

Test Effort Estimation

Software test effort estimation is the process of predicting the amount of time, resources, and personnel needed to complete a software testing project.

IMPORTANCE OF TEST EFFORT ESTIMATION

- Project Schedule and Timeline
- Project Cost
- Project Quality
- Project Team



PROBLEM STATEMENT

To design and implement a pipeline of data extraction that could be used to predict objective software metrics to measure Test Effort
Estimation using Machine Learning approaches (ML-TEE).

SOURCES OF DATA

- Architectural Smells
- Abstraction Smells
- Encapsulation Smells
- Modularisation Smells
- Hierarchy Smells
- Compute Metrics

Designite

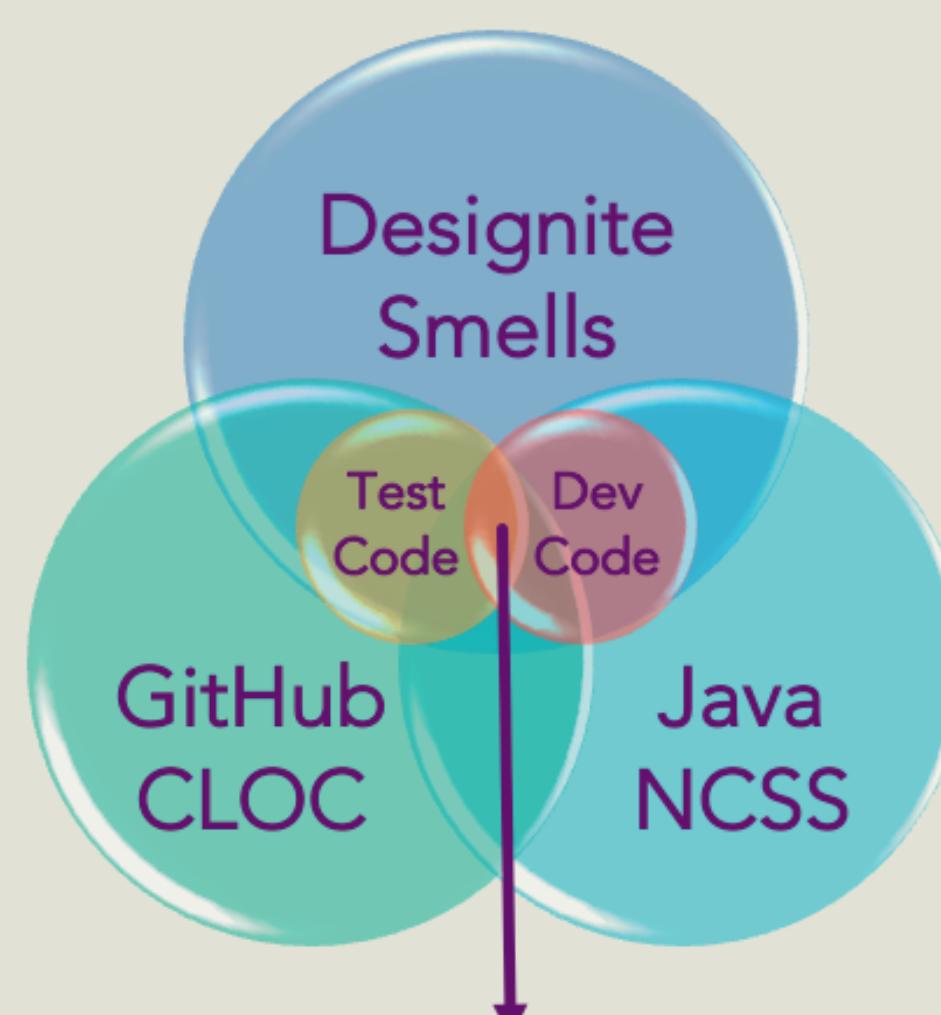
- Cyclomatic Complexity
- Non Commenting Source Statements
- Javadoc comments
- Number of Packages, classes, Functions

Java NCSS

- Changes in Lines of Code at each Commit
- Changes in Lines of Code in each Pull Request
- Changes in Lines of Code in each Release

