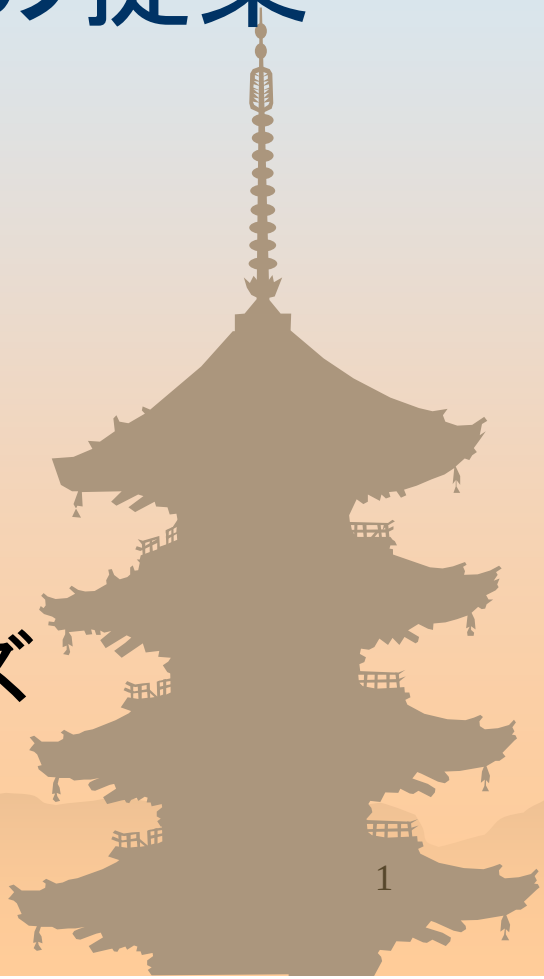
ソフトウェア開発における予測モデル構築手法の提案

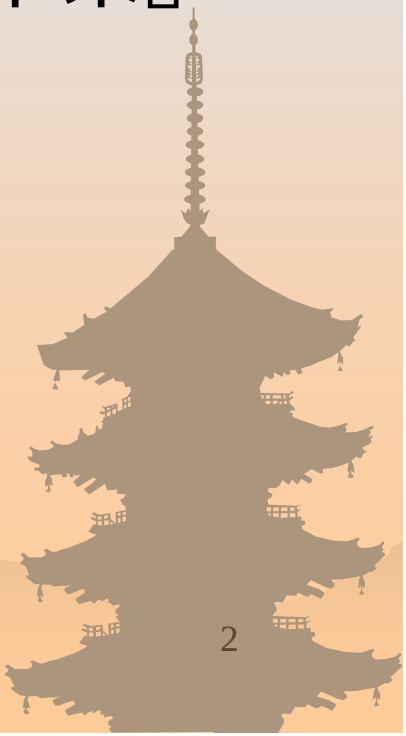
小室　睦

（株）日立ソリューションズ



未来を語る前に、今の現実を
知らなければならない。
現実からしかスタートできないからである。

―ドラッカー『産業人の未来』



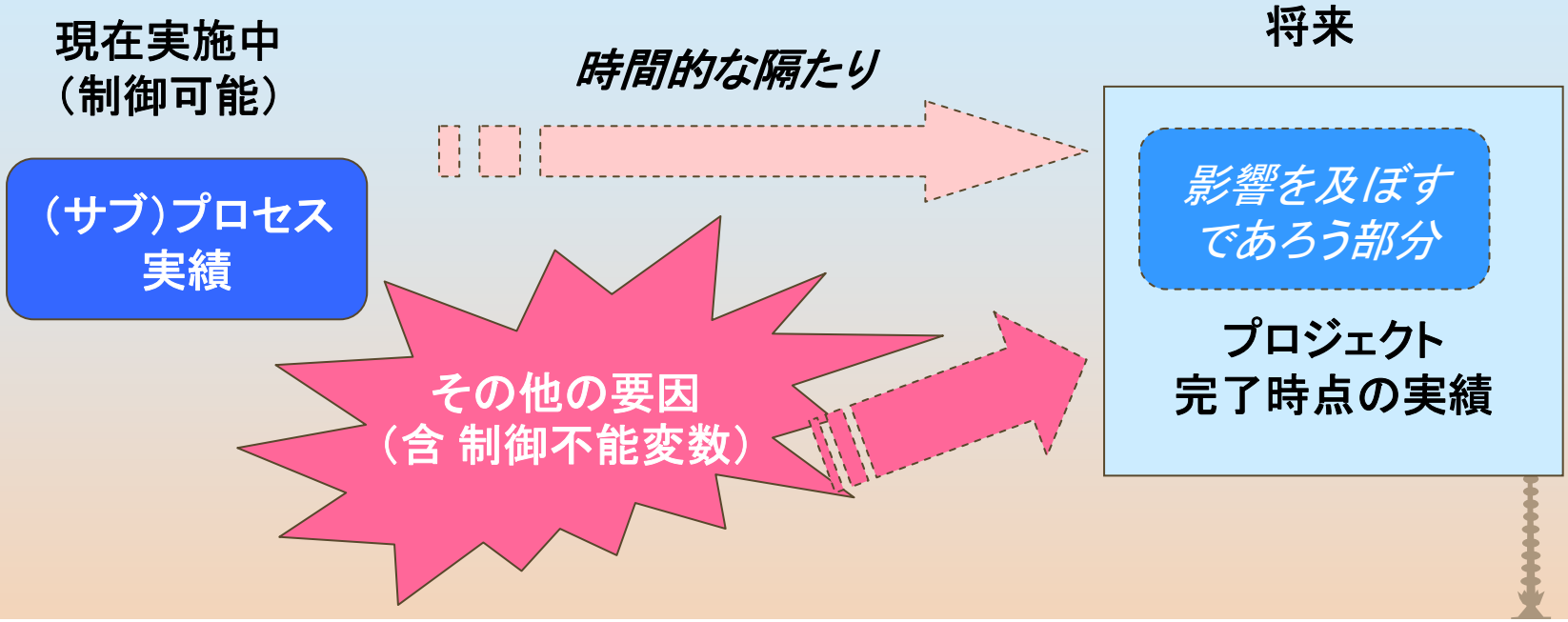
目次

- 背景・動機
- 予測モデル構築の難しさ
- プロセスモデルの発展
- 実施例１:ピアレビューと品質予測
- 実施例２:プロジェクトのトラブル予測
- まとめ

背景・動機

- ソフトウェア開発は難しい！
  - ソフトウェアプロジェクトの50 %は失敗(J.Johnson,1995)
  - 不確実性との戦い
    - 途中経過が明確に把握できない
    - 最終的な実績がはっきりわからない
- 予測可能性
  - プロジェクト管理での最大の課題の一つ
  - トラブルの未然防止、事前処置
- 予測の難しさ
  - CMMIではレベル4で予測を可能にする「プロセス実績モデル」の確立が想定されている
  - 離れた時点でのプロセスデータの関係付け
  - さまざまな要因から来る雑音の存在

予測モデル構築の難しさ



- 典型的な失敗例（ビッグバンアプローチ）
  - 利用可能な測定データを統計分析にかけ、相関関係を探索
  - 「雑音」が大きく探知したい信号は小さい
    - 適切な関係／関数関係の特定が困難
  - 一旦関係が見つかったと思っても別のデータで試すと消えてしまう
  - 関係が求まっても開発経験上の意味づけとあわない

