Towards Better Understanding Developer Perception of Refactoring

Eman Abdullah AlOmar

September 25, 2019

Rochester Institute of Technology, NY, USA

Introduction

Introduction

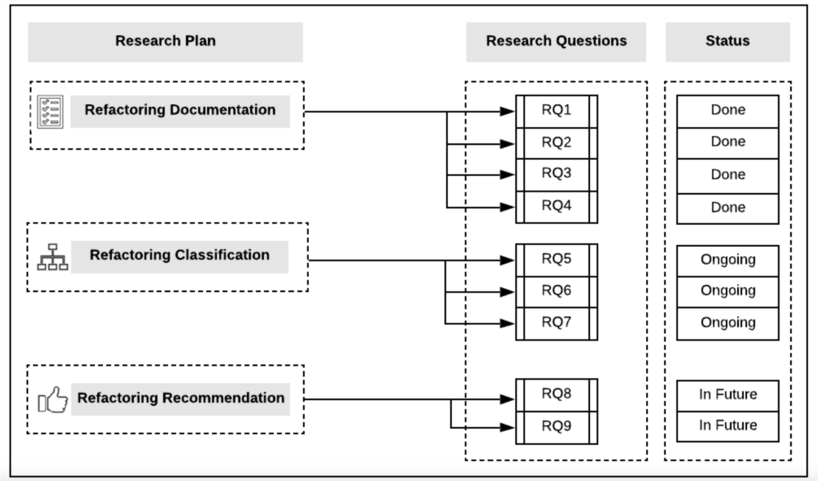
- Refactoring is the art of improving the quality of software design without altering its behaviour
- Accounts for up to 75% of the total development cost
- Other studies shown that refactoring is incorporated in other development activities

Research thrusts

- **RT1**. Mining-to-define the developer’s taxonomy of refactoring documentation
- **RT2**. Mining developer perception of refactoring
- **RT3**. Supporting the automation of refactoring recommendation

Research plan

Research agenda



Research Plan	Research Questions	Status
Refactoring Documentation	RQ1	Done
	RQ2	Done
	RQ3	Done
	RQ4	Done
Refactoring Classification	RQ5	Ongoing
	RQ6	Ongoing
	RQ7	Ongoing
Refactoring Recommendation	RQ8	In Future
	RQ9	In Future

Figure 1: Research plan overview

Research plan

- **Documentation**
  - Refactoring documentation
  - Refactoring documentation assessment
- **Classification**
  - Refactoring Developers Perception & Practices
- **Recommendation**
  - Refactoring automation

Methodology

- Data Collection & Refactoring Detection
- Self-Affirmed Refactoring Analysis
- Selection of Quality Metric

Conclusions

- **RQ1**. *What patterns do developers use to describe their refactoring activities?* – **87 SAR patterns have been identified**
- **RQ2**. *What are the quality issues that drive developers to refactor?* – **SAR patterns classified as: internal quality attributes, external quality attributes and code smells**
- **RQ3**. *Do commits containing the label "Refactor" indicate more refactoring activity than those without the label?* – **the use of "refactor" is not a great indication of refactoring activities**
- **RQ4**. *Do the developer perception of quality improvement align with the quantitative assessment of code quality?* – **different degrees of improvement and degradation of software quality**

Questions?