Exploring Quality Assurance Practices and Tools for Indie Games

Jeff Cho
University of Alberta, Canada
jeff.cho@ualberta.ca

Karim Ali
University of Alberta, Canada
karim.ali@ualberta.ca

2023 IEEE/ACM 7th International Workshop on Games and Software Engineering (GAS) | 979-8-3503-1203-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/GAS59301.2023.00010

***Abstract*—The games industry is growing worldwide, even eclipsing the global film industry as a premier entertainment solution. Developing a commercial game is a complex, lengthy, and costly process. Therefore, quality assurance (QA) is critical for producing high-quality games that are fun and reasonably defect-free. Prior studies have explored game development methodologies and testing approaches, goals, and automation. However, they have not addressed the disparate contexts of independent (indie) and non-indie game development with respect to available funding and resources. Since indie games make up the lion's share of newly released games each year, we want to empower their developers by maximizing their QA opportunities within their resource constraints. To lay the foundation for such support, we surveyed 19 game developers who have experience with commercially released games to learn about their QA experiences and perspectives based on 22 of their released game projects. Our survey results show that indies have less clear goals and plans for testing, perform tests on a conditional basis over a regular testing schedule, and have subjective test results.**

***Index Terms*—games, testing, indie**

I. Introduction

The games industry is constantly growing worldwide, being worth over $180B USD in 2021 [1], eclipsing the global film industry and North American sports industries combined in 2020 [2]. This rise was fuelled in part by the COVID-19 pandemic environment which saw a need for accessible and engaging entertainment at home and the ability to connect with others virtually. Games also serve important societal and educational functions. In the United States, 74% of parents play video games with their children at least once a week, and 80% of Americans believe that games are educational [3]. Despite some arguments to the contrary [4], games are also largely considered to be an art form, with selected video games having been curated and displayed at the Smithsonian American Art Museum [5] and the Museum of Modern Art [6]. In short, video games are no longer the niche that they once were; they are a global industry with wide reach and social impact.

Developing commercial games is a costly process that continues to increase in complexity with longer development duration [7]. For example, the rise in cost for a blockbuster PlayStation console game has risen from $100M per title for PlayStation 4 to $200M per title for PlayStation 5 [8]. Given these rising costs, game studios are under pressure to maximize their return on investment and minimize unnecessary or unforeseen costs. Quality Assurance (QA) and testing is one aspect of development that game studios can leverage to increase efficiency and decrease potential downstream costs. Bugs and defects in software cost more to fix the later they are discovered in the development cycle; fixing a bug found in testing can be 15× more costly than if it had been found while it was in design [9], [10]. It is then in the developers' best interests for bugs to be detected and corrected as early as possible. Blockbuster—so-called "AAA"—games and studios typically have millions in funding, which enables them to have dedicated QA teams for testing with software and hardware infrastructure provided for them. On the other hand, smaller, independent (i.e., "indie") studios have major resource constraints that limit their QA abilities. Although they have fewer resources, indie games are an important pillar of the gaming industry. For example, on Steam [11], indie games account for more than 95% of all titles listed, just under 40% of units sold, and 28% of revenues [12].

In this paper, we study the current practices, goals, and needs of commercial game developers to identify how we can improve quality assurance practices and tools for indie game developers. In particular, we are trying to answer the following research questions:

RQ1: What are the most significant differences in game testing between indie and non-indie developers?

RQ2: How much automated or scripted testing do indie developers use compared to non-indie developers, and what are the biggest pain points for automation?

To answer these questions, we conducted an anonymous online survey of 19 indie and non-indie game developers who detailed their previous experiences with QA on 22 commercially-released game projects. Our survey focuses on 7 primary areas of interest: test performance, test planning, testing goals, test automation, testing tools, test results, and testing resources. We then conducted both quantitative and qualitative analyses on the responses to gain further insights into our research questions. Our analysis shows that indie developers have less clear goals and plans for testing, and perform tests on a conditional basis over a regular testing schedule. Open-ended answers to our survey questions have also revealed several cultural and management issues that respondents have raised. Artifacts for this study consist of the survey questions, anonymized answers, and analysis results, which are available online [13].

979-8-3503-1203-4/23/$31.00 ©2023 IEEE
DOI 10.1109/GAS59301.2023.00010

II. Survey on Game Testing

To understand the current state of testing for indie games, we have conducted an online survey of 19 indie and non-indie commercial game developers who have shared their experience working on 22 of their released games.

A. Survey Construction

Our survey consists of two sections: background information and project-based responses.

1) Background Information: Our survey participants must have had 2+ years experience in commercial game development in the past 10 years, and worked for 1+ year on a commercially released game. We chose these criteria to restrict responses to the domain of commercial games, because the dynamics of hobbyist or amateur games have different considerations. In particular, working on a commercially released game and understanding release management and post-release support leads to specific pressures not experienced on non-released commercial games. Therefore, this section asked the following questions:

(Q1) *Do you consent to participate in this survey?* Yes/No.
(Q2) *Do you have at least two years' experience in commercial game development in the past 10 years?* Yes/No.
(Q3) *Have you worked for at least one year on a game that has been commercially released?* Yes/No.
(Q4) *How many years have you been active in the games industry?* Decimal text input between 0–100.
(Q5) *Which of the following roles have you previously held in the games industry?* Select: Developer, Programmer, and/or Engineer; Quality Assurance and/or Game Tester; Team Lead, Producer, and/or Project Manager; Game Designer, Level Designer, Gameplay Designer, and/or Product Owner; Other (specify).
(Q6) *How many years of indie game development experience do you have, in the past 10 years?* Integer input between 0–10.
(Q7) *How many years of AA game development experience do you have, in the past 10 years?* Integer input between 0–10.
(Q8) *How many years of AAA game development experience do you have, in the past 10 years?* Integer input between 0–10.
(Q9) *What is your age range?* Under 18; 19–24; 25–34; 35–44; 45–54; 55–65; 65+.
(Q10) *What country do you currently reside in?* Text response.

