
不具合管理システム利用時の 不具合修正プロセス改善のための 滞留時間分析手法の提案

伊原 彰紀 大平 雅雄 松本 健一

背景

- ❖ 不具合管理システムの利用
- ❖ Linuxの不具合の残存期間
平均1.8年間
- ❖ 複雑なバグ修正プロセス

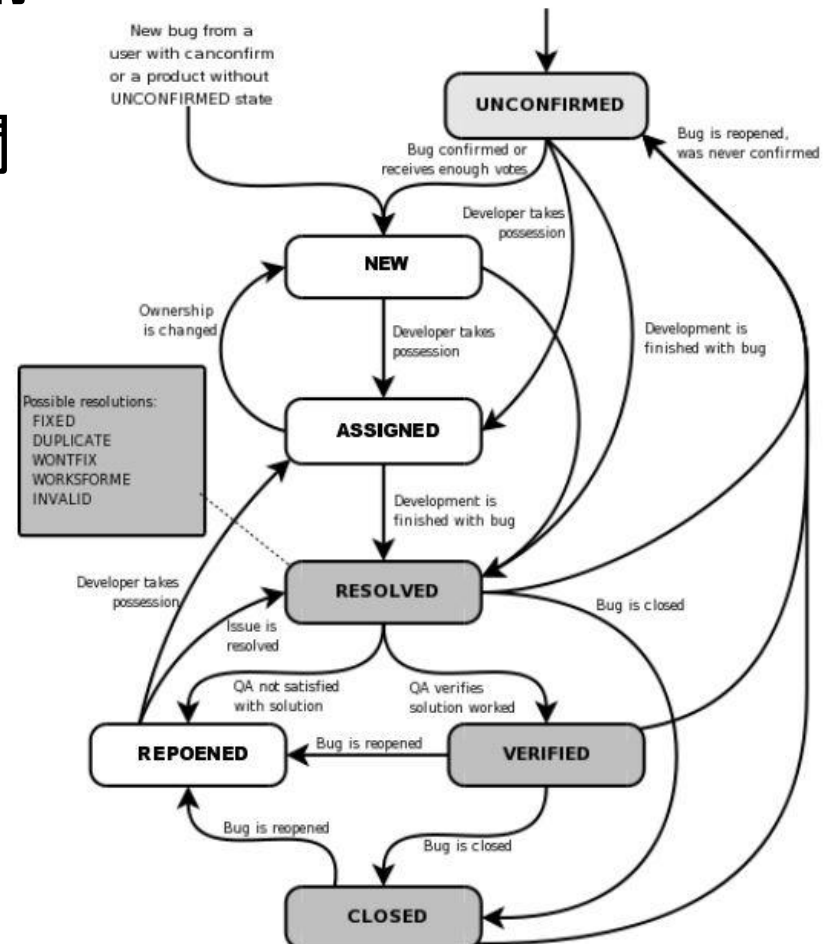


図 1 Bugzilla 利用時の不具合修正プロセス¹⁰⁾ 1/10

不具合の滞留時間の研究

- ❖ Two Case Studies of Open Source Software
Development: Apache and Mozilla — AUDRIS MOCKUS
 - 広範囲で利用されるモジュールの不具合は早く修正される.
- ❖ Towards a Simplification of the Bug Report form in Eclipse — Israel Herraiz
 - 重要度が高い不具合は滞留時間が短い.
 - 重要度, 優先度は簡素化が可能である

本研究

❖ 修正プロセスの分割

- 未対応時間と修正時間
- 修正開始から完了までの時間を詳細にする.
- ボトルネックの理解

どうしたら不具合を早く修正できるかを解明

❖ Firefoxを対象としたケーススタディ

修正時間分析

❖ 不具合の分類

- 未対応時間のみの長い不具合
- 修正時間のみの長い不具合
- 未対応時間と修正時間が共に不具合

遷移パターンを分析することで、修正プロセスを改善できる。

❖ 修正作業の分類

- 担当者決定→ 解決
- 不具合承認→ 担当者未決定→ 解決
- 不具合未承認→ 解決

修正時間に最も影響を与えている状態を特定。

❖ 修正確認の分類

- 解決→ 修正確認
- 解決→ 再修正決定

Firefoxを対象としたケーススタディ

表 3 Firefox における不具合の滞留時間

	滞留時間	未対応時間	修正時間
平均 (日)	144.14	7.85	136.29
中央値 (日)	54.88	0.02	47.05
標準偏差	225.85	37.10	221.27
分散	50987.71	1375.96	48940.58
最大値 (日)	2415.40	478.38	2376.38
最小値 (日)	0.00	0.00	0.00

