



変化点検出とトピック分析を用いた リポジトリマイニング手法の提案

和歌山大学システム工学部

©久木田 雄亮, 柏 祐太郎, 大平 雅雄

背景

ソースコードの規模
ソフトウェアの連携



バグの数



バグ管理の必要性

- 例えば，予期せぬ（工程の手戻りが必要な）バグが発見された時修正にかかる工数が膨大となる
- ソフトウェアの**品質状況**を**随時把握**することが必要

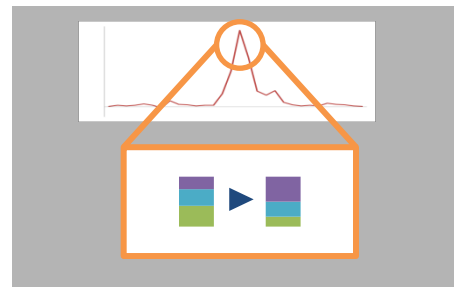
目的・アプローチ

- 目的

- プロジェクトの品質状況の悪化などの悪い方向に向かう変化を早い段階でとらえる手法の構築

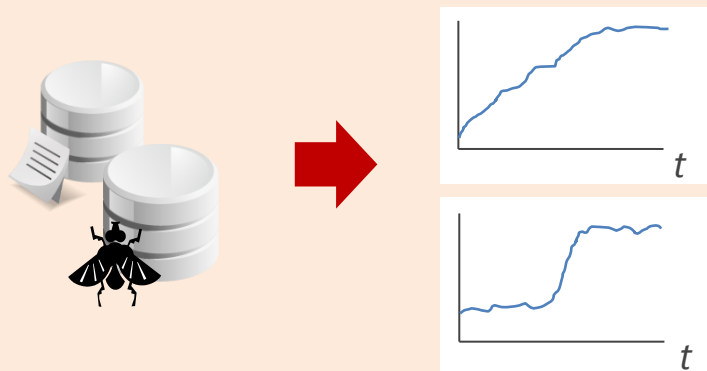
- アプローチ

- 変化点検出を用いてプロジェクトにおける変化をとらえ，その変化がどのような意味を持つ変化であるか理解するためにトピック分析を用いる



