MSR 2012

An Empirical Study of Supplementary Bug Fixes

Jihun Park[1], Miryung Kim[2], Baishakhi Ray[2], Doo-Hwan Bae[1]

1. Korea Advanced Institute of Science and Technology
2. The University of Texas at Austin

Motivation

- Human is five times more accurate at locating errors of commission than **errors of omission** [Fry and Weimer '10]

- Several tools recommend **supplementary changes** to r educe omission errors.

- However, there has not been **a comprehensive study of the characteristics of omission errors**.

Study Findings

- **A considerable portion (22%~33%) of bugs** requires supplementary patches.
- Incomplete patches are **larger in size and more scatt ered** than regular patches.
- Predicting a supplementary fix location using **clone a nalysis alone is insufficient**.
- About 15% of supplementary change **locations are b eyond the scope of the direct neighbors** of initial ch ange locations.

Outline

- Research Questions
- Bug Categorization
- Study Results
- Related work
- Discussions

Research Questions

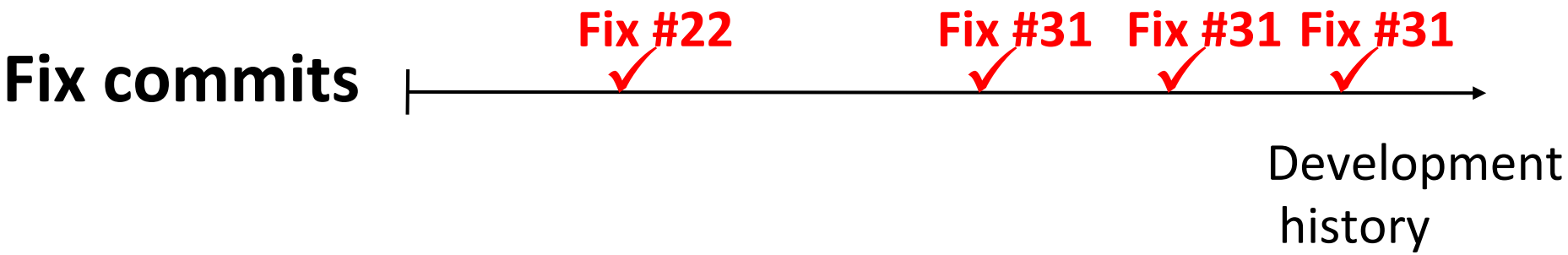
Q1. What is the **extent** and **characteristics** of supplementary changes?

Q2. What are the common **causes of incomplete bug fixes**?

