

ソフトウェア見積もり工数に
影響を与える因子に関する研究

筑波大学審査学位論文（博士）

2018

志 田 剛

筑波大学大学院
ビジネス科学研究科 企業科学専攻

ソフトウェア見積もり工数に 影響を与える因子に関する研究

目次

第1章 緒論	1
第2章 ソフトウェア開発見積もりの現状	5
2.1 ソフトウェア見積もりの重要性	5
2.2 ソフトウェア見積もりの問題点	7
2.3 ソフトウェア見積もりの流れ	11
2.4 ソフトウェアによる訴訟問題	12
2.5 ソフトウェア見積もり技法	15
2.5.1 類推法	16
2.5.1.1 類推法の特徴	16
2.5.1.2 類推法の利点	19
2.5.1.3 類推法の問題点	20
2.5.2 積算法(ボトムアップ法)	20
2.5.2.1 積算法の特徴	20
2.5.2.2 積算法の問題点	22
2.6 係数モデル見積もり	23
2.6.1 LOC 法	23
2.6.1.1 LOC 法の特徴	23
2.6.2 ファンクションポイント法	27
2.6.2.1 ファンクションポイント法の特徴	27
2.6.2.2 ファンクションポイント法の問題点	28
2.6.3 CoBRA 法	29
2.6.3.1 CoBRA 法の特徴	29

2. 6. 3. 2 CoBRA 法モデル	29
2. 6. 4 COCOMO II	32
2. 6. 5 ソフトウェア見積もり要因に関する研究	33
2. 6. 6 その他分析手法	35
2. 7 結言	36
第3章 統計手法によるソフトウェア開発モデル	38
3. 1 データの収集	40
3. 2 生産性変動要因の抽出	40
3. 2. 1 生産性変動要因とは	40
3. 2. 2 生産性変動要因の設定	42
3. 3 重回帰分析による実験	47
3. 3. 1 重回帰分析の概要	47
3. 3. 2 見積もり工数と生産性変動要因の重回帰分析	53
3. 4 業種ごとによる分析	63
3. 4. 1 業種の分類	63
3. 4. 1 業種別による分析	65
3. 4. 2 実行結果の考察	66
3. 5 結言	71
第4章 因子分析による因子の抽出	73
4. 1 探索的因子分析の概要	74
4. 2 実験データ	76
4. 3 探索的因子分析の実験	76
4. 3. 1 因子数の推定	76
4. 3. 2 因子パターン(因子負荷)の推定	78
4. 3. 3 因子の回転方法	81
4. 3. 4 カテゴリカル因子におけ母数の推定	83
4. 4 探索的因子分析の実験の評価	88
4. 5 探索的因子分析の考察	95
4. 6 確認的因子分析	96

4.6.1 モデルの仮説	98
4.6.2 母数推定	104
4.6.3 モデル適合度の評価	107
4.6.4 パス図の描画	111
4.6.5 構成概念スコア	112
4.7 因子分析による考察	114
4.8 結言	116
第5章 ソフトウェア開発見積もりに影響を与える因子モデルを利用した業種別相違性...	118
5.1 業種別によるソフトウェア見積もり因子の比較	119
5.2 対象データ	120
5.3 評価実験	121
5.3.1 多母集団同時分析	121
5.3.2 クラスタ分析	129
5.4 実験結果の考察	132
5.5 結言	134
第6章 結論	135
参考文献	141

論文概要

博士（システムズ・マネジメント）

ソフトウェア見積もり工数に 影響を与える因子に関する研究

筑波大学大学院
ビジネス科学研究科
企業科学専攻
志田 剛

近年、AI ブームに伴い企業が IT の投資にかかるコストは、年々増加している。特に建築・土木業界では他業種と比較しても、IT に対する予算が増加している。これは 2020 年の東京オリンピックの開催を受け、深刻化している人手不足を解消するために IT 活用を進め、生産性向上を図りたいとの狙いがあると考えられる。このように日本の人口の減少に伴い AI に投資しようとしている企業が増加していると考えられる。

また、業種ごとの KGI(Key Goal Indicator)は異なっているが、企業はアプリケーション分野に関して、BtoB(Business To Business)や BtoC(Business To Customer)向けに対して、よりサービスのよい、アプリケーションを期待している。それに伴い、ソフトウェアの開発も複雑化され、ソフトウェア開発ベンダが見積もるソフトウェア開発見積もりも複雑化している。

IT 発注企業は、投資するシステムについての費用対効果(ROI : Return On Investment)などの意思決定を重要視している。そのため、IT 発注企業の経営層にとっては、そのシステムに対しての投資が妥当かどうか、的確に把握出来ることが重要な要素となっている。

一方、IT 発注企業からソフトウェア開発の見積もりの依頼を受けたソフトウェア開発ベンダは、システムを利用する IT 発注企業からの要望に対して、ソフトウェア開発見積もりを行う。しかしながら、ソフトウェア開発ベンダは、初期の段階で正確な見積もりを作成することは非常に難しく、初期のソフトウェア開発見積もりは、下流工程でのソフトウェア開発見積もりと比べ大きく外れると言われている。特に初期の段階ではソフトウェア開発見積もりの生産性に影響を及ぼす要因のうち、不確定な要因が多く存在している。そのため、その不確定な要因をソフトウェア開発見積もりの工数に含めることは難しく、それが課題となっている。IT 発注企業はソフトウェア開発ベンダからのソフトウェア見積もりから、予算を確定する。そのため、ソフトウェア開発ベンダが当初のソフトウェア見積もりより、実績工数を大きく外した場合、それがクレームとなり、場合によっては訴訟となるケースが発生している。このような情勢を鑑みると、IT 発注企業とソフトウェア開発ベンダの双方にとって、ソフトウェア見積もりは、経営的に非常に重要となっている。

システムの開発手法の 1 つであるウォーターフォール型モデル開発は、日本の企業でこの手法を取り入れているソフトウェア開発ベンダが多い。それは、上流工程からしっかりと要件を決め、外部設計へ落とし込んでいく。そしてその外部設計内容に問題がなければ内部設計、実装へと進めていく。この開発手法を採用するのはソフトウェア開発ベンダが、古くからのこの手法に慣れていると考えられる。そして、ソフトウェア開発ベンダが使用する規模見積もり技法として、Line Of Code(以下 LOC)法や Function Point 法(以下、FP 法)など機

能や実装を考慮した規模見積もりが主流となっている。しかしながら、上流工程で LOC 法や FP 法を用いた規模見積もりには限界があり、多くのソフトウェア開発見積もりの誤差が発生している。そこでこの課題に対して、LOC 法や FP 法以外でソフトウェア見積もり工数に影響を与えている要因を把握することでソフトウェア開発見積もりの誤差を縮めることができるのではないかと考えた。そこで生産性変動要因に着目した。その生産性変動要因を数値化し、統計手法を用いることで、ソフトウェア開発見積もりの工数以外に影響を与えている要因や因子が何かを見つけることが必要であると考えた。

先行研究では COCOMO II や CoBRA 法という生産性変動要因に着目したソフトウェア見積もり技法がいくつか存在する。これら見積もり技法では生産性変動要因から見積もり工数を算出する技法である。ところがどの生産性変動要因がソフトウェア見積もりに影響があるのかについての研究は少なく、また業種ごとにどの生産性変動要因がソフトウェア見積もりに影響があるのかについての研究は、ほとんどないことが分かった。

そこで本研究では、生産性変動要因に着目し、統計手法を用いてモデルを作成した。そしてそのモデルから特に影響している生産性変動要因を抽出した。また生産性変動要因から共通している因子モデルを作成し、各生産性変動要因に共通する因子を抽出した。そしてその抽出した因子から業種ごとに因子を比較し、ソフトウェア見積もりに影響のある生産性変動要因を提案した。

本論文では、まず、目的変数を見積もり工数、説明変数を 10 の生産性変動要因とした重回帰分析を実施し、見積もりに影響を与える生産性変動要因を抽出した。また、9 業種からそれぞれの業種に影響を与える生産性変動要因が何かについても業種別で明確にした。この結果からどの業種がどのような生産性変動要因に影響があるのかについて明確にすることが出来た。次に、観測変数を生産性変動要因とし、探索的因子分析を実施した。その結果、生産性変動要因から潜在的な 2 つの因子を見つけ出すことが出来た。この抽出した因子の 1 つ目は「上流工程の管理能力」、2 つ目の因子は「開発の容易性」であることが分かった。そして、その見つけ出した 2 つの因子の他に考えられる 5 つのモデルを構造化し、共分散構造分析を行った。その結果、最もあてはまりのよいモデルを見つけ出すことが出来た。

ところがこの因子分析では9業種を含めた単一母集団での分析をしており、各業種での多母集団間の比較は実施していなかった。したがって、単一母集団でのパラメータは多母集団間で必ずしも一致するとは限らないことになる。つまり2因子を抽出したモデルには複数の母集団のパラメータが異なるという仮説を検証する必要がある、業種別に分析する方法を考えた。したがって、業種別にそれぞれ分析した場合、全てのパラメータが異なるという仮説をおくことになる。多母集団を同時に分析することは全てのパラメータが同じである仮説を置いていることになるため、母集団間のパラメータを比較する場合、比較の枠組みを共通化させた上で比較したい仮説を検証する必要がある。そこで9業種の母集団のパラメータが全て等しいと仮定したモデルと等しくないと仮定したモデルの適合度を比較し、母集団間のパラメータを統計的に分析した。その結果から、業種ごとの2つの因子の分散や因子間相関係数を抽出した。そして業種の特性を抽出することが出来た。また、業種間でどのような分類となるかを見つける為、クラスタ分析にて業種群を分類した。その結果、BtoBの業種とBtoCとの業種に分けることが出来た。また因子の分散や因子相関係数からそれぞれの業種間での相違性について判別することが出来た。

これらにより、ソフトウェア見積もりに影響を与える生産性変動要因から、各ソフトウェア開発ベンダが見積もりを実施する際、予め業種ごとにソフトウェア開発見積もりに影響を与える生産性変動要因が何かを見定めることが可能となった。また、ソフトウェア開発ベンダはIT発注企業に対して、これら見積もりに影響を与える生産性変動要因や因子を予め伝えることが可能となった。つまり、それらの内容をソフトウェア開発ベンダがIT発注企業に伝えることで、お互いが納得いく見積もりが出来ることになる。その結果、トラブルの発生を抑制することができると考えられる。つまり、ソフトウェア開発ベンダはIT発注企業に対し、見積もりの妥当性を判断する手段としてソフトウェア開発見積もりに影響を与える生産性変動要因や因子を利用することが有効であると言えることになる。

第 1 章

緒論

近年、IoT や AI/機械学習が注目されるに伴い企業が IT の投資にかかるコストは業種ごとに異なっており、その投資額は年々増加している。日本情報システム・ユーザ協会(JUAS)による『企業 IT 動向調査 2018』[JUAS 18]では、東証一部上場企業向けに 2018 年度ユーザ企業を対象とした IT 予算に関する投資額の増減について調査した。その結果、2018 年のユーザ企業の IT 投資は 2017 年に比べて「増加」と回答した企業は全体の 40.7%と過去 10 年間で最高となった。特に建築・土木業界では IT 予算を「増やす」割合から「減らす」割合を引いた DI(Diffusion Index)が 27 ポイントと 17.7 ポイントであった 2017 年に比べ 9.3 ポイントを上回った。これは 2020 年の東京オリンピックの開催を受け、深刻化している人手不足を解消するために IT 活用を進め、生産性向上を図りたいとの狙いがあると報告されている。また、ITR 社『IT 投資動向調査 2017』[ITR 16]では、日本企業の業種別に今後見込まれる製品/サービス分野について調査した。インフラ/デバイス分野では、ネットワークの仮想化やサーバの仮想化、ストレージの仮想化など仮想化への投資期待が大きく、製造業や情報通信業では、IoT(Internet of Thing)や M2M(Machine To Machine)への期待が見込まれるという結果が報告されている。IoT とはネットワークを通じて、物の情報を収集する仕組みを示す。また、M2M とは人が介在することなく、機械同士が相互に情報のやりとりを行うことを指しており、機械や情報を収集する機械、機械をコントロールする機械を示す。

一方、アプリケーション分野においては製造業やサービス業では BI/データ分析などの分析アプリケーション、公共や情報通信業ではスマートフォンデバイス向けのアプリ開発、建設・不動産業については経営管理システム、そして金融・保険業ではエンタープライズ・コンテンツ管理への投資が見込まれてい

ると報告されている。つまり業種ごとの KGI (Key Goal Indicator) は異なっているが、企業はアプリケーション分野に関して、BtoB (Business To Business) や BtoC (Business To Customer) 向けに対しても、よりサービスのよい、アプリケーションを期待している。それに伴い、ソフトウェアの開発も複雑化され、ソフトウェア開発ベンダが見積もるソフトウェア開発見積もりも複雑化されている。ソフトウェア開発の見積もり依頼を受けたソフトウェア開発ベンダは、システムを利用する IT 発注企業からの要望に対して、初期の段階で正確な見積もりを作成することは非常に難しく、初期のソフトウェア開発見積もりは、下流工程でのソフトウェア開発見積もりと比べ大きく外れると言われている [JUAS 16]。特に初期の段階ではソフトウェア開発見積もりの生産性に影響を及ぼす要因のうち、不確定な要因が多く存在している。そのため、その不確定な要因をソフトウェア開発見積もりの工数に含めることは難しく、それが課題となっている。

ソフトウェア開発の見積もり工数の技法は歴史が古く、数多くの手法が存在する。その中でも Line Of Code (以下 LOC) 法や Function Point 法 (以下 FP 法) に関しては多くのソフトウェア開発ベンダが利用している。『ソフトウェア開発データ白書 (2016-2017)』 [IPA 16] では、ソフトウェア開発見積もり規模の尺度と種類について報告している。ソフトウェア開発ベンダが使用する規模見積もり方法として、LOC 法が 55% と半分以上を占めている。また、FP 法での規模見積もりが 26% となっており、LOC 法と FP 法合わせると 8 割以上を占めていることになる。ところが、これら上流工程で用いた規模見積もりには限界があり、多くのソフトウェア開発見積もりの誤差が発生している。そこで、見積もり工数以外に要因となり、その課題を解決する方法として、変動要因を使った COCOMO II モデルや CoBRA 法が存在する。これら技法は変動要因について評価してソフトウェア開発見積もりを予測する見積もり技法である。これらの技法を使って、変動要因を考慮した見積もり工数を算出することが可能となった。ところがこれら技法では変動要因をパラメータとして、開発工数の推定値を得るには有効であるが、どの変動要因に影響があるのか。また、変動要因から構成概念を明確にすることは難しく、業種ごとにどの変動要因に影響力が強いのか

について示すことができない。つまり、業種ごとでの相違性について確認することも難しい。

そこで本研究では、ソフトウェア開発ベンダが見積もり工数を算出するにあたり、影響を与える要因を明確にすることを目的とした。具体的には、プログラム構造、人員やプロジェクトの特性、そして生産時の特性など、過去のプロジェクト実績データを使い、統計分析から見積もりの生産性変動要因を因子としたモデル式を導出する。そして、導出したモデル式や統計を用いた分析結果から、これまで IT 発注企業にとってブラックボックスとなっている阻害要因について明示化を行うことを提案した。また、生産性変動要因からの因子を抽出し、抽出した因子ごとに影響のある生産性変動要因を明確にした。そしてその因子モデルから業種ごとの相違性について検討し、類似している業種について見つけ出すことが出来た。

本論文では、まず第 2 章にて、長い歴史をもつ見積もり技法について述べ、それら見積もり技法の特徴と課題について述べる。次に、第 3 章にて、本研究で実施したデータの定義について明記する。その定義したソフトウェア見積もりデータを使い目的変数を見積もり工数、説明変数を生産性変動要因に設定し、重回帰分析を実施する。その重回帰分析から導出したモデルより、影響の高い生産性変動要因が何かを明らかにする。また、より深掘り分析する為に、業種ごとに重回帰分析を実施し、業種ごとに影響のある生産性変動要因が何かについても述べる。第 4 章では、生産性変動要因を観測変数とし、どのような因子が抽出されるのかを確認するため、探索的因子分析を実施し因子数や因子負荷量などの結果について述べる。そして、抽出したモデルの評価として抽出した因子モデルを使い、測定方程式を作成し共分散構造分析から抽出した因子と観測変数から RMSEA や AIC などの適合度指標から評価し、最もあてはまりのよいモデルについて検証する。また、そのモデルに対して、因子間の構成概念スコアや信頼区間を考慮し、その結果について述べる。第 5 章では、前章で抽出した因子モデルを利用して業種ごとに絞り、配置不変モデルや測定不変モデルなどの 5 つのモデルと比較し、制約条件の一番適合するモデルを検証する。そしてその抽出したモデルを使い、多母集団同時分析を実施する。実施した結果を共分散構造分析と同様に、モデルの適合度指標から業種ごとの因子の分散と因

子相関を抽出する．そして抽出した結果からクラスタ分析を実施し，業種の相違性について検証する．最後に第 6 章にて結論を述べる．

第2章

ソフトウェア開発見積もりの現状

本章では，ソフトウェア開発見積もりの重要性と問題点，ソフトウェア見積もりの流れについて示す．そして，ソフトウェア工数の見積もりを失敗したことにより，IT 発注企業がソフトウェア開発ベンダに対し，訴訟を起こした事例について示す．そしてソフトウェア見積もり技法について，これまでの先行研究を概観する．結言において，ソフトウェア見積もりの課題について示す．

2.1 ソフトウェア見積もりの重要性

「見積もり」という言葉を広辞苑で引いてみると，見積もるという言葉は以下の2つが示されている．

1. 目で見えて大体を測る．目分量で測る．
2. 物事のあらましを考え計算して予測を立てる．つもる．概算する

ソフトウェアの工数見積もりでは，上記2.の意味となる．つまり，対象となるソフトウェア開発の全体を頭に描き，事前にプロジェクトの中で投入される可能性のある，あるいは，投入する必要のある工数を予測することになる．

ソフトウェア見積もりは，IT 発注企業やソフトウェア開発ベンダの双方にとって経営的に重要である[IPSEC06]．IT 発注企業はシステムの開発コストの妥当性，適切性，費用対効果(ROI: Return On Investment)などの判断を重要視しており，経営層がシステムに対する投資額について妥当な金額かどうか的確に把握することが大きな課題となっている．一方，ソフトウェア開発ベンダにとって，見積もりがIT 発注企業との開発プロジェクト全体の要素を決めること

になる。ソフトウェア開発の金額が決まった後、WBS (Work Breakdown Structure) や DFD (Data Flow Diagram) [Abdel 86] を作成し、開発規模、開発体制そして作業量などが基本的に決まってくる。つまり、ソフトウェア見積もりはプロジェクトの実施すべき内容や方法まで規定してしまうことになるため、もっとも基本的なデータとして捉えなければならない。概算見積もりでも一旦契約額として見積額が決定してしまうとソフトウェア開発ベンダはその契約額を基準として開始し、開発完了時には妥当な利益を確保する必要がある。また、IT 発注企業からの要望を満足させるソフトウェアを構築しなければならない。つまり、ソフトウェア開発見積もり時の規模や金額がプロジェクト全体のスケジュールや完了までの目標となってくる。今日のシステム開発において、ソフトウェア開発見積もりを誤ることは、ソフトウェア開発ベンダにとってもシステム開発を発注した IT 発注企業にとっても経済的に非常に深刻な結果をもたらしかねず、場合によっては訴訟問題まで発展するケースも発生している。つまりソフトウェア開発の見積もりはシステム開発において非常に重要な要素となっている。

2.2 ソフトウェア見積もりの問題点

2.1 節ではシステムの見積もりは重要であることを示したが、システムに対する投資額を経営層が意思決定するためには、ソフトウェア開発ベンダが開発のプロセスにおける初期の段階でシステム開発全体の規模を見積もる必要がある。システム開発にはいくつかの手法があり、ウォーターフォール型開発やスパイラル型開発、反復型開発、アジャイル開発などいくつかの種類がある。これら手法は、プロジェクトごとにそれぞれの開発スタイルに合った方法を選択し、プロジェクトを進めていく必要がある。その中でもメジャーな手法が、ウォーターフォール型モデルである。図 2-1 は、「ソフトウェア開発データ白書 2014-2015」[IPA 15]より、ソフトウェア開発プロジェクトにおける開発作業の進め方に関するアンケート結果である。

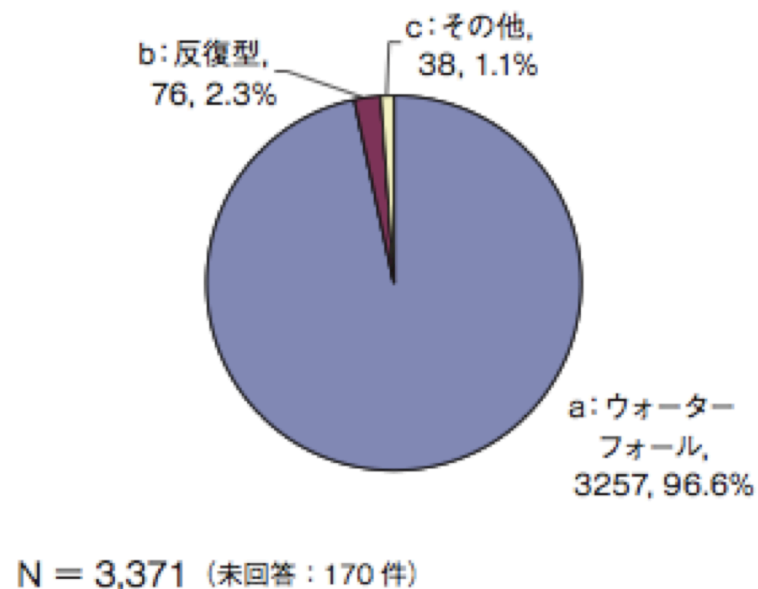


図 2-1 開発ライフサイクルモデル [IPA 15]

その中でウォーターフォール型の開発が 96.6%と多くの企業がこのウォーターフォール型モデルを採用していることが分かる。ウォーターフォール型モデルで開発を進めた場合、プロジェクトの初期段階では不確実な要素が多い。図 2-2 では妥当な見積もり時期と見積もり誤差・リスクについて示している[IPA 05]。

図 2-2 から、上流工程(システム化の方向性, システム化計画, 要件定義)での見積もり時の誤差が非常に大きく、設計から製作の下流工程に向かい見積もり誤差が少なくなることが分かる。これはプロジェクトの計画時には多くの不確定な要素が考えられる。例えば、業務知識の習熟度、開発技術スキルレベル、ユーザの関心度など人的な要因がある。また、システムの要求内容の曖昧さやシステムの複雑性、業務(データ)の複雑さなどのプロダクト要因がある。プロジェクトごとに変動する要因は違うため、それぞれに取り入れた工数を提示する必要がある。そして、ソフトウェア見積もり作業はソフトウェア開発ベンダの担当者が自らの経験や勘を基準にして見積もりを行うため、同じシステムの開発でも担当者によって見積もり結果が異なるという現象が発生する。また、担当者と開発するシステムの内容によって見積もりの精度も一定ではないことが容易に推測できる。

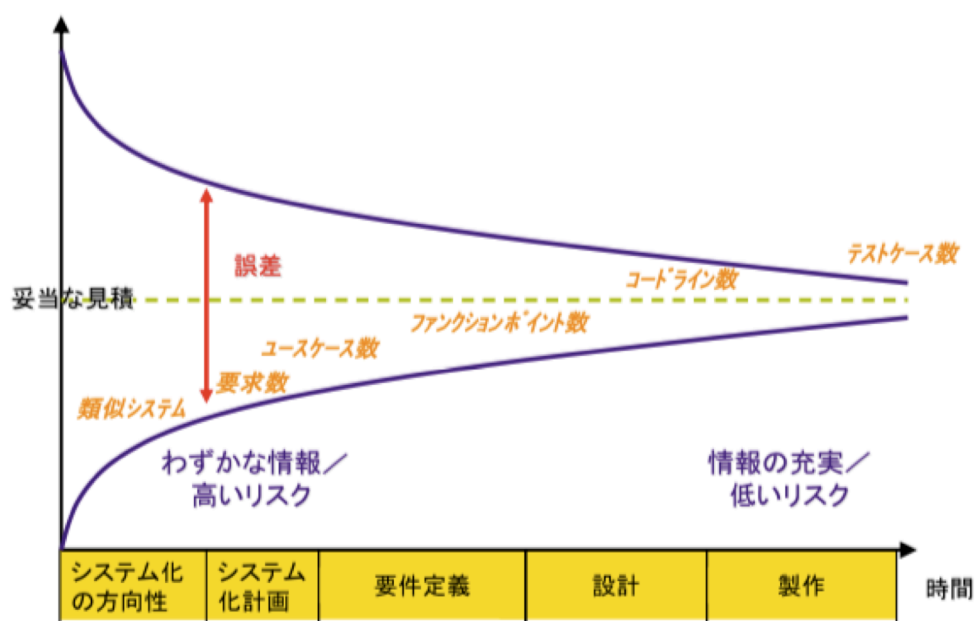


図 2-2 見積もり時期と見積もり誤差・リスク [IPA 05]

それでは一般的に良い見積もりの定義とは何かについて考えてみる。[Conte 86]によると、「プロジェクトのコントロールが適正であれば、見積もりに対してプラスマイナス 10%の正確性を達成することが可能である」。また「良い見積もりのアプローチとは、実績値 75%のケースで誤差が 25%以内に収まる見積もりを提供することである。」と述べている。これら共通することは、プロジェクト開始時の見積もりとプロジェクトが終了となった実績の工数の誤差が少なければ良いものであると定義している。ところがプロジェクトを進める中で、多くのリスク要素が発生する。例えば図 2-3 ではプロジェクトの最初と最後とでは誤差がなしということが前提であるが、途中経過で数々の変動要因が発生している。これはたまたまプロジェクトの開始と終了で見積もり工数が一致したと考えられるが、本来、リスクとして考えた場合には、工数は膨らむことが想定される。

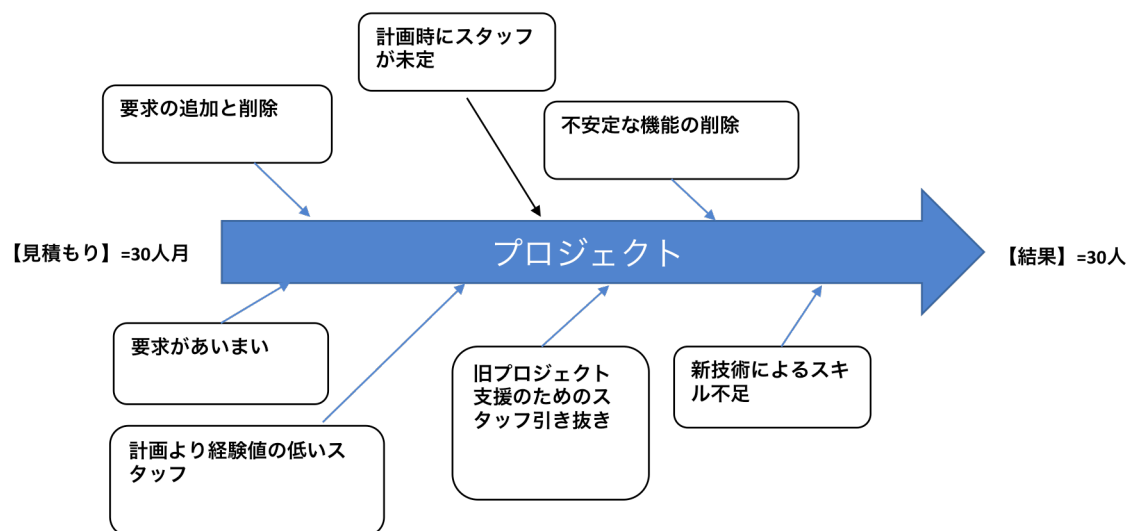


図 2-3 プロジェクトで発生するリスク

見積もりが難しくなっている原因について以下の4つの要因が考えられる。1つ目が、要求仕様があいまいであること。ビジネスのスピードアップに伴い、ソフトウェア開発ベンダは見積もりに当てられる時間が短縮されている。そのため、高いスキル経験が要求されることになる。ところが、ソフトウェア見積もりは、要求仕様が曖昧である計画の早い段階で見積もりを算出しなければならない。つまり、どういうソフトウェアが欲しいのか、何を作りたいのか、はっきりしていないことになる。2つ目が人的要素(ヒューマンファクタ)の影響が大きい。ソフトウェアの開発コストのほとんどが人件費である。目に見えない知的労働の工数を定量的に見積もることは難しい。3つ目が見積もりに関するノウハウが蓄積されていない。見積もりは過去の実績値や経験則が見積もりの根拠となる。新しい技法や経験のない分野での利用が増えており、過去の延長線上では考えられないプロジェクトが増加している。そして4つ目が、ソフトウェアビジネス環境が未成熟である。社会性や公共性の高い分野や戦略性の高いソフトウェア開発が増加し、ハードウェアからソフトウェアへとIT業界の主軸が変わっていった。

