QUALITY ASSURANCE TESTING IN VIDEO GAMES

The importance and impact of a misunderstood industry

Essi Ruuska

Bachelor's thesis
May 2015
Degree Program in Film and Television
Cinematography & Film Lighting

TAMPEREEN AMMATTIKORKEAKOULU
Tampere University of Applied Sciences

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Program in Film and Television
Cinematography & Film Lighting

Essi Ruuska
Quality assurance testing in video games
The importance and impact of a misunderstood industry

Bachelor's thesis 62 pages, appendices 6 pages
May 2015

The aim of this research was to provide a more holistic insight of the video game quality assurance industry to video game industry professionals and prospective employees in order to promote the importance and impact of quality assurance testing in video games. The motive for this thesis came from the author's work experience in video game quality assurance testing, and from realizing how little is known about the industry. The research question was defined as 'what is video game quality assurance, and what is its impact on an end product?' This thesis was executed as a qualitative research, as it used structured interviews as the main method of data collection. Other methods of data collection were an online survey, literature and Internet sources.

The results gained from this research were that quality assurance consists mainly of functionality quality assurance, localization quality assurance, and platform certification quality assurance. Quality assurance testing is an important part of game development, because it detects and reports defects within a game, which helps improve the product. Another result gained was that the future of the industry is expected to boom, as video games become the major form of entertainment, and thus the role of quality assurance will increase. Hence it is important to understand quality assurance testing as an essential part of game development.

The conclusion of this thesis was that quality assurance is an important matter in today's game development, but some game companies still do not view it as necessary for their games. With the time and money invested in quality assurance testing, the game companies could, however, avoid financial and reputation losses in the eyes of end users. By understanding the importance of quality assurance testing and investing in it on time, the booming future of the industry can also be secured.

Due to the author's signing of a non-disclosure agreement with their previous employer, confidential information such as intellectual property, any detailed information about the company, its clients or any other game industry company was excluded from this thesis.

Key words: quality assurance, video games, video game industry, game development process, issues, future

CONTENTS

GLOSSARY

AI	Artificial intelligence.
Beta	A version of a game in which the software has its primary features completed but contains of known and unknown issues.
Build	A version of the video game software to be tested.
Company X	Pseudonym for a video game quality assurance company used as an example in this thesis.
CQA	Certification Quality Assurance. A practice of assuring the platform's certification quality, i.e. terminology.
FPS	Frames per second. The amount of how many frames per second are displayed in a game. The higher the fps, the smoother the graphics and movement are in a game. Also short for first-person shooter, a game in which the user only sees the character's arms and gun, as in first person.
FQA	Functionality Quality Assurance. The process of preventing and isolating mistakes and defects in a game's functionality.
Gold	Final version of a game that gets released.
Hardware manufacturer	Manufacturer of a game console or a system, such as Sony (PlayStation®) or Microsoft (Xbox).
Hardware manufacturer's guidelines	Quality guidelines set by the hardware manufacturer.
IP	Intellectual Property. A work or invention that is the result of creativity, such as a manuscript or a design, to which one has rights and for which one may apply for a patent, copyright or trademark.
LQA	Localisation Quality Assurance. The process of preventing and isolating linguistic issues and defects in game's translation.
NPC	Non-player character. A character programmed with artificial intelligence in the game.
Placeholder	A part in a game that is not yet completed, but has a temporary graphic or a string holding place for the final graphic or text.

QA	Quality Assurance. Way of preventing mistakes or defects in manufactured products and avoiding problems when delivering solutions or services to customers.
String	A piece of text implemented in a game. Can consist of a single word, a sentence or a complete paragraph.
Submission	Evaluation by a hardware manufacturer to determine if the game is ready for release.

1 INTRODUCTION

Game testing jobs are often glorified and seen as a cool and easy way to make money. However, although many higher-ups in the video game industry started as video game testers, testing jobs are not always an easy way to step up to bigger shoes within the video game industry.

While working as a video game quality assurance tester in Montreal, Canada, I ran across various misperceptions about testing jobs, and realized that the video game quality assurance is still not a well-known or understood industry. Video games are said to become the major form of entertainment in the near future, which is why I find understanding and appreciating quality assurance testing as a part of game development an important matter. In order to provide better understanding of the industry, this thesis focuses on the Canadian video game and video game quality assurance industries, and covers the common testing workflow, possible careers within the industry, discloses how to master the art of testing, covers the importance and impact of testing, and discusses the video game and QA industries' future. The will to explain the industry, what game testing really is, and to have the reader ponder the future of video games and quality assurance are the reasons behind doing this research.

Let us start with the misperceptions I have come across during my time in the quality assurance industry. Number one is that testers merely play games, which is partially incorrect, as a tester's job is to test the game and find issues, not play it for fun. In fact, testers are often performing specific tasks given by the client and are actively minimizing time spent playing in order to increase production, which can sometimes lead up to hefty amounts of work. Another common misperception of video game quality assurance, shortly QA, is that it is not as sophisticated as software quality assurance, which is a process that ensures that the developed software meets and complies with defined or standardized quality specifications. The meaning of both software and video game QA is to ensure the development of high-quality software, and while there is less of a need for formal training in video game quality assurance, the complexity of testing wildly different creative games has made video game QA a very flexible and accommodating discipline. Another misperception is that QA is a commodity. However, many game companies still see quality assurance as expenditure that has no real value. Without test-

ing a game, it is safe to say the game will not succeed – the worst possible scenario for anyone willing to make money out of their games. These are all matters that encouraged me to choose this topic for my research.

Collins states (2010, 12) that a research process commonly has nine phases, of which some are in concurrent communication with each other, some are not. Before starting with this thesis, along with coming across the aforementioned misperceptions, I had a firm knowledge base gained from work experience, which enabled me to choose the theme for my thesis. The definition of the research problem was, however, a bit more complicated, and as described by Collins (2010, 12), it was in constant interaction with my familiarization with the sources from the Internet, literature, structured interviews, and an online survey. The Internet and literature sources were carefully investigated using critical reading techniques for their accuracy and reliability before their use as sources in this thesis. These techniques consisted of examining the evidence of arguments presented in each source, investigating the possible influences for the arguments and evidence, finding out about the limitations for the study design and focus, pondering about the interpretations in each source, and deciding what arguments I wanted to use in this thesis. (Collins 2010, 180.)

Structured interviews were used as the main method of collecting data for the research, and were performed by sending a list of similar questions about each area of quality assurance to the recipients via email. These questions consisted of general information about quality assurance – what is it, why is it important, who is it for – and of more open questions, such as what are the biggest issues within the industry when it comes to new technology, or to understanding the impact of QA testing. These were all essential questions in order to explain the causal relations between the different areas of QA testing, and the impact on end user experience and game companies' reputation. The full list of questions can be found in Appendix 1. The professionals interviewed were Localization Quality Assurance Manager Olivier Chrun, Functionality Quality Assurance Manager Mathieu Lachance, Localization Quality Assurance Project Manager Michael Cartier, Certification Quality Assurance Project Manager Roxanne Beriault, and Localization Quality Assurance Test Lead Eric Chan. Each one of these individuals has extensive knowledge of the quality assurance industry and its relation to the game industry as a whole, and they provided a professional insight to each of the different areas of video game quality assurance. These interviewees were chosen for the interviews due to their

high level of professionalism, which all of the interviewees have gained through working on the quality assurance field in various occupations.

The online survey was conducted as an anonymous survey for gamers (people who play video games) by using Google Docs questionnaire template. The questions consisted of what types of games and on which platform the participants mainly play, and if they had ever come across issues that had an effect on their gaming experience, and whether they think quality assurance testing had anything to do with how well the game was received. The questions were the same for each participant, and the answers were a list of multiple-choice answers, to which the participants could answer 'Other' and comment to specify. In total 129 participants completed the survey, of which 97% answered they had encountered bugs when playing. An amazing 82% of the participants thought quality assurance was an important part of game development, even if it sometimes causes delays in release dates. The full list of questions can be found in Appendix 2.

After the interviews and other data collection, the data was analyzed to finally define the research problem. Brainstorming and mind mapping were my principle methods of constructing the research problem, as well as the guidance from my thesis instructor. With the help of these, and Collins (2010, 28-33), I understood that a good research topic is interesting to its writer, has a clearly defined focus and set of research questions, is able to provide the readers with new insight about the topic, is feasible to complete within the time and resources given, and helps its author by giving more information about the topic, which will later help the author in their career or studies. Based on Collins' (2010) advice, after careful consideration and some trial and error, I decided to define the research question as 'What is video game quality assurance, and what is the impact of quality assurance testing on an end product?' This topic is not only interesting to game industry professionals, prospective employees, the general public and me, but was also feasible to execute within the time and resources given. Thus the main goal of this thesis is to explain video game quality assurance testing, and uncover the importance of quality assurance testing and the impact it has on the end product. Moreover, the future of the video game industry is discussed in the end of this thesis.

Understanding the quality assurance and video game industry as a whole, and the impact of quality assurance testing on video games are the biggest matters I wanted to acquire from the research, as I have determined to continue pursuing a career on the field,

and the holistic understanding of these matters are quintessential for my professional growth. These matters come especially handy in management positions, as they provide me with the assets and the ability to think objectively, and consider various alternatives and seeing their causal relations before decision-making. On the other hand, the readers that can benefit from this research are video game industry professionals, new and experienced, who are looking for a reason to invest in quality assurance testing; prospective employees, who are looking for a more comprehensive insight on testing jobs; and for the general public to gain a better understanding of quality assurance as a part of the video game development process.

The conclusions gained from this research are that quality assurance testing is an essential part of today's game development, and testing jobs are not as easy and fun they are often seen. According to Localization Quality Assurance Manager Olivier Chrun (Interview, 2015), testing and judging a game takes skills and knowledge, and while no formal education is required, being a successful tester requires gaming abilities, attention to detail, critical thinking and team working skills, the ability to work hard under pressure, and for localization quality assurance testers, linguistic abilities. Without quality assurance testing subpar games with game-breaking features would be released, which would lead to the end users heavily criticizing the games and their makers, and eventually to the decrease of the game companies' clientele.

As video games become even more popular, the meaning of quality assurance testing becomes paramount. Understanding the quality assurance testing processes as early as possible could help many game companies in maintaining their reputation in end users' eyes – and to financially benefit from their games. In the end customers are what keep game companies running, so everything possible should be done in order to provide top quality video games to satisfy the customers' needs.

2 VIDEO GAME QUALITY ASSURANCE

2.1 What is video game quality assurance?

This chapter explains what video game quality assurance is, how it is tested, and why it is an important factor in game development. In the following subchapters different areas of quality assurance and their impact on the final product are clarified.

The video game industry consists of three main players: The developers, who provide intellectual property (IP) of game design; the middleware companies, who provide the tools by which the design is created and delivered; and publishers, who are the financiers and take responsibility of bringing the product to the market (ESAC 2009). According to the 2009 survey done by Entertainment Software Association of Canada (ESAC), most of the jobs within the video game industry in Canada come from developers, but the growing sector of middleware companies, such as quality assurance companies, cannot be left unmentioned.

According to Localization Quality Assurance Project Manager Michael Cartier (Interview, 2015), quality assurance in video games is not considerably different from any other type of software QA. Before a game is released, an important and often forgotten part is to make sure it is behaving as intended, and reaches the quality standards end users are expecting, and meets the requirements provided by the hardware manufacturers. This is when QA testing takes place. (Interview, Cartier 2015.) QA testing in video games includes functionality quality assurance, which consists of ensuring the game is technically ready for release; localization quality assurance, which aims to find and analyze issues within a title's translation; and platform certification, which verifies that a hardware manufacturer's requirements are met within a game. The different areas of video game quality assurance are further explained in the chapters 2.1.1-2.1.3.

Additionally, QA testing helps a game pass submission, which is the final test that determines whether a game's software is ready to be released. It is essential to pass the submission to receive the green light to actually sell the game, to avoid production delays and costly resubmission fees. In my interview with Functionality Quality Assurance Manager Mathieu Lachance (Interview, 2015), he states that to pass submission a

team of testers detect defects on all designated areas; the developers fix the issues reported, and the quality of the product is improved (Table 1) (Interview, Lachance 2015).

Table 1 was combined by utilizing work experience from quality assurance testing. The left side of the table depicts game development as a whole, which is followed by different steps (steps 1-7). On the top, different participants in game development are presented, and their role in each phase is explained below. The whole process of development is further explained in chapter 3.1.

