QUEST LAB

UNIVERSITY OF CALGARY

Beyond Single-PR Issues:

Understanding Multi-PR Issue Resolution in Game Development

Nimmi Weeraddana

Developers submit PRs to fix reported issues and update the main codebase

Stakeholder

Feature request/bug

Issue report

CLOSED

Single-PR issue

Developer

Make a code change

Initiate a PR

MERGED

Developers submit PRs to fix reported issues and update the main codebase

Stakeholder

Feature request/bug

Issue report

Multi-PR issue

CLOSED

Developer

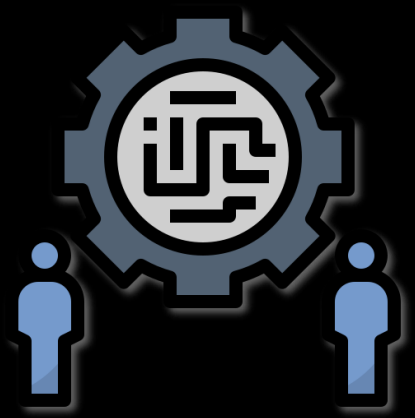
Make a code change

Initiate a PR

Initiate another PR

MERGED

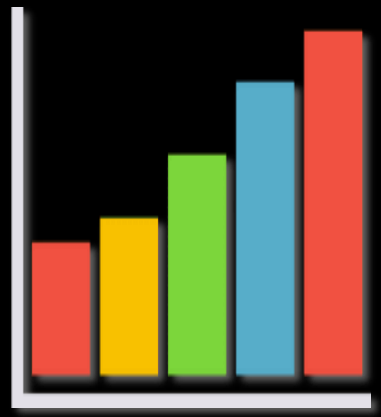
Importance of studying multi-PR issues in video game development



Reveal cross-system complexity

Leave players affected for longer

Despite their practical relevance, little is known about…

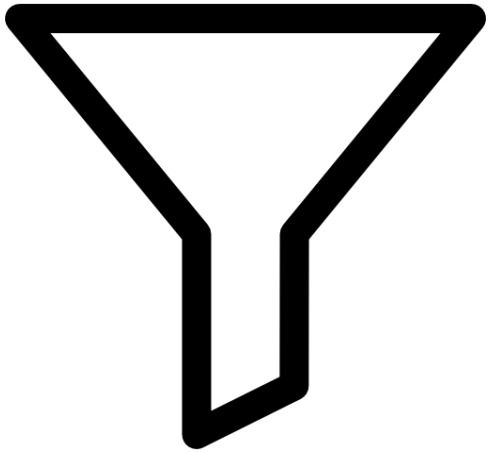


Prevalence Sources Impact

of multi-PR issues in video games

Explore multi-PR issues in video games

950 projects



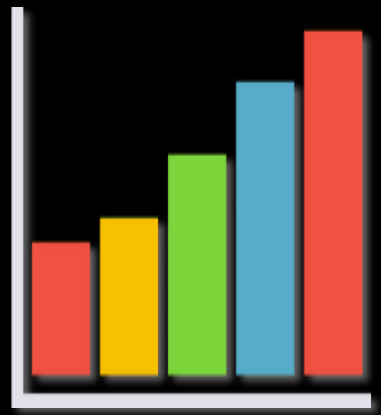
392 projects

OSSGameBench
(Marsad et al.)

Filter projects with more than 100 PRs

Final dataset for analysis

Explore multi-PR issues in video games



Prevalence

Sources

Impact

Approximately **one in twenty** issues is resolved through multiple PRs

Cross-project distribution of single- vs. multi-PR issues

Issue category	Number of issues	Percentage (%)
Single-PR issues	48,979	95.2%
Multi-PR issues	**2,387**	**4.8%**
Issues linked to at least one PR	51,366	100.0%

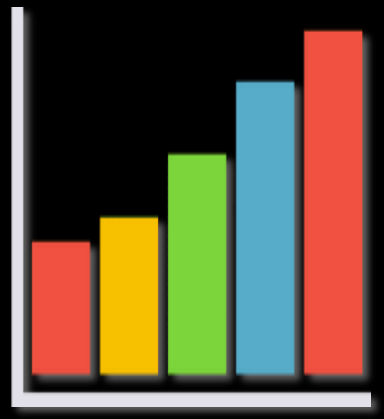
A subset of projects shows a **high proportion** of multi-PR issues

The **project-level multi-PR rate is 1.4%** (median), indicating that a typical project has a relatively low proportion of issues resolved through multiple PRs.

