

Breaking the cycle of ineffective game feedback

The key to better games,
cleaner code and happier –
and healthier – developers

2023 GAMING EXPERIENCE REPORT



Contents

3 < Introduction >

4 < Key findings >

5 SECTION 1 < Glitchy games: Is the pressure for early release internal? >

12 SECTION 2 < Developers struggle when negative gamers dominate feedback avenues >

18 SECTION 3 < Better feedback and the quest for “respawn gamers” >

23 SECTION 4 < Leveling up developer feedback tools >

27 < Conclusion >

28 < Methodology >



INTRODUCTION

Video game development is often a battle royale of pressure

Millions of dollars on the line.

Thousands of hours in labor from countless contributors — developers, testers, engineers and designers. Throngs of impatient fans, accelerated and looming deadlines, unrealistic media expectations and a long list of bugs to address before reaching gold master.

Mix these variables together, put them on a map, and you've entered hostile territory. Anxiety about game release is especially tense given the consequences of recent, messy launches.

We've seen firsthand how a disappointing game kills the hype surrounding a new release:

- 1 [Battlefield 2042](#) proved even AAA games can launch with Grade-A glitches — which damage brand reputation.
- 2 At the same time, launches such as Fallout76 and Cyberpunk 2077 show sub-optimal releases aren't always game over — [but the effort to save them can feel like it](#)
- 3 Mobile developers and their games also suffer unique launch challenges — device inconsistencies, a broader player base and platform limitations compound pressures and inspire unique gamer frustrations. Add in factors like device farms and the propensity for more fraud in mobile gaming, and mobile gaming developers are working at a nightmare-mode level of difficulty.

Disappointed fans' loud voices and management's unrealistic expectations often lead to professional and personal challenges for developers. These struggles impact developer work, fueling a negative feedback loop with gamers, burnout and future mistakes.



Key Takeaways

In a survey of both 150 video game developers and 500 gamers, we asked developers to help us understand the unique pressures that lead to buggy releases — and we asked gamers to tell us how they react when they encounter unpolished production.

We learned

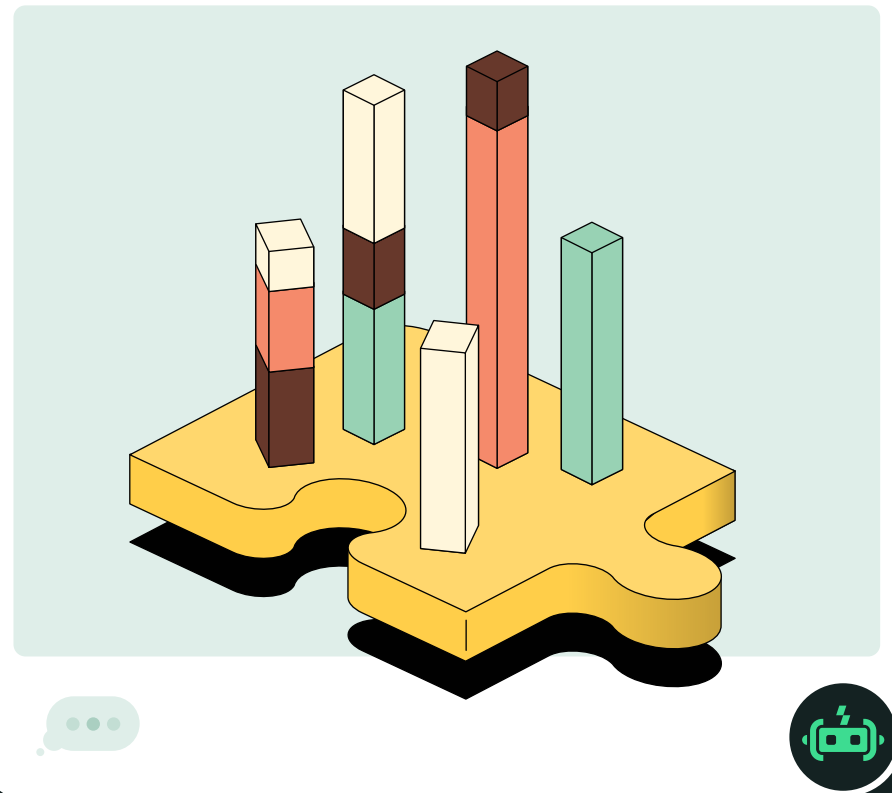
- 1 Why games that should stay in beta get pressured into release
- 2 What gamers really do when games glitch
- 3 The true mental and physical toll bullying and harassment have on developers
- 4 Why addressing bugs during pre- and post- release is such a challenge
- 5 Why mobile developers need better feedback solutions