TABLE 1: Quality assurance testing in relation to game development and end user experience (Ruuska 2015).

		Game companies	Testing studios	End users
Game development	1	Request testing from testing studios	Test games for LQA, FQA and CQA	
	2		Find and report issues	
	3	Fix the issues received and send a new build to be tested	Test the new build by regressing the issues and report new ones	
	4		Deliver the means of creating an intact, functional game	
	5	Utilize the means in order to improve the game		
	6	Release the game		Buy and play the game, and give feedback based on their experiences
	7	Receive either good reputation and financial gain or a loss of clientele	Receive either a good reputation or a loss of clientele	

If all the areas of testing applicable to the game are covered, it is more likely that the end users will be able to appreciate the title the way developers conceived it, in all languages and in all platforms (Interview, Cartier 2015).

2.1.1 Functionality Quality Assurance

A piece of software crashes unexpectedly – a problem many have encountered while using smartphones, computers and video game platforms. Functionality quality assur-

ance – referred to as FQA in this thesis – is a process of testing the software to ensure it is acting as intended and is not, in industry terms, broken (Interview, Lachance 2015).

FQA is an important step in game development, as it tests the game for possible defects that affect end user experiences. As an example, according to Penn's documentary *Atari: Game Over*, Atari's game *E.T.* received a reputation of being the worst game ever. The reason *E.T.* received such reputation was likely due to the fact that the game was developed from beginning to the end in only 5 weeks – and was thus not tested properly for its functionality. As a result, E.T. would fall into pits that it could not get out of, and the player's progression was blocked. (Penn 2014.) FQA is a practice of quality assurance, where functionality defects such as the aforementioned are investigated, analyzed and reported.

The basic methodology of FQA testing is to test the game's software. Finding a collision issue (a surface is not solid though it is supposed to be) (Picture 1), such as the aforementioned E.T. falling to pits, can include running the character into every item and surface possible, and finding which of these supposedly solid surfaces and objects have no collision.

PICTURE 1: Collision issue in THQ's *Cars* video game. (Picture: Miles Messenger 2011)

An example of a collision issue is depicted in the Picture 1. FQA testers spend endless hours testing every area of the game for finding and reporting issues from minor graphic mistakes to game-breaking features. According to Localization Quality Assurance Test Lead Eric Chan (Interview, 2015), to help find submission failing issues (Table 2) within a game's functionality, testers pay special attention to progression blockers, such as the E.T. example.

Table 2 depicts common functionality issues within video games, and gives examples on how these issues can affect end user experience. Submission failing issues are revealed on the right column to further illustrate the importance of functionality quality assurance testing. The information on the table is based on work experience in video game quality assurance.

TABLE 2: Common functionality issues in video games (Ruuska 2015).

Issue type	Issue name	Example	Effect	Submission failing
Performance	Crash	Game quits unexpectedly.	Unsaved data will be lost. Possible progression blocker.	Yes
	Hang	Game freezes for x amount of time.	Possible progression blocker.	Yes
Graphic	Placeholder	Temporary texture implemented instead of the final texture.	Game looks incomplete. Possible progression blocker.	Possibly
	Incorrect graphic	Incorrect texture is implemented.	Game looks incomplete.	No
	Rendering	Graphics flicker.	Game looks incomplete. Graphics are hard to look at. Possible progression blocker.	Possibly
	Missing texture	Graphics are not implemented.	Game looks incomplete. User can get confused. Possible progression blocker.	Possibly
Collision	No collision	User can walk through walls or fall off the world.	Possible progression blocker.	Yes
	Incorrect collision	Invisible walls or obstacles.	Possible progression blocker.	Possibly
Artificial Intelligence (AI)	Non-responsive NPCs (non-player characters)	User cannot interact with the NPC, user might not be able to receive quests.	Possible progression blocker.	Possibly
	NPC stuck on a loop	NPC is performing the same action repeatedly.	Game looks incomplete. Possible progression blocker.	Possibly
	Aggressive NPCs	NPC starts attacking the user for no rea-	Possible progression blocker.	Possibly

		son.		
Audio	Clipped audio	Audio stops in the middle of a sentence or too early.	User might get confused and not understand the audio. Possible progression blocker.	Possibly
	No audio	No audio is played.	Possible progression blocker.	Possibly
	Low or high volume	Audio is played on volume too high or too low.	Audio might be hard to understand or too loud. Possible progression blocker.	Possibly
	Wrong audio	Wrong audio is played.	User might get confused. Possible progression blocker.	Possibly
	Audio and subtitle mismatch	Audio and subtitles are displayed on different times.	User might get confused.	No
	Placeholder	Temporary audio is played.	User might get confused and not understand the audio. Possible progression blocker.	Possibly
	Timing	Audio is played at an incorrect time.	User might get confused and not understand the audio. Possible progression blocker.	Possibly
Text	Incorrect message	Incorrect message is displayed.	User might get confused and not understand the message. Possible progression blocker.	Yes
	Corrupted font	Characters are not displayed completely, i.e. umlauts or accents are missing.	User might get confused and not understand the message. Possible progression blocker.	Possibly
	Cut off	Text is displayed cut off.	User might get confused and not understand the message. Possible progression blocker.	Possibly
	Overlap	Text is overlapping with another piece of text or graphics.	User might get confused and not understand the message. Possible progression blocker.	Possibly
	Font size	Font is displayed too small or too large.	User might get confused and not understand the message. Possible progression blocker.	Possibly
	Incorrect instructions	Incorrect instructions are given to the user.	Possible progression blocker.	Yes
	Terminology	Incorrect terminology is displayed.	User might get confused.	Yes
	Lack of required information	Not enough information is given to the user to start or com-	Possible progression blocker.	Yes

		plete a quest.		
Camera	Wrong angle	In a first person shooter game the camera is displayed in third person angle, or the other way around.	User might get confused. Possible progression blocker.	Possibly
	Clipping through objects	Camera clips through objects, such as walls. User is able to see inside the objects.	Game looks incomplete.	No
	Frozen camera	Camera does not move.	Possible progression blocker.	Possibly
Balance	Game difficulty not balanced	Game is too difficult or too easy to play despite choosing the difficulty level.	Possible progression blocker.	Possibly
	Overpowered enemies	Enemies are too powerful and can kill the user in a single strike even when wearing armour.	Possible progression blocker.	Possibly
Design	Game controls	Game controls are difficult to handle, i.e. cars are difficult to drive or moving around is hard.	Possible progression blocker.	Possibly
	Action possible when not allowed / action not possible	User cannot interact with objects, or can interact when action is not supposed to be available.	User might get confused. Possible progression blocker.	Possibly
	Scripting	The game does not follow the action it is supposed to follow, i.e. pushing a button does not trigger an action.	Game looks incomplete. Possible progression blocker.	Possibly
	Unreachable areas	The user is unable to reach parts of the map.	Game looks incomplete. Possible progression blocker.	Possibly

By documenting and collecting evidence of these defects, testers are able to provide the tools for developers to reproduce and fix the issues. These documentations can consist of bugs submitted in the client's bug database, crash logs and performance reports. To be specific with their bugs, testers often check if the issue in question occurs on another level, with another character and with another object or area, as this information could be quintessential in fixing the issue.

According to Lachance (Interview, 2015), a good FQA tester has attention to detail, the will to break the game, and thinks ahead of the developers. An FQA tester will think of

things a developer might never spare a single thought for – and find an issue. (Interview, Lachance 2015.) The best FQA testers are competent gamers with years of experience in gaming, who are also strong communicators. Along with these abilities, both team working skills and the ability to work autonomously are desired when looking for a good candidate for an FQA tester.

Without investing in FQA video games could fail submission, and would oftentimes receive a bad reputation with its end users. With enough time and resources allocated for FQA, these issues affecting the end user experience or failing submission can be mapped, analysed and fixed. An intact, functional game is not only a joy for the end users, but will also bring wealth and reputation to its quality assurance company, and most importantly to its developers and publishers.

2.1.2 Localization Quality Assurance

No native speaker is willing to read a text that sounds like it was automatically translated by a piece of software. Localization quality assurance – referred to as LQA in this thesis – is the practice of assuring a translation's quality within a title. It is intended to review the work for proper use of certain terminology, idioms, translations and cultural accuracy (Pérez 2011).

LQA testing is a process of assuring a translation's quality in software. Whereas FQA tests the software itself, LQA tests the translated text implemented in the software, and ensures all translations are correct, fluent and match the source language. Unlike FQA, LQA is often seen as an expenditure that has no real value, because whereas FQA testing provides imminent benefit for developers, LQA has more impact on end user level (Interview, Cartier 2015). In reality LQA is an essential part of the translation work and a localized game's development. In her blog article, Pérez (2011) explains that in it is very important that the linguistic editor and the proof reader are not the same person as the one who is responsible for the original translation. A translator doing proofreading and editing is not a reliable way to assure the quality of the translation (Pérez 2011). In order to check a translation's quality native speakers of the respective languages are used for LQA testing.

LQA priorities are that the game boots in and is localized in all the respective languages, all strings are translated for all languages, all languages respect the terminology requirements provided by the hardware manufacturers, and no major text implementation issues are found. Implementation issues mean that important messages are unreadable due to corrupted font, are displayed cut off or overlapping in the text, a placeholder is displayed (Picture 2), or unlocalized or incorrectly translated text is implemented in the game. (Interview, Cartier 2015.)

The hardest part of LQA testing is finding strings that are generated randomly, or need a very precise and complex course of action to trigger. Usually developers help LQA testers trigger all of the strings by adding debug in the game. Finding strings can be tracked with various types of files or software, in which testers mark whether a string has been found and whether it needs to be edited. This process makes it possible for both the test lead and the client know when all the content has been verified. (Interview, Cartier 2015.) On developer and publisher level LQA testing ensures submission failing issues are found (Table 3). In LQA the submission succeeding rate is generally high, as the aforementioned issues are oftentimes easy to find, given that enough time is allocated for LQA testing.

PICTURE 2: Placeholder string is displayed instead of the final string in English. (Photo: http://citadel.prophpbb.com/topic5579.html, modified)

Table 3 depicts common localization issues within video games, and gives examples on how these issues can affect end user experience. Submission failing issues are revealed

on the right column to further illustrate the importance of localization quality assurance testing from developers and publishers' point of view. The information in the table is based on work experience in video game quality assurance.

TABLE 3: Common localization issues in video games (Ruuska 2015).

Issue type	Issue name	Example	Effect	Submission failing
Implementation	Cut off	Text is cut off from its end, beginning, or from above or below.	Text can be illegible for the user, and might block the user's progression.	Possibly
	Overlap	Text is overlapping with another piece of text, or with an image.	Text can be illegible for the user, and might block the user's progression.	Possibly
	Placeholder	A placeholder text is implemented instead of the final text.	The user does not see the final text. Progression might be blocked.	Yes
	Font	Text is too big or too small to read. Font is corrupted. Text is bidirectional.	Text can be illegible for the user, and might block the user's progression.	Possibly
	Wrong language	Wrong language is implemented in the game.	The user does not understand the text, and progression might be blocked.	Yes
Linguistic	Grammar	Text contains a grammar issue, i.e. incorrect conjugation.	Can make the user confused or not understand the text.	No
	Spelling	The text contains a spelling mistake, i.e. missing accent, no space between words, typo.	Can make the user confused or not understand the text.	No
	Translation error	Translation does not match the source. Translation's quality is poor. Cultural expressions or idioms are incorrectly translated.	Can make the user confused or not understand the text.	Possibly
Audio	Audio cut off	Localized audio is cut off too early or in the middle of a sentence.	User might not understand the audio.	No
	Subtitle and audio mismatch	Subtitles and audio do not match.	User might get confused by the mismatch.	No
	Timing	Audio is timed incorrectly, audio and subtitles are not timed correctly.	User might get confused by the mismatch.	No
	Wrong audio	Wrong localized audio is implemented.	User might not understand the audio. Progression can be blocked.	Possibly

Other	Terminology	Wrong terminology is used in the text or the audio.	Certification issue.	Yes
	Instructions	Wrong instructions are given in the text or the audio.	The user might get confused. Progression can be blocked.	Yes

LQA testing does not require a tester to have formal education in translation or game design, but to be able to isolate and fix linguistic issues LQA testers have to be skilled in their native language, have to have attention to detail and have the will to make the translation the best it can be (Interview, Chrun 2015). Additionally, assuring a translation's quality requires not only linguistic abilities, but also involves cultural understanding of the area the language is from. Identifying expressions that are used or not used in the language tested, and changing them to the accurate translation is an essential part of LQA testing. For example, a Spanish tester can test a game that is localized to Mexican Spanish for the general linguistics, but it requires a Mexican to assure that cultural expressions, idioms and dialect are correctly used. This is especially true with languages that have two or more variables, such as Portuguese and Brazilian Portuguese, or European French and Canadian French. Therefore LQA testing requires people from various countries and backgrounds to ensure the highest possible quality of a title's translation.

