Automated Game Testing and the Tester: A Search for Meaning

Nathan Hahn

George Mason University

May 18, 2020

Contents

Abstract

The rise of automation suggests reflection on the value of the human touch in many industries, and in game development the rise of automated testing could potentially change the way game testers relate to their work. Trends in automation may be welcomed by some testers who desire more challenging work, but cause hardships for others unable to adapt to those changes. This paper examines how automated testing changes the roles and responsibilities of testers on game development teams, and whether those changes contribute towards making game testing more or less meaningful as a human work endeavor.

Automated Game Testing and the Tester: A Search for Meaning

Introduction

In the early days of video game development, games were created by individual generally knowledgeable individuals balancing the various tasks of game development on their shoulders. But as games have grown more complex in terms of systems and larger in terms of content, the teams that are required to make those games have become more complex, and roles more specialized. While the primary roles of the game development team have received academic attention in terms of their operations and the goals of their work, the role of game tester on these teams has received relatively less attention. The role of a game tester deserves attention and analysis because the role of game tester is commonly perceived as a gateway to other roles in the industry, but the lack of knowledge on the meaning of the game tester's work on the game design team can result in misuse of testing resources by the development team as well as disillusionment in the people hired as games testers hoping to break into the industry.

Because the tester is not viewed as a central creative role, the activities that the tester performs might undergo a change in the same way that the software delivery process has been transformed by software delivery automation technologies. This work seeks to unmask the meaningfulness of testing work and determine how automation will transform the meaningfulness of the testing role within the game development team.

Meaningful Work

A definition of meaningful work must be provided prior to determining whether automation in games testing will increase or decreasing the meaningfulness of work as a game tester. By looking at an overview of meaningful work, as well as meaning within work from the game development perspective, a model will be established that can be used to evaluate automation and game testing to determine whether meaning is added or subtracted through the work automation process for game testers.

Overview of Meaningful Work Perspectives

The concept of meaningful work is broad and can be tackled from many different perspectives. Researchers often grapple with the difficulty of determining subjectivity and objectivity within meaningful work. In addition, it is difficult to determine the actual impacts of meaningful work through developing a systematic analysis.

In her book *Meaningful Work*, ethical and political philosophy professor Veltman (2016) uses both angles and states that meaningful work can both impact the workers subjectively through the process of working and the world around them objectively through their impact on history. By summarizing other sources on meaningful work, she identifies four key dimensions in her model. She identifies work where the worker's capabilities are exercised and rewarded with recognition and esteem, work that supports virtues and honest morality, work providing purpose for self or others, and work integrating the worker's life, environmental, or relational context. In combining all these different traits, she acknowledges the over-inclusion of meaning, as nearly all work has at least some part of one of these elements, and she identifies that occupations that seem meaningless are often either jobs with indirect effects or jobs with low social esteem.

Lips-Wiersma et al (2018) create a definition of meaningful work based on connecting individuals effectively with their organization with seven dimensions, composed of four pathways, two tensions, and one overall context. The four pathways are "Integrity with Self, Unity with Others, Service to Others, and Expressing Full Potential." The two tensions are between the sense of self and meeting needs of others, and "the need for Being (reflection) as well as the need for Doing (action).” The overall context of meaning is placed within both inspiration towards an ideal, and realistic reflection of current circumstances.

In performing analysis through an extensive study of research on meaningful work as faculty of practical theology, Martela and Pessi (2018) describe meaningfulness through three elements: work significance, self-realization, and broader purpose. Significance is the individual’s perspective on their own work and how much value it provides, self-realization is how much individuals can express themselves through their work, and broader purpose is the extent to which their work impacts the world around them.

As a professor of counseling psychology, Allan (2017) studied more direct impacts to work results from perceived meaningfulness. His research determined that task significance impacted meaningful work, but this was also dependent on subjective responses from the individual, based on perceiving work as improving the welfare of others results in subjective feelings of meaningfulness in work. When discussing meaningful work from the business management approach, meaningful work is used as an input for worker productivity, not an end of itself.