これらの問題を解決し、ソフトウェア開発ベンダはIT発注企業に納得してもらえる見積もり方法を定量的に示す必要がある。

2.3 ソフトウェア見積もりの流れ

ソフトウェア見積もりの流れはシステム開発の流れに沿って見積もっていくのが基本である。ウォーターフォール型モデルの場合、図 2-4 のようにアナリストが要求分析から始まり、要件定義を明確にしていく。そして、要件定義で作成した要件定義書を input とし、基本設計を作成していく。次に、完成した基本設計書を input として、詳細設計書を作成する。次に完成した詳細設計を input として、実装へと進んで行く。一旦実装まで完了すると、その後にテストフェーズへ進んでいく。テストフェーズ(図 2-4 右側)では要件や設計、実装の成果物を input とし、それぞれの局面での成果物を元にテストフェーズへ進んでいく。例えば、図 2-4 が示す通り、単体テストでは、詳細設計で作成した成果物を input とし、単体テスト設計書を作成し、単体テストを実施していく。最終的には要求分析した内容を input として、ユーザの受入テストまで進み、要件や仕様の認識違い、実装のバグがなくなったと判断した後、システムのリリースが開始される。つまりウォーターフォール型モデルの場合、この V 字モデルに沿ってソフトウェアの見積もりの必要があり、見積もりに必要なタスクに抜けや漏れがないようにリスクも含めて考えていく必要がある。

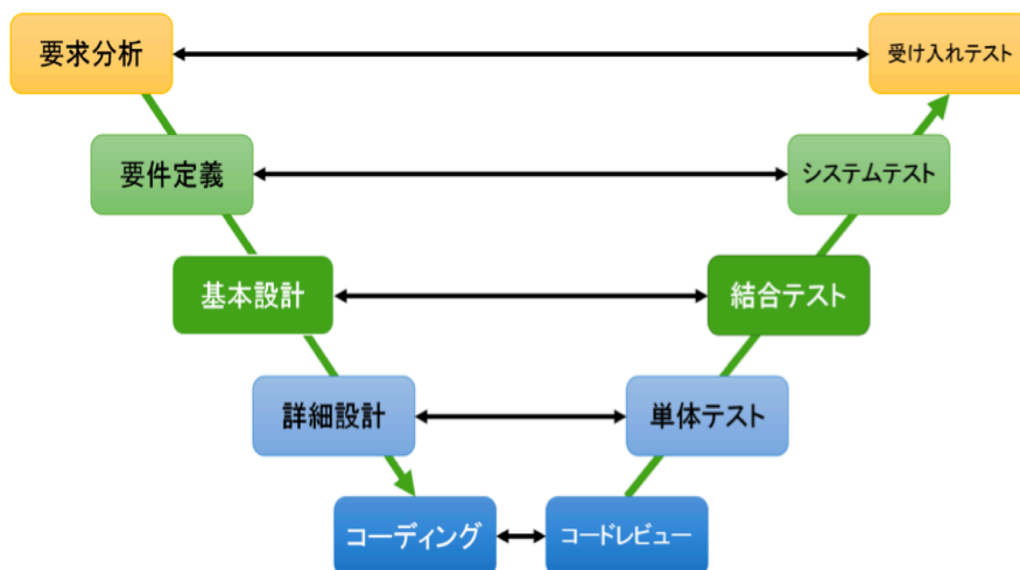


図 2-4 ウォーターフォール型モデルにおける開発の流れ(V 字モデル)

2.4 ソフトウェアによる訴訟問題

ソフトウェアの見積もりが大きく外れた場合、後の工程で実作業時間が足りなくなり、ソフトウェア開発ベンダは IT 発注企業へ別途追加金額を請求することになる。しかし IT 発注企業は、当初のソフトウェア見積もりで社内稟議を通すことで、追加予算をとることが非常に難しくなる。その結果、IT 発注企業とソフトウェア開発ベンダ間で訴訟問題まで発展するケースが発生している。以下の事例[Nikkei 18]は IT 発注企業がソフトウェア開発ベンダへ要件の認識の違いから訴訟に発展したケースである。

[27 億円の賠償巡り新たな IT 裁判始まる、文化シャッタが提訴]

アルミ建材大手の文化シャッタ社が、販売管理システムの開発が頓挫した責任は委託先の日本 IBM 社にあるとして、2017 年 11 月、約 27 億 4000 万円の損害賠償を求めて、東京地方裁判所へ日本 IBM 社を提訴した。同社は 2017 年度第 2 四半期決算(2017 年 7-10 月)で、販売管理システムの開発継続断念に伴う 17 億 4500 万円の特別損失を計上済み。同システムの開発委託で日本 IBM 社に支払った費用などの返還を求めている。

文化シャッタ社が既存の販売管理システムを刷新するプロジェクトを始めたのは 2015 年 3 月、文化シャッタ社は日本 IBM 社に RFP(Request For Proposal)の作成を委託。その RFP に基づき複数のソフトウェア開発ベンダから提案を受けたうえで、日本 IBM 社をシステム構築の委託先として選定した。

日本 IBM 社の提案は、販売管理システムの構築に ERP（統合基幹業務システム）などのパッケージを使わず、米セールスフォース・ドットコム

(Salesforce.com) 社のクラウド開発基盤「Salesforce1 Platform」を利用してシステムをスクラッチ(手作り)するというものだった。稼働時期は約 1 年半後の 2016 年 7 月、総開発費用は約 12 億 3500 万円を見込んでいた。

両社は当初、アジャイル開発とウォーターフォール型開発の併用によるシステム構築を目指していたが、途中からウォーターフォール型開発のみに方針を転換。要件定義、設計・開発、システムテストと工程を進めた。プロジェクトは当初

予定より数カ月遅れ、両社は稼働時期を 2016 年 7 月から 11 月に延期。新たな日程を前提に、同年 8 月にユーザ受け入れテストを始めた。ここで問題が噴出した。

同テストで多数の不具合が発見された。その数は同年 10 月までに 600 件以上にのぼったという。両社の会議で日本 IBM 社の担当者は、受け入れテストの段階で不具合が多数見つかった理由として「要件定義フェーズ、設計フェーズの遅延に伴う開発フェーズの期間圧縮・テスト検証不足」を挙げた。加えて、受け入れテスト段階で要件の変更に当たる事項も顕在化し、その理由として「機能要件および外部設計に関するヒアリング・確認が不十分」などを挙げた。

日本 IBM 社は立て直しを図るため、10 月末にプロジェクトマネージャ (PM) を交代させるとともに、米セールスフォース・ドットコム社の技術担当者もプロジェクトに参画させたがその後、受け入れテスト工程で発生した追加工程の費用支払いをめぐる文化シャッタ社との間で意見が対立。日本 IBM 社は 2017 年 1 月から作業を中断した。この時点で見つかった不具合は 1000 件ほど。日本 IBM 社はこのうち約 800 件を「プログラムのバグでなく仕様の変更に当たる」とし、バグは 200 件弱と主張した。文化シャッタ社はこの分類に異論を唱えつつも、システムの早期稼働を優先。まず日本 IBM 社がバグと認めた 200 件弱の不具合のみを修正してシステムを稼働させるよう、日本 IBM 社に要請した。

だが日本 IBM 社は文化シャッタ社の案を受け入れず、2017 年 2 月に全く異なる提案を示した。それは Salesforce Platform を使ったカスタム開発から、Salesforce Platform の標準機能を活用した開発へと方針を転換する内容であった。開発を進めたシステムは、Salesforce Platform の標準機能で実装した部分が 5%、同基盤上でカスタム開発した部分が 95% だった。これを標準機能が 80% 以上、カスタム部分が 20% 以下になるよう開発し直す。標準機能を多用するため、画面のレイアウトやシステムの機能にも制約が加わることになる。

日本 IBM 社が 5 月に提示した具体案は、標準機能とシステム要件の適合性を見極めるため、要件定義からやり直す内容で、開発に 2 年 4 カ月と従来の 1.6 倍の期間を要するものであった。

文化シャッタ社は、この提案は実質的に従来のプロジェクトの成果を破棄するものであり、この段階でプロジェクトは頓挫したと判断。開発失敗の責任は

日本 IBM 社にあるとして、同社に支払った開発委託費約 22 億円を含む 27 億 4475 万円の損害賠償を求めて同社を訴えた。この原因はソフトウェア開発ベンダからすると、IT 発注企業へ合意形成をとっておらず前に進めたことが原因と考えられる。一方、IT 発注企業はシステムが完成後の受入テストで当初の仕様と違うことに気づき、要件の曖昧さが最終的にソフトウェア開発ベンダへ伝わらず、このような結果になったと考えられる。

上記事例はほんの 1 部の事例に過ぎず、表 2-1 に示す通り、多くの損害賠償請求が発生している。このような結果は、IT 発注企業とソフトウェア開発ベンダがお互いの合意形成が出来ずにもたらした結果と考えられる。つまり、ソフトウェア開発ベンダは、IT 発注企業の要件を明確に把握し、お互いの合意形成をとることでこのようなトラブルを回避する必要があると考えられる。

表 2-1 損害賠償請求一覧 [Keisan 10]

発生年	原告	被告	倍賞請求金額	原因
2018 年	文化シャッタ	IBM	27 億円	要件定義の不十分により当初の見積もり工数からの大幅な工数の超過
2008 年	放送会社	ソフトウェア開発ベンダ	19.3 億円	開発要素設計書において、全ての要件を決めきれなかったことによるスケジュールが遅延
2006 年	ソフトウェア開発ベンダ	IT 発注企業	25 億円	要件定義の不十分により当初の見積もり工数からの大幅な工数の超過
2002 年	ソフトウェア開発ベンダ	大手通信業者	1 億円	要件変更による追加請求
2002 年	システム開発ベンダ	書籍管理・配送	0.6 億円	要件定義の不十分により当初の見積もり工数からの大幅な工数の超過
2001 年	国民健康保険組合	ソフトウェア開発ベンダ	2.5 億円	プロジェクト管理の能力不足とユーザの不参加によるスケジュールの遅延
1998 年	ソフトウェア開発ベンダ	石材加工・販売業者	1.2 億円	納品後、信頼性や品質の低いシステムの為
1995 年	開発下請け業者	開発元請け業者	1.4 億円	初期見積もり工数と実績工数との乖離

2.5 ソフトウェア見積もり技法

ソフトウェア見積もりには大きく 2 種類の見積もり技法がある。1 つ目がソフトウェアの大きさ, つまり規模を算出するための「規模見積もり」と 2 つ目が, 作業にかかる工数を算出する「工数見積もり」である[Hatsuda 06]。規模見積もりでは LOC や FP 法や UML(Unified Modeling Language)のユースケース数で規模を測定する「ユースケース・ポイント」などがある。工数見積もりについては, プロジェクト体系の PMBOK(Project Management Body Of Knowledge)が出版している「PMBOK ガイド」によると, ソフトウェア見積もり工数見積もりの技法には 3 つあると紹介されている。それは「類推(トップダウン)見積もり」, 「係数モデル見積もり」, 「ボトムアップ見積もり」である。その構成が図 2-5 になる。

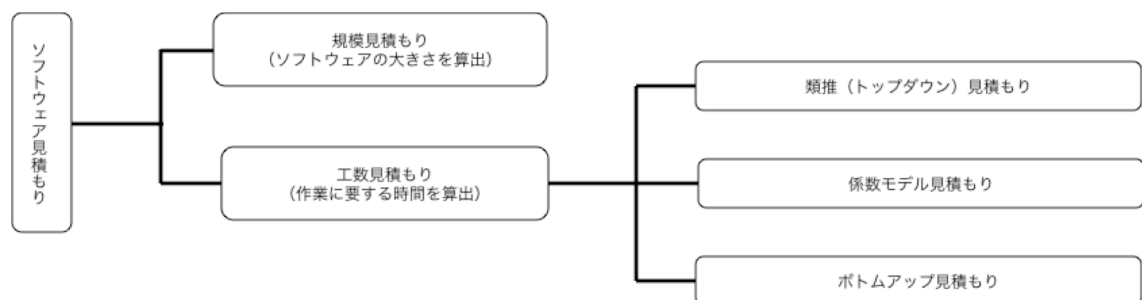


図 2-5 ソフトウェア見積もりの分類 [Hatsuda 06]

また, COCOMO を提唱した Dr. Boehm は著書「Software Engineering Economics」[BOEHM 81]の中でソフトウェアの工数を見積もる分類は以下の 7 分類あることを挙げている。その中の 1 から 4 については PMBOK に含まれている。

1. 類推法 (Estimation by Analogy)
2. トップダウン法 (Top-Down Estimation)
3. ボトムアップ法 (Bottom-Up Estimating)
4. 数式モデル (Algorithmic Model)
5. 専門家判断法 (Expert Judgement)

6. プライス・ツー・ウィン法 (Price to Win Estimating)

7. パーキンソン法 (Perkinsonian Estimation)

5. 専門家による判断法とはデルファイル法のことを示す。デルファイル法とは、コーディネータの数人の専門家に見積もりを行わせる。この時に専門家同士では話をせずに、コーディネータが試算した見積もりを統合させ、そしてサイドメンバへフィードバックし、再見積もりを行わせる。

6. プライス・ツー・ウィンでは顧客の予算に合わせて見積もる手法である。IT 発注企業の予算を意識することは重要であるが、システムがその予算に合わせることで予想以上にコストが膨らむ可能性があることを示唆している。

7. パーキンソン法とは、利用可能な人的リソースを元に見積もる手法である。つまりプロジェクトの期間や招集できるメンバの数に合わせて見積もりを行う技法である。プロジェクトメンバは作業が少ない場合、何かしら作業を行わせようとし、結果的にそれがルーチンワーク化して業務力が多くなった場合でもその仕事を減らすことが出来なくなってしまうことになる。Dr. Boehm は、上記 6, 7 の技法はやってはいけない技法であるとして言及している。

次節では PMBOK で提唱されている 1 から 4 の見積もり技法について説明する。

2.5.1 類推法

2.5.1.1 類推法の特徴

システム開発規模を非常に迅速に決定する方法として、類推法 [Capers 98] がある。これは、過去の類似の開発事例から開発規模を類推する実績ベースの手法である。

最も単純な類推法は、図 2-6 のように過去に開発されたシステムや経験から規模リストを眺め、1 つまたはそれ以上の類似のシステム開発プロジェクトを選択して概算規模を導き出す手法である。

開発規模リストの非常に単純な例を表 2-2 として示す。過去の事例とまったく同一パターンのシステム開発プロジェクトが見つかるのであれば、これは確実な見積もり方法だと言える。事例が 3 つ以上あれば理想的だが、2 つでも対象プロジェクトと比較することでどこに違いがあるのかをつかみやすくなる。この手法の重要なポイントとして、過去の事例を記憶に頼るのではなく、なるべくそれを詳細なドキュメントとして記録し、それを活用することである。

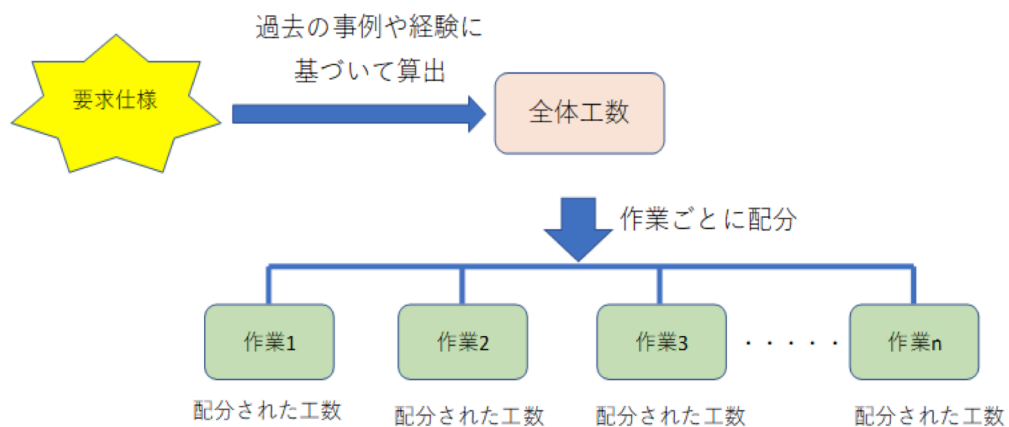


図 2-6 類推見積もりの手順

表 2-2 ソフトウェアアプリケーションの概略規模

ソフトウェアアプリケーションの概略規模

アプリケーション	タイプ	目的	開発言語	規模(K LOC)	規模(FP)	LOC/FP
Graphics Design	市販	CAD	Objective C	54	2,700	20.00
IEF	市販	CASE	C	2,500	20,000	125.00
Visual Basic	市販	コンパイラ	C	375	3,000	125.00
IMS	市販	データベース	アセンブラ	750	3,500	214.29
CICS	市販	データベース	アセンブラ	420	2,000	210.00
Lotus Notes	市販	グループウェア	混合言語	350	3,500	100.00
MS Office Professional	市販	オフィスツール	C	2,000	16,000	125.00
Smart Suite	市販	オフィスツール	混合言語	2,000	16,000	125.00
MS Office Standard	市販	オフィスツール	C	1,250	10,000	125.00
Word	市販	オフィスツール	C	315	2,500	126.00
Excel	市販	オフィスツール	C	375	2,500	150.00
MS Project	市販	プロジェクト管理	C	375	3,000	125.00
Knowledge Plan	市販	プロジェクト管理	C	134	2,500	56.67
CHECKPOINT	市販	プロジェクト管理	C++	225	2,100	107.14
FP管理	市販	プロジェクト管理	混合言語	56	450	125.00
SPQR/20	市販	プロジェクト管理	C	25	350	71.43
WMCCS	軍需	防衛	Quick Basic	18,000	175,000	102.86
航空機レーダ	軍需	防衛	Jovial	213	3,000	71.00
火砲コントロール	軍需	防衛	Ada83	250	2,336	107.00
航空会社予約	MIS	ビジネス	CMS2	2,750	25,000	110.00
保険	MIS	ビジネス	混合言語	1,605	15,000	107.00
電話	MIS	ビジネス	COBOL	1,375	11,000	125.00
税務	MIS	ビジネス	C	180	2,000	90.00
Windows 95	システム	オペレーティングシステム	混合言語	11,000	85,000	129.41
MVS	システム	オペレーティングシステム	C	12,000	55,000	218.18
UNIX V5	システム	オペレーティングシステム	アセンブラ	6,250	5,000	125.00
DOS 5	システム	オペレーティングシステム	C	1,000	4,000	250.00
SESS	システム	テレコミュニケーション	C	1,500	12,000	125.00
System/12	システム	テレコミュニケーション	CHILL	800	7,700	103.90

2.5.1.2 類推法の利点

本来，ソフトウェア見積もりは要件定義が完了していないと見積もりをとることはできない．ところがユーザ数やデータ量，業務などある程度の根拠のある工数やコストを算出することが出来る．また，同業種や同業務など一度開発経験をしたことがある場合，そのプロジェクトの差分から考慮して見積もることが出来る．図 2-7 はソフトウェア産業の実態把握に関する調査[IPA 12]の中から，過去のエンタープライズ開発に見積もり方法についてアンケートをとったグラフである．過去の実績に基づいた見積もりが 36.6%，個人の経験に基づいた見積もりが 12.0%となっており，合わせると 50%弱と半数近くをしめている．つまり類推法を利用している企業が多いことがわかる．

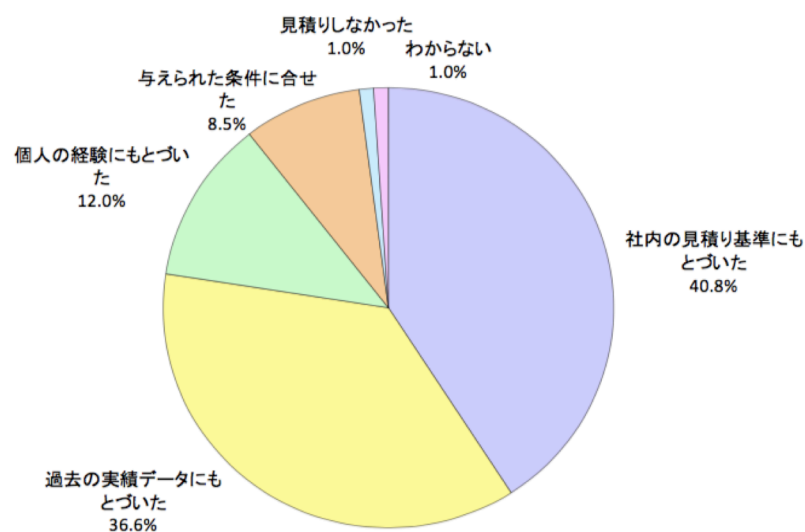


図 2-7 ソフトウェア産業の実態把握に関する調査 [IPA 12]

2.5.1.3 類推法の問題点

ところが、実際にはシステム開発プロジェクトのパターンは複雑であり、過去のシステム開発プロジェクトの中に多くの点で同一開発内容のものが存在するとは限らない。そのため、表 2-2 のような非常に単純な項目のみでのパターン比較では同一内容と判断できても、実際には過去の事例と異なる内容である場合も多く、非常に危険を伴う。従って類推法は、非常にリスクの高い曖昧な方法である。現実のシステム開発においては、経験と勘で見積もりが行われることが多い。経験と勘での見積もりとは、見積もり担当者が過去に経験してきたシステム開発プロジェクトの規模リストと、見積もりを行うシステム開発プロジェクトを頭の中で比較し、規模を見積もることに他ならない。このことから、類推法は、経験と勘による規模見積もりをまとめたものであるとも言える。

2.5.2 積算法（ボトムアップ法）

2.5.2.1 積算法の特徴

積算法[Okumura 09]は、システム開発において成果物や実施すべき作業など、図 2-8 のように、行うべき事柄を小さなタスクに分解し、それらのタスクを実行するために必要な工数や費用を積み上げることで、見積もりを行う方法である。開発するソフトウェアを構造化(サブシステム化)し、機能単位に見積もる

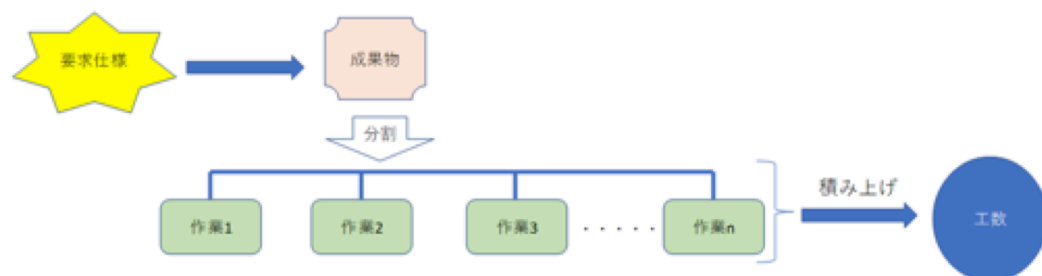


図 2-8 ボトムアップ見積りの手順

方法や、プロジェクトで実施する作業をタスク化に分解し、WBS 等を用いて行う。分解されたタスクは、必要な工数を誤差なく見積もれるほど小さな規模である必要がある。WBS の分解例 [m & t Consulting 02]については表 2-3 に示す。積算法は類推法や係数モデルの見積もりよりも精度が高いと言われており、システム開発プロジェクトを進めるための計画立案やスケジュール管理等においては非常に有効である。

表 2-3 WBS

ECサイト構築	A.マイルストーン	A.01 全体設計完了		
		...		
	B.プロジェクトマネジメント			
	C.システム	C.01 全体設計		
		C.02 システムインストール		
		C.02 システムテスト		
	D.Webプログラム	D.01 画面	D.01.01 設計	
			D.01.02 製作	
		D.02 CGI	D.02.01 設計	
		
	E.商品データ	E.01 自社商品	E.01.01 画像データ	E.01.01.01 撮影
				E.01.01.02 データ化
			E.01.02 商品紹介テキスト	E.01.02.01 作成
		
		E.02 他社商品
	
	F.ハードウェア	F.01 PC	F.01 調達	
		...		