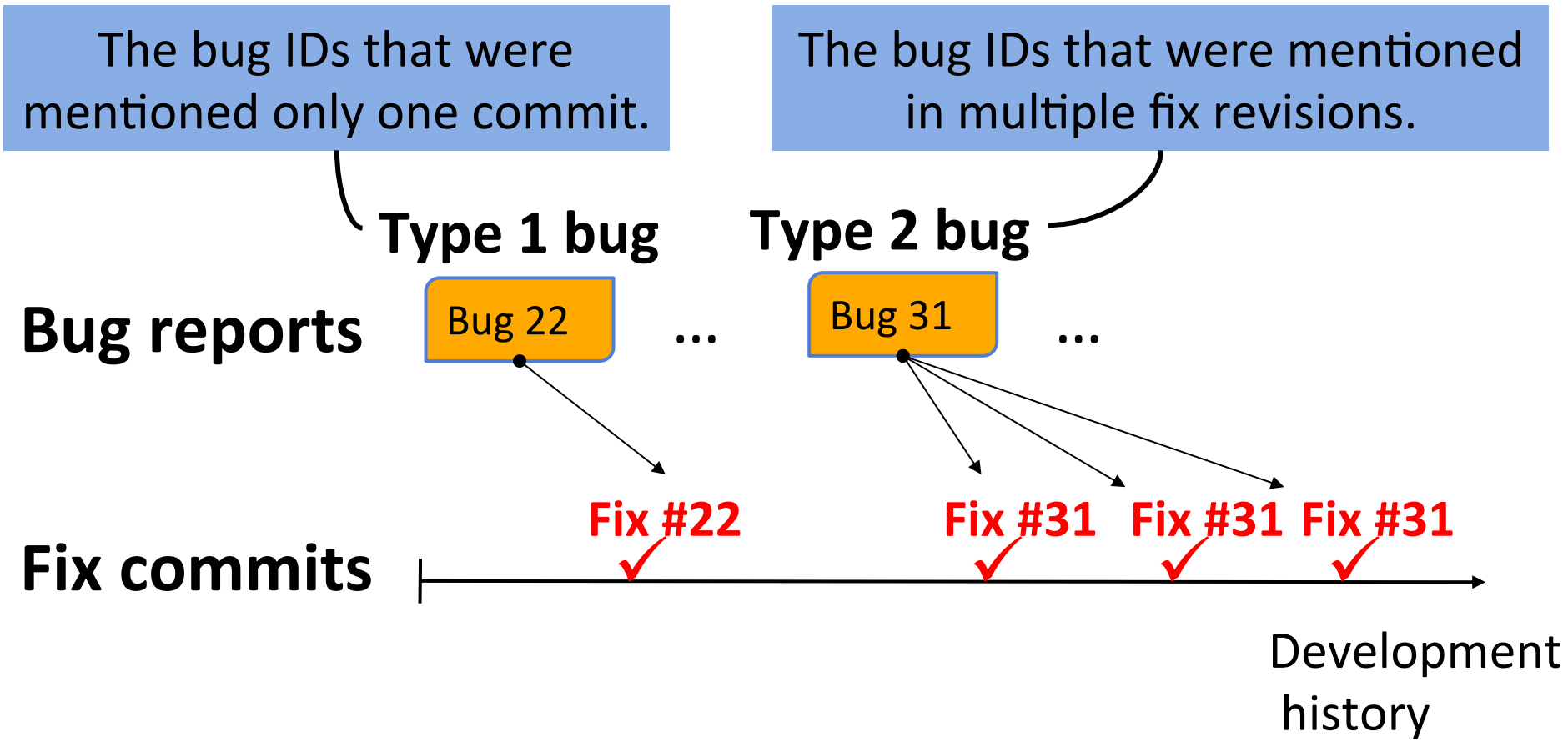
Q3. Are supplementary bug fixes **similar** to corresponding initial fixes?

Q4. Where is **the location of supplementary bug fixes** in relation to initial fixes?

