

# EASE研究の適用事例

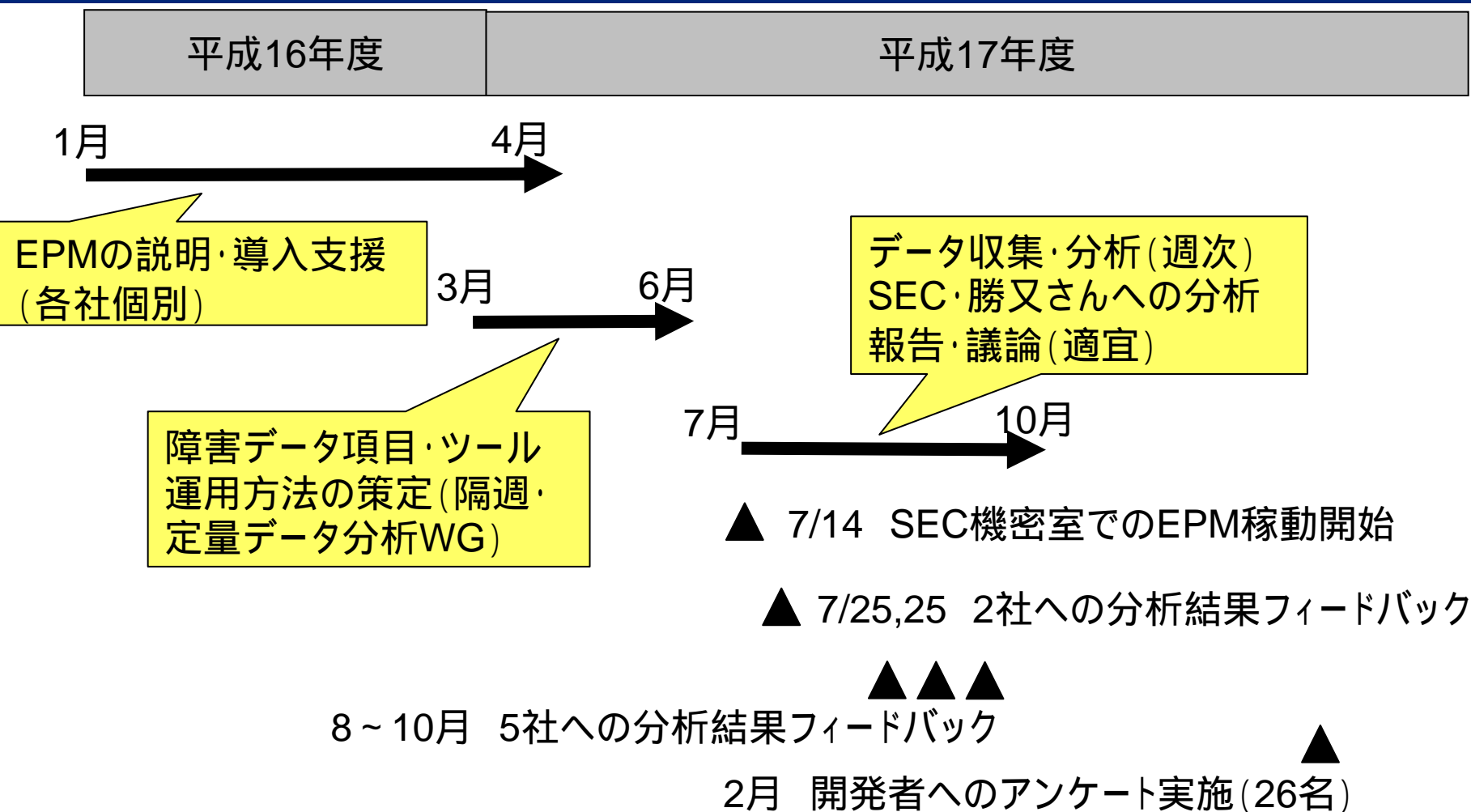
奈良先端科学技術大学院大学  
STAGEプロジェクト研究員 松村知子

エンピリカルソフトウェア工学研究会  
於 キャンパスイノベーションセンター一階 国際会議室  
2007年12月18日

# 発表概要

- EASEのCOSEにおける研究活動の流れ
- 個別研究活動報告
  - コードクローン分析
  - ロジカルカップリングによるファイル間依存関係の分析
  - 協調フィルタリングによる見積り支援
  - マイクロプロセス分析によるプロセス品質の推定
  - 相関ルールマイニング(Needle)による傾向分析
  - 上流工程分析
  - 修正工数分析
  - EPM Pro\*によるインプロセス分析