2.5.2.2 積算法の問題点

ところが、積算法で見積もりを行うためにタスクへの分解を実施するには、当然、システム開発において行うべき事柄をすべて把握しなければならない。そのため、システム開発における早期の段階での概算見積もりには適用し難い。また、工数やコストが膨らむ傾向にある。そして、非常に手間と時間を要する欠点もある。つまり成果物や作業内容を詳細に洗い出さなければいけない。行うべきタスクへ分解する方法については多く議論されているが、分解された個々のタスクにかかる工数や費用を見積もること自体は経験と勘に過ぎない。

2.6 係数モデル見積もり

係数モデル見積もりは数式として提供される「見積もりモデル」に基づいて工数やコストを算出する方法である。図 2-9 のようにあらかじめ蓄積しておいた過去のプロダクトやプロセスのパラメータを使い、規模見積もりから工数を算出する。パラメータの適用方法には数式を使ったり、基準値を使ったりする見積もり技法がある。

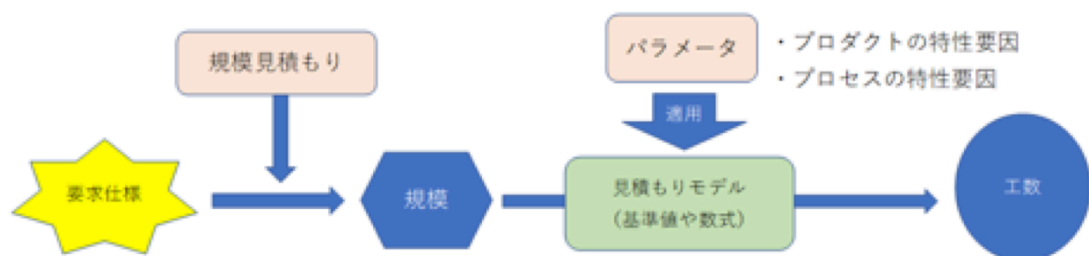


図 2-9 係数モデル見積もりの手順

2.6.1 LOC 法

2.6.1.1 LOC 法の特徴

古くから用いられている分析ベースの方法として、LOC (Line of Code) [Albrecht 83] がある。LOC は、システムで開発されるプログラムのステップ数を開発規模とする方法で、開発するソフトウェアの行数を把握し、その行数を開発工数に換算する。LOC は、汎用機における COBOL 言語を用いてのシステム開発にて考案されて以来、長年に渡り様々な開発環境にて用いられてきており、オブジェクト指向の概念が導入されてからは、機会は減っているものの、現在でも根強く使用されている。

LOC による 1 つの見積もり方法として、まず、システム開発プロジェクトのプログラミングフェーズにおける開発ステップ数を想定し、想定ステップ数から開発規模を見積もる。その上で、設計やテスト等の他のフェーズの開発規模を経験則からくるプログラミングフェーズとの開発規模比率により見積もり、フェーズ毎の合計を全体の開発規模とする方法がある。長く用いられてきたことから LOC では、アセンブラ、COBOL、FORTRAN、PASCAL、PL/I 等の手続き型言語を中心に経験則(表 2-4)が蓄積されてきている。

表 2-4 手続き型言語を用いるプロジェクトの LOC 尺度による概算ルール

手続き型言語を用いるプロジェクトのLOC尺度による概算ルール

プログラム規模	コーディング (LOC/人時)	コーディング工数 (人時)	テスト工数 (%)	設計工数 (%)	総工数 (人時)	ネット (LOC/時)
1	18.94	0.05	10.00	10.00	0.06	15.78
10	17.05	0.59	20.00	20.00	0.82	12.18
100	15.15	6.60	40.00	40.00	11.88	8.42
1,000	13.26	75.43	50.00	80.00	173.49	5.76
10,000	11.36	880.00	75.00	100.00	2,420.00	4.13
100,000	9.09	11,000.00	100.00	150.00	38,500.00	2.60
1,000,000	7.58	132,000.00	125.00	150.00	495,000.00	2.02

また、その中でも代表的なモデルが Doty 法と Putnam 法である。以下、Doty 法と putnam 法について説明する。

<Doty モデル>

Doty モデルは比較的小さいシステム向けのモデルである。初期段階における総開発工数は以下の式 2.1 で定義されている。尚、定数に関しては表 2-5 になる。

$$MM = aI^b \quad (\text{式 2.1})$$

※ MM：人月，I：プログラム規模(ソースコードまたはオブジェクトコード)，a, b：定数

表 2-5 Doty モデルの係数表

アプリケーション	オブジェクトコード		ソースコード	
	a	B	a	b
全体（総平均）	4.790	0.991	5.258	1.057
指揮管制	4.573	1.228	4.089	1.263
科学技術計算	4.495	1.068	7.054	1.019
事務処理	2.895	0.784	4.495	0.781
ユーティリティ	12.039	0.719	10.078	0.811

<Putnam モデル>

1978 年，Putnam により考案されたモデルであり，大規模/特大規模開発に適したモデルである．開発工数は以下の式 2.2 で定義されている．

$$K = \frac{L_3}{ck^3td^4} \quad (\text{式 2.2})$$

※K：ライフサイクル総工数(人年)～保守を含む．開発工数のみならこれを 0.4 倍する．L：出荷されるコード行数(LOC)，td：開発期間(年)，ck：補正係数

そして，LOC による別の見積もり方法として，大規模なシステム開発においては Brooks の法則を用いる方法がある．Brooks の法則[Kogure 00]とは，LOC において大規模なシステムと全体の開発工数の間には，工数がステップ数の 3/2 乗に比例(図 2-10)するという法則である．ステップ数から，直接にシステム全体の開発規模を見積もる．

このように，LOC は他の方法よりも比較的定量的な方法であると言える．

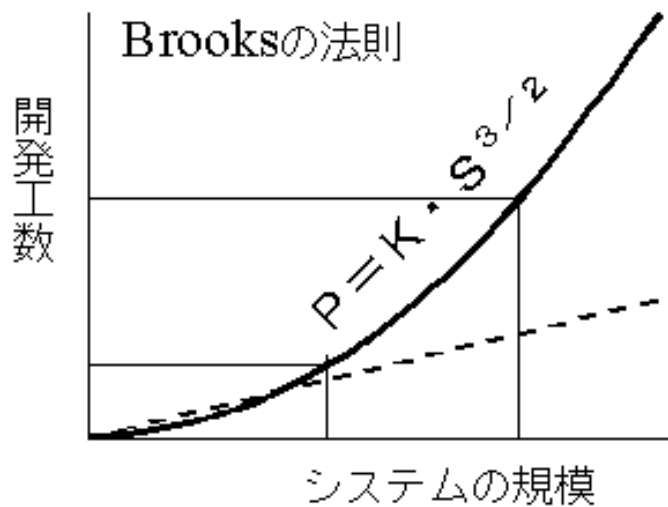


図 2-10 Brooks の法則 [Kogure 00]

2.6.1.2 LOC 法の問題点

明らかにある種の手続き型言語による経験をベースにしており，Visual Basic や Realizer 等のオブジェクト指向を取り入れたビジュアル開発環境が導入されたことによって，複雑なプログラミング部分がボタンコントロールやプルダウンメニュー等，再利用可能なコンポーネントによって実現されるようになった．その為，現在ではプログラムの内容がステップ数に反映され難くなっているという問題がある．また，プログラムの構成は様々であり，一般的には開発難易度が考慮されないため見積もり誤りを生じやすいという問題点も指摘されている．さらに，システム開発プロセスにおける早期の段階での概算見積もりという視点から最も重大な問題として，まだ開発していないプログラムの行数をどのように把握するのかという問題がある．つまり，プログラムのステップ数を想定するためにはシステムの詳細な設計が必要であり，要件定義の早期のような段階ではとうてい LOC を適用できないということである．

2.6.2 ファンクションポイント法

2.6.2.1 ファンクションポイント法の特徴

1979 年に Allan Albrecht が発表した方法に、ファンクションポイント [Garmus 12], [Kotani 98]がある。ファンクションポイント法は、システムの開発規模をユーザと開発者にとって意味のある外部的な特性によって決定しようとする概念に基づき、システムの機能量（ファンクションポイント）を測定して開発規模を見積もる方法である。システムが提供する機能を分割し、機能の種類ごとに開発規模の重み付けを行う。ファンクションポイント法が広く用いられるようになるにつれ、国際的非営利団体である IFPUG（International Function Point Users Group）が形成され、ファンクションポイントの計算ルールの近代化と更新を行っている。

ファンクションポイントは、システムの画面や帳票、データベースのレコードなどを洗い出し、それぞれの複雑さを点数化することで計算する。具体的には、まず洗い出しを図 2-11 で示す通り、5 つの分類で行う。

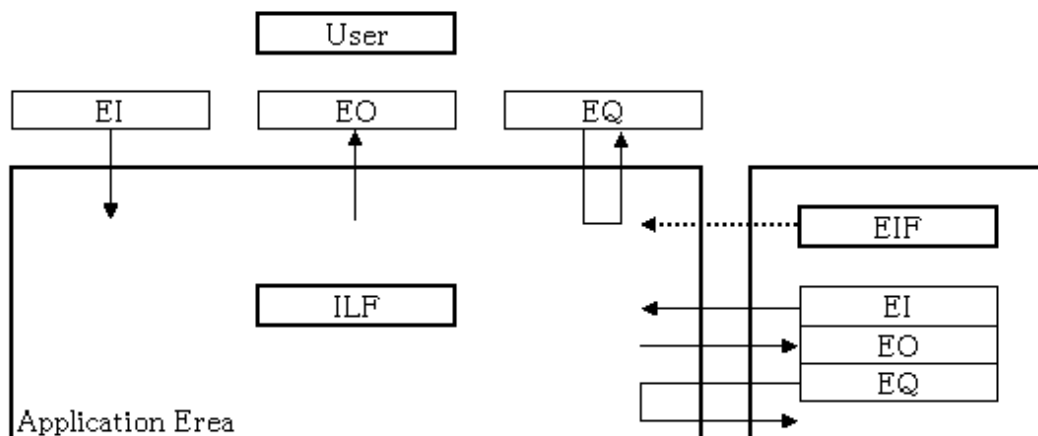


図 2-11 IFPUG の定義するファンクション

EI (External Input)

EO (External Output)

ILF (Internal Logical File)

EIF (External Interface File)

EQ (External Query)

洗い出した各要素のレコード数やファイル数により高，中，低の重み付けを行い，未調整ファンクションポイントを求める．この未調整ファンクションポイントにシステム特性係数を乗じて最終的なファンクションポイントを求める．システム特性係数は，通信の有無やトランザクションの多少などの 14 項目について 5 段階の評価を行い，これを点数計上し，これらの合計値に対して式 2.3 で算出する．そして，最終的なファンクションポイント値は，次式 2.4 で算出する．

$$\text{システム特性係数} = (\text{合計値} \times 0.01) + 0.65 \quad (\text{式 2.3})$$

$$\text{最終 FP} = \text{未調整 FP} \times \text{システム特性係数} \quad (\text{式 2.4})$$

このように計測されたファンクションポイントは機能量であるため，LOC の様に開発環境に依存せずにシステム開発規模を見積もることができる．また，機能というユーザからもわかりやすい視点で開発するシステムを捉えている．さらに，難易度（複雑さ）を評価するしくみの実装されている．

2.6.2.2 ファンクションポイント法の問題点

ファンクションポイント法は，事前に情報を得ることができれば比較的に高い精度での開発規模見積もりを行うことができる．ただし，それは画面等の 5 つの分類項目に該当する要素の洗い出しが漏れなく行えることが前提である．しかし，システム開発プロセスの早期の段階における見積もり作業は，5 つの分類の要素が確定する以前に行わなければならないという根本的な問題がある．つまり，詳細な機能分割を要件定義の段階で行うことは困難であるため，結局

は機能分割自体が開発担当者の経験と勘により実施されるという問題が発生し、精度の高い開発規模見積もりは困難となってしまう。

2.6.3 CoBRA 法

2.6.3.1 CoBRA 法の特徴

CoBRA 法[IShitani 06]は、経験豊富なプロジェクトマネージャ等の見積もり熟練者の経験・知識を抽出し、それを変動要因として定義・定量化することで、透明性と説明性が高い見積もり(コストマネジメント)を実現する方法である。開発規模がほぼ同じでもかかる工数に違いがある。つまり現実の工数をベースの生産性 α と、そこからのオーバーヘッドにより説明を行う技法である。

2.6.3.2 CoBRA 法モデル

開発工数は理想的な状態においては開発規模だけで決定される。これは理想の生産性があり、開発が理想の状態では式 2.5 で示すことができる。

$$effort = \frac{\text{規模}}{\text{理想の生産性}} = \frac{1}{\text{理想の生産性}} \times \text{規模} = \alpha \times \text{規模} \quad (\text{式 2.5})$$

ここで α は(1/理想の生産性)とする。ところが現実的には理想的な状態はほとんどなく、理想の生産性を下げる数多くの変動要因が存在し、変動要因が悪化させる影響の総和分、開発工数が増加する。これは現実のソフトウェア開発で必要な工数は理想の生産性から求められる開発工数に対して変動要因の総和分の工数が増加するという考え方である。そこで CoBRA 法モデルは次式 2.6 で定義されている[Briand 98] [IShitani 06]。

$$effort = \alpha \times size \times (1 + \sum CO_i) \quad (\text{式 2.6})$$

Size...LOC や FP などの工数規模

α …変動要因がない理想的状態での単位当たり工数

CO_i …関連する i 個の変動要因 ※ CO は cost overhead

i …変動要因の ID

理想状態の工数に対して，変動の総和が付け加えられたものになっている．

CoBRA 法の作成は図 2-12 の手順となる．また，上記式 2.6 の見積もり式を用いた手順は以下となる．

① 変動要因の選定

過去のプロジェクトメンバのプロジェクトマネージャ (PM) やプロジェクトリーダー (PL) 2, 3 名のブレインストーミング形式で要因を選定する．

② 過去の実績データの適用

過去のプロジェクトデータから，そのプロジェクトを担当した PM, PL, そしてメンバから要因のレベルを評価し増減割合を変動要因ごとに設定する．

③ シミュレーションの実行

それぞれのプロジェクトに対して，モンテカルロシミュレーションを実行して ΣCO_i の安定した三角分布を作成する．

③ 規模 $\times (1 + \Sigma CO_i)$ と工数に対して直線回帰分析から α 係数を算出する．

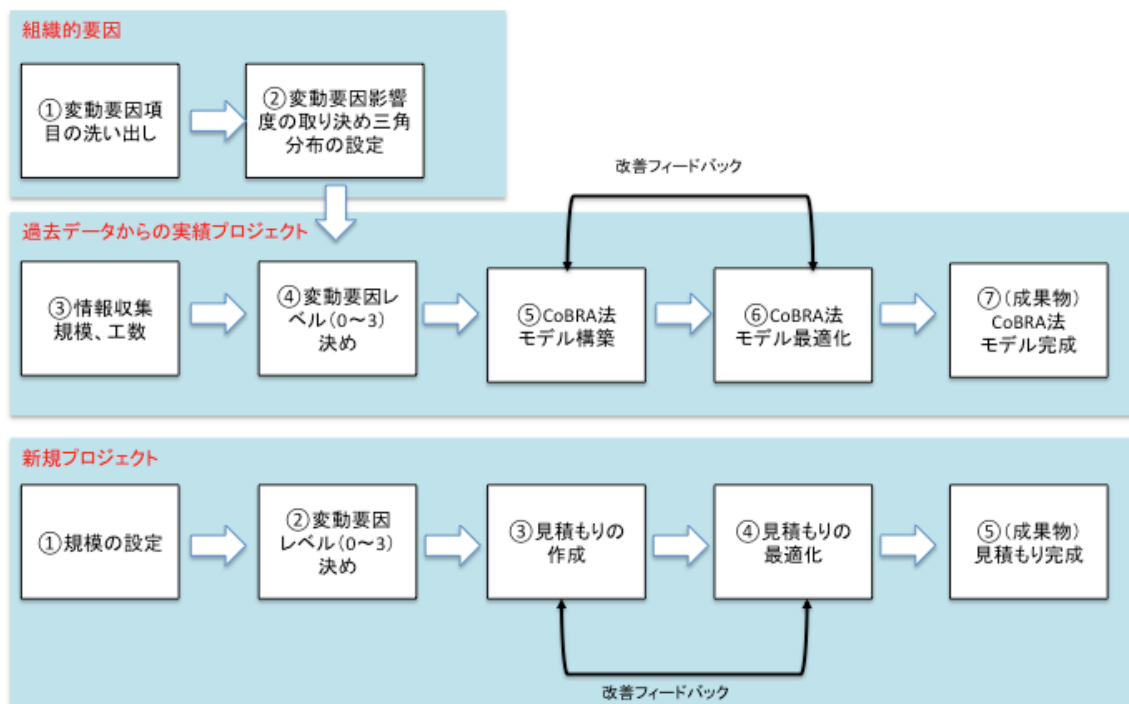


図 2-12 CoBRA 法の手順 [Ishitani 06]

2.6.4 COCOMO II

COCOMOIIモデルは、プロジェクトの持つ特徴や独自性に対応するためのモデルであり、プロジェクトの実績を蓄積しそれに合致するように微調整をしているモデルである[Kurihara 10]。このCOCOMOIIモデルには、3種類のモデルがある。プロジェクトの初期のソフトウェア開発見積もり時に行い、プロジェクトの初期段階で使用するモデルでソフトウェア工数見積もり精度が低い時点でもソフトウェア開発見積もりを可能とするアーリ・デザイン・モデル。また、新システムのアーキテクチャが決定した後のソフトウェア開発見積もりとして使用するポスト・アーキテクチャ・モデルがあり、この2つのモデルは、プログラムの新規開発やパッケージを含む既存システムの修正の開発、既存のシステムを別のプラットフォームに載せ替える再利用開発など考慮している。そして、スプレッドシートやGUI などを使用した簡単なシステムの開発見積もりを行うアプリケーション・コンポジション・モデルである。特にアーリ・デザイン・モデルとポスト・アーキテクチャ・モデルについては以下の式から工数を算出することができる[Center 00], [University 97]。

◆ アーリ・デザイン・モデル

$$PM = A \times Size^E \times \prod_{i=1}^7 EM_i \quad (\text{式 2.7})$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j \quad (\text{式 2.8})$$

※Size：調整開発規模，PM：人月，

A：2.94，EM：努力乗数，

SF：スケールファクタ，B=0.91，

Eは式2.8として置換

◆ ポスト・アーキテクチャ・モデル

$$PM = A \times Size^E \times \prod_{i=1}^{17} EM_i \quad (\text{式 2.9})$$

※PM，A，E，EMに関しては上記同様

努力乗数は, LOC や FP などの規模からは計算することができない要因である. COCOMO II Model Definition Manual ([Center 00] [Ellis 95])では努力乗数の関係をプラットフォームの難しさ (PDIF) や再利用要求 (RUSE), 開発スケジュール制約などを定義しており, 利用者のこの努力乗数に対して, 「きわめて高い」や「非常に高い」「高い」など評価をしていく. 式 2.7 から式 2.9 はいずれも努力乗数が重要なキーとなっていることが分かる.

2.6.5 ソフトウェア見積もり要因に関する研究

ソフトウェア見積もり要因に関する研究はいくつか存在する. 以下, ソフトウェア見積もり要因に関する先行研究について示す.

1. ソフトウェア開発プロジェクトの遂行に影響を与える要因モデル

プロジェクトの実績評価を目的変数とし, 規模, 工数, 工期などの量的データ及び, さまざまなプロジェクトの特性を表す質的データを説明変数とする研究がある [Furuyama 07]. この研究ではプロジェクトのQCDに影響を与える要因は以下の結論を導いている.

① コストの超過要因

コストの超過は大規模プロジェクト, 特に工期あたりの規模の大きいプロジェクトで起きやすい. また, ユーザとの役割分担の明確度が高いプロジェクト及び要因スキルが高いプロジェクトはコスト超過を起こしにくい傾向にあると言及している.

② 納期遅延及び, 品質低下の要因

要求があいまいなプロジェクトは納期遅延を起こす可能性が高く, 品質低下を行う傾向がある. つまりGlassの法則で知られている, 「要件の不備はプロジェクト失敗の根本な原因である」ことを立証している. また, PMスキルの高いプロジェクトは品質低下を起こしにくい傾向にあると示している. 但し, この筆者が傾向にあると示しているのは, 統計的有意水準を10%

としているため、その傾向が見られる程度と言及している。そのため、これら内容については信憑性が低いと考えられる。

③ テスト体制の効果

優れたテスト体制はコスト・納期・品質いずれの面からもプロジェクトの遂行に良い結果もたらしていることを示した。

2. 生産性要因に基づいて層別されたソフトウェア開発工数見積もりモデル

開発規模あたりの開発要員数と生産性との関係の研究がある[Tsunoda 09]。生産性と関連の強いプロジェクト特性(9つのカテゴリデータと10の数量データ)を特定するために、一元配置分散分析の有意差と寄与率から生産性と各プロジェクト特性との関係の有無の強さを分析している。つまり開発規模当たり要員数(大中小)に基づいてプロジェクトを層別にして構築したモデルは、平均要因数やFPなどを用いて層別にしたモデルや層別に行わず、構築したモデルよりも工数見積もりの精度が高いことを示している。

3. 開発要員数とその誤差を考慮した工数見積もりモデル

説明変数に誤りが含まれることに注目した研究がある[Tsunoda 13]。この研究では、説明変数を開発要員数として重回帰分析を実施した際に、開発要員数とその誤差から開発要員数を説明変数に含めるかどうか判断する。つまり、過去のプロジェクト平均と比較し、要員数が非常に小さい場合は要員数が小さいことを前提としたモデルを採用し、非常に大きい場合は要員数が大きいことを前提とするモデルを採用する。また、これらに当てはまらない場合は要因数を説明変数に含めないモデルを用いて工数を見積もる方法である。この先行研究では要因数の誤差が不明な場合や誤差が大きい場合、本手法が有効となることを示している。

4. 欠損データからの見積もり手法

見積もりデータは欠損しているデータが多く、一貫していないコスト要因データ、つまり、欠損値を含むデータから見積もりを行う手法に関する研究があ

る．回帰分析に対して欠損値処理法を用いるもの[Strike 01]，協調フィルタリングを用いるもの[Tsunoda 05]などである．これらの研究では，協調フィルタリングを用いた見積もりが最も高い精度となることが知られている[Kakimoto 06][Ohsugi 06]．

2.6.6 その他分析手法

これまで述べた方法の他にも，多くの手法が用いられている．Dr. Boehm が LOC にエンジニアの能力や要求の信頼性といった補正係数を加味するように発展させた COCOMO (Constructive Cost Model) 法[Center 00]．また，ファンクションポイント法とは異なる視点から開発するシステムを捉えたオブジェクトポイント法[Takahashi 01][Hashimoto05]やユースケース・ポイント法[Philippe 00]，少々変わった方法では複数の専門家に複数回の見積もりを依頼し，収束した見積もり値を採用するという Delphi 法[Boehm 81]等が存在する．
手法ごとに利点や欠点があるが，いずれの手法においても同様に，システム開発プロセスの早期の段階では情報が不足しているため適用できないという問題がある．

2.7 結言

本章では、多くのソフトウェア見積もり技法から、その技法の特徴と問題点について説明した。ソフトウェアは目に見えない成果物であり、それを計画の時点で見積もることは難しい。そのため、それら見積もり誤差を減らす為に古くから見積もり技法が研究されてきた。その代表する見積もり技法が PMBOK に示されている。PMBOK ではいくつかの見積もり技法を推奨しており、それが類推法、ボトムアップ法、数式モデルを使った見積もり技法である。

類推法については利用している企業が多いことを示した。ところがこの技法では機能が似ていても、実際には過去の事例と異なる内容も多いため非常にリスクの高い曖昧な技法であった。次に小さな作業タスクに分化して機能単位で見積もるボトムアップ法について説明した。この方法では細かいタスクに分けることで抜けや漏れがなくなるため、有効な技法であるが、早期段階でそれを見分けることは難しいことが分かった。また手間と時間がかかり、結局この手法も類推法と同様に経験と勘に過ぎないことを述べた。次に統計手法を使った技法について述べた。見積もりの時点で、RFP や要件定義より事前に機能に分けて、それをソースコードの量に換算する LOC 法。また機能の難易度やボリュームに分けてポイントをつけて見積もりを算出する FP 法など、これら技法を利用している企業も多い。ところが、ソースコードや機能以外に影響する部分を洗い出すことは難しい技法である。そこで、ソースコードや機能以外に見積もりに影響する変動要因を導入した技法である COCOMO II 法や CoBRA 法について述べた。この技法では LOC 法や FP 法以外に変動要因を係数に加えることで見積もりの精度を上げることになるため、非常に有効であることを述べた。しかしこの技法からでは、過去のデータから導出するため、どの変動要因に影響があり、事前に判断することは非常に難しかった。また、ソフトウェア開発ベンダが対象の業種の過去のデータを持っているとも限らない。そこで筆者はこの生産性変動要因に着目し、過去のソフトウェア見積もりのデータから、生産性変動要因を使い、見積もり工数や生産性変動要因の因子に関する研究についてサーベイした。その結果、それに関連する研究はいくつか存在していた。これら研究では

QCD (Quality, Cost, Delivery) に与える要因の研究や生産性要因を層別に分けた見積もり技法、また、データのコスト因子データの欠損データから見積もる技法など存在していた。ところが、ソフトウェアの質的要因に着目した研究はなかった。そこで本研究では質的要因となる生産性変動要因に着目し、生産性変動要因に影響を与える要因を用いてモデルを作成し、その因子を発見した。以下、現状解決しなければいけない以下問題に点についてまとめた。

- 1) 経験や勘での見積もりを行っている企業が多く占めているが曖昧な部分が多い。
- 2) LOC 法やFP 法などを使い定量的に見積もりをしている企業も多いが、定性的な部分が見えないため、誤差が大きい。
- 3) COCOMO II や CoBRA 法のように LOC や FP と変動要因を使い定量的かつ定性的に算出可能であるが、どの変動要因に見積もりに影響があるのか一般化できない。

上記解決法については次章からは統計手法を使い、見積もりモデルを定義し、そこから変動要因に影響を与えている要因を確認していく。具体的には、第3章では生産性変動要因と見積もり工数から見積もり工数に影響を与える要因は何かについて言及する。また、業種ごとに分けた場合に業種ごとに影響を与える要因について述べる。第4章では見積もり工数に影響を与える生産性変動要因から構成概念を見つけ出し、因子に影響を与える変動要因を見つけ出す。第5章ではその見つけ出した因子から業種間での相違性について述べる。

第3章

統計手法によるソフトウェア開発モデル

システム開発プロセスの早期の段階における開発規模見積もりにおいて、類似法やLOC法、FP法などの不確定要素の多くを取り入れられていない見積もりモデルから見積もり工数を算出することは難しい。したがってCoBRA法やCOCOMO IIなどのように不確定要素を取り入れた見積もりモデルは非常に有効であると言えることは前章より述べた。ところが、これら不確定要素を取り入れた技法でも、どのような不確定要素が原因なのかについて推定することは難しく、その工数のみをIT発注企業に提示し、理解を求めても納得することは難しい。それではどのようなモデルであればIT発注企業はソフトウェア開発ベンダから提示した見積もりに対して納得してもらえるのか。つまりプロジェクト規模を見積もる際、変動する要因の中のどの要因が一番ボトルネックになり、見えなくなっているか。その原因を追求する。

本章では、まずは本研究で使用するデータの内容や特徴について説明する。そしてCoBRA法やCOCOMO II法でも利用されている変動要因CO(Cost Overhead)について説明し、本実験で使用する生産性変動要因について定義する。そして見積もり工数は生産性変動要因からどのように影響を受けているのかを確認するため、見積もり工数と生産性変動要因からまずはフルモデルから重回帰分析を実施する。そして生産性変動要因の変数選択を行い、汎化能力の高いモデルを抽出する。その抽出したモデルから、見積もり工数に影響力のある生産性変動要因を抽出する。次に業種ごとではどの生産性変動要因が見積もり工数に影響を受けているのか深掘り分析する。そのためにまずは同データから業種ごと

に分類する．そしてその分類した業種ごとに重回帰分析を実施し，その結果となる重回帰式から，業種ごとにどのような生産性変動要因が見積もり工数に影響を受けているのかを明確にする．

3.1 データの収集

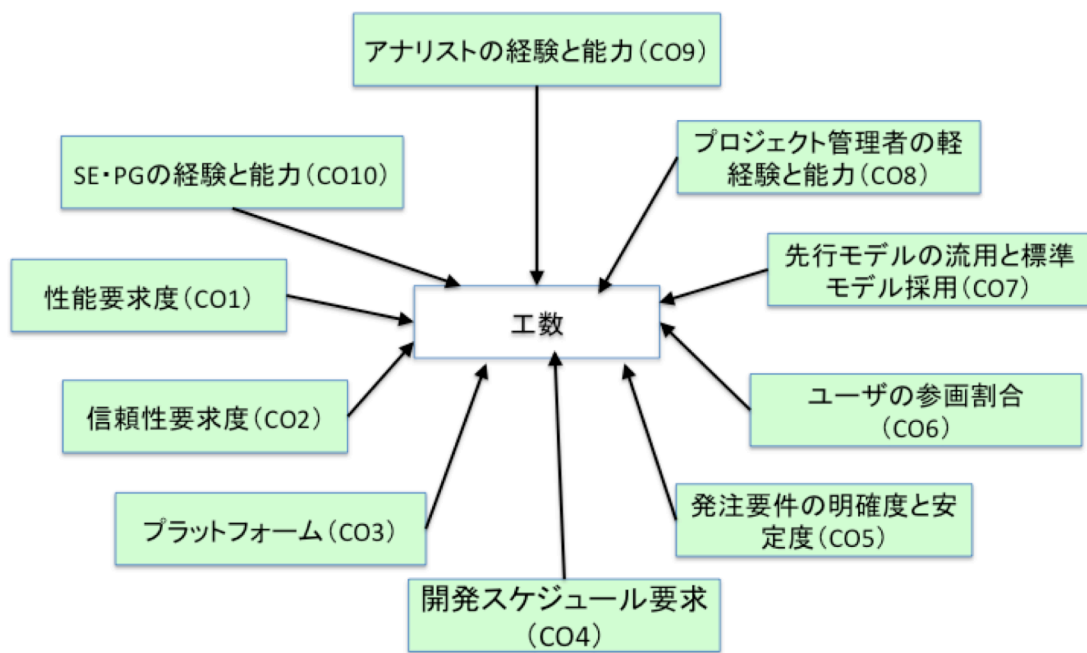
本研究のデータ提供元は、一般財団法人 経済調査会[Keizai 15]が 2001 年～2012 年度までに大企業から中小企業までの 344 社を対象にアンケートを実施し、収集した蓄積データである。データ項目は 375 あり、データ総件数は 2,009 レコードある。そのデータ項目の中から見積もり工数と生産性変動要因項目を抽出したが、欠損しているデータがいくつか存在していた。そこで、見積もり工数と生産性変動要因の欠損のないデータのみを抽出した結果、本章で利用した対象データ件数は 1,311 件となった。抽出したデータ項目は、プロジェクトごとに設定した見積もり工数（人月）と生産性変動要因である。本データの特徴としては、独立行政法人情報処理推進機構（以下、IPA）でも同様の調査を実施している。ところが IPA の場合、その対象となる企業数が 30 社[IPA 16]であるのに対し、本研究でのデータは 344 社と幅広い企業データが収集されている。

3.2 生産性変動要因の抽出

本節では見積もりに影響を与える生産性変動要因について説明する。生産性変動要因はプロジェクトごとに定義され、見積もり工数とどのように関係があるのかについて示す。

3.2.1 生産性変動要因とは

生産性変動要因とは、見積もり工数を決定するにあたり、直接変動する要因のことを指す。そして、チームの知識などの人的要因、開発経験などのプロセス要因、そしてシステムの複雑さなどの開発における工数の増加要因を示す。図 3-1 は見積もり工数に対して関連している 10 の変動要因のデータについて示している。



