business to be considered, including marketing, sales, and HR. However, for the purposes of the study, only the development process was investigated.

While domain experts were consulted and steps were taken to ensure comprehensiveness of the interview questions, the study was not, however, piloted. Hence, it is possible some aspects of the interview script were lacking, which would affect its content validity.

The primary researcher conducted the data collection and data analysis phases alone and without blinding processes. Hence researcher bias or data torturing is possible. However, quality checks by the research team as well internal validity reviews of the analysis interpretations by industry stakeholders were carried out to reduce these potential issues.

There is a myriad of contextual factors and other complexities inherent in VGD that could potentially influence or contribute towards team dynamics and leadership challenges. Hence, it is possible that there are unknown confounding factors not captured by the interviews that may have influenced the results of this study. Just as for Koutonen and Leppänen [19], the use of case studies for direct observation of studio practice would help triangulate these results.

Generalization of the findings was impacted by the small sample size. The most common type of NZ studio is an emerging independent startup that have very small teams producing online or mobile games in short development cycles [50] – which is like the situation for most game developers in general [11] [18] [19] [20] [21]. Eight studios, representing about 27% of the NZ game studios at the time of the interviews, participated in this study. However, these studios concentrate around half of the NZ game development employees population at the time of the study [12]. Hence the findings can be considered representative. Nevertheless, more detailed demographic information about each studio (e.g. the studio's primary game genre and platform they develop for, average years of developer experience, among others) could improve the generalization of the findings.

VII. CONCLUSION

This paper details an empirical study carried out in the NZ VGD industry using semi-structured interviews with the intent of understanding the perception, adoption and adaptation of agile practices, and how these factors may be linked to common VGD challenges.

Scrum and Kanban are popular agile frameworks that are used within the VGD industry, and when properly understood and applied, such agile SE best practices seem to be effective in addressing some known collaboration challenges. However, these frameworks must necessarily be adapted for each studio's unique context, as well as for the multidisciplinary needs and constraints inherent to the VGD process. One such constraint includes the diverse goals for what commonly appears to be four development stages that comprise a waterfall-like

milestone delivery superstructure. Another constraint is the parallel two-track production pipelines of art and asset creation (which often uses Kanban-like elements) and game feature or mechanic creation (which often uses Scrum-like iterative development). Hence, in general, VGD is often an amalgamation of different development frameworks, and so VGD sometimes can be ad-hoc, and at other times it only appears to be.

The actual limitations of agile SE frameworks within VGD are complicated by the fact that the frameworks are also commonly misunderstood and misapplied in key areas around project management and multidisciplinary collaboration. Development problems, antipatterns, or failures caused by these misapplications can be subsequently misattributed to the frameworks. Further, only a minority of studios expressed a genuine agile mindset. This seems to substantiate many of the propositions of previous studies by fairly establishing there is a relationship between a studio's (mis)perception and (mis)implementation of agile frameworks and the collaboration challenges they face. One of the root causes of this particular problem appears to be a common lack of experience or training with agile practices within the VGD industry.

Still, currently no single 'out of the box' agile framework by itself seems to be sufficient to meet both the SE and creative production needs of VGD – the frameworks must be adapted and/or be used in conjunction. For example, a hybrid process resembling Scrumban is reported by some studios to alleviate this tension. These findings suggests that new video game specific development framework(s) or process models incorporating best principles and practices of agile SE, complemented by those of creative production, ought to be explored. This would provide a significant benefit to both tertiary education intuitions and the industry itself.

As future work, to perform industrial case studies for obtaining a more granular analysis of the practices, roles, and outputs of each development phase and how these elements are affected by the contextual factors of differing game projects is suggested. This would include studios' perception and adaptation of both agile SE and the hitherto unexplored creative production practices, contributing to a more holistic best practice VGD model.

ACKNOWLEDGMENTS

Funding and support for this study came from the Computer Science and Software Engineering Department, School of Product Design, and Applied Immersive Gaming Initiative at the University of Canterbury.

The authors wish to thank the New Zealand Game Developers Association and the Christchurch Game Developers Meetup for their support, as well as the NZ game studios who participated in this study.

