Analyzing the IMGD Playtesting Requirement at WPI

An Interactive Qualifying Project submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
In partial fulfillment of the requirements for the degrees of
Bachelor of Science and Bachelor of Art

By

Owen Lacey

Michael Gatti

Evan Arenburg

February 28th, 2023

Advisor: Prof. Walt Yarbrough

WORCESTER POLYTECHNIC INSTITUTE

Abstract

The Playtesting requirement at WPI as it currently stands is outdated, and does not address the problem that it was created for. It was intended to give MQP and graduate projects access to early testing. However, most projects do not get tested by the general student body and instead rely upon internal testing or friends. This proposal researches the intended effect of the playtesting requirement and the way it currently manifests and recommends improvements to the system.

Table of Contents

Introduction

Bethesda's *The Elder Scrolls IV: Oblivion* is well known for being one of the best selling games of its time, and for being one of the best RPGs ever created. When released, however, it came with multiple glaring flaws, such as serious problems in the voice acting. Some voice lines have the wrong inflection, too much emotion, or too little, and at one point, the voice actor can be heard in game saying, "Let me do that one again" (Shinkle, 2021). While it might be tempting to blame it all on the method of recording the voice lines—giving the actors a list of lines in alphabetical order and having them read through it—the problem should really be traced to a lack of quality assurance, or testing, of the game.

Quality assurance (QA) as it is used today was first seen during the Industrial Revolution, when it shifted from being the job of the crafter or artisan, to being a separate group of workers in a factory. Over the years, the process was refined to the modern QA departments we see today. However, it is still a somewhat unknown process to most people, even those developing the products that get tested. Developers will simply send their product off to the QA department, and get a list of problems back, with little to no input or insight on the process.

At WPI, every class in the Interactive Media and Game Development (IMGD) field requires that students complete two playtesting credits per term. This process was put in place by the leaders of IMGD to promote early testing of Major Qualifying Projects (MQP) and graduate projects. Testing these projects early is critical for creating a polished final project. Missing out on early tests can leave large flaws in the project which aren't noticed until too late in the development process. Needing to fix large flaws late can force the scope of the project to become

smaller, or affect the quality of the overall product. The requirement was put in place to help projects get tested early by many students in order to provide an easier development process for MQP and graduate teams. Unfortunately, our research leads us to believe that 'playtesting' has largely failed to provide an easier development process, with few exceptions such as Professor Moriarty's classes, and the yearly Alphafest run by IMGD.

This paper will start with a background on testing, both for games and other software products. Then it will move on to playtesting games, and more specifically, the playtesting methods in the industry and at Worcester Polytechnic Institute. The research done on industry testing methods will be compared to Worcester Polytechnic Institute's playtesting practices in order to give a critique of the playtesting practices performed at WPI in an effort to improve upon them. This will be done first with a report summarizing the findings of the current best playtesting practices actively performed in the industry for testing games. Following with a report of recommendations based upon the findings from the industry report and tested software in order to illustrate potential changes to Worcester Polytechnic Institute's playtesting practices. This will give recommendations based on the findings from our research and testing which we hope can be used to improve upon said present playtesting practices at Worcester Polytechnic Institute. Any testing methods used will be described within this report as well as listing any software in which these methods were tested on.

Background

History of Testing

During the middle ages in Europe, artisan guilds began to develop standards for the quality of the service they provided. They would stamp, or in some way mark the product they were creating with a unique symbol, intended to show that the product was verified to come from their workers. Over time, these began to be seen as a stamp of quality, a symbol that a particular product was better or more reliable than others (ASQ, n.d.). During the Industrial Revolution, attempts to increase productivity led to drops in the quality of the products. To combat this, the individual process of quality control seen in the middle ages was replaced with dedicated departments in factories checking products as they were made (Kolb & Hoover, 2012, p7). This was the start of the quality assurance process, leading to the QA departments seen in most companies today.

After the destruction of World War II, Japanese manufacturing quality fell apart, leading to Japanese products being seen as substandard and inferior . William Deming, an American physicist and business production expert, was brought in to help with the problem. His methods had been rejected by American manufacturers, who preferred the American style inspection system, and he had developed a method which placed the consumer as the highest priority of every worker involved in the manufacturing process, using charts and checklists to ensure that everything matched the required specifications (Kolb & Hoover, p9). Kold and Hoover conclude that this process allowed Japanese manufacturing to become known as one of the highest quality manufacturing systems in the world.

Purposes of Testing

In an ideal world, testing would be made nearly redundant by improving the manufacturing processes so that they could not produce the errors that testing is meant to find. However, wear and tear, unexpected circumstances, and human error are always present in the process of creating a product, and cannot be completely eliminated without prohibitive expense. The process of testing a product finds flaws that production created, and either rejects the product, or sends it back for further polish.

With a physical product like airplane parts, most flawed products are rejected and their materials recycled; An example of this being when the A220-100/Bombardier prototype's fuselage was cut into pieces and used for training (Loh, 2021)**.** This requires the manufacturer to start over from scratch, and often costs time, which can be lessened by testing products as early as possible. In software development, the product can usually be fixed without completely scrapping it, making early testing even more helpful for time saving.

General Testing versus Playtesting for Games

Testing most products is a process that can only be done once the product is fully completed. You can test individual components of larger, composite, creations, such as ensuring that the engines work on an airplane before attaching them to the wings, but trying to test an airplane fuselage before attaching the wings will not reveal flaws with the flight characteristics. You can check for errors in construction, but design work can only be determined to be correct with a finished product. Games, on the other hand, can and should be tested as early as possible, even before a playable build can be made. A simple text version of a game, missing all of the

graphics and behind-the-scenes automation, can still reveal whether a concept is fun to play, or dreadfully unengaging.

Testing games also requires more people than testing most other products. An airplane has only one proper configuration that will fly properly, and the same group of testers can review every available aircraft to ensure that they all meet the specifications. Most products can be tested with objective criteria and either accepted, or sent back for fixing or recycling. Games must be tested based on the amount of engagement they provide, and the ease of access for players, both of which are subjective criteria. A single playtester will only give developers one data point, which is inadequate for a proper test.

Playtesting in the Industry

In the industry, there are various forms of "testing" before the game gets to be released to stores and online marketplaces. There are "internal design review" (Fullerton, 2019 p277) sessions where the designer and team play through the game and test/question features. As described by Tracy Fullerton, "playtesting is something that the designer performs throughout the design process" . It has many forms, and is heavily dependent on how large the studio itself is. We conducted an interview with Alex Bettios who is a solo dev that is working on a MOBA (Multiplayer Online Battle Arena) game called Orion. As a solo dev there are obvious obstacles that come with not having the extra hands and mind that would come with working at a medium to large sized game development studio. He cited in particular that he no longer categorizes the tests because you test with a goal in mind but sometimes they blur the line. And follows that up with essentially conducting the previously mentioned internal design reviews/ other internal tests whenever he reaches a point where there is a mechanic that needs to be tested. This also leaves

him in a constant cycle between creating features, establishing if they work with the existing mechanics, and then putting them into the main branch. He is able to do this through a series of self written python scripts that enable automated testing, as well as heavily utilizing the Unreal Engine debugger which has been cited as being “amazing”.

Playtesting at WPI

The playtesting requirement at WPI was put in place to help get Major Qualifying Projects (MQP) and graduate projects tested earlier than the late phases of development. Professor Brian Moriarty claimed that the intention was for all projects to get tested early and often. He said that he often would give his classes an assignment to test one of the projects developed by an MQP that he was advising as a way to fulfill the playtesting requirement. He would also add playtesting opportunities to his classes, in which students would test each other's work throughout the term (Prof. Brian Moriarty, personal communication, Oct 10, 2022). However, Moriarty’s classes were somewhat unique in including the playtesting in the class. Most IMGD classes do not have playtesting built into the course, and even fewer offer MQP projects for testing. Some courses do not even mention the requirement in the syllabus for the class.