プロセスモデルの発展

定性的モデル
このプロセスを実施すると最終的な実績がよくなるという理解

定量化による効果確認
改善前、改善後の実績を検定等により定量的に比較

改善効果の説明モデル
プロセス実績と改善効果を定量的に関連付ける

効果の説明／評価と予測の役割を分離して、困難を分割

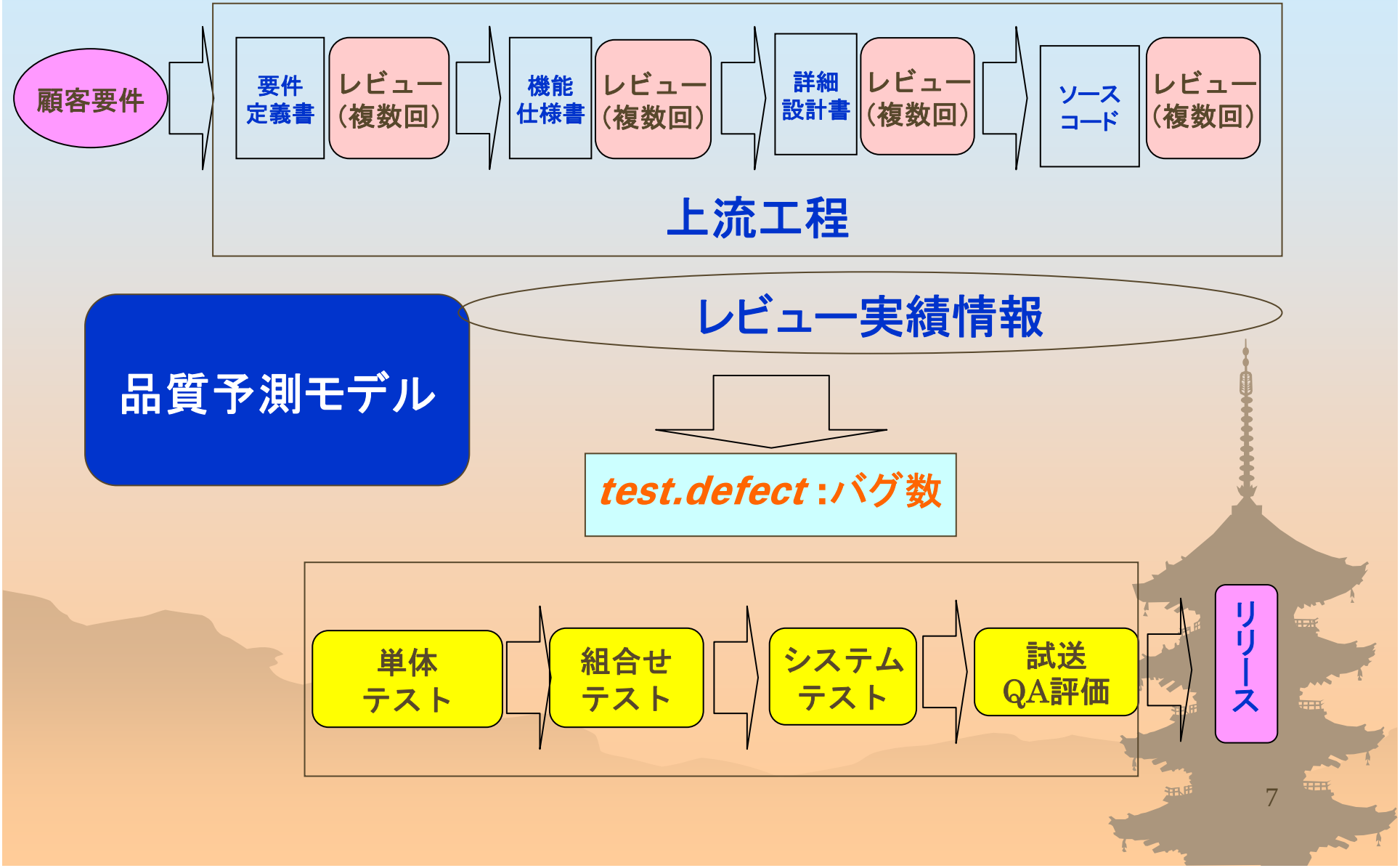
予測モデル
実施途中の実績データから完了時の結果を予測

どう関係しているか／すべきかという「意味づけ」をまず確立

プロセス実績モデル
将来結果を予測し、目標達成できるようサブプロセスを制御

サブプロセスの制御

実施例1:ピアレビューと品質予測

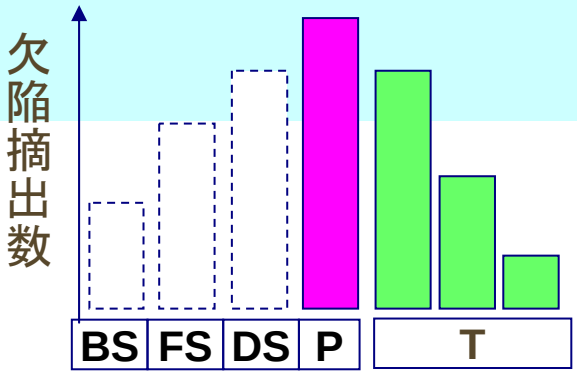


用語の定義と仮定

- *size*: 開発規模, 単位はキロステップ数.
- ウォータフォールのライフサイクルモデルを仮定
- p_j: j番目の開発工程
  - 適用組織では以下のような工程記号を用いている:
    - BS: 基本設計, FS: 機能設計 ,DS: 詳細設計
    - P: プログラミング, T : テスト
- $f(p_j)$: 工程p_jで摘出された欠陥の件数
- テスト工程 Tに対する $f(T)$ のことを *test.defect*とも書く。
- $d(p_j)$: 工程 p_jでの前倒し率