Project-level distribution of multi-PR issues

Project-level metric	Number of projects	Percentage (%)
Projects analyzed	392	100.0%
Projects with more than 10% multi-PR issues	**28**	**7.1%**
Projects with more than 20% multi-PR issues	**6**	**1.5%**

Explore multi-PR issues in video games



Prevalence

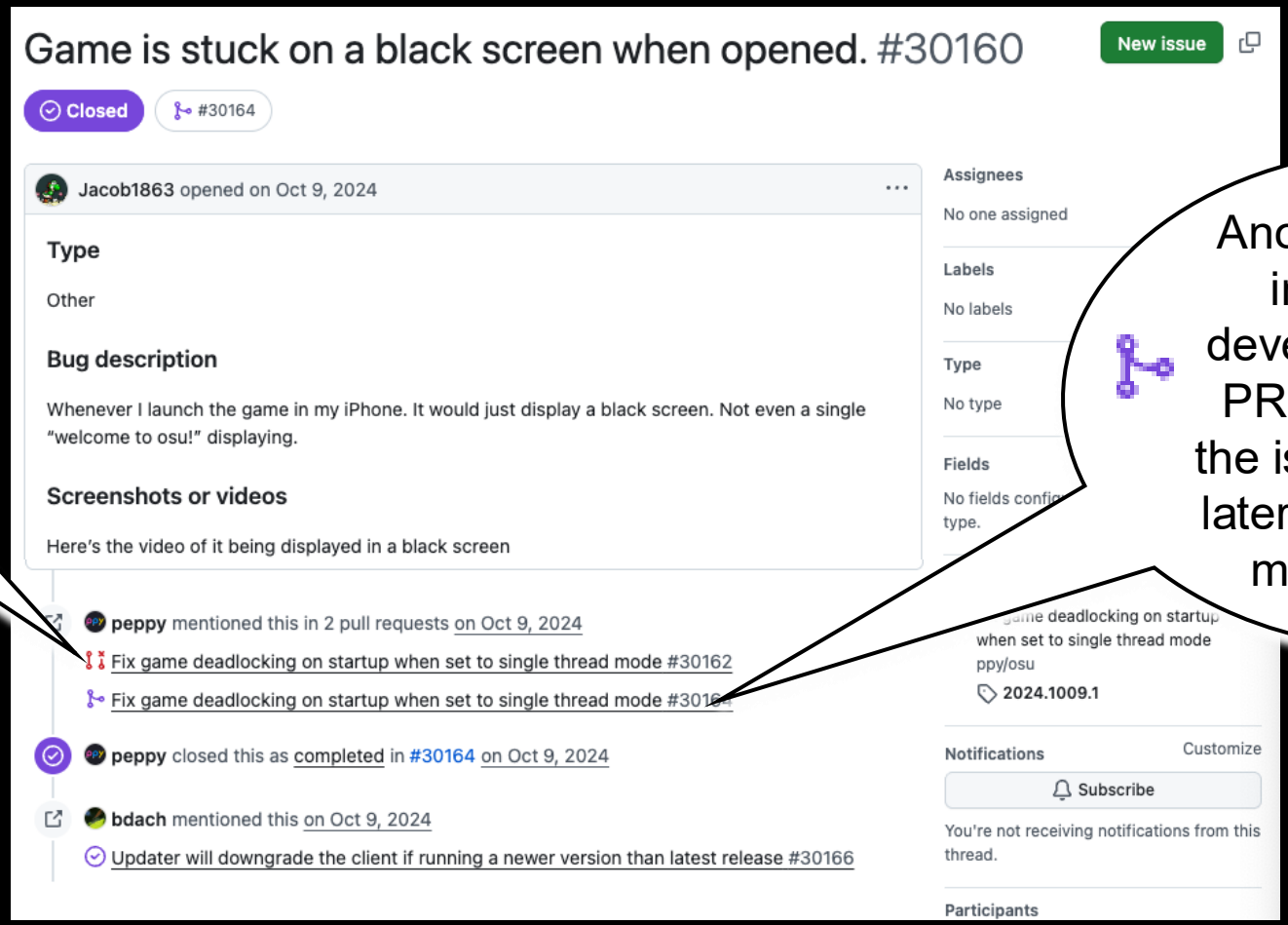
One in twenty issues is resolved through multiple PRs; a subset of projects shows a high proportion of multi-PR issues

Sources

Impact

The most frequent (48%) source is **incomplete** or **buggy PRs**

The first PR was closed without merging due to **failing test cases**



Another contributor independently developed a second PR that **fully fixed** the issue, which was later merged into the main codebase.