Moreover, LQA testers often have to know platform specific terminology in their own language – and in every platform they test. Certification Quality Assurance Project Manager Roxanne Beriault (Interview, 2015) states that some QA houses or publishers only verify terminology used in the platform with native speakers – LQA testers – whereas others have decided to survive without them. There are presumably as many methods of testing terminology as there are QA teams, but the most common way is to employ LQA testers for the tasks. These checks are fairly simple, as the testers are provided with a list of terminology used within the platform tested. (Interview, Beriault 2015.) However, every single word affected with the platform terminology has to be checked to ensure its correctness. The process of certification quality assurance is further explained in chapter 2.1.3.

According Cartier (Interview, 2015), if a developer decides to translate their game, they should invest in LQA testing. If LQA testing is not performed on a translated game, it would be like developing a game but not checking it for functionality issues: the poor quality of the localization would be noticeable. Translation issues are sensitive, as they

can negatively impact on how the game's creators are seen. Translation issues can cost a company their reputation within a language that the game is localized in, and is therefore worth investing in. (Interview, Cartier 2015.)

In conclusion, LQA testing aims to assure translation's quality within software. By using native speakers of the respective languages, translation and certification issues that can affect the end user experience and the company's reputation can be found, analyzed and fixed before the release of the end product.

2.1.3 Platform Certification Quality Assurance

Platform Certification Quality Assurance – referred to as CQA in this thesis – consists of verifying that the requirements and standards of the hardware manufacturers are conformed to (Interview, Beriault 2015).

CQA testing includes verifying functionality, mastering and build packaging (ensuring the software distributed meets the sizing and naming conventions of the hardware manufacturer) and terminology requirements set by the hardware manufacturer. CQA testing follows the hardware manufacturer's guidelines for proficient functionality and terminology within the platform. (Interview, Beriault 2015.) According to Beriault (Interview, 2015), CQA is an essential procedure in game development, and while its primary benefit is to the developer and publishers of a title, there are benefits for all individuals involved with the software. End users benefit from the requirements that ensure quality in hardware and software, i.e. preventing crashes and ensuring proper display. Therefore CQA is a part of the manufacturer's effort to provide high quality standards to the end user, and not allow subpar games to be released on their platform. (Interview, Beriault 2015.)

CQA testing process depends heavily on the platform a video game is released on, as every platform has their individual requirements in both hardware and terminology. These requirements are provided to the testing team, along with detailed instructions on how to perform every relevant check. These checks can include testing that the hardware is compatible with the specific software, and acts as intended under certain situations. (Interview, Beriault 2015.) Due to the non-disclosure agreement signed by the

author and the interviewee more detailed information about the different CQA issue types cannot be disclosed in this thesis.

The best CQA testers are skilled in critical thinking, are detail oriented, and are highly logical and technical in nature (Interview, Beriault 2015). CQA testing can be repetitive and complex, thus an ideal CQA tester is someone with the aforementioned qualities. Additionally, testers need to be verbally talented, as communication is a key factor in successful CQA testing. (Interview, Beriault 2015.) In Company X, locals rather than foreign workers generally test CQA. This is due to the fact that while LQA testers verify the hardware manufacturer's requirements for terminology, the actual hardware can be tested without having linguistic skills in a foreign language.

There are various reasons why some developers choose not to invest in CQA at all. Some are simply overconfident in presuming their title is already following the guidelines, others have budget constraints, and many face schedules and deadlines too tight to allow CQA testing. Usually developers invest in a minimum of CQA testing to ensure there are no delays due to submission. (Interview, Beriault 2015.) Nonetheless, investing as much money and time as possible for CQA testing could prove useful in the submission process. As stated before in this thesis, finding major issues before submission can result in a title being released on time. However, it is important to note that not every game developer faces the need to invest in CQA, as not every platform for which games are created have strict requirements to follow. For example more traditional video game platforms, such as PlayStation® and Xbox will require some form of CQA, but some titles released on PCs do not meet any particular set of requirements. (Interview, Beriault 2015.) Indie developers can also benefit from Microsoft's new policy of Xbox 360 market being free from submission fees. This is beneficial to Microsoft, as new titles keep targeting the platform, and for indie developers as they do not have to worry about spending thousands of dollars in resubmission fees. (Interview, Beriault 2015; Yin-Poole 2013.)

CQA is a quintessential factor in receiving the permit to sell the game, and in avoiding delays in release dates and additional expenses in resubmission fees. However, even if most major certification issues are found in the submission process, it is possible for the game to contain some issues even after submission. Investing enough time in CQA prior to submission is important, as the amount of large impact issues in the title will be sig-

nificantly reduced, which benefits both the game makers and the end user. (Interview, Beriault 2015.) Additionally, ensuring the quality and consistency of a title leads to maintaining the brand's image. Without proper CQA it is easy for a title to fail submission, and lead to a long list of issues, which can prove difficult and time consuming to fix in the later stages of development. Investing in CQA as early as possible during the development assures that issues within the title's architecture are caught as early as possible, hence reducing the time and effort required to redo these stages. (Interview, Beriault 2015.)

3 QUALITY ASSURANCE TESTING IN VIDEO GAMES

3.1 Quality assurance testing process in video games

The quality assurance testing processes in video games are different for each project and quality assurance company. The processes usually depend on the client, but there are some similarities. In this chapter, the most common ways of QA testing and workflows are described. The following subsections handle the means of mastering successful testing, such as the importance of concurrent communication.

A project's workflow consists of finding, analyzing and reporting new issues. Knowledge and fluency of each project's workflow is important to effectively test a game, and thus minimize the time needed for testing. The most common bug workflow, or workflow upon finding an issue, is demonstrated in the next paragraphs.

Upon finding an issue, the tester takes a screenshot, and searches the client's bug database for duplicates. This enables the tester to find out whether or not a bug for the issue in question has already been submitted by another tester. If a duplicate is found, no new bug will be entered. If no duplicates are found, the tester submits a new issue in the database. LQA testers often have to enter new bugs for their respective languages, even if the same issue has already been submitted in another language. To track which issues occur in which languages, LQA testers add the issues to a file for other testers to check if the same bug occurs in their language; an act called crosschecking. Depending on the client, issues that occur in various languages either get merged into one bug, or a new bug is entered for every affected language. In FQA crosschecking is not common, as everyone is testing in the same language – mainly English – and therefore FQA testers seldom search for language specific issues.

At the end of the day, bugs are forwarded to the client in question, who then forwards the issues to either the game's developers, if the problem is within the code, or translators, if the issue is a linguistic one. The fixes of each bug will be implemented in the new build (version of the game), which is sent to the testing team as soon as possible. In the new build the bugs reported earlier have been claimed fixed by the client, and the testers will regress the issues upon receiving a new build. Regressing a bug means that

the tester looks into the bug database in order to remember how to reproduce the bug again, and thus finds out whether the issue has been fixed or not. If the issue is fixed, it receives a status as verified fixed; the bug has been regressed and can no longer be found within the game. If the issue still occurs, it will be marked as failed, and will be sent back to the client, who then further investigates how to fix the issue in question.

The loop of sending bugs back and forth from the client to the testing teams continues until no major issues can be found in the game. However, a number of issues can usually be found in the Gold version (final product). When asked why there are still issues in Gold versions, Lachance (Interview, 2015) states that it is practically impossible to create a flawless game, be it within or outside developers' control. Some of the bugs reported are not reproducible, which means that they do not occur every time, even though a tester clearly has instructed what needs to be done in order to trigger the issue. Bugs can also occur in a blink of an eye, and in that blink a tester needs to take a screenshot, memorize what they did to trigger the issue, and write down the instructions on how to make the bug trigger for someone who has never seen the issue before (Thang 2012). Specifying which actions lead to the discovery of the issue are often complicated to report, which is why some known bugs never get fixed. However, depending on the severity of the bug, testers will try everything and as long as is needed to accurately instruct how to reproduce the issue in question.

Versioning issues are another good example for why bugs end up in a Gold product. Versioning issues are bugs that occur in build 1.1.1, are verified fixed in 1.1.2, but occur again in 1.1.3, or even in later builds. The simple reason for versioning issues is that upon fixing 1.1.2, a developer went back to an older version of the code in order to update to 1.1.3, but the code for 1.1.3 did not contain the fix for 1.1.2 anymore, as it was the same as the code for 1.1.1. (Interview, Chan 2015.)

Additionally, the developers need to make decisions on which bugs to fix, since development schedules are usually tight. Therefore some minor bugs that will not make the game fail submission or cause remarkable inconvenience to the end user, such as graphic or minor grammar issues, never get fixed even if they are reported accordingly.

There are also millions of ways to trigger an issue, and trying out all of these ways is practically impossible with the limited time and resources testing usually has designated

for it. No gamer plays the same, and thus no tester can emulate all the possible situations every single end user is going to try while playing. (Thang 2012.)

3.1.1 Intercommunication – key to successful testing

As stated above, quality assurance testing processes are heavily dependent on the client's method of development. The two most popular methods in software development are the waterfall method and the agile method. In his paper entitled *Managing the Development of Large Software Systems*, Dr. Winston Royce criticizes sequential development, such as the waterfall method (Figure 1). In his opinion, software should not be developed like a car on the conveyor belt, where each part is added after another and each part is dependent on the previous. Instead, the best approach for developers would be to gather all the project's requirements, complete all of its architecture and design, then write the code, test it, and make each phase communicate with each other (Figure 2).

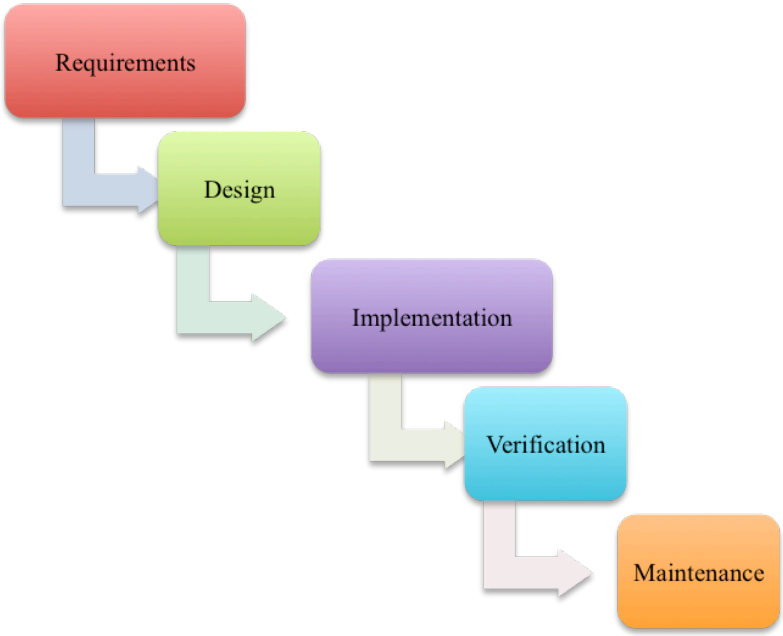


FIGURE 1: The waterfall method as depicted by Royce (Ruuska 2015).

From a testing point of view, the problem with the waterfall method is easy to see. It assumes that every issue can be identified before any design or coding occurs – a scenario that is practically impossible in today's software development (Royce 1970, 328-

331). In 1970, Royce already realized that the agile method provides opportunities to constantly be in communication with different phases of development – a method still relevant for today's software development. The agile method was and is still achieved by regular cadences of work, also known as iterations. At the end of these iterations, the team will be able to present a product ready for release, whereas in the waterfall method the teams only get one chance to get every aspect of the project right. (Royce 1970, 328-331.) The iterative method of working reduces development costs and thus time to market, because teams can develop the software at the same time they are gathering requirements.