2) Project-Specific Questions: Our main survey questions aim at gathering information about 7 main aspects of quality assurance: test performance, test planning, testing goals, test automation, testing tools, test results, and testing resources. To ensure non-generic responses, we asked respondents to evaluate a real project that they have worked on while answering these questions. Respondents provided responses for at least one project, but they could optionally provide responses for up to 3 projects. For each project, we first ask the following questions:

(Q11) *Please enter a nickname for the project you have chosen. This is used purely for reference purposes and does not have to be the actual name of the game.* Text response.
(Q12) *How would you classify the project?* Indie, AA, AAA.
(Q13) *What was your primary role on the project?* Same selection options as Q5.
(Q14) *What were your other roles on the project, if any?* Same selection options as Q5.
(Q15) *What engine did you use for the project? If it was a proprietary engine that you do not want to identify due to privacy reasons, please enter"Proprietary".* Text response.
(Q16) *How long were you working on this project?* 1 year or less; 2–3 years; Over 3 years.
(Q17) *What was the size of the team for this project?* 1–49; 50–99; 100–499; 500 or more.
(Q18) *What was the approximate budget for this project?* Unknown; Under $1M USD; $1M–$10M USD; Over $10M USD.
(Q19) *Did you have a publisher for this project?* Yes, we had a publisher who provided marketing support; Yes, we had a publisher, but they did not provide marketing support; No, we did not have a publisher.
(Q20) *How long was this project in development?* 1 year or less; 2–3 years; Over 3 years.

For each quality assurance aspect, we asked one free-text question, telling the respondent to "Please be as specific and detailed as possible without compromising privacy and anonymity; you can always obscure details as required if you feel they would be too identifying." We followed this question with a few prompts (marked with letters below) to be answered on a Likert-like scale: Strongly Disagree, Somewhat Disagree, Neither Agree Nor Disagree, Somewhat Agree, Strongly Agree. Participants could also select "N/A" for inapplicable prompts.

(Q21) *Please describe how testing was performed on [this project]. What steps were taken by testing, by whom, and when?*
(Q22-a) *We primarily used testing tools that we developed in-house.*
(Q22-b) *We primarily used third-party testing tools.*
(Q22-c) *Team members were able to give actionable input and feedback to the testing process.*
(Q22-d) *The majority of testing was done by a dedicated quality assurance team.*
(Q22-e) *We routinely ran unit tests on this project.*
(Q22-f) *We routinely ran regression tests on this project.*
(Q22-g) *We routinely ran integration tests on this project.*
(Q22-h) *We routinely ran smoke tests on this project.*
(Q23) *Please describe the test planning process for [this project]. How were test plans made, by whom, and when?*
(Q24-a) *The testing process was well documented and openly available to the team.*
(Q24-b) *The testing process followed the testing plan in most cases without requiring ad-hoc changes.*
(Q24-c) *At any given point during development, I had a good idea*

of the current testing plan including tools, priorities, and schedule.

(Q24-d) *Each major feature for development had a testing or validation plan prior to implementation.*

(Q24-e) *There were no concrete testing plans or priorities for the majority of this project.*

(Q24-f) *Test plans and priorities were primarily set by developers.*

(Q24-g) *Test plans and priorities were primarily set by quality assurance personnel.*

(Q24-h) *Test plans and priorities were primary set by supervisors or leads.*

(Q24-i) *Test plans or priorities were primarily set by the producer, game director, or senior management.*

(Q25) *Please describe the goals of testing that you and your team had on [this project]. What were you testing for, and how did your testing process accomplish that goal (or fall short)?*

(Q26-a) *We routinely performed testing for game mechanics (e.g. game rules, balance between features, overall content design).*

(Q26-b) *We routinely performed testing for technical aspects of the game (e.g. functionality, stability).*

(Q26-c) *We routinely performed testing for user experience (e.g. fun factor, user impression, satisfaction).*

(Q26-d) *The results of testing were often subjective and open to interpretation.*

(Q27) *Please describe the degree of manual and automated testing that you experienced on [this project], including whether there were specific areas that were prioritized for testing manually or with automation.*

(Q28-a) *We used more manual testing than automated testing.*

(Q28-b) *We had personnel dedicated to writing scripted or automated tests on this project.*

(Q28-c) *The automated testing tools that we used required frequent adjustments as the project progressed.*

(Q28-d) *Manual testing yielded more actionable results than automated testing.*

(Q28-e) *Automated testing yielded more actionable results than manual testing.*

(Q28-f) *Manual testing yielded results that could not be determined through automated testing.*

(Q28-g) *Automated testing yielded results that could not be determined through manual testing.*

(Q28-h) *Automated testing is important for game development.*

(Q28-i) *I am happy with the proportion of automated testing to manual testing that was done for this project.*