13

図 3-1 工数と変動要因の関係

3.2.2 生産性変動要因の設定

工数、工期、機能規模以外の開発生産性に影響を与えることが想定される生産性変動要因は、経済調査会が設置した委員会にて、JIS X0129-1 で定義されているソフトウェアの品質特性やその他の文献を参考に議論し、10 の生産性変動要因（CO:Cost Overhead）を設定した。経済調査会が設置した委員会は、国内の大手ソフトウェア開発ベンダなど実際の開発現場を知る 10 名強の委員で構成される。さらに、この委員会は、4 年におよぶ歳月をかけて各生産性変動要因の内容を定義し、表 3-1 に示す通り、1 から 5 までの 5 段階のレベル区分で設定した。レベルの尺度水準順序としては、レベル 1 の方がその変動する要因への影響は小さく、レベル 5 になるほど影響度は大きくなる。例えば、CO₁ の「機能性」の場合、レベル 1 では難易度の高い機能や他システムとの接続などを要求されなかった。レベル 2 では部分的に要求された。レベル 3 では適度に要求された、レベル 4 では部分的に過度に要求された。レベル 5 では全体的な過度に要求されたという評価設定を行っている。つまり変動要因のレベルが高いほど見積もり工数に深く影響することを意味する。表 3-1 は変動要因の説明とそのレベルに対する評価を説明している。

表 3-1 生産性変動要因一覧 [keizai 15]

変動要因	変動要因の説明	レベル	評価
機能性 (CO1)	機能性とは、下記の要件を指す ・ 難易度の高い機能（合目的性） ・ 精密性の高い計算（正確性） ・ 他システムとの接続（相互運用性） ・ 税法 OSI (Open System Interconnection) 企画等の公的標準、社内標準等（標準適合性） ・ 機密保護やアクセス管理（セキュリ	1	要求されなかった
		2	部分的に要求された
		3	適度に要求された
		4	部分的に過度に要求された

	ティ)	5	全体的に過度に要求された
信頼性 (C02)	信頼性とは以下の要件を指す ・故障発生率（成熟性） ・システムダウン（障害許容性） ・システムダウンからの再開・回復時間（回復性）	1	ソフトウェア障害は軽微な損失が想定された
		2	ソフトウェア障害は簡単に復旧可能な小規模な損失が想定された
		3	ソフトウェア障害は復旧可能な中規模の損失が想定された
		4	ソフトウェア障害は財政上の大規模な損失が想定された
		5	ソフトウェア障害は人命に関わる損失が想定された
プラットフォーム (C03)	プラットフォームとは、ハードウェアとソフトウェア(OS や DBMS など)の複合体でありその適合性（ニーズ、性能など）を指す.	1	非常に適合性があった
		2	かなり適合性があった
		3	適合性があった
		4	部分的に適合性に欠けた
		5	全体的に適合性に欠けた
開発スケジュール 要求 (C04)	開発スケジュール要求とは、開発スケジュールの制約度合を指す.	1	スケジュールは当初の想定より50%以上許容された
		2	スケジュールは当初の想定より10%～50%前後許容された
		3	スケジュールは当初の想定通りであった

		4	スケジュールは当初の想定より10%～25%前後短縮された
		5	スケジュールは当初の想定より25%以上短縮された
発注要件の明確度と安定度 (C05)	発注者の明確度とは、発注時点における発注仕様書の明確度合を指す。 発注要件の安定度とは、ソフトウェア開発期間中に発注者の要因により仕様変更が発生する度合を指す。	1	先行モデルがあり非常に明確で常に安定していた
		2	非常に明確で安定していた
		3	明確で安定していた
		4	部分的に不明確・不安定であった
		5	全体的に不明確・不安定であった
ユーザの参画割合 (C06)	ユーザの参画割合とは、発注者の関与度合を指す。	1	全行程について適切な担当者が適度に参画した
		2	全工程について適度に参画した
		3	主要な工程について適度に参画した
		4	一部参画した
		5	全く参加しなかった
先行モデルの流用と標準モデルの採用 (C07)	先行モデルの流用とは、対象業務のシステム化に開発済みの類似システムを流用できる度合を指す。	1	先行モデルは全面的に流用出来たまたは適合する標準モデルは全面的に採用出来た
		2	先行モデルはかなり流用出来たまた適合する標準モデルはかなり採用出来た

		3	先行モデルは部分的に流用出来たまたは適合する標準モデルは部分的に採用出来た
		4	先行モデルは存在したがほとんど流用出来なかったまたは適合する標準モデルは存在したが採用しなかった
		5	先行モデルは存在しなかったまたは適合する標準のモデルは存在しなかった
プロジェクト管理者の経験と能力 (C08)	プロジェクト管理の経験と能力とは、プロジェクト管理者に求められる経験と能力の度合を指す.	1	多数の中大規模のプロジェクトの管理を経験していた
		2	少数の中大規模のプロジェクトの管理を経験をしていた
		3	多数の小中規模プロジェクトの管理を経験していた
		4	少数の小中規模プロジェクトの管理を経験していた
		5	経験なかった
アナリストの経験と能力 (C09)	アナリストの経験と能力とは、アナリストに求められる経験と能力の度合（業務経験含む）を指す.	1	多数の中大規模のプロジェクトのアナリストを経験していた
		2	少数の中大規模のプロジェクトのアナリストを経験していた
		3	多数の小中規模プロジェクトのアナリストを経験していた
		4	少数の小中規模プロジェクトのアナリストを経験していた
		5	経験がなかった

SE・プログラマの経験と能力(C010)	SE・プログラマの経験と能力とは下記の要件を指す. ・業務の経験と能力の度合 ・開発技法の経験と能力の度合 ・開発方法論の経験と能力の度合 ・プラットフォームの経験と能力の度合 ・言語とツールの経験と能力の度合	1	過半数は十分な経験をしていた
		2	半数は十分な経験残り半数はある程度経験してしていた
		3	過半数はある程度経験していた
		4	半数はある程度経験していた
		5	要因の過半数は経験なかった

3.3 重回帰分析による実験

3.3.1 重回帰分析の概要

重回帰分析とは図 3-2 のように、2 つ以上の説明変数を設定し、目的変数との関係から関係式を作成し、予測・潜在能力などを評価する。さらには、説明変数が目的変数に及ぼす影響度や説明変数の重要性の格付けを行う。

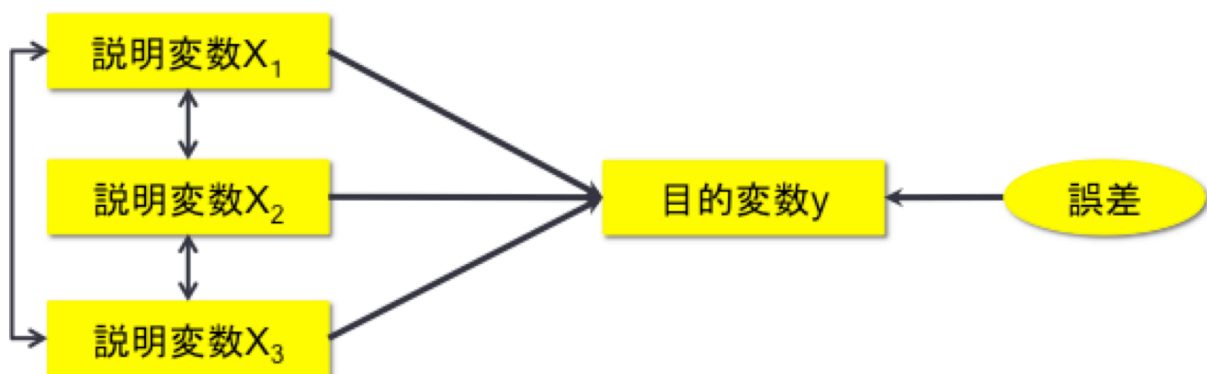


図 3-2 重回帰モデル

重回帰分析は、複数個の独立した説明変数 x_{ij} と目的変数 y_i の間に以下式 3.1 の線形の関係があることを仮定する。

$$y_i = \alpha + \sum_{j=0}^k \beta_j x_{ij} + \varepsilon_i \quad \varepsilon_i \sim N(0, \sigma^2), \quad i = 1, 2 \dots n \quad (\text{式 3.1})$$

α : 切片, β : 偏回帰係数, ε_i : 誤差 (残差)

これら α , β , k パラメータを推定することが重回帰分析の目的である。

【重回帰式 3.1 の導出】

重回帰式の例として、式 3.2 の目的変数に対して 2 つの説明変数がある場合について以下示す。 y の x_1, x_2 に対する重回帰式は以下のようになる。

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \varepsilon_i \quad \varepsilon_i \sim N(n, \sigma^2) \quad (\text{式 3.2})$$

β_0 は定数項, β_1 は y の x_{i1} に対する偏回帰係数, β_2 は y の x_{i2} に対する偏回帰係数である。この式 3.2 に実データの n 組の y, x_1, x_2 を代入し残差がもっとも小さくなる係数 α, β_1, β_2 の組み合わせを最小二乗法から重回帰式を求める。予測値と残差が式 3.3, 式 3.4 となる。

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} \quad (\text{式 3.3})$$

$$e_i = y_i - \hat{y}_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2}) \quad (\text{式 3.4})$$

このような残差平方和は以下となる。

$$\begin{aligned} S_e &= \sum_{i=1}^n (e_i)^2 \\ &= \sum_{i=1}^n \{y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2})\}^2 \end{aligned} \quad (\text{式 3.5})$$

つまり式 3.5 を最小にする $\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2$ の組み合わせが最小二乗法で求める解となる。そこで式 3.5 を $\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2$ で偏微分すると以下の通りとなる。

$$\frac{\partial S_e}{\partial \beta_0} = -2 \sum_{i=1}^n \{y_i - (\widehat{\beta}_0 + \widehat{\beta}_1 x_{i1} + \widehat{\beta}_2 x_{i2})\} = 0 \quad (\text{式 3.6})$$

$$\frac{\partial S_e}{\partial \beta_1} = -2 \sum_{i=1}^n x_{i1} \{y_i - (\widehat{\beta}_0 + \widehat{\beta}_1 x_{i1} + \widehat{\beta}_2 x_{i2})\} = 0 \quad (\text{式 3.7})$$

$$\frac{\partial S_e}{\partial \beta_2} = -2 \sum_{i=1}^n x_{i2} \{y_i - (\widehat{\beta}_0 + \widehat{\beta}_1 x_{i1} + \widehat{\beta}_2 x_{i2})\} = 0 \quad (\text{式 3.8})$$

したがって式 3.6 から式 3.8 より，以下の正規方程式が導出される．

$$\widehat{\beta}_0 n + \widehat{\beta}_1 \sum x_{i1} + \widehat{\beta}_2 \sum x_{i2} = \sum y_i \quad (\text{式 3.9})$$

$$\widehat{\beta}_0 \sum x_{i1} + \widehat{\beta}_1 \sum (x_{i1})^2 + \widehat{\beta}_2 \sum x_{i1} x_{i2} = \sum x_{i1} y_i \quad (\text{式 3.10})$$

$$\widehat{\beta}_0 \sum x_{i2} + \widehat{\beta}_1 \sum x_{i1} x_{i2} + \widehat{\beta}_2 \sum (x_{i2})^2 = \sum x_{i2} y_i \quad (\text{式 3.11})$$

ここで y_i , x_{i1} , x_{i2} それぞれの期待値を \bar{y} , \bar{x}_1 , \bar{x}_2 とすると式 3.9 は以下となる．

$$\widehat{\beta}_0 = \sum \frac{y_i}{n} - \widehat{\beta}_1 \frac{\sum x_{i1}}{n} - \widehat{\beta}_2 \frac{\sum x_{i2}}{n} = \bar{y} - \widehat{\beta}_1 \bar{x}_1 - \widehat{\beta}_2 \bar{x}_2 \quad (\text{式 3.12})$$

$$\therefore y_i = \widehat{\beta}_0 + \widehat{\beta}_1 \bar{x}_1 + \widehat{\beta}_2 \bar{x}_2 \quad (\text{式 3.13})$$

つまり式 3.13 より，推定された重回帰式は点 $[\bar{x}_1, \bar{x}_2, \bar{y}]$ を通ることになる．

式 3.12 を式 3.10, 式 3.11 に代入し， β_0 を消去すると

$$\begin{aligned} & \left(\sum_{i=1}^n \frac{y_i}{n} - \widehat{\beta}_1 \frac{\sum_{i=1}^n x_{i1}}{n} - \widehat{\beta}_2 \frac{\sum_{i=1}^n x_{i2}}{n} \right) \sum_{i=1}^n x_{i1} \\ & + \widehat{\beta}_1 \sum_{i=1}^n (x_{i1})^2 + \widehat{\beta}_2 \sum_{i=1}^n x_{i1} x_{i2} = \sum_{i=1}^n x_{i1} y_i \end{aligned} \quad (\text{式 3.14})$$

$$\begin{aligned} & \left(\sum_{i=1}^n \frac{y_i}{n} - \widehat{\beta}_1 \frac{\sum_{i=1}^n x_{i1}}{n} - \widehat{\beta}_2 \frac{\sum_{i=1}^n x_{i2}}{n} \right) \sum_{i=1}^n x_{i2} \\ & + \widehat{\beta}_1 \sum_{i=1}^n x_{i1} x_{i2} + \widehat{\beta}_2 \sum_{i=1}^n (x_{i2})^2 = \sum_{i=1}^n x_{i2} y_i \end{aligned} \quad (\text{式 3.15})$$

となり，式 3.14，式 3.15 を整理すると，以下となる．

$$\begin{aligned} & \widehat{\beta}_1 \left(\sum (x_{i1})^2 - \frac{(\sum x_{i1})^2}{n} \right) + \widehat{\beta}_2 \left(\sum x_{i1} x_{i2} - \frac{(\sum x_{i1})(\sum x_{i2})}{n} \right) \\ &= \sum_{i=1}^n x_{i1} y_i - \frac{(\sum x_{i1})(\sum y_i)}{n} \end{aligned} \quad (\text{式 3.16})$$

$$\begin{aligned} & \widehat{\beta}_1 \left(\sum x_{i1} x_{i2} - \frac{(\sum x_{i1})(\sum x_{i2})}{n} \right) + \widehat{\beta}_2 \left(\sum (x_{i1})^2 - \frac{(\sum x_{i1})^2}{n} \right) \\ &= \sum x_{i2} y_i - \frac{(\sum x_{i2})(\sum y_i)}{n} \end{aligned} \quad (\text{式 3.17})$$

ここで各変数の平方和と偏差積和を次のように定義する．

$$S_{11} = \sum_{i=1}^n (x_{i1} - \bar{x}_1)^2 = \sum x_{i1}^2 - \frac{(\sum x_{i1})^2}{n} \quad (\text{式 3.18})$$

$$S_{22} = \sum_{i=1}^n (x_{i2} - \bar{x}_2)^2 = \sum x_{i2}^2 - \frac{(\sum x_{i2})^2}{n} \quad (\text{式 3.19})$$

$$S_{12} = \sum_{i=1}^n (x_{i1} - \bar{x}_1)(x_{i2} - \bar{x}_2) = \sum x_{i1} x_{i2} - \frac{(\sum x_{i1})(\sum x_{i2})}{n} \quad (\text{式 3.20})$$

$$S_{yy} = \sum_{i=1}^n (y_i - \bar{y})^2 = \sum y_i^2 - \frac{(\sum y_i)^2}{n} \quad (\text{式 3.21})$$

$$S_{1y} = \sum_{i=1}^n (x_{i1} - \bar{x}_1)(y_i - \bar{y}) = \sum x_{i1} y_i - \frac{(\sum x_{i1})(\sum y_i)}{n} \quad (\text{式 3.22})$$

$$S_{2y} = \sum_{i=1}^n (x_{i2} - \bar{x}_2)(y_i - \bar{y}) = \sum x_{i2} y_i - \frac{(\sum x_{i2})(\sum y_i)}{n} \quad (\text{式 3.23})$$

したがって，式 3.18 から式 3.23 を式 3.16，式 3.17 に代入すると以下となる．

$$\widehat{\beta}_1 S_{11} + \widehat{\beta}_2 S_{12} = S_{1y} \quad (\text{式 3.24})$$

$$\widehat{\beta}_1 S_{12} + \widehat{\beta}_2 S_{22} = S_{2y} \quad (\text{式 3.25})$$

この連立方程式を行列にした場合、以下となる.

$$\begin{bmatrix} S_{11} & S_{12} \\ S_{12} & S_{22} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} S_{1y} \\ S_{2y} \end{bmatrix} \quad (\text{式 3.26})$$

式 3.26 の両辺に左から逆行列 $\begin{bmatrix} S_{11} & S_{12} \\ S_{12} & S_{22} \end{bmatrix}^{-1}$ をかけると以下となる.

$$\begin{aligned} \therefore \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} &= \begin{bmatrix} S_{11} & S_{12} \\ S_{12} & S_{22} \end{bmatrix}^{-1} \begin{bmatrix} S_{1y} \\ S_{2y} \end{bmatrix} \\ &= \frac{1}{S_{11}S_{22}-S_{12}^2} \begin{bmatrix} S_{22} & -S_{12} \\ -S_{12} & S_{11} \end{bmatrix} \begin{bmatrix} S_{1y} \\ S_{2y} \end{bmatrix} \\ &= \frac{1}{S_{11}S_{22}-S_{12}^2} \begin{bmatrix} S_{22}S_{1y} & -S_{12}S_{2y} \\ -S_{12}S_{1y} & S_{11}S_{2y} \end{bmatrix} \end{aligned} \quad (\text{式 3.27})$$

以上より、偏回帰係数 β_1, β_2 が求めるパラメータとなる.

また、式 3.12 を式 3.5 に代入すると、以下となる.

$$\begin{aligned} S_e &= \sum_{i=1}^n \{y_i - (\widehat{\beta}_0 + \widehat{\beta}_1 x_{i1} + \widehat{\beta}_2 x_{i2})\}^2 \\ &= \sum_{i=1}^n \{y_i - \bar{y} - \widehat{\beta}_1(x_{i1} - \bar{x}_1) - \widehat{\beta}_2(x_{i2} - \bar{x}_2)\}^2 \\ &= S_{yy} + \widehat{\beta}_1^2 S_{11} + \widehat{\beta}_2^2 S_{22} - 2\widehat{\beta}_1 S_{1y} - 2\widehat{\beta}_2 S_{2y} + 2\widehat{\beta}_1 \widehat{\beta}_2 S_{12} \\ &= S_{yy} + \widehat{\beta}_1(\widehat{\beta}_1 S_{11} + \widehat{\beta}_2 S_{12}) + \widehat{\beta}_2(\widehat{\beta}_1 S_{12} + \widehat{\beta}_2 S_{22}) - 2\widehat{\beta}_1 S_{1y} - 2\widehat{\beta}_2 S_{2y} \\ &= S_{yy} - (\widehat{\beta}_1 S_{1y} + \widehat{\beta}_2 S_{2y}) \end{aligned} \quad (\text{式 3.28})$$

したがって重回帰モデルの誤差 ε の母分散 σ^2 を次式 3.29 で推定できる.

$$\hat{\sigma}^2 = V_e = \frac{S_e}{\phi_e} = \frac{S_e}{n-3} \quad (\text{式 3.29})$$

ϕ_e は自由度を示す.

【寄与率(重相関係数(R2))の導出】

$$\begin{aligned}
S_{yy} &= \sum_{i=1}^n (y_i - \bar{y})^2 \\
&= \sum \{y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2}) + (\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2}) - \bar{y}\}^2 \\
&= \sum \{y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2})\}^2 + \sum \{(\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2}) - \bar{y}\}^2 \\
&\quad + 2 \sum \{y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2})\} \{y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2}) - \bar{y}\}
\end{aligned}$$

ここで、式 3.6 より、 $2 \sum_{i=1}^n \{y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2})\} = 0$ であるため、上式は以下となる。

$$\begin{aligned}
S_{yy} &= \sum \{y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2})\}^2 \\
&\quad + \sum \{(\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2}) - \bar{y}\}^2
\end{aligned} \tag{式 3.30}$$

一方、 $S_R = (\hat{\beta}_1 S_{1y} + \hat{\beta}_2 S_{2y})$ とおくと式 3.28 は以下となる。

$$S_{yy} = S_e + S_R \tag{式 3.31}$$

したがって、式 3.30 と式 3.31、 $S_e = \sum_{i=1}^n \{y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2})\}^2$ より式 3.32 となる。

$$\therefore S_R = \sum_{i=1}^n \{y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2})\}^2 \tag{式 3.32}$$

求めた回帰式が有効であるためには実測データ y_i と予測値 $\bar{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2}$ の相似基準として寄与率(決定係数)を利用する。寄与率 R^2 は以下の式で表すことが出来る。

$$R^2 = \frac{S_R}{S_{yy}} = \frac{S_{yy} - S_e}{S_{yy}} = 1 - \frac{S_e}{S_{yy}} \tag{式 3.33}$$

R^2 は説明変数が増えると自動的に大きくなる性質がある。そこで以下の式のように自由度を用いて調整する必要がある。これが自由度調整済み決定係数になる。

$$R^2 = 1 - \frac{S_e / \varphi_e}{S_{yy} / \varphi_T} \tag{式 3.34}$$

尚, S_{yy} , S_e の自由度は $\phi T=n-1$, $\phi R=n-3$
 これら指標を使い重回帰分析の評価を行う。

3.3.2 見積もり工数と生産性変動要因の重回帰分析

目的変数に見積もり工数, 説明変数に 10 の変動要因を設定し, 1,311 件のデータから, 見積もり工数と各要因とで単回帰分析を実施した。その結果から線形性があると判断し, より回帰分析の精度を上げるため, 各要因を説明変数とし, まずははじめにフルモデルでの重回帰分析を実施した。尚, 実験では統計解析ツール R を使用し, lm 関数のパラメータには式 3.35 の重回帰モデルを使用した。

$$\text{effort} \sim C01 + C02 + C03 + C04 + C05 + \\ C06 + C07 + C08 + C09 + C010 \quad (\text{式 3.35})$$

effort : 見積もり工数, **C0n**($n = 1, 2 \dots 10$)

次頁の実験結果から, 式3.36となり決定係数が7%, 自由度修正済み決定係数は6.3%となりこの重回帰モデルは6.3%を説明していることになる。また, p 値は5%を基準とし, 5%以下の生産性変動要因は「機能性」, 「信頼性」, 「発注要件の明確度と安定度」, 「プロジェクト管理の経験と能力」となり, これら4つの変数が統計的に有意となった。つまりこれら生産性変動要因はいずれも見積もり工数に大きく影響を与えていることになる。

$$\text{effort} = 3.553 + 16.358 * C01 + 27.528 * C02 - 2.275 * C03 + 3.391 \\ * C04 + 12.352 * C05 - 2.438 * C06 - 2.812 * C07 - 24.633 \\ * C08 - 4.371 * C09 + 1.214 * C010 \quad (\text{式 3.36})$$

Coefficients:				
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3.553	34.341	0.103	0.9176
機能性	16.358	6.001	2.726	0.0065 **
信頼性	27.528	5.261	5.233	1.94e-07 ***
プラットフォーム	-2.275	5.948	-0.382	0.7022
開発スケジュール要求	3.391	5.681	0.597	0.5507
発注要件の明確度と安定度	12.352	5.943	2.078	0.0379 *
参画割合	-2.438	4.854	-0.502	0.6156
先行モデルの流用と標準モデルの採用	-2.812	4.416	-0.637	0.5243
プロジェクト管理の経験と能力	-24.633	5.913	-4.166	3.30e-05 ***
アナリストの経験と能力	-4.371	5.051	-0.866	0.3869
SE.PG の経験と能力	1.214	4.346	0.279	0.7801

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1				
Residual standard error: 160.3 on 1300 degrees of freedom				
Multiple R-squared: 0.07007, Adjusted R-squared: 0.06291				
F-statistic: 9.795 on 10 and 1300 DF, p-value: 6.166e-16				

パラメータは数が多い重回帰モデルほど、データのあてはまりは良くなる。ところが、それは偶然得られたデータへの当てはめ向上を目的とする特殊化であり、式 3.36 の重回帰モデルの予測の良さを損なっている可能性がある。そこで、複数の重回帰モデルの中から、モデル選択するため、式 3.35 のフルモデルから、ステップワイズ法より変数を入れ替えながら重回帰モデルを構築した。そして、より汎化性の高い重回帰モデルを採択するため、統計解析ツール R の step 関数を使い、変数選択を実施した。実施した結果が以下となる。

Step: AIC=13320.34

見積もり工数 ~ 機能性 + 信頼性 + プラットフォーム + 開発スケジュール要求 + 発注要件の明確度と安定度 + 参画割合 + 先行モデルの流用と標準モデルの採用 + プロジェクト管理の経験と能力 + アナリストの経験と能力

	Df	Sum of Sq	RSS	AIC
- プラットフォーム	1	3257	33392051	13318
- 参画割合	1	5948	33394741	13319
- 開発スケジュール要求	1	8744	33397538	13319
- 先行モデルの流用と標準モデルの採用	1	9727	33398521	13319
- アナリストの経験と能力	1	18908	33407702	13319
<none>			33388794	13320
- 発注要件の明確度と安定度	1	115424	33504217	13323
- 機能性	1	192901	33581695	13326
- プロジェクト管理の経験と能力	1	446778	33835572	13336
- 信頼性	1	711131	34099924	13346

Step: AIC=13318.47

見積もり工数 ~ 機能性 + 信頼性 + 開発スケジュール要求 + 発注要件の明確度と安定度 + 参画割合 + 先行モデルの流用と標準モデルの採用 + プロジェクト管理の経験と能力 + アナリストの経験と能力

	Df	Sum of Sq	RSS	AIC
- 参画割合	1	6196	33398247	13317
- 開発スケジュール要求	1	8958	33401009	13317
- 先行モデルの流用と標準モデルの採用	1	11614	33403665	13317
- アナリストの経験と能力	1	19468	33411519	13317
<none>			33392051	13318
- 発注要件の明確度と安定度	1	112237	33504288	13321
- 機能性	1	189650	33581701	13324
- プロジェクト管理の経験と能力	1	451336	33843387	13334
- 信頼性	1	708512	34100563	13344

Step: AIC=13316.71

見積もり工数 ~ 機能性 + 信頼性 + 開発スケジュール要求 +

発注要件の明確度と安定度 + 先行モデルの流用と標準モデルの採用 +

プロジェクト管理の経験と能力 + アナリストの経験と能力

	Df	Sum of Sq	RSS	AIC
- 開発スケジュール要求	1	8080	33406327	13315
- 先行モデルの流用と標準モデルの採用	1	11867	33410114	13315
- アナリストの経験と能力	1	19037	33417283	13316
<none>		33398247	13317	
- 発注要件の明確度と安定度	1	106371	33504618	13319
- 機能性	1	192483	33590730	13322
- プロジェクト管理の経験と能力	1	465550	33863797	13333
- 信頼性	1	706343	34104589	13342

Step: AIC=13315.03

見積もり工数 ~ 機能性 + 信頼性 + 発注要件の明確度と安定度 +

先行モデルの流用と標準モデルの採用 + プロジェクト管理の経験と能力 +

アナリストの経験と能力

	Df	Sum of Sq	RSS	AIC
- 先行モデルの流用と標準モデルの採用	1	12469	33418796	13314
- アナリストの経験と能力	1	18965	33425292	13314
<none>		33406327	13315	
- 発注要件の明確度と安定度	1	107407	33513733	13317
- 機能性	1	196018	33602345	13321
- プロジェクト管理の経験と能力	1	465056	33871382	13331
- 信頼性	1	721440	34127766	13341

Step: AIC=13313.52

見積もり工数 ~ 機能性 + 信頼性 + 発注要件の明確度と安定度 +

プロジェクト管理の経験と能力 + アナリストの経験と能力

	Df	Sum of Sq	RSS	AIC
- アナリストの経験と能力	1	18982	33437778	13312
<none>		33418796	13314	
- 発注要件の明確度と安定度	1	96547	33515343	13315
- 機能性	1	186112	33604908	13319
- プロジェクト管理の経験と能力	1	490991	33909787	13331
- 信頼性	1	726759	34145555	13340

Step: AIC=13312.26

見積もり工数 ~ 機能性 + 信頼性 + 発注要件の明確度と安定度 +

プロジェクト管理の経験と能力

	Df	Sum of Sq	RSS	AIC
<none>		33437778	13312	
- 発注要件の明確度と安定度	1	87553	33525331	13314
- 機能性	1	182034	33619812	13317
- 信頼性	1	745855	34183634	13339
- プロジェクト管理の経験と能力	1	844863	34282641	13343

Call:

lm(formula = 見積もり工数 ~ 機能性 + 信頼性 + 発注要件の明確度と安定度 + プロジェクト管理の経験と能力,
data = dat5)

Coefficients:

(Intercept)	機能性	信頼性
-1.792	15.650	28.002
発注要件の明確度と安定度	プロジェクト管理の経験と能力	
10.274	-27.904	

