



CHALMERS
UNIVERSITY OF TECHNOLOGY



UNIVERSITY OF GOTHENBURG

Game Design Feedback Collection Methods in Pre-Release Game Development

Provided by Göteborgs universitets publikationer - e-prints och e-arkiv



Upprättad av bibliotekets tekniska stödet vid Göteborgs universitet

Bachelor of Science Thesis in Software Engineering and Management

FIRAS CHEAIB
OMAR FAWAL

Department of Computer Science and Engineering
UNIVERSITY OF GOTHENBURG
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019



The Author grants to University of Gothenburg and Chalmers University of Technology the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let University of Gothenburg and Chalmers University of Technology store the Work electronically and make it accessible on the Internet.

Game Design Feedback Collection Methods in Pre-Release Game Development

© Firas Cheaib, June 2019.

© Omar Fawal, June 2019.

Supervisor: Jan-Philipp Steghöfer

Examiner: Richard Berntsson Svensson

University of Gothenburg

Chalmers University of Technology

Department of Computer Science and Engineering

SE-412 96 Göteborg

Sweden

Telephone + 46 (0)31-772 1000

Game Design Feedback Collection Methods in Pre-Release Game Development

Firas Cheaib, Omar Fawal
Dept. of Computer Science and Engineering
Chalmers | University of Gothenburg
Gothenburg, Sweden
guschefi@student.gu.se,
gusfawom@student.gu.se

Abstract—The development of games is secretive in nature due to its creative constraints, and a project can run over the course of a few years. With the advent of agile methodologies, software projects have involved customers in the development process to iterate on received feedback. By exploring the different methods game developers employ to involve customers or tackle issues that interfere with the value of the final product, this study offers an insight into what practitioners actually do to collect the feedback they deem useful. We find that there are two main categories of feedback methods, those that are internal to the company and those that involve potential customers. Within these categories, different mechanisms are employed with differing goals and targets at different stages of the development process. While there are clear patterns on what constitutes useful feedback to practitioners, the implementation of those feedback collection mechanisms differs across the industry.

I. INTRODUCTION

Agile software development encourages the inclusion of customers and end users within the process. This ensures that the product is valuable to customers and allows them to share their feedback throughout development as early as possible. Software engineers make use of this information by continuously improving the product until the time of release. Furthermore, software development practices suggest the production of a Minimum Viable Product (MVP) as soon as possible, precisely to produce value to the customer in a timely manner, and to be able to elicit feedback and minimize costs in further iterations. In contrast, large video game projects are enormous undertakings with increasing costs and large development teams [1].

The design decisions done in the early phase of development shape the rest of the game in a significant manner. That is, they influence a multitude of elements such as which mechanics will be implemented or the size and length of levels [2]. Consequently, these decisions are then implemented by developers who, as mentioned previously, benefit from early feedback. In addition to issues found in traditional software development, such as bugs, game developers must ensure game mechanics are coherent and offer value to players [3]. Furthermore, these mechanics shape the gameplay that customers will experience. If the gameplay and mechanics do not create a fun experience, they may stop playing.

Having effective means of collecting useful feedback on game design elements during development (pre-release) is

essential. Due to the creative nature of the industry and the multi-disciplinary nature of game development teams, many different concerns (design, production, etc.) must be addressed. This can often make it difficult to deliver value to the customer when video game development cycles take several years. To find ways around this, video game companies employ several methods such as playtesting, betas, and demos in order to showcase features to customers or dedicated play testers. These methods would primarily focus on the gameplay elements and whether certain mechanics provide value to the customers. Internally, developers might regularly submit level or feature tests to dedicated Quality Assurance staff or invite playtesters to play the game and submit feedback under non-disclosure agreements.

QA staff is usually in charge of specific tasks attempting to break certain elements and finding bugs. In the case of demos or beta builds open to the public, this is done so late that major overhauls are not possible during development and might be inevitable after release [4]. By the time customers finally get their hands on a playable build, they might voice concerns regarding gameplay elements or mechanics that cannot be changed without a lengthy delay or negative press. If a player thinks a basic game mechanic is detrimental to the rest of the game, this cannot be changed without overhauling other elements that depend on it. Therefore, if these concerns are not voiced early on, the ramifications can affect the game's success. Additionally, internal feedback may be handled differently depending on the concerns of the development staff and some issues might remain unresolved when the game finally ships. With all this in mind, updating or patching the game could be very costly after the game has shipped, resulting in the potential removal of features if even possible late into development and an alienation of the consumer base.

While there is research that outlines the role that Quality Assurance (QA) plays during development [5], there are no studies exploring the different methods used across the industry. Furthermore, most research on the topic uses postmortems as the main data source and this could lead to missing some details otherwise obtained through direct contact with professionals [6].

This study aims to provide insights on the game industry's experience handling these issues to explore the feedback

methods used by companies, how these methods tie into the development process and how effective they have been in allowing these companies to make game design decisions that provide value to the customer. The results should outline the methods that influence game design decisions and give insight on the consequences of using one or the other based on the experience of industry professionals. This would give potential researchers specific areas in which concrete experiments could be conducted in order to measure the effectiveness of various methods. Practitioners can also learn from the experiences undertaken by professionals in large projects outlined in the findings. Therefore, the results should provide a snapshot of industry practices, within the scope of our study, to professionals who want to implement feedback collection mechanisms in their projects.

II. RELATED WORK

The game industry has rapidly risen in popularity to become the multi-billion dollar industry of today. According to the Entertainment Software Association, over \$29 billion have been spent in 2018 on video game content in the United States alone [7]. Accompanying it was a climbing complexity in game development, which led to projects on a much larger scale as customer expectations continue to rise [1]. This presents technical challenges such as long compilation times, large file dependencies, and complex simulations [1]. During the ICST 2018 conference, a Technical Director at the Quality Engineering group at EA DICE mentioned that during the development of their most recent title, *Battlefront II*, there were “effectively 1400 people checking into the source base at any given point” [3]. Companies also face a design risk, which is to create a product that satisfies the customers’ needs with fun gameplay [1].