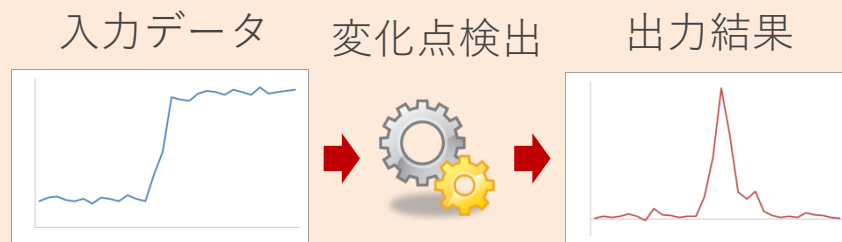
提案手法

メトリクス抽出



1

変化点検出



2

トピック分析



3

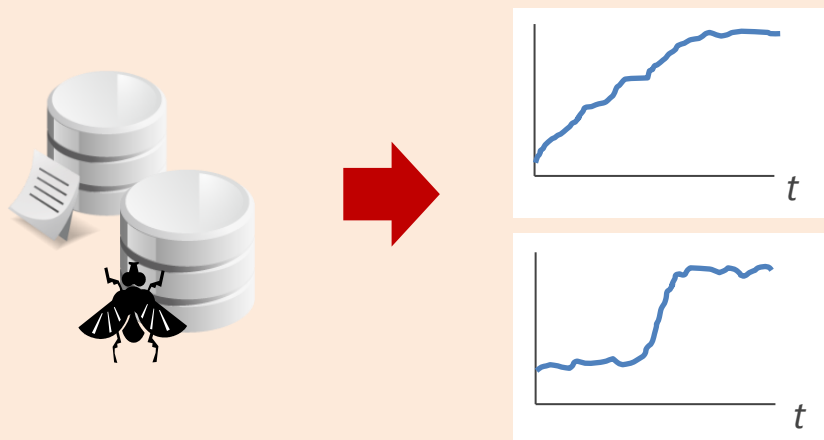
変化点要因の特定



4

提案手法

メトリクス抽出



1

- データソースからメトリクスを抽出し1日毎に集計し、時系列データとしてまとめる（コード行数，バグ報告数など）

メトリクス抽出



変化点検出



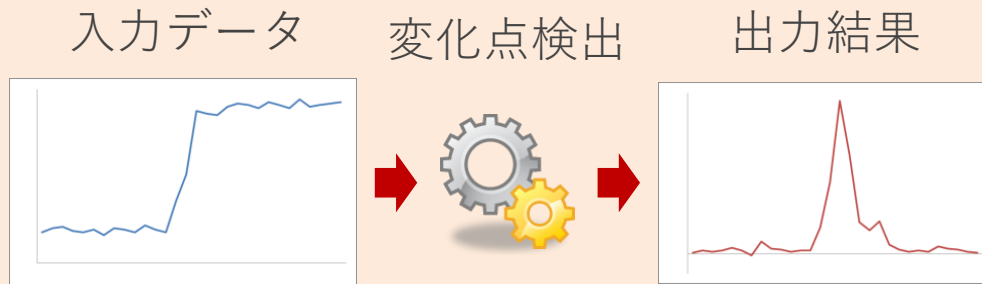
トピック分析



変化点要因の特定

提案手法

変化点検出



2

- メトリクスの時系列データを
入力として、変化点検出を行う
- 時系列モデル（データの分布）の
変わり目が変化点である
- 本研究ではChangeFinderを用いる