AIC (Akaike's Information Criterion) とは、良い予測をする統計モデルが、良い統計モデルという考えに基づいて設計された選択基準である。パラメータの数が増えた場合、あてはめ誤差が小さくなり、対象となるデータの適合度は良くなる。しかしその反面、パラメータに全く関係のない項目が入ってきた場合、測定構造と無関係な変動にも合わせてしまうことになるため、同種のデータでは適合しなくなってしまうことになる。その問題を解決するために、モデルのパラメータ数をどれだけ抑える必要があるかを AIC の最小モデルを選択することで良いモデルを選択することが出来ることになる。AIC は以下の式 3.37 で表現することができ、AIC は尤度で定義された統計モデルの良さを測るために使用され、値が小さいほど良い統計モデルと判定される。

$$\begin{aligned} AIC &= -2\{(\text{最大対数尤度}) - (\text{最尤推定したパラメータ数})\} \\ &= \chi^2 - 2df \end{aligned} \quad (\text{式 3.37})$$

式 3.37 では最尤推定したパラメータは自由度の 2 倍に等しいことを示している。最尤法では手元のデータが得られる確率（密度）が最大になるようにパラメータを推定する。推定されたパラメータによって観測変数の分布が決まることになる。この時、統計モデルによって明示された観測変数の分布と本来の分布が似ていればいるほど期待平均対数尤度が高くなることが知られている。そして期待平均対数尤度が高い統計モデルほど、AIC が低く推定されるという関係がある。この性質を利用し、複数の統計モデルが候補に挙がっている場合、AIC が最小の統計モデルを採用することで、将来得られるであろうデータの対数尤度とその統計モデルの下で最も高くなることが期待できることになる。

上記の実験結果からフルモデルで実施した場合、AIC=13322.26 となったが、変数を増減させた結果、AIC が一番小さい AIC=13312.26 が一番あてはまりのよい統計モデルということになる。したがって生産性変動要因は「機能性」、「信頼性」、「発注要件の明確度と安定度」、「プロジェクト管理の経験と能力」の 4 つの生産性変動要因が選択された。つまり切片と 4 つの偏回帰係数から構成され、その重回帰式は式 3.38 となった。

$$effort = -1.792 + 15.7C01 + 28.0C02 + 10.3C05 - 27.9C08 \quad (\text{式 3.38})$$

特に式3.38の重回帰式の偏回帰係数を確認すると、選択された変動要因の中でも CO_2 の「信頼性」と CO_8 の「プロジェクト管理の経験と能力」の絶対値が高い結果となった。また、 CO_8 の偏回帰係数については、-27.90となり「プロジェクト管理の経験と能力」が負となった。これは間接効果が考えられる。つまり、プロジェクト管理の経験と能力が高いメンバほど、IT発注企業側の予算や納期などが厳しい状況のプロジェクトに配属されていることが考えられる。

次に、10の変動要因の変数間での相関の強さを確認するため、各変数間の相関について調査した10の変動要因同士の相関について確認した。相関係数にはピアソンの積率相関係数とスピアマンの順位相関係数、ケンドールの順位相関係数などの係数がある。今回の相関係数は10の生産性変動要因から、2変数間同士の相関性を考慮することが目的のため、ピアソンの積率相関係数を使用した。尚、ピアソンの積率相関係数の式は、 r を相関係数とし、 $x = \{x_i\}$ 、 $y = \{y_i\}$ それぞれの相加平均を \bar{x} 、 \bar{y} とした場合以下の数式で示すことができる。

$$r = \frac{\sum_{i=1}^n (x - \bar{x})(y - \bar{y})}{\sqrt{\sum_{i=1}^n (x - \bar{x})^2} \sqrt{\sum_{i=1}^n (y - \bar{y})^2}} \quad (\text{式 3.39})$$

式3-39を使い、各生産性変動要因で実施した相関係数は表3-2に示す。

表 3-2 生産性変動要因の相関係数一覧

	CO1	CO3	CO3	CO4	CO5	CO6	CO7	CO8	CO9	CO10
CO1	1.00	0.30	0.24	0.07	0.33	0.04	0.21	0.00	0.05	0.15
CO2	0.30	1.00	0.13	0.09	0.00	0.01	0.01	-0.09	-0.10	0.07
CO3	0.24	0.13	1.00	0.00	0.25	0.08	0.23	0.11	0.11	0.20
CO4	0.07	0.09	0.00	1.00	0.03	0.07	-0.02	0.00	0.00	-0.03
CO5	0.33	0.00	0.25	0.03	1.00	0.20	0.29	0.10	0.17	0.22
CO6	0.04	0.01	0.08	0.07	0.20	1.00	0.08	0.12	0.07	0.14
CO7	0.21	0.01	0.23	-0.02	0.29	0.08	1.00	0.16	0.13	0.19
CO8	0.00	-0.09	0.11	0.00	0.10	0.12	0.16	1.00	0.55	0.23
CO9	0.05	-0.10	0.11	0.00	0.17	0.07	0.13	0.55	1.00	0.17
CO10	0.15	0.07	0.20	-0.03	0.22	0.14	0.19	0.23	0.17	1.00

表3-2からは、相関係数は全体的には低いことが分かった。その中でも相関係数が高い生産性変動要因は、CO₈（プロジェクト管理の経験と能力）とCO₉（アナリストの経験と能力）の2変数間の相関係数が $r=0.55$ となり、表3-3から、中程度の正の相関が計測された。この2変数間の関係は、経験や管理能力が共通しているが、業務経験豊富なシステムアナリストはプロジェクトが発足される前に業務分析を行い、そのままプロジェクトマネージャに就いていることが考えられる。また、CO₁₀（SE・プログラマの経験と能力）についても同じ経験能力となるが、CO₈とCO₁₀の相関係数は0.23、CO₉とCO₁₀の相関係数は0.17となりアナリストやプロジェクトマネージャがそのままSEやプログラマにつかず、別のメンバが割り当てられていると考えられる。

表 3-3 相関係数の値と相関(参考値)

相関係数	相関
$-1 \leq r \leq -0.7$	強い負の相関
$-0.7 \leq r \leq -0.4$	中程度の負の相関
$-0.4 \leq r \leq -0.2$	弱い負の相関
$-0.2 \leq r \leq 0.2$	ほとんど相関なし
$0.2 \leq r \leq 0.4$	弱い正の相関
$0.4 \leq r \leq 0.7$	中程度の正の相関
$0.7 \leq r \leq 1$	強い正の相関

次に説明変数間で同じような内容が含まれていないか確認するため、偏回帰係数間の多重共線性について確認した。

多重共線性とは、重回帰モデル内の一部の予測変数が他の予測変数と相関しているときに起こる状態である。重度の多重共線性は、回帰係数の分散を増加させて不安定にするため問題となる。係数が不安定になると以下の影響が生じる。

1. 係数は、予測変数と目的変数の間に有意な関係が存在する場合でも、有意でなく見える場合がある。
2. 相関の高い予測変数の係数はサンプルによって大きく異なる。
3. 重回帰モデルから相関の高い項のいずれかを削除すると、他の相関の高い項の係数の推定値に大きな影響を及ぼすことになる。また、相関の高い項の係数の符号が誤っている場合もある。

したがって、多重共線性は重回帰式を導出した式 3.27 の逆行列が存在しない場合、つまり $\det=0$ の場合、多重共線性が発生していると考えられる。多重共線性は以下の式 3.40 から導出することができる。

$$\begin{aligned}
& S_{11}S_{22} - S_{12}^2 = 0 \\
\Leftrightarrow & \frac{S_{12}^2}{S_{11}S_{22}} = 1 \\
\Leftrightarrow & r_{x_1x_2}^2 = \left\{ \frac{S_{12}}{\sqrt{S_{11}S_{22}}} \right\}^2 = 1 \\
\Leftrightarrow & r_{x_1x_2} = \pm 1 \qquad \qquad \qquad (\text{式 3.40})
\end{aligned}$$

すなわちの x_1 と x_2 の相関係数が 1 または -1 のとき多重共線性が存在することになる。 x_1, x_2 の相関が ± 1 となることは点 (x_{i1}, x_{i2}) ($i=1, 2, 3, \dots, n$) の全てが一直線上に並んでいる場合となるからである。また, 多重共線性を測定するには, 予測変数の相関の構造を調べるか, 分散拡大係数となる VIF (Variance Inflation Factor) を調べることも可能である。分散拡大係数は, 予測変数間に相関がある場合に推定された回帰係数の分散がどれだけ増加するかを測定するものである。すべての VIF が 2 より小さい場合, 多重共線性は発生していないことになる。一方, VIF が 10 より大きい場合, その項の回帰係数は適切に推定されないことになる。表 3-4 は機能性, 信頼性, 発注要件の明確度と安定度, プロジェクト管理の経験能力全てにおいて分散拡大要因 (VIF) について調査した結果である。その結果, $VIF < 2$ となり多重共線性に関しては問題なしと考えられる。つまり, 今回の重回帰モデル式からは特に多重共線性は発生していないことが分かった。

表 3-4 変動要因による VIF 結果一覧

機能性	信頼性	発注要件の明確度と安定度	プロジェクト管理の経験と能力
1. 24	1. 12	1. 14	1. 02

3.4 業種ごとによる分析

3.3 節では見積もり工数と生産性変動要因から重回帰分析を実施し重回帰モデルを導出した。全体として4種類の生産性変動要因に関係性があることが分かった。ところがIT業界全体を俯瞰した場合はこの重回帰モデルで良いが、実際のエンタープライズでの業務の場合、ソフトウェア開発ベンダの担当者は業種ごとにプロジェクトを進めていくことがほとんどである。そのため、3.3 節から抽出した重回帰モデルでは業種ごとでの重回帰モデルとして考慮している訳ではなく、各業種全てに当てはまることは考えにくいと考えられる。そこで業種ごとでは、どのような生産性変動要因が見積もり工数に影響を与えているのかを明確にする必要があると考えた。

本節ではまず業種について定義し、業種ごとで重回帰分析を実施した場合、業種ごとでどの生産性変動要因に影響があるのかについて検証した。

3.4.1 業種の分類

本研究で適用業種は、全9種類の業種を対象とした。業種別の分類は表3.5の通りである。サービス業には飲食店、宿泊業、医療、福祉、教育、学習支援事業が含まれており、運輸業には郵便業、卸売業、小売業、また、その他に関しては、不動産業、物品賃貸業、学術研究、専門・技術サービス業が含まれる。

表 3-5 業種別分類

業種	分類
建設業	建設業
製造業	製造業
電気・ガス・熱供給・水道	電気・ガス・熱供給・水道
情報通信業	情報通信業
流通業	運輸業，郵便業 卸売業，小売業
金融業・保険業	金融業，保険業
サービス業	宿泊業，飲食サービス業 医療，福祉 教育，学習支援業 生活関連サービス業 サービス業（他に分類されないもの）
公務	公務（他の分類除く）
その他	不動産業，物品賃貸業 学術研究，専門，技術サービス業，その他

3.4.1 業種別による分析

収集したデータから、見積もり工数と 10 の生産性変動要因を業種ごとに抽出し、表 3-5 で示す通り 9 種類の業種で分類を行った。そして全業種で実施した重回帰分析と同様に、見積もり工数を目的変数、10 の生産性変動要因を説明変数としてそれぞれで重回帰分析を行った。そして汎化性の高い重回帰モデルを導出し予測精度の高い重回帰モデルを導出するため、3.3.2 節と同様に、ステップワイズ法より変数選択を実施し、変数選択の評価は AIC を適用した。その結果が式 3.41 から式 3.49 となった。尚、統計ソフトは R を使い、使用した関数は step 関数を利用し分析した。

[建設業]

$$\text{effort}_1 = 21.0 + 14.0 * \text{CO}_2 + 14.2 * \text{CO}_3 - 24.0 * \text{CO}_5 + 6.8 * \text{CO}_9 \quad (\text{式 3.41})$$

[製造業]

$$\text{effort}_2 = -47.8 + 13.9 * \text{CO}_2 + 16.6 * \text{CO}_5 \quad (\text{式 3.42})$$

[電気・ガス・熱供給・水道業]

$$\text{effort}_3 = -124.0 + 68.5 * \text{CO}_4 + 116.2 * \text{CO}_5 - 110.1 * \text{CO}_9 \quad (\text{式 3.43})$$

[情報通信業]

$$\begin{aligned} \text{effort}_4 = & -55.6 + 29.2 * \text{CO}_2 + 28.3 * \text{CO}_3 \\ & + 29.3 * \text{CO}_4 - 29.4 * \text{CO}_9 \end{aligned} \quad (\text{式 3.44})$$

[流通業]

$$\text{effort}_5 = -14.4 + 32.7 * \text{CO}_2 + 19.7 * \text{CO}_5 - 27.7 * \text{CO}_8 \quad (\text{式 3.45})$$

[金融業，保険業]

$$\text{effort}_6 = 333.5 + 55.1 * \text{CO}_2 - 45.8 * \text{CO}_4 - 72.0 * \text{CO}_8 \quad (\text{式 3.46})$$

[サービス業]

$$\text{effort}_7 = -35.3 + 34.2 * \text{CO}_1 + 21.9 * \text{CO}_2 - 22.2 * \text{CO}_9 \quad (\text{式 3.47})$$

[公務]

$$\text{effort}_8 = 57.2 + 30.3 * \text{CO}_1 - 23.1 * \text{CO}_9 \quad (\text{式 3.48})$$

[その他]

$$\begin{aligned} \text{effort}_9 = 5.4 + 17.0 * \text{CO}_2 + 18.6 * \text{CO}_3 + 19.1 * \text{CO}_5 \\ - 21.9 * \text{CO}_6 - 16.4 * \text{CO}_7 \quad (\text{式 3.49}) \end{aligned}$$

3.4.2 実行結果の考察

式 3.42 から式 3.49 より、各業種ではどのように生産性変動要因が影響しているのかについて以下考察を述べる。

機能性(CO₁)については「サービス業」と「公務」の2業種が抽出された。偏回帰係数の絶対値はそれぞれ34.2, 30.3となりこの2変数の生産性変動要因の影響力は大きい。これら業種では主に他システムとの連携が多く発生していることが考えられ、それらが見積りに大きく影響していると考えられる。また、サービスを提供するにあたり機密性保護やアクセス管理などセキュリティ面を重視しており、より強固な対策をすることで見積もり工数への影響が大きくなっていることが考えられる。

信頼性(CO₂)については、9業種中7業種で信頼性の変数が選択された。これはどの業種もシステムがダウンすることで、サービスや業務に支障をきたすことになるため、故障率や障害許容度、システムの回復性については重要視してと考えられる。特に金融機関は他の業種と偏回帰係数を比べても55.1と非常に高い結果となった。金融業の場合、仮にシステムダウンが発生した場合、社会的に影響を与えることになる。そのため、止まらないシステムの安定稼働を常に考慮しているため、見積もり工数に大きく影響を与えられる。一方、信頼性に影響しないと示されたのが、「電気・ガス・熱供給・水道業」と「公務」

の 2 業種である。これら業種では故障率や障害許容度、システムダウンからの回復性などはあまり見積もりに影響を与えない業種であることが分かった。理由として、これら業種が利用しているシステムはシステムが停止しても業務やサービスに影響を与えないシステムであるか、RFP を作成した際に明示していない可能性が考えられる。

プラットフォーム(CO₃)については「建設業」「情報通信業」「その他」の 3 業種で影響があることを示した。その中でも、情報通信業の偏回帰係数は 23.8 と他の業種よりも高いことが分かった。プラットフォームとは表 3-1 で示している通り、ハードウェアとソフトウェア (OS や DBMS) の複合体でありそのニーズや性能などを示している。つまり通信業では OS や DBMS などのプラットフォーム構成を制約され上で、システム性能を重要視する業種であることが考えられる。

開発スケジュール要求(CO₄)については「電気・ガス・熱供給・水道業」、「情報通信」、「金融・保険業」の 3 業種で影響があることを示した。偏回帰係数はそれぞれ、68.5, 29.3, -45.8 となった。特に「電気・ガス・熱供給・水道業」の影響が大きいことが分かる。この業種では、開発スケジュールが他の業種と比べても納期へ要求が厳しく、スケジュールの遅延がプロジェクトに影響を与えることになり、ソフトウェア開発ベンダはスケジュールを重要視していることが考えられる。また、金融業では偏回帰係数が負の値を示した。金融業の場合、業務の統合や法律改正などの理由により決まったスケジュールを遵守する必要が考えられるが、Q(Quality), C(Cost), D(Delivery)を考慮した場合、納期が短くなるとコストが下がることで、品質の低下が考えられる。逆に、スケジュールに余裕がある場合、見積もり工数も増えることになるので、例えば月ベースで必要な人数を確保するために見積もっていると考えられる。

発注要件の明確度と安定性(CO₅)については「建設業」、「製造業」、「電気・ガス・熱供給・水道業」、「流通業」、「その他」の 5 業種で影響があることを示した。この結果からほとんどの業種で発注要件の明確度と安定性を重要視していることが分かる。特に「電気・ガス・熱供給・水道業」の偏回帰係数が 116.2 となり非常に大きく影響していることが分かった。つまり、要件の変更が発生した場合、大きく見積もり工数に影響を与えることになる。「電気・ガス・熱供

給・水道業」のインフラ機関の場合、業務プロセスが明確になっており、業務にあった要件を明確にする必要がある。したがって、予め決まった業務運用に則り、発注時に外部インターフェースやデータの I/O が明確になっていることが考えられる。「建設業」では他の 4 業種と比べ、唯一偏回帰係数が負の値となった。この業種では、発注要件の明確度と安定度の条件が厳しい場合、見積もり工数が下がる結果となった。通常、発注要件が明確にならない場合、リスクを見越してソフトウェア開発ベンダは見積もり工数を高くすることが考えられる。しかし、建設業の場合は逆の結果が示された。この原因として考えられる原因は 2 つある。1 つ目の理由として、建設業では入札形式でソフトウェア開発ベンダを選定している。つまり、ソフトウェア開発ベンダは入札する際、RFP をソフトウェア開発ベンダに提示する際、予め要件が明確になっており、要件の変更があまり発生していないことが前提であることが考えられる。2 つ目の理由として、仮に要件が変更となった場合でも、下流工程で変更することは少なく、変更後、再見積もりをとることで後続の工数が増えるような状況は少ない業種である可能性が考えられる。

ユーザの参画割合 (CO_6) については、「その他」の 1 業種で影響があることを示した。これはほとんどの業種でユーザがプロジェクトに参画するかどうかはあまり重要視していないこととなる。「その他」の偏回帰係数を見た場合、-21.9 と負の値が検出されたが、これは予算や納期が厳しいプロジェクトにおいて、高くなっていることが推測することができる。考えられる原因として、ユーザが参画することで、IT 発注企業との打ち合わせによる工数の増加や仕様変更が発生することが考えられる。

先行モデルの流用と標準化の採用 (CO_7) についてもユーザの参画割合と同様に「その他」の 1 業種で影響があり、負の値を示した。つまり先行モデルの流用と標準化を採用しない方が見積もり工数を少なく抑えられることが分かる。ソフトウェア開発では過去に同様の機能がある場合それを流用したり、パッケージを使ったりすることで開発工数を減らす方法があるが、その他業種ではこれら先行モデルの流用や標準化の採用をしない方が見積もり工数を少なく抑えることになる。その理由としては、IT 発注企業からの要件に対して、機能やサービスが先行モデルやパッケージに合わせる事が難しく、カス

タマイズが多い結果となっていることが考えられる。

プロジェクト管理経験と能力(CO₈)については、「流通業」「金融業・保険業」の2業種で影響があることを示した。特に金融業・保険業では偏回帰係数が-72.0 となり、流通業の-27.7 と比較すると絶対値が3倍弱大きくなっており、金融業・保険業ではプロジェクト管理の経験と能力の影響力が非常に高いことが分かる。また、この2業種は共に偏回帰係数は負の値となった。これは経験が豊富なプロジェクト管理者がプロジェクトに参画した場合、見積もり工数は増えることになる。つまりこの現象はプロジェクト管理の経験と能力が高いメンバほど、IT 発注企業側の予算や納期などが厳しい状況のプロジェクトに配属されていることが考えられる。

アナリストの経験と能力(CO₉)については、「建設業」「電気・ガス・熱供給・水道業」、「情報通信業」、「サービス業」、「公務」の5業種が影響あることを示した。建設業以外の4業種は全て負の値が検出された。特に「電気・ガス・熱供給・水道業」の偏回帰係数は-110.1 となり絶対値が大きな値が示された。この業種では、プロジェクトの管理経験と能力が負となった理由と同様に、経験が豊富なアナリストがプロジェクトに参画した場合、見積もり工数は増えることになる。つまりアナリストの経験と能力が高いメンバほど、IT 発注企業側の予算や納期などが厳しい状況のプロジェクトに配属されていることが考えられる。一方、建設業のみ偏回帰係数が6.8 と正の値を示した。これはアナリストの経験と能力が高い場合、見積もり工数が上がることを示しているが、厳しくないプロジェクトへ経験と能力の高いアナリストを配置しているもしくは、特に考慮せずに配置している業種であると推測することが出来る。

SE, PG の経験と能力(CO₁₀)については今回どの業種も影響があることを示さなかった。ソフトウェア開発ベンダはソフトウェア工数を見積もる際にどのSEやPGを割り当てるのかということは決まっていないことが多く、設計工程や開発工程で技術者を招集することになる。そのため、上流工程での見積もり時にはこのSE, PGの経験と能力はあまり影響していると考えていないことになる。

続いて、業種ごとで多重共線性が発生しているかどうか確認した。その結果が表3-6に示す。表3-6からどの業種も多重共線性は発生していないことが分かった。

表 3-6 業種別での変動要因による VIF 結果一覧

	機能性	信頼性	プラットフォーム	開発スケジュール要求	発注要件の明確度と安定度	参画割合	先行モデルの流用と標準モデルの採用	プロジェクト管理の経験と能力	アナリストの経験と能力	SE・PGの経験と能力
建設業	－	1.03	1.10	－	1.09	－	－	－	1.05	－
製造業	－	1.00	－	－	1.00	－	－	－	－	－
電気・ガス・熱供給・水道業	－	－	－	1.00	1.03	－	－	－	1.03	－
情報通信業	－	1.04	1.03	1.01	－	－	－	－	1.02	－
流通業	－	1.01	－	－	1.01	－	－	1.02	－	－
金融業，保険業	－	1.02	－	1.01	－	－	－	1.01	－	－
サービス業	1.11	1.11	－	－	－	－	－	－	1.02	－
公務	1.00	－	－	－	－	－	－	－	1.00	－
その他	－	1.02	1.12	－	1.18	1.04	1.13	－	－	－

3.5 結言

本章では、過去プロジェクトの実績データから目的変数を見積もり工数、説明変数を10の生産性変動要因として、全業種から重回帰分析を実施し、4つ見積もり工数とそれに影響を与える生産性変動要因の重回帰モデル式を導出した。この導出した式から、見積もりに影響を与える生産性変動要因は CO_1 (機能性)、 CO_2 (信頼性)、 CO_5 (発注要件の明確度と安定度)、 CO_8 (プロジェクト管理の経験と能力)であることが分かった。また、この重回帰モデル式から特に偏回帰係数の絶対値が大きい、 CO_2 (信頼性)と CO_8 (プロジェクトの管理の経験と能力)の2つの変動要因が見積もりに対して大きく影響していることを導いた。次に、この抽出した要因が業種ごとでどのように影響しているのか。また、業種ごとで違う要因がないかを確認した。方法は業種ごとでグループに分類し、業種ごとでの重回帰分析を実施した。そしてそれぞれで重回帰モデル式を抽出した。抽出した結果から9業種のうち、2つの要因に影響がある業種が製造業と公務の2業種となった。製造業は「信頼性」、「発注要件の明確度と安定度」に影響しており、公務は「機能性」と「アナリストの経験と能力」であることが分かった。次に3つの要因に影響がある業種は、電気・ガス・熱供給・水道業、金融業・保険業、流通業そしてサービス業の4業種となった。電気・ガス・熱供給・水道業は「開発スケジュール要求」、「発注要件の明確度と安定度」、「アナリストの経験と能力」であり、金融業は「信頼性」、「開発助スケジュール要求」、「プロジェクトの経験と能力」であり、この3業種は特に他の業種と比較しても偏回帰係数が高かった。そして流通業は「信頼性」、「発注要件の明確度と安定度」、「プロジェクト管理の経験と能力」、サービス業は「機能性」、「信頼性」、「アナリストの経験と能力」となった。そして4つの要因に影響のある業種は建設業と情報通信業の2業種である。建設業は「信頼性」、「プラットフォーム」、「発注要件の明確度と安定度」、「アナリストの経験と能力」となり、特に「発注要件の明確度と安定度」については負の値となり、アナリストの経験と能力は正の値を示した。これは他の業種とし比較し、符号が逆となった。つまりこの業種は発注要件の明確度と安定度に強く影響され、

発注要件が明確でないまたは、安定していない場合、見積もり工数が下がる結果となった。また、情報通信業は「信頼性」、「プラットフォームの適合性」、「開発スケジュール要求」、「アナリストの経験と能力」となった。これら偏回帰係数の絶対値はどれも同じ値を示しており、平均的に影響している業種であることが分かった。最後に 5 つの要因に影響がある業種はその他の 1 業種のみとなった。その他は「信頼性」、「プラットフォーム」、「発注要件の明確度と安定度」、「ユーザの参画割合」、「先行モデルの流用と標準化モデルの採用」である。その他業種ではいくつかの業種が混ざっているため、多くの要因が含まれてしまった可能性がある。

本研究では以上の結果から業種ごとに見積もりに影響を与える生産性変動要因は特徴があり、それら業種ごとに影響を与える生産性変動要因を明確にした。

本研究から抽出した要因はソフトウェア開発ベンダ側が決めることであり、IT 発注企業側にはブラックボックスになっている部分である。つまり、ソフトウェア開発ベンダがこの要因をユーザに説明しても理解されにくい部分がある。今後、業種ごとにソフトウェア見積もりを見積もる際、抽出した回帰式から影響のある生産性変動要因を考慮することでソフトウェア開発ベンダはどの生産性変動要因に注目し、見積もりをとるのか考慮する必要がある。つまり、IT 発注企業に見積もり工数をより理解してもらうためには、プロジェクト計画時にソフトウェア開発ベンダにも、これら業種ごとに見積もりに影響を与える生産性変動要因についてきちんと理解してもらう必要があり、お互いが納得のいく見積もりは不可欠であると言える。

本研究で抽出した要因は見積もり工数を抽出するための判断としては、非常に有効であると言える。

次に本章の説明変数で利用した 10 の生産性変動要因のみに着目し、生産性変動要因から背後に共通する構成概念がどのようなものなのか関係性を確認する必要があると考えた。これにより今回使用している生産性変動要因の特性を確認することができる。つまり因子分析を行うことで共通因子とそれに影響を与える生産性変動要因が何かについて次章で検討を行う。

第4章

因子分析による因子の抽出

因子分析とはさまざまな観測変数の相関関係からその背後に共通して存在し、観測変数に影響を与えているような潜在変数（因子）を特定するための分析方法である。その分析方法の中でも、探索的因子分析、確認的因子分析がある。

探索的因子分析 (Exploratory Factor Analysis) とは観測される一連の変数から直接観測することが出来ない因子 (潜在変数) を探し出す方法のことである。

確認的因子分析 (Confirmatory Factor Analysis) とは事前にある程度明確な仮説が設定されており、観測変数に基づいて仮説として立てた因子構造がかどうか検証する方法のことである。

本章では、生産性変動要因を観測変数として探索的因子分析を実施し、抽出した因子がいくつになるのかについて述べ、その抽出した因子の構成概念を推測する。そしてその構成概念から因子と関係性の深い観測変数を抽出する。

次に、探索的因子分析から見つけ出した因子と観測変数の関係からいくつかのモデルを作成し、共分散構造分析にて最も適合度のよいモデルを評価し抽出する。その抽出したモデルから分散値や因子負荷量などから因子に影響度を与えている生産性変動要因を見つけ出す。

4.1 探索的因子分析の概要

探索的因子分析とは、データ主導型の分析であり、観測変数から共通因子を見つけ出し構成概念を探る分析である。図 4-1 は因子間や因子と観測変数との関係を表したパス図である。具体的には、因子と観測変数との関係について先行する制約や仮設を分析内におかないで、全ての観測変数との相関をもとに計算した結果から因子を測定する。分析内の全ての観測変数が関連するモデルを想定し、分析で得た固有値、推定値、因子負荷量を参考に因子数を決定する方法である。

