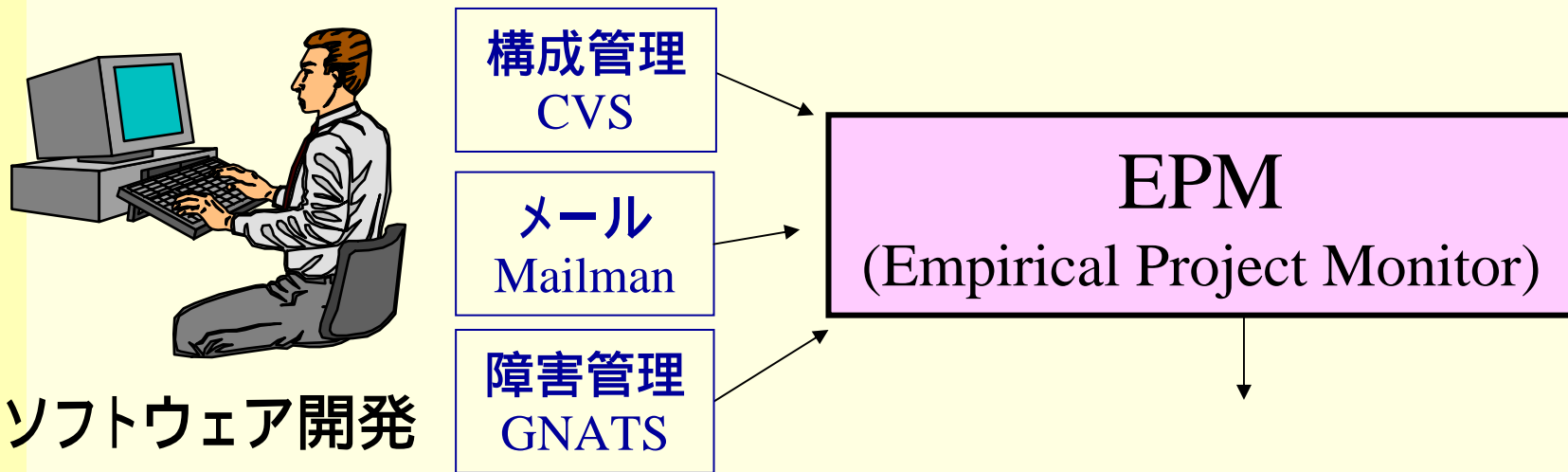


# EPMによるデータ収集・ GQMによる分析の事例報告

奈良先端科学技術大学院大学  
情報科学研究科  
門田暁人  
akito-m@is.naist.jp

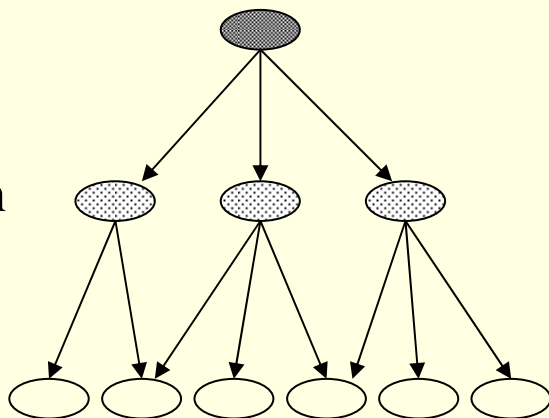
# EPMによる収集データとGQMモデル



Goal

Question

Metric



## 1日ごとのプロジェクトサマリ

- ・ソースコード総行数
- ・総ファイル数
- ・変更者がN人以上のファイル数
- ・M行以上修正されたファイル数
- ・障害発生数(種類毎, 累積)
- ・対処済み障害数
- ・障害滞留時間
- ・ . . . . .