$$d(p_j) = f(p_j) \Big/ \Sigma_{k \geq j} f(p_k)$$

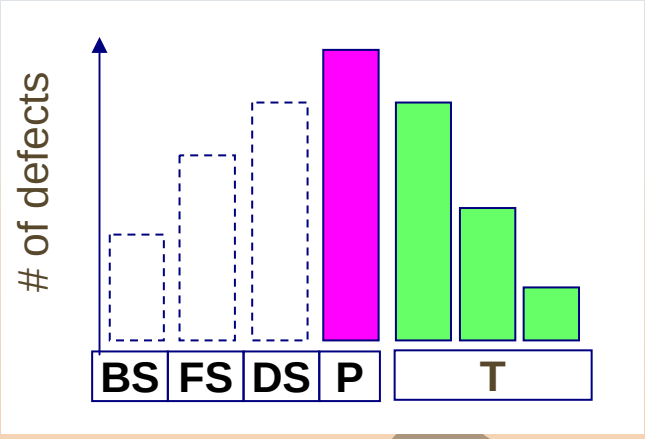
$$d(P) = \frac{\square}{\square + \square}$$



効果説明モデルの確立

- テスト工程での欠陥摘出数*test.defect* とピアレビュー実績を関係付けるモデル
  - 従来からの研究結果（レイリーモデル他）から前倒し率 $d(\mathrm{p}_j)$ が*test.defect* に指数関数的に影響すると期待される

$$d(\mathrm{P}) = \frac{\square}{\square + \square}$$



- 直感的意味づけ
  - 前倒し率 $d(\mathrm{p}_j)$ は工程p_jでのピアレビュー実施の徹底度を表わす。
  - ピアレビューを徹底すれば、それ以降の工程の品質が大きく向上する。

$$\log(test.defect) = a_1 \log(size) + a_2 d(\mathrm{p}_j) + (\text{他の要因の影響を表わすダミー変数達})$$