(Q28-j) *Automated testing was a source of frustration on this project.*

(Q29) *Please list the tools you used for testing on [this project], whether they were used in a certain order, as well as what you perceived as the major benefits and drawbacks to each one (with consideration for functionality and usability).*

(Q30-a) *The tools that we used for testing were developed specifically for games.*

(Q30-b) *The tools that we used for testing were intuitive and easy to use.*

(Q30-c) *The tools that we used for testing were well-suited to our testing goals.*

(Q30-d) *The output from testing was easy to read and understand.*

(Q30-e) *The tools that we used for testing complemented each other well.*

(Q30-f) *There was a structured approach to testing that was repeated throughout the project.*

(Q31) *Please describe the format, readability, and usability of the testing output, and how the testing results and output were used by the [project] team.*

(Q32-a) *The results of testing were used to form actionable plans to address the results.*

(Q32-b) *The results of testing could cause major changes to the game's design.*

(Q32-c) *The results of testing were used to validate internal targets.*

(Q32-d) *The results of testing were primarily used by the testing or development team.*

(Q32-e) *The results of testing were used primarily by leads and managers.*

(Q32-f) *The results of testing were monitored or visualized over time and used as metrics.*

(Q33) *Please describe your understanding of the allocation of resources for testing on [this project]. What resources were allocated to testing (human resources, funding, time allotment, hardware, etc.) and how?*

(Q34-a) *We had sufficient testing resources overall to support this project.*

(Q34-b) *We had sufficient testing/QA personnel dedicated to this project.*

(Q34-c) *We had sufficient test automation for this project.*

(Q34-d) *We had appropriate testing tools given the project and its testing goals.*

(Q34-e) *The team requested additional testing resources during the project.*

(Q34-f) *Testing was primarily contracted to an outside firm.*

(Q34-g) *Testing was primarily performed internally on the team.*

(Q34-h) *Testing was done primarily by team members who had other primary tasks, such as developer or designer.*

(Q34-i) *Testers were integrated with the development team and had regular direct access to developers.*

We then asked if the participant would like to provide additional information, in free-text, about the project, as well as describe their experience with more projects.

(Q35) *Would you like to add any additional details about [this project]?*

(Q36) *Would you like to discuss another game project that you worked on for at least one year? If you select "yes", you will be asked the same set of questions about another project.* Yes/No

(Q37-Q62) Optional Second Project-Based Response Set

(Q63-Q87) Optional Third Project-Based Response Set

Finally, we asked about past experience with testing:

(Q88) *What has made testing easier or faster for you in the past?*
(Q89) *What are the biggest pain points you have encountered with testing?*
(Q90) *What changes would you make or like to see in testing for games?*

B. Participant Recruitment

Prior to deploying our survey, we invited 3 game developers from diverse backgrounds to pilot the survey and give us feedback on each question as well as the overall flow and feel of the survey. The diversity of backgrounds includes gender minority, experiences, positions, and game engines. Each participant completed one pass of the survey for one project each. After completing the survey, we conducted debriefing interviews with each participant, which lasted 20 minutes on average. Participants were asked to keep notes as they progressed through the survey with any questions, concerns, or feedback that they had. Section II-A presents the final deployed survey, with underlined text indicating changes based on pilot feedback. We have made all details available online [13].

We set out to reach as many game developers with commercial experience as possible to take the survey. Since we had no guaranteed way of verifying an anonymous respondent's credentials, we used non-probability sampling [14]. We advertised the survey on social media platforms such as Twitter, Facebook, LinkedIn, Slack, and Discord, including public game development-focused groups and communities. To reach as many game developers as possible regardless of geographic location or time zone, we made these social media posts periodically and at varying times of day. We additionally asked colleagues and industry contacts to invite any eligible game developers they knew for snowball sampling [15]. We ensured that the participants of the survey did not overlap with the participants of the pilot survey.

C. Data Verification and Grouping

To use only valid data for our analysis, we first checked the integrity of the responses that we received. Out of 82 people who began the survey, 21 respondents completed it. Upon examining all completed for coherence, we removed two of them. The first response appeared to be a troll who wrote *"You lost me at 'he/him', pronouns in bio is a red flag."* for every free-text field in response to one of the authors including their pronouns on their social media account. The second response did not provide sensible information for free-text responses (e.g., *"me"* for describing test planning processes). There were other respondents who were unable to answer certain questions for sections of the survey, such as those who did not have experience with automated testing or those who had no visibility into test planning processes. This was evidenced by a high number of "N/A" or "Unknown" responses in these sections. However, we retained these responses because they were potentially indicative of trends in the industry and their free-text responses provided additional insight into their experiences and opinions.

In the end, we had 19 eligible respondents who submitted 22 project-specific responses, with one respondent submitting 2 projects and one respondent submitting 3 projects. We assigned an ID to each game (G1–G22), which we will use to refer to participants responses for better context of information. Table I and Table II present the 22 games. Given the low number of AA projects submitted as well as our focus on differences between indie and non-indie developers, we divided our responses into two groups: indie and non-indie. We put games reported as indie into the indie group, and games reported as AAA into the non-indie group. We reallocated the only 3 AA games (G14, G18, and G21) to these two groups. These AA games have budgets $1M–$10M USD and no publisher support. Since G14 and G18 have 100–499 members on their teams, we classified them as non-indie games. This is because the processes and tools for a team of this size are more in line with AAA practices. Since G21 has 1–49 team members, we classified it as an indie game. Grouping game types left us with 11 indie and 11 non-indie games for consideration. We add a suffix to each project identifier to indicate whether it is an indie (I) or a non-indie (N) game (e.g., G18-N and G21-I).