Our data is clear:

It's time to patch the broken processes in video game development currently leading to unfinished, unpolished and bug-filled games. Brand reputation, gamer loyalty and revenue to fund future releases depends on it.

How can developers break this negative cycle and meet gamer and corporate expectations?

Our team at Sauce Labs did the reconnaissance to find out.



1

Glitchy games:

2

Is the pressure

3

for early release

4

internal?

Pressure to release is a boss fight

The concept of “development hell” is often associated with throngs of angry gamers screaming for launch day. But even more frequently, the pressure to release games before they’re ready — and while they still contain critical errors — is coming from inside the house.

Nearly two-thirds (61%) of developers we surveyed have been pressured to release a buggy or unfinished game — and anyone close to game development can imagine what these internal pressures feel like. Investors become impatient. Mobile versions of games increasingly need to accompany major console releases. Developer churn forces teams to release “as-is” and move on. Any number of scenarios can lead to a low-quality product and a sub-par, premature launch.

61%

of developers say they have released a game under pressure before it's fully debugged or ready



A rushed job has consequences

Nearly three-quarters of gamers (74%) said they avoid buying games from a developer who previously released a buggy game.



The majority of developers who have rushed a game to market did so due to pressure from their team, investors or executives above them [Fig 1].

Despite gamers' loud voices on social media and in online forums, developers rarely feel pressure from gamers around early releases at the level they experience within their own professional networks.



FIG. 1

Pressure to rush games is coming from inside the house

32%

Your team

25%

Executives/C-suite

18%

Investors

13%

Yourself

12%

Gamers

Devs grapple with mobile shift

Increased pressure to rush to market comes as gaming demographics continue to shift — namely, gamers moving toward mobile [Fig 2].

Mobile game development presents unique challenges and mobile game fans have unique concerns. Increasingly, mobile game tie-ins accompany the release of major movies as well as the biggest gaming IPs like Fortnite. It's a big opportunity for third-party developers that can create games out of the biggest blockbusters, but it's also a risk. Buggy, glitchy or error prone releases not only reflect poorly on the developers' brand, but also the original work.



FIG. 2

Gamers increasingly skew toward mobile

Mobile (e.g., iOS, Android, etc.)

73%

Sony Playstation

45%

PC

40%

Microsoft Xbox

29%

Nintendo Switch

21%

Web/browser games

12%

Steam Deck

2%

SECTION 1

Combined with internal pressures, developers know how the game plays out with gamers and fans during a sloppy release — adding more stress to the development cycle. Rude comments, invasive DMs, rage-induced Reddit posts: The consequences of a low-quality or error-filled game making it to market are both seen and unseen by developers [Fig 3].

In development, “mean time to recovery” MTTR is a metric often discussed when a product launch goes wrong — it represents the average time it takes your organization to recover from the fallout. But the above data suggests another trend: Nearly a third (31%) of gamers have reacted to a buggy game by stopping play themselves or encouraging others to stop play through a variety of behaviors.

Developers who wince at MTTR should fear another acronym based on these numbers — [mean time to Reddit post \(MTTRP\)](#). How long after a game crashes or glitches are gamers taking to forums to express frustration? What kind of errors are so frustrating that they require venting to similarly invested Redditors?