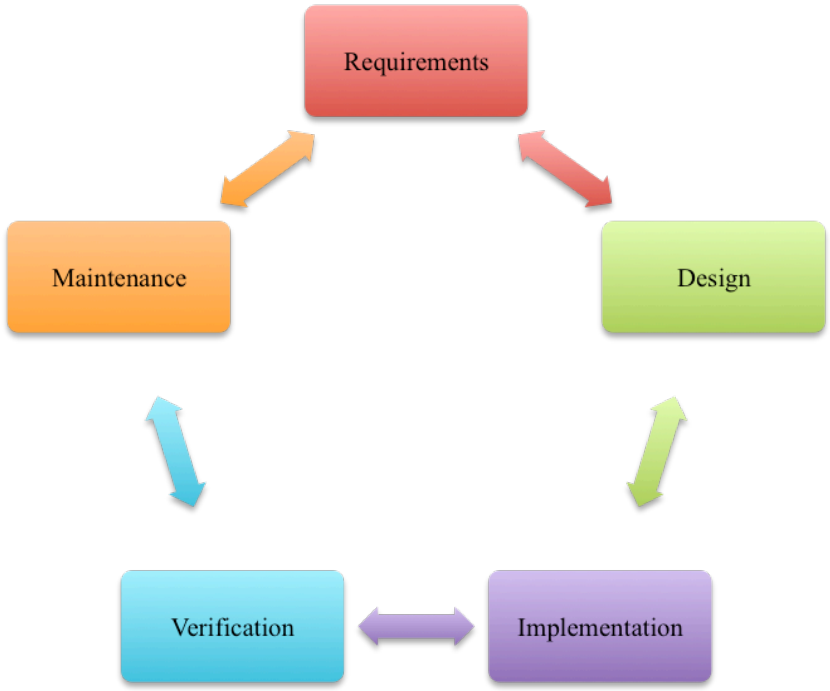


FIGURE 2: The agile method as depicted by Royce (Ruuska 2015).

A common thought is that more complex games have resulted in longer development times, hence shorter period of time for bug testing. This would be true if all the game studios used the waterfall method of development. In my interview with Lachance (2015), he states that many studios use the agile technique in their development processes, which enable developers to work on multiple steps concurrently, and have the different parties involved provide feedback to one another continuously. With this model of development, the developers can work on a game at the same time as a QA team is working on finding and reporting issues. (Interview, Lachance 2015.) A good example of simultaneous co-operation between the testing team and the developers is receiving a

new build every day or every other day. As explained in chapter 3.1, in this new build issues from the day before are claimed fixed by the developers. The issues are then regressed by the testing team, and the build tested for possible new ones. This is a great way of communicating as it provides continuous feedback between both parties.

In order to benefit from the aforementioned agile method, both the testing company and the client have to be in constant communication, and be able to rely on each other. Since testing is usually outsourced from the actual game company to a testing company, reliability is a main factor that needs to be attained through the communication channel between the testing house and the client. For example, developers often have no assets for LQA testing within their house, which is why relying on a testing company's abilities to test LQA for the client is quintessential. (Interview, Cartier 2015.)

Equally as important as the testing party's communication to the client, is for the client to inform the testing party that the information has been received and will be dealt with as soon as possible. By communicating constantly, the client in question can also advise the testing party on testing priorities, and can thus increase the effectiveness of testing. If the only communication between the client and the testing party is done with a report, the testing party will not receive information on changed testing priorities or even schedule changes in time. This causes the project to linger on, as testers cannot test effectively without good instructions, and the test lead cannot receive answers to questions that have arisen during the day. For example the test lead sometimes has to ask the client for further instructions. If no intercommunication is done during the testing hours, the answer will not be received until the next day, when it might be already too late in the game. On the other hand, the test lead and the project manager of a project also have the responsibility to communicate accurately and in a timely manner. Contacting the client as soon as issues or questions arise, being precise and truthful in the communication and acknowledging received contact is the key when dealing with clients (Interview, Cartier 2015). No matter how embarrassing an issue, or how busy the day; client communication should always be done immediately and accurately. In a worst-case scenario, false information or delays in communication could lead to an entire team being on downtime due to incompetent communication.

Another way of utilizing and benefiting from constant communication is between the teams. Oftentimes all three areas of QA – FQA, LQA and CQA – are tested at the same

time, sometimes at the same testing house. This is a good way of testing, as all three groups can be in touch through their test leads and they can communicate and ask questions about the issues they encounter. For example upon finding a functionality issue while testing the game in Hungarian, the Hungarian LQA tester in question can reach out to the FQA team to ask whether the issue has been entered, or could it be happening in Hungarian only? As functionality testers oftentimes do not test the game in other languages than English, and thus cannot test for language specific issues, the LQA testers can support the FQA team by reporting any functionality issues they find. However, the three teams sometimes work at different times, even at different testing houses or at different branches of a testing house. LQA can be tested in Canada, while FQA is tested in Germany and CQA in India. While this is not the best way of testing from a testing team's point of view, with communicating accordingly between the teams, the same questions can be brought up as if they were testing at the same studio. The same principles to concurrent intercommunication apply between the testing teams as between the client and the testing party.

In conclusion, to benefit from the agile method of development, information between the client and the testing party, or between the testing teams, needs to be accurate, precise, complete and sent on timely manner. This is especially true when dealing with critical issues that could cause submission failure or cause delays in development. The job of the testing party is to report and fix issues present within the testing scope, and to communicate properly. Failing to do so will lead to the final product's poor quality, which can lead end users badmouthing the company for its products, or even submission failures. This would be catastrophic for not only the game developer, but also to the relationship between the testing house and the client. (Interview, Cartier 2015.)

3.2 Career in Quality Assurance – Job descriptions in Company X

In Company X (pseudonym for a testing company) testing teams consist of four main groups (Figure 3). Testers are the backbone of the company as they find and report bugs. Mentoring testers are senior testers, who help testers with their inquiries and issues, in example with bug formatting. Above senior testers are test leads, who manage the group of testers and communicate with the client. Managing all of the groups men-

tioned above are project managers, who organize teams, communicate and book testing dates with the client, and take care of the project's requirements for resources.

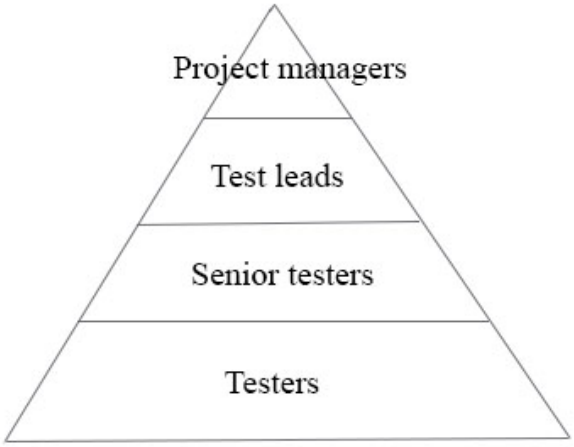


FIGURE 3: Testing team triangle (Ruuska 2015).

The common practice in the video game quality assurance field is that employees for higher positions, such as test leads and project managers, are usually hired internally, and external candidates are hardly searched for. Hence starting as a tester in video game QA could lead to a promising career within the field. However, a certain skillset is required for each of these positions.

In the following sections, these four different roles are explained as they are in Company X.

3.2.1 Tester

In her essay *Was it good for you?* Wilkinson describes testers as the group of people who are paid to bring out the bad things about a project. In video game testing, this means that testers find flaws and issues – bugs – and report them to the client. Finding and reporting bugs is what makes the tester's otherwise oppressive job enjoyable. (Wilkinson 2010, 5.)

Testers' tasks are very straightforward. They find, isolate, submit and regress bugs. Testing video games requires testers to not only be good gamers, but also curious, experimental, fearless, analytical, intelligent and most importantly – have the ability to learn fast (Wilkinson 2010, 5). In Company X LQA testers are required to work on multiple tasks at once, such as testing the game whilst searching the string database to verify the source and the translation match, communicating with the team and submitting bugs themselves. These and many other tasks in LQA, FQA and CQA require attention to detail, being logical, and having technical abilities. Additionally, testers are often required to explain technical concepts for their supervisors or even for clients. English proficiency, reading comprehension and strong verbal communication skills are thus desired. Furthermore, being autonomous is an important factor in becoming a good tester, as testers need to be able to escalate issues and request additional information when needed. (Interview, Beriault 2015.) Depending on the client's needs and schedule some minor bugs never get fixed on purpose, and it is often up to the tester to decide whether a very minor issue should be reported or not. Prioritizing which bugs get to go to the database is one of the most important tasks a tester faces every day (Wilkinson 2010, 7, 10). According to Beriault (Interview, 2015), critical thinking like this is a key feature in every tester within the quality assurance field.

Some qualities are less desired, but equally as important, and can be found especially in more experienced testers. Experienced testers are often distrustful, as they have been told over and over again that something is not an issue, and should not be tested, but they check them anyway. (Wilkinson 2010, 5.) In some cases this action could lead to great discoveries, but oftentimes it is not desired to act freely outside the client's orders and tasks for the day, as this could lead to delays and therefore cause financial issues. A skilled, experienced tester knows, however, when it is required to act slightly outside of the client's guidelines for a title's benefit.

While the abovementioned tasks might not be extremely complicated and being a tester does not require higher education, being a successful video game tester requires gaming abilities, attention to detail, critical thinking and team working skills, and for LQA testers; linguistic abilities (Interview, Chrun 2015).

3.2.2 Senior Tester

Company X uses senior testers to help a project's workflow to go as intended. Senior testers can be seen as mentors in schools; who help testers with their questions, for example with bug formatting, various databases and files for testers.

Being a senior tester requires management and communication skills. A good senior tester is someone who will put the team's needs in front of theirs. This means that a senior will often spend most of the day away from their seat, explaining a tester why testing is done the way it is, why it is important to get a bug formatted exactly as ordered by the client, or simply raising their motivation. It is essential that seniors know how each tester is best mentored. For example, for some testers it takes a lot of time to understand the game, the workflow or the meaning of their work. They get easily frustrated, and are willing to give up fast. This is where a senior tester stands by their side, helps them understand and learn the process, and at the same time make them understand they are irreplaceable in the project, so that the testers will do the best they can.

Additionally, senior testers often help the project's test lead by vetting the bugs reported. Vetting requires senior testers to verify every bug submitted, and make sure information provided in the bug is correct and formatted as per the client's request. For example in LQA bugs strings are often edited. These edits can contain grammar corrections, changing misspellings or even providing new translations. When vetting a bug, a senior will look into the string database and make sure the tester has, in fact, made the edit. In many projects there are no designated vetting seniors, who only help the lead with their work, but seniors often work as regular testers, and at the same time they are mentoring testers and helping out the lead.

All of the abovementioned facts require senior testers to be competent gamers and excellent communicators who are able to manage mentoring, vetting and testing at the same time.

3.2.3 Test Lead

Test leads coordinate testing teams for various projects. They create and implement test plans to perform testing, and brief testers to summarize what needs to be done during the day and how. Test leads are the key figures in transferring knowledge between the client and the testing team, which requires leads to be excellent communicators and managers.

As any management professional, test leads need to have a certain amount of authority to make teams respect them and understand why things are done the way they are. They have to lead an example in work ethics and remember to encourage teams especially when tasks are redundant, complicated or frustrating.

Oftentimes designated assisting seniors are not granted for projects, which is why leads are usually responsible for vetting bugs. Just as senior testers, they make sure everything is correct in the bugs, and no duplicates or invalid bugs will be sent to the client. After a bug is vetted, the lead transfers the bug to either developers or translators for them to fix.

Leads also communicate with clients to manage resources, media and testing objectives. For example, leads are in charge of receiving and requesting builds from the client. This requires constant interaction with various departments within the company to make sure enough testing devices and seats are allocated for the project, and that builds are getting transferred for the team to be able to start testing. Client communication also consists of sending reports to the client. These reports contain information on what has been done during the day, and highlight possible critical issues or pending questions. As discussed in 3.1.1, concurrent intercommunication with the client and the project manager of the team is an essential part of a lead's job.

Observing the testers and finding possible areas of improvement within an individual is also a part of a lead's job. This way everyone in the team will be the best they can be by getting help with their areas of improvement. By improving every tester's skillset and workflow they can also be provided with opportunities for growth within the company.

In conclusion leads need to be skilled communicators, great managers, and have the ability to make fast and rational decisions, as days are often hectic. They have to be able to cope with a lot of stress to provide the clients what they are paying for: an intact game as flawless as possible – within the timeframe given for testing.

