

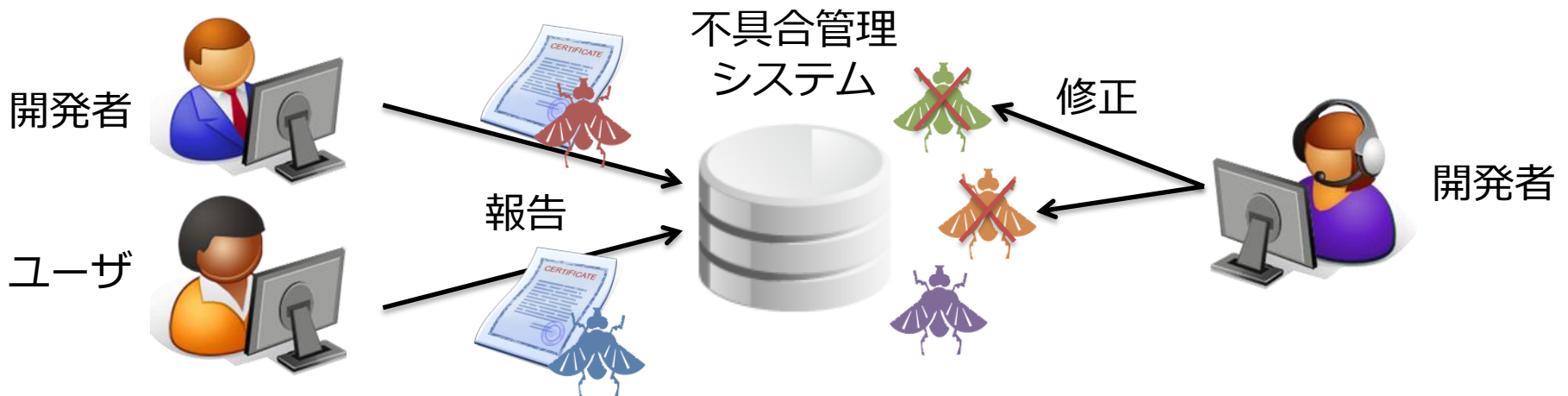
不具合修正プロセスにおける 優先度と修正作業の 関係理解のための分析

和歌山大学

○吉行 勇人 大平 雅雄

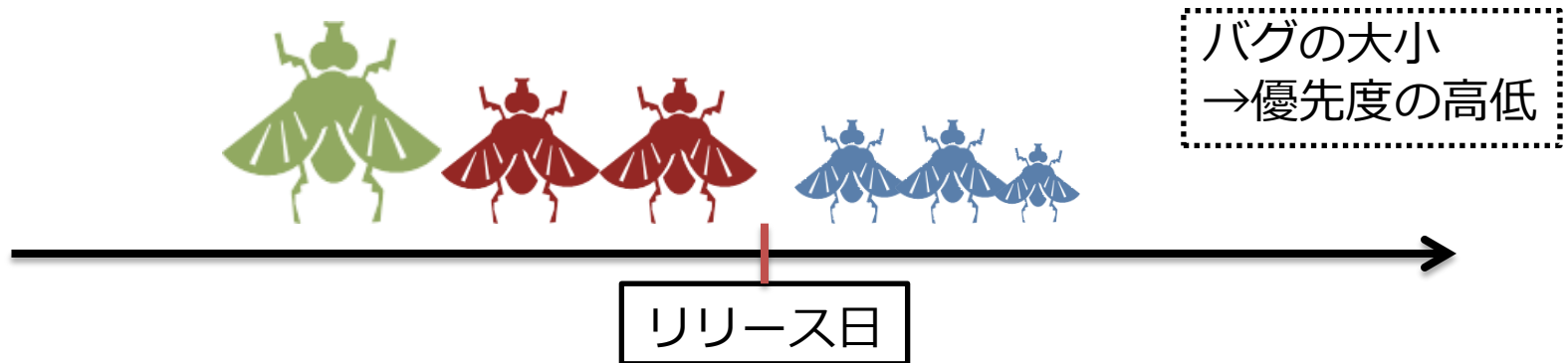
背景

- OSSプロジェクトでは不具合管理システムを用いて不具合に対応している
 - 大規模なOSSプロジェクト→数百件/日[1]



不具合の優先順位付け

- 次のリリースを目安に不具合に優先度を設定



- 各不具合報告を確認して設定する必要がある
– 優先度決定の自動化[2]

[2] Tian, Yuan and Lo, David and Sun, Chengnian, DRONE: Predicting Priority of Reported Bugs by Multi-factor Analysis, Proceedings of the 2013 IEEE International Conference on Software Maintenance, pp. 200--209, Sep 2013.