20% of the multi-PR issues are **extended PRs**

A PR that addresses the **core problem** (i.e., the PR is merged), followed up by PRs that **extend and refine** the solution

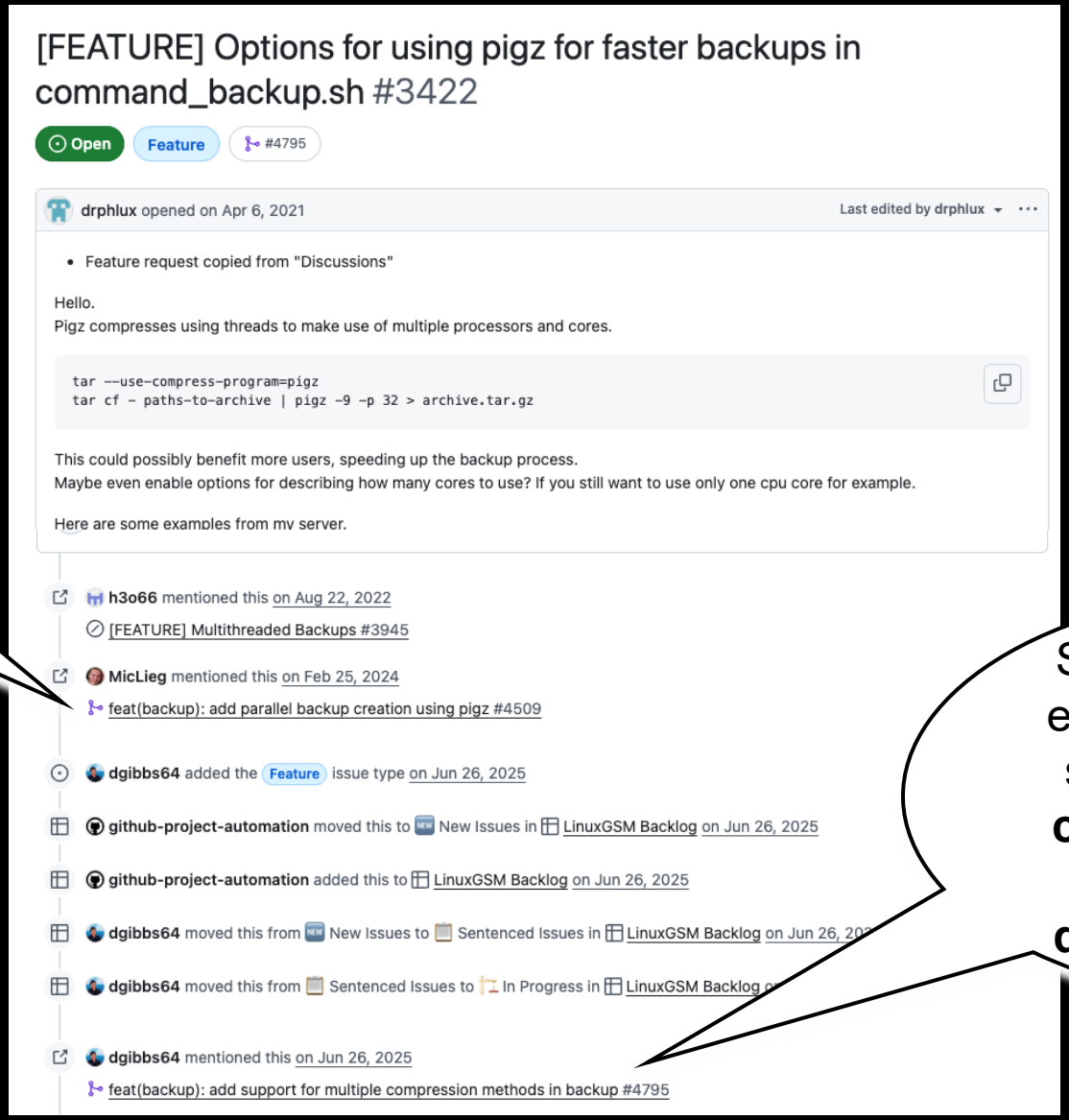
Additional features

Edge-case handling

Quality improvements that go beyond the original fix

20% of the multi-PR issues are **extended PRs**

Initial PR added support for **parallel compression**



[FEATURE] Options for using pigz for faster backups in command_backup.sh #3422

Open Feature #4795

drphlux opened on Apr 6, 2021 — Last edited by drphlux

- Feature request copied from "Discussions"

Hello.
Pigz compresses using threads to make use of multiple processors and cores.

```
tar --use-compress-program=pigz
tar cf - paths-to-archive | pigz -9 -p 32 > archive.tar.gz
```

This could possibly benefit more users, speeding up the backup process.
Maybe even enable options for describing how many cores to use? If you still want to use only one cpu core for example.