Github API

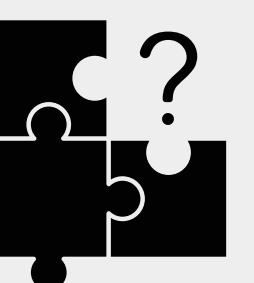
MERGING OF DATA



- Final CSV is formed by:
- Inner Join on data from the 3 Data Sources (based on common unique identifiers)
 - Outer Join on Commits with Test and Dev File changes

CHALLENGES

- Lack of industrial generalised data to work on
- Unavailability of existing pipeline to fetch data
- Lack of unique metrics to measure TEE (Test Effort Estimation)



INDUSTRIAL USE CASE EXAMPLE

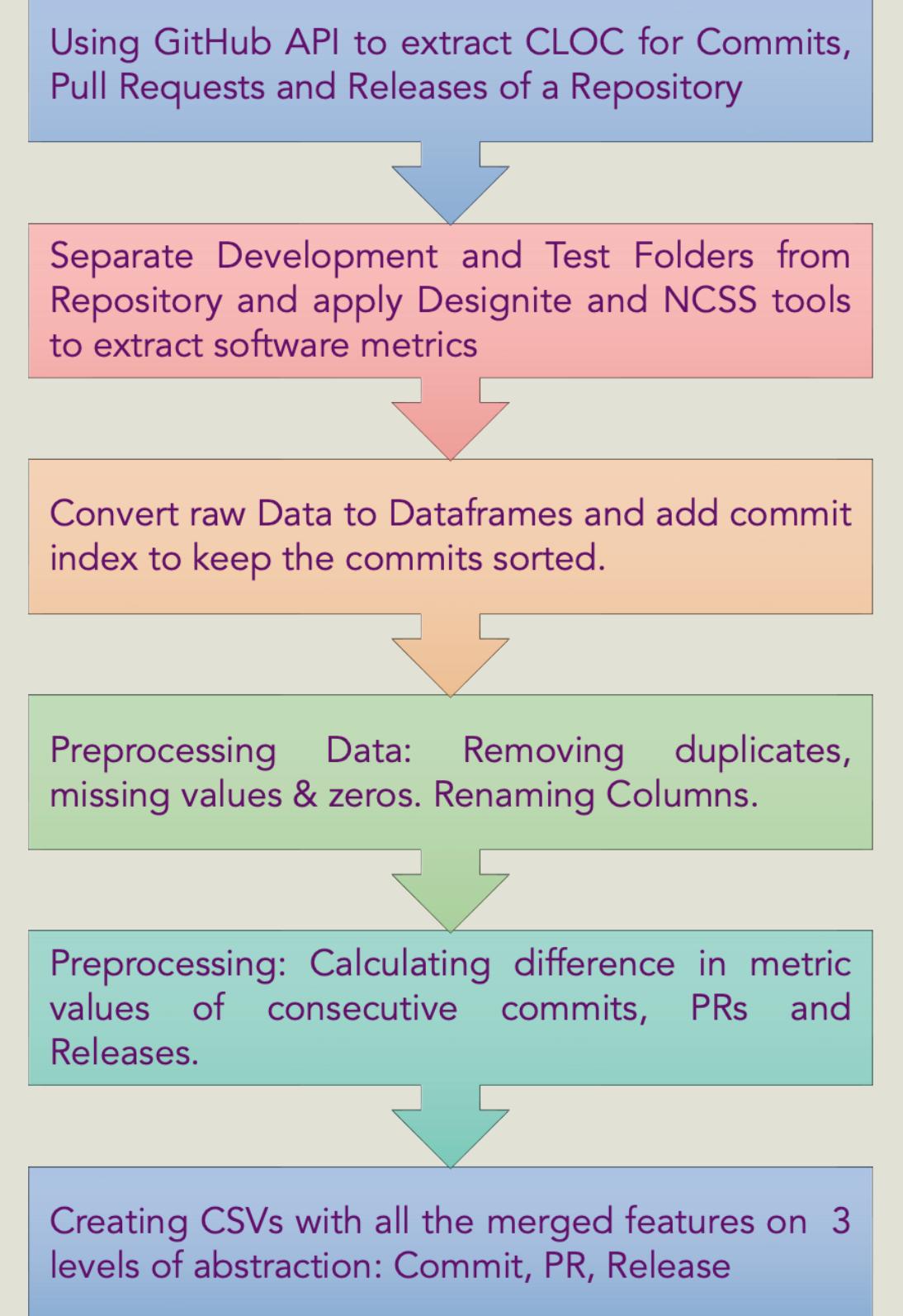


RIP

1 in every 3 project, fails due to insufficient Testing

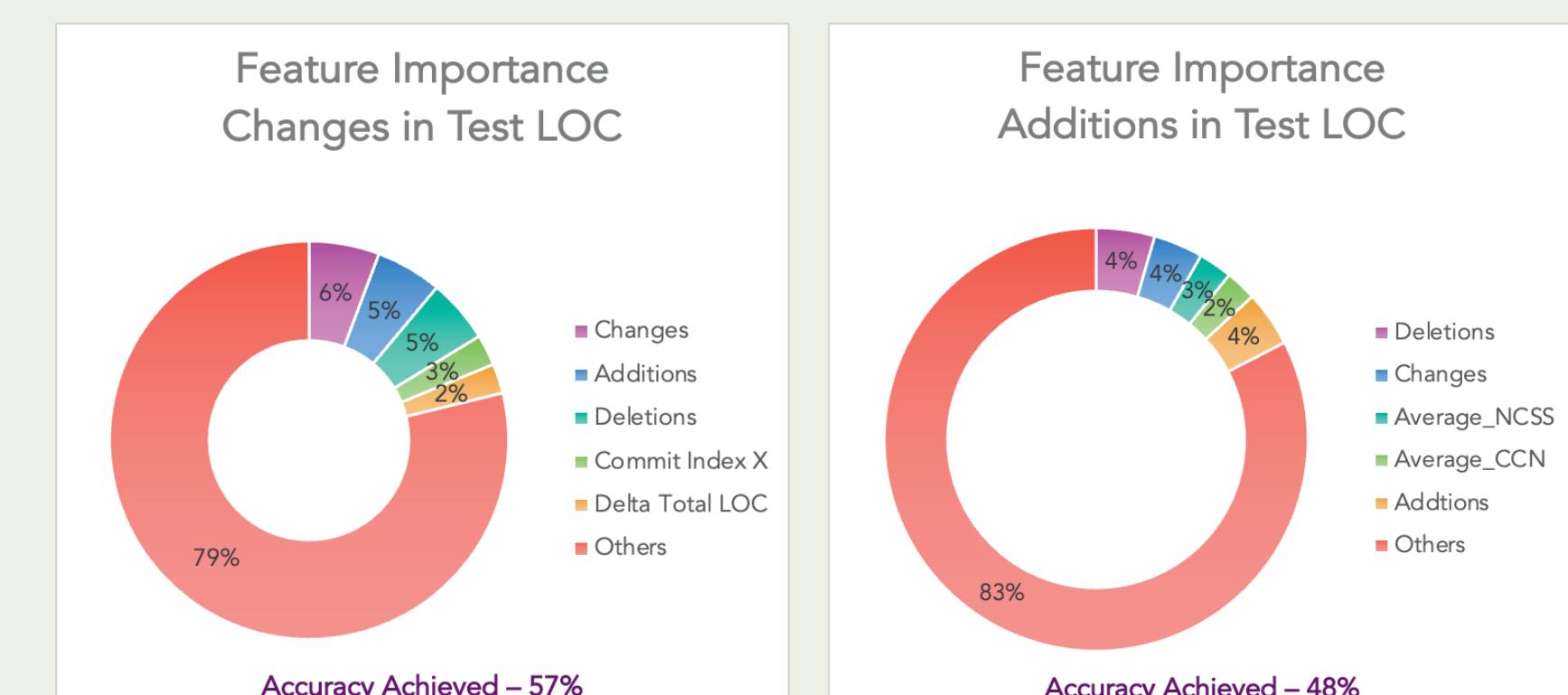
