

# ソフトウェア開発データに基づく プロジェクト管理

－ エンピリカルソフトウェア工学とEASEプロジェクト－

文部科学省StagEプロジェクト研究代表  
奈良先端科学技術大学院大学  
ソフトウェア工学講座  
松本健一

# あらまし

- 背景
- エンピリカルソフトウェア工学の基本概念と3つの段階
- 文部科学省EASEプロジェクト  
(2003年4月～2008年3月)

# 背景

# CHAOS Report in US

- 1994年 [http://www.standishgroup.com/sample\\_research/](http://www.standishgroup.com/sample_research/)
  - プロジェクト成功率 16.2%  
(大企業では9%)
  - プロジェクトキャンセル率 31.1%
  - 残り52.7%では平均89%の開発費超過
- 2003年
  - プロジェクト成功率 34%
  - 米国ソフトウェア開発費3820億ドルのうち、プロジェクトマネジメントの失敗による損失は820億ドル。

# 日本におけるプロジェクト成功率

観点 \ プロジェクト種別	ソフトウェア開発全般*	組込みソフトウェア開発**
品質	46%	71%
コスト	76%	43%
納期	55%	36%

\* “特集: プロジェクト成功率は26.7%”, 日経コンピュータ, 2003年11月17日号, pp.50-62, 2003.

\*\* 2006年版組込みソフトウェア産業実態調査報告書, 経済産業省, 2006.

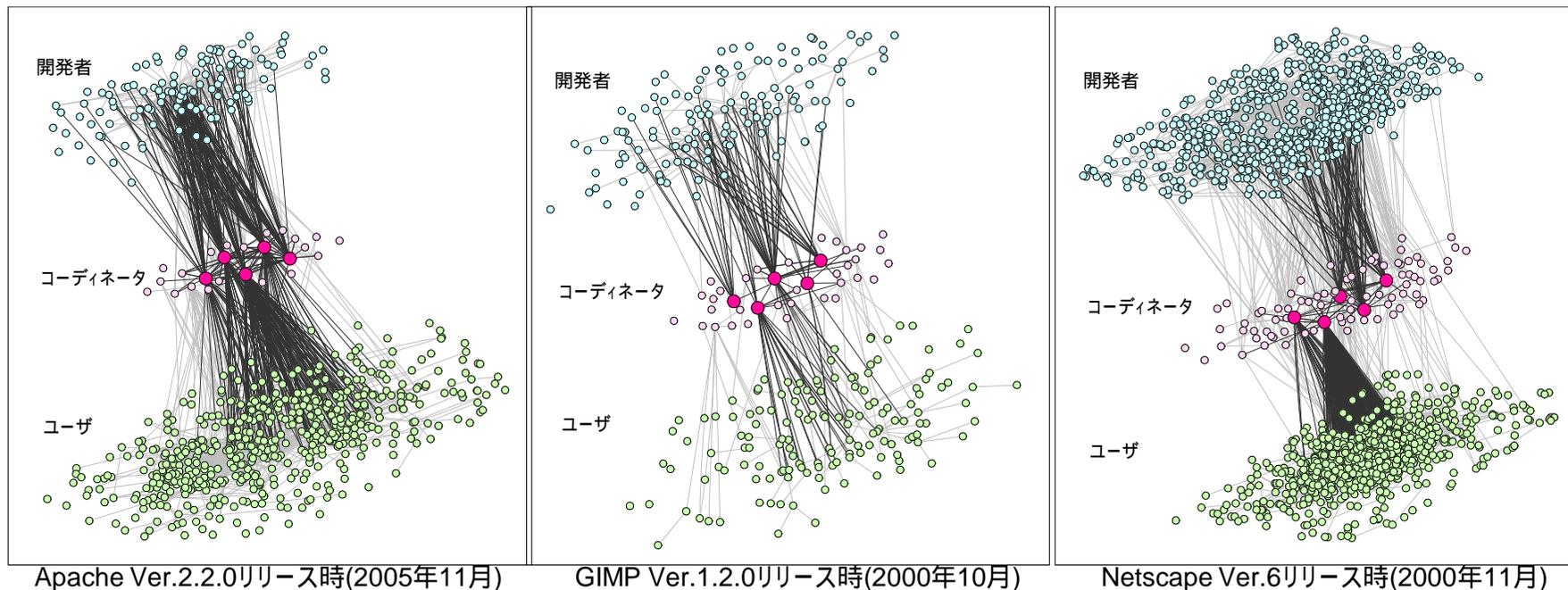
# 情報サービス・ソフトウェア産業維新

- 企業情報システムや組込み系のソフトウェアの信頼性・生産性の向上という観点からは、プロジェクトの**計画、実装、試験、運用段階における定量的な運営手法等のソフトウェアエンジニアリングに関する実証的な研究**を産学官が連携して進めるとともに、その普及に努めることが重要である。
- ソフトウェア開発の信頼性と生産性を高める新しい手法の現場への導入を促進するためには、理論やガイドラインのみでは十分ではなく、**実際の現場へ適用して成功例を示すこと**や、**現場からのフィードバック**を得て、さらに理論等を改善していくことが重要である。

経済産業省・産業構造審議会・情報経済分科会，情報サービス・ソフトウェア小委員会中間とりまとめ，63ページ，2006年9月．

ソフトウェアエンジニアリングに関する実証的な研究例

# 開発時コミュニケーションの可視化例



開発者とユーザ, 双方のコミュニティと  
バランスよくコミュニケーションをとって  
いるコーディネーターが多い.

開発者がユーザ, どちらか一方のコ  
ミュニティとしかコミュニケーションを  
とっていないコーディネーターが多い.

# バグ対応工数の削減事例

## 推測される「工数不足の原因」

- ・原因不明のまま長期間、延々と調べている。
- ・人による能力差が大きい。
- ・振り分けられたバグに他パートのバグが多い。
- ・他パートのバグをすぐに移管せず深追いしている。



## 実績データの収集(数日間)

- ・バグ対応の工数(個人別)。
- ・他パートに移管したものを含めた対応件数。



## 分析結果

- ・平均バグ対応時間 22時間/件。
- ・個人差が大きく、特に若手が長期間悩んでいる。
- ・約40%が他パートに移管されている。



## 改善策

- ・バグ対応作業を5つのフェーズに分割。
- ・2時間を上限としフェーズ毎にリーダーに報告。
- ...



## 成果

- ・平均バグ対応時間の向上  
22時間/件 → 7.1時間/件
- ・**残業時間の減少**  
**2/3以下に**
- ・開発者が効果を実感し、改善意識が向上
- ・バグ対応手法の教育効果
- ・活動の見える化 / 協力意識
- ・他パートへの移管速度向上
- ・新たな改善へのトライ

川崎 雅弘, “メトリクス活用による現場のプロセス改善～バグ対応工数の削減事例～”, ソフトウェア  
プロセス改善カンファレンス2007 (SPI Japan 2007), 2007年10月.