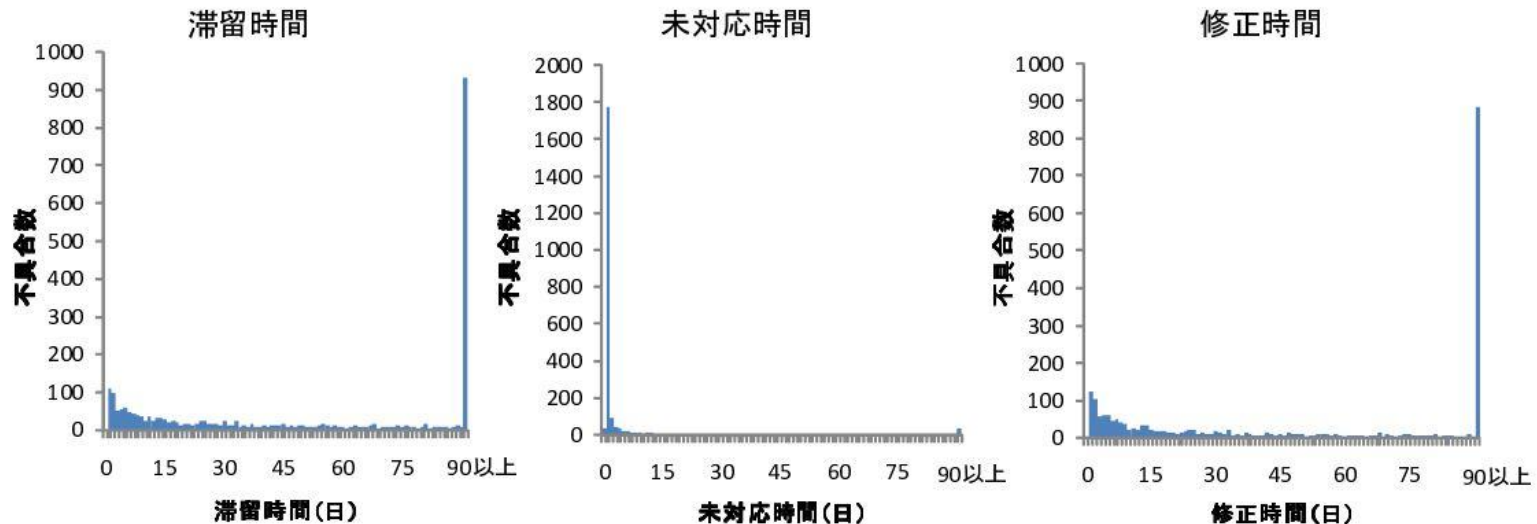


図 4 滞留時間と 未対応時間と 修正時間の分布

Firefoxを対象としたケーススタディ2

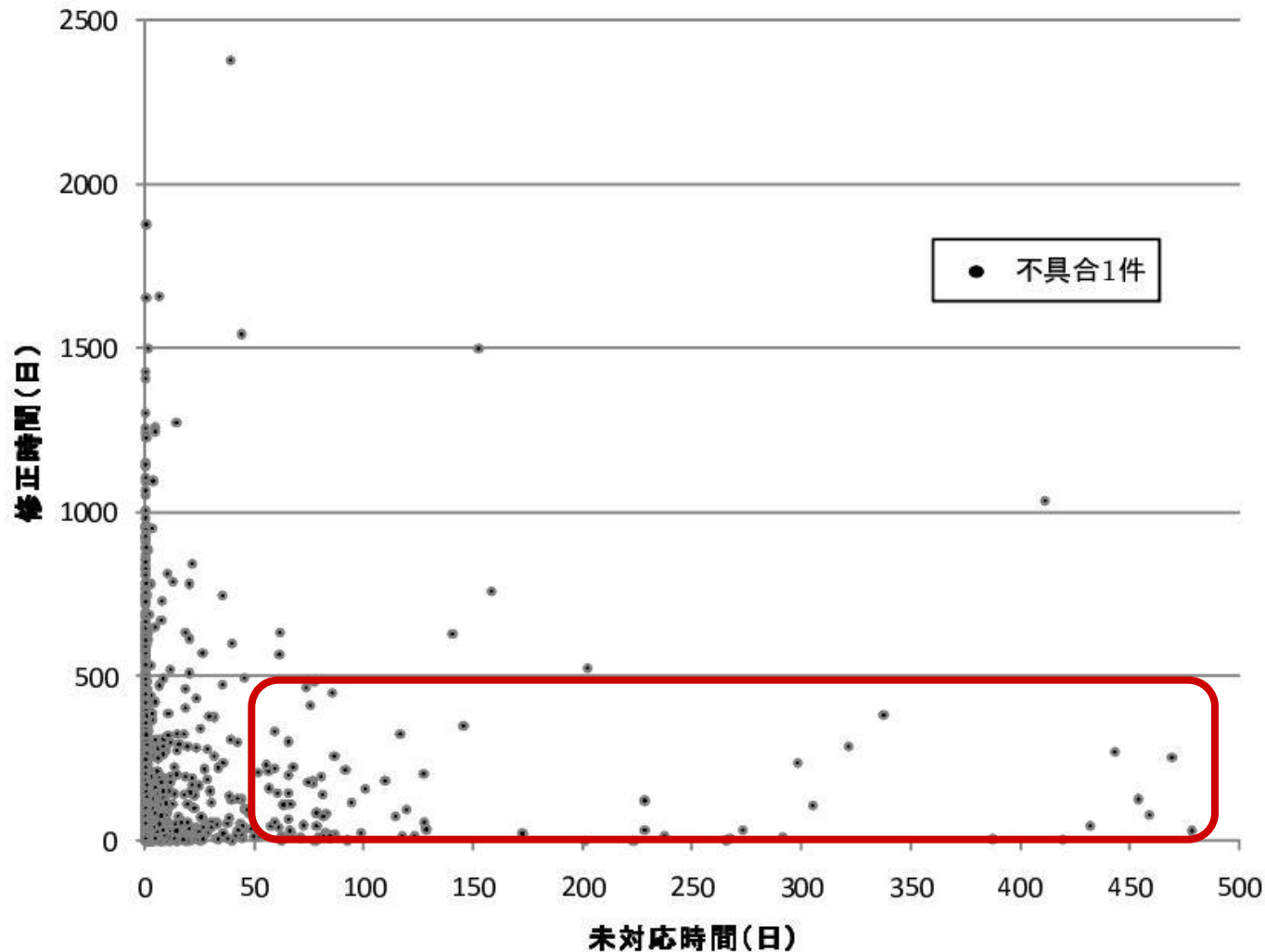


図 5 未対応時間と修正時間の関係

遷移図からの分析

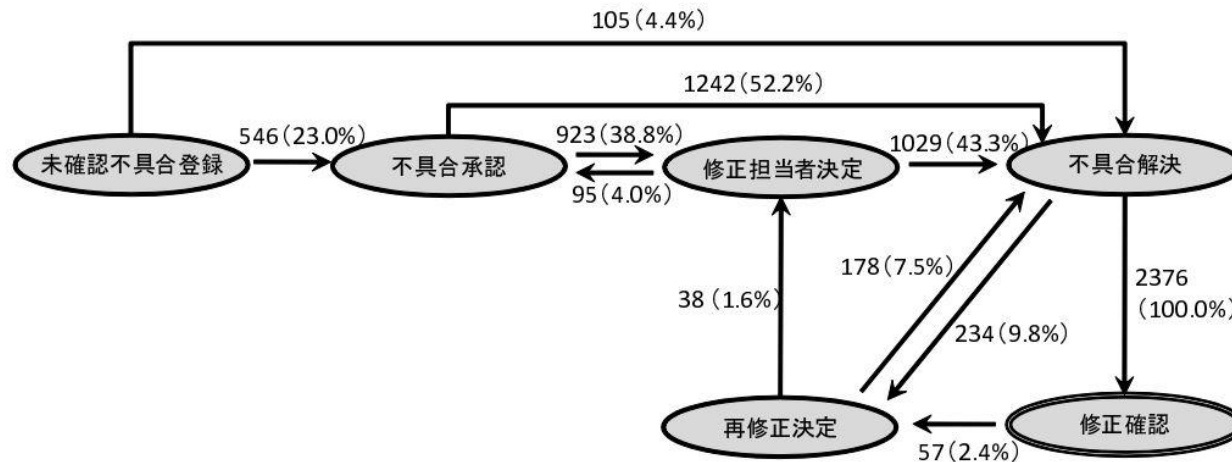


図 6 各修正プロセスにおける状態遷移件数 (数字は各状態間を遷移する不具合の件数, 括弧内の数字は分析データ数に対する割合)