3.2.4 Project Manager

Like in any company, project managers in Company X are responsible for planning, executing and closing projects. In Company X project managers are the main representatives in client communication, which requires them to be professionals in both management and communication.

Concisely, project managers discuss the needs of a client, check the resources within the company, and offer and execute testing services. In example, if a client requests testing for Spanish, French and Italian for the next four weeks, the project manager asks for testers and hardware resources from the testing house, and informs the client of the results. After an agreement has been made, project managers execute the project by forwarding information to the test lead and the testing team. After the testing is done, the project manager takes care of billing and closes the project. During the entire process project managers make sure that every party involved in the testing is satisfied with the job.

During a project, project managers are in constant communication with the client, the project's test lead and sometimes even the testers. Project managers make sure everyone is up to date on the tasks for the day and for the project. They take care of employment contracts, create tools for testing and documentation purposes, and make sure every tester in their team is coached for the tasks.

To conclude, project managers often handle various projects at the same time, which requires them to be exceptionally good in multitasking and work hard under pressure. At the same time they need to lead their team with example, advise their leads and maintain the quality of testing provided for the clients. Demanding clients, critical testing situations and employees' health and safety are all matters that make project manag-

ers in Company X true professionals with years of training and experience of management and communication.

4 THE IMPORTANCE AND IMPACT OF QUALITY ASSURANCE TESTING IN VIDEO GAMES

4.1 The importance of quality assurance testing in video games

In my interview with Chrun (2015), he mentions that one of the most common misperceptions is that video game QA is a commodity. As the video game industry grew out of infancy, increasing budgets allowed for vast expansions in QA testing. With the boom years gone, many publishers are viewing QA as simply a cost rather than as an added value. If the end user is willing to pay a higher price when also the quality is higher, then the unbiased feedback from QA is to increase the quality of the product. (Interview, Chrun 2015.)

However, financial matters and production delays are often a reason to decide against QA testing. No developer or publisher wants to give out their shelf spot to another game, or be late for the Christmas sales. (Interview, Lachance 2015.) Quality assurance is a factor that could make game companies' worst nightmares come true, but could also help games pass submission, thus reduce resubmission costs and avoid production delays (Interview, Chrun 2015). When interviewed and asked what could be done to inform new developers about the importance of quality assurance testing, Lachance (Interview, 2015) states that it is not only newer developers that are not familiar with the QA and localization processes, but also some experienced developers lack knowledge of the matter. He suggests that the best tool when relating impacts of game production decisions is using transparency early in the development process. By offering insights regarding QA or localization, QA firms help developers steer the game in a better direction and can offer solutions even before issues arise. Financially this is more sustainable than seeking help later in the development process, as it is more expensive to undo than do. On the other hand, costly choices are sometimes a great learning experience as well, but could cause bad publicity to the developer. (Interview, Lachance 2015.)

Additionally, the phrase 'any publicity is good publicity' does not always stand true with the video game industry: Publishing a broken game is something no developer wants to do – new or experienced. While a flawless game is practically impossible to make (Interview, Lachance 2015), with enough time for quality assurance testing scheduled for a

project, the amount of flaws can be decreased drastically. Nevertheless, buying a buggy and broken game is like buying a dishwasher that does not work. It spills water on the floor, leaves dishes dirty and stops working every 30 minutes. Would you buy a dishwasher from this brand ever again? A good example of a buggy game is Digital Extremes' *Star Trek: The Video Game*, which was developed for full three years before its release, and was anticipated to be a hit game. However, even after three years of development the game did not meet the expectations of the end users. It was plagued with inexplicable bugs, which made the gameplay practically impossible due to malfunctioning AI (artificial intelligence), and actions and quests that simply would not trigger (Stapleton 2013). The game was a flop: even after three weeks it had only sold 140,000 copies (Finke & Yamato 2013).

Star Trek: The Video Game did not seem to be ready even after three years, but Ubisoft Montreal decided not to make the same mistake with *Watch Dogs*, which was bound to be released for the Christmas sales of 2013. However the game got delayed, and was released in May 2014 instead. Ubisoft realized it was beneficial to let the release day slip, as this time could be used to increase the quality of the product by allowing more time for quality assurance testing. The senior producer for *Watch Dogs*, Dominic Guay, explained the delay:

> We produced an insane amount of animations and behaviours for the citizens of our Chicago. But once you do a lot of playtests you realize there are certain parts of the city where players go more than others. So look at it and we say, OK, there's all these things happening in the city that many players may never see, there's those areas they're going in, and maybe if we had more variety there it would be better. It's impossible to plan that a year ahead. You need to do it, see it, make an adjustment, iterate on it. So we actually produced more content that would fit into the areas where the players went more, moved content around a little bit, looked at it again, played it again. Iterating on this huge of a game takes a while. It takes weeks for anyone to get through our game. (Steinman 2014.)

Polishing the game, even if it means a later release date, leads to a satisfied clientele due to a more intact and complete game, thus increased publicity, and better brand image. Why some game companies decide not to further test their games, even when the games are not ready to be released could be due to the above stated reasons, such as financial and marketing matters. Sometimes the publishers decide that the game should come out on day X, ready or not, and less time for bug testing inevitably leads to more bugs ending up in the end product. An example of a recent game that was put out in a hurry due

to the aforementioned reason is Ubisoft Montreal's *Assassin's Creed Unity*. Ubisoft Montreal has made the decision to put out a yearly release of *Assassin's Creed* games due to their popularity (Maiberg 2014). *Unity* was, however, rushed out to get it in the stores in time, thus less time was designated for bug testing. Subsequently the released version contained an incredible amount of serious issues that affected the gameplay. These bugs have been very damaging for the company's image, as explained by Keza MacDonald, a UK editor of the gaming news site Kotaku:

> Assassin's Creed is Ubisoft's most important, bestselling franchise, and it's also generally the firm's most anticipated series by gamers. [...] The fact that after Unity came out, the conversation has been almost entirely about how many bugs it's got and how broken it is in certain areas, is really, really damaging for the company. (Kelion 2014.)

Based on the aforementioned facts it can be said that the claim of quality assurance not having any real value, since bugs still end up in Gold versions, is untrue. Testing is a process that takes time, and deadlines are the primary reason on why even after various testing rounds bugs do end up in games. There is more and more pressure to deliver a greater quality game at lower cost and within a shorter timeframe (Interview, Lachance 2015). Testing surely is an expenditure that might push the release dates further back, but allocating enough time for testing a game decreases the chance of releasing the next *Star Trek* or *Unity*. With the amount of money and time invested in QA testing game companies can in fact save money, and maintain their reputation that would be lost to a broken game.

In conclusion, what is important in the final product is that the game is in line or better than what the customers are expecting, not that the game is completely bug free. Investing in QA is about finding the right balance in managing both a strong financial risk and legitimate quality expectations. (Interview, Lachance 2015.), and since translation and development are expensive, investing in less expensive workforce can reduce the cost of these (Interview, Chrun 2015).

4.2 The impact of quality assurance on end user experience

An online survey was carried out to find out whether or not end users have encountered issues while gaming. The results were stunning: 97% of the 129 participants had indeed

come across issues that affected their gameplay (Figure 4). This chapter discusses the issues that are particularly affecting end user experiences based on the survey answers and Internet sources, and how much do end users think quality assurance affects the quality of the end product.

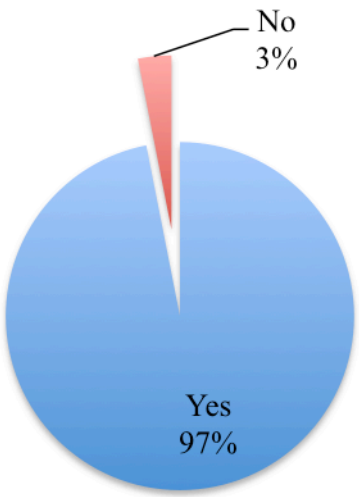


FIGURE 4: Percentage of participants affected by bugs in their gaming experience (Ruuska, online survey 2015).

According to an Internet gaming site N4G, issues that particularly affect the end user experience are to do with the game's stability and crashes. In example, an end user plays a FPS (first person shooter) game and starts attacking a big enemy with a gun that has not been in the game before. Just as the end user starts doing damage to the enemy, the frame rate of the game drops, the picture starts fidgeting, and eventually the whole game freezes and crashes. Crashing means that the software quits unexpectedly, and all unsaved data is lost. Other issues that affect the end user experience greatly are for example font issues (font too small to read, text running outside the screen, text overlapping with other text etc.); collision issues (invisible walls, characters being able to fall off the map etc.); problems with the AI, such as the NPCs (non-player characters) ignoring the player character, or the NPCs being able to execute tasks that the player character cannot for no obvious reason; and balance issues, such as the player dying from one knife wound while wearing a metal armour. (Most annoying bugs in... 2010.) All of these flaws are possible to be found during testing rounds, given that enough time is allocated for the testing for FQA, CQA, and LQA if the game is translated.

Performance issues were not liked among the survey participants either, as 33% of the participants found crashes or hangs the ones that affected their gaming experience the most (Figure 5). Design issues, such as inability to access a certain area, interacting with objects does not trigger an action, game controls difficult to handle, or action permitted when it should be and vice versa, took the second place with an astounding 19% of the answers. Translation issues were only considered affecting by 2% of the participants. This could be due to the fact that translations are generally well made and tested, or because most of the survey participants play games in a game's native language, usually English.

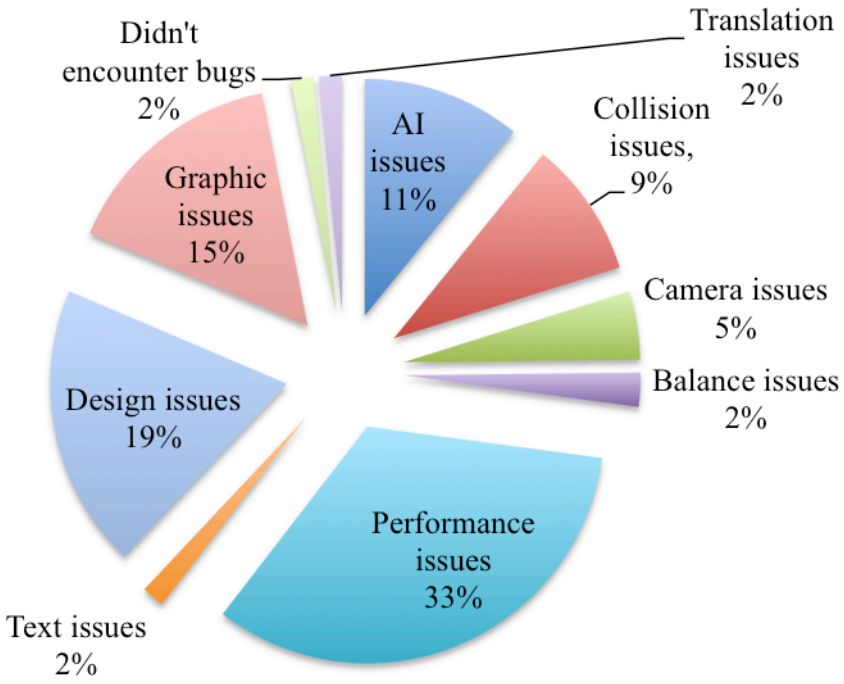


FIGURE 5: Issues that affected the participants' gaming experience the most (Ruuska, online survey 2015).

Why is end user experience so important then? Would it not be enough that the gamers simply buy the game? Unfortunately, no. End users are the pillars keeping game companies up. End users are like fans of a band: they affect the feedback and thus the reputation a game company receives. They can choose to pay for the company's next game, if they liked the previous one. Based on the survey, 39% of the participants that had encountered issues while playing stated they did not mind the minor issues, but the ma-

jor ones affected their image of the game company. 30% of the participants answered they were not impressed with the game's quality, but hoped for the next one to be better (Figure 6).