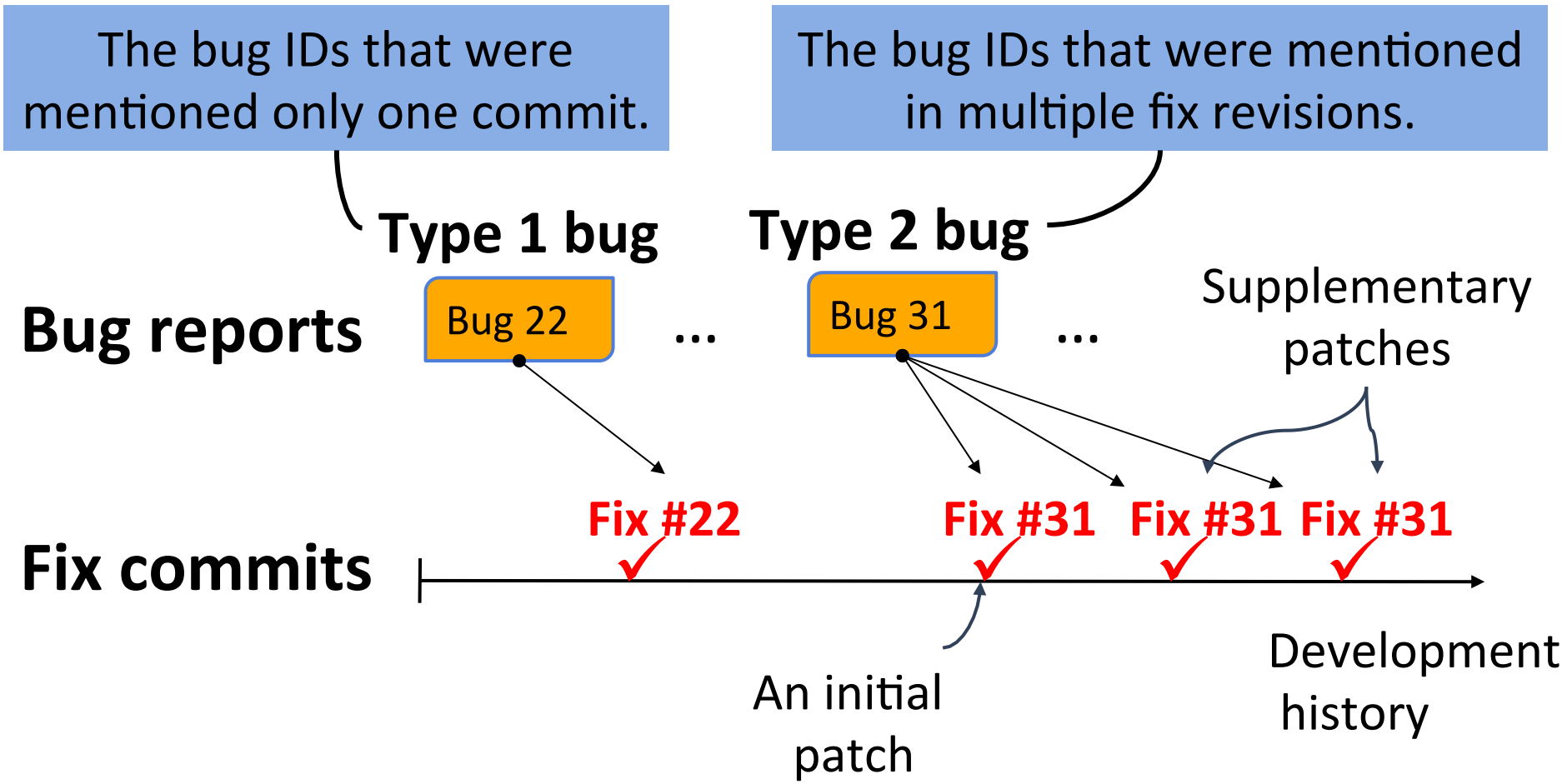
Bug Fix Categorization



Bug Fix Categorization



Bug Fix Categorization



Q1-1. What is the extent of Type 2 bugs?

	Eclipse JDT core	Eclipse SWT	Mozilla project
Study period	2004/07 ~ 2006/07	2004/07 ~ 2006/07	2003/04 ~ 2005/07
Total revisions	17000 revisions	21530 revisions	200000 revisions
# of bugs	1812	1256	11254
Type 1 bugs	1405 (77.54%)	954 (75.96%)	7562 (67.19%)
Type 2 bugs	407 (22.46%)	302(24.04%)	3692 (32.81%)

22% ~ 33% bugs require supplementary bug fixes.

Q1-2. What are the characteristics of bugs that were fixed more than once?

- We examine the **time taken to fix bugs**: the time gap between "REPORTED" to "FIXED" or "CLOSED" statu s.

- We compare **the total number of developers** involve d in Type 1 vs. Type 2 bugs.

Q1-2. What are the characteristics of bugs that were fixed more than once?

The time taken to resolve bugs

	Type 1 bug	Type 2 bug	p-value
Eclipse JDT core	120.79	188.27	3.84e-04
Eclipse SWT	176.99	337.32	2.65e-07
Mozilla	594.50	805.92	8.40e-42

The number of developers involved

	Type 1 bug	Type 2 bug	p-value
Eclipse JDT core	3.67	4.44	1.45e-12
Eclipse SWT	3.13	4.29	1.39e-09
Mozilla	4.70	7.28	2.05e-84

More developers are involved Type 2 bugs, and t hey take longer to be resolved.

Q2. What are the characteristics of incomplete patches?

- We compare patches of Type 1 bugs and initial patches of Type 2 bugs in terms of **number of files**, **patch size**, **entropy, etc.**

	Files		LOC		Added LOC		Entropy (file)		Entropy (package)	
	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2
Total	3.30	5.72	147.98	309.38	60.92	63.42	0.31	0.36	0.15	0.18
p-value	**1.15E-18**		**4.46E-05**		**4.03E-12**		**2.05E-09**		**1.04E-10**	

Initial patches of Type 2 bugs are larger in size, and more scattered than patches of Type 1 bugs.

Q2. What are the causes of incomplete bug fixes?