エンピリカルソフトウェア工学

# 基本概念と3つの段階

# ソフトウェア工学

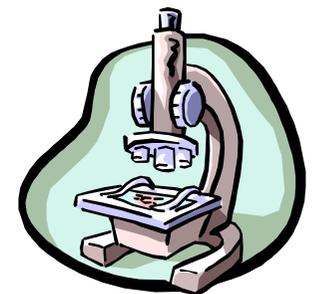
The application of a systematic, disciplined, quantifiable approach to the **development**, **operation**, and **maintenance** of software.

*IEEE Standard Glossary of Software Engineering Terminology*, IEEE Std. 610.12-1990, IEEE, New York, 1990.

ソフトウェアの開発, 運用(利用), 保守に関する理論(モデル, 方法論)を応用して, ソフトウェアの生産性や品質を向上させる技術の総称

# エンピリカルソフトウェア工学

- ソフトウェア工学における, 実証性の概念を前提とするアプローチ.
- **測定データ(実証・実績データ)に基づいてソフトウェアの生産性や信頼性の向上を目指す.**
- Empirical=**Experimental**  
(大学などでのソフトウェア開発実験)  
+**Experienced**  
(ソフトウェア開発現場での経験や実績)



Journal by Kluwer

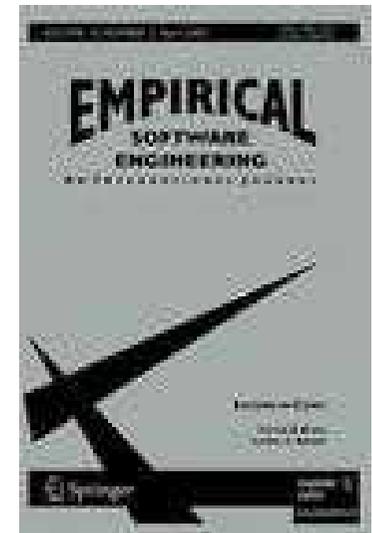
# Empirical Software Engineering

- Topics

- analysis and design methods
- verification and validation methods
- quality assurance and project management
- organization models of software development
- predictive models for software dependability
- cost estimation techniques
- AI techniques to software engineering
- measurement theory, experimental design, qualitative modeling and analysis approaches.

- Impact Factor

- 0.966 (2005)



# ICSE2008 投稿の多かった研究テーマ

1. Program Testing and Analysis
  2. Tools
  3. Software Architecture and Design
  4. Empirical Software Engineering
- ...



# エンピリカルアプローチの3段階

## 第1段階: 観察 (Empirical observations) の実施

- ・実験や調査により現象(事実)を確認する.
- ・現象を**表現**できるようになる.



## 第2段階: 法則 (Laws) の発見

- ・現象が起こるコンテキスト(状況)を理解する.
- ・現象を**予測**できるようになる.