D. Data Analysis Methodology

To analyze our data, we used quantitative analysis for Likert-like responses and qualitative analysis for free-text responses.

1) Quantitative Analysis: We transformed Likert-like data into numerical values (1 = *Strongly Disagree*; 2 = *Disagree*; 3 = *Neither Agree nor Disagree*; 4 = *Agree*; 5 = *Strongly Agree*) for ease of analysis. We also encoded answers that are not in the Likert-like range (0 = *N/A*; -1 = *Unkown*) for counting purposes. We then compared the medians between indie and non-indie responses for each prompt to discover the magnitude and trend of differences between the groups for specific items. Since the distributions are generally non-normal and the sample sizes are small ($n_i = 11$ in each group), we perform Mann-Whitney U-tests [16], [17] on each set of responses to look for statistically significant differences in distributions that would illustrate a divergence between indie and non-indie testing. Since *N/A* and *Unknown* responses are not relevant for the Mann-Whitney U-test, we omitted these values from the tests.

We ran our tests as two-tailed tests with $\alpha = 0.05$ to achieve 95% confidence in our results. In accordance with Bergmann et al. [18], we use asymptotic approximation since our sample size is greater than 10 and ties often occur in our data due to our 5-item response scale. Since we use indies as the first group for these tests, U indicates how many indie observations are greater than non-indie observations. The maximum value of U is the product of sample sizes of the two groups, $n_1 n_2$, corresponding to the number of pairwise comparisons. The closer U is to $n_1 n_2$, the more indies agree with the prompt in comparison to non-indies. We also report f, the Common

TABLE I: Project information for indie games.

ID	Type	Engine	Team Size	Budget	Publisher Support	Duration
G1-I	Indie	Proprietary	1–49	<$1M	Yes, without marketing	3 years
G4-I	Indie	Proprietary	1–49	<$1M	Yes, with marketing	3 years
G5-I	Indie	Unity	1–49	<$1M	No	1 year
G8-I	Indie	Unity	1–49	<$1M	No	3 years
G9-I	Indie	Unity	1–49	<$1M	No	3 years
G11-I	Indie	Unity	1–49	<$1M	No	1 year
G16-I	Indie	Unity	1–49	<$1M	No	3 years
G17-I	Indie	Unity	1–49	<$1M	No	2–3 years
G19-I	Indie	Proprietary	1–49	<$1M	Yes, with marketing	2–3 years
G20-I	Indie	Unity	1–49	<$1M	No	3 years
G21-I	Indie	Proprietary	1–49	$1–$10M	No	1 year

TABLE II: Project information for non-indie games.

ID	Type	Engine	Team Size	Budget	Publisher Support	Duration
G2-N	Non-Indie	Frostbite	100–499	>$10M	Yes, with marketing	3 years
G3-N	Non-Indie	Unreal	50–99	>$10M	Yes, with marketing	2–3 years
G6-N	Non-Indie	Unreal	100–499	>$10M	Yes, with marketing	3 years
G7-N	Non-Indie	Proprietary	100–499	>$10M	Yes, with marketing	3 years
G10-N	Non-Indie	Proprietary	500	>$10M	Yes, with marketing	3 years
G12-N	Non-Indie	Frostbite	500	>$10M	Yes, with marketing	3 years
G13-N	Non-Indie	Unreal	100–499	>$10M	Yes, with marketing	3 years
G14-N	Non-Indie	Unreal	100–499	$1M–$10M	No	2–3 years
G15-N	Non-Indie	Unity	100–499	>$10M	No	3 years
G18-N	Non-Indie	Unreal	100–499	Unknown	Yes, without marketing	3 years
G22-N	Non-Indie	Unreal	100–499	>$10M	Yes, without marketing	3 years

Language Effect Size (CLES) [17], expressed as a proportion of responses from the first group (i.e., indies) that are larger than the other. For example, a CLES of 75% means that in three-quarters of the pairwise comparisons, the indie value is greater than the non-indie value.

2) Qualitative Analysis: For our qualitative analysis, we used Reflexive Thematic Analysis (RTA) [19] to systematically code free-text responses, which we gather into themes to analyze trends of testing approaches, tasks, and where they are fundamentally dissimilar. Following recent work by Byrne et al. [20], our RTA process has 6 phases: familiarization, generating initial codes, generating themes, reviewing themes, defining and naming themes, and producing the report. To make sure our procedure and analyses are trustworthy, we compared our process with Braun and Clarke's checklist for proper thematic analysis [21].

III. Survey Population in Context

Our survey respondents have a median of 6 years of experience in game development. In the past 10 years, 15 respondents had indie development experience, and 17 had non-indie development experience. Throughout their careers, participants have held several roles. Lead, Designer, and QA roles each have 11 participants (57.9%) who held the role in the past, with 7 respondents (36.8%) previously holding a Dev role. Four respondents (21.1%) indicated other roles such as Business/Marketing. Overall, the participants have a breadth of experience in different roles and project scopes, and over half of them have experience in a QA role.

Table I and Table II present key project information for indie and non-indie games, respectively. Seven indie games use Unity (63.6%) and 4 use proprietary engines (36.4%). All 11 indie teams have 1–49 members, and 10 games (90.9%) have budgets less than $1M USD. The majority of indie games (72.7%) have no publishers. Only 2 indie games (18.2%) have a publisher who provided marketing support, while 1 (9.1%) has a publisher without marketing support.

