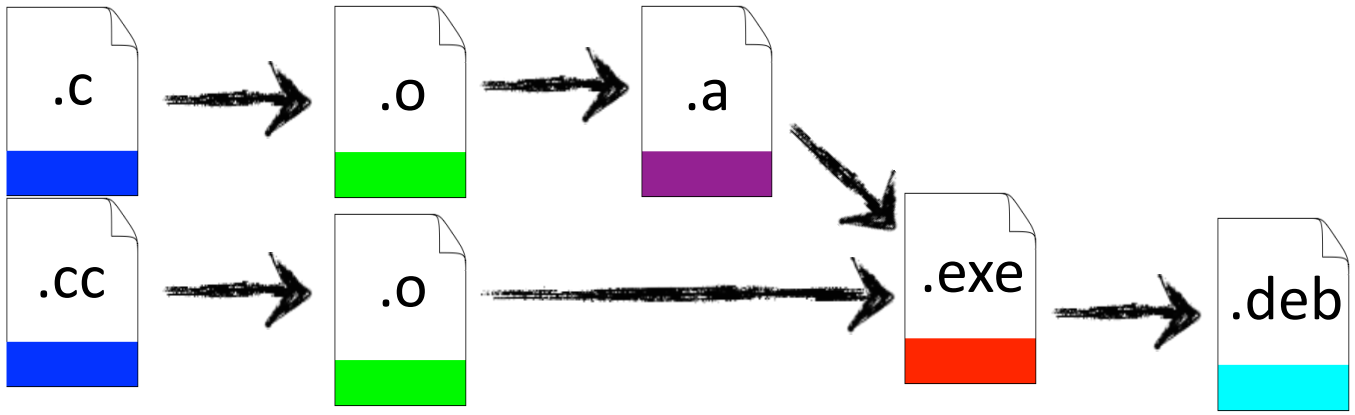
Tackling Build Failures in Continuous Integration

Foyzul Hassan
University of Texas at San Antonio

Ph.D. Advisor: Dr. Xiaoyin Wang

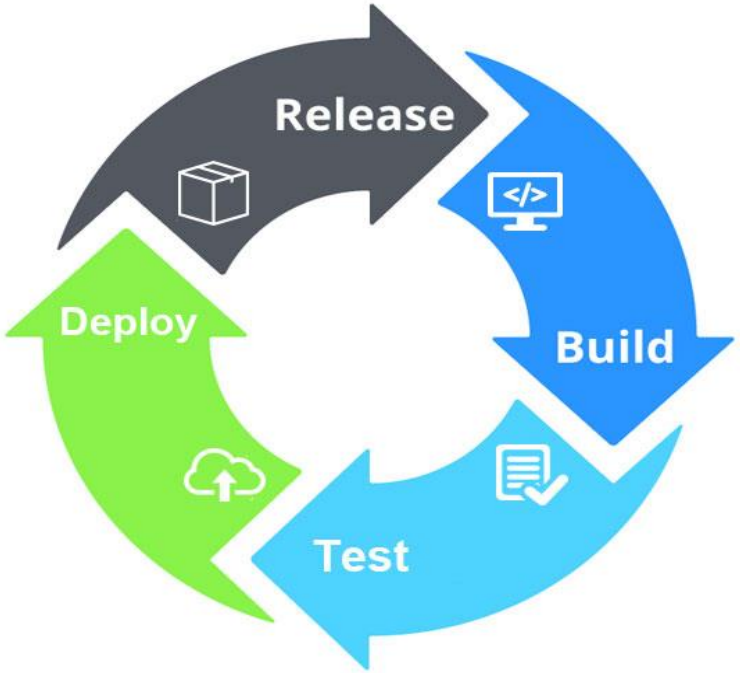
ASE 2019 Doctoral Symposium – Nov 11th 2019
San Diego, California, United States

Build Systems and Continuous Integration



Build systems describe how sources are translated into deliverables

maven



Continuous Integration(CI) is a widely used development practice for faster code integration and deployment

The build system is at the heart of processes in Continuous Integration (CI)

Build Failure and Performance for Project Progress

- According to TravisTorrent data-set, **29%** of code commits fail to go through successful build on the integration server.

- Study of Java Build system *[McIntosh et al. 2012]* shows **27%** of source code work items require an accompanying change to the build system.

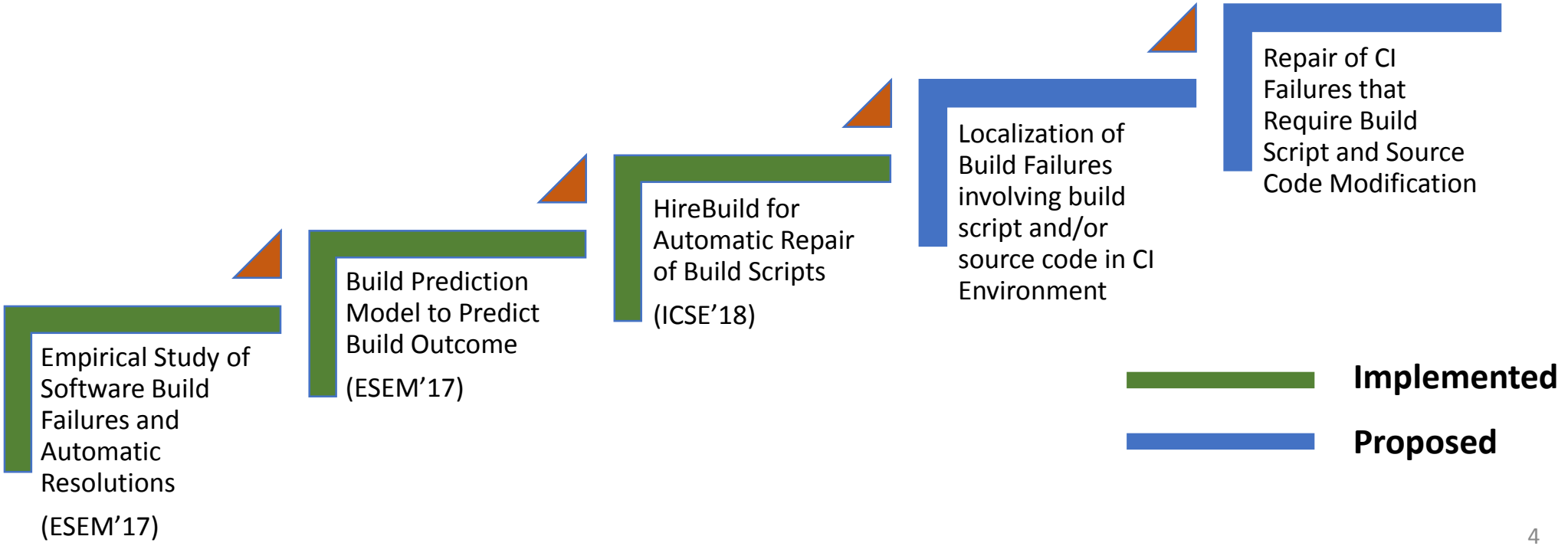
Build Failure and Waiting Time Hampers Project Progress

- Study on TravisTorrent data-set shows that 20% of code commit waits at build queue due to concurrent nature of code commit.