33%

have requested a refund of a game they considered broken or glitchy.

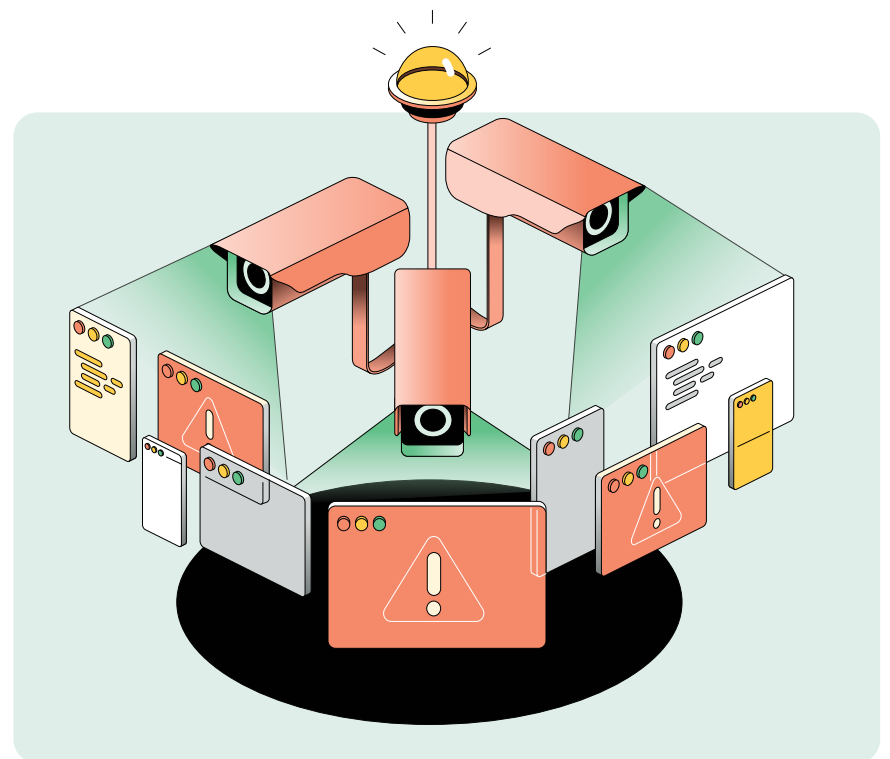


FIG. 3

When gamers rage quit

Audibly yelled or shouted at the screen

54%

Checked release notes or the developer's site to see if the bug is a known issue

36%

Abandoned the game and never played again

31%

Threw a piece of gaming equipment in anger

27%

Contacted one or more developers of the game via social media

19%

Taken a photo or video of the bug and posted to social media

14%

Wrote negative social media or forum posts

11%

Played incorrectly or interfered with other players' experiences (griefing) to make a point

11%

Encouraged friends to stop playing the game

9%

After they finish shouting, gamers' second most common reaction to a bug is checking a developer's site to see if the bug or error is known to the developer. This requires transparency. Are developers equipped with the tech to track and report known issues? Are they doing enough?

And with a third of gamers prone to ask for refunds, there's a significant concern about loss of revenue.



KEY TAKEAWAYS

There's no shortage of cautionary tales and failures in gaming launches – **so why do they keep happening?** And what are the consequences when they do, both financially and for the health of development teams?



1 Developers struggle
2 when negative
3 gamers dominate
4 feedback avenues

Video game development plays out like a recursive code loop

As if personal pressure to release the perfect product isn't enough, co-workers and bosses add to the demand that games release quickly. Reddit threads and Twitter Spaces are dedicated to the development hell video game creators now find themselves in, and the list of game-breaking bugs is completely overwhelming.

A hasty release enrages fans (who expected something more polished), and now the same co-workers and bosses who rushed products to market are demanding fast answers and solutions. **Despite high praise for a game's potential coming from many positive voices, criticism drowns out more hopeful discourse.**

48%

of developers have felt threatened or bullied online because of a game they've worked on.