Requirements in game development tend to be more subjective than traditional software development, with functional requirements being less useful overall [8]. Therefore, developers find the requirements to be frequently unclear throughout development [8], [9]. Even when a detailed concept and design is provided, it does not necessarily translate to valuable entertainment to the consumer [8]. Too much planning can limit the creative process with the developers losing sight of what produces an enjoyable experience [6]. Thus, game designers change plans and requirements often, which could cause the developers to go into architectural debt if they plan too far ahead [8]. This can be a reason why code produced by developers is not used, or thrown away, more often than in other software development fields [8], [9].

Clanton [10] provides three categories for the different game issues encountered based on human-computer interaction: game interface, game mechanics, and gameplay. *Game interface* represents what device is used to interact with the game along with its software interface. *Game mechanics* is the physics of the game, depicting what actions can be performed. *Gameplay* represents the game’s purpose, or goal, that is aimed for by users.

With the increasing customer involvement in the game development process due to agile development becoming more prevalent, customers have the potential to provide feedback that guides and detects issues in the game [11]. Usability testing can allow customers to engage with the game directly and give immediate feedback, which has been shown to be helpful in finding game design issues and bugs that developers would not have otherwise found [12]. Nielsen provides extensive heuristics to inspect usability, however, they are mainly targeted for traditional software interfaces, which may not be reflective of game development [13]. Federoff analyzed the relevance of Nielsen’s heuristics for this reason, by comparing it to game heuristics from other literature [5]. Federoff found that 16 of the 30 identified game heuristics did not have a comparable heuristic from Nielsen’s work [5]. All 16 are concerned with gameplay, confirming to be unrelated to other software fields. While these heuristics help explain how a game’s design can be lacking, usability testing is only one method that game development companies can use to get feedback to compare with heuristics.

III. RESEARCH METHODOLOGY

The research was conducted as an exploratory case study using data compiled by developers themselves after the end of projects as well as individual interviews with representatives from different video game development companies in North America and Europe. The study was conducted in several phases, namely starting with a review of literature, followed by a data collection phase (interviews and postmortems) and finally an analysis of the data collected.

A. Research Questions

Throughout this study, we aim to answer the following research questions:

- RQ1) What constitutes useful feedback resulting from QA testing and customer involvement in video game software development?
- RQ2) What are the different methods game companies utilize to acquire user feedback on game design elements during the pre-release stage?
- RQ3) How effective are these methods in providing useful feedback, according to those involved in the development of video games?
- RQ4) How do game companies include these methods in their development process?
- RQ5) Which factors affect the willingness of acting on received feedback?

B. Data Collection

We collected data from postmortems uploaded by developers on the website Gamasutra¹. As customer collaboration has become more commonplace since the Agile manifesto was published, this could have influenced how companies approach feedback collection [11]. Additionally, we wanted to gain an

¹www.gamasutra.com

TABLE I
INTERVIEW QUESTIONS

#	RQ*	Question Text
Q1	D	What was your role during the project?
Q2	D	What was the project's length span, from conception to release?
Q3	D	How many people were working on the project?
Q4	RQ1	What type of feedback, provided by customers (potential players) and/or QA, do you find most helpful when developing a game?
Q5	RQ2	What methods did you use to collect feedback on work produced during development?
Q6	RQ3	Which of these methods helped you most understand what players expected from your game? How so?
Q7	RQ3	Which of these methods helped you most improve the quality of the game? How so?
Q8	RQ4	At what point during development did you make use of feedback collection methods?
Q9	RQ4	How did you include these methods concretely in your development process (throughout the project)?
Q10	RQ5	How did you prioritize what to act upon when receiving feedback?
Q11	RQ5	Did priorities change throughout development?

*Relates research question to interview question. D represents a demographic question.

insight on postmortems from developers reflecting on modern practices. Therefore, we chose to take a sample consisting of all postmortems appearing in the last five years on Gamasutra search results but still concerning games released after the Agile manifesto. [14]. We also excluded postmortems that do not reflect critically (discussing what went right and wrong in detail) on development practices. The selection of the postmortems followed a search strategy based on these criteria in addition to including only textual postmortems related to video game development and that discuss playtesting, QA processes or usability evaluations.

Additionally, we interviewed a sound designer as well as a QA Lead in a semi-structured style. We have contacted developers and studios interested in participating in the study at the Game Developer's Conference. The companies we targeted housed multi-disciplinary development teams and separated concerns (art, production, QA, programming). In contrast, micro-sized development teams may be forced to take on multiple concerns, and may be undertaking hobbyist projects. The interviews lasted between 30 minutes to an hour, and each subject was interviewed separately with two researchers present. Additionally, we aimed for face-to-face interviews when possible, or at least through video-conference. We chose this structure in order to apply a similar standard among developers, with them answering the same questions to establish a baseline before eliciting data specific to their cases. To preserve the integrity of the interviews, we transcribed and stored them prior to conducting the analysis.

C. Data Analysis

Due to the nature of the data we collected, we chose to conduct a qualitative analysis. The data from the postmortems was coded and categorized to highlight similarities and differences across the project. The issues that went undetected until after release are coded based on Clanton's [10] categorization of them as discussed in section 5: game interface, game mechanics, and gameplay. This is done to see what issues can arise after the end of development, possibly relating to some QA oversight. After reviewing the postmortems, an extra coding has been added to represent *technical issues*. This code

encompasses technical faults that indirectly affect the quality of a game's design, such as network or matchmaking issues. Moreover, feedback collection is separated into two themes, internal feedback methods and external feedback methods. Both include testing and providing feedback, but internal methods concerns people either directly or indirectly involved in the development of the game, while external feedback methods involve customers. Codes to these themes have been added to match what has been identified in the postmortems.

Internal feedback method codes are Publisher QA, Internal QA, Outsourced QA, Internal Playtest, and Prototyping. *Publisher QA* concerns the publisher being involved in the feedback loop, providing their own QA resources in the process. *Internal QA* assumes in-house QA testers and tools are utilized to provide feedback. *Outsourced QA* concerns the delegation of QA tasks to others that are not directly involved in development, providing feedback distinct from a potential customer viewpoint. *Internal Playtest* is playtesting with people either directly or indirectly involved in development, often with a formalized process to record and discuss the feedback produced. *Prototyping* is concerned with creating models to test different game design ideas without utilizing much time or resources.