- **Manual investigation** on 100 supplementary patches

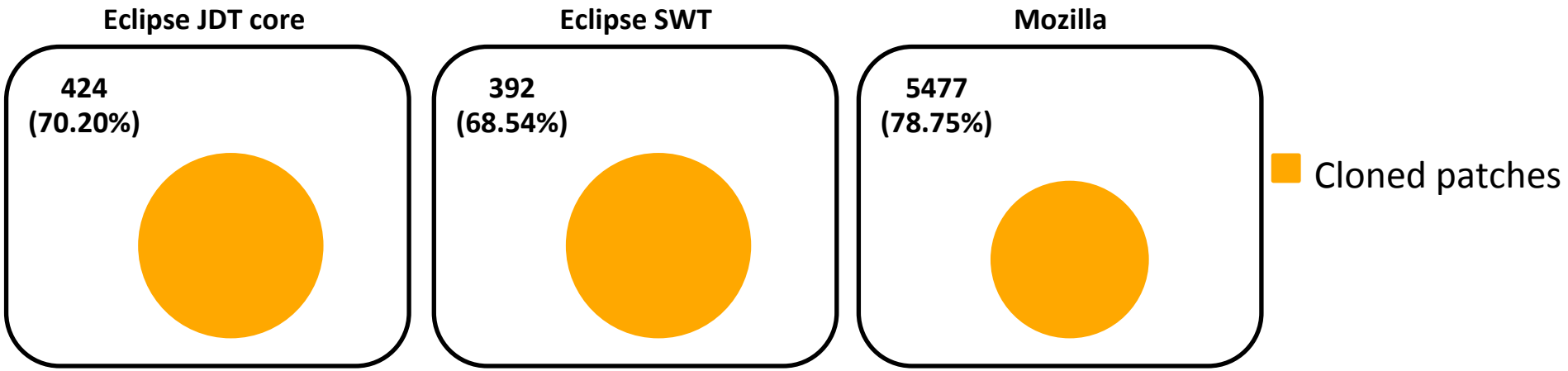
	Types	Frequency
1	**An initial patch is ported** to a different component or branch.	28%
2	The **conditional statement** of an initial fix is not correct.	23%
3	Code elements **referring to or being referenced** by changed code a re later updated.	15%
4	Two different parts **calling different subclasses of the same type** ar e not updated together.	4%
5	**Incomplete refactoring** induces a supplementary patch	3%

The common causes of incomplete fixes include missed por ting updates, incorrect conditional statements, and incomp lete refactoring.

* more types are on the paper

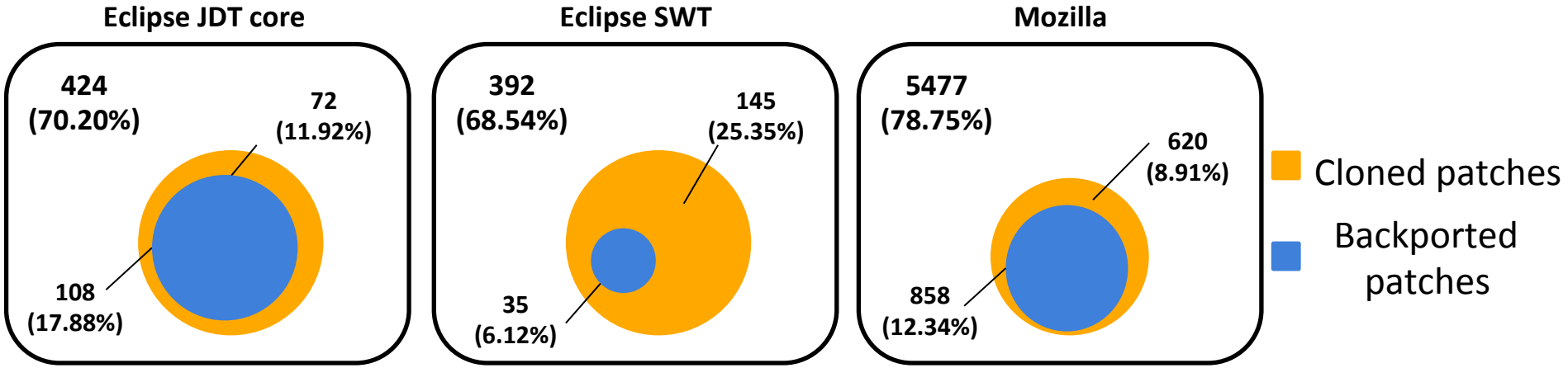
Q3. Are supplementary bug fixes similar to corresponding initial fixes?