REFERENCES

- [1] Statista. (2020) Global box office revenue from 2004 to 2019, by region. Statista. Accessed: 18-Nov-2020. [Online]. Available: <https://www.statista.com/statistics/264429/global-box-office-revenue-by-region/>

- [2] ——. (2020) Global digital music revenue from 2004 to 2019. Statista. Accessed: 18-Nov-2020. [Online]. Available: <https://www.statista.com/statistics/263109/global-digital-music-revenue/>
- [3] ——. (2020) Video game market value worldwide from 2012 to 2023. Statista. Accessed: 18-Nov-2020. [Online]. Available: <https://www.statista.com/statistics/292056/video-game-market-value-worldwide/>
- [4] T. Wijman. (2020, May) The world's 2.7 billion gamers will spend \$159.3 billion on games in 2020; the market will surpass \$200 billion by 2023. NewZoo.com. Accessed: 11-Jun-2020. [Online]. Available: <https://newzoo.com/insights/articles/newzoo-games-market-numbers-revenues-and-audience-2020-2023/>
- [5] F. Laramee, *Secrets of the Game Business, 1st Edition*. Hingham, Massachusetts, USA: Charles River Media, February 2003.
- [6] Forbes. (2008, 11) Cooking up a blockbuster game. Forbes. Interview with Geoffrey Zatkin, head of Electronic Entertainment Design and Research. Accessed 18-Nov-2020. [Online]. Available: https://www.forbes.com/2008/11/21/games-cedar-developers-tech-ebiz-cx_mji_1121cedar.html?sh=2141a8d01f00
- [7] B. Hile. (2020, February) 10 closed game studios that produced amazing games. gamerant.com. Accessed: 13-Nov-2020. [Online]. Available: <https://gamerant.com/closed-game-studios-best-games/>
- [8] D. Callele, E. Neufeld, and K. Schneider, "Requirements engineering and the creative process in the video game industry," in *Proceedings of the IEEE International Conference on Requirements Engineering*. New York, NY, USA: IEEE, 01 2005, pp. 240–250.
- [9] M. W. Jr., P. Sathiyarayanan, M. Nagappan, T. Zimmermann, and C. Bird, "“what went right and what went wrong’: An analysis of 155 postmortems from game development,” in *Proceedings of the 38th International Conference on Software Engineering (ICSE 2016 SEIP Track)*. New York, NY, USA: ACM, May 2016. [Online]. Available: <https://www.microsoft.com/en-us/research/publication/what-went-right-and-what-went-wrong-an-analysis-of-155-postmortems-from-game-development/>
- [10] T. Tschang, "Videogames as interactive experiential products and their manner of development," *International Journal of Innovation Management (ijim)*, vol. 09, pp. 103–131, 2005.
- [11] P. S. Inc, "The state of game development report: 2020 & beyond," Perforce, Tech. Rep., 8 2020. [Online]. Available: www.perforce.com
- [12] NZGDA. (2020, February) Interactive game exports double in two years to \$200m. nzgda.com. Accessed: 23-Jun-2020. [Online]. Available: <https://nzgda.com/news/survey2019/>
- [13] T. Reed, M. Wynands, Z. Hobson, J. Boggs, L. Morris, and A. Mclean, "PlaybyPlay2019 keynote talk ten: Industry panelists," New Zealand Games Festival, Personal Communication, April 2019, <https://playbyplay.co.nz/>.
- [14] S. Aleem, L. F. Capretz, and F. Ahmed, "Game development software engineering process life cycle: a systematic review," *Journal of Software Engineering Research and Development*, vol. 4, no. 1, p. 6, November 2016. [Online]. Available: <https://doi.org/10.1186/s40411-016-0032-7>
- [15] H. Engström, B. B. Marklund, P. Backlund, and M. Toftedahl, "Game development from a software and creative product perspective: A quantitative literature review approach," *Entertainment Computing*, vol. 27, pp. 10–22, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1875952117300927>
- [16] C. Kanode and H. Haddad, "Software engineering challenges in game development," in *ITNG 2009 - 6th International Conference on Information Technology: New Generations*. New York, NY, USA: IEEE, 2009, pp. 260 – 265.
- [17] K. Schwaber and J. Sutherland, "The scrum guide: The definitive guide to scrum: The rules of the game," Scrumguides.org, Tech. Rep., 2020.
- [18] J. Musil, A. Schweda, D. Winkler, and S. Biffl, "A survey on a state of the practice in video game development, report ifs-qse 10/04," Institute of Software Technology and Interactive Systems, Vienna, Tech. Rep., 03 2010.
- [19] J. Koutonen and M. Leppänen, "How are agile methods and practices deployed in video game development? a survey into finnish game studios," in *Agile Processes in Software Engineering and Extreme Programming*. Berlin, Heidelberg: Springer, 06 2013, pp. 135–149.