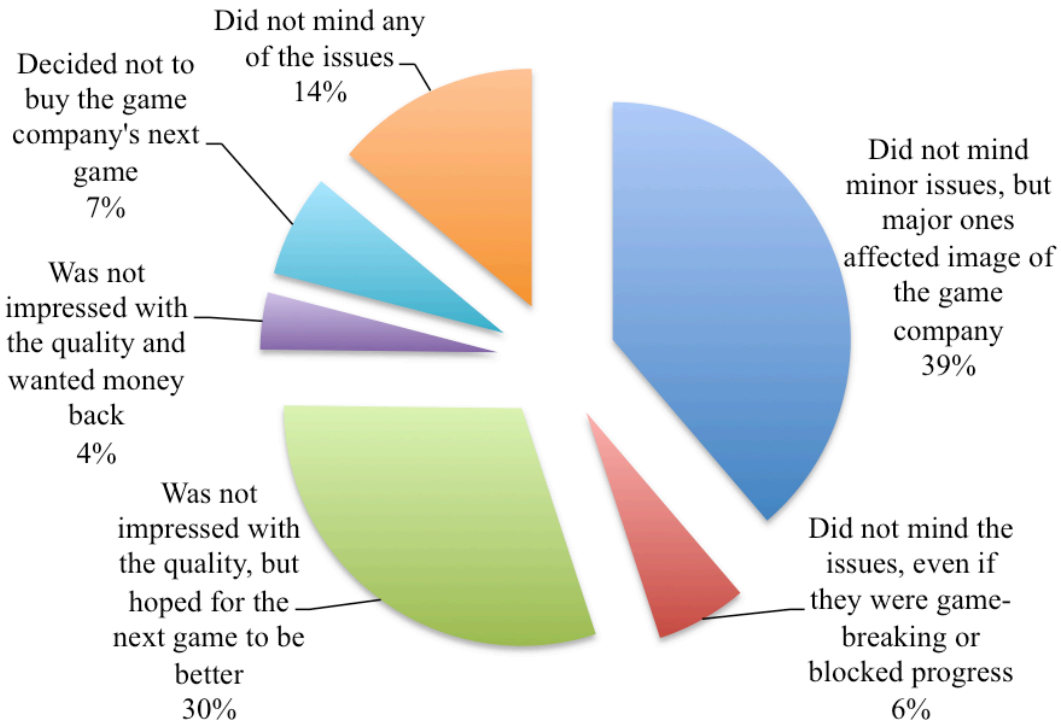


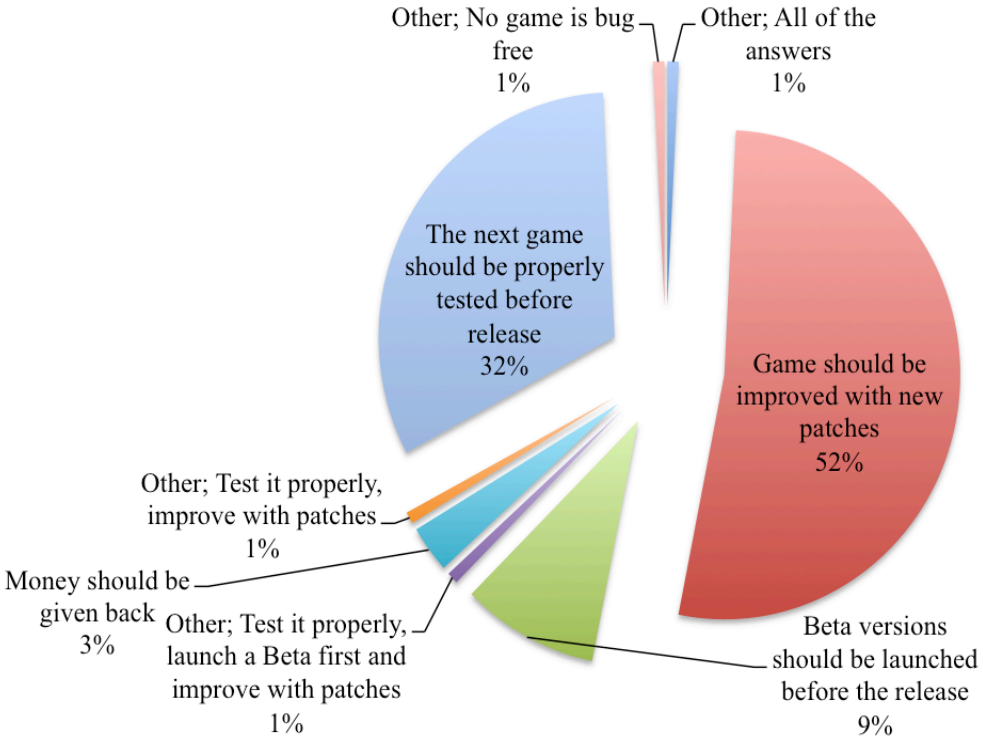
FIGURE 6: Participants' impression after encountering bugs in a game (Ruuska, online survey 2015).

End users have the power to go online and blast negative comments all over the Internet about the game and the company, which is why good end user experience is essential especially for smaller companies. Big companies such as EA Games are not expected to go under after one bad game, but even Atari had to call it off after a few flops and some misfortune (Santoso 2008). Gamers are the ones with the power to decide which game is the next hit, which one is a miss.

Fortunately, gamers these days are increasingly understanding of the complexities of game development and the difficulties of hitting release dates and making a bug-free game (Interview, Lachance 2015). As depicted on Figure 6, only 7% of the participants decided not to buy the company's next game, and only 4% wanted their money back. 14% of the participants did not mind any of the issues (Figure 6). Based on the survey,

gamers seem to be forgiving when it comes to releasing a mildly buggy game as well; 52% of the participants say the game should simply be improved with new patches (Figure 7).

FIGURE 7: What should be done if a buggy game is released? (Ruuska, online survey



2015).

Survey participants also were asked whether they thought quality assurance testing was an important part of game development or not. An astonishing 82% of the participants chose to answer that quality assurance testing was, indeed, an important factor in game development, even if it sometimes causes delays in release dates (Figure 8). 13% recognized quality assurance testing as an important part of game development, but did not think it is a valid enough reason to cause delays in the game's release (Figure 8). The rest of the participants thought QA testing was either somewhat important, or were not sure what to answer. Only one person answered that quality assurance testing was not important at all. However, this single participant also stated that they did not mind the minor issues, but the major ones affected their image of the game company, and a buggy game should be improved with new patches. But how does one improve a game if not by testing it?

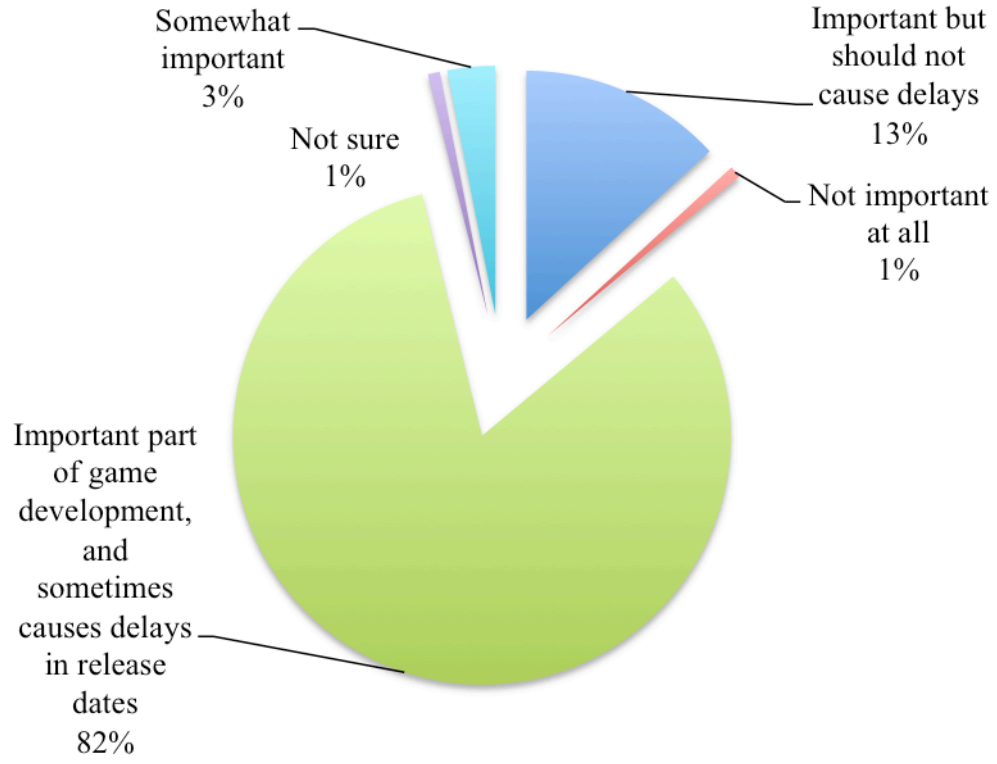


FIGURE 8: How important is quality assurance testing in game development? (Ruuska, online survey 2015).

To better understand the backgrounds of the survey participants, Figure 9 depicts how many of the 129 participants work or have worked in the video game industry. As a result 66% of the participants answered they had never worked in the industry, and 11% had worked as developers. Another 11% of the participants had worked as testers. Therefore we can see that most of the participants were regular end users with no experience of game development or testing (Figure 9); people whose experiences game companies are so concerned about. However even these participants found quality assurance testing, Beta versions and patches an important factor in improving a game's quality (Figure 7) before and after release. Based on the survey the fear of end users not being forgiving about letting a release date slip is thus most likely exaggerated; gamers seem to be willing to wait a little bit more in order to get what they pay for.

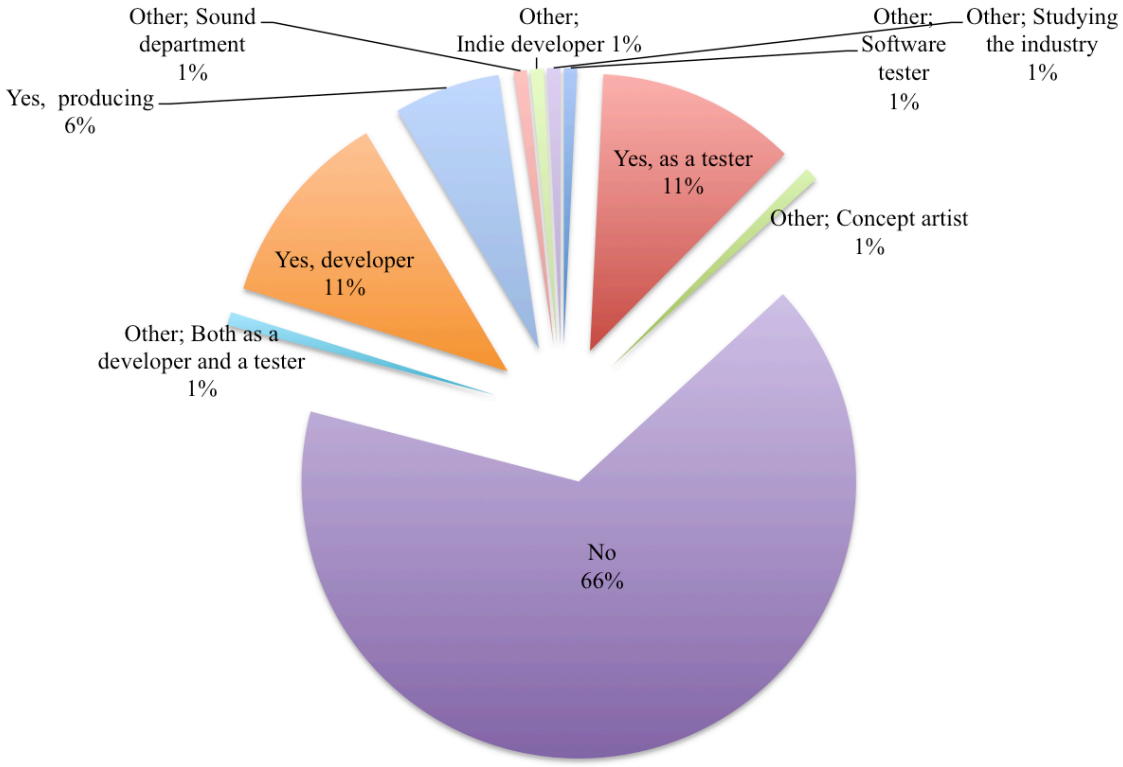


FIGURE 9: Amount of participants working on the video game or quality assurance industries (Ruuska, online survey 2015).

If the data from all of the figures is combined, we can draw a conclusion that gamers do not mind receiving a game with minor issues, and even these can be improved with new patches for the game (Figures 6 & 7). From combining the data we can also see that approximately 30% of the participants hope that the company's next game will be properly tested before its release (Figures 6 & 7). By looking at Figure 8, it can be seen that end users are aware of quality assurance testing as an important part of game development, and also understand that games are sometimes pushed back for further testing and polishing in order to improve the quality of the end product (Figure 8). Figure 9 declares that most of the participants did not have work experience on the video game industry.