This feedback loop isn't just counter-productive – it's harmful bullying and harassment [Fig 4].

Buggy game releases take a physical and mental toll on developers. Beyond the individual suffering, the statistics indicate a larger industry problem. **Nearly 1 in 5 developers have thought about leaving for another company or quitting the industry** altogether — a disheartening statistic for those who want to see technical talent enter, not exit, the industry. Especially when finances and the talent market get tight, this kind of dissatisfaction is a concern for development companies and leadership teams.

FIG. 4

Negative feedback takes its toll

Depression/disappointment

45%

Stress/overwhelm

41%

Negative impact on physical health
(i.e., insomnia, nail biting, changes in weight)

33%

Negative impact on mental health
(i.e., panic attacks, clinical depression, dissociation)

28%

Fear for safety due to threats

19%

Wanting to change careers

19%

Wanting to work for another company

17%

Anger/frustration

15%

Agreement with the negative feedback

8%

As mobile gaming audiences grow, so do expectations:

- 1 Companion apps and mobile game tie-ins are becoming normal accompaniments to Triple-A releases.
- 2 Fans of IPs like Fornite and Minecraft expect the same smooth, low-glitch experience they access on PC or console via their mobile device — and they don't want a pared down experience, either.
- 3 Poor experiences within microtransactions and progression in mobile games leave gamers feeling slighted.



SIDEBAR

Development's mobile shift

79%

Nearly 4 in 5 developers say the pressure to release unfinished or buggy games has increased over the last five years

85%

say remote work has made it easier for errors to occur as teams split work between locations



SIDEBAR

Mental health checkpoint

Much has been written about the mental health of [both gamers and developers](#), and our survey data unfortunately confirms that developers experience a range of negative consequences during stressful game releases. It's important for developers — as well as their managers and leadership teams — to understand the unique challenges that come with the job and impact mental well-being.

Here are a two resources to consider:

1 [CheckPoint.org](#):

A non-profit that explores mental well-being in gaming communities.

2 [GamingtheMind](#):

UK mental health professionals and self-professed gaming fans have assembled resources about how mental health, accessibility and social issues relate to gaming.

Despite struggles, developers still think highly of gamer feedback

91%

of developers say gamers have a generally positive attitude toward developers.

84%

of developers say gamers are more likely to give positive feedback than negative feedback.

With a horde of positive feedback to work from, **what's stopping developers** from receiving the message and making the most of that intel?



KEY TAKEAWAYS

While developers want to have relationships with gamers and see value in gamer input, **boundaries are required** to keep interactions healthy and constructive. What happens when developers tune out the loudest negative voices and instead, focus on improving experiences for everyone?



1 Better feedback
2 and the
3 quest for
4 “respawn gamers”

Why glitched out gamers respawn

A poor-quality launch leads to brand damage and deters future players. But not all gamers abandon games for good after a choppy or disappointing first impression. Meet the “respawn gamers” — the hidden majority (67%) of gamers who have jumped back into a buggy game after a period of time.

No one in gaming wants to see a flop. But for developers and their employers, there’s still opportunity for a comeback storyline after a launch gone wrong. Gamers are willing to return — and many of the experiences they come back for can be fixed and maintained more efficiently with better development tools [Fig 5].

Gen Z (81%) and millennials (79%) were more likely to play a buggy game again after it was patched than Gen X (60%) and baby boomers (38%). Older gamers seem to be much more impacted by first impressions, while younger gamers tend to be more forgiving. But older gamers — who increasingly [see video games as a family activity](#) — are too valuable to turn away.

FIG. 5

Gamers’ reasons for respawning in a glitchy game

The game recently got a big update/DLC I wanted to check out

51%

The player base grew significantly since I last tried it

36%

I read an article or watched a video that said the experience improved

34%

My friends started playing it

27%

I wanted to try the game on a different platform

24%

There was an in-game event I wanted to see

19%

I saw a film/TV show tie-in that made me interested again

18%

SIDEBAR

Bringing back mobile gamers

Mobile game developers face unique challenges (and have unique advantages) in gathering feedback and addressing issues during product launch.

