ON THE IMPACT OF RELEASE POLICIES ON BUG HANDLING ACTIVITY: A CASE STUDY OF ECLIPSE

Zeinab Abou Khalil
Motivation
Selected Case Study

7 Milestones (6~7 weeks interval)

4 Release Candidates (1~2 weeks interval)
RQ0

How do the Eclipse Core project releases evolve over time?
How do the Eclipse Core project releases evolve over time?

Resolution field

- FIXED
- DUPLICATE
- INVALID
- WORKSFORME
- NOT_ECLIPSE
- MOVED
- WONTFIX
- LATER
- REMIND

Resolution field

\[ \text{ResRate}(d_1, d_2) = \frac{\# \text{ of bugs resolved}}{\# \text{ of bugs reported}} \]

\[ \text{FixRate}(d_1, d_2) = \frac{\# \text{ of bugs fixed}}{\# \text{ of bugs resolved}} \]
RQ$_1$

How does the bug handling rate evolve across releases?
How does the bug handling rate evolve across releases?

\[ r_{n-1} \quad r_n \quad r_{n+1} \]

\[ \text{ResRate} \left( \text{before} (r_n) \right) \quad \text{ResRate} \left( \text{after} (r_n) \right) \]
How does the bug handling rate evolve across releases?
RQ$_2$

How does the bug handling time differ before and after each release?
How does the bug handling time differ before and after each release?

![Graph showing bug handling time for releases 4.2 and 4.7](image)
How does the bug handling time differ before and after each release?

**Triaging time**

- For annual releases, bugs tend to get triaged faster before than after the release.
- Beneficial transition from an annual to a quarterly release.
- Bugs are triaged faster after the switch to quarterly releases.
- No measurable effect of bug severity on bug triaging time before and after the release.
How does the bug handling time differ before and after each release?

Fixing time

- No difference between the bug fixing time before and after the release.
- Bug severity does not seem to have a measurable effect.
- Bugs are fixed faster after the transition to the quarterly releases.
- The consulted Eclipse maintainers confirmed our results.
RQ$_3$

How does the feature freeze period impact bug handling rate?
How does the feature freeze period impact bug handling rate?

There is no observable difference in fixing rate between the development period and feature freeze period of each release.
How does the feature freeze period impact bug handling rate?

The feature freeze period focuses more on bugs being reported in that same period than on bugs reported earlier.
How does the feature freeze period impact bug handling rate?

- More effort is spent on fixing bugs during the feature freeze period than during the development period.
- This difference in effort appears to have increased for quarterly releases.
RQ4

How does the feature freeze period impact bug handling time?
How does the feature freeze period impact bug handling time?

- bugs reported for current release 4.5
- bugs reported for next release 4.6
How does the feature freeze period impact bug handling time?
How does the feature freeze period impact bug handling time?

- It takes less time to triage and fix bugs during the feature freeze period of the next release compared to the current release.
- Bugs of the current release that are triaged and fixed during the feature freeze period have stayed open longer compared to those in the development period.
- The feature freeze period does not affect the triaging and fixing time of bugs for the next release.
- The severity of the bugs handled does not influence the results.
Discussion & Conclusion (1/3)

Evolution of the bug handling process of Eclipse:

• Project communities should carefully assess the pros and cons of such release changes upfront.

• Plan ahead these changes to avoid negative impacts.

• Measure if the targeted improvements have been reached after changing the process.

• Researchers need to be aware of such changes in the bug handling process when carrying out empirical.
Discussion & Conclusion (2/3)

Bug handling during feature freeze period:

• Feature freeze periods impose extra stress during rapid releases.

• Feature freeze periods need to be preserved, since they allow to spend more focused effort fixing bugs for the upcoming release.

• They should invest in test automation especially in presence of rapid releases.
Discussion & Conclusion (3/3)

Benefits and challenges in switching to a more rapid release policy:

• Careful preparation and planning is key for a successful transition to faster release cycles.

• It enables the developer community to become more effective in bug handling activities.

• Other projects can benefit from these lessons learned.