Professors of psychology Jiang and Johnson (2017) attempted to use a systematic methodology to evaluate the relation between affective commitment and meaningful work. Affective commitment is relevant in game design where workers create a product based on their own creativity. This research uses a definition of meaningful work with three elements: personal significance, meaning make through work, and motivation to influence the greater good, from Steger et all (2012). Their research showed that if workers had less work centrality, individuals required more meaningful work in order to remain committed to their work.

The Working Environment of Game Development

Game development involves designing interactive systems and delivering those systems to the people who will play the game. While the individuals who contribute to those systems may have different skill sets, they all generally work under the same conditions if they are working on the same game. In situations where workers in game development are concerned about the quality of their working conditions, a negative image of working conditions may impact individual's subjective perspective of the meaning of their work. A survey by the International Game Development Association tried to find common themes across the perspective of game developers (Westar, 2019), and 73% of survey respondents believed that working conditions were a factor that influenced negative perception of the game industry. This speaks to an awareness of working conditions on the top of the minds of workers in the game development industry.

The study of game development workers often involves reflection of how game developers' creativity fits into the larger business of game development. In looking at game development from an innovative industrial standpoint, Thompson, Parker, and Cox (2016) place game development as part of a larger value chain, that while the illusion of freedom exists in the work that game developers perform, the fact that their funding comes from relationships with their publisher, their level of autonomy is minimal. Because the game industry is based on delivery of projects, their work results in "temporary, intermittent, and insecure jobs with weak work-life boundaries," (Thompson et al., 2016, p. 5).

The role of workers was also discussed from the perspective of sociology by Gill and Pratt (2008) who stress the precariousness of work in the creation of cultural products. They focus on the different aspects of cultural labor, such as precarity, affect, temporality, subjectivity, and solidarity. Because game development work is project-based, workers' efforts are heavily tied into the cultural products they create. This setup creates a space for meaningful work to occur on individual projects if the workers feel a connection between their individual efforts and the results of the project.

Definition of Meaningful Work for the Context of Automation and Game Testing

To create a simple system of meaningful work within the context of game design work, subjective concepts about meaningful work will be labeled internal factors, and objective concepts about meaningful work will be labeled external factors. Internal factors include how individuals feel about the meaningfulness, how they feel about the level of impact their work has on themselves and others, and whether their work aligns to their personal moral code. External factors include delivering a specific quantifiable value as part of a larger organization or working for the greater good that may not directly benefit the individual performing the work. When addressing meaningful work game development, game testing, and automation, connections will be made to internal and external factors in positive and negative relations to work.

Work Automation

Work automation involves taking a manual task that a human performs and turning that into an automated process performed by a powered machine. This outcome can result in the individual being replaced by the automation and no longer having a task to perform, or receiving a new role overseeing the automation technology. One is the development and improvement of society, that by automating work, more work can be done to benefit more people. On the other hand, automation of work displaces the individuals performing that work, leading to one of two outcomes. Automation can lead to those workers being placed in new roles, performing work that has not yet been automated or cannot yet be automated. Automation can also lead to displacement of workers, who will be required to reskill in a new profession in order to work again (Chessell, 2018). I will begin by addressing concerns of automation at a broad economic level, then address in greater detail automation's impacts on game development specifically.

Work Structure Reactions to Automation

From a human resources perspective, Sorells (2018) evaluates whether work automation will result in job losses and concludes that innovative tasks are required to stave off technological unemployment. By looking at the potential impacts of work automation and job loss by country, Segal (2018) takes into account the high quantity of jobs that have the potential to be automated, but comes to the conclusion that the increased pace of technological change might not cause an increased pace of job loss, but instead create increased inequality as automation targets work that is high value but low skill. Professors in technology and engineering Madakam, Holmukhe, and Jaiswal (2019) discuss how robotic process automation (RPA) reduces repetitive work in business operations and its applications. RPA represents the technical specifics of automation, without addressing the underlying changes to the broader labor force that might come from it. To fully understand the way that automation impacts the meaningfulness of work for workers, the technical knowledge required for automation needs to work with the broader economic and labor changes caused by that technology.