- 1 **Faster release turnaround**
- 2 **Feedback from disparate devices /operating systems**
- 3 **Shorter gamer attention span**

Thankfully, there doesn't need to be a catastrophic launch failure to see the benefit of better development and testing tools.

Big names in mobile have already seen benefits, including:

- 1 **Roblox**
Used custom tooling and a singular debugging interface to meet significant demand to scale their game due to fast user growth. Now, the game's developers get immediate notifications when stability issues affect new releases.
- 2 **Cryptic**
Improved bug submission times and made updates to how they prioritize issues, leading to 100% coverage and better responsiveness to new issues.
- 3 **Amazon**
Made improvements to accuracy and prioritization of fixes, replacing their previously slow and inflexible crash reporting system.



SIDEBAR

Evolving tools for bug management

Developers need tools that allow them to test on real and virtual devices, which is especially critical when developing for mobile. With tools from Backtrace, developers can beta test, launch, debug and patch games with more confidence (and less fear of the MTTRP).

1

Expedite automated and live testing of mobile games in thousands of Android and iOS devices, even older device models or OS versions.

2

Analyze crashes more effectively with the ability to group, filter, or aggregate on any custom attribute, and utilize custom views of data designed for debugging.

3

Run tests across multiple devices in parallel, dramatically reducing test execution time.

4

Debug faster with a private device cloud using a virtual USB solution that simulates a direct connection to workstations

5

Better assess impact of fatal and non-fatal errors with the most accurate and flexible deduplication system.

6

Debug faster with support for a wide range of NDK and iOS versions, breadcrumbs, file attachments, and more.

7

Accurately test games for display (pixels, screen sizes, etc.), hardware dependencies, compatibility testing and more.



KEY TAKEAWAYS

What's stopping game developers from capitalizing on respawn gamers? And what's leading to the errors, bugs and glitches that drive gamers away in the first place?



1

Leveling

2

up developer

3

feedback tools

Devs struggle to parse user feedback

Pressure situations can lead to sub-par video game releases, along with all of the unhelpful and stressful feedback and reviews that accompany them. **But where do developers specifically struggle when it comes to managing and triaging feedback** [Fig 6/7/8]?

User feedback — both during release and beta testing — is wrought with unreliable data for developers to parse through and triage. During release, developers' biggest challenge is deciphering gamer feedback to make information actionable — and it appears they don't have much help finding clarity. **To make better games, developers require actionable, consistent and accurate feedback.**

The second biggest challenge developers face during game release is getting users to submit a manual report clarifying a bug or error. On the testing side, developers experience the most pain in organizing and triaging bugs.

