

Maintenance Effort Estimation for Open Source Software: A Systematic Literature Review

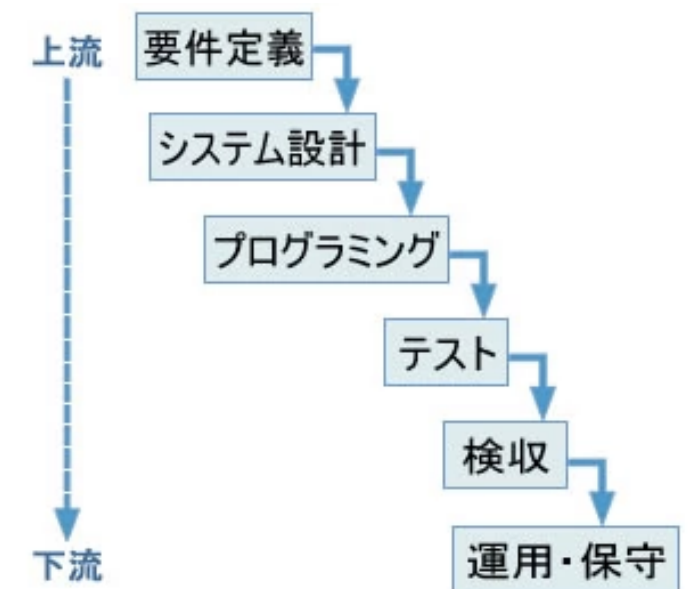
**Hong Wu , Lin Shi , Celia Chen , Qing Wang
Barry Boehm**

福井 克法

ソフトウェア保守

- ソフトウェア保守

- ソフトウェア工学において既存のソフトウェアを改良・最適化する
- バグを修正する



- 開発工程において、保守に労力がかかる

- バグを予測推定することは労力を削減できると言われている
- ソフトウェア保守に関する論文は多く存在している

目的

- 既存のソフトウェア保守に関する研究を理解することはOSS保守工数推定に大きく貢献する
 - 実際のメンテナンスの労力を削減する

そのため本研究では

→OSS保守に関する文献をレビューする

→文献を分類する

Research Question

- 1.OSSのために提案された保守工数推定にはどのような根拠があるか？
- 2.OSS開発記録から抽出される保守工数推定のメトリクスを分類することはできるか？
- 3.一般的なプロジェクトとOSSのデータセットのサイズはいくつか？
データセットがどのように保守を円滑にする手法に関係するか？
- 4.どの方法/アプローチが使用されているか
- 5.全体の推定の精度は？

文献抽出の方法

- 自動で抽出

Kitchenham et al. [19]の手法を用いて抽出

- 手動で抽出（文献を検索できるサービスを用いて）

- Inspec • Compendex • IEEE Xplore • ACM Digital Library
- Science Direct • Google Scholar

→ **PICO**のキーワードを用いて関連する文献を抽出する

PICOとは

P:Population I:Intervention C:Comparison O:Outcome

※今回はComparisonは対象外とする

- **Population : open source**

- ★ terms : “Open source”, OSS, FOSS, FLOSS, opensource, libre, free.

- **Intervention : 保守コスト見積もり**

- ★ terms : maintain, maintenance, evolve, evolution, bug fix, bug fixing, defect fixing, defect resolution, effort, cost, estimation, prediction.

- **Outcome : 保守コスト見積もりはOSSに対して有益に働く**

- ★ terms : empirical, validation, experiment, evaluation, “case study”