Automation and Game Development

The primary areas of game development, the team members creating the playable game as a product, are design, programming, and art. In attempting to describe the game development team and the relations individuals have with one another, professors of computer science McGuire and Jenkins (2012) describe three types of teams as mechanics, technology, and content, but the titles of individuals usually fall under the design, programmer, and artist. In this definition, sound design is placed as a subset under art, although development of art assets and sound assets may be very different processes. There are many other roles involved in creating games, such as testing, production, and other administrative roles, but this review will focus on the major three roles before discussing testing. Most of the other required work necessary to develop games will be discussed under management of the game development pipeline.

Design, Programming, and Art

Out of the three major roles, design is the least specific, which makes descriptions of automation in the field of design the most difficult. The computer science thesis work of Smith (2012) explores how artificial intelligence techniques can be used in the process of designing games to amplify designers' creativity, describing a theoretical role of "designer-programmer" that will use programming techniques to create designs that would be beyond what one individual could create on their own. In evaluating real-time strategy games as a game genre with automation potential, Lara-Cabrara, Nogueira-Collazo, Cotta, and Fenandedx-Leiva (2015) discuss procedural content generation (PCG) as a way of reducing the costs of development by automating map generation and AI development, but state that human creativity (in particular, tuning and evaluation of PCG results) is still a core part of the design process. The feminist approach of Phillips, Smith, Cook, and Short (2016) also discuss PCG and evaluate questions of the role of humans in relation to computational creativity. They critique the trade-off between efficiency of automation technology and the meaningfulness of hand-crafted results, and so do not view automation as a replacement, but as a means of co-authorship of designed content. Because designs need to be imbued with meaning, it is difficult for authors to make the case that automation will cause a strict replacement of designers in game development. Shaker (2016) created a textbook describing the techniques used in automation from a technical programming perspective, and acknowledges that while content generation technologies could potentially replace artists and designers as a way for game development costs to be reduced, he also stresses that PCG tools should be viewed as augmentation for human creators rather than replacement.

For game development, the trend for researchers emphasizes automation enhancing the work of programmers, artists, and designers, allowing them to create more content with less time and effort. This leads to more meaningful work considering external factors of delivering more completed game content out to players in the same amount of work time as if no automation existed. However, workers may also feel a loss of control over meaning of their work when faced with co-authorship. Worker-managed automated content generation would decrease meaningfulness of work when individuals are not trained effectively for injecting their own creativity into the automation process.

Pipeline Management

In addition to creating the design and the assets for games, the development process also includes the creation of a development pipeline that enables the creation of games. Because the technology that supports this pipeline is administrative rather than creative, it can be automated without suffering the loss of creativity felt with PCG. DevOps engineering consultants Humble and Farley (2011) describe this as part of the software delivery process with the term of Continuous Delivery, with the intention of turning the manual process of delivering software products to customers into an automated one. Because developers consider writing functional code as creative work and the software distribution process as non-creative work, automation increases meaningful work time while reducing work time spent on less meaningful tasks.

Unlike other software products, game development includes the delivery of art assets, which adds a different layer of complexity in the software development pipeline. Furtado, Santos, Ramalho, and de Almeida (2011) describe the process of pipeline management as the creation of Software Product Lines (SPL) that enable game development, and that candidates for automation as part of these product lines should be identified to improve productivity and abstraction. However, they accept the challenges that game design presents in the quantity of different domains required to create games, which makes it difficult to create reusable pipelines after automation. O'Donnell (2014) describes the advantages of having these pipeline tools for the developers, empowering creativity on the part of game developers, rather than replacing the meaningful work of game developers. Automating the game development process pipeline automation enables skilled developers who would otherwise need to perform routine manual tasks in the game development process to focus on the creative aspects of their work.