メトリクス抽出



変化点検出

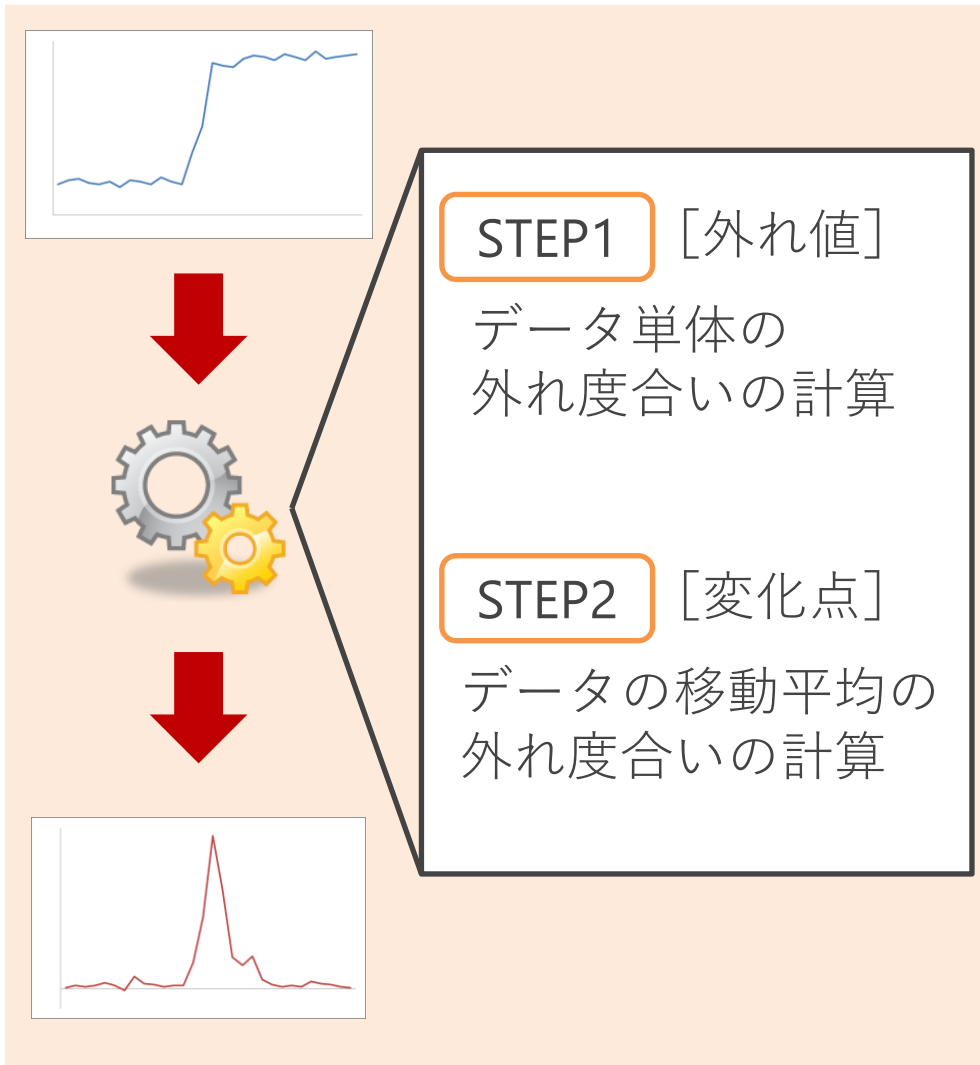


トピック分析



変化点要因の特定

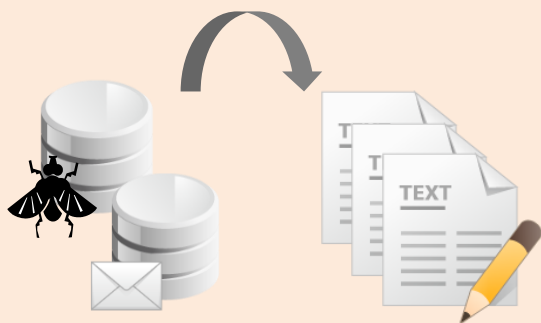
変化点検出のアルゴリズム



- 2段階の
“検出”ステップに分離
- 外れ度合いの計算に
ARモデルを導入
- 様々な分野で
用いられている
 - ネットワークへの
攻撃検知など

提案手法

トピック分析



3

- データソースから取り出したメールやコメントがどのトピックに一番近いかを振り分ける
- 本研究ではトピックモデルの**LDA**を用いる

メトリクス抽出



変化点検出



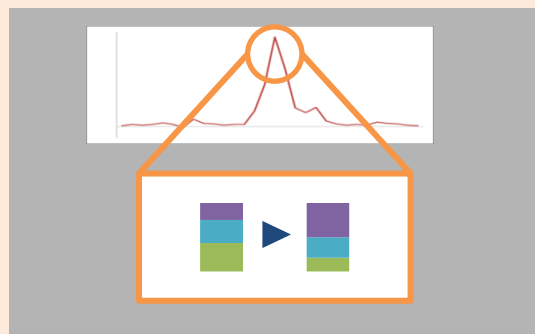
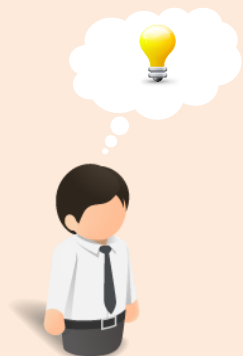
トピック分析