Six non-indie games (54.5%) use Unreal, and 4 games use proprietary engines, including 2 (36.4%) that use Frostbite and 1 (9.1%) that uses Unity. Most non-indie teams (72.7%) have 100–499 members, while 2 teams (18.2%) have 500 or more and 1 (9.1%) has 50–99 people. Unlike indie games, the majority of non-indie games (81.8%) have over $10M USD in funding. Additionally, 7 non-indie games (63.6%) have a publisher who provided marketing support, while the rest either have a publisher without marketing support or no publisher attached.

TABLE III: Prompts with significant differences ($\alpha = 0.05$) in responses between indies and non-indies. A higher U-statistic indicates stronger agreement from indies than non-indies. Sample size is 11 for all prompts except ones marked with * to indicate a sample size of 10 or † for a sample size of 9.

ID	Aspect	Prompt	Indie Median	Non-Indie Median	U	P-Value	Effect Size
(Q24-d)	Planning	Each major feature had a test plan prior to implementation	1	3†	18.5	0.0153	18.7%
(Q24-e)	Planning	There were no concrete testing plans or priorities for the majority of this project	3	1	100.0	0.0068	82.6%
(Q24-g)	Planning	Test plans and priorities were set by QA personnel	2	4	30.5	0.0477	25.0%
(Q26-d)	Goals	Testing results were subjective and open to interpretation	4	2	100.5	0.0080	83.1%
(Q32-c)	Results	Testing results were used to validate internal targets	4	4*	25.0	0.0207	22.7%
(Q32-f)	Results	Testing results were monitored or visualized over time as metrics	1†	5*	17.0	0.0184	18.9%
(Q34-b)	Resources	We had sufficient testing personnel on this project	2	4	18.0	0.0040	14.9%
(Q34-c)	Resources	We had sufficient automated testing on this project	2†	3*	19.0	0.0286	21.0%
(Q34-d)	Resources	We had appropriate testing tools for this project and goals	2	4*	9.5	0.0010	8.6%
(Q34-g)	Resources	Testing was primarily performed internally on the team	5	4	93.5	0.0061	77.3%
(Q34-h)	Resources	Testing was done primarily by team members who had other primary roles	5	1	116.0	0.0002	95.9%

IV. Survey Results

Through our reflexive thematic analysis, we have identified 10 primary themes: testing approach, goals, plans, types, tools, resources, timing, automation, results, and overall sentiment. Most of these themes reflect the categories of our survey. In this section, we discuss our survey results with respect to these categories. We present the quantitative data, and use the open-ended data for interpretation.

A. Testing Approach and Planning

Table III shows that 82.6% of indies agree that there are no concrete testing plans or priorities for their project (Q24-e). Overall, indies describe a chaotic, unstructured testing approach that is more about experiential playtesting than thorough systematic verification. For example, G4-I states that *"QA/Testing was mostly a fluent process and not necessarily structured. As a small team it didn't make sense to invest the resources to do so."* We also found that testing plans for indie developers are vague and lack details compared to non-indies. For example, G20-I states that their team *"made no test plans"* and that planning *"was not done through a formal process."*

Unlike indies, non-indies usually describe a regimented approach to testing for uncovering actionable items. For example, G15-N states that *"Test plans made and maintained by internal*

QA in test rails." and that *"[the] Engineering group separately implemented unit and regression tests to their understanding of the feature."*

B. Testing Goals and Results

Table III shows that 83.1% of indies agree that their testing results are subjective and open to interpretation (Q26-d). This is because indies focus more on experiential playtesting, lack clearly defined goals, and use unstandardized documentation procedures. Indies also describe outputs that lack variety and specificity, such as text-based logs, verbal test output, and brief text notes about issues. For example, G11-I stated that they *"used $debug.log()$ a lot."* and G17-I explained that they *"had a group discord, when things came up they were put there. We liked to call it the "explain it like I'm 5" rule."*

Only 18.9% of indies agree more than non-indies that they track or visualize results over time as metrics (Q32-f). This is not surprising given that they are testing without clear, standardized outputs. Conversely, non-indies mention formal report templates, standardized output from tools, and access to dashboards for tracking various telemetry and game health indicators. For example G22-N explained that *"Automated tests would specify which tests failed. It would also specify the particular item in the test logic that failed. Regression and Smoke tests by the QA department would yield specific and reproducible test steps that developers could look at in order to fix bugs and defects."*

C. Testing Resources

Table III shows that 95.9% of indies agree that testing is done primarily by team members who have other primary roles (Q34-h). In other words, quality assurance is a secondary or an additional role for the majority of indie developers. In contrast, the vast majority of non-indies (91.4%) feel that they have enough resources in terms of QA tools available to them (Q34-d). We have also observed this stark difference through the responses of both groups to our free-text questions. For example, G8-I states that *"There were no dedicated testing machines, no cloud-based tools, nothing of the sort."* and that their team *"weren't allotted a certain amount of developing and testing time - rather, we were expected to allocate our own time accordingly."* Indies feel that these factors lead to less availability for testing, longer testing times, and inferior product outcomes compared to what they could accomplish with more resources. G4-I emphasizes this hope when they said *"We invested in an external team for about 1 week of QA testing. This was mostly done to pass console certification. More would certainly have been nice."*