External feedback methods were differentiated as Customer Playtest, Demo, and Alpha/Beta Build. *Customer Playtest* is concerned with people neither directly or indirectly involved in a game's development being brought in to playtest and offer their views as potential customers. *Demo* offers a limited vertical slice of the game to present its various features to customers in exchange for feedback. An *Alpha/Beta Build* provides customers with early pre-release access to the game during its development.

The information extracted from the literature shaped some of the structured questions we asked the interview participants. However, as we are conducting an exploratory study, we aimed to elicit information directly from practitioners through interviews and analysis of postmortems rather than actively attempt to link information from the literature to the practices we found.

D. Pilot Study

Prior to conducting the interviews, we ran a pilot study with three software engineering students in order to guarantee the quality of our questions. The pilot interviews were conducted through the course of a week, and lasted between 30 to 50 minutes. The students in question were performing research studies at that time, and were therefore familiar with the interview process. Before reading the questions, the students were introduced to the topic of the study and the goal behind the interviews. Each question was read individually before asking the student to explain their understanding of it. If their answer diverged from our intent, we discussed those differences and reworded the question until a consensus was reached.

As a result of the pilot study, minor changes were made such as emphasis on customers and QA on Q4. Q5 was reworked to clarify the wording and shorten it as two of the students found it difficult to follow.

E. Threats to Validity

Predicated on Runeson and Höst's work in case study research [15], we categorized threats to the validity of the study into construct validity, internal validity, external validity, and reliability.

Construct validity is concerned with what the research is designed to investigate compared to what the researchers intend for it to study. Misrepresentation of the results is a possible threat as qualitative data can bear multiple interpretations. We reduce the likelihood of misinterpretation by ensuring the two researchers responsible for the analysis independently examine the data, followed by a comparison of the two analyses. Discrepancies are then discussed, and the original data source can be contacted if clarification is required. The first-degree data from interviews further fights bias and misinterpretations of the industry that can arise from reading second-degree data from postmortems [15].

Internal validity considers the correct identification of cause and effect for studied factors. An interview setting can influence the interviewees in various ways which can cause hidden factors affecting the data collected. For example, the answer of one interviewee affecting the response of others in a group interview situation. This risk is avoided by conducting interviews with participants individually, giving full attention to the single interviewee in that time. Additionally, participants are more inclined to provide complete details when their identity is left anonymous, which is ensured to them at the beginning of every interview. In regards to the postmortems, authors not mentioning some issue or feedback method does not necessarily mean it was not applied in their project, though it can suggest that it was insignificant to the project as a whole from the author's perspective. Fortunately, the postmortems are written by various stakeholders with different perspectives.

External validity is the aspect concerned with how generalizable a study's findings are. Despite the study covering companies from around Europe and the United States, it may not be representative of other companies' experience

in feedback collection. Furthermore, the small amount of interviews could also be a threat to generalizability. However, the study explores processes used by industry leaders that often shape or inspire the workflows of others within the field. This view of the industry is formed from the triangulation of data from both, the postmortems and the interviews.

Reliability is concerned with how consistently the study can be applied independent of the researchers themselves. As mentioned earlier, by independently analyzing and coding the data from both interviews and postmortems, we reduce possible bias stemming from one researcher. While the semi-structured interview questions allow for some leeway when necessary, the predefined research questions are foundational and can be consistently applied irrespective to the game development company in question.

IV. FIRST DEGREE DATA SOURCES

This section presents the sources of first degree data we have collected throughout the duration of this study. We have interviewed, and therefore, have had direct contact with the individuals listed in the following companies. Both candidates were interviewed through video-conferencing.

A. Company A

Company A is a small game development studio based in the United Kingdom. Its employees have previously worked on large commercial titles, and the company is currently working with industry leading publishers. Company A is currently self-publishing its first title.

The candidate we interviewed at Company A is a QA Manager. His responsibilities include testing and leading teams that test games, and has had a decade of experience in the industry. This includes smaller independent titles as well as large titles with development team sizes of more than a thousand people.

B. Company B

Company B is a large game development studio based in the United States, with over 300 employees. Its titles include several commercially successful franchises. Company B is currently working on a sequel to one of its popular franchises.

The candidate we interviewed at Company B is a sound designer with over 10 years of experience. His work includes creating and designing interactive assets for the games, and collaborating with designers and programmers on a daily basis. He has worked at Company B for 10 years. Previously, he worked at Volition on *Saints Row 2*.

V. RESULTS & DISCUSSION

In this section, we will present the results of the study (what we found in the postmortems that was also corroborated in the interviews) and attempt to answer the research questions (see Section III-A).

We collected a total of 131 postmortems throughout the course of the study. 72 of them were discarded as per our search criteria: 15 did not discuss any aspects of QA or customer involvement in any detail, 19 covered games published