Game Testing

Game testing creates value in the game design process by determining if the design specified and marketed to potential players matches the implementation of the game by artists and programmers. Unlike artists, programmers, and designers, the output of their work does not directly create the game as delivered to players but rather acts as feedback to the other parts of the game design process. Programmers provide the inputs to game testers' work with game builds created in the pipeline management process.

The game testing process may also be referred to as the quality assurance process. The quality assurance title originates in manufacturing, and when describing the game development process, or software development process, with manufacturing analogies (such as the pipeline management process being the assembly line for software,) quality assurance provides an alternate description for game testing. Quality assurance also provides a broader context for the role of testers in the game design industry, as testers may be involved in the testing of hardware, server infrastructure, or game development tools, outside of the creation of games. This paper focuses on testers that test the interactive software to be delivered to the consumer, as meant to be interacted with by the consumer. When a source refers to quality assurance as either a title, a department, or a collection of responsibilities, this term will be either be kept as quality assurance to assume non-game-testing responsibilities or translated to talk about game testing as the term used.

The functional tasks of game testers can be abstracted to a broader value system, such as the two rules Schultz (2012) creates for game testing: “Don’t Panic” and “Trust No One”. The first rule applies to the attitude that testers bring to the activities they perform, while the second rule shows how testers should relate to other members of their game development team. Testing has a more difficult time justifying its existence than other areas of game development, such as programming, art, sound, and design, because the outputs of testing activities are not specifically integrated into the game, but provide feedback for the systems and content generation functions of a game development team to create a higher-quality game.

The concept of “Trust No One” as an effective rule for game testers is evaluated by Zhang (2009), who performed analysis on whether conflict between testers and developers results in a higher-quality product at a large company that also has software development functions. In his analysis, personal conflict and task conflict need to be differentiated to determine their separate impact, and while conflict in general reduced job satisfaction, task conflict increased software quality.

Activities, Relationships, and Outputs

The work of game testers involves performing tasks of testing in the context of a team to create testing artifacts. The resulting research that addresses this work usually involves a broader discussion of the game development process as a whole, but testing-focused technical resources can also provide relevant insights.

It can be difficult to discern when testing begins. The actual playtesting of a game only occurs when a build of the game is available to be played, even though preparatory work is performed before the build is ready. Meaningful testing on features occurs at the alpha stage according to Laramee's (2005) discussion of the game development process, where the features of a game are complete, and the assets are incorporated. From his work as a testing manager, Schultz (2012) provides a comprehensive look at the activities that testers perform in their work: playing games, identifying bugs, amplifying problems, notifying the team, testifying to others, and verifying fixes. Various techniques are used to determine how to perform well in a testing role, such as writing test cases, performing ad-hoc testing, and defect classification.

The level of expertise has an impact on the ability of testers to perform their responsibilities. Zhang (2009) believes testers in conflict with developers because developers maximize efficiency in the number of features delivered, and testers maximize effectiveness of the software product for end users. While testers bring broad knowledge to the table, and developers bring specialized knowledge, experienced developers working with inexperienced testers results in an imbalance in those conflicts. This imbalance is heavily reflected in the game industry, where McGuire (2009) describes how testing jobs are less desirable than other more creative work on game development teams. In a more detailed look at the roles of individuals on the game design team, Pedersen (2009) differentiates among different levels of testers, with lead testers and testing managers interacting more frequently with other game development roles, while more junior testers receive work from the tests written by their leads and managers.

The relationships of game testers with other team members are critical for their ability to provide feedback on the working state of the game in development, but game testers are sometimes not included in diagrams describing the relationships of team members. McGuire (2009) creates a diagram consisting of mechanics, content, and technology, explicitly excluding game testers from the relationships among the core game development team members. On the other hand, through the Scrum development process Keith (2010) gives testers an approval authority that they must provide approval on work items before they are considered complete. By granting game testers the authority to nullify progress of other team members they would have the potential to have their feedback and critique have more weight in the development process. Rabin (2012) talks about the relationship that testers have with producers on a game design team. This is because the producer is in charge of classifying bugs as they are handed over to the development team, and the testers may identify bugs they believe are important, but will be rejected by the producer who tries to keep the development team focused on what they believe are high-value work items.