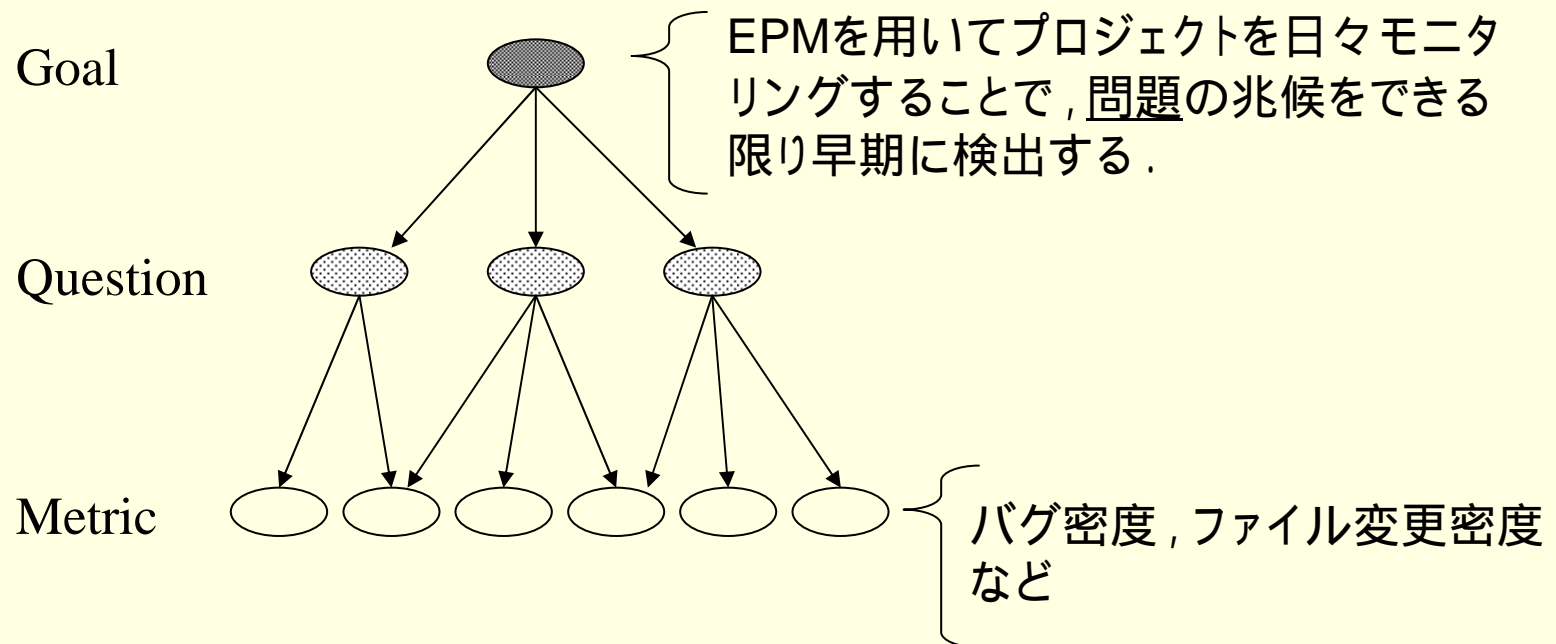
GQMモデルによるデータの解釈

# 収集データと得られる情報

- 構成管理ツール(CVS)
  - 行の追加 …… 開発が進んでいる or 停滞している.  
1人日あたりの行数は？クローンは？
  - 行の削除 …… 仕様変更, バグなど. 規模が大きいものは要注意
  - ファイル変更者 …… 何人が同一ファイルに触っているか.  
急に増えると要注意.
  - 同時変更ファイル …… ファイル間の関連の把握 など
- 障害管理ツール(GNATS)
  - 上流工程(設計書レビュー)から下流(テスト)まで
  - 対応状況 …… 累積障害, 残存障害, 障害密度, ……
  - 混入工程 …… 仕様バグ, 設計バグ, コーディングバグ
  - 原因 …… ロジック, インタフェース, ……
  - 障害箇所 …… どのバージョンのモジュールか
  - 修正工数 …… 人時 など

# GQMモデル

- **ゴール指向の分析アプローチ**
  - 1984年にBasiliとWeissによって提唱された。
  - 目的を明確化し, 収集データやメトリクスと**対応付ける**。