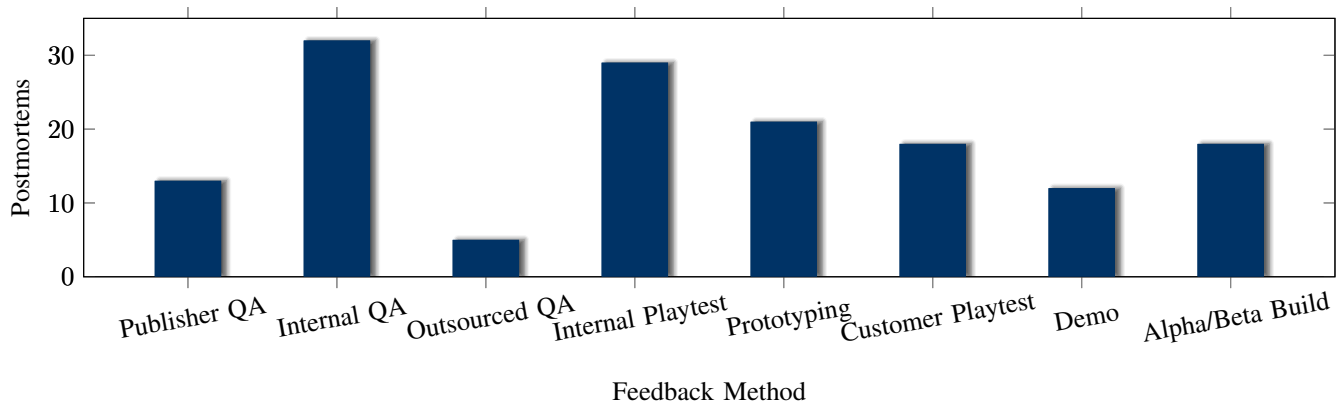


Fig. 1. Distribution of feedback methods across postmortems

before 2001, and 38 were of hobbyist or very small team size (average team size was 1.6). We also conducted two interviews with game development professionals (see Section IV).

Fig. 1 displays the feedback methods identified in the postmortems as well as the frequency of their usage. As shown in the figure, internal QA, internal playtests, and prototyping were all more common than any external feedback method. This may be due to the costs associated with dedicating time and resources to finding customer playtesters and creating something presentable to the public mid-production. The codes were briefly introduced in Section III-C and are covered more extensively in the Sections V-B and V-C.

A. What is useful feedback?

The data we’ve collected allows us to present some key points found repeatedly throughout the postmortems. Most game developers want to learn of potential design issues, in order to “*try and hone the fun factor*”, as early as possible [16]–[18]. In their postmortem of *Guitar Hero*, Daniel Sussman and Greg LoPiccolo mention that “*completed design docs are not always very useful to us, as we’re not yet sure what will be fun*” [19]. One of the reasons cited for this type of feedback is that it freed up development time spent polishing the game instead of refining design elements in the later stages [20]. Furthermore, it establishes the identity of the game which sets a clear picture for the rest of the development team [21].

Editor tools and automated builds were favorably viewed by the developers using them to receive quick feedback on the actions they make [22]–[25]. This saves companies time as developers can focus on other work rather than manually finding bugs [22], [23]. As these bugs are quickly found as they occur, it helps stop them from proliferating into severe issues further into development [22]–[24]. This quick feedback is not only useful to inform the developers about bugs, but it also helps them to directly see the effect of their changes on the product [23], [25].

When asked about what type of feedback they find most useful when developing a game, both interviewees began discussing direct player feedback, despite their different roles

within development. Ted Morris, the executive producer working on *Grey Goo*, said “*We conducted surveys, both written and digital, but the most valuable information was from watching each person play the game for the first time*” [20]. It is a direct way for companies to know what players enjoyed and what they found frustrating [26]. While it can be brutal, this feedback would quickly either validate the design decisions made or reveal its flaws [27], [28].

To answer RQ1 (What constitutes useful feedback resulting from QA testing and customer involvement in video game software development?), while there may not be a consensus on one type of feedback that is most useful, there are definite patterns in the industry on what is valuable feedback. From what was discussed above, it can be condensed into three categories. The first is learning what direction to take with the game early into development. Not doing so has led to costly backtracking with a lot of work thrown away [29], makeshift workarounds [30], and leaving the developers uncertain about the game’s trajectory or purpose [21]. The second is quick responsive feedback for changes made during development. Quicker feedback means faster detection of issues and the ability to assess more design decisions in the same time frame. The third is direct feedback from potential customers. It is a simulation of how the game will be received once it is released. Therefore, it allows designers to correct the trajectory of the product accordingly, and lowers the risk of the game not selling at release. Finally, it is important to mention that the type of feedback found by answering RQ1 will be used to answer RQ3 in later sections.

B. Internal Feedback Methods

Internal QA was the most common form of feedback method, with 32 companies opting to utilize their own tools, tests, and QA personnel. With games being released on various platforms with different hardware, companies employ compatibility testing to ensure a level of consistency among them [29], [31], [32]. Some followed the Continuous Integration development practice, deploying automated test builds either once daily or more [22], [24]. Company B set up a QA department that takes requests from various teams to test

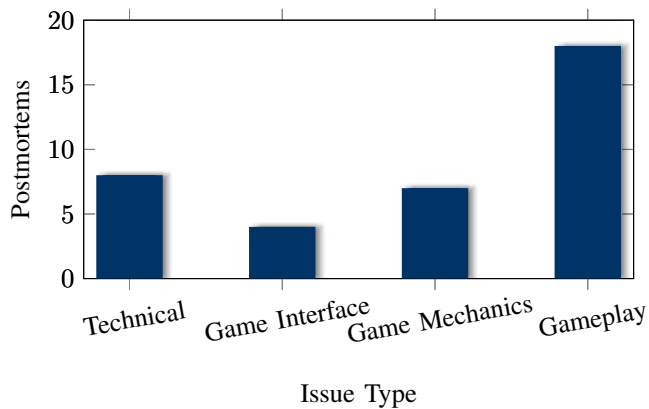


Fig. 2. Distribution of game issues encountered across postmortems

particular mechanics or sections of the game. In one case, testing and developing on other platforms was delayed until the PC version was completed [32]. The performance on other platforms was much lower when tested, and the company had to suddenly dedicate resources in rewriting and optimizing the code to achieve a similar level of quality as the completed PC version. In another case, the co-founder of the company responsible for the game *Warhammer: End Times - Vermintide* regrets not planning for more internal QA during development which would have saved them time, “*With automated tests, QA could have focused on other tasks and we would faster know if a certain map or feature were broken when running the automated tests*” [31]. When the game was released, they received many reports of issues that were tied to specific hardware, which was coded as a technical issue in Fig. 2.

Companies were also willing to spend valuable project resources to develop tools that will eventually end up saving them time [23]–[25], [33]. The tools allow for the simulation of complex scenarios [23], [33]. If built correctly, a tool can be easy and accessible enough to use that even designers with little knowledge of programming can use and assess different game designs [25]. Moreover, they can make repetitive tasks easier to perform for developers, such as debugging and evaluating data [24], [33].