優先順位付けの影響

- 先行研究において
 - 優先度は必ずしも修正時間に影響を与えない[3]
- 予備調査の結果
 - 「優先度高」と設定された不具合のうち約20~30%の不具合はリリース後に修正完了している



不具合に設定された優先度 ≠ 実際の修正作業



プロジェクトが想定した品質を満たせない

目的

- 目的

- 不具合の優先度と実際の修正作業が一致していない不具合の特徴を分析し、優先度の自動設定や修正時間予測の精度を向上させる

- サブゴール

- 不具合の優先度と実際の修正作業が一致していない不具合の特徴を明らかにする
- 分析結果を用いて優先度の自動設定や次のリリースまでに修正完了するか否かの予測の精度を向上する

分析方法

- 対象プロジェクト



2004年9月~2014年10月



2006年9月~2014年10月

- 対象不具合

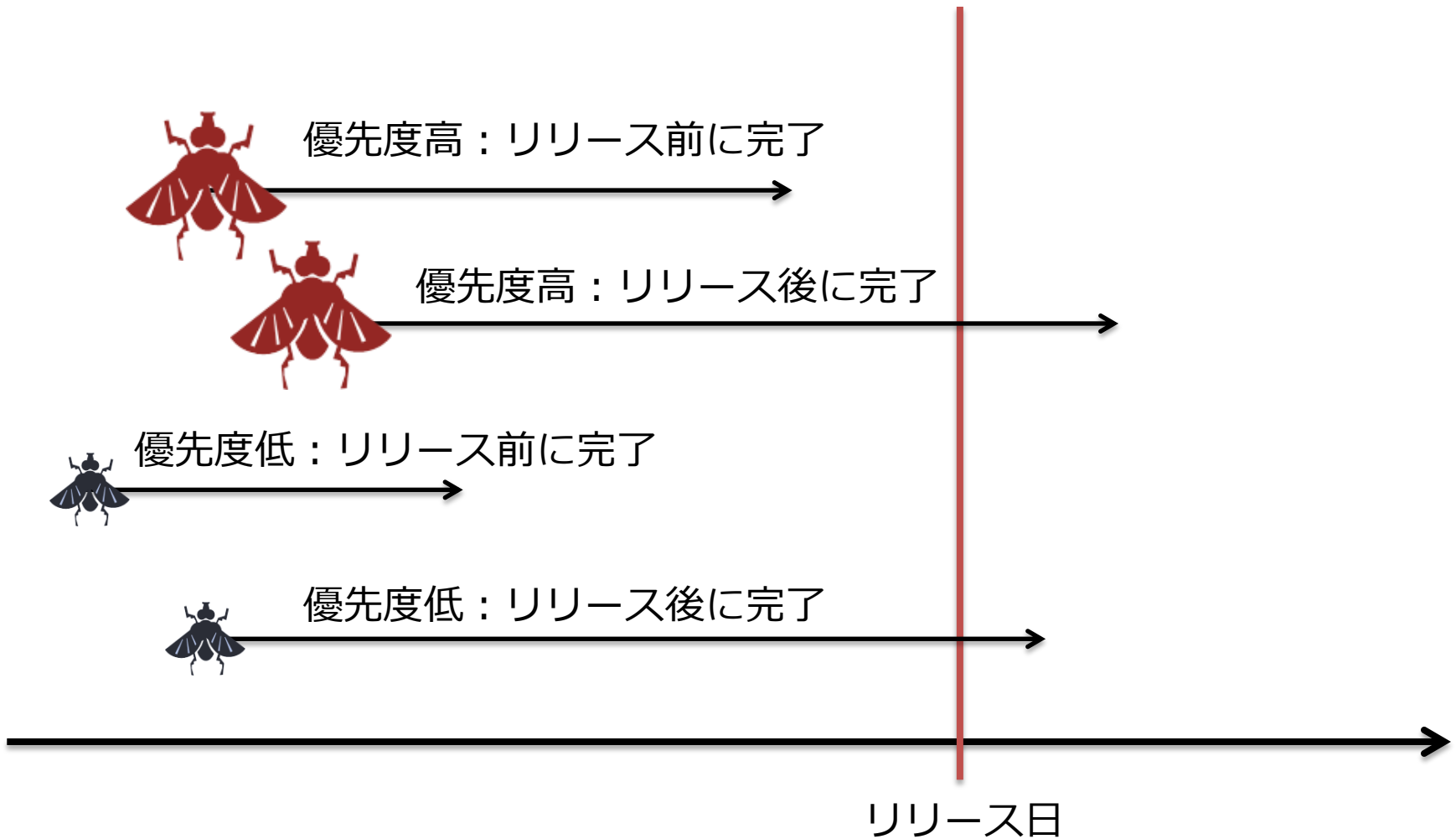
- 不具合管理システム: Jira
- Derby: 1106件 Qpid: 809件
- すでに修正が完了している不具合
- 修正のためにソースコードを変更している不具合
- 優先度がデフォルト値でない不具合