For Schultz (2012), the major outputs of testers are records of completed tests and the results of those tests, whether they passed or failed, as well as written bugs entered into the bug-tracking system that are used by other members of the team. Honest representation of completed tests provides useful information to other members of the team, and well-written bug reports provide actionable data that those members use to fix the code involved.

Connecting Meaning to Testing

The difficulty of connecting meaning to testing comes from existing in a middle space between game development and game playing. The indirect influence that testers have on the final output of the game, as mediated through the producer and the developers, designers, and artists who make changes based on tester feedback, means that the meaningfulness provided from external factors, creating a game, may not be present for testers. The lack of influence in the final output is stated by one anonymous tester, “Things you like get taken away without warning, things you hate become features” (Trenches, 03/03/2015). The meaningfulness of game play applies as an internal factor for meaningfulness, but this is also not present, as described by another anonymous tester, “This was not ‘getting paid to play games’ – this was ‘getting paid to perform monotonous, time consuming, mind numbing activities” (Trenches, 08/09/2011). The lack of creative input into the testing process positions it closer to the administrative work of creating the game development pipeline, rather than the creative design work of other members of the game development team.

Rabin (2010) positions meaningfulness in testing through two lenses. Firstly, he states testing provides a challenge because each game is unique, arguing that testing is not just repetitive work. Secondly, he presents testers as the first step in a game developer’s career, stating that “With solid QA lead experience, the step up to the studio is a natural progression. A large number of producers, designers, and executives form their start in QA” (Rabin, 2010, p. 831) These two reasons provide an effective internal factor approach to meaningfulness but lack the external factors for meaningfulness. A focus on testing as a steppingstone is also reflected in Moore (2010), who attempts to provide hope for someone looking to move beyond testing, “A tester who is bright, communicative, and observant might draw the attention of management and be offered a position in production or design” (Moore, 2010, p. 95). Even though Keith (2010) tries to broaden the role of testers in the game development process, he also explicitly refers to the role as a “gateway role”, in the context of testers contributing to non-testing activities as part of their work to prepare them more effectively for other types of work in the company. To present testing as a pathway role for other jobs in the game development space, testing's external factors, providing value that is meaningful and appreciated by other members of the team, is downplayed. Instead, a focus is made on the internal factors approach, as testers place themselves in a better position through their work to acquire their actual desired role.

Automation of Testing

In his study as an AI researcher, Thompson (2020) describes two different types of automation applied to testing, bots that are written to stress-test server infrastructure, and bots that are written to test game playability. The first type of automation performs specifically written tasks using abilities that players would never have. The second type of bot (client,) were developed to test procedurally generated content at a scale that human testers could not effectively test. As automation has been brought to design, testing has had to follow up with its own automation technology. However, this type of bot does not provide a replacement for all human testing, as use of this testing automation system required creating a level system with waypoints, and game testers would review data gathered by the bots. This type of automation represents a transitioned role, where the testers transitioned from manually testing levels as a primary function to observing and notifying developers when issues arose with the automated testing infrastructure.

The focus of Schultz (2012) on automation through the recording and altering test inputs, programming test results, and determining the success or failure of tests as they are run against the record of inputs provides a more code-separated approach to testing automation. This approach does not require the involvement of developers or require a developer-tester role to implement the automated testing functionality. Pederson (2009) also describes this type of testing but states this type of testing is the responsibility of senior testers, rather than all levels of testers.

Keith (2010) describes testing within the context of Agile development with Scrum and talks about the importance of having testers embedded within development teams from an early point in the development process. This allows for the simultaneous development of automated tests using white-box testing methods, to prevent an overload of testing required in the post-production process. Fields (2010) agrees that testers should be embedded early in the development process with a distributed team but leaves out testing automation.