While internal playtests can be a component of internal QA, it appears in 29 of the postmortems and does not necessarily involve other internal QA, making it widespread and distinct enough that it warranted a code of its own. While this playtesting is sometimes conducted by QA personnel [23], [34], it is usually done by everyone involved in the project [35]–[37], or indirectly with their extended friends and family [20], [38]. The interviewee from company A mentioned that they resort to internal playtests in the first quarter of a typical 2-year project in order to have quick feedback loops while discussing the results with other teams. This was done so that the company understood what they can develop with the technology they have. Kevin Wong, the lead designer for *Vanishing Point*, reflects this by saying “*playtesting also allowed us to confidently make deeper changes to how our mechanic worked.*

Early in the project, we realized that everything about how the mechanic worked simply was not working, and revised it from the ground up” [39].

Internal playtesting covers the earlier parts of the game more extensively than the rest because playtests usually begin from the starting point of the game [21], [40], [41]. The lead producer of the game *Civilization V* describes this in his postmortem, “*there ended up being a large disparity between the amount of playtime invested in the first half of the game versus the time spent testing the second half of the game,*” and that “*testers frequently had to start over from scratch, not always able to complete a game before the next build*” [40]. This led to gameplay issues in the later stages of the game to go unspotted by companies, such as “*imbalances that were not revealed*” until after release in the case of *Civilization V* [21], [40], [41]. Over-reliance on internal playtesting was also the source of some gameplay issues [37], [41]. For example, a company creating a puzzle game playtested with its team who were already well-versed in puzzle games, which led to complaints that the game was very difficult in its later stages [37].

Publishers taking part in the QA process was mentioned by 13 postmortems, with varying results. In many cases, publishers provided the companies with vital feedback and QA resources, either through conducting external playtests [42], [43], internal playtests [22], [38], or identifying problems in specific features [40], [44]. It seems that publishers begin testing once a specific milestone is reached that was agreed upon previously [22], [38]. In one case, the publisher gave damaging feedback that harmed the game design by reducing the difficulty greatly and introducing gameplay issues [30]. The co-designer believes the company should have defended the game’s design decisions more adamantly. In two other cases, the publisher feedback was either insufficient or less useful than internal QA [41], [44].

Outsourced QA was only used in 5 of the companies from the postmortems, but had positive results from all of them. It differs from internal QA by assigning QA responsibilities to people knowledgeable about game development but are not involved in the project. One company discusses how they outsourced some of their QA to an external company, which in turn “*wrote mock reviews and in-depth assessments*” [44]. This helped improve the game’s design and features according to the directors [44]. Company A utilized outsourced QA as well, providing a build of the game to a company to get feedback on. Two other professional companies had QA outsourced to them, working on identifying bugs and test coverage [45], [46]. Game development and design students were used in the remaining two cases, providing the companies with feedback from playtesting [33], [47].

Prototyping was used by 21 companies and has been praised in all mentions of it. While prototypes can be a model of a playable game as proof of concept, they usually test various different features and mechanics on their own [19], [37], [39]. Developers for *The Sims 2* said that prototyping allowed them to “*resolve look and feel issues, to help understand*

the key emotional connection, and most importantly, to test out the new gameplay concepts” [48]. In one case, a lead-designer regrets not prototyping a core mechanic of their game, which caused limitations in the design [39]. The project leader for the game *Stellaris* says, “There are no excuses. Even if you can’t prototype everything, you can at least isolate some parts of a system that you can try out” [21]. Developers for *Half-Life 2* write the following as a lesson at the end of their postmortem: “Don’t design using theoretical mechanics. Validate designs first using prototypes” [35]. There was also a case where a company misused this method by creating temporary prototypes and placing them in the game, intending to create a complete system later. However, they waited too long to replace them, creating many dependencies that “ripping them out and rewriting them threatened to create wide repercussions for other departments such as level and overall game design” [49].

C. External Feedback Methods

When soliciting feedback directly from customers, our data suggests that developers make use of three main approaches: customer playtests, demos and Alpha/Beta builds.

Customer playtests are present throughout the postmortems but the terminology is not always the same. Terms such as “Usability Testing” [42], [43], “Focus Testing” [19], [38], and “Gameplay Testing” [50] were encountered. In certain cases, the terms were also used interchangeably [51]. The descriptions of the terminology fit within our “Customer Playtest” code, therefore any mention of them was classified under it. 18 out of 59 considered postmortems presented some form of “Customer Playtest”. In contrast with an Internal Playtest, the feedback was elicited from potential customers who have never played the game and are not involved with the development process. Our interview candidate from Company B mentioned that in the case of a sequel to a previous game, Company B looks for players who are familiar with the previous titles. It is critical that when pooling candidates “it’s always going to be people who are potential customers”. In the case of *Half-Life 2*, players were brought inside to test a specific feature or mechanic which designers were unsure of. Their gameplay footage was recorded and used “as a way to settle design arguments” [35].

This method allows developers to receive fresh perspectives on the current design assumptions. The developers of *The Sims 2* called the process of bringing in people to play the game only once “Kleenex Testing” [48]. However, in many cases this method is problematic as it presents a security risk to the companies, which is often mitigated by making players sign Non-Disclosure Agreements [40], [52]. The QA lead from Company A mentioned that they did “as much as you possibly can to illustrate the security concern that you have” but that in one case a candidate “just pulled out this massive paparazzi camera and started taking photographs”. Because of these security concerns, Company A opts to “not bring people in until a bit later”. This likely contributes to the fact that internal

playtests seem to be more prevalent in our dataset as shown in fig. 1.

Demos present a vertical slice of the game that give players an idea of a typical gameplay iteration. As they are publicly available, they must be more polished than a simple level used as part of a playtest. They present a clear depiction of what the final product can look like. Out of the 59 postmortems we considered, 12 made use of demos in their development process. Therefore the feedback and reception generated by players can indicate whether customers will be interested and whether or not the core gameplay elements work in practice [53]. However, creating a demo takes away resources and time that are already being used to develop the game [42]. We have found that demos often satisfy both development and promotional concerns [34]. In the case of *Kingdoms of Amalur: Reckoning* or *Star Wars: Knights of the Old Republic*, a polished demo was prepared for the E3 conference to both unveil the game to the general public but also to provide players with the ability to play a vertical slice of the game themselves [42], [51].