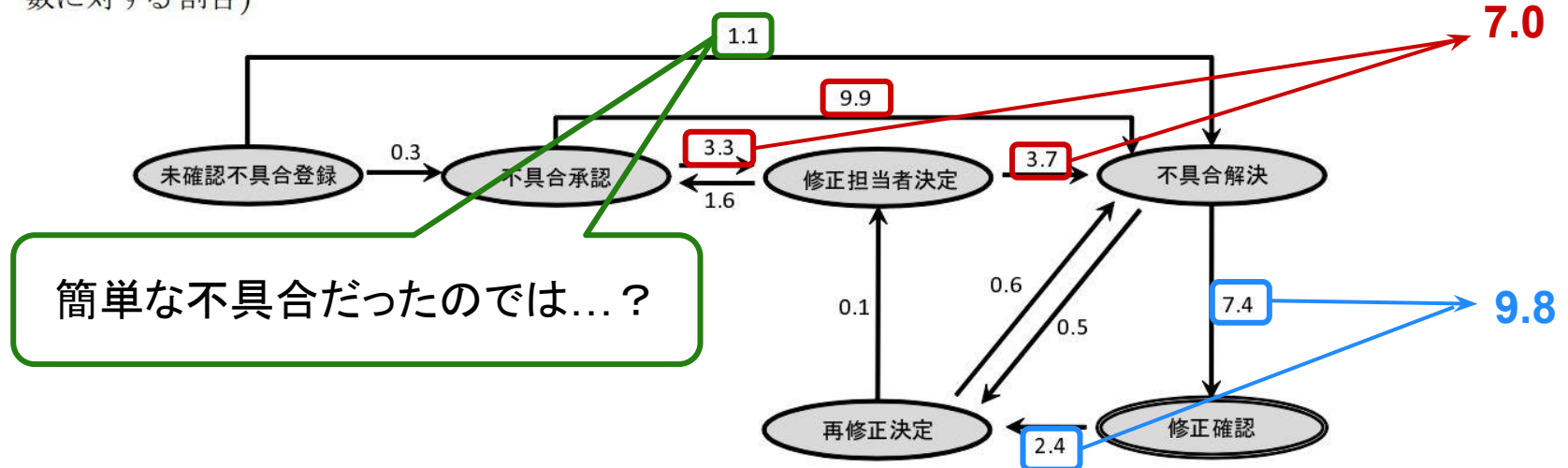


図 7 各修正プロセスに要する時間 (数字は各状態間に要する日数の中央値)

修正に要する時間

❖ 修正担当者決定に時間がかかる.

- 担当者が決まった場合 7.0日
- 担当者が決まらない場合 9.9日

担当者を早く決めることで、時間短縮

❖ 修正確認作業に7.4日かかる.

- 修正確認→再修正決定に9.8日.

修正確認を効率化して、時間短縮

考察

❖ Herriaizらの研究⁽¹⁾の指摘

- 滞留時間だけで重要度, 優先度の簡素化は不適.
- 未対応時間と修正時間を考慮すべき.

❖ Wangらの研究⁽²⁾の裏付け

- 修正担当者の有無が滞留時間に影響

❖ 修正時間にかかる時間

- 修正確認に時間がかかる

(1) Towards a Simplification of the Bug Report form in Eclipse

(2) Measuring the evolution of open source software systems with their communities, SIGSOFT Softw.

今後の展望

- ❖ 本研究では, 不具合の特徴を考慮していない.
- ❖ 不具合の特徴にあわせた効率的な修正活動を支援.



Firefox®