# 研究活動の流れ(第1フェーズ)



# 分析技術の導入手順

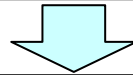
何を目的とするか  
何を知りたいか  
どう役に立つか

何を集めるか

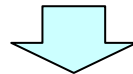
どのように集めるか

何を/誰に/どのよう  
うな手段で/いつ報

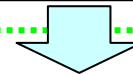
分析案の作成・提示



収集データ項目の決定



ツール運用方法の決定



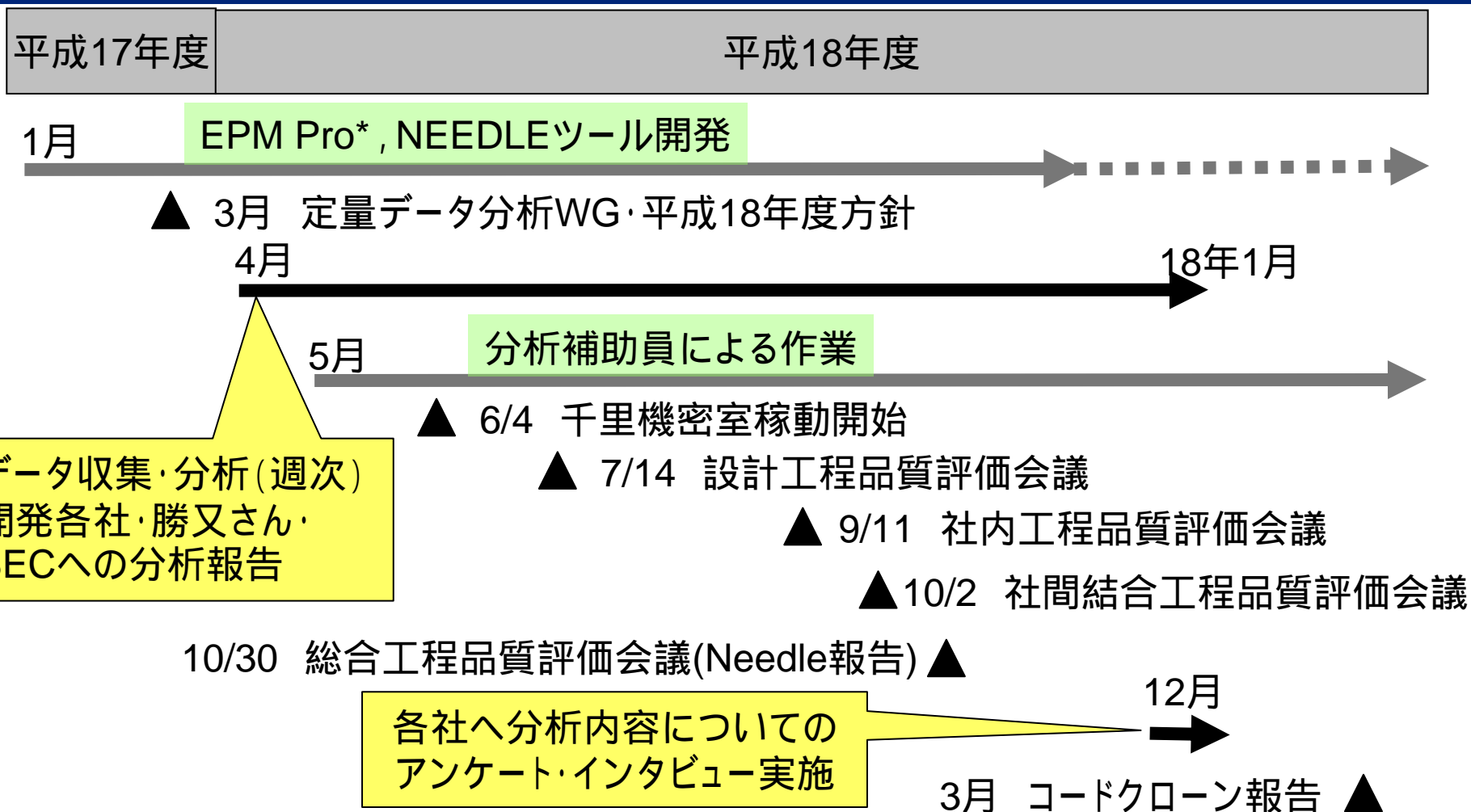
フィードバック方法の  
決定

平成17年度の  
手順

平成18年度の  
手順

構成管理ツール  
障害管理ツール  
etc.