効果説明モデル

上流と下流の摘出欠陥数の比

実際のプロジェクトデータを用いた線形回帰で検証

ポアソン分布に対する一般化線形回帰

p_j=Pの場合：p値 3.4×10^{-5}

擬似R^2 92.6%

レビューの徹底度

減少方向に強く影響（指数関数的）

他の要因
（難易度・開発経験など）

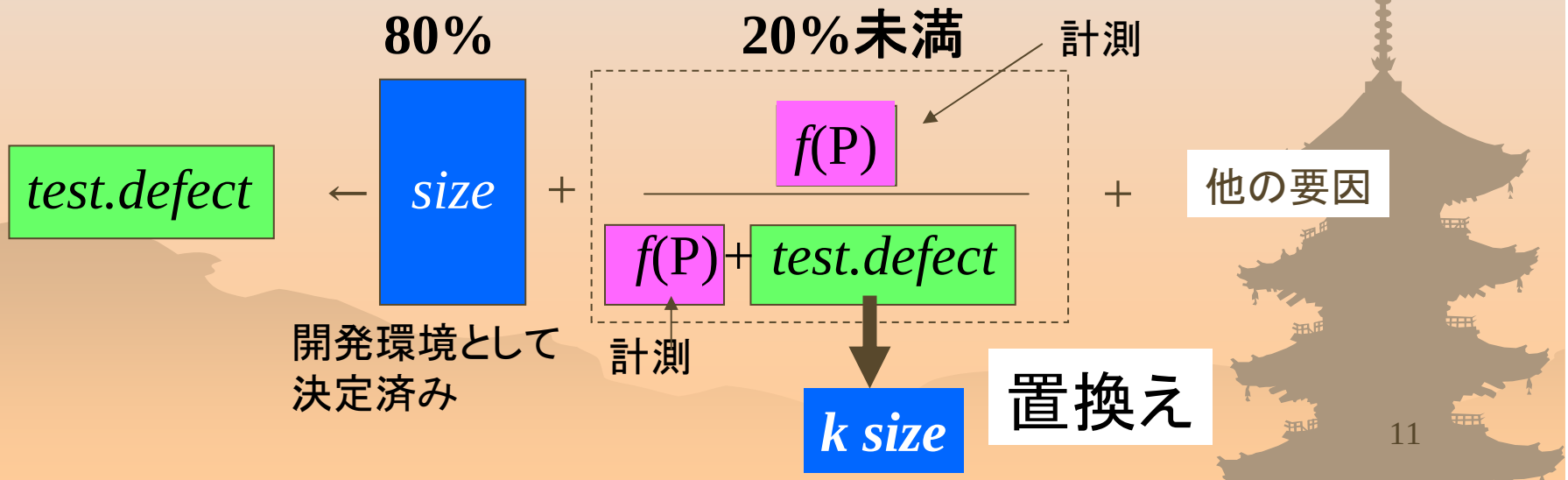
雑音を消去

開発規模
（制御可能ではない）

テスト工程での欠陥数

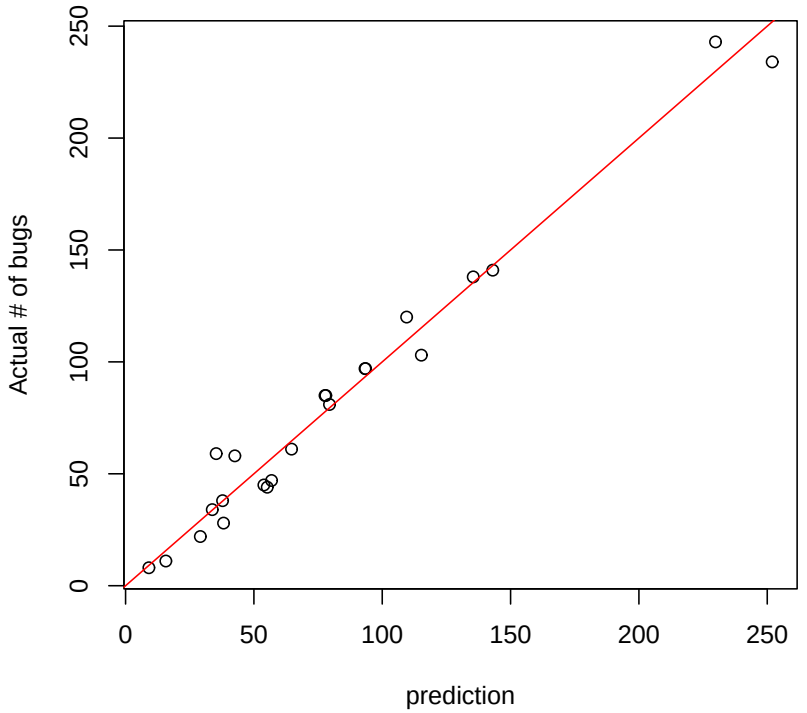
予測モデルへの変形

- アイデア
  - 回帰式は（誤差項つきの）近似式なので、*test.defect* の「十分良い」近似値が得られればそれで十分。
  - 回帰式の主要項は*size*である。
    - *size* との単回帰分析を実施すると寄与率 R^2 は 80%以上。
      - 適当な定数 $k>0$により　*test.defect* = *k size*　という近似式が成立つ
    - 開発規模 *size* は制御可能変数ではないことに注意