– Blocker > Critical > ~~Major~~ > Minor > Trivial

優先度高

優先度低

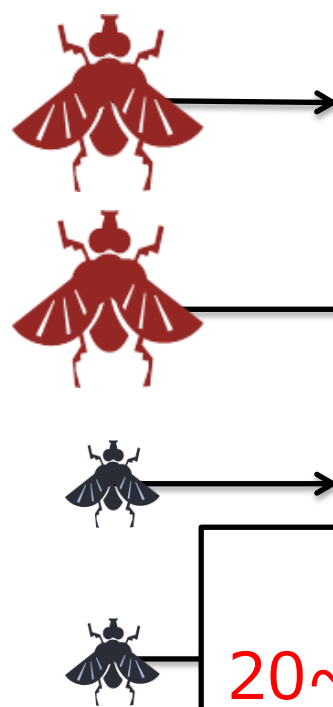
不具合の分類



分析内容

- 分類された不具合ごとにそれぞれ算出
 - 件数
 - 報告されてから解決までの時間（修正時間）
 - リリース日を超えてからの時間（リリース後のみ）
 - 修正時に変更したファイル数および変更したソースコードの行数
 - 不具合に対するコメント数
- 件数以外の結果は中央値

分析結果：件数



	Derby	Qpid
高：前	68	152
高：後	25	37
低：前	649	488
		2

優先度の高低にかかわらず
20~30%の割合でリリース後に修正完了している

分析結果：修正時間

リリース後の（）内はリリース日を超えてから何日かかったか

Derby

Qpid

リリース前に修正完了した不具合のほうがリリース後に修正完了した不具合よりも修正時間が長い

低：前

5.21

1.80

リリース後に修正完了した不具合はリリース日を超えてから長期間放置されやすい

分析結果：修正ファイル数・行数

		Derby	Qpid
	高：前	6 : 44	6 : 22
	高：後	8 : 103	6 : 64
	低：前	6 : 24	4 : 22
 優先度が高く、リリース後に修正完了した不具合は修正規模が大きい			38