# 研究活動の流れ(第2フェーズ)



# 千里機密室



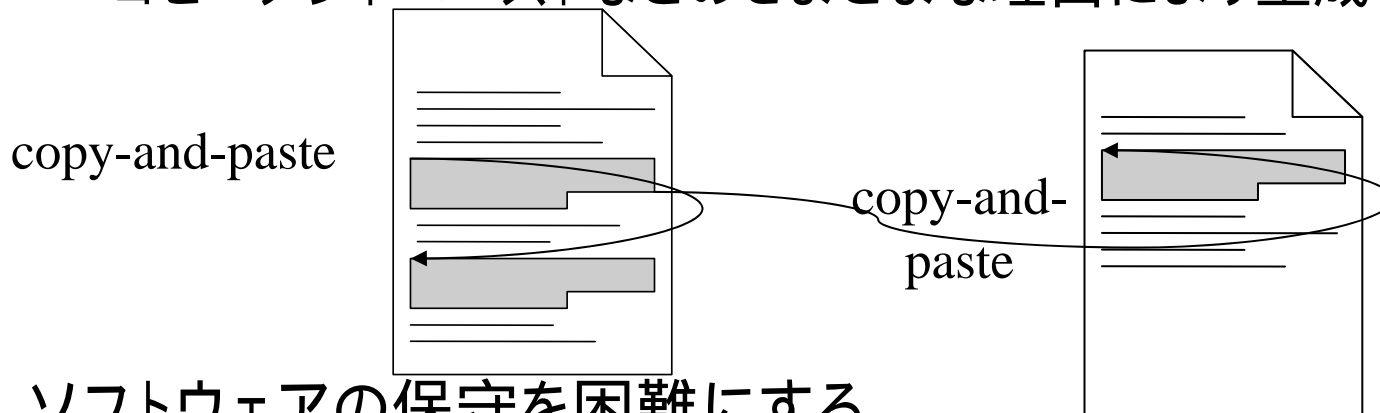
# コードクローン分析

## 出典・参考文献:

- 肥後芳樹, 吉田則裕, 楠本真二, 井上克郎: “産学連携に基づいたコードクローン可視化手法の改良と実装”, 情報処理学会論文誌, Vol.48, No.2, pp. 811-822 2007/2
- 楠本真二: “産学連携による実証的ソフトウェア工学の事例”, SPI ジャパン 2007 講演

# 背景

- コードクローンとは
  - ソースコード中に存在する一致または類似したコード片
  - コピーアンドペーストなどのさまざまな理由により生成される



- ソフトウェアの保守を困難にする
  - あるコード片にバグがあると, そのコードクローン全てについて修正の検討を行う必要がある
  - 機能を追加する場合も同様のことがいえる



# コードクローン分析ツール群

- コードクローン検出・分析ツールの開発
  - クローン検出ツールCCFinder (字句単位のクローン検出)
  - 分析環境ICCA
    - CCFinder をコードクローン検出エンジンとして使用
    - 目的別にサブコンポーネントが存在
      - 可視化, 理解支援 - Gemini コンポーネント
      - リファクタリング支援 - Aries コンポーネント
      - デバッグ支援 - Libra コンポーネント\*
- ツールの適用
  - フリーソフトウェア, 大学の演習
  - 配布先各社・大学等での利用

\*適用事例:

“コードクローン分析を利用した類似バグ検出” パナソニックMSE株式会社 佐々木 健介様 第14回 EASE研究会発表, [http://www.empirical.jp/download/past/publicdata/14th\\_kenkyukai/5.pdf](http://www.empirical.jp/download/past/publicdata/14th_kenkyukai/5.pdf)

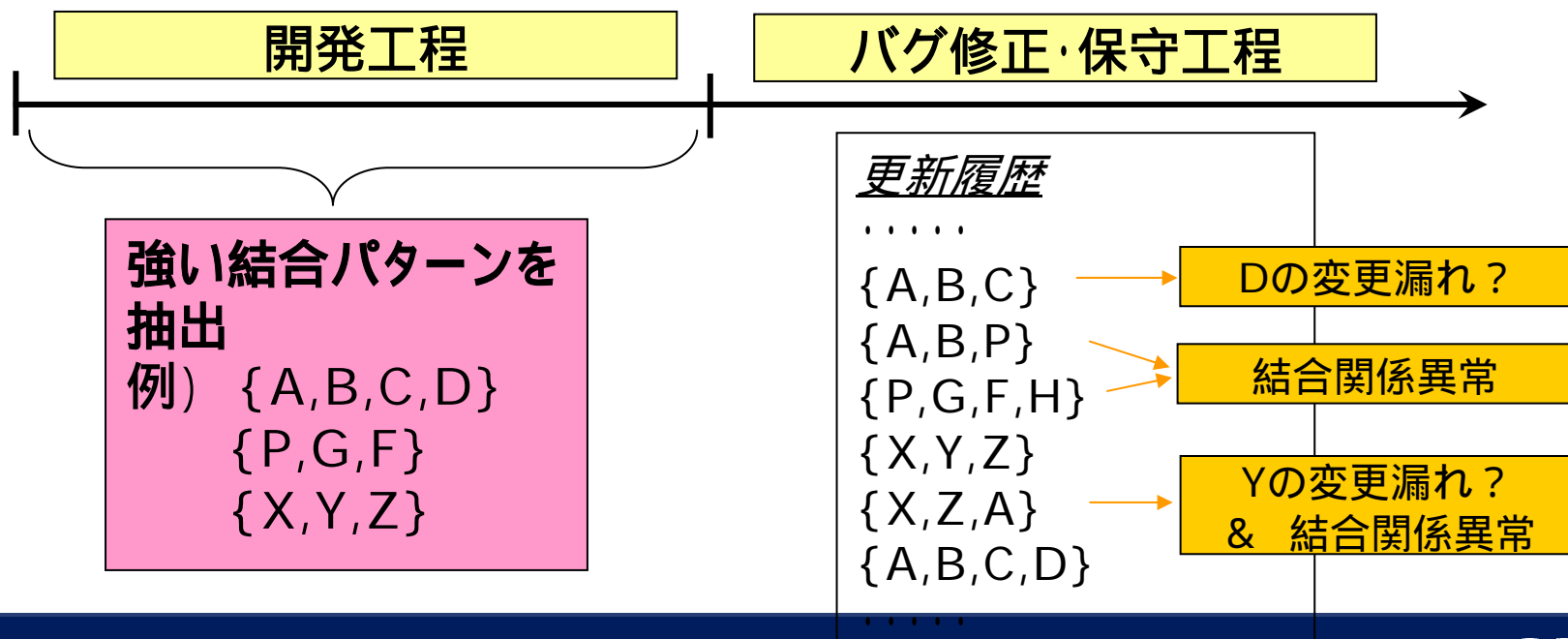
# ロジカルカップリングによる ファイル間依存関係の分析

## 出典・参考文献:

松村 知子, 横森 励士, 大杉 直樹, 川口 真司, 松下 誠: "ファイルの同時変更パターンと変更差分の分析による論理的結合関係の自動抽出", ソフトウェアシンポジウム2005論文集, 2005/6/9