Here are some examples from mv server.

h3o66 mentioned this on Aug 22, 2022
[FEATURE] Multithreaded Backups #3945

MicLieg mentioned this on Feb 25, 2024
feat(backup): add parallel backup creation using pigz #4509

dgibbs64 added the Feature issue type on Jun 26, 2025

github-project-automation moved this to New Issues in LinuxGSM Backlog on Jun 26, 2025

github-project-automation added this to LinuxGSM Backlog on Jun 26, 2025

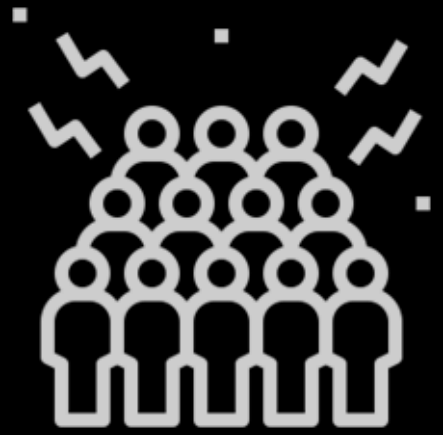
dgibbs64 moved this from New Issues to Sentenced Issues in LinuxGSM Backlog on Jun 26, 20

dgibbs64 moved this from Sentenced Issues to In Progress in LinuxGSM Backlog

dgibbs64 mentioned this on Jun 26, 2025
feat(backup): add support for multiple compression methods in backup #4795

Subsequent PR further extended this by adding support for **additional compression** methods and improving **dependency handling**

Other sources of multi-PR issues

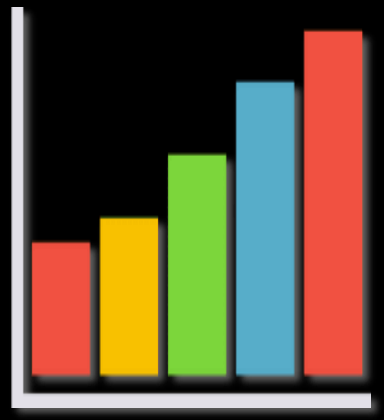


Crowded PR (8.3%)

Abandoned PR (5.0%)

Accidental PR (5.0%)

Explore multi-PR issues in video games



Prevalence

One in twenty issues is resolved through multiple PRs; a subset of projects shows a high proportion of multi-PR issues

Sources

The most frequent (48%) source is incomplete or **buggy PRs**; 20% of the multi-PR issues are **extended PRs**

Impact

Multi-PR issues exhibit significantly longer resolution

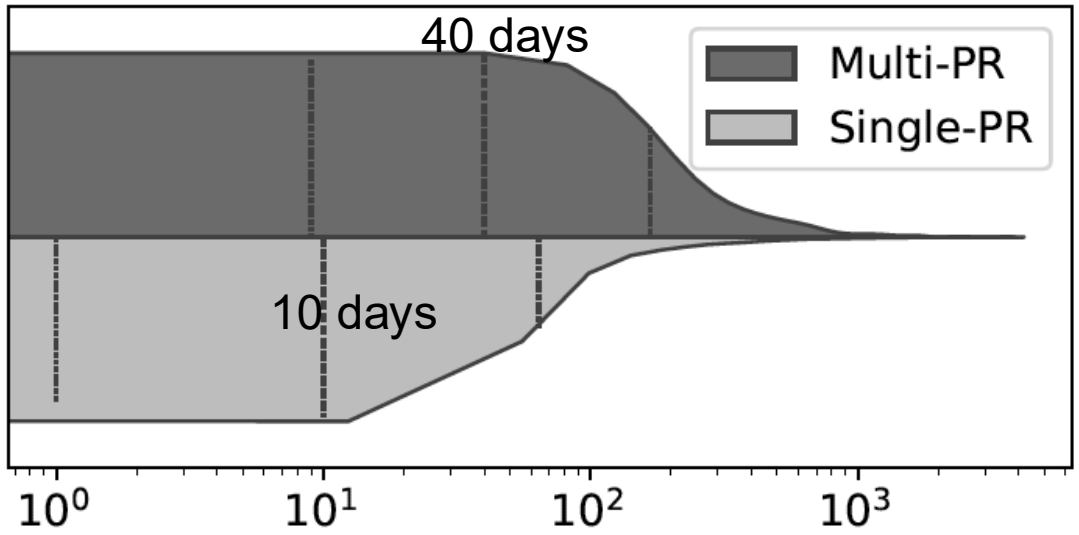


Figure 3: Distribution of issue resolution time in days.