Many of the elements of a demo are initially not ready for public release, which forces developers to “bump it all up to shippable quality long before it’s supposed to be shippable quality” [42]. Dedicating resources to do this is sometimes not possible and therefore demos are outsourced to other teams or studios [34], [42]. In some cases, previously assumed design decisions can also change as a result of the development of the demo where new ideas are found through iteration [38]. Some developers have reported that producing a demo generates some constraints, such as alienating the consumer base. The developers of *Call of Duty 4: Modern Warfare* mention that they did not want to release any content that players were not familiar with, in order to not give away too much of the final product [54]. We found several cases where the same concern was expressed. Developers want to communicate their vision effectively through a demo without giving away story elements or novelty before release [55]. This resulted in players expressing disappointment, and the developers conclude by saying “we should have realized that a pre-release demo would be likely to hurt us rather than help us” [54]. This seems to indicate that demos serve a more promotional goal than other feedback methods.

Before delving into the last method, it is important to note that the terms “Alpha” or “Beta” do not represent traditional software versioning terminology in this case. The terms are used interchangeably throughout the postmortems and definitions seem to be loose. A proper way to characterize them would be an early build showcased to a targeted group of players for the purposes of collecting some form of feedback on gameplay elements. Out of 59 postmortems, 18 fulfilled those criteria.

In the case of multiplayer games, Open Betas are used extensively in order to stress test the limitations of networks and whether or not matchmaking features work properly [31], [36]. While reflecting on a popular multiplayer franchise’s approach to Open Beta prior to release, the QA Lead from

Company A mentioned that *“the feedback that they’re looking for in terms of their last week before launch is analytics based. What they’re looking for is how long do people play the game in a single session? Who comes back after the first 24 hours of having the game installed? What classes do people gravitate towards? Why do they do that? What maps do they vote to keep?”*. Having direct access to the data from player machines is more valuable than *“the four thousand tweets every minute of people coming in saying every class is overpowered, every class is underpowered, that becomes more noise than signal”*.

Furthermore, the targets are not the same depending on the pool of players selected. In the case of a Closed Beta (invite-only), candidates *“are often people that really care about the product, they care about the game”* and *“they’re people that will give you quite detailed feedback”*. In contrast, an Open Beta is *“a lot more brutal typically because all of a sudden its people that have no investiture in your game”* and the experience must be somewhat polished because if players are *“not having fun within the first three minutes they’re out”*. We also encountered a more recent phenomenon in the form of an early access model. Certain developers used paid Alphas (Offering an early version of the game prior to final release) to iterate alongside their customer base, which also allows them to fund the game through crowdfunding [52], [56].

D. Effectiveness of Feedback Methods

As mentioned previously in Section V-A, answering RQ3 (How effective are these methods in providing useful feedback, according to those involved in the development of video games?) meant that we first needed to understand what practitioners expected from the feedback methods found in Sections V-B and V-C. The information regarding the methods themselves gives us a certain outlook on which kind of feedback it targets (tools determining responses to changes, playtests gathering fresh customer perspectives, etc). However, the answers differ and we found no clear determinations on whether one method was more effective than another at gathering feedback (e.g. differing opinions on the use of demos). More data that is not as accessible in the postmortems would be required to make an accurate determination, such as publisher relationship, level of commitment to method, and what project phase it is used in (see V-E). Consequently, it did not seem like our dataset could answer the question properly without clear quantitative results to supplement it. Collecting types of issues (see Figure 2) could have given us an insight on whether certain methods were failing to detect them or caused them. In the end, we determined that this was not enough and the information was too limited as developers could have failed to mention certain issues and we had no concrete way to link issues to a specific method.

E. Use of Methods throughout the Development Process

We had varying results into how companies utilize these feedback methods in their development process. Company A does not begin using external methods until active production

started, around 6-8 months into a 2-year project. In addition to security concerns, the company did not test the game externally as they did not have anything playable to present. In the case of company B, external testing begins in early pre-production, bringing external playtesters to evaluate very small systems that only expose the core mechanics to be tested. There are not many details about what development processes the companies in the postmortems use, but some cite iterating between feedback and development constantly [36], [51], [57]. Company A follows a similar process, iterating over 2 week sprints where the teams meet to playtest and discuss at the end of the sprint. Company B allows employees to request feedback whenever necessary by either directly tuning in to external playtest live video streams or through their research development department. Some companies also created ‘strike teams’ that involve employees from multiple disciplines [23], [24], [35], [40]. Their role included questioning design decisions, discussing feedback, and identifying issues when specific crashes occur.

Prototyping took place at the beginning of pre-production in many cases [31], [48], [53], and through multiple rapid iterations [18], [19]. As for Alpha/Beta builds, company A and two other cases organized them a few months before release [31], [36], but there was a postmortem that described it extending to 14 months prior to release [52]. A typical timing for an Alpha/Beta build cannot be drawn from the data as most postmortems did not discuss when they began.

As for demos when they are used in pre-release development, they usually coincide with events where a large amount of players will be available to play through it. We’ve seen a consistent pattern throughout the postmortems of demos being prepared for conferences and trade shows [42], [54], [55]. In addition, the demos are also made available through digital stores on the respective hardware platforms that developers want to reach. This means that demos are not created with the project’s own schedule in mind, instead relying on fixed events where they will generate the most impact and feedback from players.

F. Changing Priorities during Development

While we have determined what constituted useful feedback during development in Section V-A, there are nuances as to what will be considered depending on the project’s progress. For instance, the bug count will increase throughout development and as deadlines approach developers sometimes found it necessary to *“mark ‘will not fix’ or ‘as designed’ on as many bugs as [they] could”* in order to meet the shipping date [22]. Design decisions made at this stage may be influenced by whether it is worth implementing in relation to other pressing issues. A developer for *Spider-Man* states that in the later stages *“it’s a gray area between a ‘bug’ and a feature that ‘really has to be there.’ ”* [22]. Both our interview candidates expressed that priorities change throughout development and that when nearing deadlines *“there’s just certain things that you don’t have enough time for sadly”*. We also found certain cases where developers created a process that *“set priorities*

and assignments for each tester” on a weekly basis, to adapt the QA process to changing requirements depending on the schedule [58].