観測変数間に相関があるとき、因子分析はその相関関係を説明する潜在変数を仮定する。プロジェクト i の観測変数 j の値を x_{ij} とするとき、因子分析モデルは共通因子 k の因子得点 f_{ik} を用いて x_{ij} を以下のように分解する

$$Z_{ij} = a_{j1}f_{i1} + a_{j2}f_{i2} + \cdots + a_{jk}f_{ik} + d_j u_{ij} \quad (\text{式 4.1})$$

k は因子数、 i 番目の観測対象の m 個の共通因子が $f_{i1}, f_{i2}, \dots, f_{ik}$ である。 a_{jk} は因子 k の観測変数 j に対する因子負荷量、 d_j は独自因子からの観測変数に対する係数であり、観測変数 j が独自因子 j から受ける影響の程度の指標である。また、 u_{ij} は独自因子スコアを示し、観測対象 i の観測変数 j の独自因子であり、共通因子からは説明が出来ないデータの振る舞いを説明する。つまりこれらを用いて k 個の構成概念を示すことになる。因子モデルを仮定するには3つの仮定が考えられる。

1 つ目の仮定が共通因子と独自因子の平均は 0 であり、分散が 1 となる。共通因子や独自因子は、実在しない仮想変数より計算の利便性を考慮し固定する。また、平均と分散を固定することで因子負荷量や独自係数の大きさを互いに比較することが可能となる。

2 つ目の仮定が共通因子と独自因子は互いに無相関であることが必要である。これは共通因子で表現できない因子が独自因子になるため、仮に共通因子が大

きくなるにつれ、独自因子も大きくなる場合、独自因子の設定の目的に反することになる。

3つ目の仮定が、独自因子は互いに無相関である。独自因子が大きくなるにつれ、別の独自因子も同時に大きくなる場合、その因子間には共通の因子が含まれることになる。したがって、独自因子はお互いに無相関で仮定する必要がある。

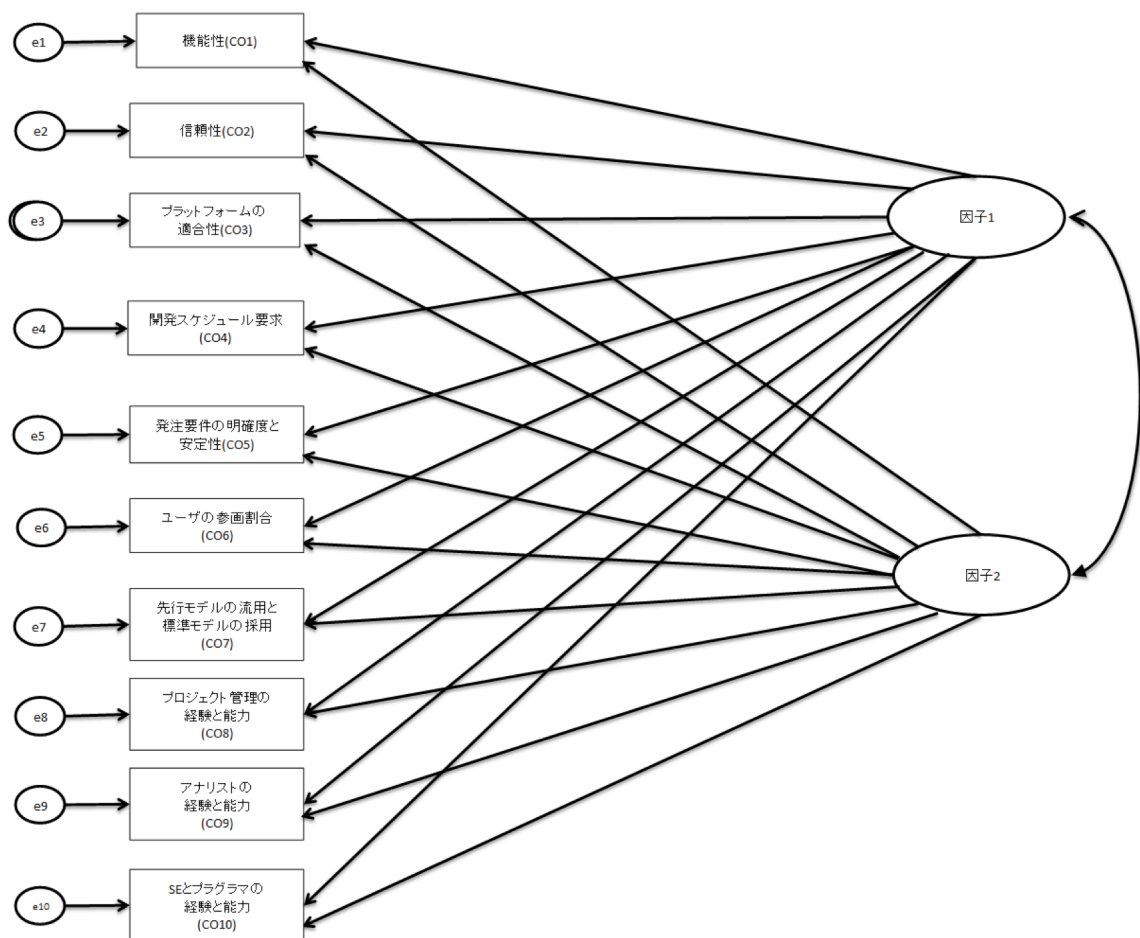


図 4-1 探索的因子分析におけるパス図

4.2 実験データ

本章で使用したデータは全2,008件のデータから生産性変動要因のみのデータを抽出し、その中から欠損データを除く1,721件を対象に実験した。

4.3 探索的因子分析の実験

ソフトウェア開発の見積もりがどのような因子に分かれるのか、観測変数から因子を探すことを目的に、探索的因子分析を実施した。実施した手順は以下、(i)から(iv)で実施した。統計ソフトは R を使用し、パッケージは psych における fa 関数から以下の概観について検証した。

- (i) 因子数の推定
- (ii) 因子パターン（因子負荷）の推定
- (iii) 因子の回転方法
- (iv) カテゴリカル因子分析における母数の推定

4.3.1 因子数の推定

前章表3-2 の生産性変動要因の相関係数一覧から抽出した相関行列を用いて、スクリーテストを実行した。スクリーテストは、相関行列の固有値を固有値順位に対してプロット（スクリープロット）し、これを最下位固有値から傾向線を引く。その傾向線から離れる固有値の順位が因子数となる。その結果を図4-2に示す。図4-2のスクリープロットの結果より、因子数が2であることが推定された。また、因子数を決定する基準として、カイザー・ガットマン基準がある。この基準は、固有値が1以上の因子数の採用を行う基準である。因子数に対する固有値を調査した結果を表4-1に示した。その結果から、固有値が1以上の因子数は3と推定された。したがって、スクリーテストでの結果とカイザー・ガットマン基準での因子数の結果に差異が発生した。但しこれら因子数は目安であるため、因子数が2と3の前後つまり1から4までを検証する必要がある。

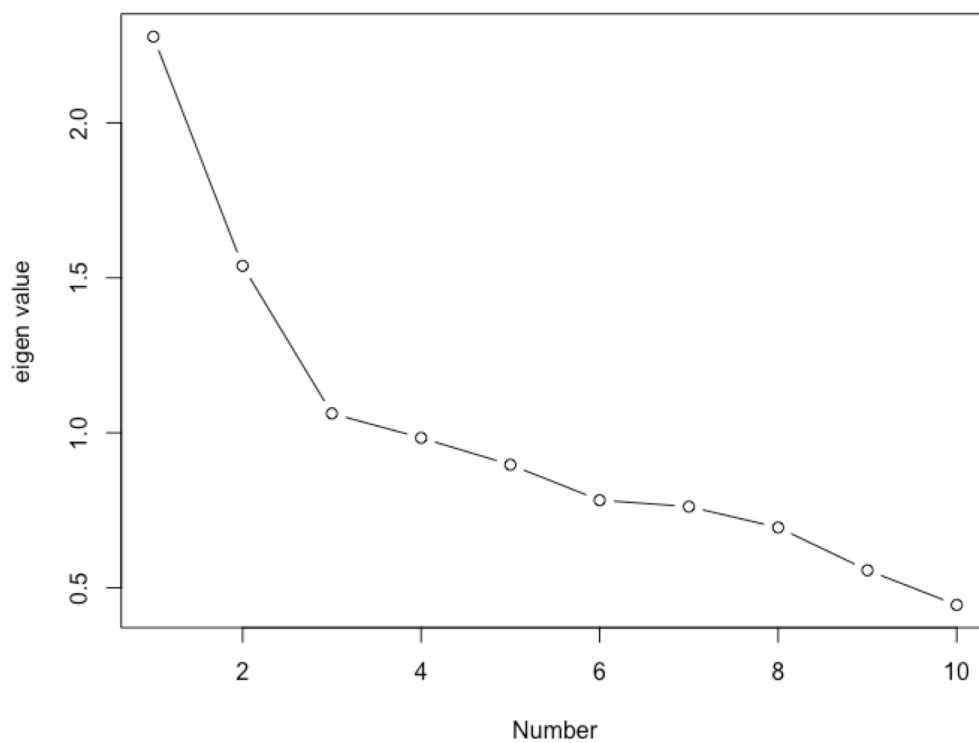


図 4-2 スクリープロット

表 4-1. 因子数に対しての固有値一覧

因子数	固有値	因子数	固有値
1	2.281	6	0.781
2	1.533	7	0.760
3	1.061	8	0.700
4	0.983	9	0.558
5	0.898	10	0.445

4.3.2 因子パターン（因子負荷）の推定

本研究での因子分析での因子抽出法では、 $n = 1,721$ より、第一因子から順に因子寄与が最大となるように最尤法を選択した。最尤法は漸近一致性、漸近有効性、漸近正規性といった性質を持っている。また、適合度指標を用いて因子数の決定を行うため、最尤推定法を適用した。

最尤推定法(maximum likelihood estimation)とはデータの発生機構に論理的な確率分布を仮定し、実際に想定されたデータを得る確率が最大となるような母数の値を推定値とする方法である。因子スコアは直接観測することができないため、分布を仮定する必要がある。最尤法では確率分布の1つであり、図4-3で示している標準正規分布を因子スコアの分布で仮定することになる。また、独自因子スコアに対しても標準正規分布を仮定することになる。尚、図4-3は因子1で分析した場合の因子負荷量を示している。

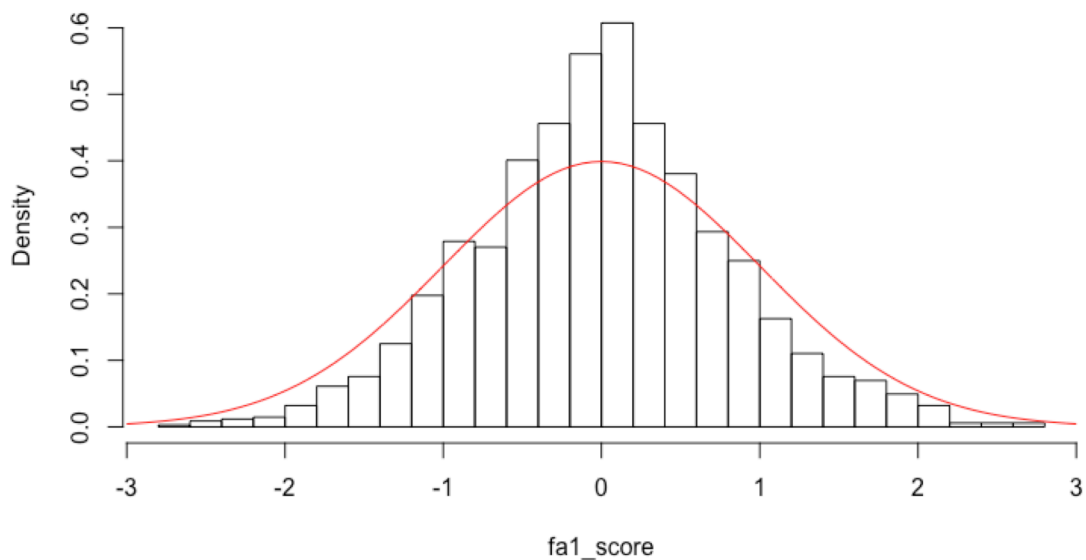


図 4-3 1 因子モデルの場合の正規分布

＜尤度関数の導出＞

単変量正規分布とは以下の式4-2で表すことが出来る．

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(x-\mu)}{2\sigma^2} \right\} \quad (\text{式4.2})$$

※ μ は平均, σ^2 は分散を表す．

標準正規分布は平均が0, 分散が1の正規分布を示す．つまり $N(0, 1)$ となる．

多変量モデルの場合, 共通因子を f_i , 因子負荷量 A と独自因子スコア u_i , 独自係数を D の対角行列とした場合, 因子分析モデルは式4.3となる．また, 式4.3の転置行列を考えた場合, 式4.4の通りである．

$$z_i = \begin{bmatrix} z_{i1} \\ z_{i2} \\ z_{i3} \end{bmatrix}, \quad A = \begin{bmatrix} a_{11} & \dots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{j1} & \dots & a_{jm} \end{bmatrix}, \quad f_i = \begin{bmatrix} f_{i1} \\ \vdots \\ f_{im} \end{bmatrix}$$

$$u_i = \begin{bmatrix} u_{11} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & u_{jp} \end{bmatrix}, \quad D = \begin{bmatrix} d_{11} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & d_{jp} \end{bmatrix}$$

$$z_i = Af_i + Du_i \quad (\text{式4.3})$$

$$Z = FA' + UD \quad (\text{式4.4})$$

式4.3は多変量因子分析モデルより f_i や u_i の分布は多変量正規分布が仮定され, 多変量正規分布は平均ベクトルと分散共分散行列によって規定される．

式4.2の単変量の正規分布の場合と同様に, 共通因子スコア, 独自因子スコアをそれぞれ, $f_i \sim N_m(\mathbf{0}, I)$, $u_i \sim N_p(\mathbf{0}, I)$ と仮定する． I は共通因子が無相関であるとし, 初期解を求める． z_i の平均ベクトルは $\mathbf{0}$ である．一方分散共分散行列は式4.5で表すことが出来る．

$$E[(Af_i + Du_i)(Af_i + Du_i)'] = AA' + D \quad (\text{式4.5})$$

となるので、 z_i は $N_p(O, AA' + D^2)$ に従うことになる。

ここで、 $\Sigma = AA' + D^2$ として、平均ベクトルが0、分散共分散行列が Σ の多変量正規分布は以下となる。

$$p(z|\Sigma) = (2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left[-\frac{1}{2} z' \Sigma^{-1} z\right] \quad (\text{式4.6})$$

そして、式4.6に従うそれぞれ独立な観測対象を z_i ($i = 1, 2, \dots, n$)をまとめた行列 Z とするとデータとして得られる確率は以下の確率分布の積となる。

$$p(Z|\Sigma) = \prod_{i=1}^n p(z_i|\Sigma) \quad (\text{式4.7})$$

式4.7に対数をとると、以下の式となる。

$$\log p(Z|\Sigma) = -\frac{1}{2} \sum_{i=1}^n \text{tr} \left[\Sigma^{-1} z_i z_i' \right] + \frac{n}{2} \log |\Sigma^{-1}| - C \quad (\text{式4.8})$$

$$= -\frac{1}{2} \text{tr} \left[\Sigma^{-1} \frac{1}{n} \sum_{i=1}^n z z' \right] + \frac{1}{2} \log |\Sigma^{-1}| - C \quad (\text{式4.9})$$

$$= -\frac{n}{2} \left[\text{tr} \left(\Sigma^{-1} R \right) - \log |\Sigma^{-1}| \right] - C \quad (\text{式4.10})$$

※尚、 R は標準化データ z_i の分散共分散行列とし、 C は母数を含まない項を示す。したがって対数尤度関数を $LL(\Sigma|Z)$ とすると式4.11となる。

$$LL(\Sigma|Z) = \log |\Sigma^{-1}| - \text{tr}(\Sigma^{-1} R) \quad (\text{式4.11})$$

よって式4.11の対数尤度関数の $LL(\Sigma|Z)$ の A と D^2 を最大化した結果が最尤推定法となる。

4.3.3 因子の回転方法

因子負荷には初期解と回転解がある．因子の回転にはいくつか種類があり，2因子以上の場合は回転解が適用される．その回転解の種類にはいくつかあり，代表的な回転解がバリマックス（直交）回転と，プロマックス（斜交）回転である．

バリマックス回転による直交解は因子が無相関である数学的制約の下で，単純構造を前提に初期解を変換する．一方，プロマックス回転による斜交解は，2軸を別々に回転させることで因子を見つけやすくし，制約なしに単純構造を前提に初期解を変換する．因子1と因子2における相関係数を r_{f_1, f_2} とすると，2つの因子間の共分散を各因子の標準偏差の積で除することによって求めることができる．各因子軸をベクトル表記 f_1 , f_2 で表記した場合，幾何学的には因子間共分散は内積 $\mathbf{f}_1 \cdot \mathbf{f}_2$ で，各因子の標準偏差はスカラー値， $|f_1|$, $|f_2|$ で表現できる．つまり以下の式4.12が成り立つ．

$$r_{f_1, f_2} = \frac{\text{因子1と因子2の共分散}}{\text{因子1の標準偏差} \cdot \text{因子2の標準偏差}} = \frac{\mathbf{f}_1 \cdot \mathbf{f}_2}{|f_1||f_2|} \quad (\text{式4.12})$$

各因子の標準偏差は1であり且つ， $\mathbf{f}_1 \cdot \mathbf{f}_2 = |f_1||f_2|\cos\theta$ から，以下式4.13が成り立つ．

$$r_{f_1, f_2} = \cos\theta \quad (\text{式4.13})$$

ただし， θ は因子1の軸と因子2の軸のなす角を示し，2因子が直交している場合は，そのなす角は， $\frac{\pi}{2}$ ラジアンとなるため， $r_{f_1, f_2} = \cos\frac{\pi}{2}=0$ となり無相関となる．一方，2つの因子軸が斜交した場合，なす角度は $0 < \theta < \frac{\pi}{2}$, $\frac{\pi}{2} < \theta < \pi$ となるため，相関係数は0を除く-1から1の範囲で動くことになる．したがって因子に相関を認めることによって，因子は斜めに交わることになる．

本研究では，因子数1から因子数4の可能性もあるため，初期解と回転解それぞれで実施する．また因子間相関行列の相関は 上記の通り，因子間の相関がな

しということは考えにくいため、プロマックス（斜交）回転を選択した。尚、分析後に2因子以上の場合は因子間相関について検証する。

4.3.4 カテゴリカル因子におけ母数の推定

4.3.1 節の因子パターンの推定からは, 因子数が 2 または 3 どちらにするのか判断することが難しかった. そこで因子数が 2 の前後を含めた因子数が 1 から 3 と, 因子数 3 の前後を含めた因子数が 2 と 4, つまり因子数 1 から 4 までのそれぞれのパターンでどの因子数が最適なのか確認し評価した. その結果が以下となる. 尚, 因子数が 1 の場合は解の推定方法は最尤推定, 回転は初期解 (none) を指定し, 因子分析を実施した. また, 因子数が 2 から 4 では, 解の推定方法を最尤推定, 回転はプロマックス回転を指定し, 因子分析を実行した結果より因子負荷行列を抽出した. その結果が以下である.

< 因子数 1 での実行結果 >

Factor Analysis using method = ml

Call: fa(r = business, nfactors = 1, rotate = "none", fm = "ml")

Standardized loadings (pattern matrix) based upon correlation matrix

	ML1	h2	u2	com
機能性	0.44	0.1909	0.81	1
信頼性	0.12	0.0150	0.98	1
プラットフォーム	0.44	0.1924	0.81	1
開発スケジュール要求	0.04	0.0016	1.00	1
発注要件の明確度と安定度	0.55	0.3064	0.69	1
参画割合	0.25	0.0634	0.94	1
先行モデルの流用と標準モデルの採用	0.48	0.2351	0.76	1
プロジェクト管理の経験と能力	0.38	0.1454	0.85	1
アナリストの経験と能力	0.37	0.1389	0.86	1
SE.PG の経験と能力	0.43	0.1887	0.81	1

	ML1
SS loadings	1.48
Proportion Var	0.15
Mean item complexity = 1	
Test of the hypothesis that 1 factor is sufficient.	
The degrees of freedom for the null model are 45 and the objective function was 1.09 with Chi Square of 1867.92	
The degrees of freedom for the model are 35 and the objective function was 0.49	
The root mean square of the residuals (RMSR) is 0.09	
The df corrected root mean square of the residuals is 0.1	
The harmonic number of observations is 1721 with the empirical chi square 1305.08 with prob < 1.3e-251	
The total number of observations was 1721 with Likelihood Chi Square = 833.76 with prob < 1.8e-152	
Tucker Lewis Index of factoring reliability = 0.436	
RMSEA index = 0.115 and the 90 % confidence intervals are 0.108 0.122	
BIC = 572.99	
Fit based upon off diagonal values = 0.71	
Measures of factor score adequacy	
	ML1
Correlation of scores with factors	0.81
Multiple R square of scores with factors	0.65
Minimum correlation of possible factor scores	0.30

< 因子数 2 での実行結果 >

Factor Analysis using method = ml

Call: fa(r = business, nfactors = 2, rotate = "promax", fm = "ml")

Standardized loadings (pattern matrix) based upon correlation matrix

	ML1	ML2	h2	u2	com
機能性	-0.05	0.63	0.3774	0.62	1.0
信頼性	-0.19	0.36	0.1212	0.88	1.5
プラットフォーム	0.10	0.41	0.2064	0.79	1.1
開発スケジュール要求	-0.03	0.09	0.0076	0.99	1.3
発注要件の明確度と安定度	0.13	0.51	0.3212	0.68	1.1
参画割合	0.14	0.15	0.0547	0.95	2.0
先行モデルの流用と標準モデルの採用	0.19	0.36	0.2107	0.79	1.5
プロジェクト管理の経験と能力	0.82	-0.19	0.6044	0.40	1.1
アナリストの経験と能力	0.71	-0.14	0.4632	0.54	1.1
SE.PG の経験と能力	0.29	0.22	0.1735	0.83	1.9

	ML1	ML2
SS loadings	1.36	1.18
Proportion Var	0.14	0.12
Cumulative Var	0.14	0.25
Proportion Explained	0.53	0.47
Cumulative Proportion	0.53	1.00

With factor correlations of

	ML1	ML2
ML1	1.00	0.32
ML2	0.32	1.00

Mean item complexity = 1.4

Test of the hypothesis that 2 factors are sufficient.

The degrees of freedom for the null model are 45 and the objective function was 1.09 with Chi Square of 1867.92

The degrees of freedom for the model are 26 and the objective function was 0.11

The root mean square of the residuals (RMSR) is 0.04

The df corrected root mean square of the residuals is 0.05

The harmonic number of observations is 1721 with the empirical chi square 234.96 with prob < 1.5e-35

The total number of observations was 1721 with Likelihood Chi Square = 192.63 with prob < 2.3e-27

Tucker Lewis Index of factoring reliability = 0.842

RMSEA index = 0.061 and the 90 % confidence intervals are 0.053 0.069

BIC = -1.09

Fit based upon off diagonal values = 0.95

Measures of factor score adequacy

	ML1	ML2
Correlation of scores with factors	0.86	0.80
Multiple R square of scores with factors	0.74	0.64
Minimum correlation of possible factor scores	0.48	0.27

< 因子数3での実行結果 >

警告メッセージ:

fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, で:

A loading greater than abs(1) was detected. Examine the loadings carefully.

< 因子数4での実行結果 >

警告メッセージ:

fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, で:

A loading greater than abs(1) was detected. Examine the loadings carefully.

上記結果から、因子数が1または2の場合は、不適解(heywood case)は検出されなかったが、因子数が3以上の場合は不適解 (heywood case) が抽出された。不適解とは計算途中で共通性の推定値が 1 を超えるなど、導かれた解が適切でないことを示す。原田ら[13]は、その原因を下記 3 つと述べている。

< 不適解で考えられる原因 >

- (1) サンプルサイズが小さい。
- (2) データ内部で局面的に相関が高い項目がある。
- (3) 真の因子数よりも過大に因子数を見積もって推定してしまっている。

本研究のデータは1,721件とサンプルサイズが小さいことは考えにくい。また、データ間の相関係数は表3-2より、相関係数の一番高い生産性変動要因はCO₁とCO₅の相関係数が0.34と弱い相関ありと示しており、つまりこの結果からも不適切と考えられる原因の(1)、(2)については考え難い。したがって(3)の真の因子数よりも過大に因子数を見積もって推定している可能性が高いと考えた。そこで因子数3と因子数4については実験対象から外した。

4.4 探索的因子分析の実験の評価

以上の結果より、今回対象のモデルは因子数が1か2が対象となった。そこでこれら2つのモデルのうち、どちらが最適なモデルとなっているのかについて評価した。探索的因子分析にはRMSEA(Root Mean Square Error of Approximation)やBIC(Bayesian Information Criterion)などの指標があり、その評価基準でモデルを判断していく。その評価基準について以下説明する。

＜RMSEA＞

RMSEAは、モデルの分布と真の分布との乖離度を1自由度あたりの値として評価する指標である。RMSEAは最尤推定法を利用して母数の推定を行う必要があり、最尤推定法を用いて因子分析モデルの推定を行うと、 χ^2 値と尤度関数の最小値 f_{min} 値が得られることになる。以下式4.14で表すことができる。

$$RMSEA = \sqrt{\max\left(\frac{\chi^2}{df \times n} - \frac{1}{n}, 0\right)} \quad (\text{式4.14})$$

式4.14の χ^2 はモデル適合度を検討するための χ^2 検定であり、観測対象の数が十分大きい時は式4.16の自由度df (degree of freedom) が式4.15の χ^2 分布に近似的に従うことを利用して検定を行う。

$$\chi^2 = n \times f_{min} = n \{ \text{tr}(\widehat{\Sigma} S^{-1}) - \log |\widehat{\Sigma} S^{-1}| - p \} \quad (\text{式4.15})$$

Sは標本共分散行列、 $\widehat{\Sigma}$ は母数で再現された共分散行列、nは観測対象の数、pは観測変数の数を示している。

$$df = \frac{1}{2} p(p+1) - \text{自由母数の数} \quad (\text{式4.16})$$

<BIC>

BICとは、式3.37のAICと同様に複数のモデルが候補とあげられる時、モデルごとに情報基準を算出し、それが最小となるモデルを採用することになる。情報基準 χ^2 値と自由度dfを利用して以下のように計算される。

$$BIC = \chi^2 - \log(d)df \quad (\text{式4.17})$$

■ 因子1のケース

因子負荷量が0.4以上を基準に考えた場合、表4-2より因子1は「機能性」、「プラットフォーム」、「発注要件の明確度と安定度」、「先行モデルの流用と標準モデルの採用」、「SE、PGの経験と能力」の5つが抽出された。ところが、発注要件の明確度と安定度の共通性は高いが、全体的に共通性が低く、独自因子が高い。つまり共通因子で説明ができない他の変数が大きく影響していると考えられるため、因子1は良いモデルと判断することは難しいと考えられる。

表 4-2. 1因子で実行した因子分析結果

	因子 1	共通性	独自因子
機能性	0.44	0.191	0.81
信頼性	0.12	0.015	0.98
プラットフォーム	0.44	0.192	0.81
開発スケジュール要求	0.04	0.002	1.00
発注要件の明確度と安定度	0.55	0.306	0.69
参画割合	0.25	0.063	0.94
先行モデルの流用と標準モデルの採用	0.48	0.235	0.76
プロジェクト管理の経験と能力	0.38	0.145	0.85
アナリストの経験と能力	0.37	0.139	0.86
SE、PGの経験と能力	0.43	0.189	0.81

■ 因子数が 2 のケース

因子負荷量が 0.4 以上を基準と考えた場合、表 4-3 より因子 1 は「プロジェクト管理の経験と能力」，「アナリストの経験と能力」となり，因子 2 は「機能性」，「プラットフォーム」，「発注要件の明確度と安定度」が抽出された。これら観測変数は共通性も高いため，モデルとしては適合している可能性がある。そこで以下，適合度指標から確認し，モデルの評価を行った。

表 4-3. 2 因子で実行した因子分析結果

	因子 1	因子 2	共通性	独自因子
機能性	-0.05	0.63	0.38	0.62
信頼性	-0.19	0.36	0.12	0.88
プラットフォーム	0.10	0.41	0.21	0.79
開発スケジュール要求	-0.03	0.09	0.01	0.99
発注要件の明確度と安定度	0.13	0.51	0.32	0.68
ユーザの参画割合	0.14	0.15	0.05	0.95
先行モデルの流用と標準モデルの採用	0.19	0.36	0.21	0.79
プロジェクト管理の経験と能力	0.82	-0.19	0.60	0.40
アナリストの経験と能力	0.71	-0.14	0.46	0.54
SE. PG の経験と能力	0.29	0.22	0.17	0.83

表 4-4 の適合度指標より，因子数 1 と因子数 2 それぞれの RMSEA と BIC を検証し，それぞれの結果が $RMSEA_{f1}=0.115 > RMSEA_{f2}=0.061$ ， $BIC_{f1}=572.99 > BIC_{f2}=-1.09$ となり，因子数 2 の方は BIC が低く，RMSEA は 0.061 と 5%を上回ってしまったが，あてはまりのよいモデルであると判断することが出来た。

表 4-4 因子適合度の評価

因子数	RMSEA	BIC
1	0.115	572.99
2	0.061	-1.09

そこで因子数が2の表4-3から共通性を観察すると、「信頼性(C02)」、「開発スケジュール要求(C04)」、「ユーザの参画割合(C06)」、「SEとPGの経験と能力(C010)」が低いことが分かった。そのため、これら4つの観測変数を削除し、再度因子分析を実施した。その結果が以下となる。

<因子数2の探索的因子分析（再分析）>

Factor Analysis using method = ml

Call: fa(r = business_next2, nfactors = 2, rotate = "promax", fm = "ml")

Standardized loadings (pattern matrix) based upon correlation matrix

	ML1	ML2	h2	u2	com
機能性	-0.11	0.57	0.30	0.70	1.1
プラットフォーム	0.02	0.43	0.19	0.81	1.0
発注要件の明確度と安定度	-0.01	0.62	0.38	0.62	1.0
先行モデルの流用と標準モデルの採用	0.09	0.45	0.24	0.76	1.1
プロジェクト管理の経験と能力	0.86	-0.06	0.72	0.28	1.0
アナリストの経験と能力	0.63	0.04	0.41	0.59	1.0

	ML1	ML2
SS loadings	1.14	1.09
Proportion Var	0.19	0.18
Cumulative Var	0.19	0.37
Proportion Explained	0.51	0.49
Cumulative Proportion	0.51	1.00

With factor correlations of

	ML1	ML2
ML1	1.0	0.3
ML2	0.3	1.0

Mean item complexity = 1

Test of the hypothesis that 2 factors are sufficient.

The degrees of freedom for the null model are 15 and the objective function was 0.75 with Chi Square of 1284.54

The degrees of freedom for the model are 4 and the objective function was 0.01

The root mean square of the residuals (RMSR) is 0.01

The df corrected root mean square of the residuals is 0.03

The harmonic number of observations is 1721 with the empirical chi square 10.94 with prob < 0.027

The total number of observations was 1721 with Likelihood Chi Square = 11.67 with prob < 0.02

Tucker Lewis Index of factoring reliability = 0.977

RMSEA index = 0.033 and the 90 % confidence intervals are 0.012 0.056

BIC = -18.13

Fit based upon off diagonal values = 1

Measures of factor score adequacy

	ML1	ML2
Correlation of scores with factors	0.88	0.79
Multiple R square of scores with factors	0.77	0.62
Minimum correlation of possible factor scores	0.54	0.24

表 4-5. 2 因子で実行した因子分析結果

	因子 1	因子 2	共通性	独自因子
機能性	-0.11	0.57	0.30	0.70
プラットフォーム	0.02	0.43	0.19	0.81
発注要件の明確度と安定度	-0.01	0.62	0.38	0.62
先行モデルの流用と標準モデルの採用	0.09	0.45	0.24	0.76
プロジェクト管理の経験と能力	0.86	-0.06	0.72	0.28
アナリストの経験と能力	0.63	0.14	0.41	0.59

BIC が-18.13, RSMEA が 0.033 となり, 観測変数 10 の場合の分析と比較し, 明らかにモデルとしては改善され, 非常に良いモデルとなった. また, 因子負荷量についても因子 1 では「プロジェクト管理の経験と能力」, 「アナリストの経験と能力」, 因子 2 では「機能性」, 「プラットフォーム」, 「発注要件の明確度と安定度」, 「先行モデルの流用と標準モデルの採用」全て 0.4 以上となった. そして, 累積寄与率も 25%から 37%と改善された. したがって, 表 4-5 の因子負荷量と独自係数の結果から式 4-1 に適用した場合, 式 4-18 から式 4-23 となる.

$$z_{i1} = (-0.11)f_{i1} + 0.57f_{i2} + 0.83u_{i1} \quad (\text{式 4.18})$$

$$z_{i2} = 0.02f_{i1} + 0.43f_{i2} + 0.90u_{i2} \quad (\text{式 4.19})$$

$$z_{i3} = (-0.10)f_{i1} + 0.62f_{i2} + 0.79u_{i3} \quad (\text{式 4.20})$$

$$z_{i4} = 0.09f_{i1} + 0.45f_{i2} + 0.87u_{i4} \quad (\text{式 4.21})$$

$$z_{i5} = 0.86f_{i1} - 0.06f_{i2} + 0.53u_{i5} \quad (\text{式 4.22})$$

$$z_{i6} = 0.63f_{i1} + 0.14f_{i2} + 0.77u_{i6} \quad (\text{式 4.23})$$

また、因子相関は表 4-6 より相関係数は 0.30 と示した。これは表 4-7 から、弱い正の相関があることを示している。今回の実験では、回転解でプロマックス回転を指定したが、結果的に因子間に相関があることで問題ないことが立証された。

表 4-6 因子間相関

	因子 1	因子 2
因子 1	1.00	0.30
因子 2	0.30	1.00

表 4-7 相関係数の値と相関(参考値)

相関係数	相関
$-1 \leq r \leq -0.7$	強い負の相関
$-0.7 \leq r \leq -0.4$	中程度の負の相関
$-0.4 \leq r \leq -0.2$	弱い負の相関
$-0.2 \leq r \leq 0.2$	ほとんど相関なし
$0.2 \leq r \leq 0.4$	弱い正の相関
$0.4 \leq r \leq 0.7$	中程度の正の相関
$0.7 \leq r \leq 1$	強い正の相関

4.5 探索的因子分析の考察

2つの因子については、表4-8が示す通り、因子1については、「プロジェクトの管理能力」、「アナリストの経験と能力」の2つの観測変数との関係性が強いことが分かった。因子2については、「機能性」、「プラットフォームの適合性」、「発注要件の明確度と安定度」、「先行モデルの流用と標準モデルの採用」の4つの観測変数との関係性が強いことが分かった。そこで、これら2つの因子それぞれを次のように命名した。因子1はプロジェクトの上流工程での業務がメインとなるため、「上流工程の管理能力」と命名した。また、因子2は下流工程で開発をする際に必要な条件となりこの条件が揃うことで開発がしやすくなることを考慮し、「開発の容易性」と命名した。

本実験では因子数が3以上の探索的因子分析では、不適解が見つかり、探索的因子分析では確認することが出来なかった。それでは因子数が3以上のモデルは適正でないのか。因子数が2つのモデルが正しいモデルなのか。また、他のモデルでより適合度のよいモデルがないかについて検証する必要があると考えた。そこで探索的因子分析では出来なかったモデルを独自に作成し、確認的因子分析よりいくつかのモデルについて検証を行った。

表 4-8 因子と観測変数の割り当て

因子	生産性変動要因
因子1 上流工程の管理能力	プロジェクト管理の経験と能力(C08)
	アナリストの経験と能力(C09)
因子2 開発の容易性	機能性(C01)
	プラットフォームの適合性(C03)
	発注要件の明確度と安定度(C05)
	先行モデルの流用と標準モデルの採用(C07)

4.6 確認的因子分析

確認的因子分析とは、事前に仮説を立て観測変数を基に仮説として立てた因子構造が正しいかどうかを検証する分析である。また、構成概念と観測変数の関係性以外にも、パス図での解析、構成概念間の影響関係や観測変数間の影響など測定方程式や構造方程式などを使い検討することが可能である。4.5節までの因子負荷量と共通因子スコア、独自因子の式を確認的因子モデルで表すと以下の通りとなる。

$$\begin{aligned}
 v_1 &= a_{11} \times f_1 && + u_1 \\
 v_2 &= a_{21} \times f_1 && + u_2 \\
 v_3 &= a_{31} \times f_1 && + u_3 \\
 v_4 &= a_{41} \times f_1 && + u_4 \\
 v_5 &= && a_{52} \times f_2 + u_5 \\
 v_6 &= && a_{62} \times f_2 + u_6
 \end{aligned} \tag{式4.24}$$

これを行列にすると式 4-24 となり、式 4-25 の式で表すことができる。

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix} = \begin{bmatrix} a_{11} & 0 \\ a_{21} & 0 \\ a_{31} & 0 \\ a_{41} & 0 \\ 0 & a_{52} \\ 0 & a_{62} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} \tag{式 4.25}$$

$$v = A\mathbf{f} + \mathbf{u} \tag{式4.26}$$

探索的因子分析では、全ての要素を推定していたが、確認的因子分析モデルでは各因子から影響を受けない観測変数は全て 0 固定で定義する。この時、観測変数の分散共分散行列は $(A\mathbf{f} + \mathbf{u})(A\mathbf{f} + \mathbf{u})'$ の期待値を計算することで以下となる。

$$\Sigma = AE[\mathbf{f}\mathbf{f}']A' + E[\mathbf{u}\mathbf{u}'] = A\Psi A' + \Theta \tag{式4.27}$$

Ψ は因子間相関行列， Θ は誤差変数の分散共分散行列を示す．

式 4-27 より分散共分散行列について母数の関数で表現することを構造化と呼び， Σ を構造化された分散共分散行列と言われ，共分散構造分析 (SEM : Structural Equation Model) と呼ばれる．

4.5 節までは探索的因子分析から 2 因子つまり 2 つの構成概念を抽出した．そしてそれぞれの因子と観測変数の関係性を導いた．まず，探索的因子分析から抽出した因子構造が正しいかどうか確認する．そして，観測したモデル以外に 1 因子や探索的因子分析では不適解が出力された 3 因子の場合など仮説モデルを立て，母数の推定を実施する．尚，共分散構造分析の手順は以下の流れで検証を行う．

- ① モデルの仮説
- ② モデルの適合度の評価
- ③ パス図の描画
- ④ 構成概念スコアの検証
- ⑤ 信頼性の確認

4.6.1 モデルの仮説

共分散構造分析では、モデルの仮説にはパス図による仮説、測定方程式と構造方程式の2つの方法がある。本研究では、これら3つの方法から仮説を立て進めていく。

【仮説モデル】

表4-9では、5つのモデルを仮定し、因子と観測変数とのグループ化をまとめた。モデル1については、探索的因子分析から抽出したモデルを検証する。このモデルは2つの因子を抽出したが、上流工程の管理能力が高い場合、開発の容易性に繋がることを考慮し、因子間の影響関係についても仮定する。モデル2は探索的因子分析ではあまり良いモデルとならなかったが、再度因子数が1つの場合についての検証を行う。モデル3については、因子名が「ベンダの経験と能力」の可能性を考慮し、モデル1の因子1に「SEとPGの経験と能力」を加えたモデルを作成した。モデル4については、モデル3で加えた「SEとPGの管理と経験」が上流工程ではなく、下流工程での「開発のしやすさ」に影響がある可能性があることを考慮し、モデル3因子1から外し、因子2に「SE、PGの経験との能力」を追加した。モデル5に関しては、信頼性に着目し、信頼性は2つの因子には含まれてないことを考慮し、因子3を定義し、因子数が3であることを考慮した。つまり、モデル1の2因子以外に信頼性は別の因子と仮定し、モデル1に因子3「信頼性」として追加しモデルを仮定した。

表4-9 モデルの仮説

モデル	因子	観測変数
モデル1	因子1： 上流工程での管理能力	<ul style="list-style-type: none"> ・プロジェクト管理の能力と経験(CO₈) ・アナリストの経験と能力(CO₉)
	因子2： 開発のしやすさ	<ul style="list-style-type: none"> ・機能性(CO₁) ・プラットフォーム(CO₃) ・発注要件の明確度と安定度(CO₅) ・先行モデルの流用と標準化モデルの採用(CO₇)
モデル2	因子1	<ul style="list-style-type: none"> ・機能性(CO₁) ・プラットフォーム(CO₃) ・発注要件の明確度と安定度(CO₅) ・先行モデルの流用と標準化モデルの採用(CO₇) ・プロジェクト管理の能力と経験(CO₈) ・アナリストの経験と能力(CO₉) ・SE, PGの経験と能力(CO₁₀)
モデル3	因子1： ベンダの管理能力	<ul style="list-style-type: none"> ・プロジェクト管理の能力と経験(CO₈) ・アナリストの経験と能力(CO₉) ・SE, PGの経験と能力(CO₁₀)
	因子2： 開発のしやすさ	<ul style="list-style-type: none"> ・機能性(CO₁) ・プラットフォーム(CO₃) ・発注要件の明確度と安定度(CO₅) ・先行モデルの流用と標準化モデルの採用(CO₇)
モデル4	因子1： 上流工程での管理能力	<ul style="list-style-type: none"> ・プロジェクト管理の能力と経験(CO₈) ・アナリストの経験と能力(CO₉)
	因子2： 開発のしやすさ	<ul style="list-style-type: none"> ・機能性(CO₁) ・プラットフォーム(CO₃)

		<ul style="list-style-type: none"> ・発注要件の明確度と安定度 (CO₅) ・先行モデルの流用と標準化モデルの採用 (CO₇) ・SE, PGの経験と能力 (CO₁₀)
5モデル	因子1： 上流工程での管理能力	<ul style="list-style-type: none"> ・プロジェクト管理の能力と経験 (CO₈) ・アナリストの経験と能力 (CO₉)
	因子2： 開発のしやすさ	<ul style="list-style-type: none"> ・機能性 (CO₁) ・プラットフォーム (CO₃) ・発注要件の明確度と安定度 (CO₅) ・先行モデルの流用と標準化モデルの採用 (CO₇)
	因子3： 信頼性	信頼性 (CO ₂)

【測定方程式/構造方程式】

上記仮説モデルから以下、測定方程式と構造方程式を定義した。測定方程式とは構成概念が複数の観測変数によってどのように測定されているのかを表す方程式である。また、構造方程式とは変数間の影響関係を表現した式である。具体的には、「構成概念→構成概念」，「観測変数→観測変数」，「観測変数→構成概念」の3通りのパターンがある。そこで表4-9をそれぞれ測定方程式、構造方程式で示したモデルが以下、式4.28から式4.32となる。

モデル1: $Factor1 = \sim CO_8 + CO_9$

$Factor2 = \sim CO_1 + CO_3 + CO_5 + CO_7$

$Factor2 \sim Factor1$ (式4.28)

モデル2: $Factor1 = \sim CO_1 + CO_3 + CO_5 + CO_7 + CO_8 + CO_9$ (式4.29)

モデル3: $Factor1 = \sim CO_8 + CO_9 + CO_{10}$

$Factor2 = \sim CO_1 + CO_3 + CO_5 + CO_7$

$Factor2 \sim Factor1$ (式4.30)

モデル4: $Factor1 = \sim CO_8 + CO_9$
 $Factor2 = \sim CO_1 + CO_3 + CO_5 + CO_7 + CO_{10}$
 $Factor2 \sim Factor1$ (式4.31)

モデル5: $Factor1 = CO_8 + CO_9$
 $Factor2 = CO_1 + CO_3 + CO_5 + CO_7 + CO_{10}$
 $Factor3 = CO_2$
 $Factor2 \sim Factor1$ (式4.32)

そして、上記式のモデルは以下の図4-4から図4-8のパス図で表現することができる。

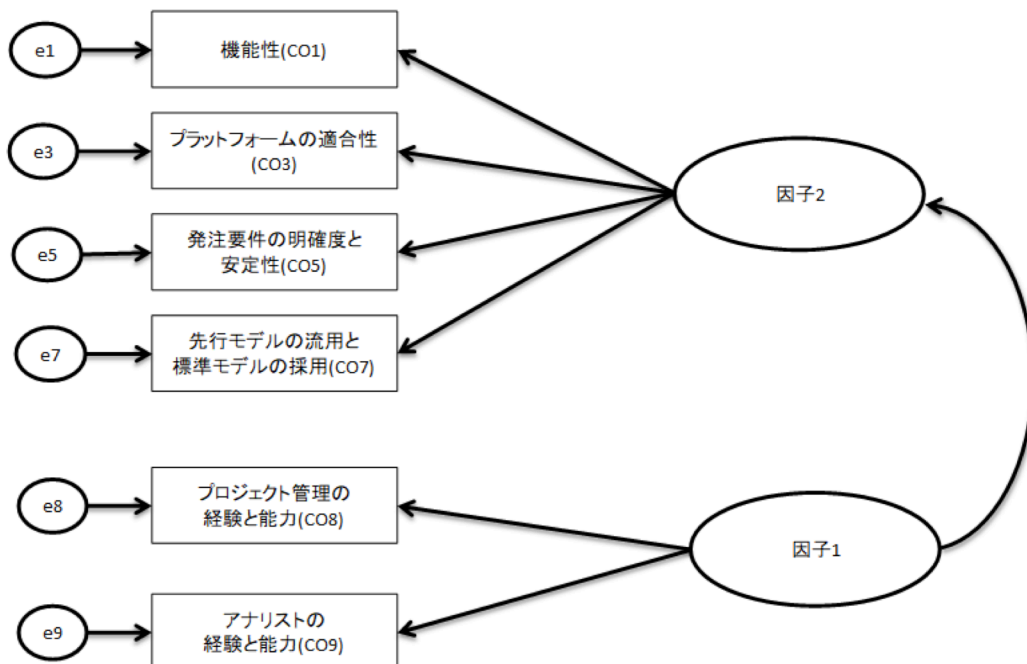


図 4-4 モデル1パス図

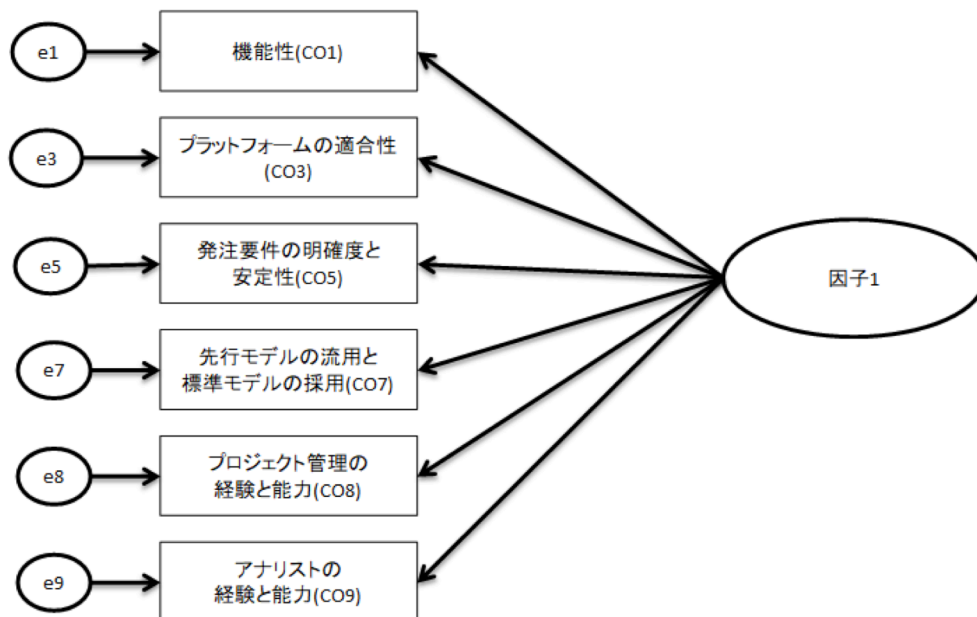


図 4-5 モデル 2 パス図

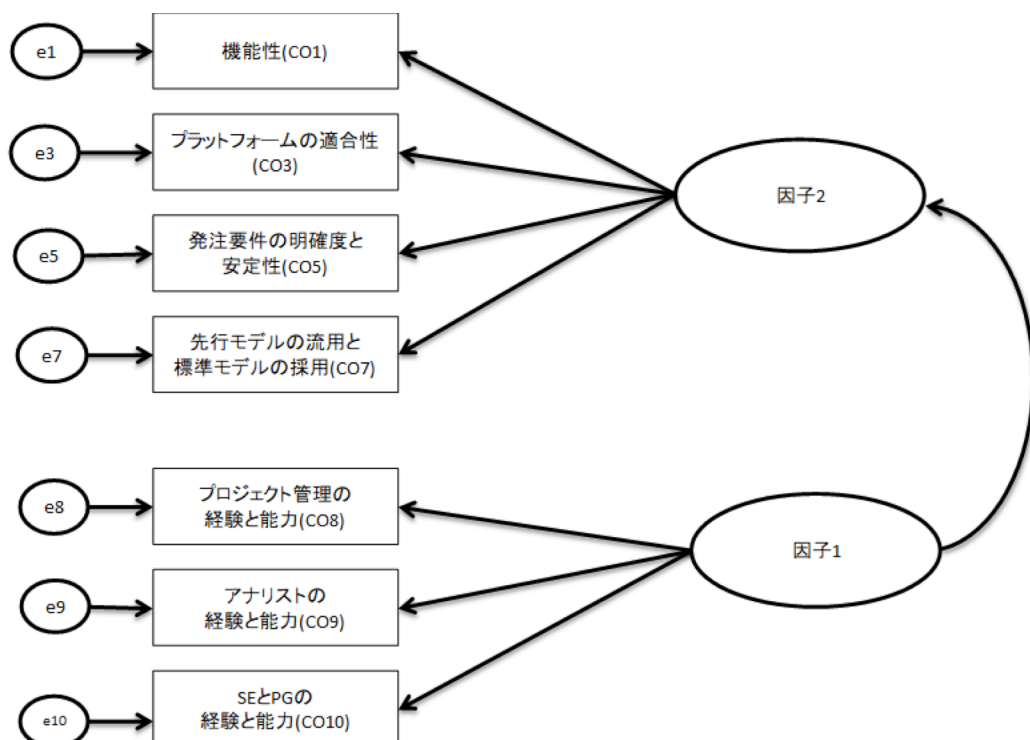


図 4-6 モデル 3 パス図

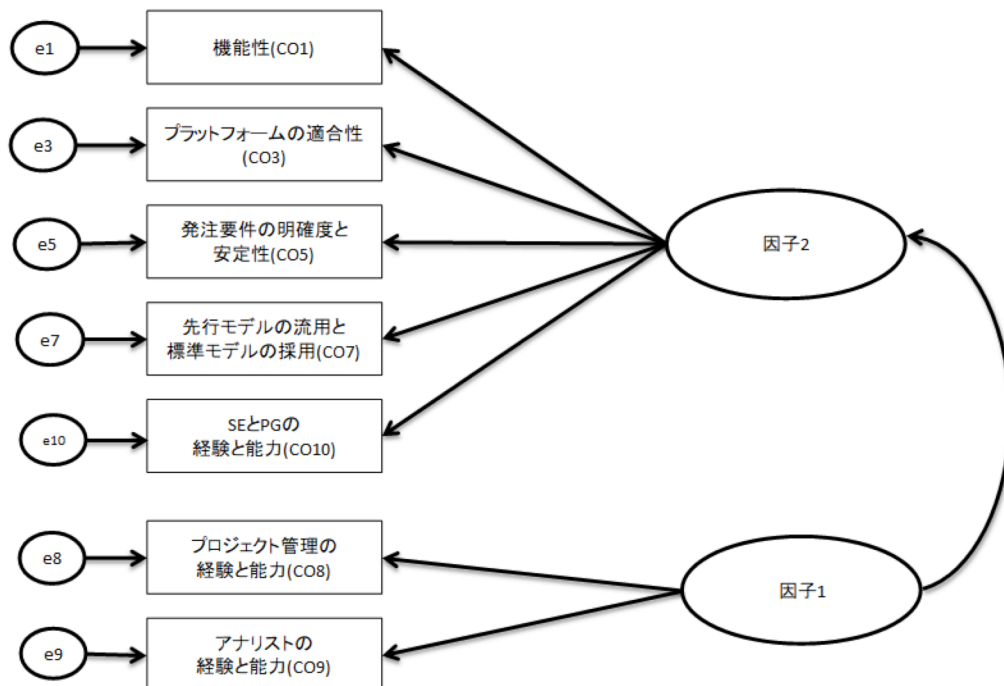


図 4-7 モデル 4 パス図

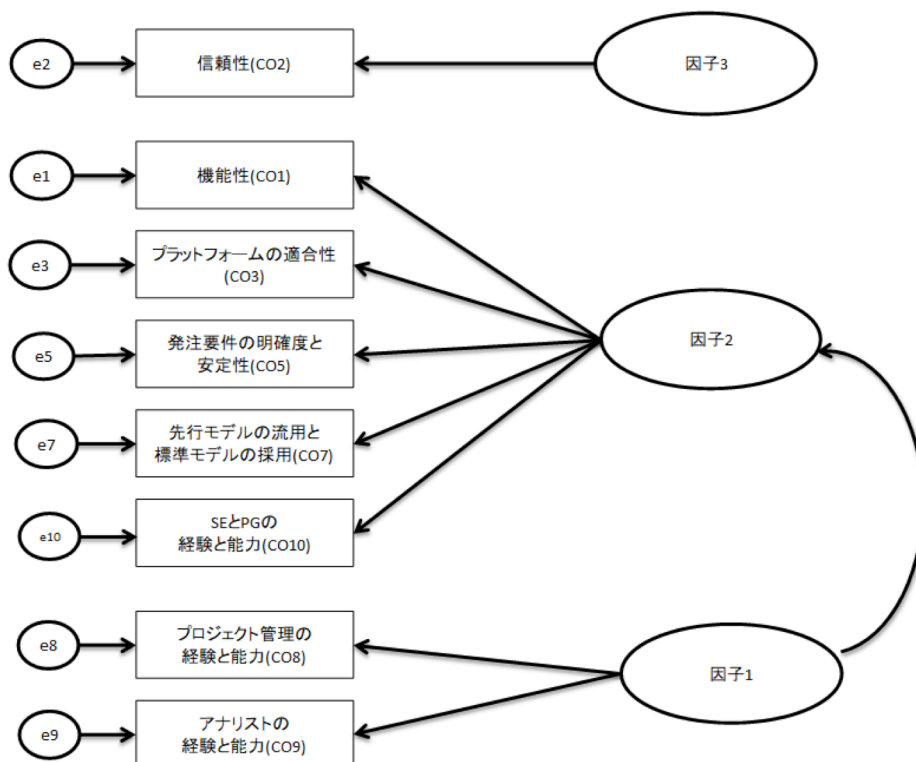


図 4-8 モデル 5 パス図

4.6.2 母数推定

上記の仮説モデルより分析を実施した。統計解析ツールはRを使い、パッケージは、lavaanパッケージにおけるsem関数を使用し共分散構造分析を実施した。実施したモデル1の結果を以下に示す。

◆モデル1の実行結果

lavaan (0.5-23.1097) converged normally after 40 iterations

Number of observations	1729
Estimator	ML
Minimum Function Test Statistic	42.170
Degrees of freedom	8
P-value (Chi-square)	0.000

Model test baseline model:

Minimum Function Test Statistic	1292.977
Degrees of freedom	15
P-value	0.000

User model versus baseline model:

Comparative Fit Index (CFI)	0.973
Tucker-Lewis Index (TLI)	0.950

Loglikelihood and Information Criteria:

Loglikelihood user model (H0)	-13232.417
Loglikelihood unrestricted model (H1)	-13211.333
Number of free parameters	13
Akaike (AIC)	26490.835
Bayesian (BIC)	26561.754
Sample-size adjusted Bayesian (BIC)	26520.454

Root Mean Square Error of Approximation:

RMSEA	0.050
90 Percent Confidence Interval	0.036 0.065
P-value RMSEA \leq 0.05	0.482

Standardized Root Mean Square Residual:

SRMR	0.031
------	-------

Parameter Estimates:

Information	Expected
Standard Errors	Standard

Latent Variables:

	Estimate	Std.Err	z-value	P(> z)
f1 =~				
CO8	1.000			
CO9	1.349	0.196	6.865	0.000
f2 =~				
CO1	1.000			
CO3	0.852	0.076	11.247	0.000
CO5	1.264	0.103	12.307	0.000
CO7	1.275	0.108	11.825	0.000

Regressions:

	Estimate	Std.Err	z-value	P(> z)
f2 ~				
f1	0.194	0.029	6.722	0.000

Variances:

	Estimate	Std.Err	z-value	P(> z)
.CO8	0.451	0.057	7.919	0.000
.CO9	0.420	0.101	4.167	0.000
.CO1	0.517	0.023	22.663	0.000
.CO3	0.514	0.021	24.560	0.000
.CO5	0.470	0.027	17.642	0.000
.CO7	0.883	0.038	23.000	0.000
f1	0.382	0.060	6.391	0.000
.f2	0.164	0.020	8.209	0.000

モデル 2 からモデル 5 も同様に実施し、それぞれの実行結果からモデルの適合度の評価となる値を抽出した結果を表 4-10 に示す。

表 4-10 適合度評価

指標	モデル 1	モデル 2	モデル 3	モデル 4	モデル 5
RMSEA	0.050	0.185	0.076	0.063	0.093
CFI	0.973	0.585	0.914	0.940	0.833
TLI	0.950	0.309	0.862	0.904	0.755
SRMR	0.031	0.103	0.059	0.044	0.072
AIC	26490	26985	31557	31518	36056
BIC	26561	27051	31639	31600	36149

4.6.3 モデル適合度の評価

モデルの適合度を評価するために、以下の適合度指標を元に、表4-10の結果から適合度評価を行った。

<モデルの適合度指標>

◆Comparative Fit Index(CFI)

CFI は、比較適合度指標と言われ、その値の範囲が 0 から 1 までの範囲に収まるように定義されている。自由度を考慮した上で分析モデルの乖離が独立モデルの乖離からどれくらい減少したのかを示す指標であり、1 に近いほど適合度がよい。

$$CFI = 1 - \frac{\chi^2_{\text{分析モデル}} - df_{\text{分析モデル}}}{\chi^2_{\text{独立モデル}} - df_{\text{独立モデル}}} \quad (\text{式 4.33})$$

χ^2 値と自由度は以下となる。

$$\chi^2_{\text{独立モデル}} = \text{tr}((\text{diag}(S))^{-1}S) - \log |(\text{diag}(S))^{-1}S| - p$$

$$\begin{aligned}
&= p - \log \left| (\text{diag}(S))^{-1} S \right| - p \\
&= -\log \left| (\text{diag}(S))^{-1} S \right|
\end{aligned}$$

$$\therefore df_{\text{独立モデル}} = \frac{1}{2}p(p-1) \quad (\text{式 4.34})$$

◆ Tucker-Lewis Index(TLI)または NonNormed Fit Index(NNFI)

TLI に関しても同様に 1 に近い方が適合度良いモデルである.

$$TLI = \frac{(T_b/df_b) - (T_h/df_h)}{\left(\frac{T_b}{df_b}\right) - 1} \quad (\text{式 4.35})$$

但し T_b はベースラインモデルでの, T_h は関心のモデルでの T の値である. df_h は制約のあるモデルと飽和モデルの母数の数の差. つまり関心のあるモデルでの自由度であり, df_b はベースラインモデルと飽和モデルの母数の数の差となり, つまりベースラインモデルでの自由度となる. 尚, ベースラインモデルとは, 各観測変数が独立であること前提にモデルを立て, それとの乖離を見る指標である.