Impact

Multi-PR issues generally require slightly more commits

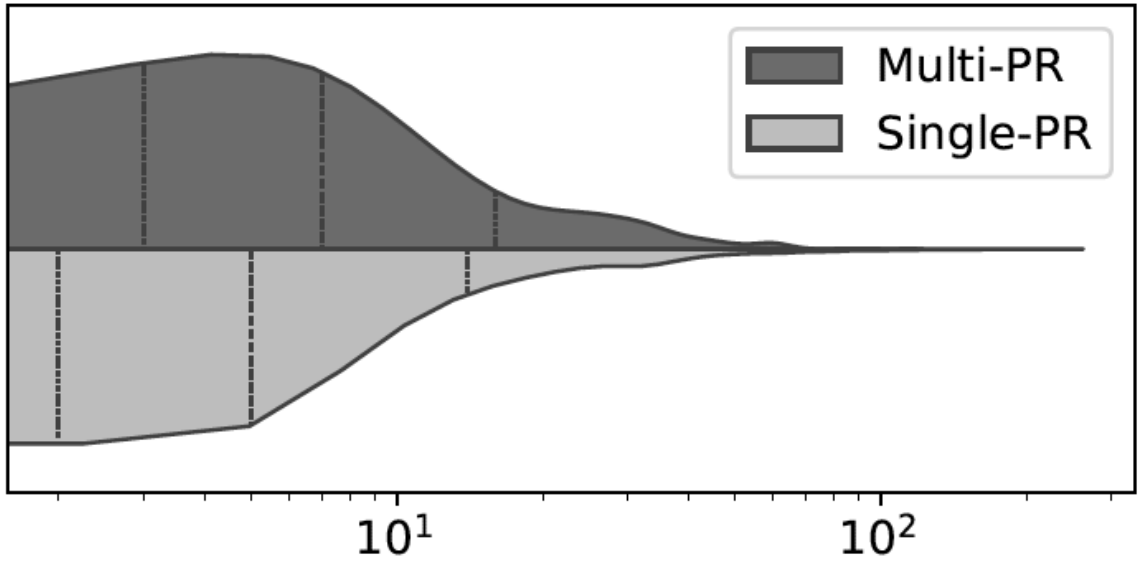
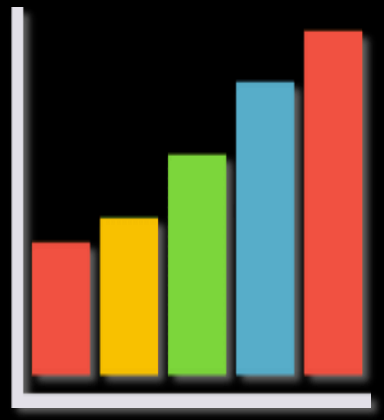


Figure 4: Distribution of the number of commits.

Explore multi-PR issues in video games



Prevalence

One in twenty issues is resolved through multiple PRs; a subset of projects shows a high proportion of multi-PR issues

Sources

The most frequent (48%) source is incomplete or **buggy PRs**; 20% of the multi-PR issues are **extended PRs**

Impact

Multi-PR issues exhibit **significantly longer resolution**; Multi-PR issues generally require slightly more commits

QUEST LAB

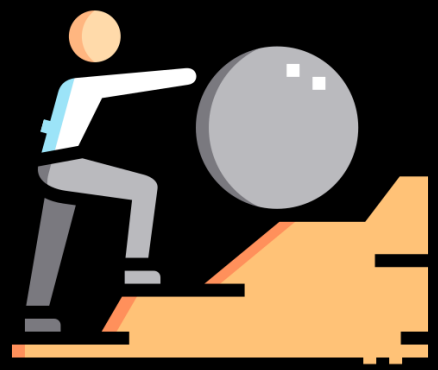
UNIVERSITY OF CALGARY

Implications of our study

Nimmi Weeraddana
nimmi.weeraddana@ucalgary.ca

For empirical researchers: Account for **resolution effort across multiple PRs** to avoid underestimating time, coordination, and complexity.

For tool builders: Model issue resolution as an **iterative multi-PR process**, not a single-shot fix.



For future studies: Apply the taxonomy to other domains to test whether multi-PR patterns are **game-specific or general software-engineering phenomena**.