On the other hand, non-indies have dedicated QA staff, access to external testers, time and budget allocated for testing, and extensive hardware availability. Nevertheless, several non-indies report the need for more testing resources, which is likely connected to their reports of extreme overtime and crunch. This contradiction between abundance of resources and still needing more is highlighted by G10-N as their team *"had and have hundreds of testers on this project and it didn't/doesn't feel like enough sometimes."*

D. Test Automation

Nearly all indies state that they perform very little to no automated testing on their project. Indies use automation for build processes and build verification, data and scene validation, and integrated tests that shipped with the game to make bug reporting easier. These tools are successful at achieving their aim but require some configuration for effective use. Indies who do not use automation approach it with skepticism; they find it to be several times more work than manual testing, which is costly given their already-constrained time resources. G16-I expressed this concern when they said *"I think I tried one automated tool but it didn't play well with the laptop I was using for development at the time. I also wasn't sure if taking the time to learn such tools would ultimately pay off with the limited resources."*

Non-indies are more evenly split in their proportion of automated to manual testing, but report more manual testing overall. Non-indies use automation to take over for common and tedious tasks, increase test coverage, perform integration tests, and run network tests. Developers are responsible for writing automated tests for the features they implement which uses up time they could have spent on the game itself. However, they feel that the time spent on automation was worth it in the end. G7-N summarized the main uses of automation in their project as *"In terms of automated testing we had an AI-driven bot system that would run 24/7 to increase test coverage and trigger asserts and crashes."* Nevertheless, non-indies say that pressure from above to implement new features interferes with their ability to spend time on automation and that not understanding the role of automation and not dedicating enough resources to their maintenance leads to poor results.

E. Testing Tools

Despite not finding any quantitative differences in sentiment between indies and non-indies when it comes to the tools they use, the qualitative results show stark differences. Indies primarily use free tools, many of which are general-purpose tools not specific to games or testing, such as Discord [22], a messaging and community platform; Git Large File Storage [23] for large assets; Redmine [24], a free and open-source project management webapp; Trello [25], a project management software that can be used without a paid subscription.

Conversely, non-indies use a large number of paid tools specifically developed to aid with testing, planning, documentation, and tracking, such as Azure DevOps [26], DevTrack [27], Team City [28], and Test Rail [29]. The first-party tools that non-indies develop to help with testing are more complex in their functionality and specific to their goals, whereas indie first-party tools are more simple and broad in their use. However, the complexity of non-indie first-party tools leads to scenarios where they are improperly

maintained over time and become ineffective as a result. G2-N summarized it all when they stated that *"[we use a] lot of in-house tools. However, most tools were either 'This is old and doesn't work that well' or 'This is new and it COULD be cool if it worked'."*

F. Culture

Non-indies have the most concerns about culture and treatment of QA. They describe nightmare projects with an excess of overtime that lead to burnout, being laid off en masse upon project completion, and QA personnel who are treated as inferior to the rest of the development team. For example, G12-N describes one of their projects as *"brutal, and one of the best examples I've ever been in as far as "traditional" QA practices. There was a ton of burn out, a ton of overtime, and everyone got laid off at the end."* Another participant, G2-N, raised their concern about how *"Full time QA in games is primarily seen as unskilled work, leading to low wages and staff trying to use it as a springboard."*

Indies do not report experiencing the same level of workplace toxicity, most likely due to QA generally being an added responsibility to existing primary roles instead of a dedicated position. However, they are affected in other aspects such as *"[having] proper means of communication, being able to openly ask questions without "feeling dumb""* as G4-I has explained. Equal pay, compared to game developers, is a major concern that both indies and non-indies have raised. G17-I has clearly made the point that *"QA is game dev and a skilled dicipline. Anyone who says otherwise is wrong. VALUE YOUR QA AND PAY THEM BETTER."* G10-N has echoed the same sentiment of *"Valuing QA more, both in including it as part of development and in supporting QA staff. Pay QA on the same level as other employees. QA are developers as well."*

As the responses have shown, culture is a major issue in the game industry, which has been receiving more public focus recently, especially the exploitation of game workers and toxic development practices such as forced crunch.

V. Discussion

Given our survey results, we would like to contextualize our findings with respect to our original research questions.

A. Most Significant Differences (RQ1)

The most significant differences in game testing between indies and non-indies are that indies have access to fewer resources, do not have clear testing goals and plans for features or the overall project, and do not use as much automated testing compared to non-indies. The majority of testing for indie games is done by team members who have other primary roles and not dedicated QA personnel. Finally, indies feel that they need more testers and better tools for their project needs.

B. Use of Automated Testing (RQ2)

Indie developers mainly use manual testing with very little reliance on automation, whereas non-indies use slightly more manual than automated testing. The biggest pain points for indies is the lack of knowledge of automated testing, inaccessible or incompatible automation tools, and uncertainty about investing time into learning and creating automated tests. Non-indies face difficulties with contradicting priorities taking time away from automation, and warn against using automation without enough time or resources dedicated to maintaining them.

C. Implications

Given our data, we provide a few recommendations to indie developers that improve their approach to testing. First, we recommend that indie developers focus on defining clear testing goals. The more specific and defined the goals, the easier it is to test for them and obtain actionable results. Indies should keep current about testing methodologies and tools, including automation, and share their experiences and learnings with others. Without knowing the state of the practice, they cannot select the right tools for the job and may be missing out on more effective or efficient solutions. A key activity for increasing productivity is identifying the most critical tests and making them as frictionless as possible. Finding or creating tooling around these critical tests, such as automating the most frequently-run or time-consuming tests, frees up valuable developer time.