Since certain elements have to be completed a few months before launch, the interview candidate from Company B suggested that at some point whoever is working on those elements “can’t touch anything else after a while” and therefore feedback is likely to be ignored. Similarly, he made the point that core elements are usually not altered in a significant manner. More minor elements would be changed “to fit into the system better” rather than changing the major element itself. This seemed to be different in every team, as sound designers were more protective of their own work whereas developers working on the narrative team would do the same on narrative elements of the game.

Additionally, the candidate from Company A confirmed that sometimes a developer will find or tinker with a feature that is deemed important enough to be incorporated into the game. As a result, priorities change suddenly and “you have to go around and think how does this affect the design, how does this affect the flow of these maps?”. This process requires a balance between incorporating new elements that can bring value to customers or prioritizing already established design decisions instead. We have found there is no established process to deal with these impromptu events and the interview candidate from Company A reiterated that “It’s about trying to find a balance and that balance is different in every company”.

Finally, to answer RQ5 (Which factors affect the willingness of acting on received feedback?), we find that sudden new features brought to the attention of the development team can result in changing priorities that affect established design decisions that may already bring value to the customer. Whether there is strong ownership of core features can also affect whether or not feedback is considered or prioritized differently. Finally, strict deadlines and long hours of overtime had an effect on the prioritization of issues related to collected feedback [22]. There is already some research regarding crunch time in the gaming industry [14], where developers will work overtime during the last few months of development. This phenomenon coupled with strict deadlines affects whether or not developers will fix minor bugs or change design elements depending on how much time is left.

CONCLUSION

The game development industry relies mostly on feedback related to game design decisions during the pre-release development stage. This can take the form of player feedback, or it is generated through the experience of the team members. Internal feedback collection methods rely on the skills and diligence of those directly or indirectly involved with development, and external methods make use of customer involvement through various means. The methods in question are used at different times during development, and companies vary their implementation. Some may opt to involve customers as soon as possible, while others do so in the later stages when a polished product can be shown. While these methods successfully

extract the feedback deemed most useful by developers, it will not always be taken into account or may be re-prioritized if the project has reached its later stages or if the feedback concerns core elements that represent a key vision. Most developers are aware of the advantages and disadvantages of using one or another method, but there is no consensus on which methods are most effective. We see further work on the topic in the form of a mixed-method study taking a closer look at the artifacts produced by each feedback method and their consequences on the work product.

ACKNOWLEDGEMENTS

The authors would like to thank the interview candidates for their useful insights and their time. We would also like to thank our supervisor Jan-Philipp Steghöfer for his diligent feedback.

REFERENCES

- [1] J. Blow, “Game development: Harder than you think,” *Queue*, vol. 1, no. 10, p. 28, 2004.
- [2] J. Schell, *The Art of Game Design: A book of lenses*. AK Peters/CRC Press, 2014.
- [3] D. King, M. Nordin, and S. Posthuma, “But is it fun?” Apr 2018. [Online]. Available: <http://www.es.mdh.se/ficst2018/keynotes/>
- [4] N. Lawrence, “Why most beta tests are really just demos,” Nov 2016. [Online]. Available: <https://www.ign.com/articles/2016/11/21/why-most-beta-tests-are-really-just-demos>
- [5] M. A. Federoff, “Heuristics and usability guidelines for the creation and evaluation of fun in video games,” Ph.D. dissertation, Citeseer, 2002.
- [6] D. Callele, E. Neufeld, and K. Schneider, “Requirements engineering and the creative process in the video game industry,” in *13th IEEE International Conference on Requirements Engineering (RE’05)*. IEEE, 2005, pp. 240–250.
- [7] E. S. Association, “2018 sales, demographic and usage data-essential facts about the computer and video game industry.” *The Entertainment Software Association*, 2018.
- [8] 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*. ACM, 2014, pp. 1–11.
- [9] L. Pasarella, F. Palomba, M. Di Penta, and A. Bacchelli, “How is video game development different from software development in open source?” in *2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR)*. IEEE, 2018, pp. 392–402.
- [10] C. Clanton, “An interpreted demonstration of computer game design,” in *CHI 98 conference summary on Human factors in computing systems*. ACM, 1998, pp. 1–2.
- [11] R. Al-Azawi, A. Ayesh, and M. A. Obaidy, “Towards agent-based agile approach for game development methodology,” in *2014 World Congress on Computer Applications and Information Systems (WCCAIS)*. IEEE, 2014, pp. 1–6.
- [12] S. Laitinen, “Do usability expert evaluation and test provide novel and useful data for game development?” *Journal of usability studies*, vol. 1, no. 2, pp. 64–75, 2006.
- [13] J. Nielsen, “Usability engineering,” *Fremont, California: Morgan*, 1993.
- [14] 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*. IEEE Press, 2017, pp. 43–52.
- [15] P. Runeson and M. Höst, “Guidelines for conducting and reporting case study research in software engineering,” *Empirical software engineering*, vol. 14, no. 2, p. 131, 2009.
- [16] B. Wardell, “Postmortem: Ironclad/stardock’s sins of a solar empire,” 2008, [Accessed 20-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/feature/132039/>