- [20] J. Kasurinen, "Games as software: Similarities and differences between the implementation projects," in *Proceedings of the 17th International Conference on Computer Systems and Technologies 2016*, ser. CompSysTech '16. New York, NY, USA: ACM, 2016, pp. 33–40. [Online]. Available: <http://doi.acm.org/10.1145/2983468.2983501>
- [21] C. Politowski, D. De Vargas, L. Fontoura, and A. Foletto, "Software engineering processes in game development: a survey about brazilian developers' experiences," in *SBGames*. New York, NY, USA: ACM, 2016.
- [22] T. McKenzie, M. Morales-Trujillo, and S. Hoermann, "Software engineering practices and methods in the game development industry," in *Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*. New York, NY, USA: ACM, October 2019, pp. 181–193. [Online]. Available: <https://doi.org/10.1145/3341215.3354647>
- [23] B. Berg Marklund, H. Engström, M. Hellkvist, and P. Backlund, "What empirically based research tells us about game development," *The Computer Games Journal*, vol. 8, no. 3, pp. 179–198, December 2019. [Online]. Available: <https://doi.org/10.1007/s40869-019-00085-1>
- [24] F. Petrillo, M. Pimenta, F. Trindade, and C. Dietrich, "What went wrong; a survey of problems in game development," *Comput. Entertain.*, vol. 7, no. 1, pp. 13:1–13:22, Feb. 2009. [Online]. Available: <http://doi.acm.org/10.1145/1486508.1486521>
- [25] E. Murphy-Hill, T. Zimmermann, and N. Nagappan, "Cowboys, ankle sprains, and keepers of quality: How is video game development different from software development?" in *Proceedings of the 36th International Conference on Software Engineering (ICSE 2014)*. New York, NY, USA: ACM, June 2014.
- [26] R. V. O'Connor and S. Basri, "The effect of team dynamics on software development process improvement," *International journal of human capital and information technology professionals*, vol. 3, no. 3, pp. 13–26, July 2012. [Online]. Available: <https://dl.acm.org/doi/10.4018/jhcitp.2012070102>
- [27] S. Sarin and G. C. O'Connor, "First among equals: The effect of team leader characteristics on the internal dynamics of cross-functional product development teams," *Journal of Product Innovation Management*, vol. 26, no. 2, pp. 188–205, January 2009. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-5885.2009.00345.x>
- [28] E. Crevoshay, S. Hays, R. Kowert, R. Boccamazzo, K. Dunlap, J. Cocks, R. Skimmons, C. Kocurek, J. VanDenBogaard, and L. Rogers, "State of the industry 2019: Mental health in the game industry," Take This Inc, 9805 NE 116th St, Suite 7411, Kirkland, WA 98034, Tech. Rep., July 2019.
- [29] J. Weststar and S. Kumar, "IGDA developer satisfaction survey 2019 industry trends and future outlook report," International Game Developers Association, Industry trends 111820, 11 2020. [Online]. Available: https://s3-us-east-2.amazonaws.com/igda-website/wp-content/uploads/2020/11/25095744/IGDA-DSS-2019-Industry-Trends-Report_111820.pdf
- [30] M.-J. Legault and K. Ouellet, "So into it they forget what time it is?: Video game designers and unpaid overtime," in *Managing dynamic technology-oriented businesses: High-tech organizations and workplaces*, J. Dariusz & M. Abigail, Ed. Hershey: IGI Global, 2012, pp. 82–102. [Online]. Available: <https://www.igi-global.com/gateway/chapter/67430>
- [31] H. Edholm, M. Lidström, J.-P. Steghöfer, and H. Burden, "Crunch time: The reasons and effects of unpaid overtime in the games industry," in *Proceedings of the 39th International Conference on Software Engineering: Software Engineering in Practice Track*, ser. ICSE-SEIP '17. IEEE Press, May 2017, pp. 43–52. [Online]. Available: <https://www.computer.org/csdl/proceedings-article/icse-seip/2017/07965428/12OmNviZlpt>
- [32] J. Weststar and M.-J. Legault, "Why might a videogame developer join a union?" *Labor Studies Journal*, vol. 42, no. 4, pp. 295–321, December 2017. [Online]. Available: <https://doi.org/10.1177/0160449X17731878>
- [33] C. Politowski, F. Petrillo, G. C. Ullmann, and Y.-G. Guéhéneuc, "Game industry problems: An extensive analysis of the gray literature," *Information and Software Technology*, vol. 134, p. 106538, June 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0950584921000252>
- [34] H.-R. Ruonala, "Agile game development : A systematic literature review." Pietari Kalmin katu 5 FI-00014 University of Helsinki, Finland: University of Helsinki, Faculty of Science, Department of Computer Science, 2017. [Online]. Available: <https://helda.helsinki.fi/handle/10138/186579>
- [35] F. Petrillo and M. Pimenta, "Is agility out there? agile practices in game development," in *SIGDOC 2010 - Proceedings of the 28th ACM*