# ロジカルカップリングの活用

- ファイルの更新履歴から同時更新関係を抽出し、同時更新回数からファイル間の関係を推定する
- 抽出された関係を用いて、ファイル変更時の修正漏れ・フォールトを検出する



# ロジカルカップリング分析結果例

信頼度	前提事象 発生数	結果事象 発生数	前提事象		結果事象
0.8	5	4	FileA.cpp FileB.h FileC.cpp	->	FileD.h
1	4	4	FileA.cpp FileB.h FileD.h	->	FileC.cpp
1	4	4	FileA.cpp FileC.cpp FileD.h	->	FileB.h
1	4	4	FileB.h FileC.cpp FileD.h	->	FileA.cpp

FileD.hが一回修正漏れの可能性はある

FileY.cppは, FileX.cppのコピーで作成されたか, FileY.cppはFileX.cppと強い依存関係にある.

信頼度	前提事象 発生数	結果事象 発生数	前提事象		結果事象
0.5	8	4	FileX.cpp	->	FileY.cpp
1	4	4	FileY.cpp	->	FileX.cpp

# 協調フィルタリングによる見積り支援

## 出典・参考文献:

- 大杉 直樹, 角田 雅照, 門田 暁人, 松村 知子, 松本 健一, 菊地 奈穂美: "企業横断的収集データに基づくソフトウェア開発プロジェクトの工数見積もり", SEC journal, No.5, pp.16--25, February, 2006.
- 角田 雅照, 大杉 直樹, 門田 暁人, 松本 健一, 佐藤 慎一: "協調フィルタリングを用いたソフトウェア開発工数予測方法", 情報処理学会論文誌, Vol.46, No.5, pp. 1155-1164, May, 2005.
- Naoki Ohsugi, Masateru Tsunoda, Akito Monden, Ken-ichi Matsumoto: "Effort Estimation Based on Collaborative Filtering", Proceeding of the 5th International Conference on Product Focused Software Process Improvement (PROFES2004), April, 2004.

# 協調フィルタリングによる見積りステップ

- ステップ1: 類似度計算

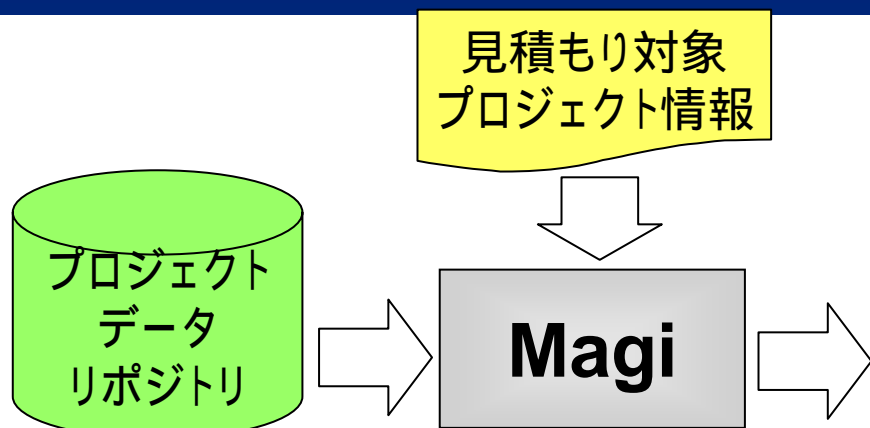
- 現行プロジェクトと過去プロジェクト間の類似度を計算する。
- 類似度の高い  $k$  (例えば  $k = 2$ ) 個のプロジェクトを選ぶ。

- ステップ2: 予測値計算

- 類似プロジェクトの工数を加重平均し, 現行プロジェクトの工数を予測する。

	開発言語	開発種別	概算 FP	要員数	開発総工数
現行プロジェクト X	Java	新規	1500	10	予測値: 402.5
類似度: +1.0 プロジェクト 1	Java	新規	欠損値	8	400
類似度: +0.9 プロジェクト 2	Java	欠損値	2500	6	350
類似度: -1.0 プロジェクト 3	欠損値	保守	3000	20	2500