Proper critique is another piece of the process which is limited or nonexistent in most IMGD classes. Art classes sometimes spend a single class period describing the proper forms for critique (Prof. Adryen Gonzlaez, personal communication, Sep 2022), but most classes do not mention it at all, leading to a lack of proper critique skills in students. This lack of skills often manifests as a reluctance to give proper feedback, instead preferring to only praise it, or just ‘smile and nod’. While Professor Gonzalez has found success in teaching students to give proper

critique in her classes, that success has not traveled outside of the art world into the technical, design, production, and writing fields.

Methods

This project will recommend changes to the Playtesting requirement and process at WPI based on feedback from professors, current MQP project members, and research into the playtesting process in the industry. We started by interviewing professors who use the playtesting system about how effective they believe it is. We also spoke to the professors who helped put the system in place in order to better understand its original purpose. Once we gathered information on the current playtesting requirement and the methods used in the game industry, we worked with an MQP team to provide improved testing for their game. When that was complete, we provided a recommendation for improvements to the current Playtesting requirement.

Evaluate Current System

In order to evaluate the current system for playtesting at WPI, we interviewed some of the professors who helped put it into place. The three that we know of are Brian Moriarty, Keith Zizza, and Gillian Smith. Unfortunately, we were unable to talk to Professor Zizza or Professor Smith. However, Professor Moriarty was able to give us information on why the playtesting system was implemented, and how it was supposed to work. It was originally designed to create opportunities for MQP teams and graduate students to test their projects as early as possible, and

to provide continuous testing throughout the year. It also was meant to expose students to the critique and testing process as early as possible. Moriarty has seen some success in his own classes, requiring students to test each other's projects in order to gain experience, and sometimes providing MQP projects that he was currently advising to his class for additional testing credit.

While Prof. Moriarty has seen good results with his method of testing MQP projects, most other professors and MQP groups have not seen the same success. A few of the ideas he suggested to improve the system are a registry of all MQP and graduate projects that need testing, running an Alphafest-like event every term, and requiring professors to give their classes MQP projects for testing each term.

While there are very few MQPs available for testing in A term, there are still graduate projects that could be distributed. Additionally, Prof. Moriarty suggested the idea of a playtesting and critique focused 1000 level class to be run each A term, to fill in for the lack of MQP projects to test. On top of an individual class, he said that all 1000 levels should also have playtesting built-in more than they currently do. At the moment, 1002 doesn't have much in the way of playtesting in the class (Moriarty, Personal Communication, 2022).

Research Industry Playtesting

In addition to speaking to professors, we plan to interview a few industry testing professionals to gain insight into the professional game testing process. We plan to ask them the questions in Appendix B, and research the methods they describe for applicability to the Playtesting process at WPI.

We interviewed Professor Walt Yarbrough and asked him about his experience in the industry. Walt was Live Producer, Senior Producer and Executive Producer of Dark Age of