# これまでの経緯

---

- 2004年10月：第3回EASE研究会にて、分析のゴールについてのアンケート調査
  - プロジェクト遅延の原因、遅延の兆候の調査
  - プロジェクト進捗管理における課題の調査
- 2004年11月：EASEプロジェクトアドバイザー Basili教授を講師に迎え、2日間のワークショップ開催
  - EASEプロジェクト向けの初期GQMモデルを5つ作成した。
  - 12月：第4回EASE研究会でのアンケート調査
- 2005年春～夏：実プロジェクトへの適用
  - データ収集、現場へのフィードバック、GQMモデルの改良
- 2005年秋～2006年冬：
  - 改良したGQMモデルに基づく分析、現場へのフィードバック

# 初期のGQMモデルの例1

## ■ Goal

- バグに関する構成管理データを用いて、ソースコードの品質を、マネージャの視点から評価する。

## ■ Question

- 重要度の高いバグの修正数の曲線は？
- 優先度が高いバグが未解決となっている時間の平均は？

## ■ Metrics (Model)

- 発生バグ(重要度大)の累積曲線  $>> 0$  ならばソースコードの品質が悪い。
- 未解決バグ(重要度大)の累積曲線  $> 0$  ならばプロジェクトでトラブルが発生している。(人員不足)
- 未解決バグ(優先度大)の滞留日数  $> 0$  ならばプロジェクト管理に問題がある。(タスク割り当ての問題)

# 初期のGQMモデルの例2

## ■ Goal

- CVSとGNATSからファイル変更パターンを解析し、**要求の不安定さ**、**設計の不完全さ**、**ソースコードの品質**を、マネージャの視点から評価する。

## ■ Question

- ファイル変更レベル(FCL)は？
- 変更行数(LCC)は？
- 発生した障害の種類(バグ or 変更要求)は？
- 各ファイルの変更者数(Owner)は？

## ■ Metrics (Model)

- $FCL > 0.5$  かつ  $LCC/\text{file size} > 5\%$  かつ 全障害数に占める変更要求が60%以上 ならば **要求が不安定**
- $FCL > 0.5$  かつ  $(LCC/\text{file size} > 5\%$  かつ 変更者数 2のファイルの割合が25%以上 ならば **設計が不完全**
- $FCL > 0.5$  かつ 1KLOCあたりのバグ数が10以上ならば **ソースコード品質が悪い**

# 初期のGQMモデルの改善すべき点


- Goal が大きすぎる (複数のゴールが混在している) .
- G-Q-Mの間のつながりが明確でない .
  - 取ってつけたようなQuestionがある .
- モデルの背後にある「**仮説**」が不明確
- モデル式に出てくるベースライン値を決めることが難しい .
- 企業からのコメント :
  - プロジェクト全体でのデータだけでなく , モジュール単位でのデータも見たい .
  - 密度データ (バグ密度など) の推移のグラフだけでは不十分 . 元データ (規模 & バグ数) の推移も見たい .

## 改善策

- より小さく具体的なGoalを設定する .
- モデルの背後にある仮説を明確にする .
- 値の変化に着目する .
- モジュール単位でも分析する .

# Goalの明確化

- EPMを用いてプロジェクトを随時モニタリングすることで、問題の兆候をできる限り早期に検出する。

- 
- ・要求が不安定
  - ・設計が不完全
  - ・リソース配置が不適切
  - ・コーディング品質が悪い

これら4つの問題の兆候を検出するためのGQMモデルをそれぞれ作成した。

# Model 1 ... 5

---

- Model 1: 要求の不安定さの評価
- Model 2: 設計の不完全さの評価
- Model 3: リソース配置の評価
- Model 4: コーディング品質

# 分析例 1 - 5

---

- 事例 1 : 不安定な仕様
- 事例 2 : 頻繁なソースコード変更
- 事例 3 : 人手不足
- 事例 4 : 要追加テスト
- 事例 5 : 開発要員の変更

# まとめ

---

- 改良したGQMモデルによるEPMデータの分析事例の紹介

- その他, データから分かること

- CVSとGNATSのデータの不整合 (GNATSでバグ登録されてるけどCVS上での記録がないなど)