# ワンクリック見積り&データ品質診断ツール Magi

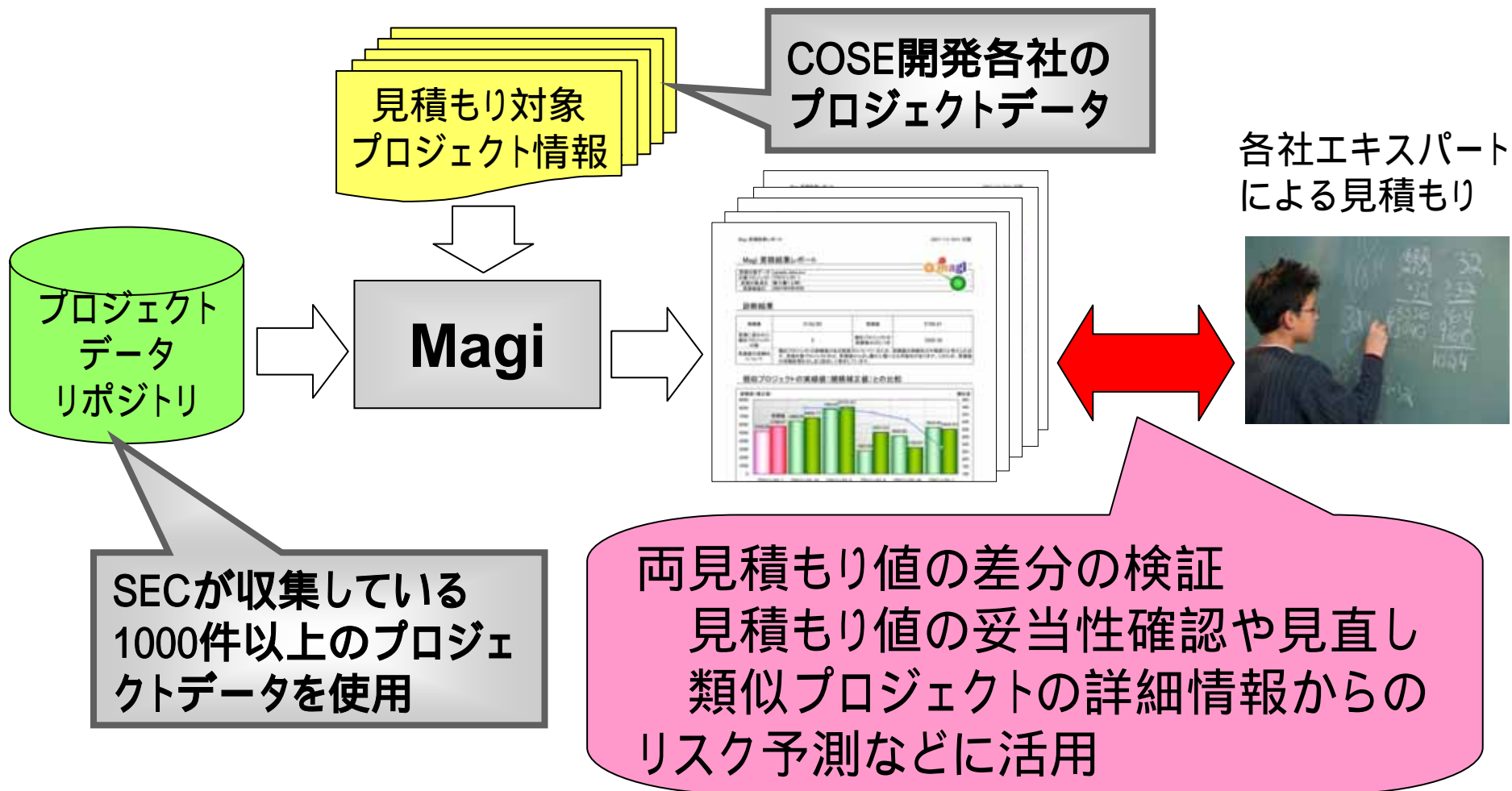


欠損値が多くても、精度の高い結果が期待できる  
 既存のデータベースを用いることができる

類似プロジェクトを具体的に示す  
 見積もりの根拠がわかる  
 類似プロジェクトからさらに詳細な情報を収集することができる



# COSEでの適用例





# マイクロプロセス分析による プロセス品質の推定

## 出典・参考文献:

- 森崎 修司, 松村 知子, 大蔵 君治, 伏田 享平, 川口 真司, 飯田 元, “エンピリカルデータを対象としたマイクロプロセス分析,” 情報処理学会研究報告, Vol.2006, No.125, pp.9-15, November 2006.
- Shuji Morisaki, Tomoko Matsumura, Kimiharu Ohkura, Kyohei Fushida, Shinji Kawaguchi, Hajimu Iida, "Fine-grained Software Process Analysis to Ongoing Distributed Software Development", In Proc. Workshop on Measurement-based Cockpits for Distributed Software and Systems Engineering Projects, pp. 26-30 (Aug. 2007)

# マイクロプロセス分析の背景と目的

- 背景

- ソフトウェアプロダクトから品質予測をするのと同様にプロセスからソフトウェアの品質を予測したい
- 「正しいプロセスで開発されていれば、プロダクトの品質も良く、プロジェクトの遅延や工数超過のリスクが小さい」