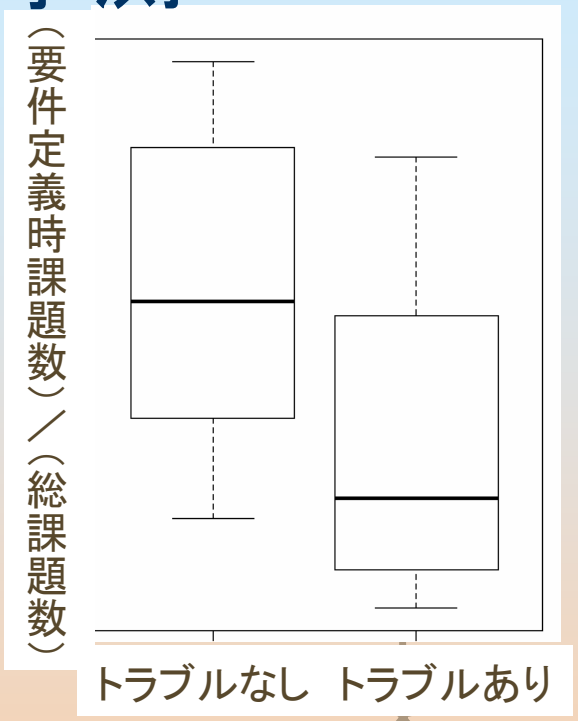
実データによる検証

- 以上の検討で見出した説明変数(とその関数形)を用いて線形回帰分析を実施(擬ポアソン分布に対する一般化線形回帰)
  - 同一事業部からのデータ23件を使用し、
  - Web系の開発で比較的小規模、言語はJava
- コーディングレビュー結果からの予測式
  - $test.defect = 5.3 size^{1.003} exp(-0.040 \log(base.size+1) + 0.62 size/f(P) - 0.60\ PM - 0.50\ SF)$
    - base.size:母体規模,
      PM:プロジェクトマネージャの経験(ダミー変数)、
      SF: システム開発の自由度(ダミー変数)
  - 逸脱度:948.95,
    残差逸脱度: 52.196,
    残差自由度:17,
    擬似R^2:94.5%
  - プログラミングレビューでの欠陥摘出数f(P)を増やすとバグ数*test.defect*は減少する



実施例2:要件管理状況からのプロジェクトのトラブル予測

- プロジェクトは要件および要件変更を「課題一覧」として管理
- 仮説:早めに課題を出し切っているプロジェクトはトラブルに陥らない



- 次の2変数の間には関係がある
  - 要件管理に起因するトラブルがあったかどうか
    - プロジェクトマネジャーとQAの合議による評価
  - 説明変数:「課題一覧」中の課題のうち要件定義工程で見出したものの割合
- 説明モデル: 2項変数に対するプロビット分析
  - 目的変数 T:要件管理に起因するトラブルがあったかどうか(値域0,1)
  - 説明変数
    - R: 要件定義時の課題数
    - K: 課題総数
    - E: 要件定義工程完了時のプロセス評価（値域0,1）