変化点要因の特定

提案手法

変化点要因の特定



4

- 変化点の前後1週間のトピックの変遷を見ることでどのようなことが議論され、変化点の要因になっているか特定する

メトリクス抽出



変化点検出



トピック分析



変化点要因の特定

ケーススタディ

- 対象プロジェクト
 - Eclipse Platform
- 対象データセット
 - 期間：2003年7月～2008年7月

ソースコード



Gitリポジトリ

不具合管理システム



Eclipse Bugzilla

メーリングリスト



Eclipse News Forum

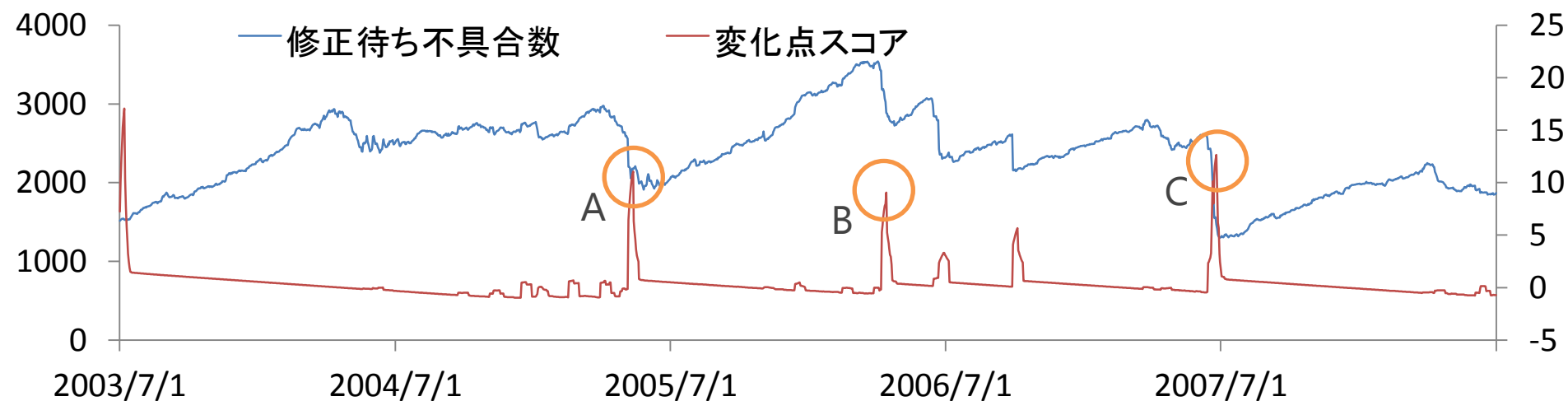
変化点検出で用いたメトリクス

- コード行数 (LOC)
- クラス数
- ファイル数
- メソッド数
- サイクロマティック数の平均
- 修正待ち不具合数
- 不具合報告数
- 不具合解決数

変化点検出で用いたメトリクス

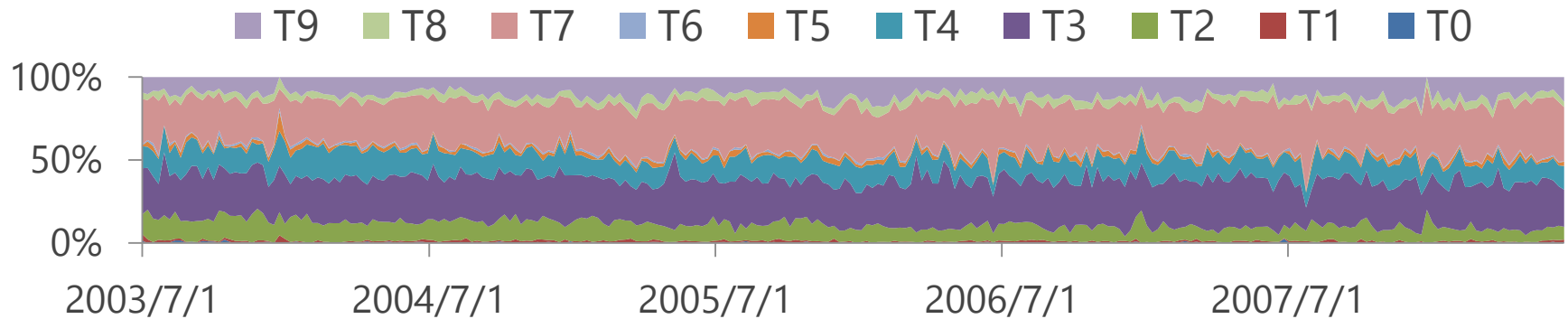
- コード行数 (LOC)
- クラス数
- ファイル数
- メソッド数
- サイクロマティック数の平均
- 修正待ち不具合数
- 不具合報告数
- 不具合解決数