- 目的

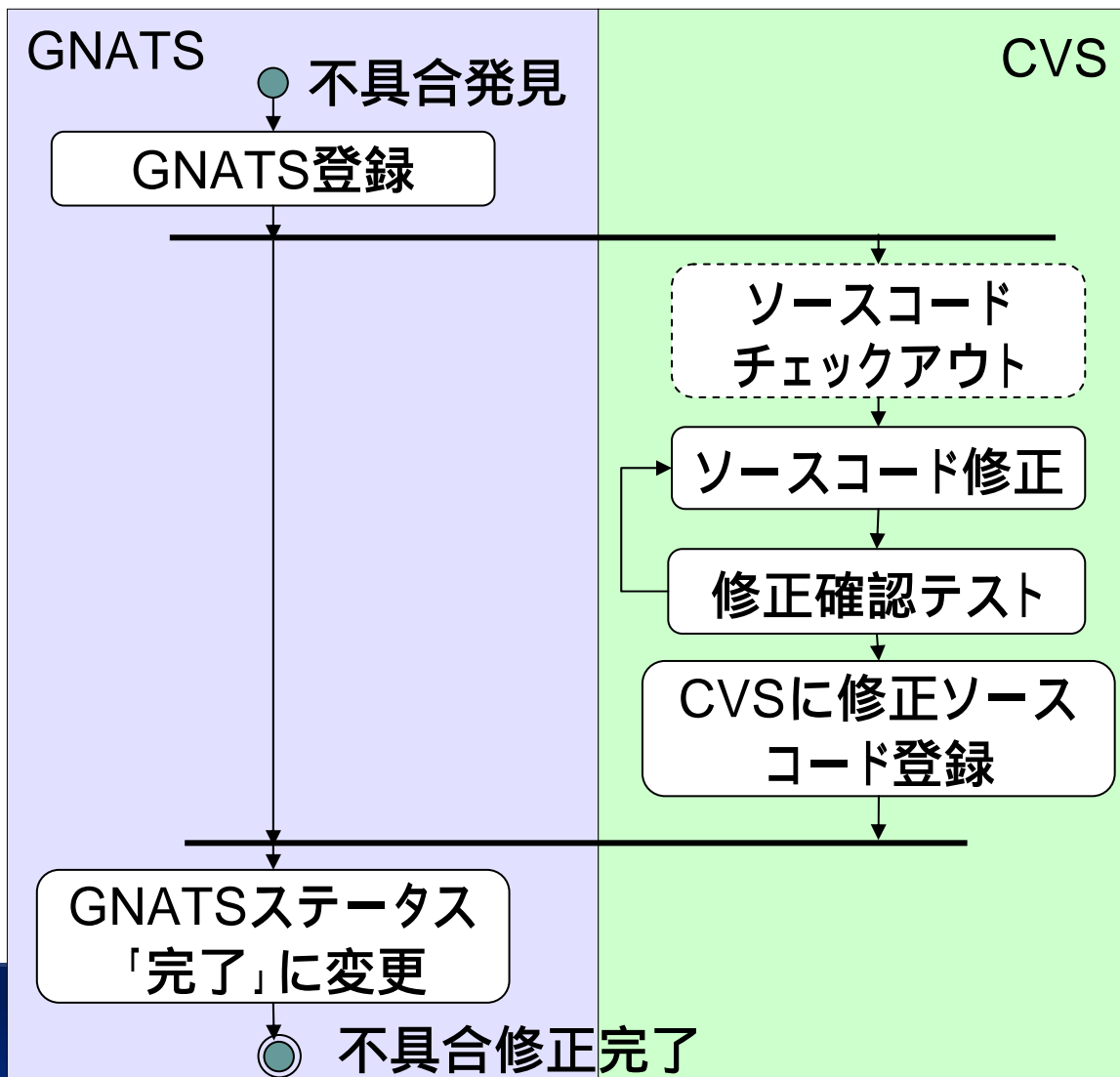
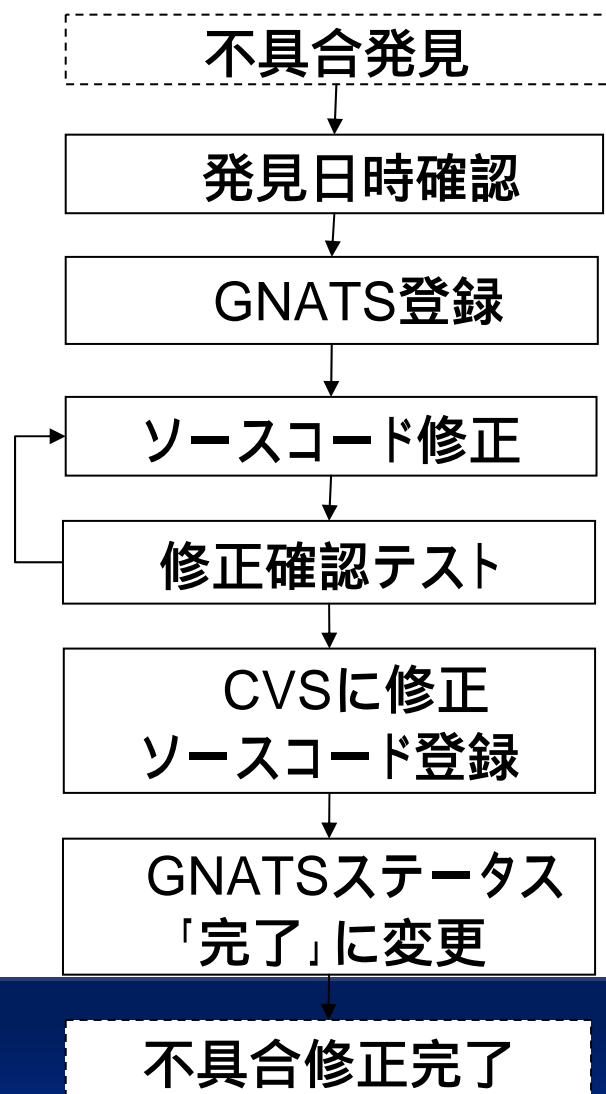
- 定量的な指標を自動計測することによりプロセスの品質を測る(マイクロプロセス分析)