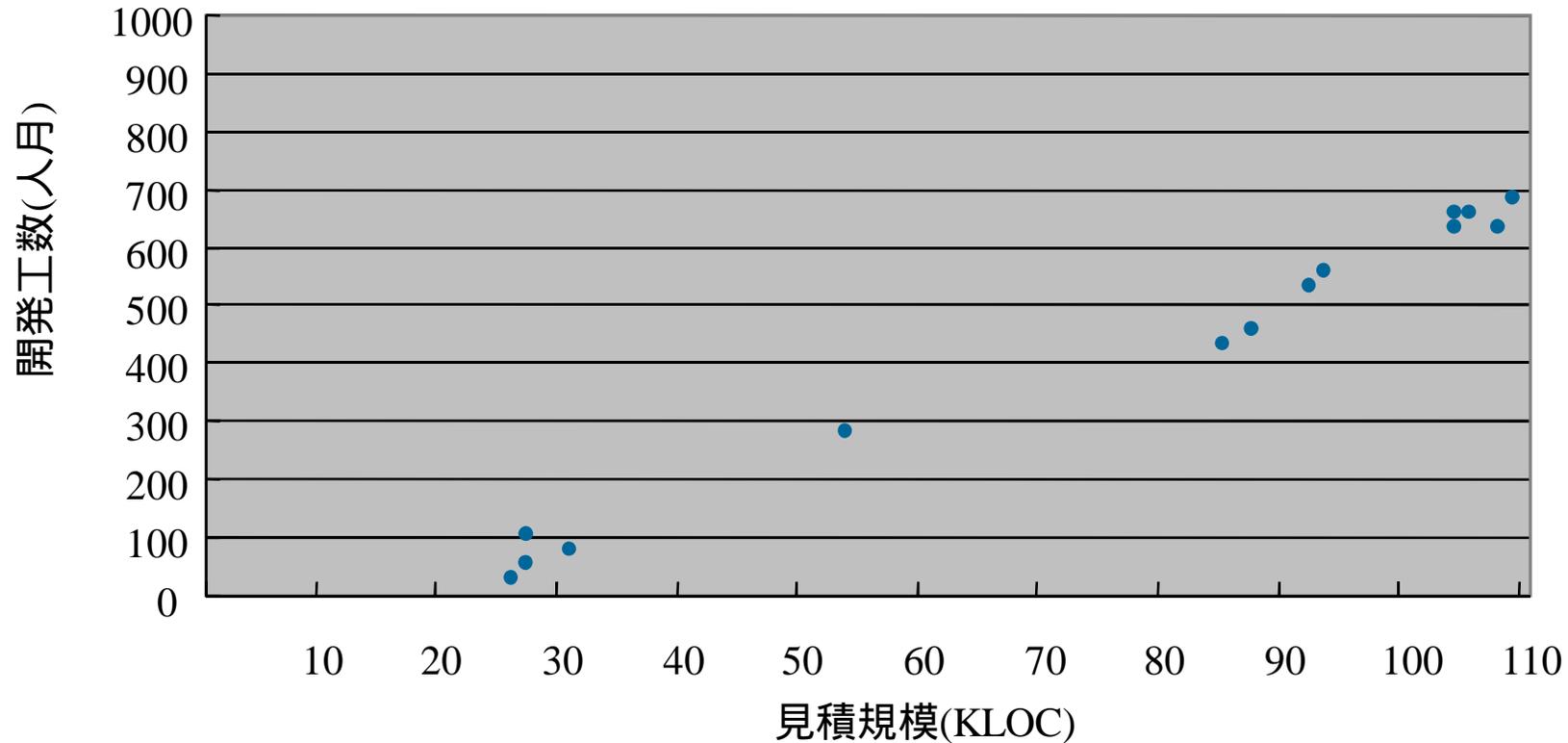


## 第3段階: 理論 (Theories) の確立

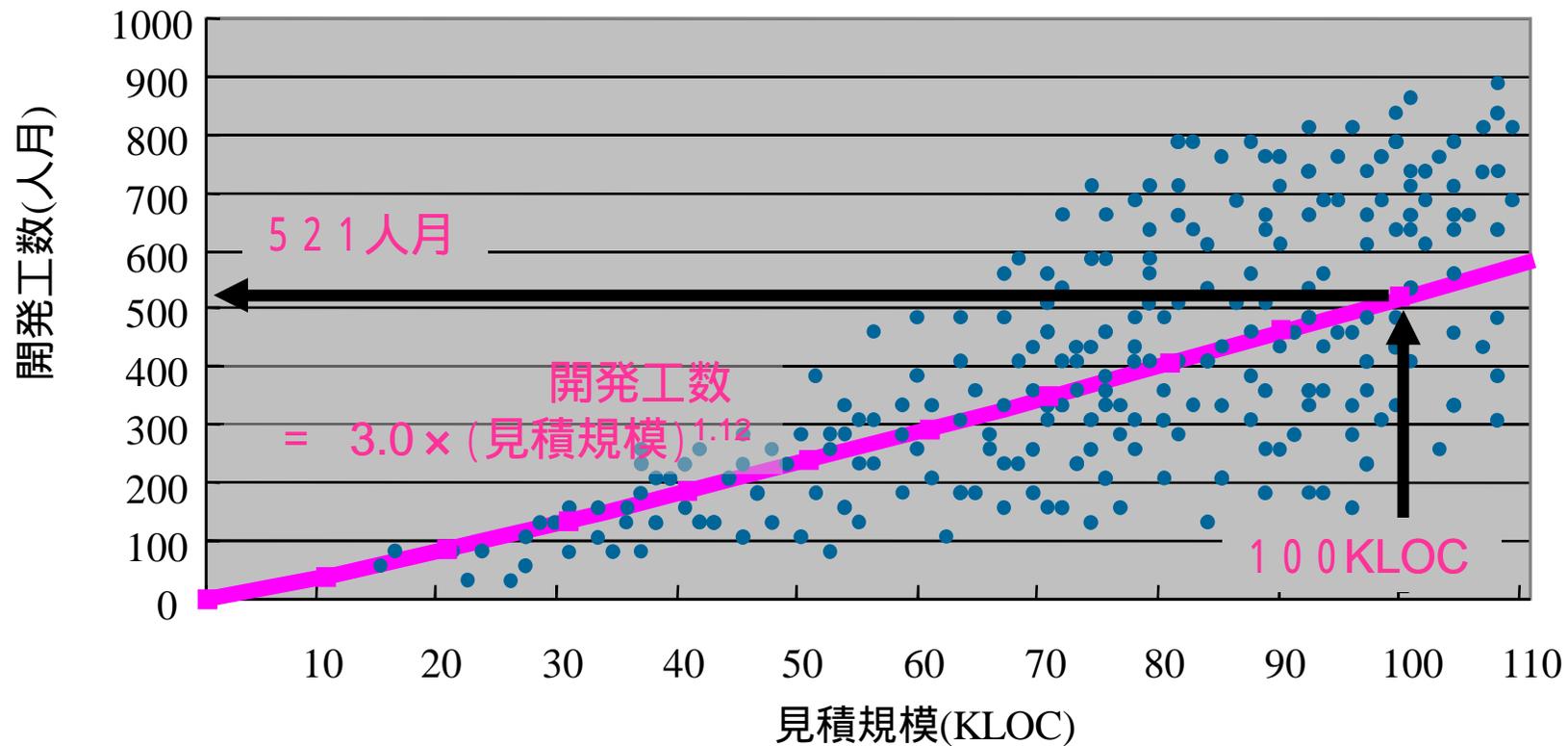
- ・因果関係を明らかにする.
- ・現象を**説明**できるようになる.

吉舖紀子訳, EASEプロジェクト監修, ソフトウェア工学・システム工学ハンドブック: エンピリカルアプローチによる法則とその理論, コンピュータ・エージ社(2005).