文献の抜粋方法

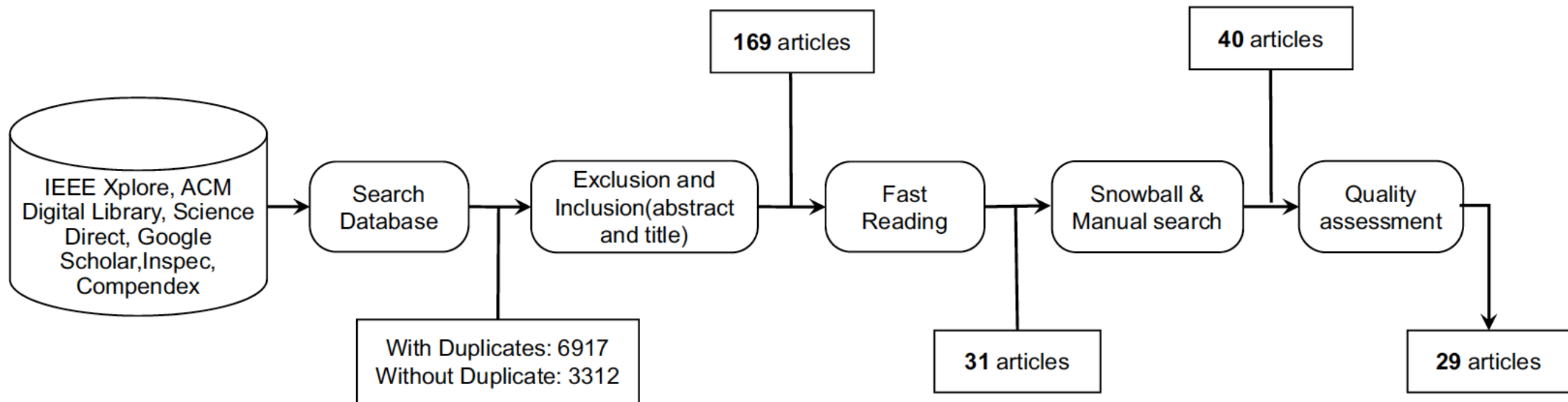


Fig. 1. Number of included primary studies during the study selection process

Snowball : 関連研究を探す手法

[8] D. Badampudi, C. Wohlin, and K. Petersen. Experiences from using snowballing and database searches in systematic literature studies. In EASE '15: Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering, pages 17:1-17:10, 2015.