Camelot from 2000-2006, working closely with an internal QA team. Walt was also Group Producer of DAoC, Ultima Online, and The Sims online from 2006-2007, working with 3 internal QA teams. In addition, Walt was the Senior Producer at Turbine for Lord of the Rings Online, working with an internal QA team and was Director of QA at QuickHit for Quickhit Football from early prototype to soft launch. Although this industry information is a little out of date with the most recent practices, it is still important to see how playtesting has developed in the industry in order to potentially develop the playtesting requirements here at WPI. We first asked him what the standard terms for different testing types were in the industry. He told us that at Turbine, each individual contributor had to present work to everybody on the team. Associate producers took notes the entire time and assigned issues to people on the team. The Senior Producer or The Executive Producer would give it a thumbs up or thumbs down, nothing would get into the product without it. Every commercial MMO Walt worked on had a large brute force QA team reviewing bug submission from the active customer base. Over 99% of bugs submitted were closed as "could not reproduce". The same team was responsible for testing new features on internal servers before they went to the test server. None of these teams had the authority to prevent the build from being published. None of the commercial products Walt worked on had Playtesting as we define it. The quality of content was typically measured with user feedback by a dedicated community relations team. However, the community team had very little power. We then asked him about regression testing at Mythic, where they had a huge number of confirmed bugs in the tracker, about 1500 confirmed bugs, and instead of fixing them, they would recommend that all the bugs were closed to see if they would reappear. The good thing about having a huge number of bugs in the tracker is that the QA team could spend a healthy amount of time checking the bugs and have a good ecosystem for knowing what was wrong with the

product. It led to good patch notes as bugs were fixed since they had to be very specific about what was being fixed.

For the information on the smaller questions, we were given that from 2000-2006 there were no functional test plans but in 2007-2008, Turbine had begun using functional test plans. For build verification, Turbine had a continuous build system where a build was automatically created with every developer check in to the main repository. If a build failed to build correctly, an email alert was sent to the entire team naming the developer and check in that caused the build to fail. Quickhit delivered a candidate build once a day with a BVT ran by 10am and a functional test plan done from 10am to 5pm everyday. From 2000 to 2008, none of the testers had certifications for testing. For teams reporting to a member of the development team, at Turbine the testing team was a separate and equal department, and at Quickhit, they had the same structure until Walt quit working there. After that, testing there was reported to development and it went sideways, with the daily builds coming late and testing going late into the night. We were also told that the QA team controls the severity of the bugs and the development team controls the priority of the bugs. For both severity and priority, many game studios use scales from 0 to 9 with 0 priority bugs being 'Must Fix' and 0 severity bugs being 'System Crash' bugs (0 is the highest classification). All of the statused are accurate as seen in Walt's commercial work: Open, Confirmed, Closed, Closed Confirmed, Will Not Fix, As Designed, and Could Not Reproduce. The QA team retested closed and open bugs to see if they could reproduce them.

Interview MQP Teams

We interviewed two IMGD MQP teams currently making games, or game-adjacent projects this year about their interactions with the testing process. Specifically, we wanted to

know if the current testing methods are providing enough feedback, useful feedback, and specific feedback. We spoke to two MQP teams, one of which had a testable prototype, and one which didn't. The team without the prototype was able to provide some information on how they had tested parts of their game up to this point.

The team with a testable build of a single level in an RTS game provided us with feedback on how their/WPI's current playtesting system has helped their development process. The summary from the responses made it clear that a combination of specific questions along with overall experience was the best way to approach asking the tester, along with ensuring that the testers aren't able to provide short answers because of confusing or non-useful questions. Overall the MQP groups were very helpful when providing feedback on how the current system is running and also are going to be the testing grounds for our implementation going forward.

Work with Teams to Provide Improved Testing

If we find teams to work with, we will work with them to improve the quality of the feedback they are getting to be more in line with their goals.If the problem is a lack of feedback overall, we will try to find a way to get their project more testing from other IMGD students. If there is a lack of quality feedback, we will work with the team to improve their methods and try to get better feedback for them. If possible, we would create different categories for how teams should approach the feedback process. Once we have devised an improved plan for getting feedback to an MQP team, we would work on how the current IMGD Playtesting requirements could be revised to allow future teams to get improved feedback as well.

After finding a team to work with, we organized a playtest session with them to research how helpful playtesting in single sessions can be. We reserved a room and advertised the session,

offering free pizza and playtesting credit for IMGD students. The MQP team had a few computers set up to allow testers to play the game, and some students also played it on their own computers. After they completed the level, they took a survey that we collaborated with the MQP team on to review both the game and the playtesting experience.