Connecting Meaning to Testing in the Automated World

Based on the internal-external factors model, automated testing lines up with the way that procedural content generation is perceived in the broader game development context. Namely, automation should enable greater impact by humans using the automation but does not replace them. However, testers occupy a different space than the rest of the development team because of the power differential, and the lack of long-term expertise in roles, as mentioned by Zhang (2009), so testers may be at a disadvantage when it comes to automation. Keith (2011) also brings up the problem that testers are not well prepared for their roles when they are brought on the team late in the production process, “It is impossible to hire and properly train a small army quickly, so testers are minimally trained for each game” (Keith, 2011, p. 250). Temporary roles will be less likely to adapt to changes in roles than other positions, and the creation of automation in the testing environment creates more of a likelihood for testing to shift to be the responsibility of programmers or a programmer-tester role. The next level of research on this topic should address the likelihood that a programmer-tester role will develop and how many testers could potentially shift into that role, in the same way that Smith (2012) describes a programmer-designer role managing procedural content generation from a design perspective.

Conclusion

Game testers occupy a different role than other members of the game development team, so their circumstances and responses to automation will be different from other game development team members. A better understanding of game tester as a role with unique characteristics, looking beyond the gateway role that testing currently represents, is important to understand reactions of testing activities and testers to automation. In order to gain that understanding, the work of game testers must be seen as meaningful by researchers in the academic community. With additional focus on the experiences of game testers and perspectives on testing that place testing inside of a distinct career path, meaningfulness in testing can be addressed in future research.

Bibliography

Allan, B. A. (2017). Task significance and meaningful work: A longitudinal study. *Journal of Vocational Behavior*, *102*, 174–182. https://doi.org/10.1016/j.jvb.2017.07.011

Anonymous (2015, March 3). QA testing: The psychological problem. [Blog Post]. http://trenchescomic.com/tales/post/qa-testing-the-psychological-problem

Anonymous (2011, August 9). Man of the Match. [Blog Post]. http://trenchescomic.com/tales/post/man-of-the-match

Chessell, D. (2018). THE JOBLESS ECONOMY IN A POST-WORK SOCIETY: HOW AUTOMATION WILL TRANSFORM THE LABOR MARKET. *Psychosociological Issues in Human Resource Management*, *6*(2), 74–79. https://doi.org/10.22381/PIHRM6220187

Dieterich, R. O. (2017). *Using Proof-Of-Concept Feedback to Explore the Relationship Between Artists and Procedural Content Generation in Computer Game Development Tools*.

Deepak, R. D. S., & Swarnalatha, P. (2019). Continuous Integration—Continuous Security—Continuous Deployment Pipeline Automation for Application Software (CI - CS - CD). *International Journal of Computer Science and Software Engineering; Dubai*, *8*(10), 247–253.

Furtado, A., Santos, A., Ramalho, G., & de Almeida, E. (2011). Improving Digital Game Development with Software Product Lines. *IEEE Software*, *28*(5), 30–37. https://doi.org/10.1109/MS.2011.101

Gill, R., & Pratt, A. (2008). In the Social Factory?: Immaterial Labour, Precariousness and Cultural Work. *Theory, Culture & Society*, *25*(7–8), 1–30. https://doi.org/10.1177/0263276408097794

Humble, J., & Farley, D. (27). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Video Enhanced Edition* (1st ed.). Addison-Wesley Professional.

Jiang, L., & Johnson, M. (2018). Meaningful Work and Affective Commitment: A Moderated Mediation Model of Positive Work Reflection and Work Centrality. *Journal of Business and Psychology*, *33*(4), 545–558. https://doi.org/10.1007/s10869-017-9509-6

Keith, C. (2010). *Agile game development with Scrum*. Addison Wesley.

Lara-Cabrera, R., Nogueira-Collazo, M., Cotta, C., & Fernández-Leiva, A., J. (2015). Procedural Content Generation for Real-Time Strategy Games. *International Journal of Interactive Multimedia and Artificial Intelligence*, *3*(2), 40–48. https://doi.org/10.9781/ijimai.2015.325

Laramée, F. D. (2005). *Secrets of the game business* (2nd ed.). Charles River Media.