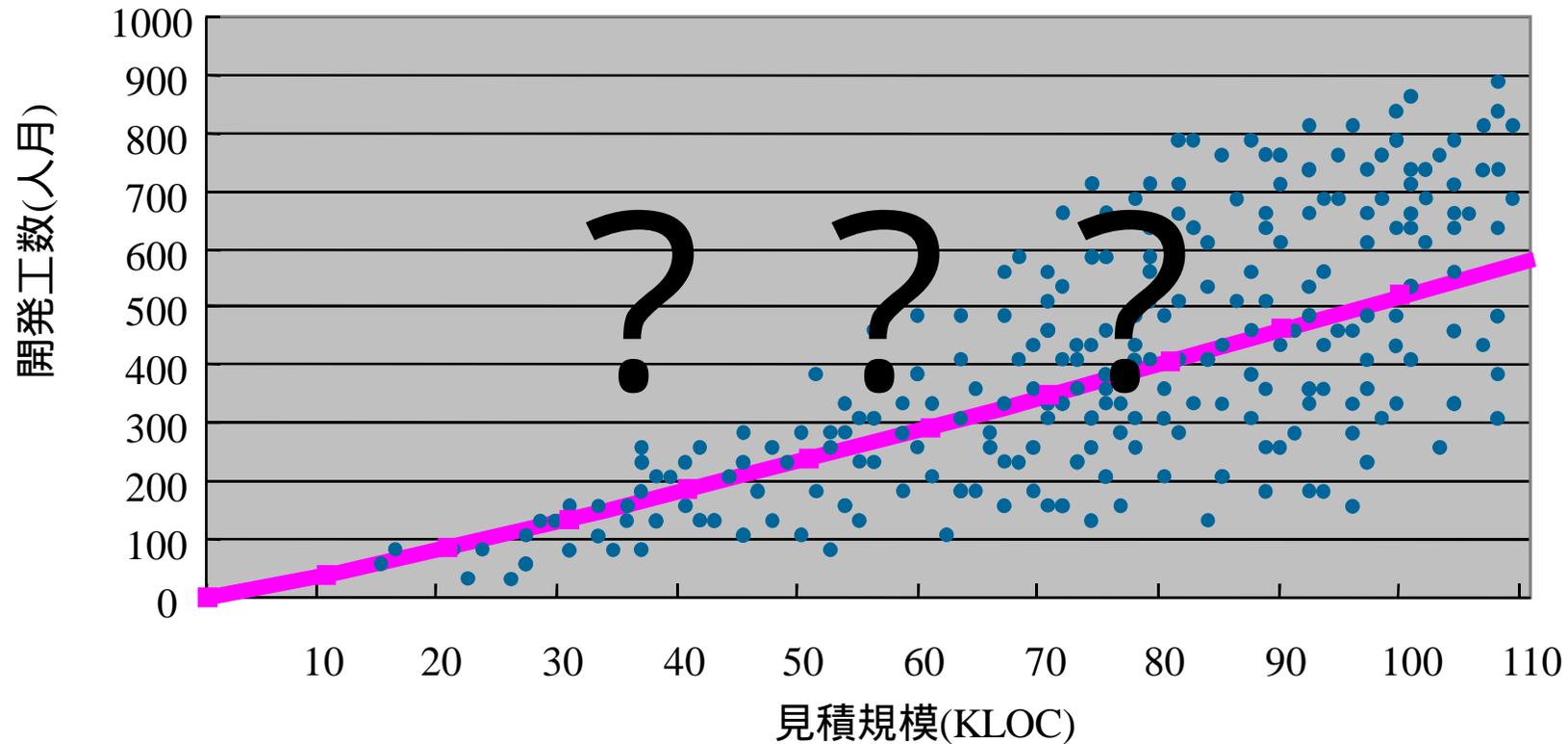
# 第1段階：観察の実施



# 第2段階: 法則の発見



# 第3段階:理論の確立???



人間を含む数多くの要素が複雑に絡み合って構成されるソフトウェア開発では、「理論の確立」まで成し遂げることは、少なくとも現状では非常に難しい。

# エンピリカルアプローチの3段階(改)

## 第1段階: 観察 (Empirical observations) の実施

- ・実験や調査により現象(事実)を確認する.
- ・現象を**表現**できるようになる.



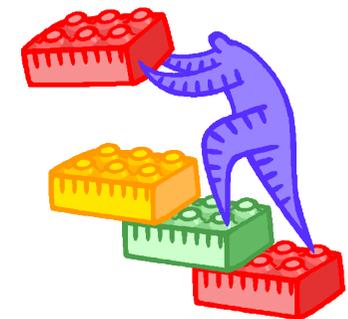
## 第2段階: 法則 (Laws) の発見

- ・現象が起こるコンテキスト(状況)を理解する.
- ・現象を**予測**できるようになる.



## 第3段階: 支援 (Supports) の実現

- ・法則を利用して現象を予測, 制御する.
- ・開発を**効率よく行い, 改善**できるようになる.



18

文部科学省リーディングプロジェクト

「e-Society基盤ソフトウェアの総合開発」（2003年4月～2008年3月）

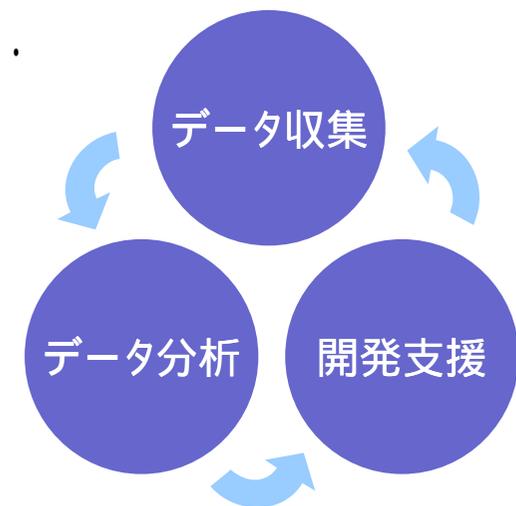
# データ収集に基づくソフトウェア開発支援 システム（EASEプロジェクト）

# EASEプロジェクト

- Empirical Approach to Software Engineering
- 文部科学省リーディングプロジェクト(2003～2007年度)
  - e-Society基盤ソフトウェアの総合開発。
    - データ収集に基づくソフトウェア開発支援システム。
- 主要組織
  - 奈良先端科学技術大学院大学, 大阪大学
  - 日立公共システムエンジニアリング, 日立製作所, NTTソフトウェア, SRA先端技術研究所
- [www.empirical.jp](http://www.empirical.jp)

# 目標

- エンタープライズ系，組込み系を問わず，広くソフトウェア開発において，他の科学や工学分野と同様に，データ収集，分析，そしてフィードバックによる改善（開発支援）というエンピリカルアプローチ（実証的手法）を実践する。
  - エンピリカル環境の提案と構築．
  - エンピリカル環境の配布・実プロジェクトへの適用．
  - プロジェクトや組織の壁を越えた，エンピリカルデータやその分析による知見の蓄積．
  - 環境適用先の組織のソフトウェア開発における生産性や信頼性の向上．



# 「法則の利用」が可能にする開発支援

- プロジェクトの工数, 日程, 品質を高精度で見積もる.
- 進行中のプロジェクトをコントロールする.
- 進行中のトラブルプロジェクトを再計画する.
- 組織内の全プロジェクトに対して, リソース割り当ての全体計画を作成する.
- 開発プロセスがどの程度改善されたかをモニターする.

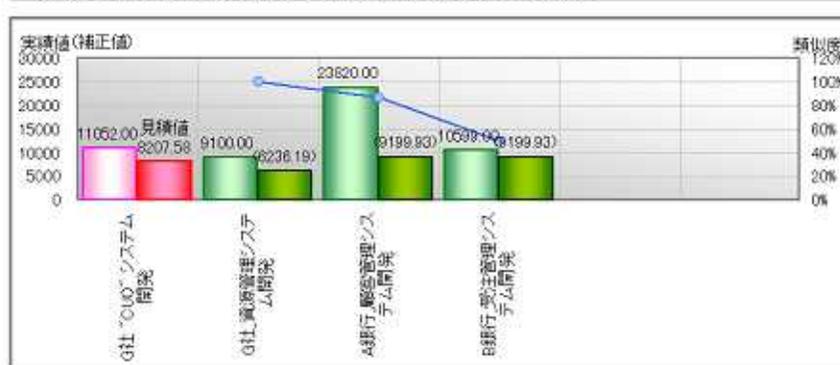
L. H. Putnam, W. Myers (著), 山浦恒央 (訳), 初めて学ぶソフトウェアメトリクス ~ プロジェクト見積もりのためのデータの導き方 ~, 日経BP社, 2005.