$$\Phi^{-1}(T) = a \log(R) + b \log(K) + c\,E + d$$

ただし、Φ^{-1}はプロビット関数（正規分布の累積分布関数の逆関数）

課題の出方（上流と下流の課題数の対比）

要件定義工程での課題数

総課題数

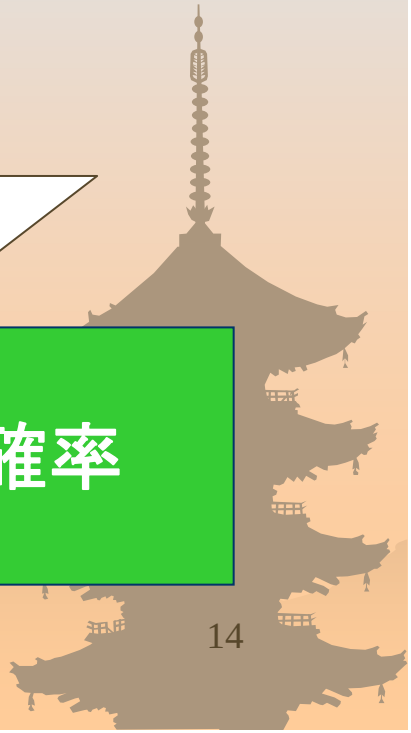
効果説明モデル

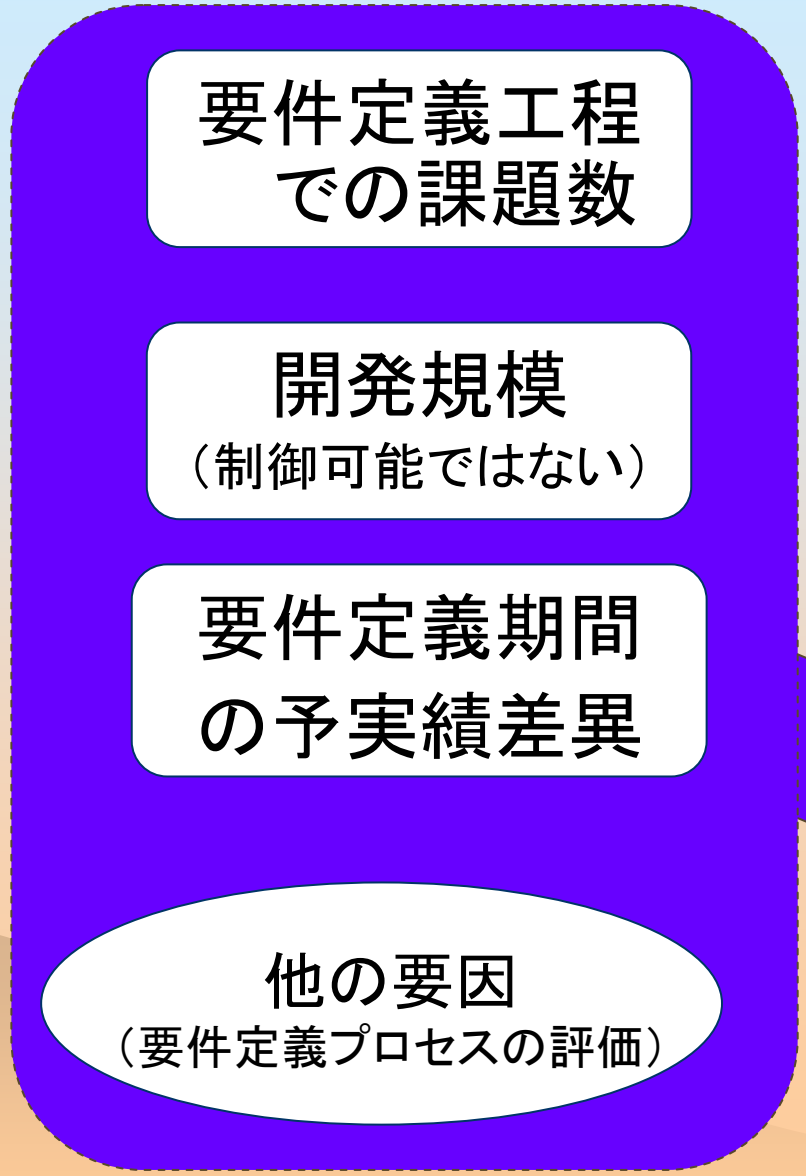
減少方向に影響

トラブル確率

他の要因
（要件定義プロセスの評価）

雑音を消去





予測モデル

トラブル確率

予測モデルへの変換

- 「総課題数」