- International Conference on Design of Communication*. New York, NY, USA: ACM, 01 2010, pp. 9–15.
- [36] D. Hodgson and L. Briand, “Controlling the uncontrollable: ‘agile’ teams and illusions of autonomy in creative work,” *Work, Employment & Society*, vol. 27, no. 2, pp. 308–325, 2013.
- [37] A. Osborne O’Hagan, G. Coleman, and R. O’Connor, “Software development processes for games: A systematic literature review,” in *Communications in Computer and Information Science*, vol. 425. New York, NY, USA: Springer, June 2014.
- [38] H. Engström, “GDC vs. DiGRA: Gaps in game production research,” in *DiGRA - Proceedings of the 2019 DiGRA International Conference: Game, Play and the Emerging Ludo-Mix*. DiGRA, August 2019. [Online]. Available: http://www.digra.org/wp-content/uploads/digital-library/DiGRA_2019_paper_182.pdf
- [39] H. Engström, *Game Development Research*, 1st ed. University of Skövde, Sweden, November 2020.
- [40] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas, “Manifesto for agile software development,” 2001. [Online]. Available: <http://www.agilemanifesto.org/>
- [41] D. Thomas, “A general inductive approach for analyzing qualitative evaluation data,” *American Journal of Evaluation*, vol. 27, no. 2, pp. 237–246, 06 2006.
- [42] S. Aleem, L. F. Capretz, and F. Ahmed, “A digital game maturity model (DGMM),” *Entertainment Computing*, vol. 17, pp. 55 – 73, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1875952116300246>
- [43] C. Keith, *Agile Game Development: Build, Play, Repeat*, 2nd ed. Upper Saddle River, NJ, USA: Addison-Wesley, Pearson Education Inc., 2020.
- [44] V.-P. Eloranta, K. Koskimies, and T. Mikkonen, “Exploring ScrumBut—an empirical study of scrum anti-patterns,” *Information and Software Technology*, vol. 74, pp. 194–203, December 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0950584915002050>
- [45] M. Cohn, *Succeeding with Agile: Software Development Using Scrum*, 1st ed. 501 Boylston Street, Suite 900 Boston, MA 02116, USA: Addison-Wesley Professional, 2009.
- [46] D. West, M. Gilpin, T. Grant, and A. Anderson, “Water-scrum-fall is the reality of agile for most organizations today,” *Forrester Research*, vol. 26, no. 2011, pp. 1–17, July 2011.
- [47] J. Martin, *Rapid Application Development*. Equitable Building, New York, USA: Macmillan Publishing Co., Inc., 1991.
- [48] C. Ladas, *Scrumban-essays on kanban systems for lean software development*. Modus Cooperandi Press, 2009.
- [49] K. Beck and C. Andres, *Extreme Programming Explained: Embrace Change*, 2nd ed. Addison-Wesley Professional, 2004.
- [50] S. Knightly, “Interactive aotearoa: Driving growth and wellbeing through interactive media,” New Zealand Game Developers Association, Tech. Rep., 2019.