Instead of relying on ad-hoc practices, indies should set a regular recurring schedule for testing. Testing routinely, indies can be more thorough and systematic about their testing tasks and coverage, and improve expectations and transparency around the QA process. Indies should also be careful to test in a way such that the results are not subjective and open to interpretation. Some subjectivity is impossible to avoid, especially in matters of user experience and player enjoyment, but results should generally be bounded and measurable to be actionable. Using mock tests or outputs to trial how those results will be used to effect change on the project can help illustrate misconceptions or gaps in the process. Objective results can also help them track and visualize key metrics over the course of the project, which can be used to demonstrate development progress and game health.

VI. Threats to Validity

Our collected sample size is relatively low, but is within acceptable boundaries for the Mann-Whitney U-tests and reflexive thematic analysis that we conducted. Our respondents also covered a broad range of industry experiences, with varying experiences in roles and project types. While we did not have much representation for developers with AA experience, the organization of our data into indie and non-indie groups obviated the need for a third category.

The respondents for our survey were self-selected from social media and online game development communities. The distribution of their demographics may reflect our recruitment methods. Therefore, our findings may not be generalizable to populations outside of this surveyed group.

Our survey was long with a completion rate of 23% and an average completion time of 48 minutes. The length of our

survey may have led to respondent fatigue, resulting in reduced attention and motivation in answering questions. However, we did not see evidence of a decrease in the quality of answers towards the latter end of the survey, such as straight-line answers on Likert-like responses or short, vague responses in open-ended text responses.

We used Likert-like prompts extensively in our survey. These instruments are susceptible to central tendency bias, or the likelihood of selecting more neutral choices than extremes, but we found most of our responses to be skewed toward one of the extremes. They can also show acquiescence bias, which is the tendency for respondents to simply agree with the prompt. To avoid this bias, we phrased our prompts in different ways. For example, the two prompts "We primarily used testing tools we developed in-house" and "We primarily used third-party testing tools" would be in conflict if a respondent simply agreed with all prompts. We also randomized the order of rows for each set of Likert-like prompts.

For our project-specific questions, we asked participants to answer based on their experiences on a completed commercial product. Respondents may be more likely to remember or report certain types of experiences than others, and their memory or sentiment regarding the process may have changed since that time.

VII. Related Work

Murphy-Hill et al. [30] have found that practices used in software engineering are not well-suited for games, primarily because the goal of game development is *fun*, which is highly subjective and difficult to form objective requirements around. The authors note little reuse of tools and code between games because games need project-specific performance tuning, as well as low use of automation due to its cost and fragility to frequent changes. These findings are confirmed by Politowski et al. [31], who conducted a review of 96 academic papers as well as gray literature (i.e., post-mortems, game development conferences, web articles) on QA for games. Moreover, they found in the gray literature that developers feel there is insufficient testing overall and that they have difficulty setting up testing tools.

Politowski et al. [32] analyzed 927 problems from 200 post-mortems from games between 1997 to 2019 that they divide into 20 categories. From the testing category, the most mentioned issues are insufficient test coverage and issues with process and testing plans, followed by specific project requirements and a scope too large to properly test. The postmortems mention playtesting but not unit or integration tests or automated testing.

Kasurinen and Smolander [33] conducted semi-structured interviews with 27 game developers from southeast Finland to find out what and how game developers test their products. They apply grounded theory to their interviews and found 3 main categories for testing: (1) game mechanics (e.g., encompassing game rules, balance between features, and overall design), (2) technical aspects (e.g., functionality and stability of the software), and (3) user experience (e.g., focusing on the fun of the game, player impressions, and overall user satisfaction). The interview participants stated that user experience testing is the most important piece, and that they consider themselves to be creatives, not software engineers.

Politowski et al. [34] have shown that game developers are concerned about the amount of data and time required to train AI agents and the cost of setting up the end-to-end process. Respondents have also stated that they want automated testing tools that are easy to maintain and not specific to the game being tested.

In a comprehensive classification of automated game testing methods, Albaghajati and Ahmed [35] have identified 5 primary categories of automated tests: search-based, goal-directed, human-like, scenario-based, and model-based. The authors discuss a number of shortcomings in the field, including a lack of automated methods to verify procedurally generated content, difficulties in avoiding agent bias for certain genres, and game testing examples having limited state space complexity (e.g., tile-matching games).

VIII. Conclusion

The video game industry is ever growing and has its effects on entertainment, culture, and art. In this paper, we examined the state of testing for games to better understand QA practices in indie game development, and how they differ from non-indie game development. While indie games represent the majority of games released each year, indie developers are constrained by their limited resources in comparison to large studios releasing blockbuster games. Surveying 19 game developers who have participated in releasing 22 commercial games, we were able to identify the most significant differences between indie and non-indie testing, automation usage, and major pain points. We then provided several suggestions for indie testing to maximize their resources.

We see this work as being an initial exploration into the differences between indie and non-indie game development and testing with ample avenues for further research. In particular, performing a similar study as a series of semi-structured interviews, as opposed to open-ended text responses on an online survey, may provide richer and more specific information from respondents. We also think it would be worthwhile to compare practices between different indie studios, instead of comparing them against non-indies, due to the extreme variance that can exist within that group.

Acknowledgment

This research has been partially funded by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] J. Clement, "Video game market value worldwide 2020 to 2025," 11 2021. [Online]. Available: https://www.statista.com/statistics/292056/video-game-market-value-worldwide/

[2] W. Witkowski, "Videogames are a bigger industry than movies and North American sports combined, thanks to the pandemic," 12 2020. [Online]. Available: https://www.marketwatch.com/story/videogames-are-a-bigger-industry-than-sports-and-movies-combined-thanks-to-the-pa

[3] "2021 essential facts about the video game industry," 2 2022. [Online]. Available: https://www.theesa.com/resource/2021-essential-facts-about-the-video-game-industry/
[4] R. Ebert, "Video games can never be art." [Online]. Available: https://www.rogerebert.com/roger-ebert/video-games-can-never-be-art
[5] "The art of video games." [Online]. Available: https://americanart.si.edu/exhibitions/games
[6] O. Solon, "MoMA to exhibit videogames, from Pong to Minecraft," 11 2012. [Online]. Available: https://www.wired.com/2012/11/moma-videogames/
[7] Superannuation, "How much does it cost to make a big video game?" 1 2014. [Online]. Available: https://kotaku.com/how-much-does-it-cost-to-make-a-big-video-game-1501413649
[8] J. Schreier, "Former PlayStation chief muses on the future of gaming," 2021. [Online]. Available: https://www.bloomberg.com/news/newsletters/2021-09-03/ex-playstation-chief-mulls-future-of-gaming-and-his-new-job
[9] M. Dawson, D. N. Burrell, E. Rahim, and S. Brewster, "Integrating software assurance into the software development life cycle (SDLC) meeting Department of Defense (DoD) demands," *Journal of Information Systems Technology and Planning*, vol. 3, 2010.
[10] A. Saini, "Cost to fix bugs and defects during each phase of the sdlc," 10 2021. [Online]. Available: https://www.synopsys.com/blogs/software-security/cost-to-fix-bugs-during-each-sdlc-phase/
[11] "Steam store." [Online]. Available: https://store.steampowered.com/
[12] "Games industry data and analysis." [Online]. Available: https://vginsights.com/insights/article/indie-games-make-up-40-of-all-units-sold-on-steam
[13] J. J. Cho, "Game testing survey artifacts." [Online]. Available: https://github.com/gojeffcho/game-testing-survey-artifacts
[14] M. Kasunic, "Designing an effective survey," Software Engineering Institute, Carnegie Mellon University, Tech. Rep. CMUSEI-2005-HB-004, 2005.
[15] L. A. Goodman, "Snowball sampling," *The annals of mathematical statistics*, pp. 148–170, 1961.
[16] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The Annals of Mathematical Statistics*, vol. 18, 1947.
[17] K. O. McGraw and S. P. Wong, "A common language effect size statistic," *Psychological Bulletin*, vol. 111, 1992.
[18] R. Bergmann, J. Ludbrook, and W. P. J. M. Spooren, "Different outcomes of the Wilcoxon-Mann-Whitney test from different statistics packages," *The American Statistician*, vol. 54, no. 1, pp. 72–77, 2000. [Online]. Available: http://www.jstor.org/stable/2685616
[19] V. Braun and V. Clarke, "Reflecting on reflexive thematic analysis," *Qualitative Research in Sport, Exercise and Health*, vol. 11, no. 4, pp. 589–597, 2019. [Online]. Available: https://doi.org/10.1080/2159676X.2019.1628806
[20] D. Byrne, "A worked example of Braun and Clarke's approach to reflexive thematic analysis," *Quality and Quantity*, 2021.
[21] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77–101, 2006. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1191/1478088706qp063oa
[22] "Discord (app)." [Online]. Available: https://discordapp.com/
[23] "Git Large File Storage (software)." [Online]. Available: https://git-lfs.github.com/
[24] "Redmine (software)." [Online]. Available: https://www.redmine.org/
[25] "Trello (software)." [Online]. Available: https://trello.com/
[26] "Azure devops services: Microsoft Azure (software)." [Online]. Available: https://azure.microsoft.com/en-us/services/devops/
[27] "Devtrack, best project task management software." [Online]. Available: https://techexcel.com/products/devtrack/
[28] "TeamCity: The hassle-free CI and CD server by jetbrains." [Online]. Available: https://www.jetbrains.com/teamcity/
[29] "Test Rail: Test management & QA software for agile teams." [Online]. Available: https://www.gurock.com/testrail/
[30] E. Murphy-Hill, T. Zimmermann, and N. Nagappan, "Cowboys, ankle sprains, and keepers of quality: How is video game development different from software development?" 2014.
[31] C. Politowski, F. Petrillo, and Y. G. Guéhéneuc, "A survey of video game testing," 2021.
[32] C. Politowski, F. Petrillo, G. C. Ullmann, and Y. G. Guéhéneuc, "Game industry problems: An extensive analysis of the gray literature," *Information and Software Technology*, vol. 134, 2021.
[33] J. Kasurinen and K. Smolander, "What do game developers test in their products?" 2014.
[34] C. Politowski, Y.-G. Guéhéneuc, and F. Petrillo, "Towards automated video game testing: Still a long way to go," 02 2022.
[35] A. M. Albaghajati and M. A. K. Ahmed, "Video game automated testing approaches: An assessment framework," *IEEE Transactions on Games*, 2020.