  - 正確には、プロジェクト完了時にならないと確定しない
  - 次の2つの量から推定できる:開発規模(Kstep),要件定義期間の予実績比
- 予測モデル
  - 「総課題数」を上記の2つの変数で置換えて、重回帰分析を実施
    - 同一事業部からの24プロジェクトのデータ
  - 結果

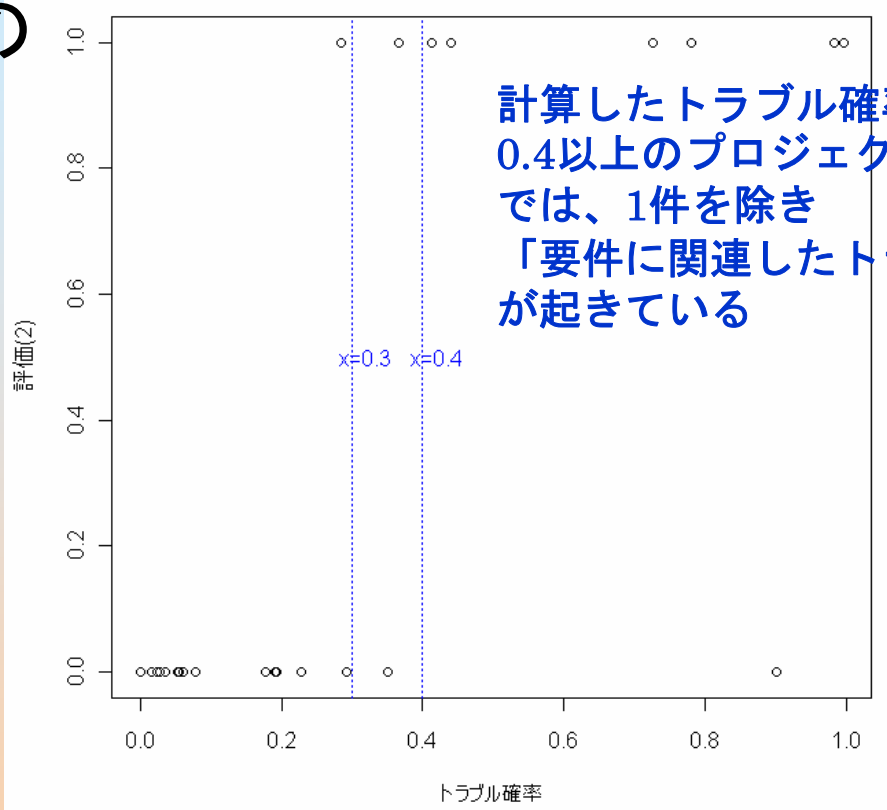
$$(\text{トラブル予想確率}) = \Phi(3.82 - 0.85*\log(\text{要件定義時課題数}) + 0.80*\log(\text{開発規模}) - 0.097*\text{要件定義期間} - 2.99*E + 0.075*\text{計画期間})$$