FIG. 6

Challenges in the developer feedback pipeline

Insufficient details or lack of context to be actionable

55%

Not enough users willing to spend the time to draft and submit a manual report

45%

Not enough users providing feedback due to ethical/privacy reasons

39%

Filtering out fake reports/trolling

37%

Too many reports to process

34%

Reproducing reported problems

20%

FIG. 7

Challenges unique to beta testing

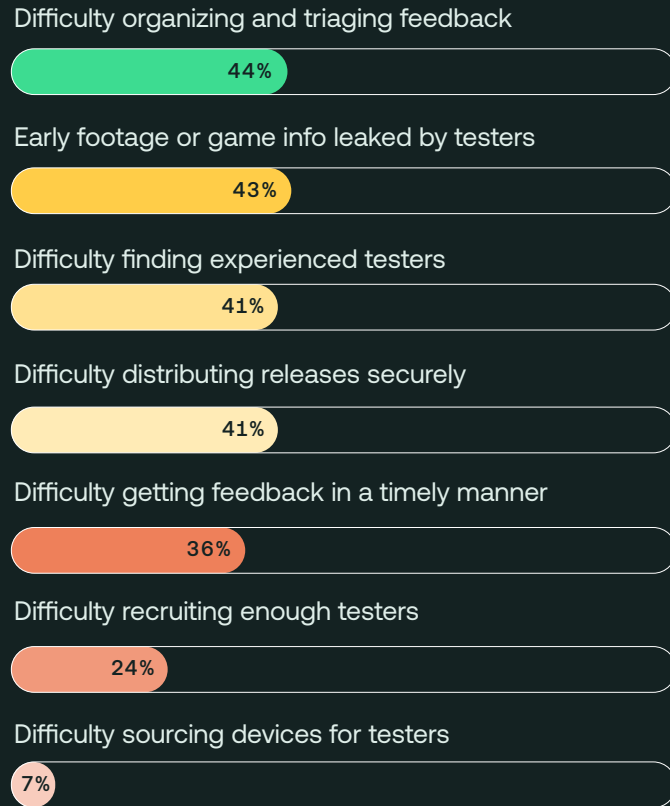


FIG. 8

What developers need



KEY TAKEAWAYS

The inability for developers to prioritize and efficiently address bugs points to tech and transparency deficits – **developers are most likely using an inefficient or inconsistent system to track bugs and errors based on gamer feedback**, meaning they miss out on key insights and opportunities to course correct.

With tools from Sauce Labs and Backtrace, developers bypass the need for user opt-in and manual crash reporting — they automatically receive contextual information about crashes and errors.



CONCLUSION

Build a positive feedback loop, release better games

Developers require real-time, actionable insights into gamer feedback, especially in critical situations like a game launch. But as it stands now, developers lack the tools to triage, organize and discover feedback generated in non-traditional ways — like via social media or forum posts.

However, with tools capable of parsing feedback in more meaningful ways, finding the source of major problems during launch is far easier and effective.

In situations where poor experiences make it to market, pivots are more possible than ever thanks to tools like Backtrace, which offer:

- 1 Custom report attributes:**
Add runtime data — like map data or server info — to crash reports for better triaging of major issues.
- 2 Data scrubbing:**
Automatically remove unwanted and personal (PII) data from large data dumps, eliminating compliance worries while working quickly.
- 3 Power query and search:**
Search, filter and analyze error reports and other data without the need for export.

Better tools can help developers make sense of the important but hard to surface feedback in places like Reddit forums and social media posts.



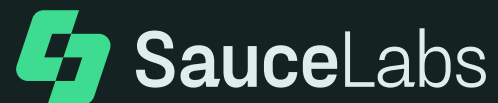
Backtrace and Sauce Labs are the power tools devs need in the fight against glitched games

But developers eager to avoid a launch day scramble in the first place are also positioned to level up. Better tools can help developers launch cleaner, better tested games so their work sidesteps unfortunate, unnecessary and embarrassing attention. Sauce Labs and Backtrace offer:

Methodology

Sauce Labs surveyed 150 full- and part-time game developers in the U.S. and 500 U.S.-based gamers in a survey conducted in November, 2022.

- 1 **True cross-platform error and crash capture:** desktop, web, mobile, device, game console and server.
- 2 **Integrated with leading game engines and consoles including:** Unity, Unreal, PS4 and Xbox One.
- 3 **Fast and accurate deduplication,** bucketing, correlation, and classification that filters out noise.
- 4 **Handle high ingestion volumes** without generating duplicate reports and notifications.
- 5 **Easy to use tools and integrations** to support your human workflows and collaboration efforts.
- 6 **Web-based debugger** provides sharable detailed crash reports, attachments, symbolicated call stacks, full thread, process and frame details.
- 7 **Issue-level resolution status,** comments, and tagging.
- 8 **Custom data analytics** and visualization tools designed for crash and error investigation.
- 9 **Filters and aggregation** on dimensions such as platform, release, region, user type, and more.
- 10 **Easy zooming** from individual error details out to issue trends across all deployments.



Looking to **respawn players**?

Want to **improve software** and game quality before launch?

Inspired to **help developers** feel happier and healthier?

Let's talk