- Study method: We identify similar patches using **clone analysis tool** (CCFinder, Kamiya et al. 2002)



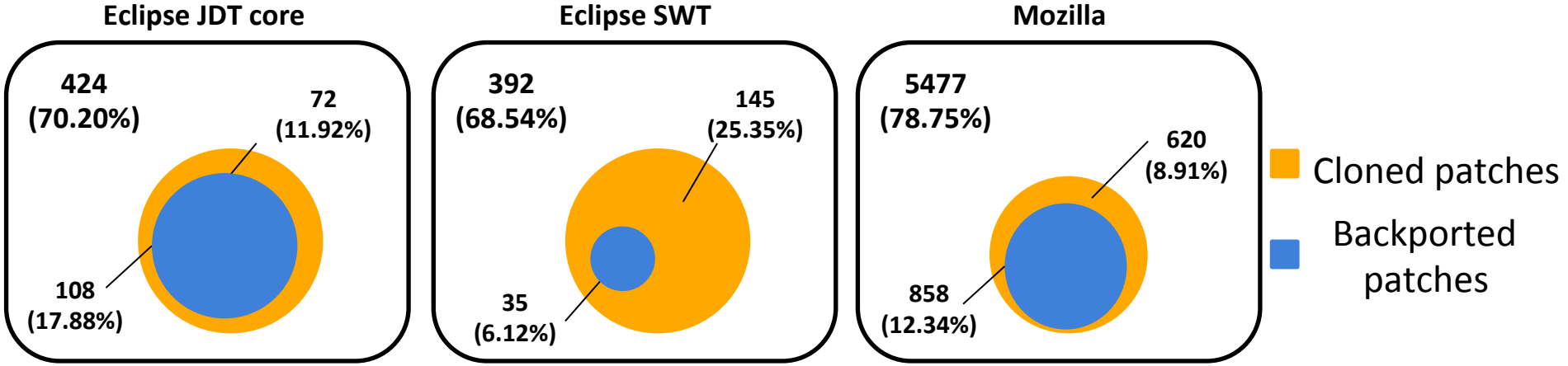
Q3. Are supplementary bug fixes similar to corresponding initial fixes?

- Study method: We identify similar patches using **clone analysis tool** (CCFinder, Kamiya et al. 2002)
- We exclude **backported patches**, because they are simply identical patches applied to different branch locations.



Q3. Are supplementary bug fixes similar to corresponding initial fixes?

- Study method: We identify similar patches using **clone analysis tool** (CCFinder, Kamiya et al. 2002)
- We exclude **backported patches**, because they are simply identical patches applied to different branch locations.
- Only **12%, 25%, and 9%** include at least five similar lines.



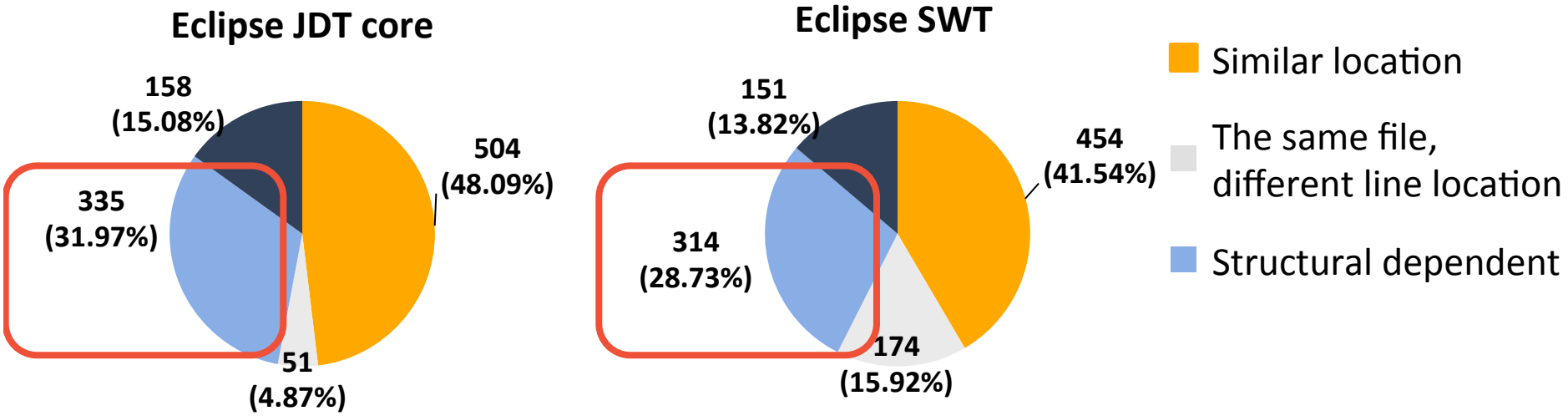
Predicting a supplementary fix location using code clone analysis alone is insufficient.

Q4. Where is the location of supplementary bug fixes in relation to initial fixes?

- 48% and 42% are made at the similar line location of an initial patch. (similar heuristics with Yin et al. '11)

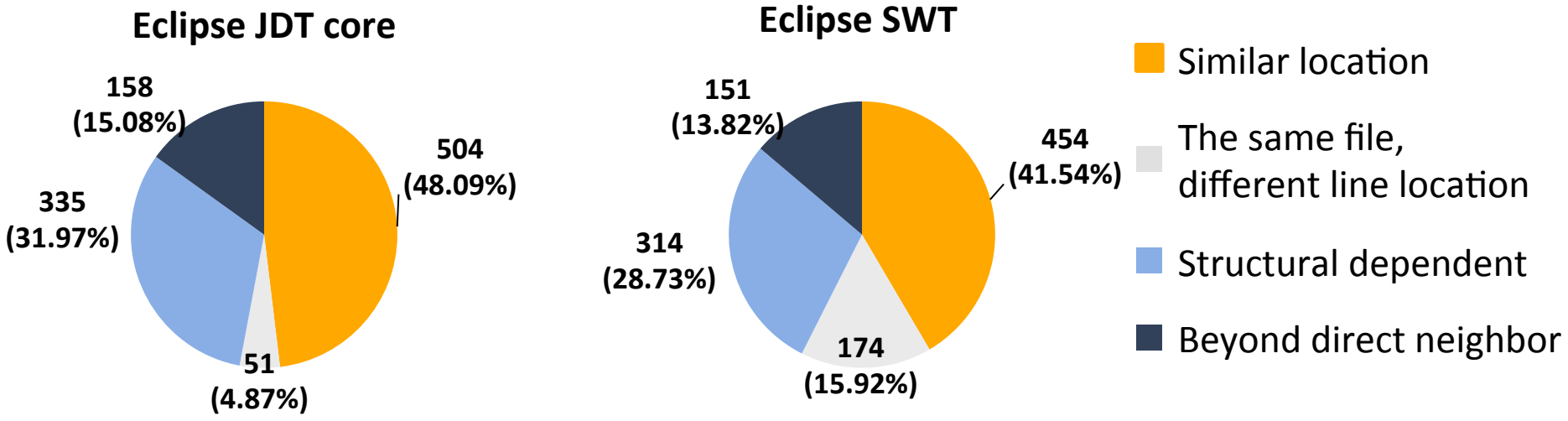
Q4. Where is the location of supplementary bug fixes in relation to initial fixes?

- 48% and 42% are made at the similar line location of an initial patch. (similar heuristics with Yin et al. '11)
- 32% and 29% have **structural dependences** with an in itial patch. (using LSdiff by Kim and Notkin '09)



Q4. Where is the location of supplementary bug fixes in relation to initial fixes?

- 48% and 42% are made at the similar line location of an initial patch. (similar heuristics with Yin et al. '11)
- 32% and 29% have **structural dependences** with an initial patch. (using LSdiff by Kim and Notkin '09)



About 15% of supplementary change locations are beyond the scope of the direct neighbors of initial patch locations.

Related Work

- 17% to 45% of fixes are recurring and they can be ide ntified using similar code units. (Nguyen et al.)

- 14.8% to 24.4% of post release patches are incorrect. (Yin et al.)
- 9% of all bugs are re-opened. (Gu et al.)

Limitations and Future Work

- Expand the studied period of bug reports.
- Investigate the relationship among supplementary fix es.
- Develop new tools for reducing incomplete bug fixes. (e.g., detection of incomplete refactoring)

Summary

- A **considerable portion of bugs** requires supplement ary patches.
- Incomplete patches are **larger in size and more scatt ered** than regular patches.
- Predicting a supplementary fix location using **code cl one analysis** alone **is insufficient.**
- About 15% of supplementary change locations are **beyond the scope of the direct neighbors** of initial p atch locations.

Acknowledgements

- This research was sponsored by the Agency for Defe nse Development, Republic of Korea, under the grant UD100031CD.
- This work was in part supported by National Science Foundation under the grants CCF-1149391, CCF-1043 810, and CCF-1117902, and by a Microsoft SEIF awar d.

Jihun Park
Email: jhpark@se.kaist.ac.kr
PhD Candidate
Software Engineering LAB
Computer Science Dept.
KAIST, Daejeon, South Korea