Quality Assessment

下記の4つの問いに、

“YES” “Partially” “No”で判断し、 YES = 1 Partially = 0.5 No = 0で点数付けする

Q1 調査の目的は明確に定義されているか？

Q2 研究の背景は記述されているか？

Q3 解決策がきちんと提示されているか？

Q4 研究の限界を提示しているか？

<スコアの分類>

High:score=4 **Medium:**2 < score <4 **Low:**score<2

HighとMediumの文献を残す

結果：RQ1 part1

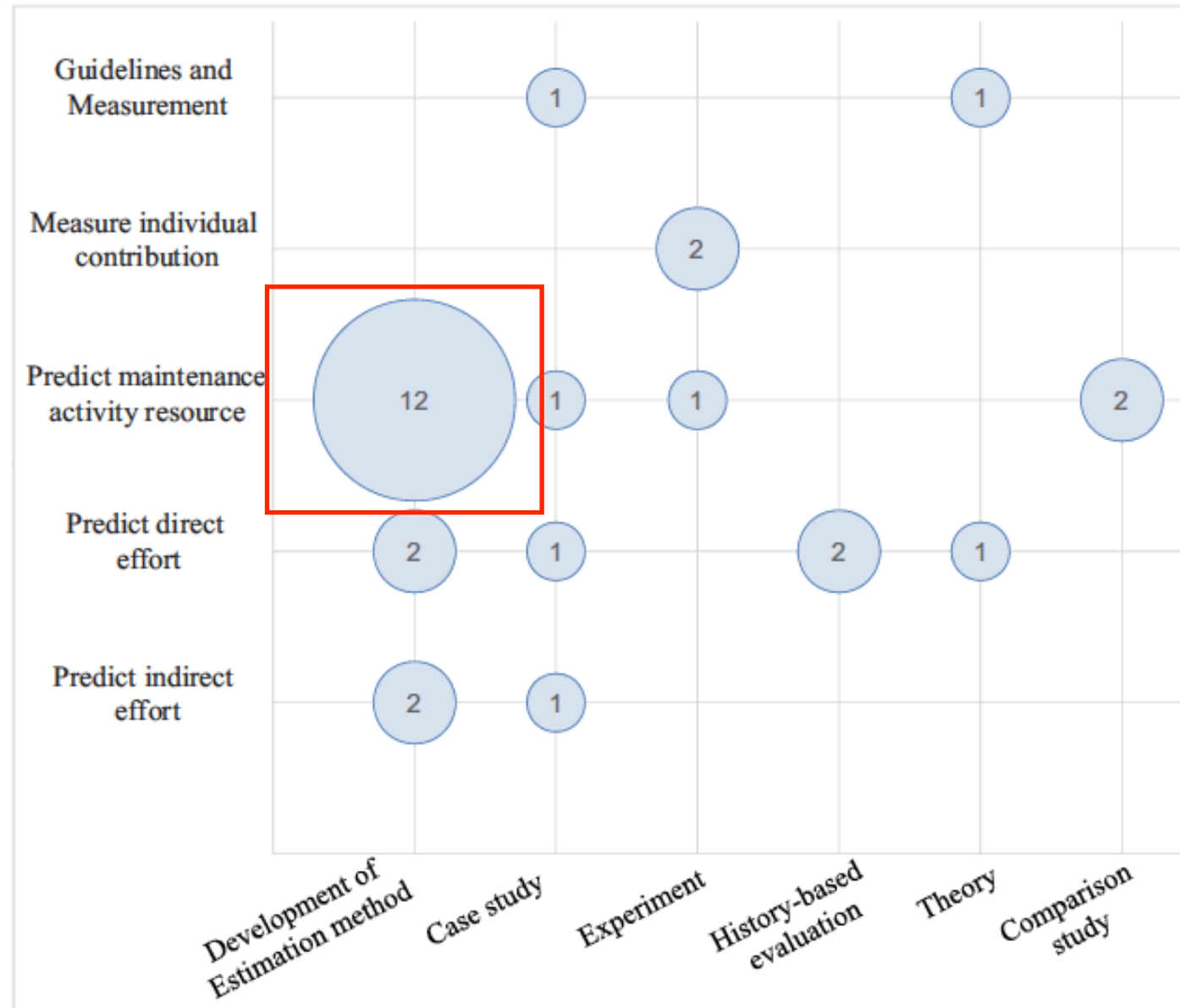
RQ1:OSSのために提案された保守工数推定にはどのような根拠があるか？

TABLE III
CLASSIFICATION OF TOPICS

Topic	Sub-topic	Paper Reference
Indirect effort prediction of the entire project	Source code-based estimation	[40],[4]
	Process-based estimation	[23]
Direct effort prediction of the entire project	People-based estimation	[21],[12],[20],[35]
	Activity-based estimation	[5],[11]
Effort prediction of maintenance activity	Peer code review	[25]
	Duplicate issues identification	[32]
	Bug fixing	[24],[6],[3],[38],[30],[14],[10],[16],[36],[29],[13],[2],[26],[27]
Individual contribution measurement		[15],[33]
Guidelines and discussion		[7],[37]

- 文献は5つのTopicに分類できる
- 全てそれぞれのTopicについて工数見積もり推定について書かれた論文である

結果：RQ1 part2



- ・ メンテナンスの活動を予測する分野の予測方法を発展させる研究が多い

結果：RQ2 part1

RQ2：OSS開発記録から抽出される保守コスト推定のメトリクスを分類することはできるか？

メトリクスを5つのカテゴリーに分類した

- プロジェクト （プロジェクト全体の特徴）
 - size, time, task, bugs, commits, developers
- 変更 （パッチのソースコードの特徴）
 - Cyclomatic complexity (CC)(循環的複雑度), function, LOC, file, class, operator
- 報告書
 - info(基本情報), stakeholder(利害関係者), time, similarity(類似性), process, quality
- 参加者
 - bug reporter, bug owner, bug trigger, bug collaborator
- コミュニティー
 - contributor, activity, workload(仕事量)