22

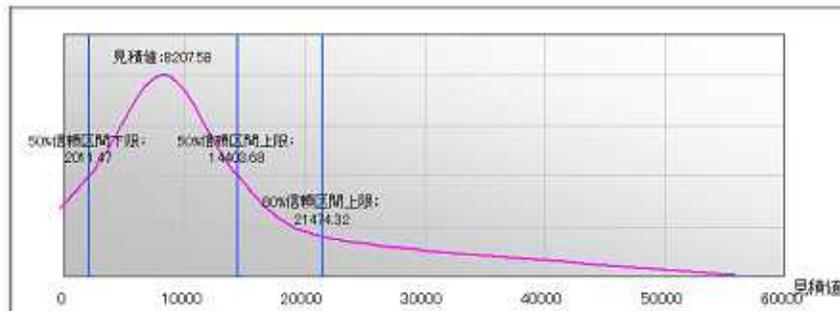
工数等の見積りに利用できるデータ・分析・法則

# 見積りモデルによる開発工数予測

類似プロジェクトの実績値(規模補正值)との比較

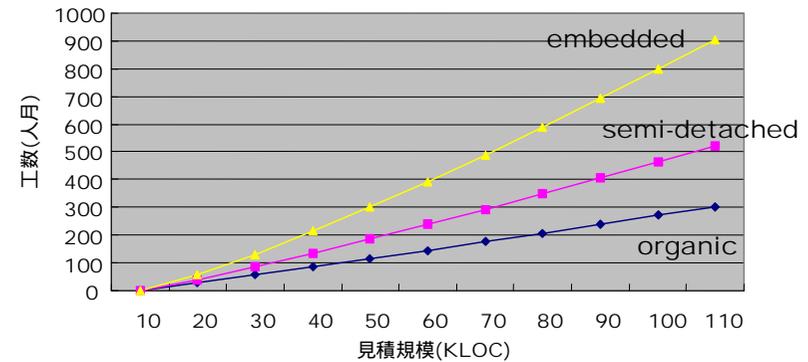


見積り値の信頼区間



Magiシステム

<http://se.naist.jp/cgi-bin/magi/licensing/te/getkey.cgi>

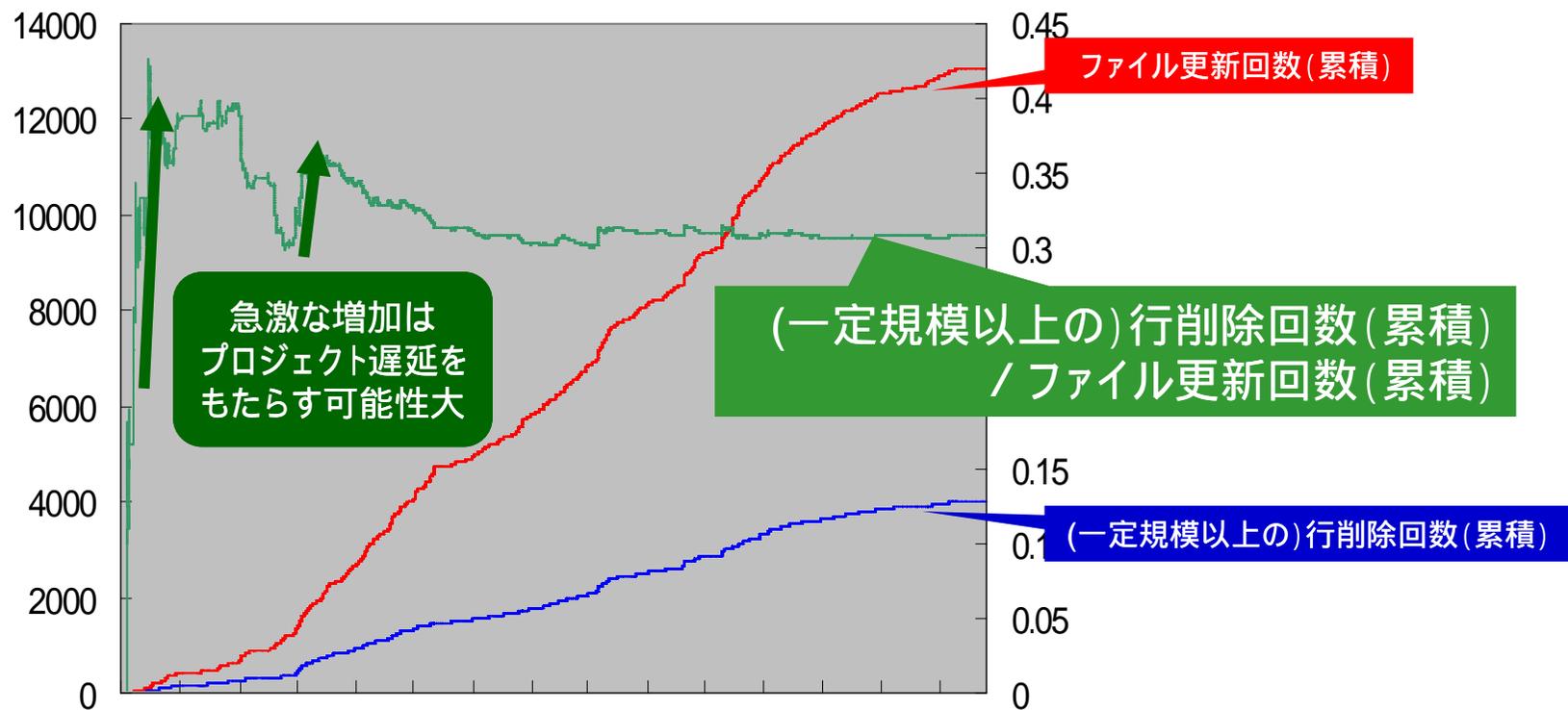


COCOMO

B.W. Boehm: "Software Engineering Economics, Prentice-Hall (1981).