Lips-Wiersma, M., Morris, L., & Morris, L. (2017). *The Map of Meaningful Work (2e): A Practical Guide to Sustaining our Humanity*. Routledge. https://doi.org/10.4324/9781351252065

Madakam, S., Holmukhe, R., & Jaiswal, D. (2019). THE FUTURE DIGITAL WORK FORCE: ROBOTIC PROCESS AUTOMATION (RPA). *Journal of Information Systems and Technology Management : JISTEM*, *16*, 1–17. https://doi.org/10.4301/S1807-1775201916001

Mazilescu, V. (2011). Automation of the Work intensively based on Knowledge, a Challenge for the New Technologies. *Annals of Dunarea de Jos University. Fascicle I : Economics and Applied Informatics*, *1*(2), 37–48.

Martela, F., & Pessi, A. B. (2018). Significant Work Is About Self-Realization and Broader Purpose: Defining the Key Dimensions of Meaningful Work.(Report)(Brief article). *Frontiers in Psychology*, *9*, 363. https://doi.org/10.3389/fpsyg.2018.00363

McGuire, M. (2009). *Creating games: Mechanics, content, and technology*. AK Peters, Ltd.

O'Donnell, C. (2014). *Developer's Dilemma: The Secret World of Videogame Creators*. The MIT Press.

Pedersen, R. E. (2009). *Game design foundations* (2nd ed.). Wordware Pub.

Phillips, A., Smith, G., Cook, M., & Short, T. (2016). Feminism and procedural content generation: Toward a collaborative politics of computational creativity. *Digital Creativity: Post-Anthropocentric Creativity*, *27*(1), 82–97. https://doi.org/10.1080/14626268.2016.1147469

Rabin, S. (2010). *Introduction to game development* (2nd ed.). Course Technology.

Schultz, C. P., & Bryant, R. D. (2012). *Game testing all in one* (2nd ed.). Mercury Learning and Information.

Segal, M. (2018). How automation is changing work. *Nature*, *563*(7733), S132–S135. https://doi.org/10.1038/d41586-018-07501-y

Shaker, N. (2016). *Procedural Content Generation in Games*. Springer International Publishing. https://doi.org/10.1007/978-3-319-42716-4

Smith, A. M. (01). *Mechanizing Exploratory Game Design* [eScholarship, University of California]. https://escholarship.org/uc/item/4600g227

Sorells, B. (2018). WILL ROBOTIZATION REALLY CAUSE TECHNOLOGICAL UNEMPLOYMENT? THE RATE AND EXTENT OF POTENTIAL JOB DISPLACEMENT CAUSED BY WORKPLACE AUTOMATION. *Psychosociological*

Issues in Human Resource Management, *6*(2), 68–73. https://doi.org/10.22381/PIHRM6220186

Thompson, P., Parker, R., & Cox, S. (2016). Interrogating Creative Theory and Creative Work: Inside the Games Studio. *Sociology*, *50*(2), 316–332. https://doi.org/10.1177/0038038514565836

Thompson, T. (2020, March 4). The Secret AI Testers inside Tom Clancy's The Division [Blog Post]. Retrieved from https://gamasutra.com/blogs/TommyThompson/20200304/359028/The_Secret_AI_Testers_inside_Tom_Clancys_The_Division.php?

Veltman, A. (2016). *Meaningful Work*. Oxford University Press USA - OSO. Retrieved from http://ebookcentral.proquest.com/lib/gmu/detail.action?docID=4707129

Weststar, J. & Kwan, Eva & Kumar, Shruti. (2019) *Developer Satisfaction Survey 2019 – Summary Report* [Summary Report] Retrieved March 23, 2019 from Independent Game Developer's Association: https://s3-us-east-2.amazonaws.com/igda-website/wp-content/uploads/2020/01/29093706/IGDA-DSS-2019_Summary-Report_Nov-20-2019.pdf

Zhang, X. (2009). *Understanding conflict between developers and testers in software development: Sources and* impact. Retrieved from ProQuest Dissertations and Theses database. (UMI No. 3400169)