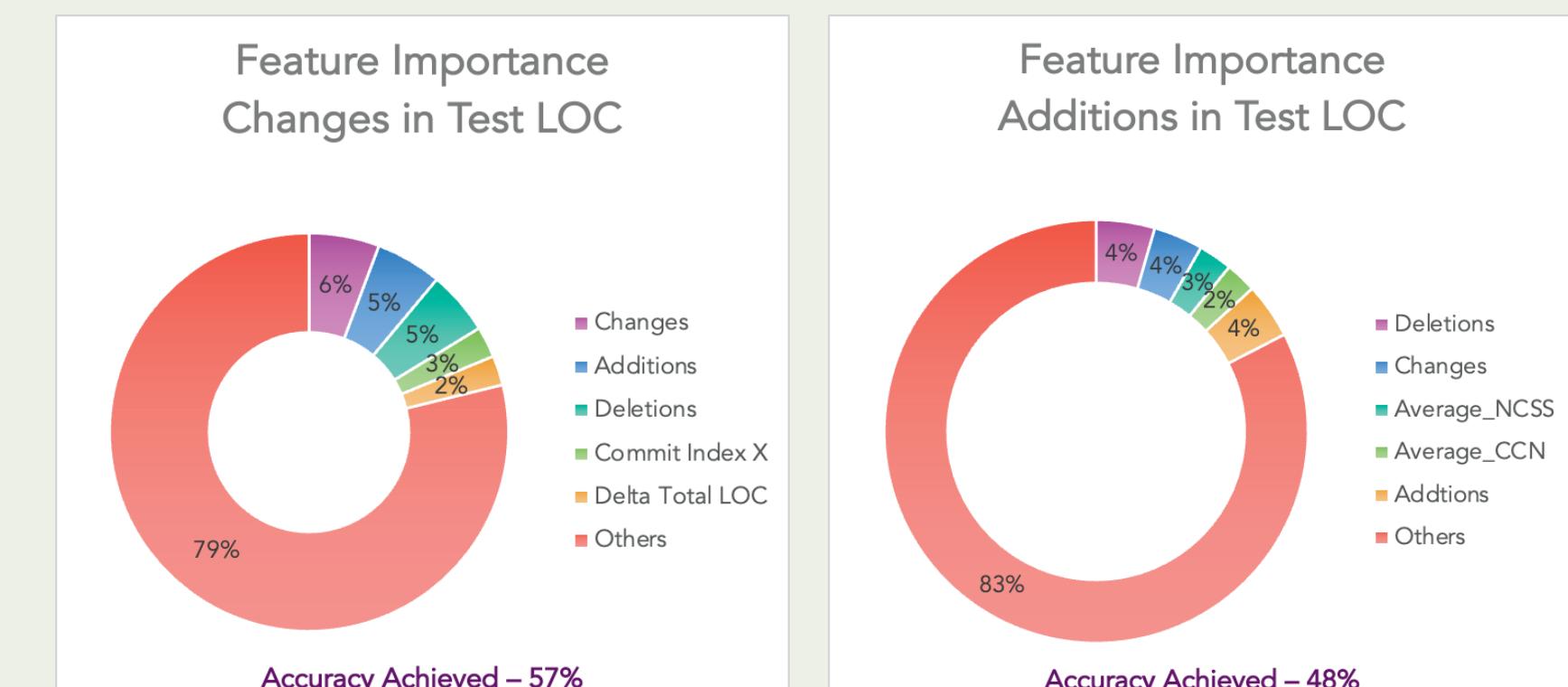
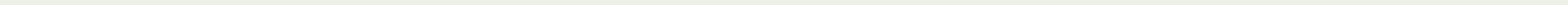
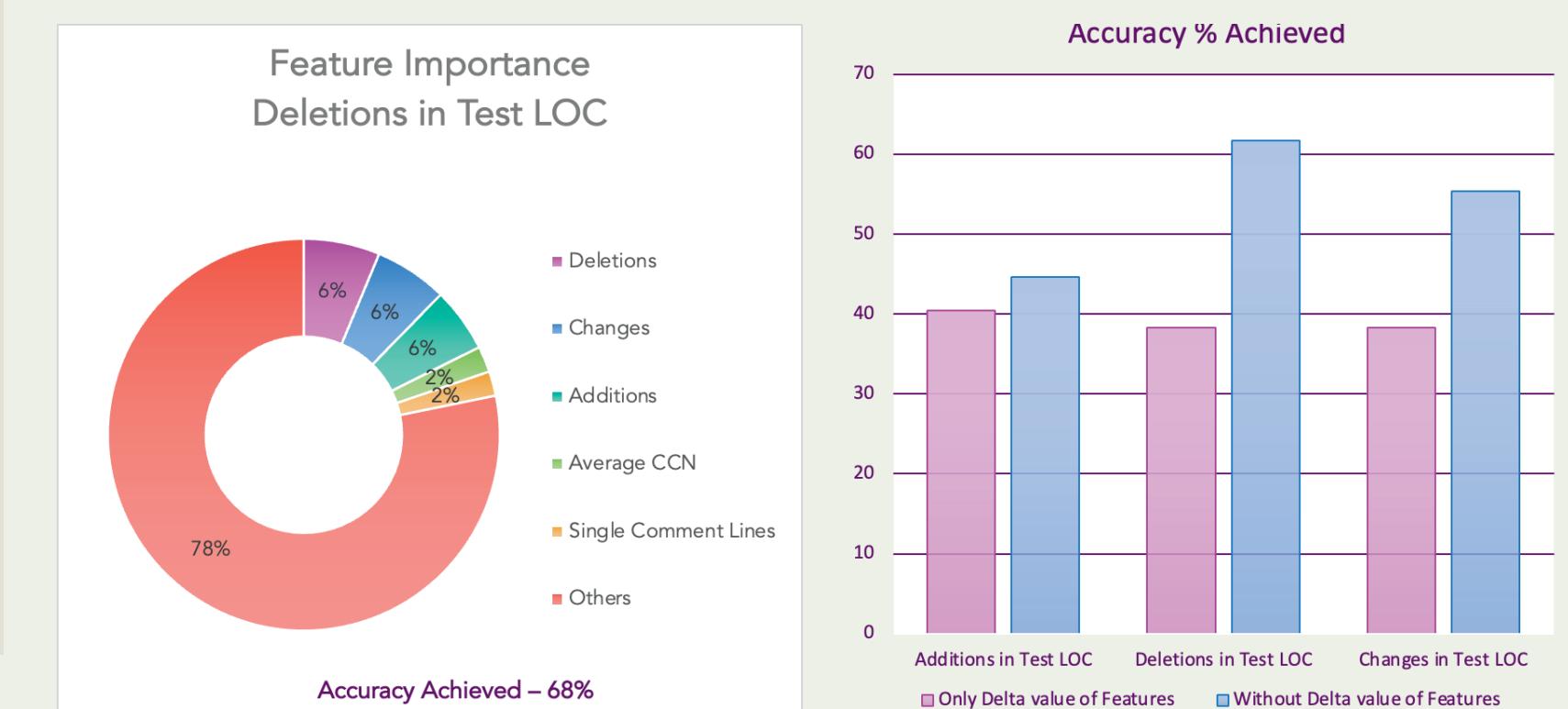
Source:
<https://www.teamwork.com/blog/project-failure/>

STEPS OF DATA EXTRACTION



RESULTS

- Results were analysed on three levels of Project Lifecycle:
 - Commits
 - Pull Requests
 - Releases (primarily)
- Best Results were obtained while analysing code changes for each existing Release and predicting test effort for the future Release.
- Best Performing ML Models were :
 - Random Forests (primarily)
 - Decision Trees



FUTURE SCOPE

- Improve the prediction results
- Work on industry provided data