作業の順序, 時間, 作業量

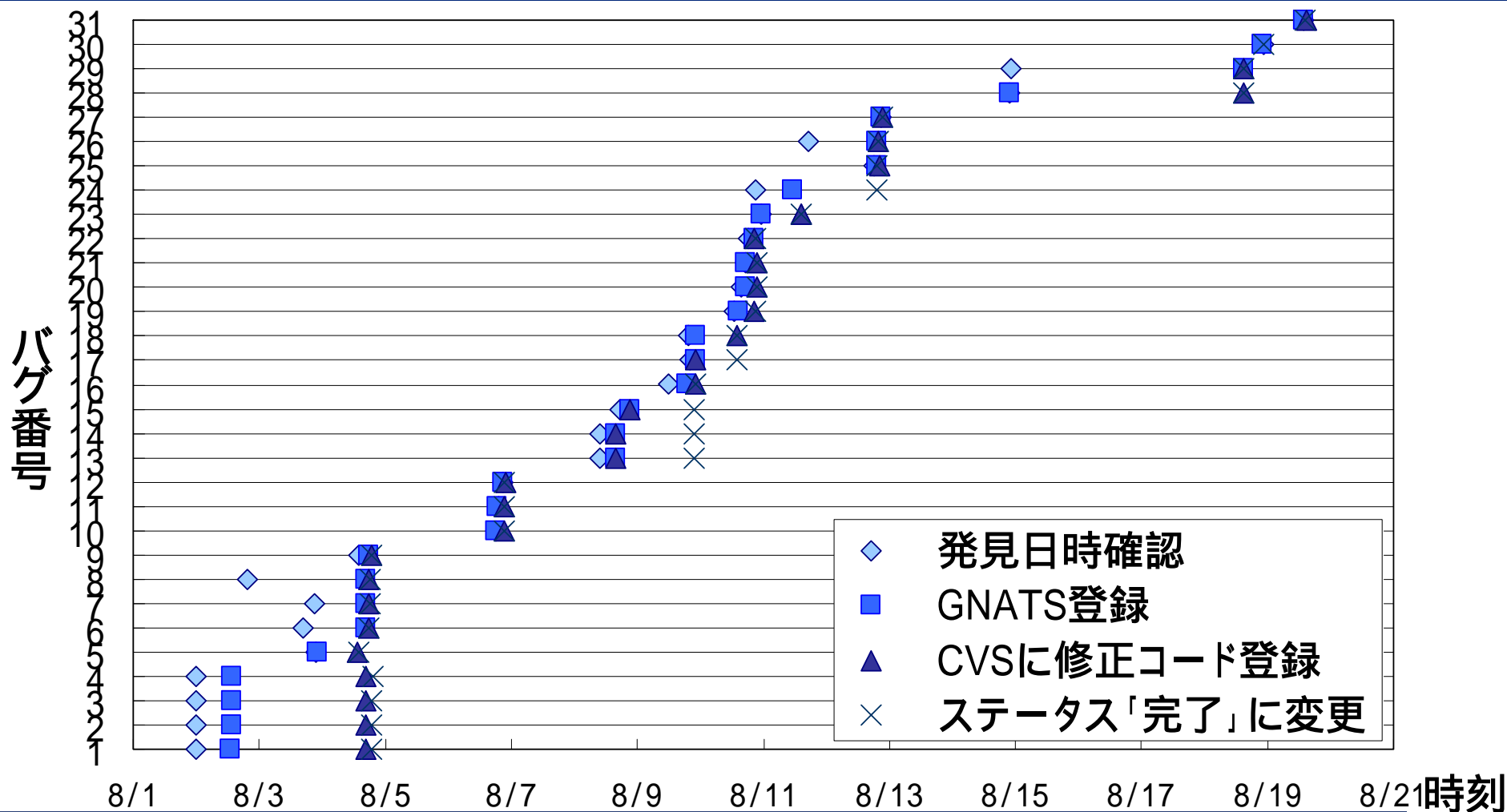
# 対象プロセスモデル

開発メンバーにお願いした手順

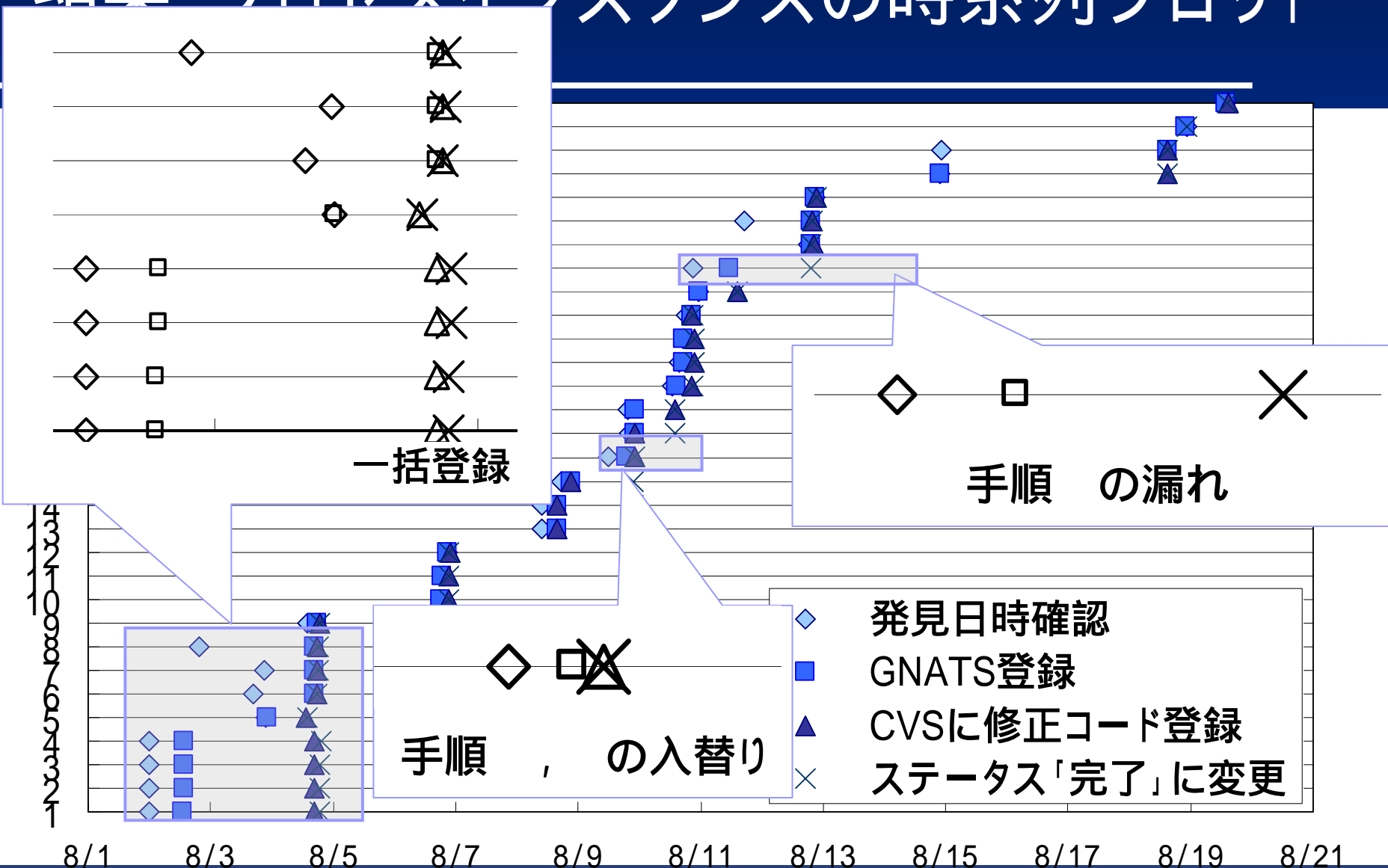
想定プロセスモデル(アクティビティ図)



# 結果：プロセスインスタンスの時系列プロット



# 結果・プロセスインスタンスの時系列プロット



# 相関ルールマイニング(Needle)による 傾向分析

## 出典・参考文献:

- 森崎 修司: "相関ルール分析 NEEDLEのバグ票とプロジェクトデータへの適用", 第14回EASE研究会, 2007/10, [http://www.empirical.jp/download/past/publicdata/14th\\_kenkyukai/3.pdf](http://www.empirical.jp/download/past/publicdata/14th_kenkyukai/3.pdf)
- 森崎, 門田, 玉田, 松村, 松本: "Defect Data Analysis Based on Extended Association Rule Mining", Proceedings of International Workshop on Mining Software Repository, pp.17-24. <http://se.naist.jp/~morisaki/publications/#i-200705>