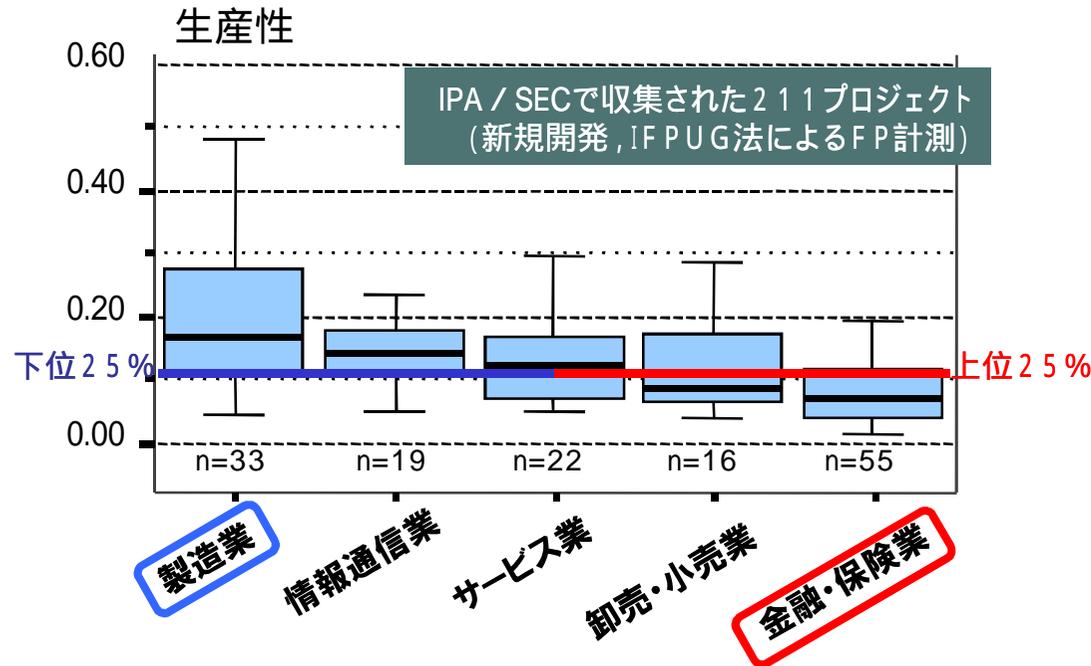
進行中のプロジェクトのコントロールに利用できるデータ・分析・法則

# ProPFVによるプロジェクト遅延リスク検出



Y. Mitani, N. Kikuchi, T. Matsumura, S. Iwamura, Y. Higo and K. Matsumoto, "Effects of software industry structure on a research framework for empirical software engineering," Proc. ICSE 2008, pp.616-619, Shanghai, China, May 2006.

# 業種間での生産性比較

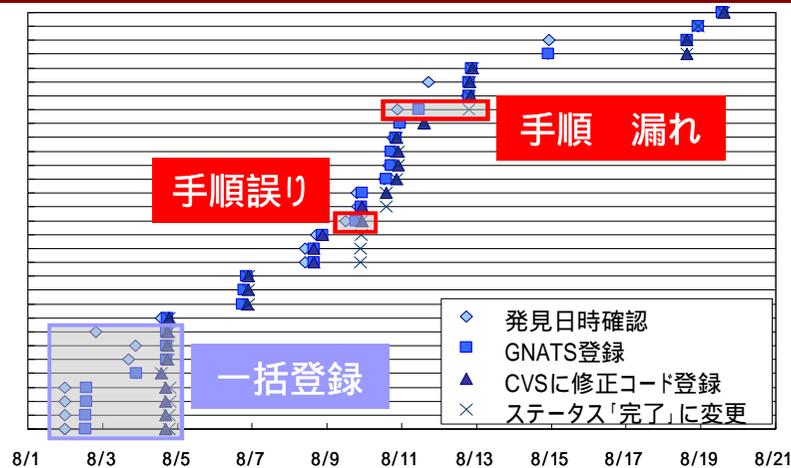


角田, 門田, 宿久, 菊地, 松本, “外部委託率に着目したソフトウェアプロジェクトの生産性分析”, 電子情報通信学会技術報告, ソフトウェアサイエンス研究会, SS2006-11, pp.19-24 (2006年4月).

海外の研究でも同様の傾向を示す.

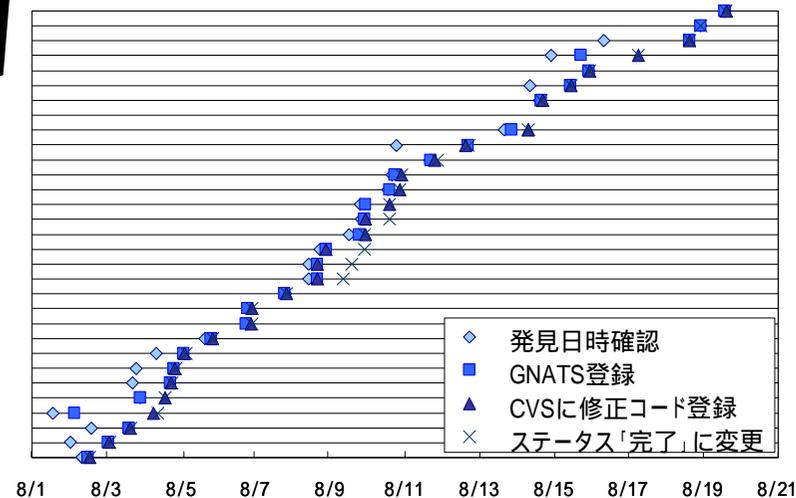
- K. Maxwell, and P. Forselius, “Benchmarking Software-Development Productivity,” IEEE Software, Vol.17, No.1, pp.80-88, 2000.
- R. Premraj, M. Shepperd, B. Kitchenham, and P. Forselius, “An Empirical Analysis of Software Productivity over Time,” In Proc. of 11th IEEE International Software Metrics Symposium (METRICS'05), Como, Italy, pp.37, Sep., 2005.
- C. Lokan, T. Wright, P. Hill, and M. Stringer, “Organizational Benchmarking Using the ISBSG Data Repository,” IEEE Software, Vol.18, No.5, pp.26-32, 2001.

# マイクロプロセス分析 (デバッグプロセス)



- 作業記録 (エンピリカルデータ) から直接プロセスに関する情報 (作業の順序, 時間, 作業量) を得る.
- 基準となるプロセスモデルを与え, 作業記録からプロセスインスタンスを得る.
- プロセスインスタンスに基づいて, プロセス品質を評価する.

デバッグ作業の見直し  
構成 / 障害管理ルール of 徹底

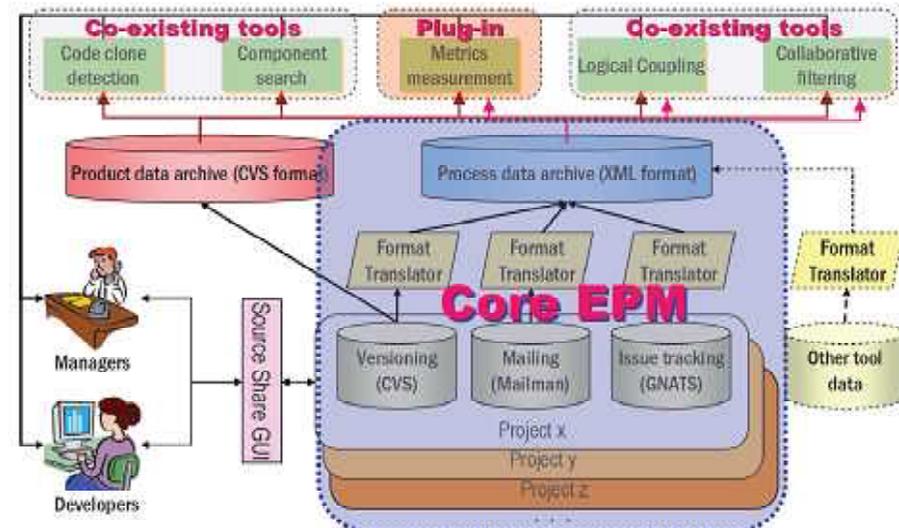


# EASEツール群



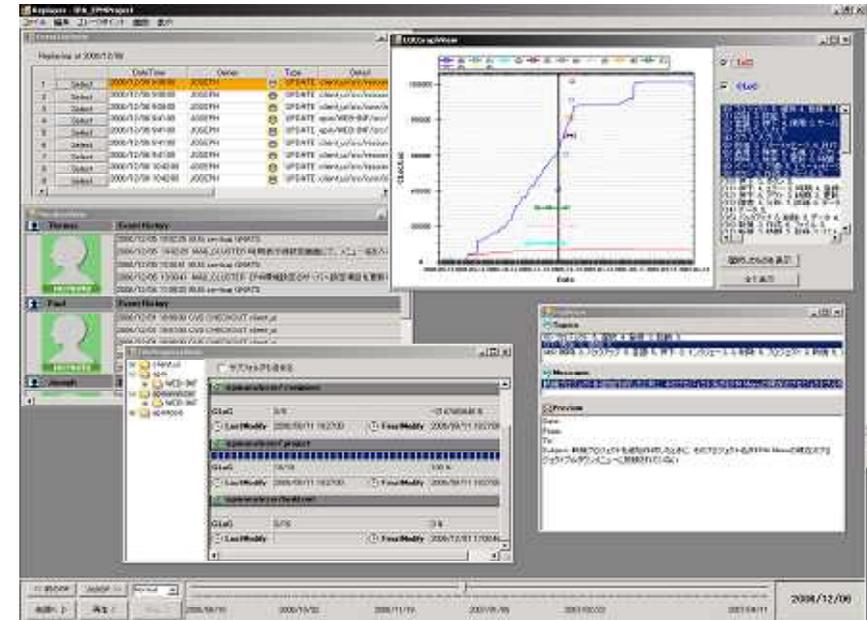
# EPM

- ソフトウェア開発における自動的なデータ収集と分析のためのプラットフォーム(エンピリカル環境)である。
- 開発者の負担を小さくするため,構成管理システム,メール管理システム,障害管理システムからデータ収集する。
- 標準エンピリカルデータのフォーマットは公開されている。



# Project Replayer

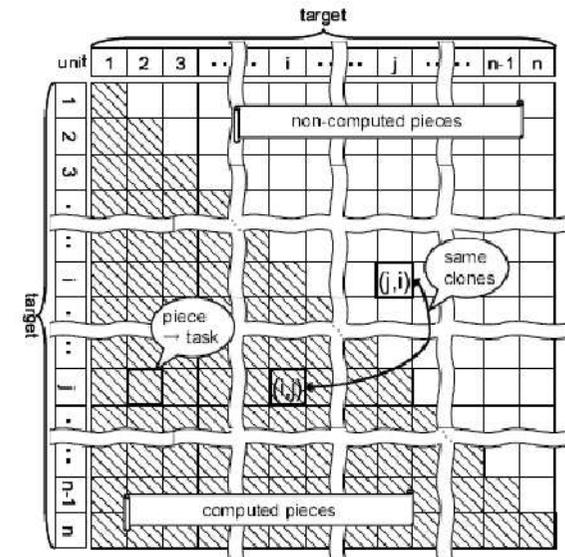
- EPM等で収集されたエンピリカルデータを基に，ソフトウェア開発プロジェクトを再現し，ビデオプレイヤーのような感覚でプロジェクトの流れを分析するツールである．
- 適用実験では，当該プロジェクトに全く関わっていない被験者であっても，開発が滞っていた期間の特定や，その理由の把握など開発プロセスの問題点が容易に認識できた．



K. Goto, N. Hanakawa, and H. Iida, "Project Replayer - An investigation tool to revisit processes of past projects," Proc. SPW/ProSim 2006, pp72-79, May. 2006.

# D - CCFinder

- 80台のコンピュータを用いた分散処理型コードクローン検出システムである。
- ソースコード群を小さなピースに分割し、ピース単位でCCFinderを実行することにより、コードクローンを検出する。
- FreeBSD用のソフトウェア集合Portsに含まれるCソースファイル約4億行から51時間でコードクローン検出を完了し、散布図等を用いてコードクローン共有状態の全体像を把握することができた。



S. Livieri, Y. Higo, M. Matsushita, K. Inoue: "Very-Large scale code clone analysis and visualization of open source programs using distributed CCFinder: D-CCFinder", Proc. ICSE 2007, pp.106-115, Minneapolis, MN, May 2007.

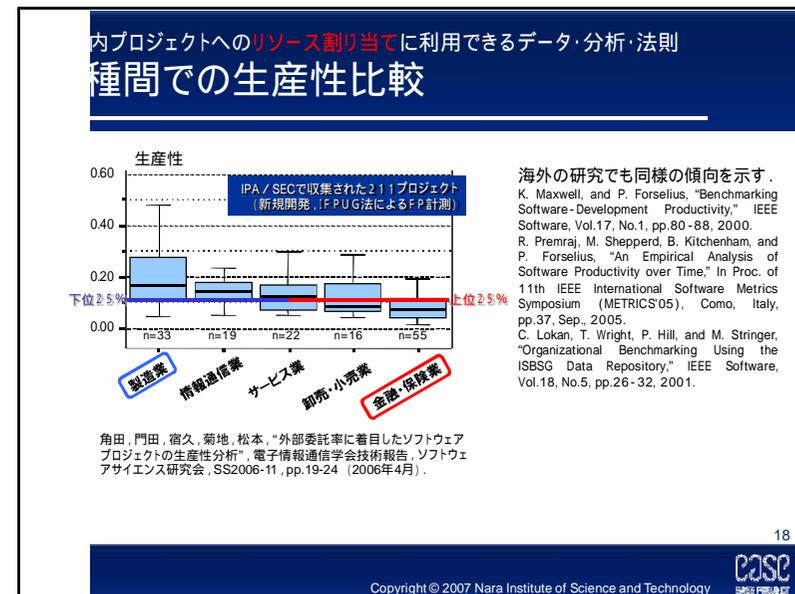
30

# 情報処理推進機構ソフトウェアエンジニアリングセンター (IPA / SEC) との連携 (1)

- エンピリカル環境普及事業・プロジェクトへの技術協力
  - ソフトウェア開発技術普及ツール開発事業
  - EPMツール検証プロジェクト
    - 参加企業 (順不同): 野村総合研究所, クエスト, 日本ユニシス, NEC ソフト, デンソー, 日本システムディベロップメント, SRA, オムロン, 日本電子計算, 高崎共同計算センター, 中電シーティーアイ, 横河電機, コア, サイオステクノロジー, セコム, アルプス電気, エヌアイデイ, NTT データシステムズ, 日本電気, オートネットワーク技術研究所, アルファシステムズ, アルゴ21, パナソニックコミュニケーションズ, ジャステック, 日立ソフトウェアエンジニアリング, フュートレック, ミラクル・リナックス, 東芝ソリューション, 日立公共システムエンジニアリング, 日本証券テクノロジー 他 計 60社以上
  - 情報サービス産業協会 (JISA) と Information Technology Alliance (ITA) では, EPM 専門部会が組織されている。

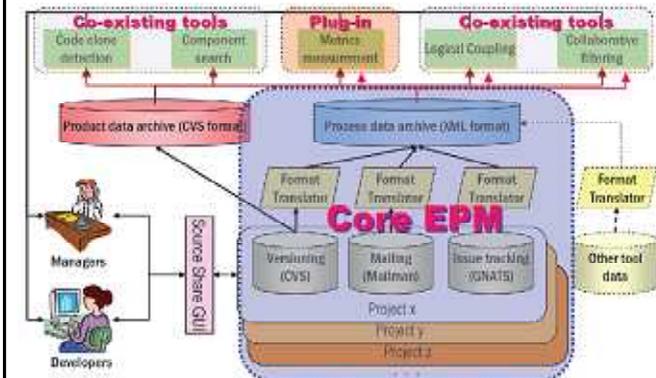
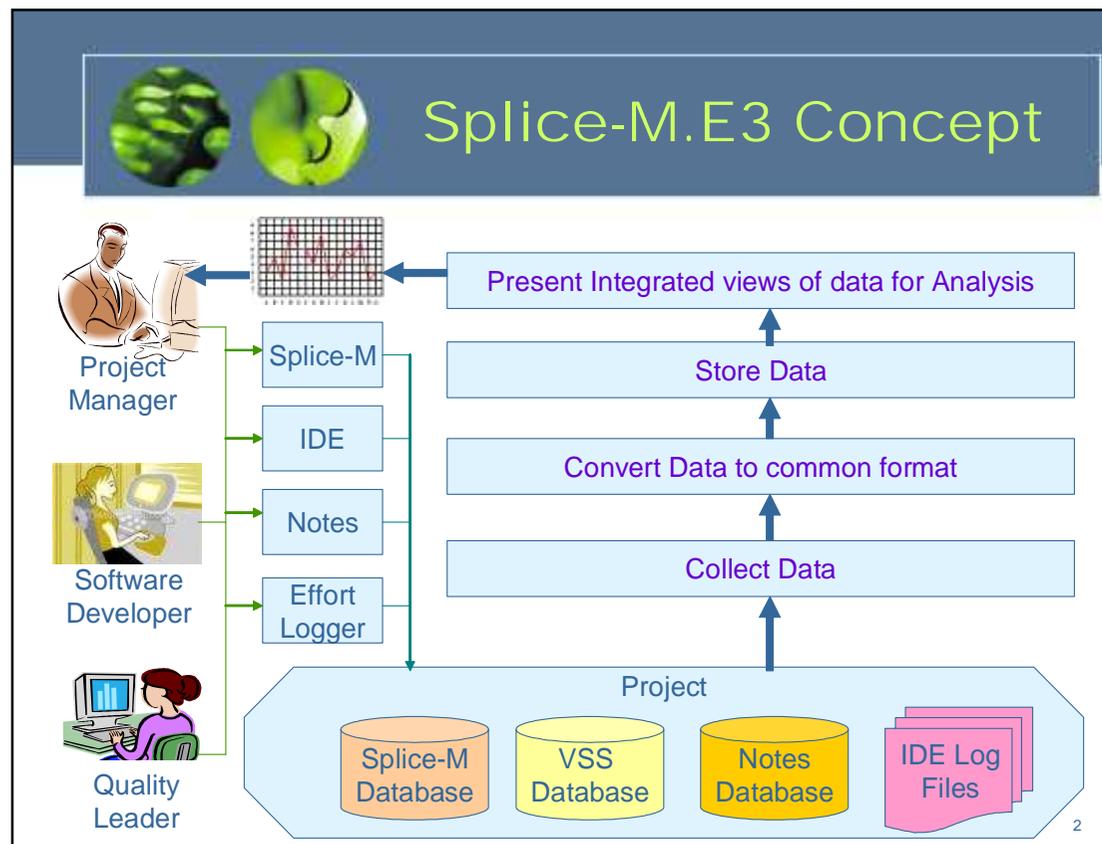
# IPA / SECとの連携(2)

- 組織横断的データ分析技術の開発
  - SECデータベース(1770プロジェクトデータ)を対象とする.
  - 欠損値を含む大量のプロジェクトデータに対する分析フレームワークの確立を目指す.
    - 欠損値・外れ値の除去・補完手法
    - 協調フィルタリングを活用したプロジェクトデータ層別方法
    - 開発管理向けベースライン構築技術
    - 類似するプロジェクトデータサブセットの算出



# 海外への波及例

- インドL & T Infotech社 SpliceM.E3



# 解説本の市販，DVDによる成果配布

- 鳥居 宏次 監修，ソフトウェア開発におけるエンピリカルアプローチ，アスキー（2008）。



- 成果DVD ([www.empirical.jp/EASE\\_DVD](http://www.empirical.jp/EASE_DVD))
  - 研究発表: PDFファイル。
  - EASEツール: インストーラ, 説明書, 関連文献等。

# まとめ

- 測定データ(実証・実績データ)に基づいてソフトウェアの生産性や信頼性の向上を目指す。
  - Empirical = Experimental  
(大学などでの実験)  
+ Experienced  
(開発現場での経験や実績)
- 文部科学省EASEプロジェクトにより、エンピリカルアプローチが産業界で広く実践された。
- 現在、ソフトウェアの発注者やエンドユーザにも対象を広げ、エンピリカルソフトウェア工学のより先進的な実践を行うため、文部科学省Stageプロジェクト(2007年8月～2012年3月)が実施されている。



# 関連URL

- EASEプロジェクトホームページ [www.empirical.jp](http://www.empirical.jp)
- Stageプロジェクトホームページ [www.stage-project.jp](http://www.stage-project.jp)
  
- 奈良先端科学技術大学院大学  
ソフトウェア工学講座 [se.naist.jp](http://se.naist.jp)  
ソフトウェア設計学講座 [sdlab.naist.jp](http://sdlab.naist.jp)
  
- 大阪大学  
ソフトウェア工学講座 [sel.ist.osaka-u.ac.jp](http://sel.ist.osaka-u.ac.jp)  
ソフトウェア設計学講座 [sdl.ist.osaka-u.ac.jp](http://sdl.ist.osaka-u.ac.jp)