Results

Interview Results

A registry of projects would allow students to search out and find testing opportunities to fulfill the requirement as it currently stands, without needing the projects to proactively set up testing opportunities. It would require the least amount of work over time, though the initial cost could be prohibitive, and making sure each project was added to the system properly could get difficult.

Running an event like Alphafest each term would be much more difficult; however, it would provide the easiest opportunity for each project to get playtesters, especially if IMGD were to require attendance at the event. Alphafest currently is likely the most reliable testing that most MQPs get, but it only happens once a year, during B term.

Requiring each professor to distribute MQPs to their classes for testing could also provide reliable testing, but the results might vary wildly with different professors. Some teach multiple large classes a term, and some only teach one or two small classes, so the amount of testing that projects would get could vary depending on who their advisors are.

We also spoke to two MQP teams about how they had been testing their games. The first MQP team, Purradice City, is a digital board game which is a mix of an engine builder and a city

builder. They do not currently have a testable digital prototype, but they have tested a few times with a small paper prototype. Their tests had about six non-team-members participating, and resulted in some useful feedback. They did get some unactionable feedback, but for the most part they were able to use what they got to improve the game. In the future, they plan to use a digital prototype to test over a paper prototype, as the game is designed to be run by a computer, not by hand.

When asked if their current methods were providing enough useful feedback, Matt of the Imperius MQP team disclosed that "For the most part, yes…" but included in his response that "a survey with X amount of questions is quite useful, specifically when we ask for detailed feedback about specific mechanics". This supports our thesis and shows the effectiveness of asking questions about specific features that were recently implemented instead of only doing more generalized questions. At the end of this specific question he did include the interesting ending when he said that the testers helped most when they are "motivated to critically analyze our game". He was then asked "Are the current questions/methods used when testing providing specific feedback that can provide the team a direction on how to improve the product" Matt provided a mixed response. He explained how "When people actually care and criticality analyze the material, we get good feedback that we can use. One or two word responses are useless" . Analyzing the responses that we got from the Imperius group gave us some useful information that we could work from when making our own testing documentation for our later implementation.

Test Results

After running the playtest, we got 23 responses on the form. Most of them agreed that this playtesting form was an improvement over most prior forms, though the organization of the testing was a bit haphazard, lacking proper instructions for playing the game.

Conclusion

After researching how testing has been done in the industry, we have compiled a few recommendations for improving the IMGD Playtesting requirement. They are to create a database of student projects, both MQP and Graduate, to allow undergraduate students to find testing opportunities on their own; encourage professors to share projects they are currently advising with their classes to get more feedback; encourage creation of early prototypes in MQP teams; and rework some of the 1000 level IMGD classes to have a more testing/critique focused curriculum.

Creating a database of projects would allow project teams to upload their work there when they have a build that needs to be tested by a wider community. Other IMGD students could then access the project, download and test it, and then fill out a form, which would provide feedback to the team, and inform their professor that they have completed playtesting for the requirement.

Encouraging professors to share projects they are advising with their classes would create a much more direct path for testing. By assigning it directly to classes, it would make it much more likely that students would test the projects for credit. They would also not need to select a project themselves, and would instead be given one already selected for them. This is a strategy

that Brian Moriarty used while teaching at WPI to get MQPs that he was advising more testing, as well as teaching students how to properly test and critique a game.

As a final option, Professors could rework the intro-level IMGD courses to be more focused on testing and critique. IMGD 1000 especially could involve much more testing of games as a part of studying the different parts that go into a game. While this option would take the most work, and be the most invasive to the system already in place, it would likely prove to be the most helpful in the long run by exposing more students to proper playtesting and critique practices early in their time at WPI.

