



研究計画

重大な影響を及ぼす不具合の検出手法の構築
~重大な影響を及ぼす不具合の定性・定量的評価~

背景

- OSS PJには日々、大量の不具合が報告されている
- プロジェクトの管理者によって、以下を決定
 - どの不具合から優先して修正すべきか
 - 不具合の担当者を誰にするべきか
- 多くの割当てが失敗している
 - 不具合修正の長期化
 - リソースの無駄遣い

関連研究

- バグトリージの支援手法が数多く提案されている
 - Anvikらは、過去の修正した不具合の内容を分類器に学習・推薦→優秀な開発者に多くの不具合を割当てる
 - 柏らの手法は、Anvikらの手法を改良し、優秀な開発者に多くの不具合を割当てないように割当て
- 多くの手法が不具合を同様に扱っている
- 重大な不具合は優秀な開発者に割当てるべき

目的とアプローチ

- **最終の研究目的**

- 重大な影響を及ぼす不具合を検出
 - できれば影響を数値化したい（経済効果の算出手法のように）

- **直近の研究目的**

- 開発者にとって不具合の重大な影響とは何かを理解
 - 重大な影響を及ぼす不具合はどんなものかインタビュー
 - インタビューを基に不具合を抽出し、データ分析

研究手順

1. High Impact Bugを抽出

1. 開発者にHigh Impact Bugはどんなものをインタビュー
2. 回答をもらった不具合を分類（カードソート）
3. 分類ごとにキーワードを抽出(一般化のため)
4. キーワードを元に不具合を検索・追加

2. 定量的な分析を実施

1. 不具合の特徴
2. 開発者の行動

※ 手順1はあくまで個人の主観やドメインによるので、手順2で定量的に評価

定量的なデータ

- **不具合の特徴**
 - 優先度
 - 重要度
 - パッチの大きさ
- **開発者の行動**
 - 報告されてからアサインまでの時間
 - 議論の回数

修正者の熟練度も測ってみたい. . .

その他 調べたい事 (考察)

- 修正は大変なのか
- High Impact Bugは本当にHigh Impactか
 - Surprising bug
 - Dormant bug
- 今回の分類の中で、最も影響が大きい分類は？
- 現場の開発者は、結果に同意するか
 - debianの開発者に結果を見せてもらう