結果 : RQ2 part2

- 研究のトピックごとにメトリクスを分類すると

TABLE IV
TYPE OF METRIC SET USED BY EACH TOPIC

Topic	Type of Metric Set
Predict indirect effort of the entire project	Project{Size, Task}, Changes{CC, Function}
Predict direct effort of the entire project	Project{Time, Commits, Developer}, Participant{Bug Collaborator}, Community{Contributor}
Predict effort of maintenance activity	Project{Size, Time, Task, Bugs}, Changes{ALL}, Issue Report{ALL}, Participant{Bug Reporter, Bug Collaborator}, Community{workload}
Measure individual contribution	Participant{ALL}, Community{Activity}
Guidelines and discussion	Project{Size}, Changes{CC}

結果：RQ3 part1

- RQ3：一般的なプロジェクトとOSSのデータセットのサイズはいくつか？
データセットがどのように保守を円滑にする手法に関係するか？

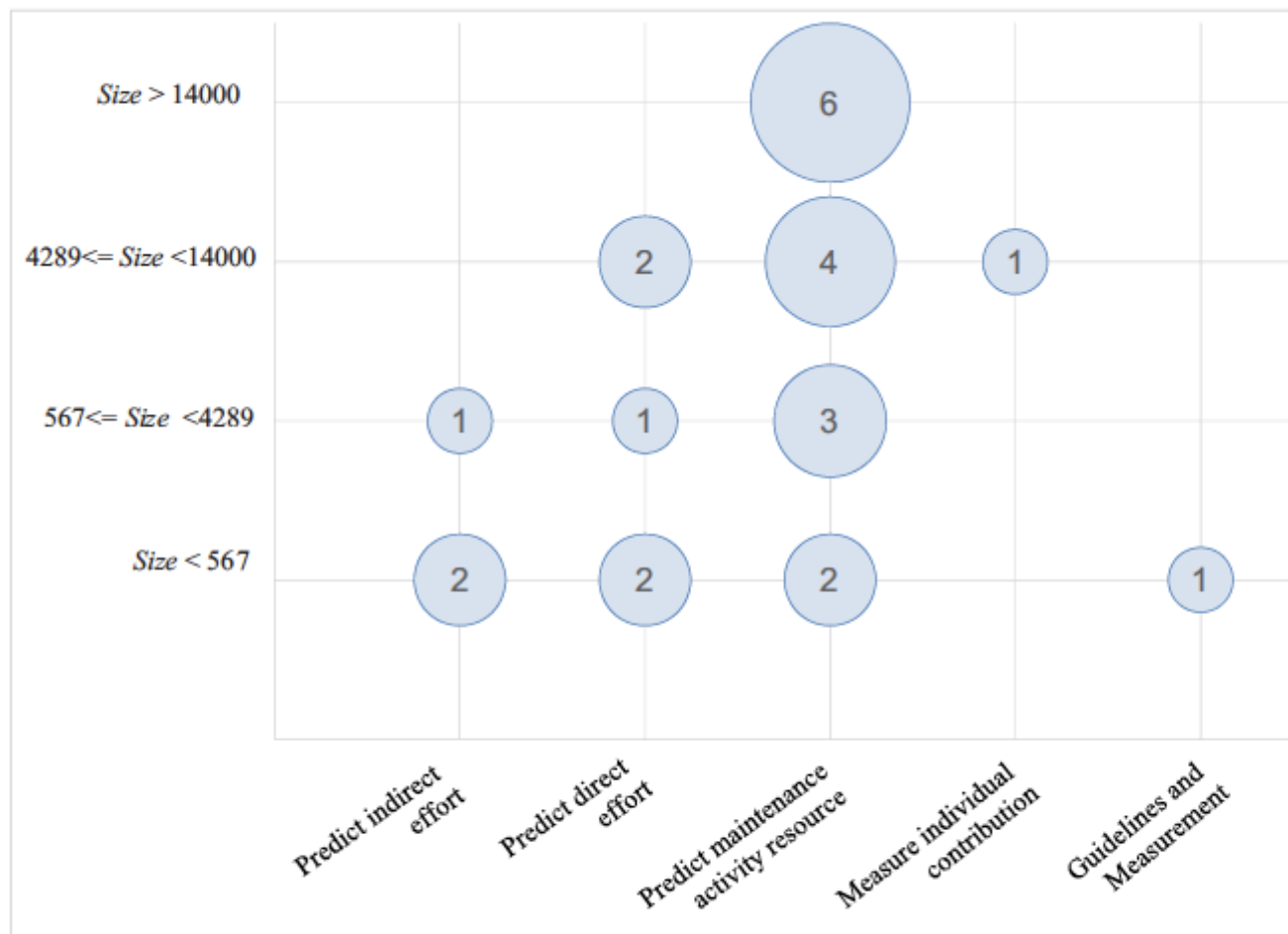
TABLE VII
OSS PROJECTS USED IN THE SELECTED PRIMARY STUDIES

OSS Project	Frequency	Paper
Mozilla	6	[24],[3],[14],[32],[36],[2]
Eclipse	6	[32],[24],[30],[14]
Jboss	3	[38],[16],[37],[29],[2]
Linux	3	[40],[4],[11]
GNOME	3	[21],[14],[2]
Soureforge	2	[12],[20]
Openstack	1	[35]
Filezilla, WxWidgets, Lighttpd	1	[23]
Google Chromium Project	1	[33]
Hibernate, Mule, Spring	1	[37]
Google Android	1	[25]
Ubuntu	1	[6]
FreeBSD	1	[10]
Codehaus	1	[16]
Thunderbird, Add-on SDK	1	[36]
QT,Qpid,Geronimo	1	[13]

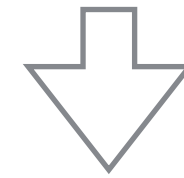
- Mozillaを分析する研究が
もっともっと多い
- 大規模プロジェクトを分析
している

結果 : RQ3 part2

- それぞれのトピックのdatasetのサイズの分布



29個中6個の研究が、より大きいデータセットを使用している



大きなデータセットから得られるデータは、より正確である可能性が高い

