

High Impact Bugの特徴を明らかにするための目視による調査

大平研究室 B4

前田欣耶

背景

- OSS開発では数多くのバグが発見されるため、管理者はどのバグを優先すべきかを決める必要がある
- この問題を解決するための研究は数多く存在するが、一つ一つのバグのプロセスとプロダクトへの影響力を等しく扱っている

High Impact Bugの特徴を明らかにする

データセット

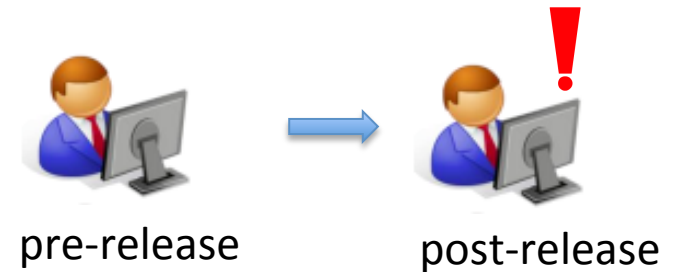
- 選択したプロジェクト
 - Camel, Ambari, Derby, Wicket
- 各プロジェクトからランダムに1000件ずつ選択したバグの条件
 - すでに修正が完了している
 - 影響を及ぼしたバージョンが一つ以上ある
 - 修正のためにソースコードを変更している

分析方法

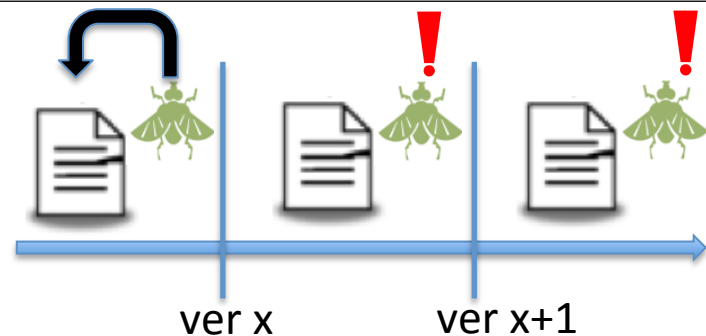
1. 選択した4000件のバグを目視で6種類の High Impact Bugに分類する
2. 分類が正しく行われたどうかを第三者により確認する
3. 分類結果を集計し, 分析する

High Impact Bugの種類(1)

- Surprise bug
 - 予想していなかったタイミングで出現するバグ

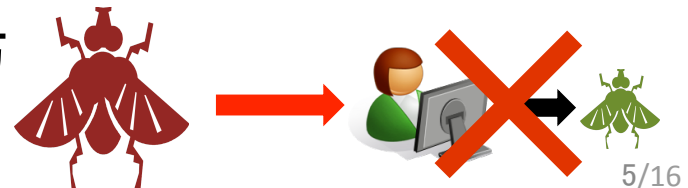


- Dormant bug
 - 影響を及ぼしている最古のバージョンの次のバージョンがリリースされた時より、報告された時の方が遅くなっているバグ



報告された時 < ver x+1がリリースされた時

- Blocking bug
 - 他のバグが修正されるのを防いでいるバグ



High Impact Bugの種類(2)

- Security bug

- セキュリティのリスクを伴っているバグ

ex

- Session error
- Password is shown in the log file

- Performance bug

- システムのパフォーマンスの低下を伴っているバグ

ex

- Memory leak
- Infinite loop

- Breakage bug

- 新機能を追加もしくは他のバグが修正されたことにより発生したバグ

ex

- Regression test error
- Ver.x works, but Ver.x+1 doesn't work

分析内容

- 各種バグの
 - Priority
 - 修正時間
 - 修正担当者数
 - 修正ファイル数
 - コメント数

分析結果：修正時間(中央値)

	Sur	Dor	Blo	Sec	Per	Bre	$\overline{\text{Sur}}, \overline{\text{Dor}}, \overline{\text{Blo}}, \overline{\text{Sec}}, \overline{\text{Per}}, \overline{\text{Bre}}$
camel	9.78	72.08	4.20	1.90	0.08	0.06	0.04
ambari	0.54	19.73	2.95	2.08	0.49	5.54	0.09
derby	158.64	43.62	14.21	7.14	5.42	580.69	2.99
wicket	12.25	0.53	36.18	36.30	0.27	3.38	18.76

High Impact Bugはそうでないバグに比べて修正時間が長くなるといえる

分析結果 : Priority

- あ

分析結果：修正担当者数

- あ

分析結果：修正ファイル数

- あ

分析結果：コメント数

- あ