- [17] C. Esmurdoc, "Postmortem: Double fine's brutal legend," 2015, [Accessed 20-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/feature/132696/>
- [18] T. Mahler, "Postmortem: Moon studios' heartfelt ori and the blind forest," 2015, [Accessed 20-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/242530/>
- [19] D. Sussman and G. LoPiccolo, "Classic postmortem: Guitar hero," 2015, [Accessed 20-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/258820/>
- [20] T. Morris, "Postmortem: Petroglyph's grey goo - getting back to the roots of rts," 2015, [Accessed 20-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/237666/>
- [21] H. Fähræus, "Postmortem: Paradox development studio's stellaris," 2016, [Accessed 20-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/274018/>
- [22] J. Fristrom, "Postmortem: Treyarch's 2002 hit, spider-man," 2002, [Accessed 21-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/feature/194852/>
- [23] Y. Mallat, "The making of prince of persia: The sands of time," 2015, [Accessed 21-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/258603/>
- [24] C. Esmurdoc, "Classic postmortem: Double fine's psychonauts," 2015, [Accessed 21-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/251220/>
- [25] B. Spasov, "Starting from scratch: Haemimont games' tropico 5 postmortem," 2015, [Accessed 21-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/237667/>
- [26] K. Stavola, "Perfecting the recipe for mobile success: Restaurant story 2 post-mortem," 2015, [Accessed 23-May-2019]. [Online]. Available: <https://www.gamasutra.com/blogs/KateStavola/20150507/242865/>
- [27] S. Thompson, T. Walsh, E. Evans, and D. Evans, "Postmortem: Pinball-rpg hybrid rollers of the realm," 2014, [Accessed 23-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/feature/233340/>
- [28] M. Wakeley, "Going mobile the right way: A crack attack postmortem," 2015, [Accessed 24-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/243406/>
- [29] G. Roberts, "Postmortem: E-line media and upper one games' never alone," 2015, [Accessed 24-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/236049/>
- [30] P. Howell, "Postmortem: The chinese room's amnesia: A machine for pigs," 2014, [Accessed 24-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/feature/218372/>
- [31] M. Wahlund, "Postmortem: Fatshark's warhammer: End times - vermintide," 2016, [Accessed 21-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/284228/>
- [32] D. Ab and J. Roth, "Postmortem: Mimimi's shadow tactics: Blades of the shogun," 2017, [Accessed 27-May-2019]. [Online]. Available: <https://gamasutra.com/view/news/310894/>
- [33] P. Tejada, "Postmortem: Chasing carrot's pressure overdrive," 2018, [Accessed 28-May-2019]. [Online]. Available: <https://www.gamasutra.com/blogs/PatrickTejada/20180322/315717/>
- [34] M. de Plater, "Postmortem: Monolith productions' middle-earth: Shadow of mordor," 2015, [Accessed 24-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/234421/>
- [35] B. Jacobson and D. Speyer, "Classic postmortem: The making of half-life 2," 2015, [Accessed 20-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/259479/>
- [36] W. Wade and C. Sonny, "Postmortem - sony santa monica's god of war: Ascension," 2013, [Accessed 23-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/feature/193695/>
- [37] B. Dillon, "Double fine's 'heartfelt and personal' hack 'n' slash: A postmortem," 2015, [Accessed 30-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/237644/>
- [38] A. Finley, "Postmortem: 2k boston/2k australia's bioshock," 2008, [Accessed 20-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/feature/132168/>
- [39] K. Wong, "Vanishing point postmortem," 2015, [Accessed 30-May-2019]. [Online]. Available: <https://www.gamasutra.com/blogs/KevinWong/20150520/243620/>
- [40] D. Shirk, "Classic postmortem: Firaxis' civilization v," 2017, [Accessed 31-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/306040/>
- [41] L. Hyvärinen and J. Kinnunen, "Postmortem: Frozenbyte's trine," 2010, [Accessed 31-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/feature/134198/>
- [42] M. Fridley, "Postmortem: Kingdoms of amalur: Reckoning," 2013, [Accessed 20-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/feature/197269/>
- [43] S. McCabe, "Ratchet & clank (2016) post-mortem," 2016, [Accessed 20-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/272694/>
- [44] J. Klose and T. Lange, "Postmortem: Deck13 interactive's lords of the fallen," 2015, [Accessed 1-June-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/236275/>
- [45] T. Oster, "Postmortem: Overhaul games' baldur's gate: Enhanced edition," 2013, [Accessed 1-June-2019]. [Online]. Available: <https://www.gamasutra.com/view/feature/190432/>
- [46] S. Casen, "Development post-mortem of project lake ridden," 2018, [Accessed 1-June-2019]. [Online]. Available: <https://www.gamasutra.com/blogs/SaraCasen/20180924/326638/>
- [47] D. Jones, "Postmortem: Building the turing test around a secret mechanic," 2017, [Accessed 1-June-2019]. [Online]. Available: <https://gamasutra.com/view/news/308654/>
- [48] L. Bradshaw, "Classic postmortem: How maxis avoided sequel-itis on the sims 2," 2017, [Accessed 27-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/305424/>
- [49] C. Rhinehart and R. Jackson, "Into the asylum: A postmortem of human head studios' lost within," 2015, [Accessed 1-June-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/252242/>
- [50] R. Rouse, III, "Postmortem: The game design of surreal's the suffering," 2004, [Accessed 20-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/feature/130523/>
- [51] C. Hudson, R. Muzyka, and J. Ohlen, "Classic postmortem: Bioware's star wars: Knights of the old republic," 2017, [Accessed 20-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/299385/>
- [52] S. Johnson, "Postmortem: Offworld trading company's early access campaign," 2016, [Accessed 24-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/274240/>
- [53] C. Harvey, "Postmortem: Drinkbox studios' guacamelee!" 2013, [Accessed 21-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/feature/200658/>
- [54] Z. Rieke and M. Boon, "The making of call of duty 4: Modern warfare," 2015, [Accessed 24-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/258315/>
- [55] A. Chmielarz, "Classic postmortem: People can fly's bulletstorm," 2016, [Accessed 24-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/266261/>
- [56] H. Taso and M. Hartman, "Postmortem: Muse games' guns of icarus alliance," 2017, [Accessed 23-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/298271/>
- [57] B. Mitsoda, "Postmortem: Doublebear's dead state - controlling scope in an rpg," 2015, [Accessed 1-June-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/236847/>
- [58] K. Hallahan, "Remaking gabriel knight: A 20th-anniversary postmortem," 2015, [Accessed 30-May-2019]. [Online]. Available: <https://www.gamasutra.com/view/news/240049/>