As stated before, based on the survey results, game companies should avoid publishing a broken game on a specific release date, or be concerned with end users not under-

standing game development. On the contrary they should focus on delivering an intact, functional and complete game that end users are paying for.

5 CURRENT ISSUES AND FUTURE OF VIDEO GAME QUALITY ASSURANCE INDUSTRY

5.1 Evolution of technology

According to Entertainment Software Association of Canada (2009), Canada's entertainment software industry is facing certain challenges, as new technology is brought to the market. With new technology and platforms new issues may arise, and all the new hardware has to be tested before they go into stores (Calvin 2014).

Unlike the common perception, new platforms such as PlayStation®4 and Xbox One have not resulted in more issues due to developers not being comfortable with the platforms. The tools used for development are still either very similar or within the same development family. Moreover, stability on the newer consoles and systems is better than on the older generation ones due to the better capabilities of multiplatform development. As an example, some major issues that would break the game on PlayStation®3 can become minor or even completely absent on PlayStation®4. (Interview, Lachance 2015.)

However, it has not been long since dancing games came out, and with them a number of issues concerning the users' health and safety (Calvin 2014). Additionally newer devices, such as Oculus Rift and Morpheus, need to be tested, since the use of them might give certain users motion sickness or other symptoms. While not software issues, these are all factors that need to be taken into account both for employees' and end users' safety. (Interview, Lachance 2015.) Failing to pay attention to every detail during testing, whether in hardware or software, can lead to submission failures, prolonged projects, financial issues and legal matters.

For CQA, the arrival of new hardware like Morpheus for PlayStation®4 brings new official terminology, and thus needs to be respected in all new software developed. With every new platform comes new tools and operating system features used for CQA testing, resulting in an entirely new processes and methods of testing – and terminology is only a tip of an iceberg in this matter. Hardware malfunctions, such as instability are common upon releasing a new platform, and the hardware manufacturers will judge

these malfunctions as acceptable or unacceptable. This will lead to updating or creation of new requirements for the new platform, in order to cover the newly discovered edge cases. (Interview, Beriault 2015.)

In conclusion, new devices and platforms have not resulted to new major issues, but as in all development, there are certain things to watch out for.

5.1.1 The threat of emulating devices

As the amount of new platforms has increased, testing companies are relying on emulating and testing devices. Many wonder if investing in QA really matters, since the testing environment and the platforms are not the same as that of an end user's. (Calvin 2014.)

Emulating devices are devices that emulate a gaming platform. An emulator can be software installed on a PC, which simulates the settings and capabilities of the original platform, and can thus be used as an emulating device for the desired platform. In his article Calvin (2014) declares that emulating devices always pose a threat of the emulator itself causing an issue. In the worst case a client receives an invalid issue that does not occur because of the game, but because of the device – and critical issues on the actual devices are missed. (Calvin 2014.) Emulators are a cheap but not a very reliable way to simulate a retail gaming platform. Fortunately for developers and publishers very few testing houses depend on emulators anymore.

Instead of emulating devices many testing houses use testing kits for testing games. Testing kits are platforms that are almost like the retail versions of the platform in question, but they come with some additional features, such as debug. They are made to run builds with additional capabilities, and are designed to record data that the user does not see, such as crash logs, performance information, etc. These kits are not exactly money savers for testing houses, as they cost much more than retail versions. However, testing builds would not launch on retail kits, which requires testing houses to invest in testing kits. In a nutshell testing kits allow the testing team to use tools the end user is not meant to have, such as debug, crash logs and system performance information, but these tools can be turned off, which leads testers to see exactly what end users see. (Interview, Chan 2015.) There are still minor differences between the testing kits and retail, namely

the ability of the game servers to handle increased traffic, and the availability of additional content on the retail virtual stores cannot be verified. However, some of these differences can be tested for. For example in CQA, testing is done during the submission process at the hardware manufacturer, in order to ensure the least amount of issues upon release of a title. (Interview, Beriault 2015.)

In today's world there are hundreds of platforms on which games can be released. Emulating devices can pose a threat of causing issues retail versions would not have, but with these devices more can be done with less money. However, emulators are not as trustworthy as testing kits, which provide the tools that are also quintessential in the testing process, for example launching the builds in progress would be impossible on retail kits, and simply playing the game in order to make progress would be time consuming and ineffective. With testing kits testing can be made easy and efficient, as the game can be stopped, milestones are easier to reach and matches less time-consuming to win. Therefore the talk that emulators pose a threat to testing is partly true, but does not apply to real testing kits that are used in the majority of the testing houses. Hence game companies should not worry about QA testing not being worth it just because the testing environment is not exactly the same as an end user's living room.

5.2 Future

In the future game development is said to become more professional than the film industry. Game development has its roots in software development, but in the future it will most probably outpace film and television, and become the major form of entertainment. (Levy & Novak 2010, 208.)

As the vast majority of the world's population will be playing games as their primary source of entertainment, game development will be seen as a strong pillar in global economy, and thus testing jobs will get stabilized. On top of this game testing will become paramount due to the cost of current and future game development. Whereas in the 1990s $100,000 was a big budget for a game, nowadays game development costs millions of dollars. (Levy & Novak 2010, 211.) With the development costs going up, the games become more expensive and testing will become more and more important, as no end user wants to pay more money than before for a broken piece of game. As

games become more expensive and more sophisticated, the role of QA will become even more important, rigorous and professionalized. (Levy & Novak 2010, 209.)

The future of testing seems to be in good hands, but the ever-growing amount of new platforms and technology will bring new difficulties to testing procedures. For example the sudden boom in mobile platform games has resulted in many more projects and opportunities for developers, publishers and testers. According to the report from ESAC (2014) the Canadian video game industry grew by 5% in the number of employees between 2011 and 2013. 2 out of 5 companies predicted to grow by 25% in the next two years. The president and chief executive officer of the Entertainment Software Association of Canada, Jason Hilchie, said he expected the video game industry to keep growing worldwide:

> We have a good sense that these types of jobs will be the jobs of the future, because of the growth in the popularity of the interactive entertainment industry all over the world. The video game industry is the fastest growing entertainment industry globally. The global market is currently estimated to be approximately $67 billion U.S. That's bigger than the box office revenues for movies. With a growth rate of 7.2% annually, this industry will be worth $83 billion by 2016. (House of Commons 2013.)

While more job opportunities are always a good thing, one cannot help but wonder how is a tester going to master all of these new platforms and games? Should testers start getting trained for one specific platform only? *Game Development Essentials: Game QA & Testing* (Levy & Novak 2010) has a potential answer to the high demand of skilled workforce within testing companies: Having testers specialize in a certain area of a game. This way teams could be assembled with testers good at fighting, others good at spotting audio flaws, etc. This technique is already in use in longer and bigger projects, where there is room and time for testers to specialize in a certain area of the game. Within a game, this may work, but the reality is that testing companies receive hundreds of various games to be tested, and a person who is good at fighting in an FPS game might not be suitable for a strategic game. As tempting and reasonable as the proposition may sound, this could end up in a mess within testing companies, as specializing in only one or two things makes testers technically unusable for any other platform. Imagining that suddenly someone from a major project got sick, and no replacements were found within the platform's specified testers, the testing house would have to bring in someone who does not know the platform or the genre at all. This could result in longer

times in finding, analysing, reproducing and reporting bugs. Ineffective testing such as this is not desired, as clients have certain expectations and tasks assigned for the day. If these tasks cannot be completed within the timeframe given, such as an inexperienced tester possibly could not, the testers and further the testing company would receive bad reputation, which would eventually lead to loss of clientele.

Another way of solving the issue could be less rigid. Testers could specialize in a type of genre, on a platform or even a client, as every client has their own requirements for testers, but they would also be cross-trained for other purposes, such as for different genres and platforms, or from FQA to CQA and from LQA to FQA, etc. This way testers could become especially skilled and experienced in certain areas of testing, while still having the ability to work on other tasks as well. This method is already used in certain projects in Company X.

A quality assurance company Testronic has also found another way of utilizing testers by using them for customer service during low season. By utilizing this method game companies can cut down their cost of customer service, and receive the service for cheaper by gamers who are already familiar with the game in question. (Calvin 2014). This can also cut down the cost of holding employees sitting on downtime at the office, and provides the employees with the opportunity to learn a new set of skills that is useful for any kind of job: communication and customer service.

We cannot foresee the future, but we can predict certain patterns in the upcoming video game sector as well. As mentioned above, more and more people are going to switch from traditional media like newspapers, TV and radio to video games (Levy & Novak 2010, 208-209), which means that games are going to develop as well – not only by their software, but also by their content. In the future we might see more and more video games with stronger and less subtle messages, combined with newest events from around the globe and the ability to keep up with the world through video games. Whereas today's video games are often considered to be addictive and causing social withdrawal, maybe in the future video games will be competent and respected ways of communicating and sharing information. Imagine a game in which the events are based on real-time and real-life events from around the world, and by playing both the good guys and the bad guys the user will be able to understand both sides of the coin. Imag-

ine a game where watching TV actually means you are watching real-time television. Combining all the media could be a real goldmine for video game companies.

But with new ways of gaming we can assume new ways of breaking the games will also be involved, and the constant support from quality assurance could become inevitable. More complex bugs would require more experienced testers to solve them, and more developers to fix them. With the requirement for more workforce, the importance of experienced testers will increase, as these testers are not only efficient and fast in their work, but are also capable of helping newcomers out with their tasks.

The future looks bright but complicated for quality assurance and the whole video game industry. How are quality assurance companies going to keep up with the speed of new technology, new games, new everything? Keeping experienced testers on board and hiring testers that are easily adaptable and quick learners would be a good start. Along with these, recycling testers as much as possible from one platform and game genre to another should be helpful. With keeping testers aboard and having chances in training them in various platforms, areas of testing and genres of games the testing companies could rest assured that in time, these testers would become the best they can be in any kind of game, on any given platform. And by doing this, the future of video games should be somewhat secured. Just like Rome was not built in a day, the video game industry is not going to change overnight. But preparing for the day everything will change should begin right about now.

6 DISCUSSION

Quality assurance has seen its boom years, but understanding of game development has provided quality assurance a firm base in game companies' eyes. The importance and the effectiveness of good, professional quality assurance testing has been noticed, and understanding the impact of a quality assurance testing in video games has lead game companies to put their trust more and more in testers' hands. This trust between the parties is based on efficient testing that delivers great results, and accurate communication, which is the key to fluent game development. While concurrent intercommunication during every day of testing is at times impossible due to time differences and tight schedules, negotiating the time needed for testing, calculating a budget that satisfies everyone, and using the agile method of game development, the preservation of the high quality of the games tested can be assured.

All areas of quality assurance – LQA, FQA and CQA – aim for the same purpose: to detect and report defects in order to improve a game. QA testing is a cheap and efficient way to make sure the game in question meets the requirements and is ready for release, without the fear of it being broken and backfiring due to angry end users. However, even with the tools provided for testing – testing kits, debug etc. – the time seems never to be enough to cover all the areas requested, and bugs always seem to make their way into a final product. As mentioned, Lachance (Interview, 2015) states that the meaning of quality assurance testing is not to make sure every single game is entirely bug free, but that these games meet and top the expectations of the end users. By bringing quality assurance in the project as soon as possible the risk of fatal mistakes decreases, and game breaking or submission failing defects can be detected more efficiently and less expensively. QA testing is therefore not only important to its developers and publishers, who inevitably will receive money and fame from delivering a great game, but also to the end users who pay top dollar to receive an intact game in a timely manner.

One of the biggest challenges in both video game development and quality assurance is the ever changing and improving technology. Technology surely is something game makers are slightly scared of, as new technology brings out new issues and new requirements both on hardware and software level. The question is, however, whether end users will understand that adjusting to the new technology takes time, and it is practical-

ly impossible to create flawless games in a short notice for all of these new platforms. Based on the survey it seems that end users are more forgiving than we imagine, and as mentioned in 5.1, just as the bugs from PlayStation®3 did not actually transfer to PlayStation®4, we can expect that the new platforms are going to be more stable, have the ability to execute more and more complicated tasks in a shorter time, and display computer graphics like never before.