- Shuji Morisaki, Akito Monden, Haruaki Tamada, Tomoko Matsumura, and Ken-ichi Matsumoto, "Mining Quantitative Rules in A Software Project Data Set," IPSJ Journal, Vol.48, No.8, pp.2725-2734, August 2007.

# 相関ルールとは？

- 対象データに含まれる「A(前提部)ならばB(結論部)」という関係の強い規則(ルール)
- 列挙されたルールから解釈を与えることができるルールを人手により探し、役立てる
- 事例) コンビニの購買履歴から得た相関ルール
  - 休日に「レジャーシート」を買う顧客は、「おにぎり」と「お茶」も同時に買っている
  - 「(曜日 = 土日) and おにぎり and お茶」 「レジャーシート」
  - 休日には、レジャーシートの配置をおにぎりかお茶に近づけ、発見率、併せ買い率を上げる

# 相関ルール分析ツール: NEEDLE

- ソフトウェア開発プロジェクトの特性をまとめたデータ(コスト, プロフィール, バグ)から規則性や傾向, 例外を, アソシエーション分析手法を用いて抽出する機能を実現
- 特長
  - データの自動前処理プログラム
    - データ種類の判別, 数値データの分割
  - 結論部(「AならばB」のBの項目)を固定にできる
  - 量的ルール, 例外ルールを抽出できる



# 障害情報と修正工数の分析例(量的ルール)