逸脱度:30.553, 残差逸脱度: 20.793, 残差自由度:19.

- 定性的な振舞い
  - 評価Eの結果が悪いとトラブル確率も高い
  - 評価Eで表現されている要件定義工程でのトラブルの分を除くと、(計画された要件定義期間程度に)要件定義の時間を割いたプロジェクトの方がトラブル確率は低くなる
  - 開発規模の割りに要件定義時課題数が少ないプロジェクトはトラブル確率が高い

評価

- プロジェクトデータとのあてはまりは良い



計算したトラブル確率が
0.4以上のプロジェクト(7件)
では、1件を除き
「要件に関連したトラブル」
が起きている

計算したトラブル確率が
0.3以下のプロジェクト(15件)
では、1件を除き
「要件に関連したトラブル」
は見られない

プロジェクトデータとの照合結果

- 実施中のプロジェクト(4～5件)の予測では専門家の評価とツールの評価がおおむね一致

まとめ

- 予測モデル構築のための発展的アプローチを提案した
  - 定性的モデル→効果説明モデル→予測モデル
- 適用事例を2つ示した
  - ピアレビュー実績からの品質予測
  - 課題件数からのトラブル予測
- 一般的な指針
  - 上流プロセスと下流プロセスの実績の対比を用いる
  - 制御可能でない変数(例：開発規模)も説明変数に採用することで近似式としての変形を可能とする
- ベテランのプロジェクトマネージャの思考を定量化している：
  - このシステムは規模がこれくらいだから、最終的な実績はこれぐらいになるはず
  - 現在の実施工程までの実績がこれくらい
  - この調子なら大丈夫だろう／ちょっと不安だ／手を打たないと危ないぞ

教訓:モデル構築に向けて

- 効果を定量的に明確にすることが予測にも（自然に）つながる
- モデル構築に役立つノウハウ
  - 専門家の知識
    - ソフトウェア工学の知識、プロジェクト管理の実践経験など
  - 上流工程と下流工程の実績の対比をとる
    - 「上流工程が重要」という常識の定量的定式化
  - データの分布を調べる
    - 依存関係についての適切な関数形の推定の助けになる
  - 説明変数に制御不能(uncontrollable)な変数も入れる
    - 実際どう効いているのか正確に知ることがモデルの変換を可能にする

募集中！

- 本発表の提案手法はとても自然なもの
  - プロセスモデル発展の仕方として自然
  - プロジェクトマネージャの自然な思考を定量化
- 本発表のモデル化手法の適用対象を募集します
- サポートします
  - 弊社での適用事例のチュートリアル
    - ツールの使い方、分析事例含む
  - 分析ワークショップ
    - 分析を一緒に実施して手法の適用法を例示
- ご協力方法
  - (1) JASPIC参加企業であれば、JASPICを通じて
  - (2) それ以外の場合は、個別にご相談ください
  - 連絡先： mutsumi.komuro.ej@hitachi-solutions.com

参考文献

- Johnson, J., “Chaos: The Dollar Drain of IT Project Failures.” *Application Development Trends*, No.1, pp. 41-47, 1995.
- Komuro, M. and Komoda, N., ”A Model to Explain Quality Improvement Effect of Peer Reviews,” *IEICE Trans. on Info. & Systems*, E93-D, No.1, pp.43-49, 2010.
- 小室睦,薦田憲久,ピアレビューデータに基づく品質予測モデル,電子情報通信学会論文誌, Vol.J94-D,No.2,pp.439-449, 2011.

The End