What is interesting is to ponder when is enough going to be enough? Are we going to create games so live like you can almost feel them on your skin? Are end users going to grow numb to the greatness of these games? Is gratitude and respect going to be even harder to gain? Is the video game industry going to be an elite field in which only the best of the best will succeed, or will more and more indie developers come out and try their luck?

No one probably knows the answer to these questions. But with this image in mind, can we expect that quality assurance and coding video games are going to be professions of the new generations? Like miners and farmers once were, can we expect the future to revolve more and more around video games and their makers? I believe so, yes. Even if the future does not go to the extent described in chapter 5.2, I do believe we are going to play more, we are going to expect more from games, and we are going to demand more from these games and from the upcoming platforms. Just like black and white TV or dial up connections are now history, games with the features they have nowadays will soon be in the past as well. With this new, grand era to come, we can expect the importance of quality assurance testing to boom once again. We can expect more and more job openings, more and more specialized career paths to open up, and more and more people from various backgrounds such as linguistics, arts and IT to join the workforce. In the future testing jobs might become more permanent than they are today, as the need for skilled, experienced testers increases. In the future testers might be considered just another group of people working on IT and media instead of an odd bunch of dirty gamers as they often are perceived now. Maybe in the future even more people who start as testers will end up as project managers, or even in higher positions. I believe that from today's testers many will succeed and pursue a successful career in the video game industry.

The future of the video game industry could be an interesting topic for a longitudinal study, as it could include the evolution of new technology, issues, workforce and types of games. We could study what kind of people are involved in the game business today and in 10 years, and the attitudes towards game jobs. We could study the reform from old technology to new, and the development of issues from today to the future. What kind of games will we have in 10 years, how about in 20? What does testing look like in 10 years? With this study we could provide information about the future to video game and QA companies – like fortune telling but for business purposes. The study could be beneficial for all these companies to learn what lead to massive lay offs back in the day, what to avoid, and what to expect.

As for this research, the limited amount of time that I had for doing this thesis is definitely something I consider a possible problem when it comes to the reliability of this research; I did not have the time to interview people from the other side of game development, such as developers, producers and game designers, but rather focused on interviewing people who in fact work on the QA field. If there was enough time, there would have been even more interviews that could have contributed in making this research even more reliable. The same goes for the online survey, which provided me with 129 participants. If there had been more time, the number of these participants could have gone up, and the questions could have been even more in-depth than now. The problem with the survey especially seemed to be the option to answer with open comments, which lead to some of these participants answering something that did not help me with my research at all. However, the number of these people who decided to give unhelpful answers was very low compared to the number of serious participants. The clearly not serious answers were also left out from the charts and diagrams.

However, I do feel that this research is rather reliable and ethically performed. All interviewees were asked whether their name, company and occupation could be listed in this thesis, to which all of the interviewees answered yes. All survey participants were informed their answers would be anonymous. Moreover, due to signing the non-disclosure agreement with my previous employer, it was thoroughly discussed which topics and how much of each topic could be disclosed in this thesis, so as not to reveal too much of my previous employer or any other video game company. It was essential for me not to display any of the mentioned companies, people or industry in a bad light that could damage any of the participants, their business or their reputation. I would like

to thank everyone who contributed to this thesis as an interviewee, survey participant, instructor, proofreader or as mental support.

I suggest everyone to start perceiving both video game and quality assurance jobs with a new vision, as I truly believe this is the entertainment form of the future, and that it has great potential for incredible growth. I hope this thesis has provided with a little bit of insight to what game testing is now in relation to game development, and that the importance of quality assurance testing has become at least a little bit clearer. For those who would like to pursue a career in game testing, I wish luck and success. Testing is definitely not as glamorous as it seems, but it provides testers with the opportunity to grow within the industry and create contacts from the field. And, to be honest, it is a fun and challenging field to work at. Additionally, I encourage game companies to take some time to get acquainted with quality assurance testing if they are not that already, and to visit testing houses if in doubt. They will see that game testing is something most testers take pride in, as they want to help make video games the best they can be.

We do not know how the video game industry is going to look like in the future, but for certain we know that video games are the entertainment form of the century, and that quality assurance is going to be required, and maybe even understood and appreciated more. With all these said, I hope video games will be more appreciated both as a form of entertainment and as an art form, as I claim that game developing, designing and testing are truly forms of art. And art belongs to everyone.

LIST OF REFERENCES

Atari: Game Over. 2014. Direction: Penn, Zak. Production: Fuel Entertainment USA, Lightbox, Red Box Films. Production country: USA.

Beriault, R. Certification Quality Assurance Project Manager. 2015. Interview 4.3.2015. Interviewer Ruuska, E. Montreal.

Calvin, A. 2014. QA and localisation: What's bugging the bug testers? MCV. Read 3.23.2015. http://www.mcvuk.com

Canada's Entertainment Software Industry. The Opportunities and Challenges of a Growing Industry. 2009. ESAC. Ottawa: Hicklings Arthur Low Corporation.

Cartier, M. Localization Quality Assurance Project Manager. 2015. Interview 4.12.2015. Interviewer Ruuska, E. Montreal.

Chan, E. Localization Quality Assurance Test Lead. 2015. Interview 3.21.2015. Interviewer Ruuska, E. Montreal.

Chrun, O. Localization Quality Assurance Manager. 2015. Interview 3.19.2015. Interviewer Ruuska, E. Montreal.

Collins, H. 2010. Creative Research: The Theory and Practice of Research for the Creative Industries. Lausanne: AVA Publishing SA.

Essential facts about the Canadian video game industry. 2014. ESAC. Read 3.26.2015. http://theesa.ca

Finke, N. & Yamato, J. 2013. A Tale Of Two Movie Tie-In Games: 'Star Trek' Flops, 'Fast & Furious 6′ Tops. Deadline. Read 5.4.2015. http://www.deadline.com

Lachance, M. Functionality Quality Assurance Manager. 2015. Interview 3.20.2015. Interviewer Ruuska, E. Montreal.

Kelion, L. 2014. Ubisoft apologizes for Assassin's Creed Unity's bugs. BBC News. Read 5.5.2015. http://www.bbc.com

Levy, L. & Novak, J. 2010. Game Development Essentials: Game QA & Testing. Clifton Park: Delmar.

Maiberg, E. 2014. Ubisoft: We'd be stupid not to release Assassin's Creed every year. Gamespot. Read 5.5.2015. http://www.gamespot.com

Pérez, C. I. 2011. Linguistic Quality Assurance: Essential for Successful Language Translation. GPI Translation Blog. Read 3.19.2015. http://blog.globalizationpartners.com

Royce, W. W. 1970. Managing the Development of Large Software Systems. Proceedings. Los Angeles: IEEE WESCON, 328-338.

Santoso, A. 2008. The Rise and Fall of Atari. Neatorama. Read 4.28.2015. http://www.neatorama.com/

Stapleton, D. 2013. Star Trek: The Game Review. I cannae take much more of this! IGN CA. Read 5.4.2015. http://ca.ign.com/

Steinman, G. 2014. Why was Watch Dogs delayed? Ubiblog. Read 4.28.2015. http://blog.ubi.com/

Thang, J. 2012. The tough life of a games tester. IGN CA. Read 3.23.2015. http://ca.ign.com/

The Canadian Software Entertainment Industry. 2013. Report of the Standing Committee on Canadian Heritage. Ottawa: House of Commons.

The most annoying bugs in video games. 2010. N4G. Read 4.14.2015. http://n4g.com/

Wilkinson, L. Was it good for you? Riley, T. & Goucher, A. (ed.) Beautiful Testing. Leading Professionals Reveal How They Improve Software. Sebastopol: O'Reilly Media Inc, 3-13.

Yin-Poole, W. 2013. Microsoft unveils the Xbox One indie developer self-publishing program. Eurogamer.net. Read 4.15.2015. http://www.eurogamer.net/

APPENDICES

Appendix 1. Research questions. **(1/3)**

1. What is video game FQA, LQA or CQA?

2. How are these different from one another?

3. Do all these different areas of QA have something in common?

4. Who are LQA, FQA and CQA for?

5. Why is QA testing important? What is the impact of good QA versus no QA done one a product? What could be the biggest issues in
    a) Submission
    b) User experience
    c) Something else?

6. What are the submission failing issues in LQA, FQA or CQA?
    a) Why don't all reported issues get fixed?
    b) Why is failing submission a bad thing for game development?

7. Should every game developer invest in QA testing?

8. It is impossible to create a flawless game?

9. When is the right time to release a game?

10. What are the benefits and dangers of pushing a game back for polishing?

11. What are the reasons for which some decide not to have their product tested?

12. What can be done to inform developers and publishers about the importance of testing?

Research questions. **(2/3)**

13. What should game developers know about investing in LQA, i.e. how much time is usually needed for an AAA title vs. mobile game, why a Canadian French tester should not test European French, etc.? Can QA ever be tested entirely as the testing devices cannot completely emulate an end user device or experience?
    a) How is an end user -like testing environment created in the office?
    b) How is terminology tested in translated games, as not all LQA testers are trained for CQA and the other way round?

14. How do new devices and platforms such as Oculus Rift or Morpheus affect QA testing?
    a) New terminology?
    b) Platform specific issues?

15. Have more complex games resulted in longer development times and thus less time for QA testing?

16. Game testing is often glorified a lot. What are the biggest misperceptions of the video game QA industry and the testers' job description?

17. What are the qualities a good tester has?

18. How is good client communication created and maintained?
    a) Is it better to keep in concurrent intercommunication, or does communication always depend on the project?
    b) Are there any main principles in client communication?

19. What is the biggest mistake that could happen in intercommunication between the testing house and the client?

20. What are the biggest issues on video game QA industry at the moment?

Research questions. **(3/3)**

21. Anything else you would like to add?

Appendix 2. Online survey questions. **(1/3)**

How often do you play video games?

a) Every day
b) Every other day
c) Few times a week
d) Few times a month
e) Occasionally

How often do you buy video games?

a) I buy basically all the new games when they get released
b) I buy some new games when they get released
c) Everytime there's a Steam sale
d) Occasionally
e) I don't buy games

What kind of games do you enjoy the most?

a) Strategic games and building games, such as Minecraft or Limbo
b) Shooters, such as Counter Strike or Call of Duty
c) Story-based, such as Heavy Rain or Beyond: Two Souls
d) MMOs, such as WoW
e) Fighting games, such as Mortal Kombat or Street Fighter
f) I don't have a type
g) Other, please specify:

Which platform do you prefer when gaming?

a) PC, Mac or Linux
b) PlayStation®3
c) PlayStation®4
d) Xbox One
e) Xbox 360
f) Wii
g) Other, please specify:

Online survey questions. **(2/3)**

Have you ever encountered bugs that affected your gaming experience?

a) Yes
b) No

What kind of bugs affected your gaming experience the most?

a) Graphic glitches, such as graphics not rendering, missing graphics, placeholders
b) AI issues, such as derpy or unresponsive NPCs
c) Text issues, such as font too small or too big to read, overlaps, cut offs, incorrect information, missing instructions etc
d) Balance issues, such as game too easy to play even when selecting the highest difficulty level and vice versa
e) Collision issues, such as invisible walls, going through walls, falling off the map
f) Camera issues, such as frozen camera, wrong angle etc.
g) Design issues, such as inability to access a certain area, interacting with an object does not trigger an action, game controls difficult to handle, action not permitted when it should be and vice versa
h) Performance issues, such as crashes or hangs
i) Translation issues, such as bad translation, incorrect translation, wrong language, grammar or spelling mistakes etc.
j) I didn't encounter any bugs

How did these bugs affect your image of the game company in question?

a) I decided not to buy their next game
b) I wasn't impressed with the quality and wanted my money back
c) I wasn't impressed with the quality, but hoped for their next game to be better
d) I didn't mind the issues, even if they were game-breaking or blocked my progression
e) I didn't mind the minor issues, but the major ones affected my image of the company
f) I didn't mind any of the issues

Online survey questions. **(3/3)**

If a buggy game is released, what should the game company do in order to make it up for the end users?

a) They should give us our money back
b) They should improve the game with new patches
c) They should make sure the next game is properly tested before release
d) They should launch Beta versions before the release
e) Other, please specify:

Do you feel that quality assurance testing is important for the end product?

a) It's an important part of game development and sometimes causes delays in release dates
b) It's important but shouldn't cause delays
c) It's somewhat important
d) It's not important at all
e) I'm not sure

Have you ever worked on the video game industry?

a) Yes, on the development side
b) Yes, on the producing side
c) Yes, as a tester
d) No
e) Other, please specify: