

エンタープライズ系ソフトウェア プロジェクトの生産性への 影響要因の分析

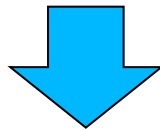
古山恒夫
東海大学

furuyama@tokai-u.jp

エンタープライズ系ソフトウェア
プロジェクトの生産性への
影響要因の分析

研究の動機

大規模なプロジェクトデータベースの存在

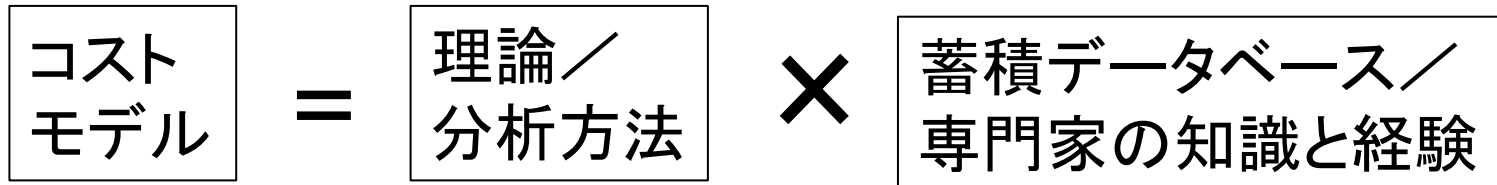


コストモデルが構築できないか？

エンタープライズ系ソフトウェア
プロジェクトの生産性への
影響要因の分析

- 背景
- 分析データ
- データ分析方法
- 分析結果と考察
 - 量的データの回帰分析
 - 質的データの一次元分散分析
 - 質的データの二次元分散分析
 - COCOMO IIとの比較
- まとめ

背景



- (重)回帰分析
- アナロジー
- 専門家の判断
- ニューラルネット
- ベイジアンネット
- ...

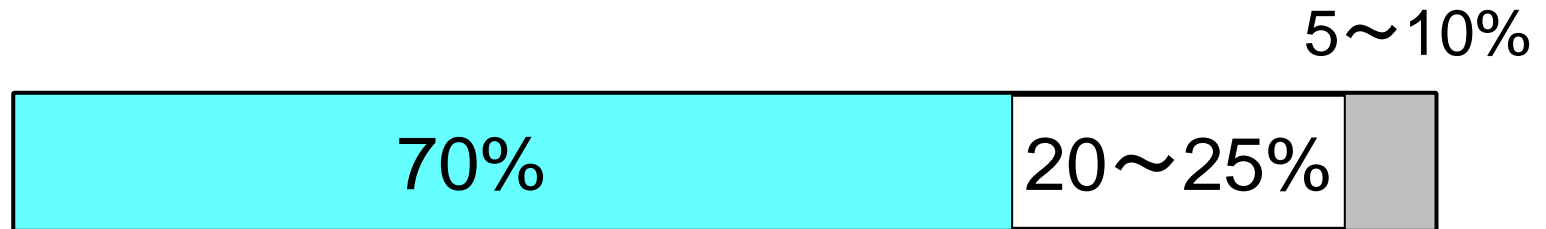
重回帰分析

- 最も基本的／さまざまな分析法のベースライン
- 蓄積データベースと重回帰分析によるコストモデルが最も多く、最近になっても過半数を占めている [1]
- さまざまなタイプのモデルの中で対数変換と回帰分析の組み合わせが最もよい [2]

[1] Jorgensen, M, and Shepperd, M.: A Systematic Review of Software Development Cost Estimation Studies, IEEE Trans. SE, Vol.33, No.1, pp.33-53 (2007).

[2] Dejaeger, K., Verbeke, W., Martens, D., and Baesens, B.: Data Mining Techniques for Software Effort Estimation: A Comparative Study, IEEE Tr. SE, Vol. 38, No.2, pp. 375-397 (2012).

工数への影響要因



規模 (FP)



- ・量的変数 (規模以外)
- ・質的変数 (名義尺度)
- ・ " (順序尺度)

特定不可能

COCOMO と COCOMO II

- COCOMO:

$$\text{工数} = A \times (LOC)^B \times \prod_{i=1}^{15} Cd_i$$

- COCOMO II:

$$\text{工数} = A \times (\text{規模})^{B + \sum_{j=1}^5 Sf_j} \times \prod_{i=1}^{17} Cd_i$$

Cd:コストドライバー
Sf:スケールファクタ
規模: SLOC または FP

PROMISEレポジトリの概要

データベース名	coc81	coc81-dem	coc2000	nasa93	nasa93-dem	Maxwell	usp05	China	Kitchenham
プロジェクト数	63	63	125	101	93	62	919	499	145
属性(変数)	19	27	50	24	27	27	17	19	10
順序尺度 対応の 変数	*15	22	*22	15	22	15	2	0	0
欠損値の数	0	0	0	0	0	0	83	0	13

(注) 50件以上のプロジェクトがあるものを掲載

* いくつかの既定値から値を選択する

ほとんどのデータベースが完全!

SEC データ

- 独立行政法人 情報処理推進機構 (IPA) ソフトウェア高信頼化センター (SEC) が収集したエンタープライズ系ソフトウェアプロジェクトデータ
- 特長: 1) 200 以上の変数 (データ項目) を定義
詳細なものまで含めると400以上 (*)
2) 3,000件以上のプロジェクトからのデータ
- 問題点: 大量の欠損値を含む

(*) ISBSGデータベースでは変数は少なくとも86あるが、多くは工数・欠陥・プロジェクトプロファイルを詳細化したもの

SECデータをどのように有効活用するか？

(1) リストワイズ削除後に重回帰分析

- 完全データベースを作成するとき、プロジェクト数を優先すると・・・重要な変数が抜けてしまうかもしれない
- 完全データベースを作成するとき、変数の数を優先すると・・・十分なプロジェクト数が揃わないかもしれない

(2) 前進選択法などで変数選択をしながら重回帰分析

- 毎回選択されるプロジェクトの集合が異なるため、既に除外された変数の組み合わせの方が、最新の時点のものより適切な組み合わせであることが起こりうる??

欠損値をうまく補間できるか？

- 多重代入法などが提唱されている
- 欠損値が完全にランダム (MCAR)なものやランダム (MAR)なものには有効
- ランダムでないものや大量の欠損値を含む場合には有効性は保証されない

→ 欠損値を補間することよりも

今あるデータ(情報)からわかることは何か
どうすれば現場にフィードバックできるか

SECデータの分析

最初のステップ:

- 量的変数に対する重回帰分析
 - FP, テストケース数, バグ数
- 分散分析(一次元、二次元)
 - 順序尺度に従う39個の変数
 - クロス効果についても試行分析

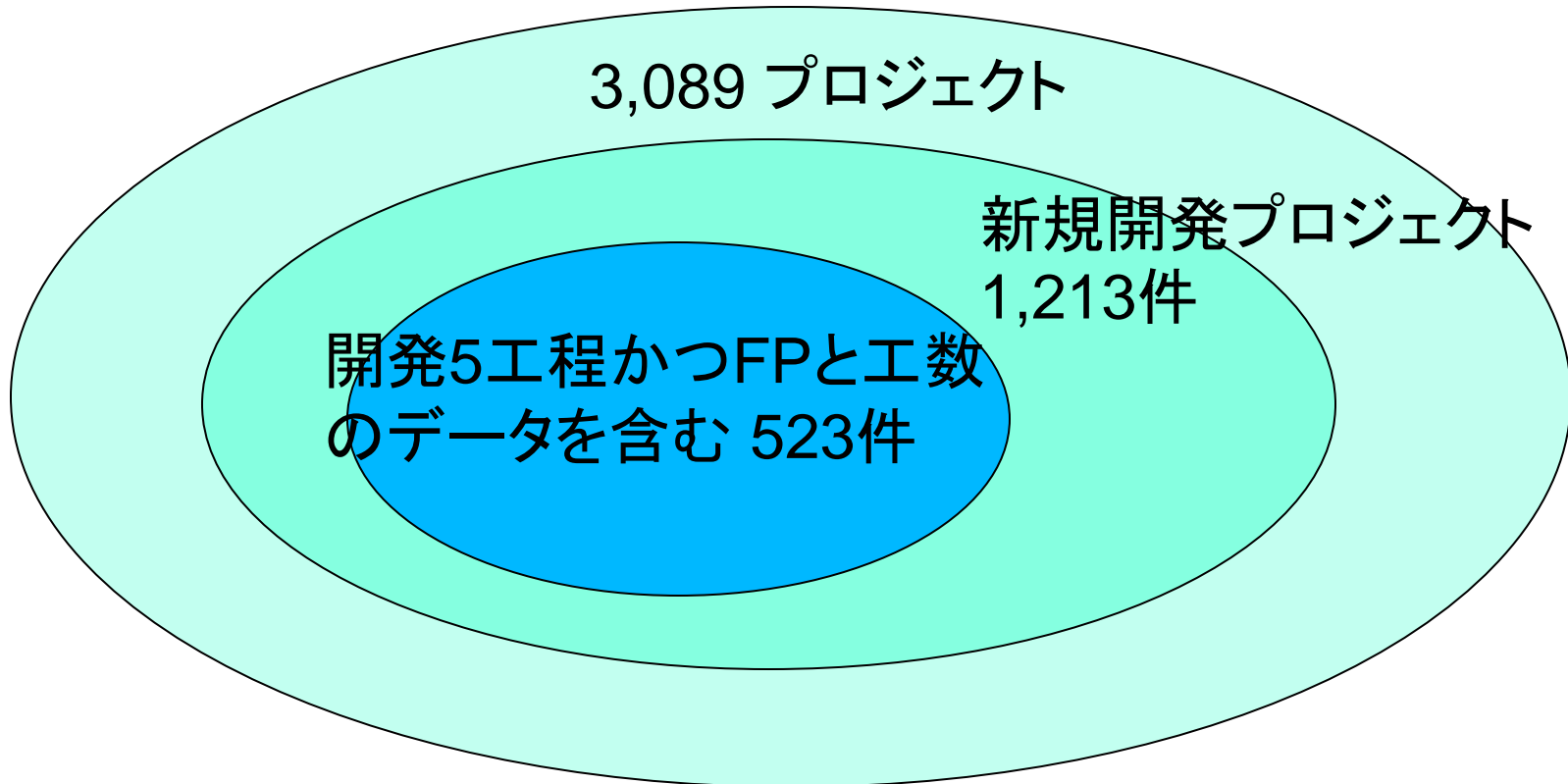
将来的には

- 多くの変数(量的・質的)を対象にした重回帰分析
- パス解析

エンタープライズ系ソフトウェア プロジェクトの生産性への 影響要因の分析

- 背景
- **分析データ**
- データ分析方法
- 分析結果と考察
 - 量的データの回帰分析
 - 質的データの一次元分散分析
 - 質的データの二次元分散分析
 - COCOMO IIとの比較
- まとめ

分析対象プロジェクト



開発5工程: 基本設計, 詳細設計, 製造, 結合テスト, 総合テスト(ベンダ確認)

目的変数と説明変数

(1) 量的変数

- 目的変数: 工数
- 説明変数:
 - 規模 (FP)
 - テストケース数
 - バグ数 (開発時)

量的変数の基本統計量(対数化後)

	工数/FP	工数	FP	テストケース数	バグ数
平均	0.988	3.858	2.870	3.133	2.061
中央値	1	3.819	2.862	3.168	2.043
標準偏差	0.368	0.667	0.489	0.754	0.663
プロジェクト数	523	523	523	324	310

(注)工数の単位は(人時/FP)

(2) 質的変数

- 目的変数: 生産性 = $\frac{\text{工数}}{\text{FP}}$

※ 値が小さくなるほど生産性は高い

- 説明変数: 順序尺度に従う、39件の変数

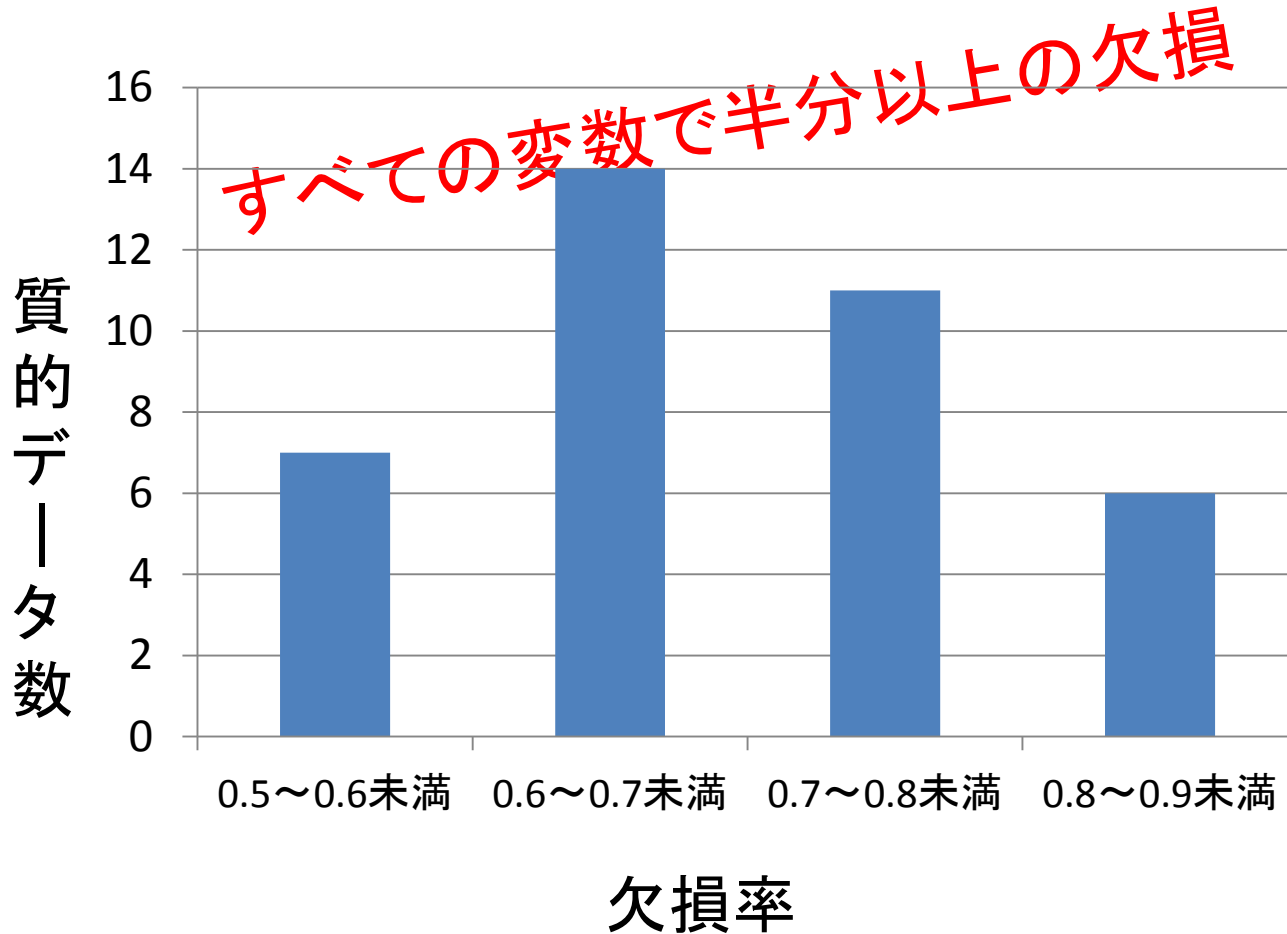
条件: 1) 50件以上のデータがある変数

2) 各変数とも各レベルに10件以上かつその変数に関するデータ総数の15%以上のデータがあること

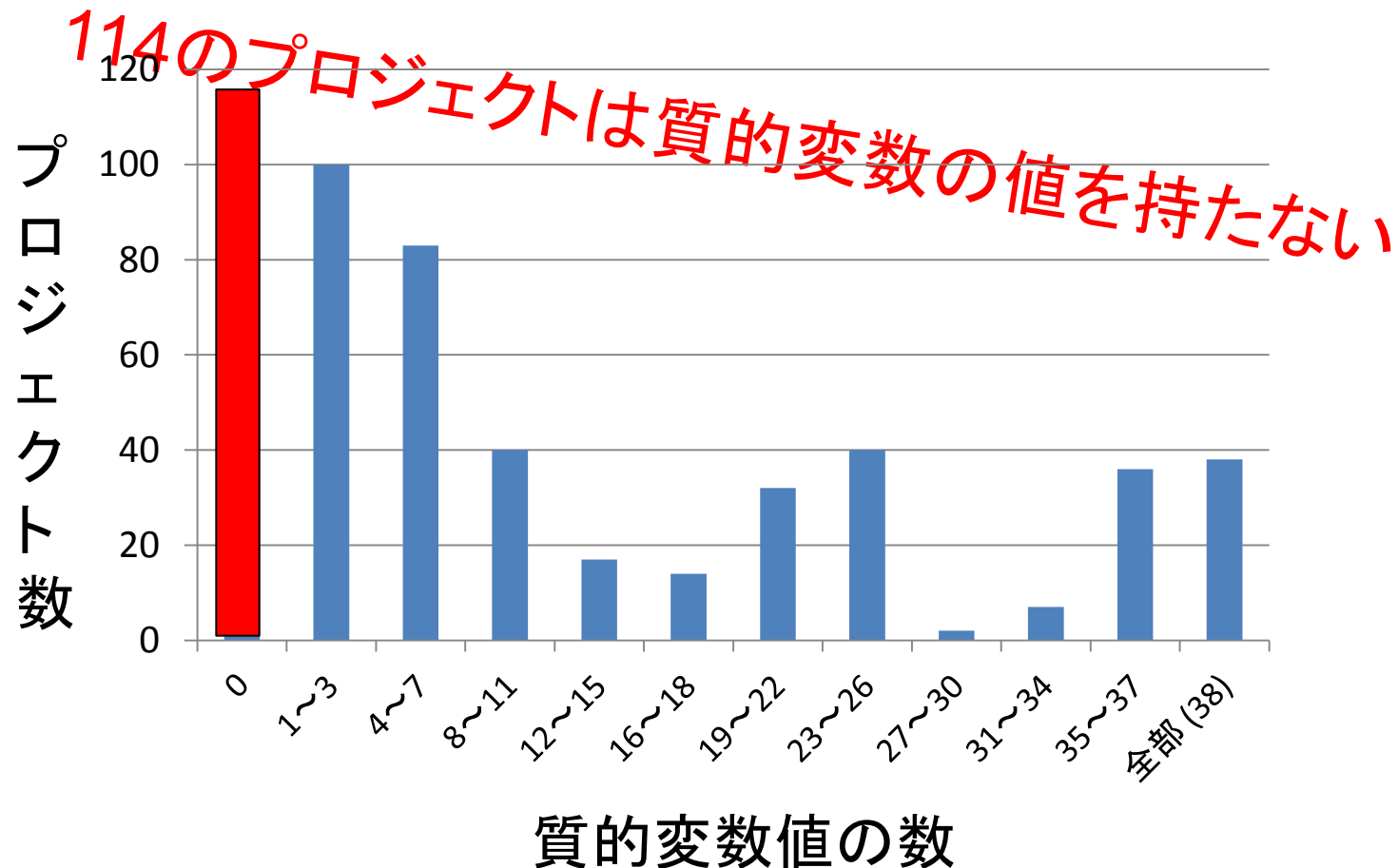
量的説明変数の候補

分類	変数
プロジェクト全体	新技術利用/役割分担_責任所在/達成目標_優先度_明確度合/定量的出荷品質基準_有無/品質保証体制_基本設計
	作業スペース/プロジェクト環境_騒音
計画の評価	計画の評価(コスト/工期/品質)
ツール利用	類似プロジェクト_有無/プロジェクト管理ツール_利用/構成管理ツール利用/設計支援ツール利用/ドキュメント作成ツール利用デバッグ_テストツール利用/コードジェネレータ利用/開発フレームワークの利用
ユーザ側のスキルレベルと関与	ユーザ担当者_要求仕様関与/ユーザ担当者_受け入れ試験関与/ユーザ担当者_システム経験/ユーザ担当者_業務経験/ユーザ担当者_設計内容理解度/ユーザとの役割分担・責任所在_明確度合
要求レベル	要求仕様_明確度合/要求レベル(信頼性/使用性/性能・効率性/移植性/保守性/セキュリティ)/法的規制有無
開発要員のスキルレベル	PMスキル
	要員スキル(業務分野経験/分析・設計経験/言語・ツール利用経験/開発プラットフォーム使用経験)
	テスト体制(スキル/要員数)

質的データにおける欠損値の比率



質的変数に対する値の数



エンタープライズ系ソフトウェア プロジェクトの生産性への 影響要因の分析

- 背景
- 分析データ
- データ分析方法
- 分析結果と考察
 - 量的データの回帰分析
 - 質的データの一次元分散分析
 - 質的データの二次元分散分析
 - COCOMO IIとの比較
- まとめ

データ前処理

(1) 量的変数の変換

生産性を含むすべての変数値は分析の前に対数変換

(2) 質的データの合併

- 3レベル以上の質的変数はすべて2レベルに合併
- 上位2レベルと下位2レベル、あるいは最上位レベルとその他、あるいは最下位レベルとその他など
- 境界はできる限り、2つのレベルのプロジェクト数のバランスがとれるように

データ分析方法

(1) 量的変数 – 重回帰分析

(2) 質的変数

- 一次元分散分析

-- ウェルチの検定 (有意水準: 5%)

-- 2つのレベルの平均が両方とも523プロジェクトの全体平均(0.988)より大きい場合あるいは小さい場合

→「バイアス」がかかっているとして保留

- 二次元分散分析

-- $2 \times 2 = 4$ つのレベルのすべての組み合わせに対して
平均値の差の検定を行い、少なくとも3つで有意差があるものだけを取り出す

エンタープライズ系ソフトウェア プロジェクトの生産性への 影響要因の分析

- 背景
- 分析データ
- データ分析方法
- 分析結果と考察
 - 量的データの回帰分析
 - 質的データの一次元分散分析
 - 質的データの二次元分散分析
 - COCOMO IIとの比較
- まとめ

量的変数に対する回帰分析結果

説明変数の数	回帰係数			定数	重相関係数	補正後決定係数	プロジェクト数
	FP	テストケース数	バグ数				
1	1.147	-	-	0.566	0.841	0.706	523
	-	0.556	-	2.237	0.653	0.425	324
	-	-	0.686	2.575	0.706	0.497	310
2	0.893	0.230	-	0.637	0.855	0.730	324
	0.850	-	0.276	0.906	0.844	0.711	310
	-	0.278	0.483	2.108	0.772	0.594	288
3	0.746	0.193	0.182	0.797	0.868	0.750	288

量的変数に対する回帰分析結果の意味

$$\log E = 0.746 \log S + 0.193 \log T + 0.182 \log B + 0.797$$



$$\frac{E}{S} = 6.26 \times S^{\sim \frac{1}{8}} \left(\frac{T}{S}\right)^{\sim \frac{1}{5} \cdot 0.193} \left(\frac{B}{S}\right)^{\sim \frac{1}{5} \cdot 0.182}$$

補正後決定係数: 70.6% → 75.0%

※ E: 工数, T: テストケース数, B: バグ数,
E/S: 生産性, T/S: テストケース密度, B/S: バグ密度

エンタープライズ系ソフトウェア プロジェクトの生産性への 影響要因の分析

- 背景
- 分析データ
- データ分析方法
- 分析結果と考察
 - 量的データの回帰分析
 - 質的データの一次元分散分析
 - 質的データの二次元分散分析
 - COCOMO IIとの比較
- まとめ

ふたつの観点

- a) それぞれのレベルに対する典型的プロジェクト(*)の生産性比率の大きさは?
- b) 生産性に影響を与えるという説明変数は文献や自分の経験に照らし合わせて妥当か？

(*) 典型的なプロジェクトとは、規模・工数・生産性に関して、対数変換後の平均値を対数逆変換した値をもつプロジェクトのこと。各値はほぼ対数変換前の中央値に等しい。

プロジェクト全体に対する分析結果

変数	レベル	プロジェクト数	生産性*		典型的プロジェクト	
			平均	分散	生産性**	生産性比率
役割分担_責任所在	非常に明確	84	0.820	0.147	6.61	1.71
	概ね明確 + やや不明確 + 不明確	130	1.053	0.151	11.31	
達成目標_優先度_明確度合	非常に明確	70	0.754	0.128	5.67	1.88
	概ね明確 + やや不明確 + 不明確	121	1.029	0.152	10.68	
作業スペース	レベル: a + b (広い)	89	0.798	0.118	6.28	1.56
	レベル: c + d (狭い)	66	0.991	0.208	9.80	
品質保証体制_基本設計	プロジェクトメンバが実施	125	0.985	0.136	9.65	1.84
	品質保証の専門スタッフが実施	59	1.249	0.121	17.73	

*対数変換後の値, **単位は、人時／FP

プロジェクト全体に対する知見

- 役割分担_責任所在が非常に明確に定義
- 達成目標_優先度が非常に明確に定義
- 作業スペースが十分に広い
 - 開発者が物理的なストレスや心理的な混乱を招くことなく作業できる環境
 - 高い生産性が期待できる
- 設計仕様書の品質保証は品質保証の専門スタッフよりプロジェクトメンバが行った方が生産性が高い
 - 教科書の記載内容や文献での報告と異なる
 - 品質(信頼性)の観点から分析すると逆の結果？

ツール利用に対する分析結果

変数	レベル	プロジェクト数	生産性		典型的プロジェクト	
			平均	分散	生産性	生産性比率
類似プロジェクト _有無	あり	54	1.009	0.165	10.21	1.47
	なし	57	0.843	0.165	6.96	
プロジェクト管理 ツール利用	あり	111	1.004	0.181	10.09	1.63
	なし	64	0.791	0.110	6.19	
ドキュメント生成 ツール利用	あり	60	0.653	0.109	4.50	2.21
	なし	93	0.998	0.133	9.95	
デバッグ_テスト ツール利用	あり	52	1.003	0.208	10.07	1.58
	なし	99	0.806	0.126	6.39	
開発フレーム ワークの利用	あり	91	0.923	0.158	8.37	1.40
	なし	75	1.070	0.156	11.75	

ツール利用に関する知見

- ドキュメント作成ツールの利用
- 開発フレームワークの利用
- 高い生産性(妥当な結果)
- 類似プロジェクトの利用
- プロジェクト管理ツールの利用
- デバッグ_テストツールの利用
- 低い生産性(期待と異なる結果)
- これらは品質(信頼性)向上に寄与しているかもしれないので、さらなる検討が必要

その他の分類に対する分析結果

分類	変数	レベル	プロジェクト数	生産性		典型的プロジェクト	
				平均	分散	生産性	生産性比率
ユーザ側のスキルレベルと関与	ユーザ担当者_要求仕様関与	十分に関与 + 概ね関与	132	0.917	0.162	8.27	1.34
		関与が不十分 + 未関与	81	1.043	0.155	11.05	
要求レベル	要求レベル_信頼性	極めて高い + 高い	81	1.016	0.194	10.38	1.85
		中位 + 低い	87	0.750	0.101	5.62	
	要求レベル_セキュリティ	極めて高い + 高い	64	1.128	0.158	13.43	2.85
		中位 + 低い	89	0.672	0.074	4.70	
開発担当者のスキルレベル	PMスキル*	レベル 5, 6, 7 (高いレベル)	58	1.088	0.195	12.25	1.81
		レベル 3, 4 (低いレベル)	108	0.831	0.140	6.77	

* ITスキル標準 (V2.0以降) の職種「プロジェクトマネジメントレベル」

他の分類における知見

- ユーザ担当者の要求仕様書への関与が不十分
 - ➔ 低い生産性(妥当な結果)
- セキュリティや信頼性への要求レベルが高い
 - ➔ 低い生産性(妥当な結果)
- 高いスキルのPMの方が低いスキルのPMより典型的プロジェクトで1.81倍生産性が低い
 - ➔ 常識とは異なる結果
 - ➔ さらなる分析が必要

スキルの高いPMと低いPMが担当するプロジェクトの比較

	FP		テストケース密度		バグ密度	
	高い	低い	高い	低い	高い	低い
PMスキル	高い	低い	高い	低い	高い	低い
平均*	3.114	2.882	0.303	0.047	-0.884	-0.932
分散*	0.221	0.199	0.166	0.388	0.543	0.396
プロジェクト数	58	108	40	65	40	65
P 値	0.3%		1.2%		18.2%	
典型的プロジェクトにおける比率	1.71		1.80		1.12	

* 対数変換後の値

PMスキルレベルと他の説明変数の間に 相関があるか？(カイ二乗検定)

		ユーザ担当者_ 要求仕様関与		要求レベル_ 信頼性		要求レベル_ セキュリティ	
		十分に関与 + 概ね関与	関与が 不十分 + 未関与	極めて 高い + 高い	中位 + 低い	極めて 高い + 高い	中位 + 低い
PM スキル	高い	40 >> 4	31 >> 12	26 >> 17			
	低い	78	27	47	58	37	67
P 値		3.9%		0.4%		1.0%	

高いスキルをもつPMの特徴

大規模かつ要求レベルの高いプロジェクトを担当

→ 品質確保のために低いスキルのPMよりも
規模あたりのテストケース数を多くする
(テストケース密度を高くする)などの手を打つため
生産性が低下

説明変数単独での生産性比率

- 要求レベル_セキュリティ: 2.85
- ドキュメント作成ツールの利用: 2.21
- 上記以外の説明変数は1.9以下

→ほとんどの説明変数単独での生産性比率は
2.0以下

エンタープライズ系ソフトウェア プロジェクトの生産性への 影響要因の分析

- 背景
- 分析データ
- データ分析方法
- 分析結果と考察
 - 量的データの回帰分析
 - 質的データの一次元分散分析
 - 質的データの二次元分散分析
 - COCOMO IIとの比較
- まとめ

二元配置の分散分析

相乗効果を与える組み合わせ				プロジェクト数	生産性		生産性比率*
変数	レベル	変数	レベル		平均	分散	
要求レベル_セキュリティ	極めて高い+高い	作業スペース	レベル: c + d (狭い)	34	1.264	0.146	3.48
				110	0.722	0.086	
		開発フレームワークの利用	なし	30	1.291	0.091	3.36
				77	0.765	0.101	
		役割分担_責任所在	概ね明確 + やや不明確 + 不明確	41	1.210	0.130	3.06
				109	0.724	0.092	

*典型的プロジェクトでの比率

二元配置の分散分析(続き)

相乗効果を与える組み合わせ				プロジェクト数	生産性		生産性比率
変数	レベル	変数	レベル		平均	分散	
要求レベル_信頼性	極めて高い +高い	達成目標_優先度_明確度合	概ね明確 + やや不明確 + 不明確	45	1.197	0.123	3.07
				108	0.711	0.098	
		役割分担_責任所在	概ね明確 + やや不明確 + 不明確	42	1.179	0.165	2.71
				111	0.746	0.115	
		類似プロジェクトの有無	なし	17	1.091	0.176	2.22
				70	0.744	0.114	

二元配置の分散分析(続き)

相乗効果を与える組み合わせ				プロジェクト数	生産性		生産性比率
変数	レベル	変数	レベル		平均	分散	
作業スペース	レベル: c + d (狭い)	開発フレームワークの利用	なし	25	1.299	0.125	3.09
				80	0.809	0.132	
		達成目標_優先度_明確度合	概ね明確 + やや不明確 + 不明確	47	1.138	0.183	2.35
				108	0.768	0.107	
プロジェクト管理ツールの利用	あり	ドキュメント作成ツールの利用	なし	50	1.124	0.117	2.48
				100	0.729	0.106	
		達成目標_優先度_明確度合	概ね明確 + やや不明確 + 不明確	46	1.048	0.138	1.98
				98	0.75	0.122	

生産性比率

- 10件の組み合わせのうち
 - 5件の生産性比率が3以上
 - 4件の生産性比率が2~3
- ➔説明変数単独の場合よりも生産性比率が大きい
- 要求レベル_セキュリティが高いソフトウェアを狭い作業スペースで開発すると、それ以外の場合よりも生産性は3.48倍低下
- 要求レベル_セキュリティが高いソフトウェアを開発フレームワークを利用せずに開発すると、それ以外の場合よりも生産性は3.36倍低下

教訓

- すべての相乗効果は生産性低下に作用
- 最も顕著な相乗効果の組み合わせ:

「セキュリティや信頼性の要求条件が高いソフトウェアを開発する場合、役割分担_責任所在や達成目標_優先度をそれぞれ文書で明確に定義し、作業スペースも十分広くとることが重要である。そうしないと、生産性の極端な低下を招く恐れが高くなる。

「セキュリティの要求条件が高いソフトウェアを開発する場合に生産性の低下を招かないようにするためには開発フレームワークの利用が大切である。」

エンタープライズ系ソフトウェア プロジェクトの生産性への 影響要因の分析

- 背景
- 分析データ
- データ分析方法
- 分析結果と考察
 - 量的データの回帰分析
 - 質的データの一次元分散分析
 - 質的データの二次元分散分析
 - COCOMO IIとの比較
- まとめ

SECデータの分析結果とCOCOMO IIのスケールファクタ／コストドライバの比較

- プロジェクト全体：それぞれに複数要因があるが対応せず
- ツール利用：COCOMO II では1つの要因のみ、SECデータでは詳細化した4つの要因
- ユーザ側のスキルレベルと関与：SECデータでは1つの要因と3つの可能性、COCOMO II にはなし
- 要求レベル：COCOMO II で8つ、SECデータで2つ（+3つの可能性）、よく対応している要因がある
- 開発担当者のスキルレベル：SECデータでは1件（+1件の可能性）、COCOMO II では5件の要因

SECデータの分析結果とCOCOMO II のコストドライバの比較(抜粋)

分類	SECデータの分析結果		類似度 **	COCOMO II	
	質的説明変数の候補	典型的プロジェクト の生産性比率*		スケールファクタと コストドライバ***	生産性 の範囲
ツール利用	類似プロジェクトの有無	1.47	~	先例性†	1.33
	開発フレームワークの利用	1.40	~	ツール活用度	1.50
	プロジェクト管理ツール利用	1.63			
	構成管理ツール利用	#1.56			
	設計支援ツール利用	++			
	ドキュメント作成ツール利用	2.21			
	デバッグ_テストツール利用	1.58			
	コードジェネレータ利用	-			
要求レベル	要求レベル_信頼性	1.85			
	要求レベル_セキュリティ	2.85	~	製品の複雑さ	2.38
	要求レベル_性能・効率性	#1.36	~	実行時間制約	1.63
開発担当者の スキルレ ベル	要員スキル_業務分野経験	-	~	アプリケーション経 験	1.51
	要員スキル_開発プラットフォームの使用経験	#1.43	=	プラットフォーム経 験	1.40
	要員スキル_言語・ツール利用 経験	-	=	言語とツールの経験	1.43

* 無印は有意水準5%で有意, “#”は5%で有意であるがバイアスあり, “++”は10%で有意.

** “=”はほとんど等しい, “~”は似ている. *** †はスケールファクタ

 :生産性向上  :生産性低下

まとめ

- SECデータベースには世界で公開されている他のデータベースに比べて、非常に多くの量的及び質的変数をもつ3000件以上のエンタプライズ系ソフトウェアプロジェクトのデータがある
- SECデータには大量の欠損値が含まれるためこれをもとにコストモデルを構築するのは容易ではない
- 最初の試みとして523件の開発5工程の揃った新規開発プロジェクトに対して重回帰分析と分散分析を試みた結果、いくつかの知見を得ることができた

まとめ(分析結果)

- (1) 生産性は、規模(FP)の8乗根、テストケース密度の5乗根、バグ密度の5乗根に比例して低下
- (2) 13件の生産性への影響要因(質的変数単独)
最も影響度の大きいもののトップ4:
 - 要求レベル_セキュリティ
 - ドキュメント作成ツールの利用
 - 達成目標_優先度_明確度合
 - 要求レベル_信頼性

まとめ(分析結果)(続き)

(3) 典型的プロジェクトの生産性比率(質的要因単独)

- 要求レベル_セキュリティ: 2.85
- ドキュメント作成ツール利用: 2.21
- それ以外: 1.9 以下。

(4) PMのスキルレベルと生産性

- スキルの高いPMの方が低いPMより低い生産性
- 高いスキルのPMは大規模で要求レベルの高いソフトウェアを開発

→ 品質確保のためにより多くのテストを実施

まとめ(分析結果)(続き)

(5) 相乗効果を起こす10件の影響要因の組み合わせ

最も特記的な組み合わせ:

- 要求レベルの高いソフトウェアを狭いスペース環境で開発する→それ以外の場合に比べて典型的プロジェクトと比較して3.48倍生産性が低下
- 要求レベルの高いソフトウェアを開発フレームワークを用いないで開発→それ以外の場合に比べて典型的プロジェクトと比較して3.36倍生産性が低下

今後の課題

- 多数の変数を加えた重回帰分析
- パス解析
- 品質モデル
- . . .

ご清聴ありがとうございました