- *At Google,* **37.4% and 29.7%** of C++ and Java *builds fail.[*Seo et al. 2014*]*

Research hypothesis

Localize, Repair and Predict CI Failures Can Significantly Improve Development Progress and Production

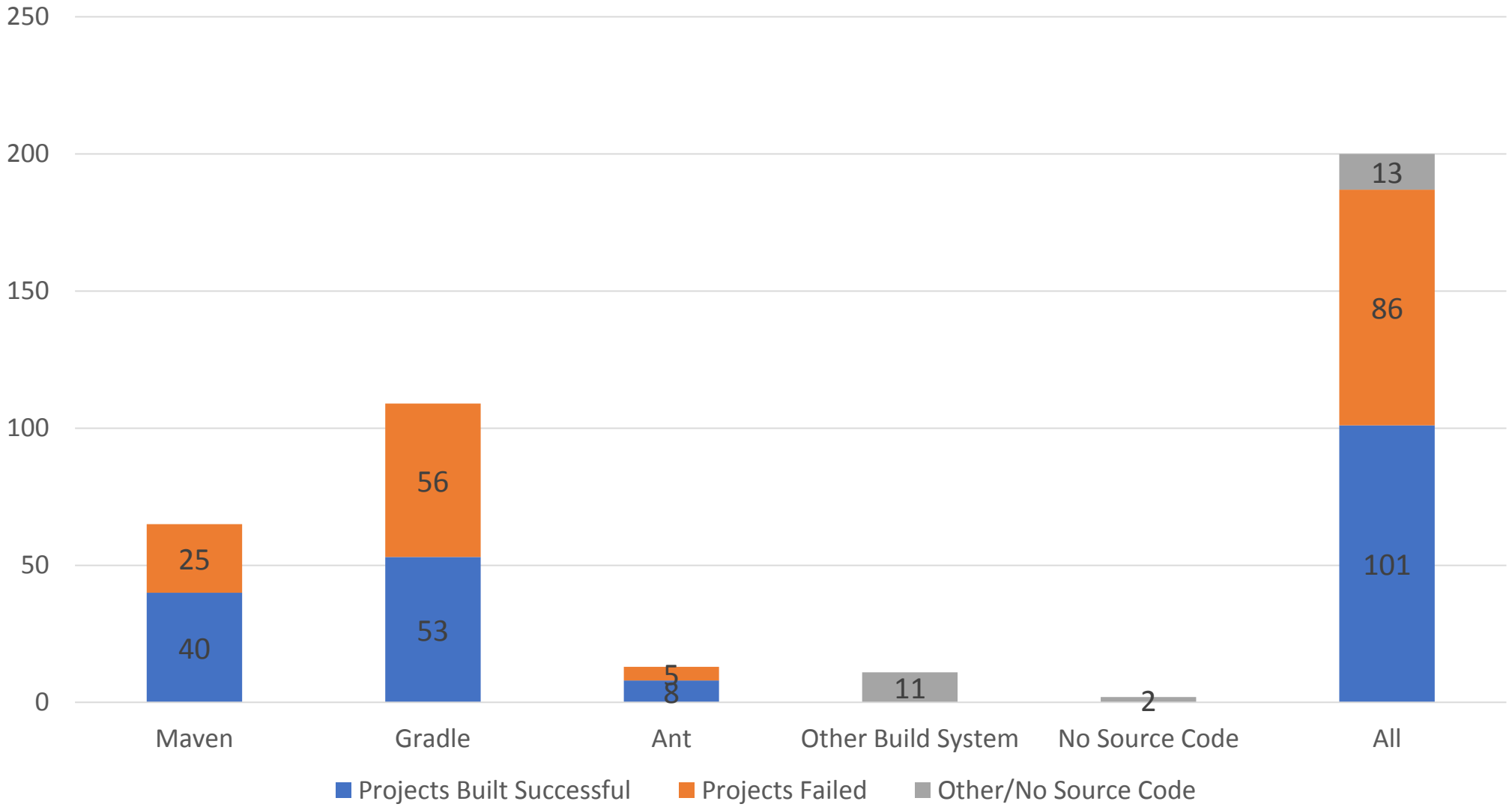


An Automatic Building of Java Projects in Software Repositories: A Study on Feasibility and Challenges

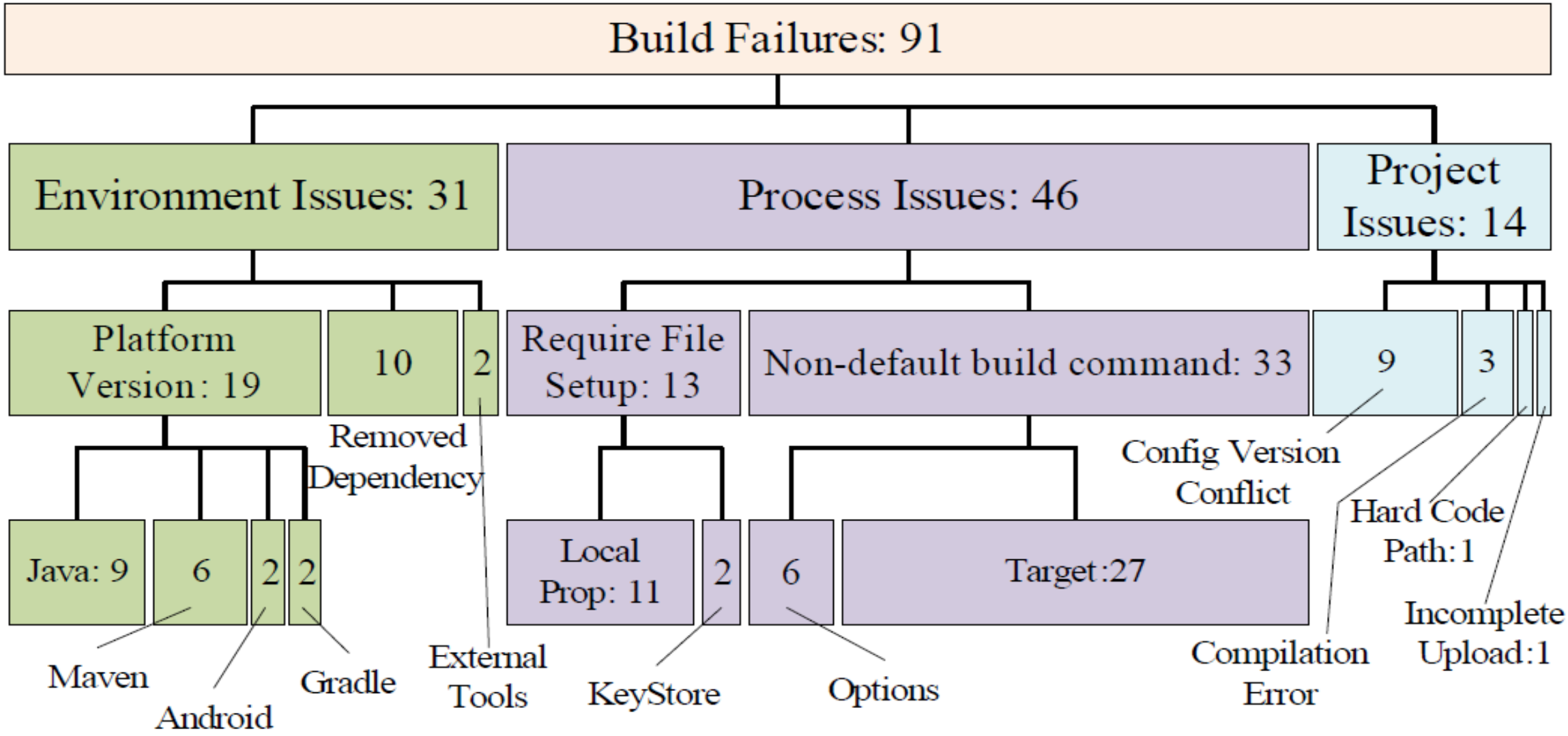
Foyzul Hassan, Shaikh Mostafa, Edmund SL Lam, Xiaoyin Wang

2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)

Build Status With Default Build Command

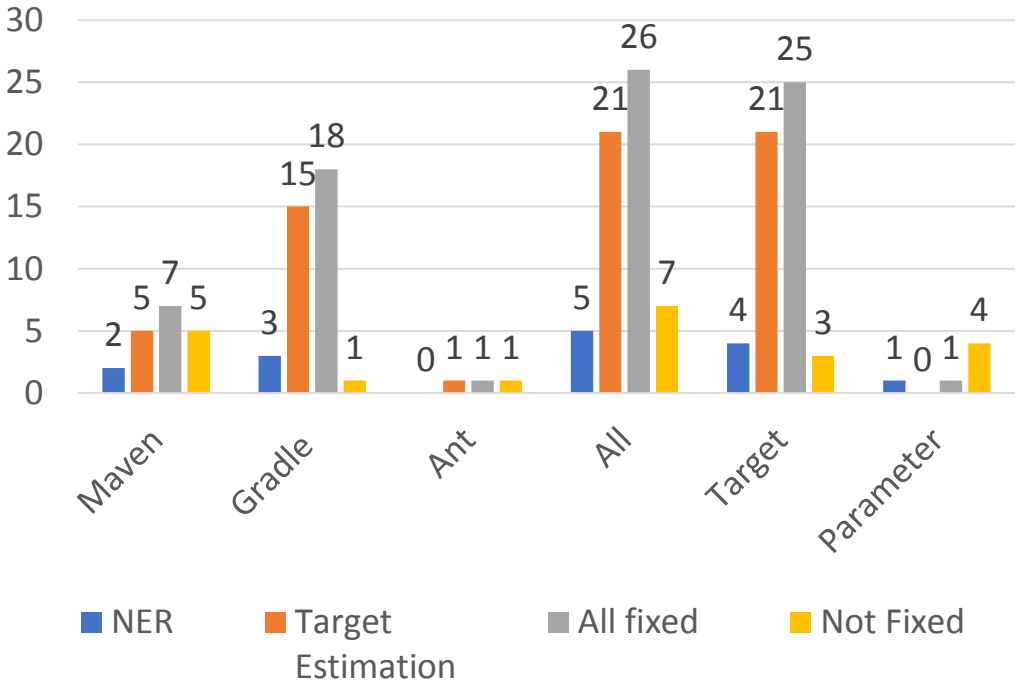


Build Failure Hierarchy

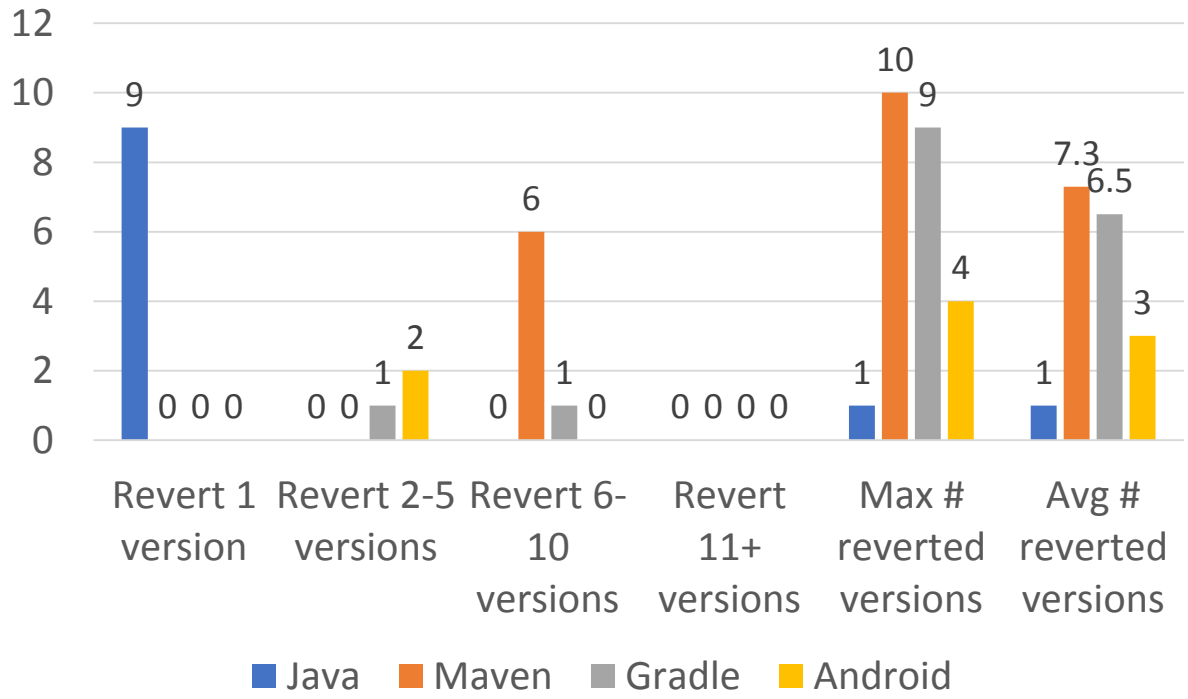


Automatic Resolution of Build Failures

Build Command Extraction and Prediction



Version Reverting



Template File Generation

We find that, simply generating an empty local file or rename sample file will resolve **7 of the 13** require file setup build failures.

Takeaways from the study

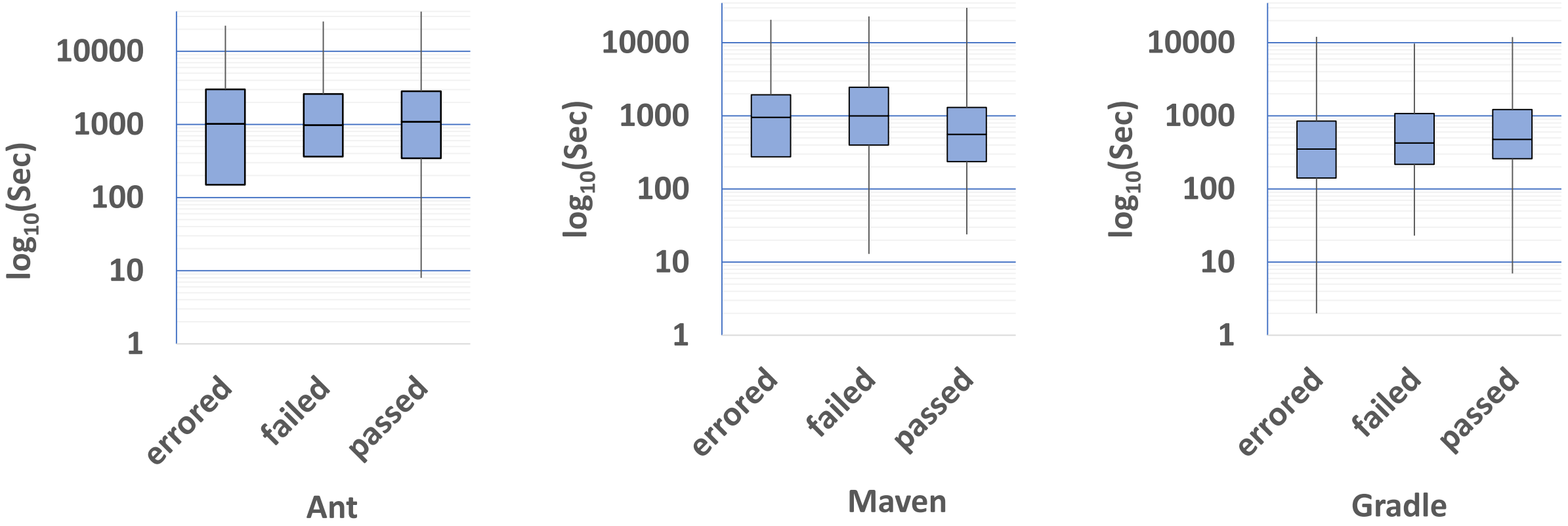
- It is a necessity.
  - **Half of the top Java projects** cannot be straightforwardly built with default build commands.
- It is feasible.
  - Among the **86 projects** with build failures, **52 projects** can be built successfully with different approach such as build command extraction and estimation, version reverting etc.
- The challenges.
  - Our study has also identified several build failure categories whose automatic resolution can be difficult.

Change-aware build prediction model for stall avoidance in continuous integration

Foyzul Hassan and Xiaoyin Wang

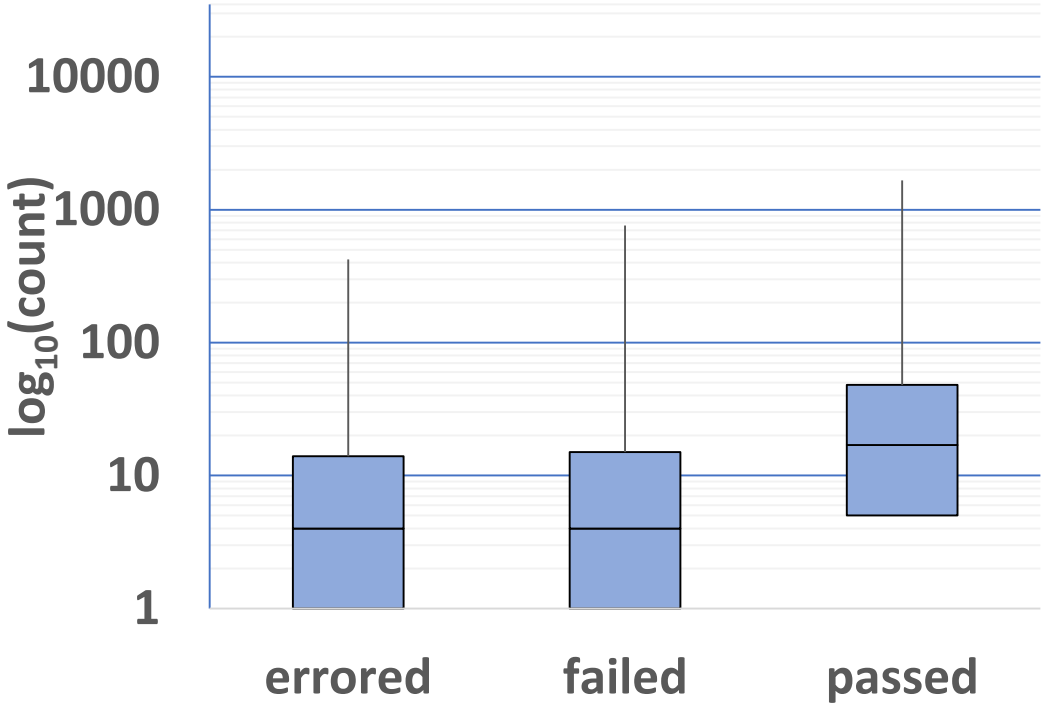
2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)

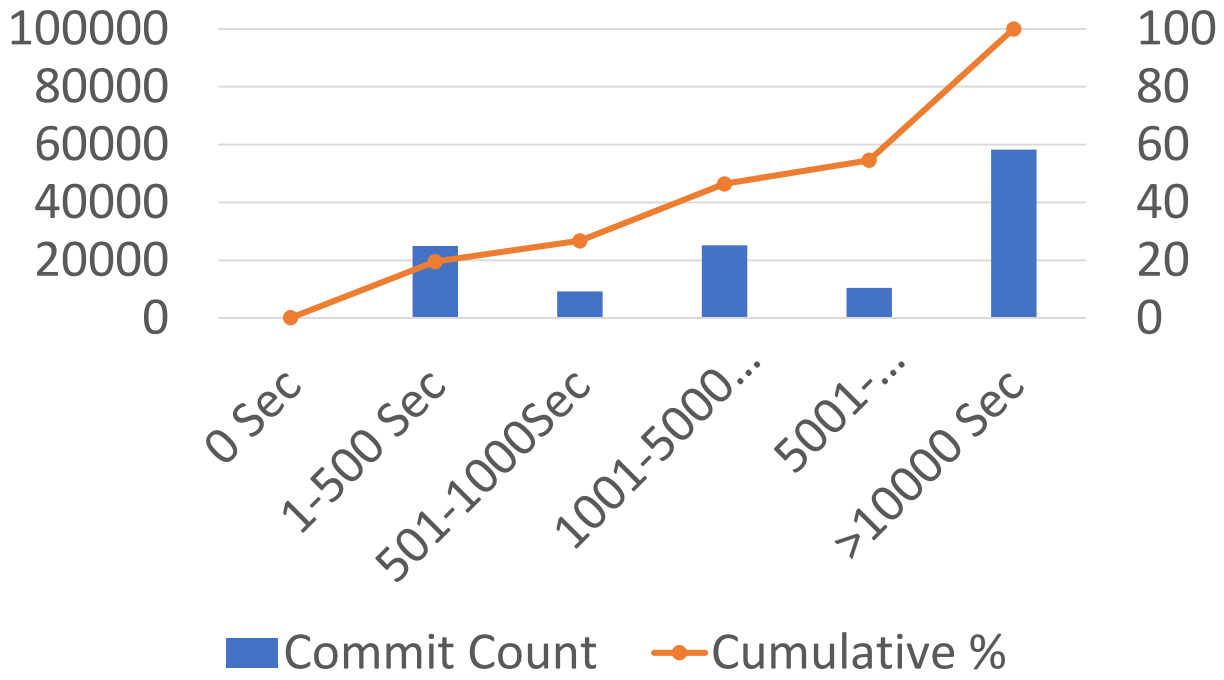
Build Execution Time



Avg. Min Execution Time in Sec: errored(0.666666667), failed(12), passed(13)
Avg. Median Execution Time in Sec: errored(775.33), failed(801.66), passed(708.16)
Avg. Maximum Execution Time in Sec: errored(18331), failed(19365.33), passed(25552.33)

Build Status Change and Commit Frequency

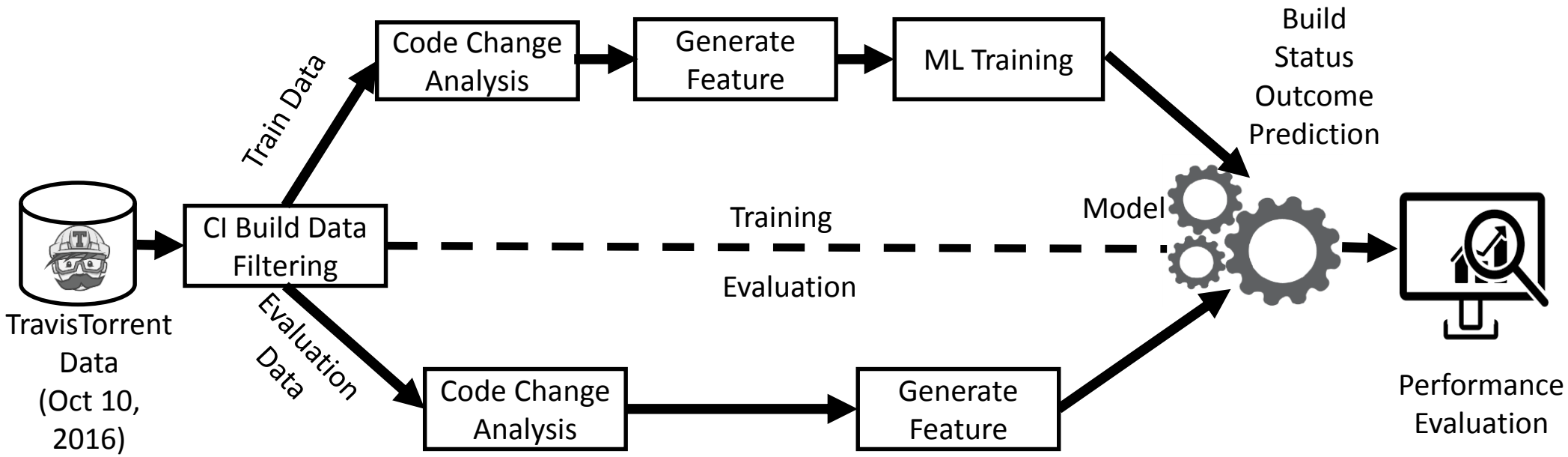




Number of Projects: 402
Number of Build Instance: 256055
Min Same Build Status: errored(0), failed(0), passed(0)
Median Same Build Status : errored(4), failed(4), passed(17)
Maximum Same Build Status : errored(422), failed(760), passed(1660)

Number of Projects: 402
Number of Commits Analyzed:236055

Build Prediction Model



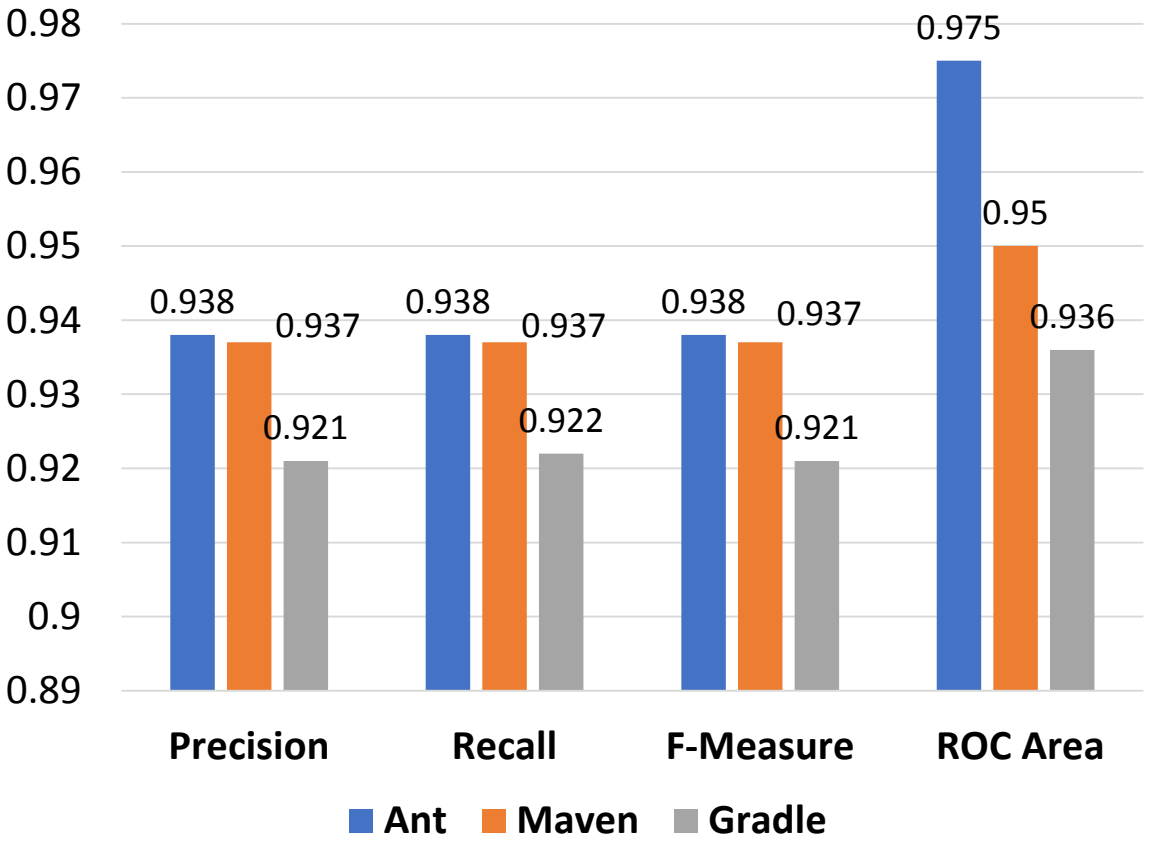
Random Forest Classifier for ML.

ML model used 20 features; 12 features already available in TravisTorrent dataset. Rest of the 8 features generated by our approach.

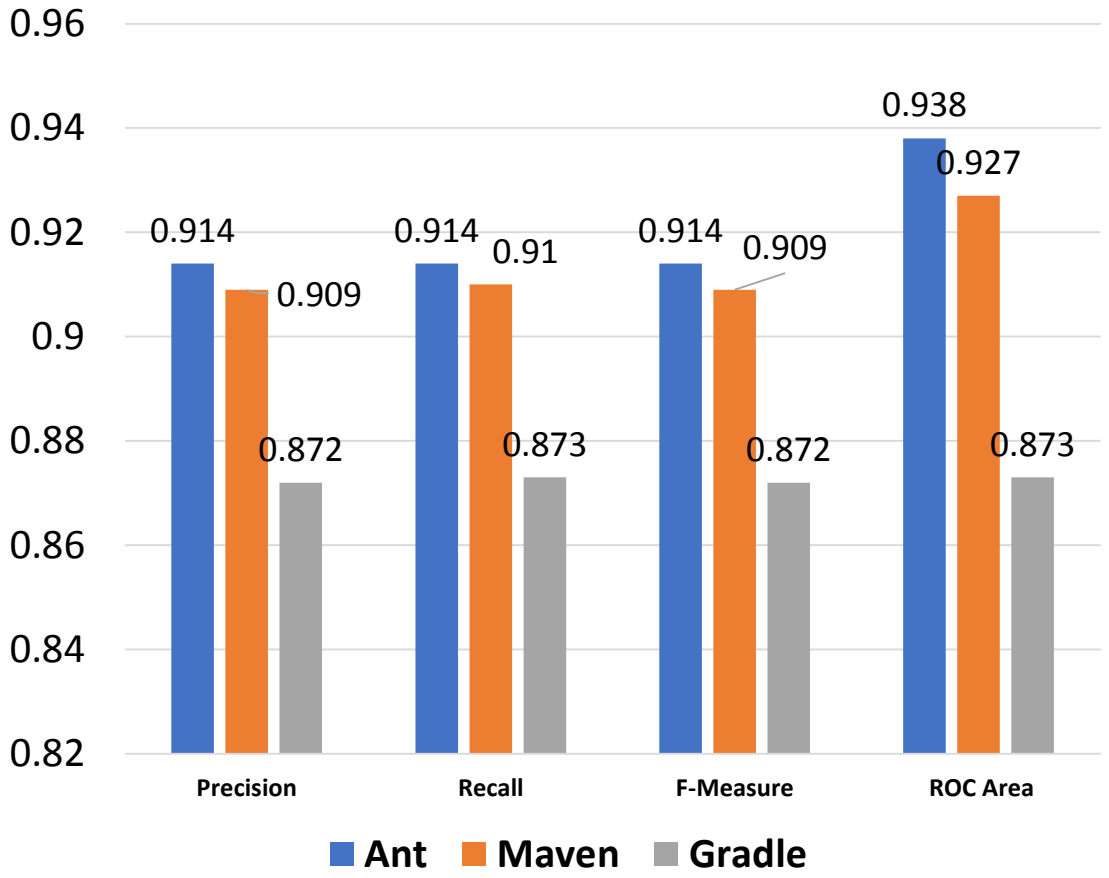
Generated Feature Names and Description

Feature Name	Description of Feature
prev_bl_cluster	Previous Build Cluster ID
cmt_buildfilechangecount	Number of build script file change
cmt_methodbodychangecount	Number of method body change count
cmt_methodchangecount	Number of method signature change
cmt_importchangecount	Number of import statement changes
cmt_fieldchangecount	Number of class attribute change
day_of_week	Day of week of the first commit for the build
cmt_classchangecount	Number of class changed

Performance Evaluation



Performed 8-Fold Cross Validation for performance evaluation.



For each build system 80 percent projects are used for Training and rest of the 20 percent projects are used for testing.

Conclusion

- Our evaluation shows that our approach predicts build outcome with over 87 percent F-Measure for all build systems in CI environment.

- Cross-Project evaluation indicates that our approach can also work for new projects having no or little build history.

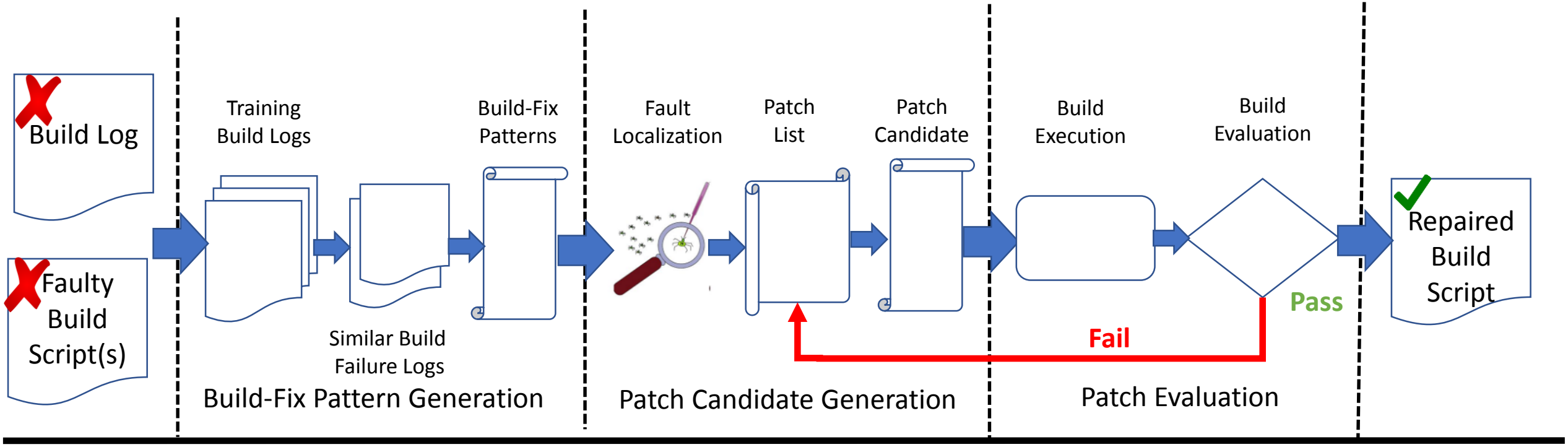
HireBuild: An Automatic Approach to History Driven Repair of Build Scripts

Foyzul Hassan and Xiaoyin Wang

40th International Conference on Software Engineering (ICSE'18)

HireBuild – In Action

- Automatic Build-Fix Pattern Generation
- Patch Candidate Generation
- Patch Evaluation

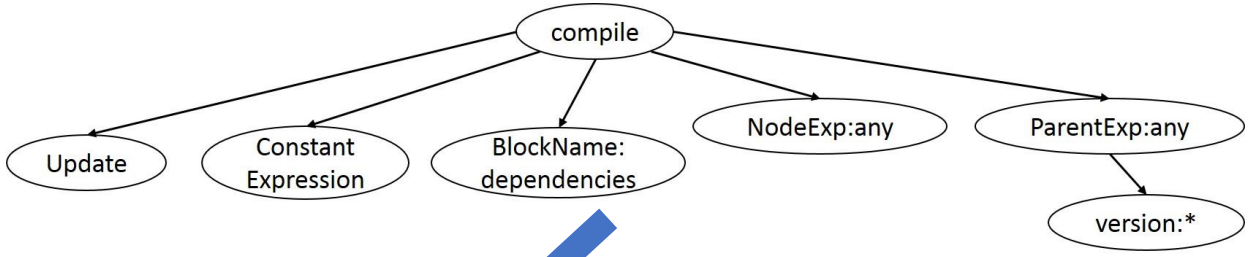


An Example of Fixing With Template

Build Log Project: aol/micro-server, ID: 9599be6

```
* What went wrong:
 Could not resolve all dependencies for configuration ':micro-boot:compile'.
 > Could not find com.aol.microservices:microserver-core:0.57
                ...
```

Fix Template



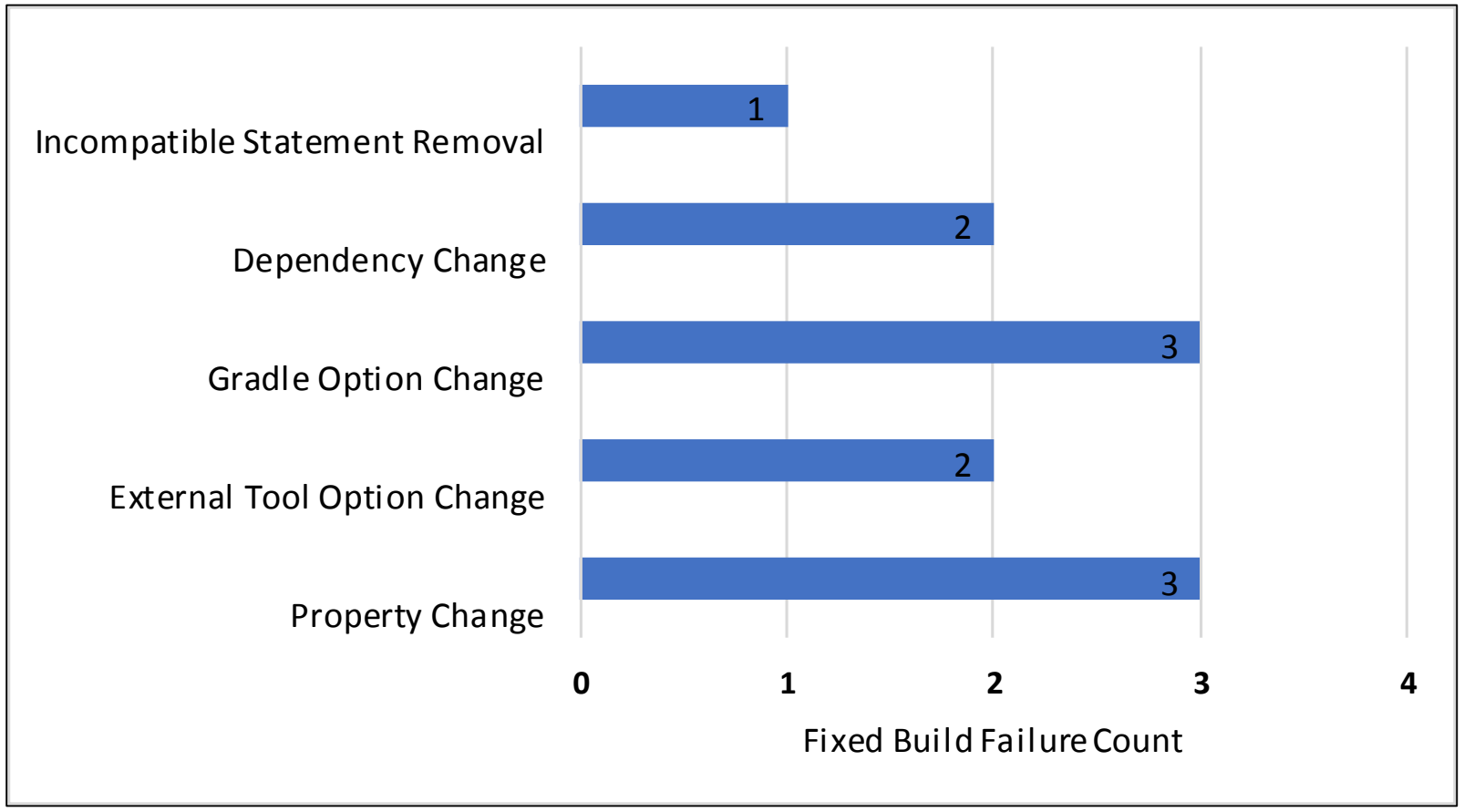
Generated Fix

```
dependencies {
        compile 'org.springframework.boot:spring-boot:1.2.1.RELEASE'
        compile 'org.springframework.boot:spring-boot-autoconfigure:1.2.1.RELEASE'
        //compile project(':micro-core')
         compile group: 'com.aol.microservices', name:'microserver-core', version:'0.56'
}
```

Successfully fixed build failures

11 of 24 Reproducible Build Failures Can Be Fixed Successfully
45.83%

Manually Checked All The Automatically Generated Fixes Are Semantically **Equivalent** to Manual Fixes



What about unsuccessful cases?

Fix Type	#of Failures
Project specific change adaption	2(15%)
No matching patterns	6(46%)
Dependency resolution failures	3(23%)
Multi-location fixes	2(15%)

Project specific change *(Netflix/Hystrix:6600947)*

```
+ if ( dep.moduleName == 'servlet-api' ) {
+ it.artifactId[0].text() == dep.moduleName
&&
+ asNode().dependencies[0].dependency.find{
+ ...
+ }}
+ }
```

Dependency resolve Issue
(BuildCraft/BuildCraft:12f4f06)

```
- mappings = 'snapshot_20160214'
+ mappings = 'stable_22'
```

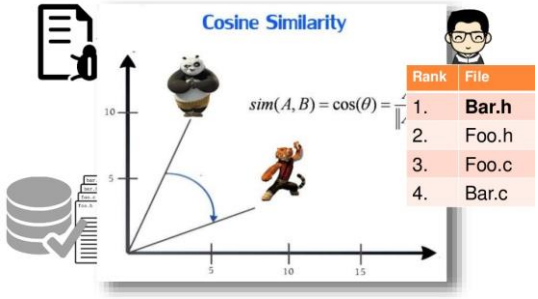
Conclusion

- We propose the first approach for automatic build fix candidate patch generation for Gradle build script.

- Our technique is featured with following challenges:
  - Build log analysis.
  - Build-fix-pattern templates.
  - Build validation.

Proposed Work(1)
Localization of Build Failures Involving Source Code and/or Build Script in Continuous Integration

IR Based Fault Localization



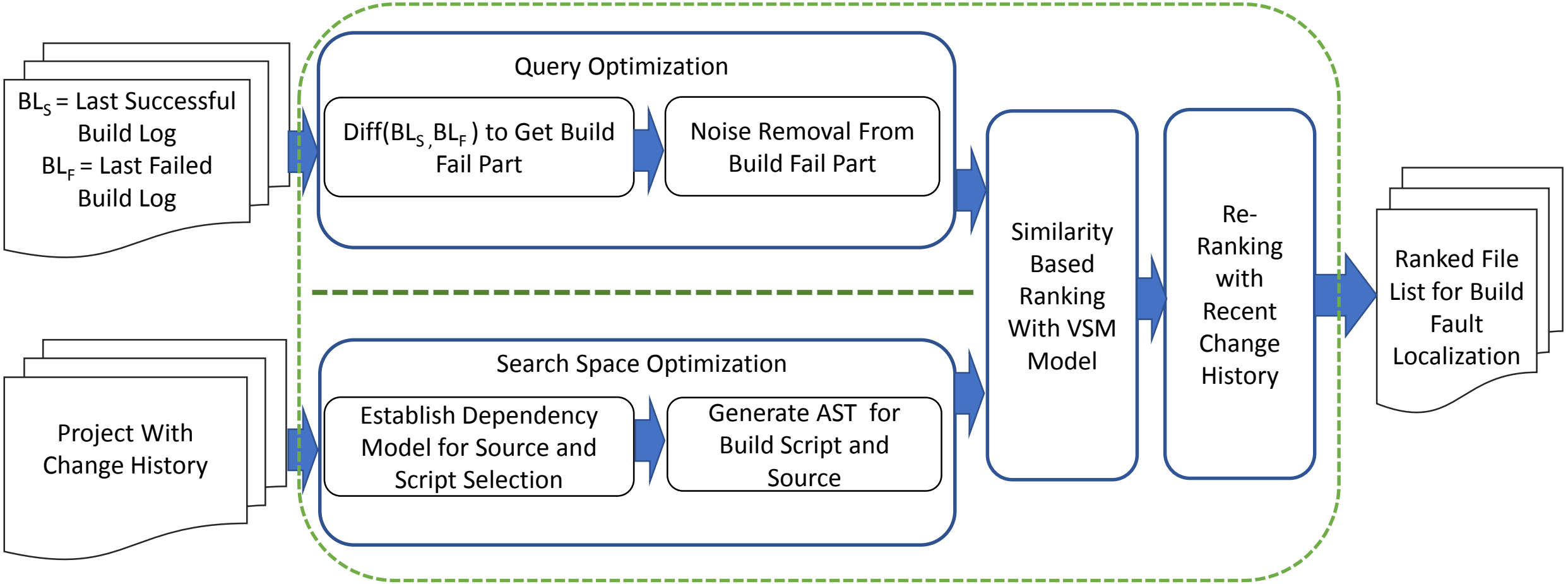
- How do they work?
  - Rank program entities based on lexical similarity of bug reports

Build Failure Log and Source Code Similarity
(tilal6991/HoloIRC: Build Failure Version:fe82cc1, Build Fix Version:4d3bf5d)

```
...
~/ServerFragment.java:37: error:
ServerFragment is not abstract and does
not override abstract method
getConversation() in IRCFragment
public class ServerFragment extends
IRCFragment;ServerEvent; {
^
...
Note: Some input files use unchecked or unsafe
operations.
FAILED
...
```

```
Source code file: IRCFragment.java
Commit ID: fe82cc1
abstract class IRCFragment<T extends Event>
extends ListFragment implements TextView
.OnEditorActionListener {
...
public abstract Conversation
getConversation();
...
}
```

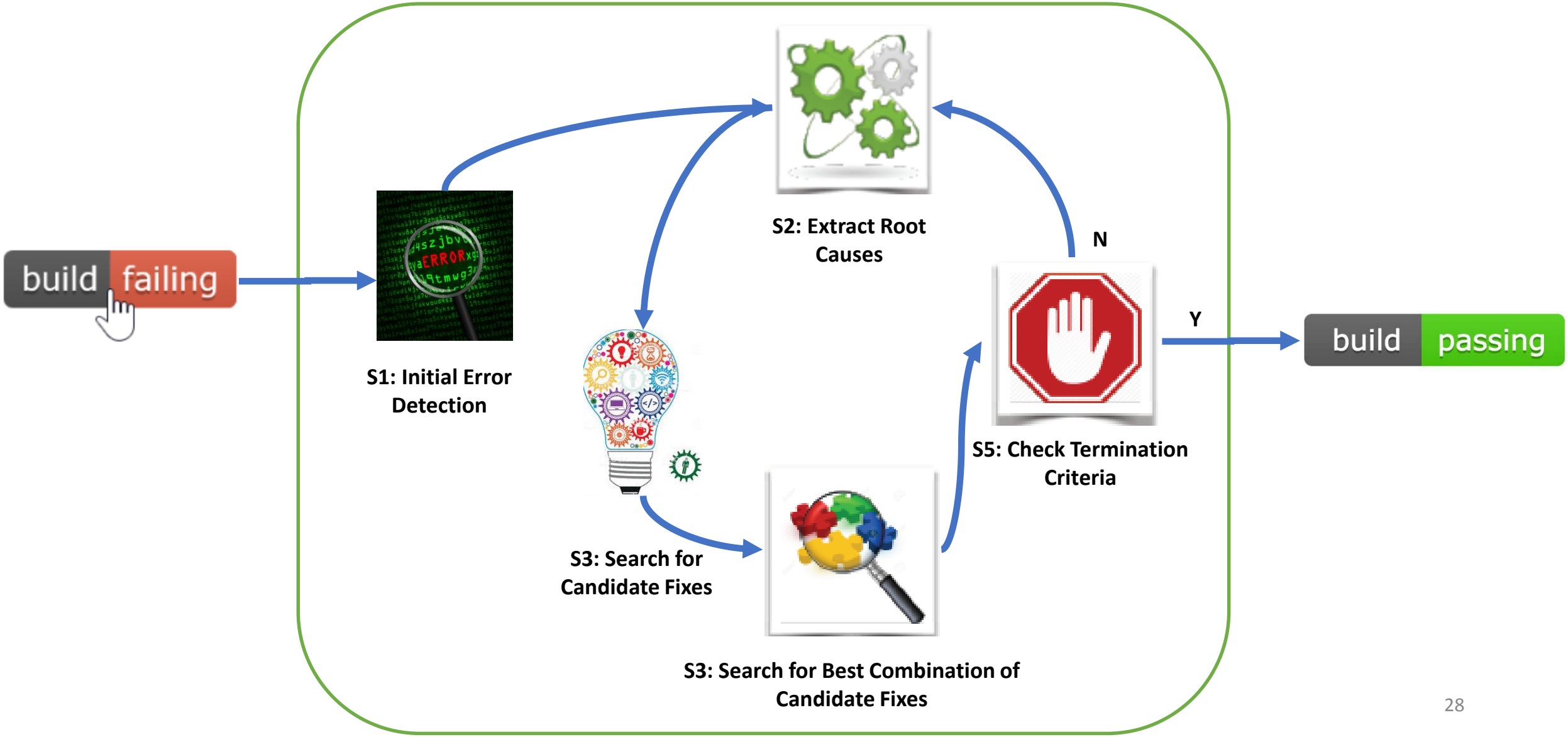
Overview of Proposed Unified Fault Localizer



Proposed Work(2)

Tool Suite to Fix Both Build Script and Source File Build Failures

Overview of Proposed Repair Tool for Build Script & Source Code

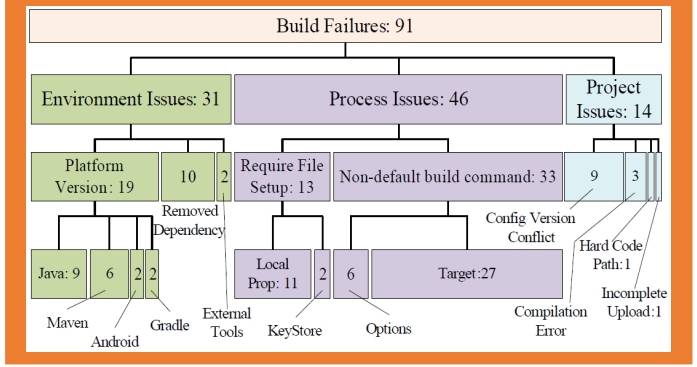


Plan

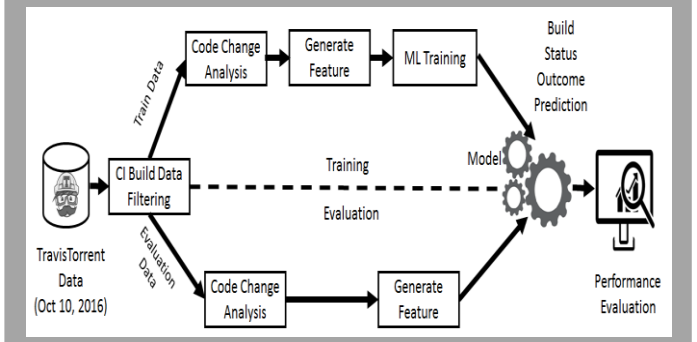
Semesters	Tasks
Fall 2019	**Attend ASE 2019 Doctoral Symposium**
	Evaluate Fault Localization work on Source and Build Script
	Submit Fault Localization work to top conferences
Spring 2020	Implement and Evaluate Repair Tool for Source and Build Script
	Submit Repair work to top conferences
Summer 2020	Write the dissertation
	Graduate

Summary

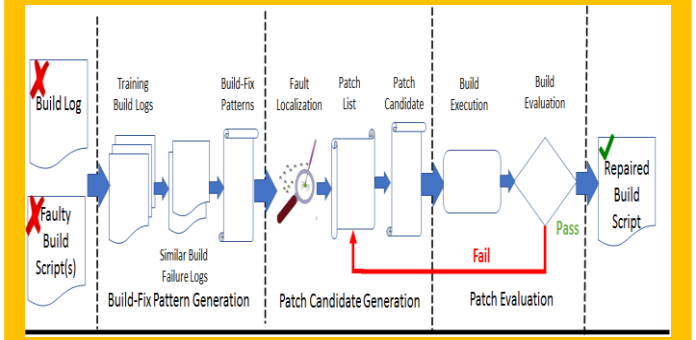
ESEM 17: Build Failure Characterization



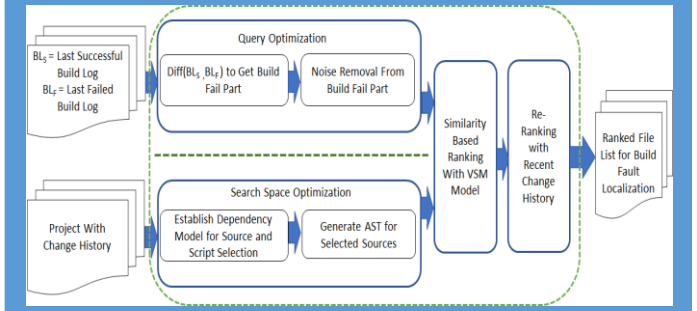
ESEM 17: Build Prediction



ICSE 18: Build Repair (HireBuild)



Ongoing: Fault Localization of CI Failures



Planned: Unified Repair Tool for CI Failures

Hire me!

Contact

foyzul.Hassan@my.utsa.edu

https://foyzulhassan.github.io/