Fig. 4. The number of studies by dataset size

size: データセットに含まれるバグIDの数と対する処理の数

結果 : RQ4

RQ2 : どの方法/アプローチが使用されているか？

Topic	Sub-topic	Paper ID	Estimation Method	Metrics
Indirect effort prediction of the entire project	Source code based estimation	[40], [4]	Linear regression model	Size, Task
	Processes-based estimation	[23]	Classification	CC, Function
Direct effort prediction of the entire project	People-based	[21], [12], [20]	Manpower function	Time, Participant, Community
		[35]	Classification	Participant, Community
	Activity-based	[11], [5]	Linear model	Commit, Community
Effort prediction of maintenance activity	Effort for peer code review	[25]	Non-linear equation	CC, Function, Class, LOC
	Effort for duplicate issues identification	[32]	Classification model	Issue-report related metrics, Participant, Community
	Bug fixing time	[6], [10]	linear regression	Issue-report related metrics
		[24], [30], [14], [13], [10]	Decision tree	Bugs, Time, Community Issue-report related metrics, Participant,
		[3]	Support Vector Machine (SVM)	Size, CC, Function, File, LOC, Participant
		[38], [16]	K-Nearest Neighbor (KNN)	Issue-report related metrics
		[36]	Apriori & K-means	Issue-report related metrics
		[30], [29], [13], [10]	Logistic Regression	Issue-report related metrics Participant
		[30], [2], [10]	Naïve Bayes	Issue-report related metrics
		[26]	Distribution Functions	Task
		[27]	Average weighted similarity	Issue-report related metrics

線形回帰

分類

労働力相関関係

分類

一般線形モデル

非線形方程式

分類モデル

線形回帰

決定木

サポートベクターマシン

k近傍法

Apriori & K-means

ロジスティック回帰分析

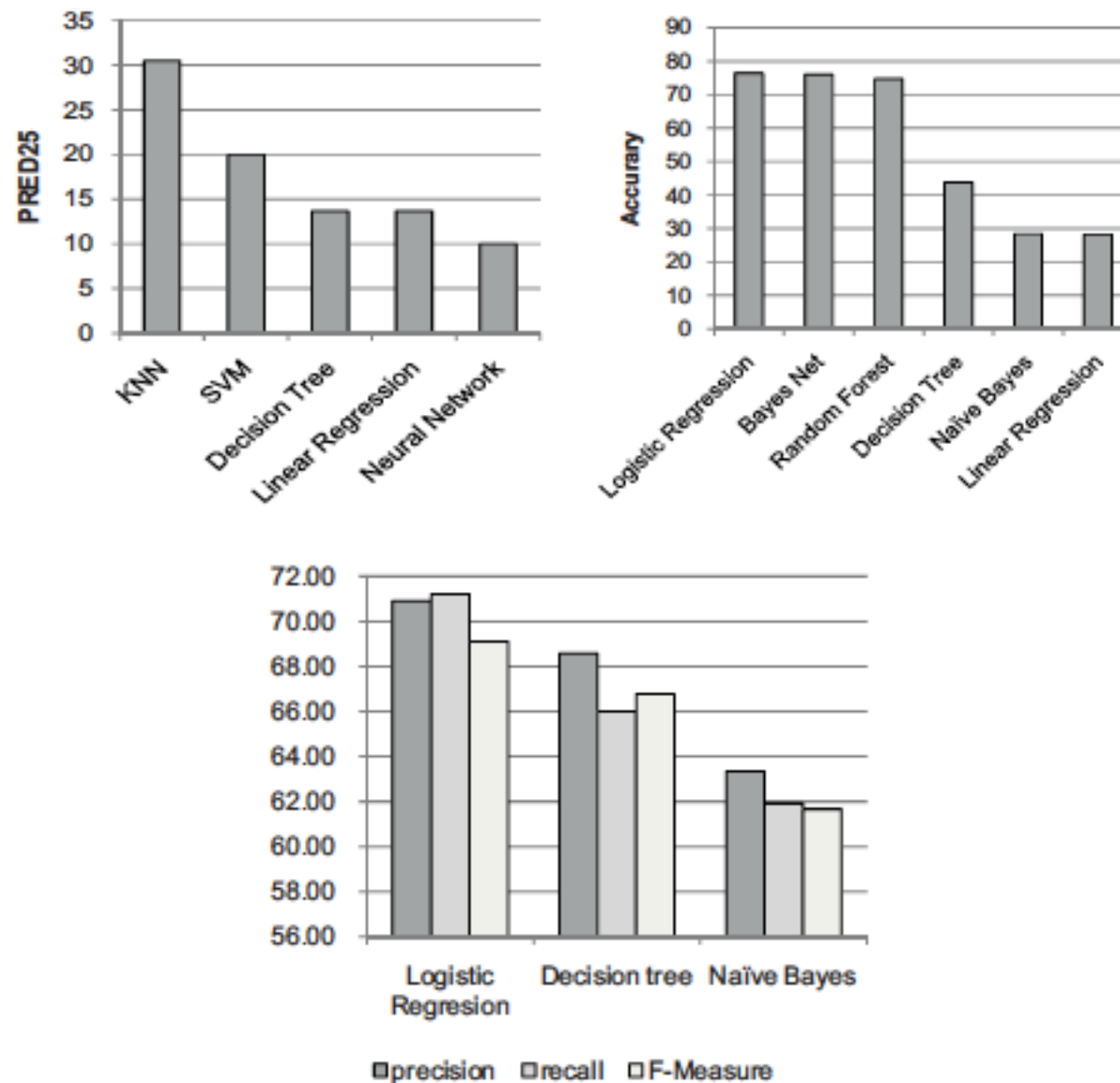
単純ベイズ分類器

分布関数

加重平均類似性

結果 : RQ5

RQ5 : 全体の推定の精度は？



[F-Measure]

F値

[recall]

再現率

[Precision]

適合率

[Accuracy]

正確さ

[PRED25]

見積り誤差率の絶対値が25%以内であるプロ

ジェクトの比率

Fig. 6. Average performance of different bug fixing time prediction methods

まとめ

- ソフトウェア保守にはさまざまなアプローチがある
- 使用されるデータセットは正確なデータを得るため大きいものが多い
- 使用される手法により精度が異なる
- 今後、現在の手法の精度の向上 改善が要求される