変化点検出結果



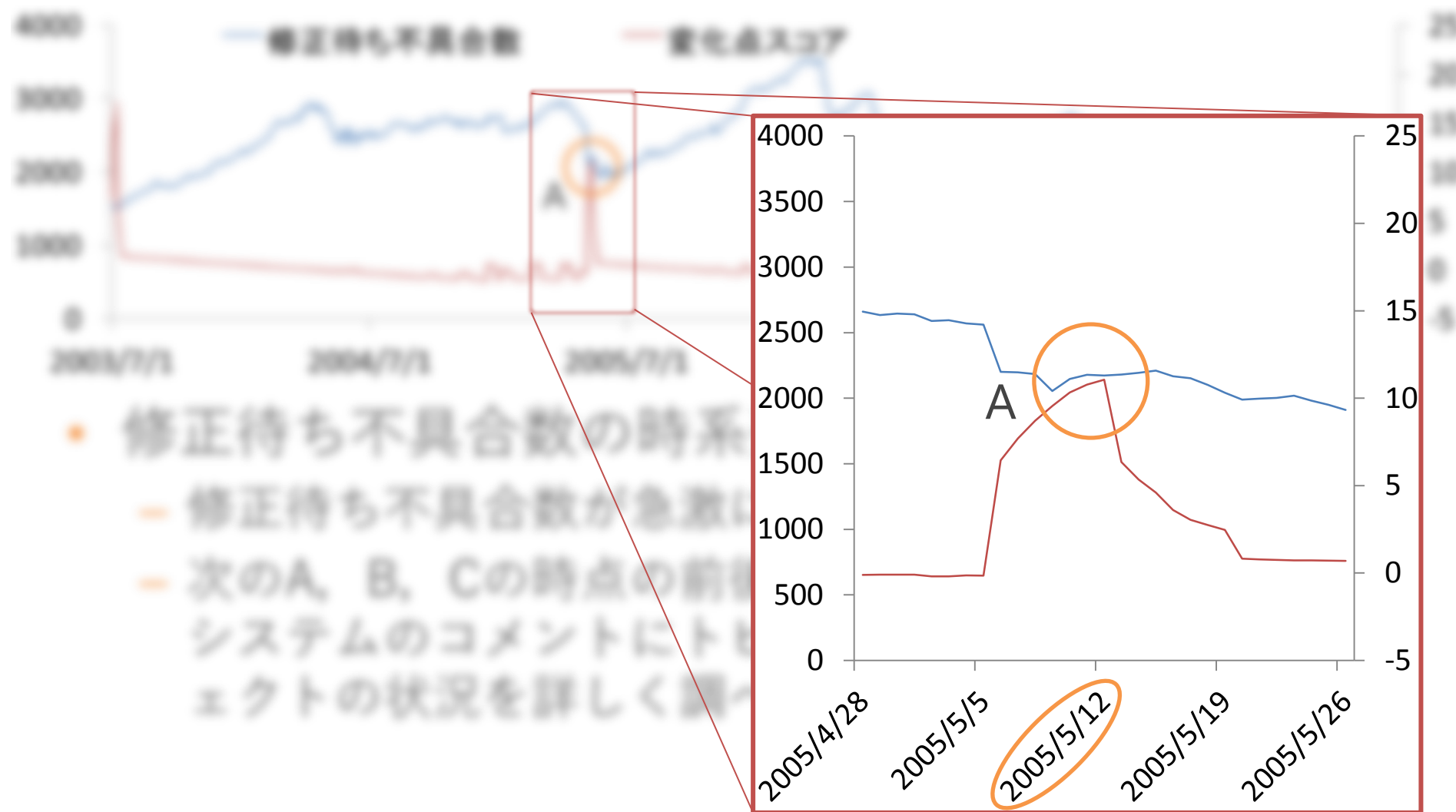
- 修正待ち不具合数の時系列データを使った結果
 - 修正待ち不具合数が**急激に減った時**に変化点が検出
 - A, B, Cの時点に変化点スコアが高い。
前後1週間について、不具合管理システムのコメントにトピック分析を用いて、プロジェクトの状況を詳しく調べる。