Appendix A: Questions for Professors

- What is your history with playtesting at WPI?
- How does playtesting work in the classes you teach?
- How many participants do you get on average for playtesting?
- How do students typically get MQPs tested?
- Do you have any MQP groups at the moment who might be able to answer questions on their testing process?
- Do your MQP members have a test plan for your project?
- Do they have a bug database? If so, what format?
- How does the playtesting requirements show up in your course/syllabus? Does it?
- Would you say that playtesting in your course fulfills the intention of sharing work early/getting feedback at every step along the process?
- Are there any improvements that can be made to the current playtesting requirements?
- Would you prefer gradual improvements or start from scratch?

Appendix B: Questions for Industry Professionals

Testing Questions:

- WPI uses the term “Playtesting”, but our research indicates functional testing may be more the industry standard. What are your standard terms for different testing types?
- Regression
- Do you have Functional Test Plans (Test Plan)?
    - Under what circumstances do you run a complete Functional Test Plan?
    - Do you test every function? If not, what percentage do you test?
    - When do you run your COMPLETE Functional Test Plan? (Release Candidate)
- Do you have a Build Verification Test (BVT or Smoke Test)?
    - Do you test every build?
    - Is the BVT time boxed? - Ivory Tower
- Do you do quality testing (Fun, engagement, polish)?
- Which of these methods do you use?
    - Manual testing vs a test plan?
    - Automated testing? Is there a specific tool for this?
    - Is Code Coverage a KPI at your company?
    - Other methods of testing?
- If you have seen other testing methods, how have you catered yours to fit your needs?
- ^Do the methods depend on the type of game/ product being tested?
- ^Do different companies have different methods or are they similar?
- ^Were there any rejected methods?
- ^Are there standards set that testing methods have to meet?
- ^What determines which method you will use for testing?
- Do your testers have certifications?
    - What certifications are they?
    - CAST, CSQA, ISTQB?
- Is your testing team considered part of the development team?
- Does your testing team report to a member of the development team? (Organization Chart)
- Does the development team do Continuous Integration Testing? (Turbine had used this)
- How often does the development team build?

Bug Questions:

- What software or interface do you use for your testing (Bugzilla,Jira, Redline, etc.)?
- What does your bug report look like (fields/data)?
- Who controls the severity of the bugs?
    - Ivory Tower - QA controls

        - What terms do you use here?
            - Crash
            - Critical
            - Bug
            - Polish
            - Feature Request
- Who controls the priority of the bugs?
    - Ivory Tower - Dev controls
    - What terms do you use here?
        - Do you use
            - High
            - Medium
            - Low
        - Do you use MoSCoW
            - Must (0)
            - Should (1-3)
            - Could(4-6)
            - Wishlist (7-9)

Database Questions:

- What is the status of your bugs?
    - Open
    - Confirmed
    - Closed
    - Closed Confirmed
    - Will Not Fix
    - As Designed
    - Could Not Reproduce
- Do you retest closed bugs?
- Do you retest open bugs?

References

ASQ. (n.d) The History of Quality. https://asq.org/quality-resources/history-of-quality. Retrieved 28/9/2022.

Fullerton, T., Swain, C., & Hoffman, S. (2004) Game Design Workshop: Designing, Prototyping, & Playtesting Games. CRC Press.

Gonzalez, A. (Sep, 2022). Personal Communication.

Kolb, R. R. & Hoover, M. L.. (2012) The History of Quality in Industry. Sandia Report. https://digital.library.unt.edu/ark:/67531/metadc843832/m2/1/high_res_d/1051714.pdf

Loh, C. (2021, April 9). *What Happens To Prototype Aircraft Once Testing Is Complete?* Simple Flying. https://simpleflying.com/what-happens-to-prototype-aircraft-once-testing-is-complete/

Moriarty, B. (Oct. 10, 2022). Personal Communication.

Shinkle, K. (2021) Elder Scrolls: Why Oblivion's Character Dialog Is So Bad. ScreenRant. https://screenrant.com/elder-scrolls-oblivion-npc-dialog-bad-weird-conversation/ . Retrieved 29/9/2022