分析結果：コメント数

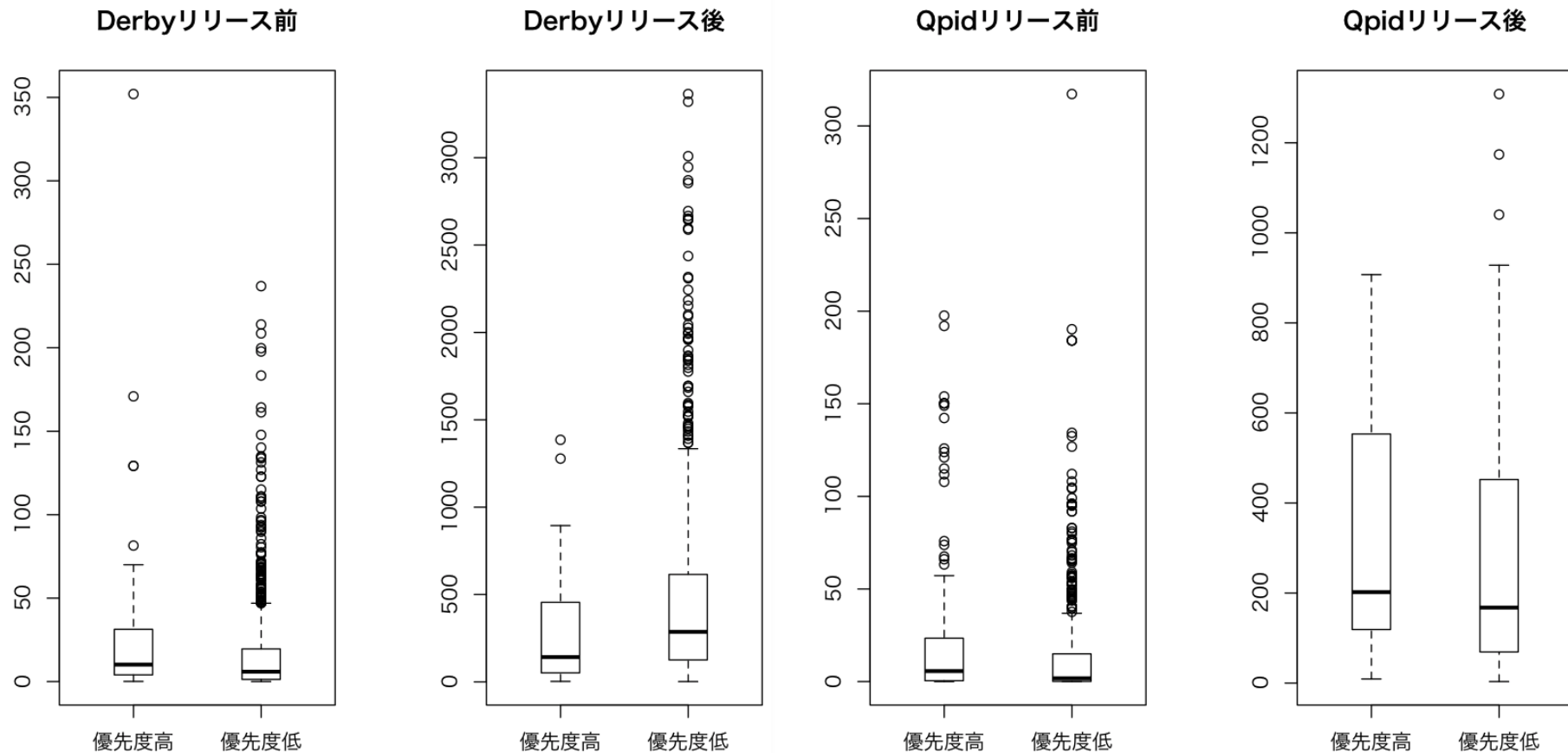
		Derby	Qpid
	高：前	9	3
	高：後	14	3
	低：前	4	2

優先度が高く、リリース後に修正完了した不具合は多くの議論を必要とした場合が多い

考察

- 優先度が高くリリース後に完了した不具合のコメント欄を参照した結果
 - 次のリリースが過ぎてから修正活動が始まる場合
 - そもそも次のリリースを目標にしているのかしていないのかを判断する必要がある
 - リリース後に完了した不具合はテストケースの追加を伴う場合が多い(要検証)

修正時間



テストケース追加検証

- ファイルパスに「test」が含まれる
 - java/systest/~~~
 - qpid/~~~~~/~~Test.java
- (プログラムの不備がありそう)

テストケース追加検証 結果

Derby	修正ファイル	修正行数	テストファイル	テスト行数
高：前	1	17	1	11
高：後	2	27	2	36
低：前	1	3	1	5
低：後	1	6	1	8

Qpid	修正ファイル	修正行数	テストファイル	テスト行数
高：前	5	40	0	0
高：後	6	114	1	4
低：前	4	37	0	0
低：後	4.5	63	0	0

まとめ

- 不具合修正の優先順位
 - リリースを目安に設定する
 - 優先度と修正作業が一致していない場合が存在する
 - 不具合に関する基本的な情報を分析した
- 分析
 - リリース前に完了した不具合とリリース後に完了した不具合の特徴の違いを明らかにした

今後の予定

- より詳細な分析
 - 不具合修正担当者や議論の参加者
 - 不具合の説明部を用いて不具合分類の詳細化
- 予測
 - 優先度の自動設定の精度向上
 - 次のリリースまでに修正完了するか否か（修正時間予測）

議論したい内容

- 報告された日から次のリリースまでに残されている時間を考慮していない
 - リリース直前に報告された不具合は修正不可
 - リリース前の一定期間は変更を行わない
- 予測モデルへの適用
 - 今回分析した内容は修正後にしか確認できない
 - 不具合報告時に得られる情報で分析を行いたい