トピック分析 | 不具合コメント

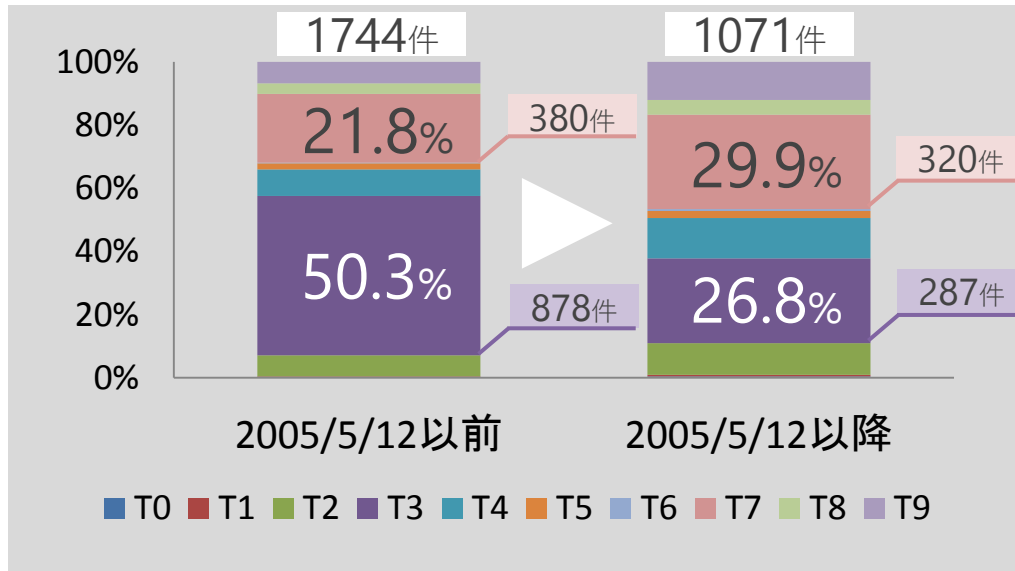


T0	バックグラウンド	line object core java run thread lang internal class worker wait
T1	ライブラリ	lib eclipse java usr gtk jar xp plugins vm opt system swt
T2	ビルド	file project eclipse cvs workspace build error files ant problem java log
T3	バグ・パッチ報告	bug created details attachment duplicate marked patch fixed verified fix head released
T4	UI	view editor text dialog menu window open key user show page problem
T5	イベント	swt shell display int public event void string table import eclipse item
T6	ワークベンチ	eclipse java org internal ui core swt run widgets main workbench jface
T7	修正活動	problem bug eclipse comment swt work time code don build fix gtk
T8	プラグイン	eclipse org ui http update plugins platform core plugin file jar xml
T9	API	api code method class change ui null case add content comment added

変化点検出 [時点A]



トピック分析結果 [時点A]



- 変化点以前は
バグやパッチの
報告・投稿活動
の割合が多い
- 変化点以降は
バグ・パッチ
報告活動の割合が
減り，修正活動の
割合が増える

番号	ラベル	主な単語
T3	バグ・パッチ 報告活動	bug created details attachment duplicate marked patch fixed verified fix head released verify
T7	修正活動	problem bug eclipse comment swt work time code don build fix gtk

考察

- 報告よりも解決しているところが変化点として検出されている

不具合はまとめて解決される傾向にある

- リリースの直前に多くのバグが修正されているということはテストが完璧でなく修正作業の品質が低い可能性がある

今後の課題

- 別の視点でのメトリクスで変化点検出
 - 変化点検出に用いるメトリクスがソースコードやバグと関係の深いものを使っている
- 変化点の発生要因の特定
 - 特定するため分析方法の検討