◆ Root Mean Square Residual(RMR)

RMRは, モデルによって説明できなかった標本共分散(相関)行列の大きさを評価している.

$$RMR = \sqrt{\frac{2}{p(p-1)} \Sigma \Sigma (S_{jl} - \hat{\sigma}_{jl})^2} \quad (\text{式 4.36})$$

モデルで説明できなかった標本共分散行列の残りの部分を 1 要素あたりの平均として算出している. つまり平均的な残差共分散行列の値である. データとモデルが完全に一致していれば差は 0 となるため, 0 に近いほど適合度が良いことになる. 観測変数のモデルには影響しないが, モデルの自由度に影響を受け, 自由度の数が増えると RMR は小さくなる性質がある. また, 生データの各観測変数の分散の違いによって, 単位の影響により値が大きく変動する.

◆Standardized Root Mean Square Residual(SRMR)

SRMR は、RMR を標準化した指標である。標準化残差平方平均であり、標本分散・共分散とモデルにより再現される分散・共分散の差である残差によってモデルの適合を検討する指標である。下限値は 0 であり、0 に近いほど適合していることになる。

$$SRMR = \sqrt{\frac{2}{p(p-1)} \sum \sum \left(\frac{S_{jl}}{\sqrt{S_{jj}S_{ll}}} - \frac{\hat{\sigma}_{jl}}{\sqrt{\hat{\sigma}_{jj}\hat{\sigma}_{ll}}} \right)^2} \quad (\text{式 4-37})$$

S_{jl} と $\hat{\sigma}_{jl}$ はそれぞれ $\hat{\Sigma}$ の j 行 j 列の対角要素である。つまり、j 番目の観測変数に関する分散であり、 S_{ll} と $\hat{\sigma}_{ll}$ も l 番目の観測変数について同様である。

RSMEAについてはモデル1以外は全て5%を超えた。RSMEAは、式4.14より0.5以下はよいモデルのため、モデル1のみが良いモデルとなった。CFIについては、モデル1がCFI=0.973、モデル3がCFI=0.914、モデル4のCFI=0.94の3つのモデルが0.9以上となったが、この中で比較してもモデル1が最適していることが分かった。TLIについては、モデル1がTLI=0.95、モデル4がTLI=0.904となり、2つのモデル共に0.9以上となったが、これもこの2つと比較してもモデル1の方が良いことが分かる。そしてSRMRについては、モデル1がSRMR=0.031、モデル4が0.044と2つのモデルは、共に0.05以下となったが、モデル4よりもモデル1の方が数値を下回った。最後にAICとBICより、モデル1はAIC=26490、BIC=26561、モデル2はAIC=26985、BIC=27051と比較すると似通っているが、モデル1の方が、値が低いため、モデル1の方がよいモデルと判断することが出来る。したがって、今回の5つのモデルの中で一番適合度が良いモデルはモデル1であることが分かった。

次にこのモデル1から係数の推定結果を確認した。その結果が表4-11となる。推定値では、因子負荷量は0.44以上と高い値が示された。標準誤差は全体的に低いため、推定値の散らばりが少ない。よって精度よく推定が出来ていると考えられる。また、P値は0.0となっているため、「母数はゼロである」という帰無仮説が棄却され、「母数はゼロでない」の対立仮説が採択されることになる。

表 4-11 係数に関する推定結果

モデル	推定値	標準誤差	Z 値	P 値
f1=^ C01	0.506	0.027	18.497	0.00
f1=^ C03	0.444	0.028	15.974	0.00
f1=^ C05	0.618	0.028	22.432	0.00
f1=^ C07	0.498	0.027	18.163	0.00
f2=^C08	0.677	0.051	13.290	0.00
f2=^C09	0.792	0.058	13.585	0.00
f1^~f2	0.279	0.036	7.8410	0.00

ところが、観測対象数が大きい場合、母数の値が0に近くても帰無仮説が棄却されやすくなることが考えられる。そこで母数の値が0に近似しているかどうかを確認する為、信頼区間を検証した。母数に関する95%の信頼区間は標準誤差を利用し、式4.38で示すことができる。そこで、因子間の係数は表4-11の推定値と標準誤差を式4.38へ代入した結果の信頼区間を表4-12に示す。この結果から、全ての母数は0に近似していないことが分かった。つまり表4-11の推定値はすべて各因子に影響していると考えられる。

$$\text{推定値} - 1.96 \times \text{標準誤差} \leq \text{母数の真値} \leq \text{推定値} + 1.96 \times \text{標準誤差} \quad (\text{式4.38})$$

表 4-12 信頼区間一覧

	下限	上限
<i>f1 = ~C01</i>	0.45308	0.55892
<i>f1 = ~C03</i>	0.38912	0.49888
<i>f1 = ~C05</i>	0.56312	0.67288
<i>f1 = ~C07</i>	0.44508	0.55092
<i>f1 = ~C08</i>	0.57704	0.77696
<i>f1 = ~C09</i>	0.67832	0.90568
<i>f1 = ~f2</i>	0.20844	0.34956

4.6.4 パス図の描画

抽出した生産性変動要因の因子負荷量を標準化した後に、SEMにて分析したパス図が図4-9である。図4-9の数値は標準化推定した値である。開発の容易性については発注要件の明確度と安定性が62%，機能性が51%と、この2因子の関連性が強いことが分かった。上流工程の管理能力については、プロジェクトの管理能力が68%，アナリストの経験能力が80%と非常に関連性が強いことが分かった。

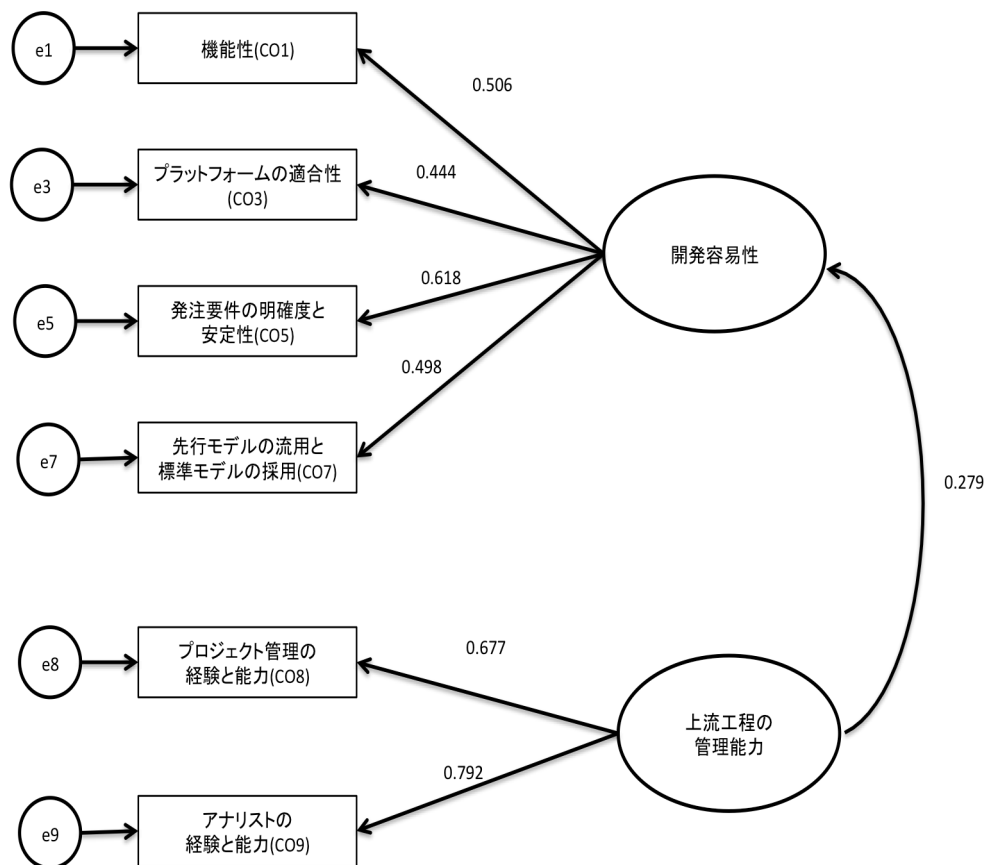


図 4-9 変動要因に関するパス図(標準化推定値)

4.6.5 構成概念スコア

構成概念スコアとは、研究仮説で仮定した構成概念の尺度上で各観測対象がどの程度その特性を持っているかを表した値のことである[Toyoda 14]。生産性変動要因データでは、「上流工程の管理能力」をどの程度を重要とし「開発の

容易性」にどれだけ影響があるのかを示した値となる。つまり因子分析における因子スコアに相当し、2つの因子の値を推定することになる。

生産性変動要因データから構成概念スコアを算出し「開発の容易性」と「上流工程の管理能力」のプロットを図4-10に示す。図4-10より、グラフの傾きが右肩上がりにプロットされており、開発がしやすい環境では上流工程の管理能力が高い傾向があるということが分かった。これは、優れている上流工程の管理者が上流工程でしっかりと取り決めすることが考えられ、プロジェクト全体の開発の容易さに導いていることが考えられる。また、逆に上流工程の管理能力が低い管理者は下流工程になるにつれ、開発の難易度が上がり、結果的に当初のソフトウェア開発見積もりよりも工数がオーバーしていることが考えられる。

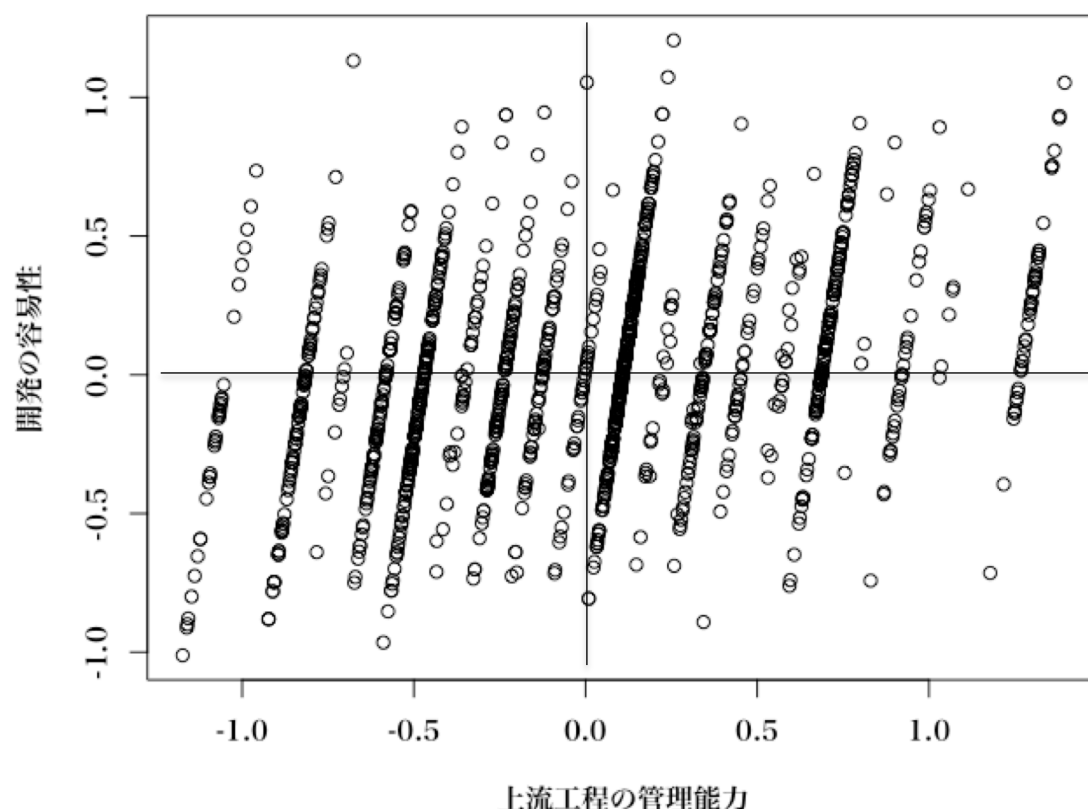


図 4-10 構成概念スコア

4.7 因子分析による考察

探索的因子分析と確認的因子分析の結果から、適合度のよいモデルを抽出し、因子と変動要因の関係性について明確にした。その関係性は表4-13で示す通りである。因子1の「上流工程の管理能力」因子は「アナリストの経験と能力」と「プロジェクト管理の経験と能力」の2つの観測変数割り当てられ、因子2の「開発の容易性」因子は「機能性」、「プラットフォームの適合性」、「発注要件の明確度と安定度」、「先行モデルの流用と標準化のモデルの採用」の4つの観測変数が割り当てられた。そして、プロジェクトの上流工程では、経験や能力の高いアナリストはそのままプロジェクト管理者になることが考えられ、最終的に上流工程の管理能力が高い技術者がソフトウェア開発の見積もりの難易度を決めることになる。また、因子間の関係性から、上流工程の管理能力が高い技術者であるほど開発の容易性が良くなると考えられる。これらは、プロジェクトの要件定義の局面でソフトウェア開発を見積もる際に、上流工程に携わる技術者のスキルによってプロジェクトに対するリスクの指標にもなり、ソフトウェア開発見積もりの重みをつける必要があると考えられる。

以上のことから今後、ソフトウェア開発ベンダはいかに早い段階で上流工程の管理能力が高いプロジェクトマネージャやプロジェクトアナリストを投入しプロジェクトに参加させることでソフトウェア開発見積もりの精度もより上がることになる。つまり、上流工程の管理能力が高いメンバがいるかどうかでプロジェクト全体が成功するかどうかが決まると考えられる。

表 4-13 SEM から導出した因子と観測変数の割り当て

因子	生産性変動要因
上流工程管理能力 (因子 1)	プロジェクト管理の経験と能力 (C08)
	アナリストの経験と能力 (C09)
開発の容易性 (因子 2)	機能性 (C01)
	プラットフォームの適合性 (C03)
	発注要件の明確度と安定度 (C05)
	先行モデルの流用と標準モデルの採用 (C07)

4.8 結言

本章では、相関行列から探索的因子分析を実施し、スクリープロットから因子数 2 が推定された。また、ガットマン基準からは、因子数が 3 と推測された。ただしこれら因子数はあくまでも目安であるため、それら因子数の ± 1 の因子数の検証を行った。つまり因子数が 1 から 4 までの探索的因子分析を実施した。ところが、実施した結果から因子数が 3 以上の解は不適解が抽出された。その為、因子数 1, 2 での因子分析を実施し、モデルの適合度評価から 2 因子が抽出された。これら因子は因子負荷量から分類し、因子 1 を「上流工程の管理能力」、因子 2 を「開発の容易性」と因子の命名とした。ところが、このモデルが本当に適正なのか。他に適正なモデルがないかを確認するため、確認的因子分析を実施した。確認的因子分析では探索的因子分析から抽出したモデルの他に生産性変動要因と因子との関係で考えられる 4 つのモデルを作成した。それら 5 つのモデルから測定方程式、構造方程式を作成し、共分散構造分析を実施した。その結果から探索的因子分析と同様に適合度指標を用いて、全体評価を実施した。適合度のよいモデルは探索的因子分析と同じ 2 つの因子と 6 つの観測変数のモデルが抽出された。

次にその抽出したモデルから深掘り分析するために、部分的評価を行った。まずは標準誤差からは、誤差が小さいことから、モデルとしては精度がよい推定であることが分かった。また、母数が 0 の可能性があるため、母数検定と信頼区間について確認した。その結果、観測変数はすべて因子に影響があることが分かった。そして最後に構成概念スコアから因子間の関係性を見出した。それは開発がしやすい環境では上流工程の管理能力が高い傾向にあるということが分かった。つまり優れている上流工程の管理者が上流工程でしっかりと取り決めることでプロジェクト全体の開発の容易さに繋がっていると考えられる。

そこで、今回の抽出した因子モデルから、3 章で定義した 9 業種のそれぞれで比較ができないかどうか考えた。今回と同様に 9 業種それぞれで 2 因子モデルを比較した場合、因子負荷や誤差分散などの母数が異なる値として推定されることになる。また、全ての業種で母集団が同一の構成概念を測定できている

とは限らない．もし母集団間で因子の意味が異なっている場合，推定された因子平均を比較することは意味がないことになる．したがって母集団間で因子負荷に等値制約を置くことで，はじめて因子平均の比較が可能となる．つまり多母集団同時分析を行うことで，因子平均が比較可能となる．これら状況を鑑みて，次章では本章で導出した 2 因子モデルを使い，9 業種で因子や因子負荷に統治制約をおき，多母集団同時分析から業種間での相違性の検証を行う．

第5章

ソフトウェア開発見積もりに影響を与える因子モデルを利用した業種別相違性

前章では生産性変動要因から見積もりに影響を与える因子を抽出し、LOC 法や FP 法では考慮されていない生産性変動要因から潜在的な因子を見つけ出し、2 つの因子を抽出した。この抽出した 1 つ目の因子は開発の容易性、2 つ目の因子は上流工程の管理能力である。ところが前章では 9 業種を含めた単一母集団での分析をしており、各業種での多母集団間での比較は実施していない。単一母集団でのパラメータは多母集団間で必ずしも一致するとは限らない。つまり 2 因子を抽出したモデルには複数の母集団のパラメータが異なるという仮説を検証する必要がある、業種別に分析する方法が考えられる。つまり業種別にそれぞれ別々で分析した場合、全てのパラメータが異なるという仮説をおくことになる。多母集団を同時に分析を行うことは全てのパラメータが同じである仮説を置いていることになるため、母集団間のパラメータを比較する場合、比較の枠組みを共通化させた上で比較したい仮説を検証する必要がある。

5.1 業種別によるソフトウェア見積もり因子の比較

4章では抽出した2つの因子と6つの観測変数となる生産性変動要因の関係モデル定義した。これらを条件として業種が違う母集団から多母集団同時分析を実施し、業種間の相違性を検証する。

5.2 対象データ

対象とした業種とそれらサンプルサイズとその相対度数を表 5-1 に示す。データが多い順に製造業が 410 件，流通業が 316 件，金融業が 236 件となり，データが少ない業種は，電気・ガス・熱供・水供給が 96 件，建設業が 49 件となっている。

表 5-1 業種ごとによるデータのサンプルサイズと相対度数

業種	サンプルサイズ	相対度数
製造業	410	24%
流通業	316	18%
金融行・保険業	236	14%
公共	206	12%
サービス業	184	11%
情報通信業	126	7%
その他・不動産業	98	6%
電気・ガス・熱供給・水道業	96	6%
建設業	49	3%

5.3 評価実験

5.3.1 多母集団同時分析

多母集団とは、互いの構造を比較したい母集団が複数個あること示す。複数個の母集団で同一の因子が想定できる時、因子が不変である、もしくは因子不変性が成り立つという。そして多母集団同時分析は各集団間で母数に制約条件を課した上でモデル適用度を確認することで、集団間で等しいと言えるのか、異なると言えるのかということを検討する分析方法である。4章の単一母集団での分析ではCFA(Confirmatory factor analysis)から抽出した2つの因子と6つの観測変数を抽出した。このモデル構造を使い、本章では、9業種(9母集団)で多母集団同時分析を実施する。多母集団同時分析ではモデルの構造として、基本的に以下の5つのモデルを比較し、適合が許容範囲になるもっとも制約条件の厳しいモデルを採用することになる。

<制約条件>

〈5・1〉配置不変モデル（適合度：model1）

配置不変モデルとは、複数の母集団で因子構造が同じモデルであることを示す。すなわち想定する母集団間で、同一の構成概念が仮定できるということを意味している。

〈5・2〉弱測定不変モデル（適合度：model2）

弱測定不変モデルとは、複数の母集団間で因子負荷が同じになるように等値制約を加えたモデルのことである。因子の性質が異なりすぎると、因子平均を比較することの意味がなくなってしまうため、因子負荷の制約が成り立つことは多母集団同時分析において、因子平均を比較する際に最低限満たしておく必要のある制約である。

〈5・3〉 強測定不変モデル（適合度：model3）

強測定観不変モデルとは、観測変数の標本平均の違いを因子平均の違いとして表現したい場合、因子負荷だけでなく、切片も母集団間で等しくする必要がある。このような等値制約を置いたモデルのことを多母集団同時分析での強測定不変モデルと呼ぶ。

〈5・4〉 厳密な測定不変モデル（適合度：model4）

厳密な測定不変モデルとは、強測定不変モデルの等値制約に加え、観測変数の誤差分散も母集団で等しいと考えるモデルである。

〈5・5〉 全母数が等しいモデル（適合度：model5）

因子の分散を除いたすべての母数が母集団間で等しいモデルである。このモデルでは、厳密な測定不変モデルの制約に加え、因子平均の値も母集団間で等しい制約を課すことになる。

上記の配置不変モデルや測定不変モデルを検討するため、共分散構造分析から導出したモデルを使い、以下のモデル式を統計解析ツール R の `lavvan` パッケージによる、`measurementInvariance` 関数を使い分析した。その結果を表 5-2 に示す。モデルの適合度指標は、4 章と同様に、AIC, BIC, CFI, RSMEA での適合度指標を使用した。これら指標に基づき、どのモデルが最も望ましいか評価した。表 5-2 の結果から、CFI が 0.911, RFMEA が 0.056, AIC と BIC が最小となる制約条件が最も厳しい `fit.means(model5)` を選択した。

この選択した `model5` を元に、因子負荷量、観測変数の切片、観測変数の残差、潜在変数の平均をグループにし、等値制約したモデルを作成した。統計解析ツールは R の `sem` パッケージを使い、多母集団同時分析を実施した。その結果を表 5-3 に示す。また、この実行した際のパス図について図 5-1 から図 5-9 に示す。

これら因子 1 の分散，因子 2 の分散，相関係数を総合的に俯瞰した場合，いくつかのグループに分類出来ないかを考えた．そこで検証方法として，因子 1 の分散，因子 2 の分散，相関係数のデータを標準化した後にクラスター分析を実施した．

表 5-2 多母集団同時分析の評価結果

Goodness-of fit	DF	AIC	BIC	Chisq diff	CFI	RMSEA
fit.configural (model1)	72	26446	27378		0.946	0.073
fit.loadings (model2)	104	26444	27201	61.621	0.924	0.072
fit.intercepts (model3)	136	26420	27004	40.974	0.917	0.065
fit.residuals (model4)	184	26382	26703	57.242	0.910	0.059
fit.means (model5)	200	26364	26598	14.316	0.911	0.056

表 5-3 因子の分散と因子間の相関係数

No	業種	因子 1 分散値	因子 2 分散値	因子間相関係数
1	金融業・保険業	0.35	0.12	0.33
2	情報通信業	0.35	0.20	0.30
3	公務	0.47	0.17	0.32
4	サービス業	0.38	0.24	0.23
5	その他	0.43	0.23	0.05
6	電気・ガス・熱供給・水道業	0.44	0.15	0.16
7	流通業	0.31	0.19	0.42
8	建設業	0.25	0.07	0.17
9	製造業	0.29	0.16	0.70

1

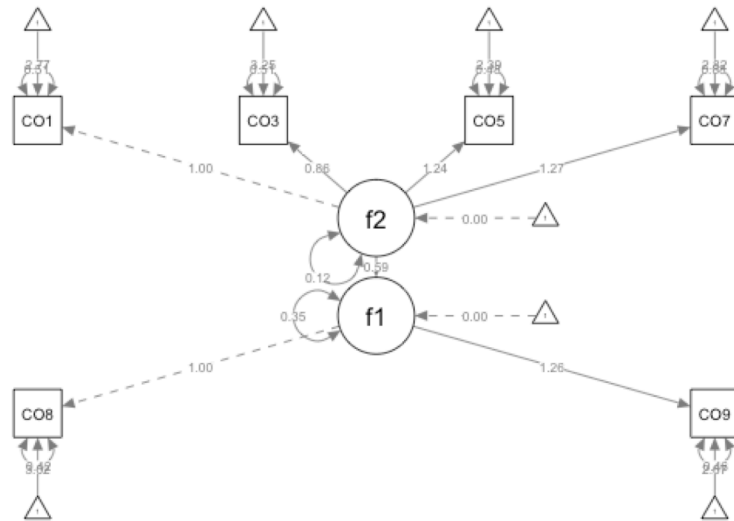


図 5-1 金融業・保険業のパス図

2

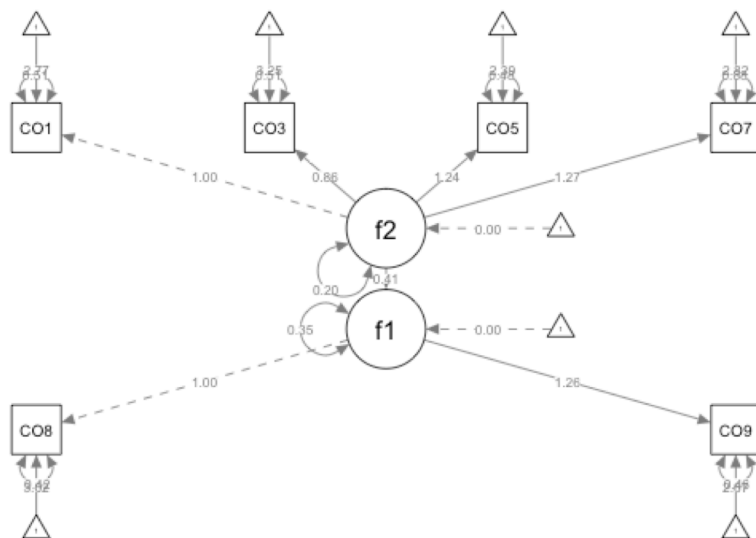


図 5-2 情報通信業のパス図

3

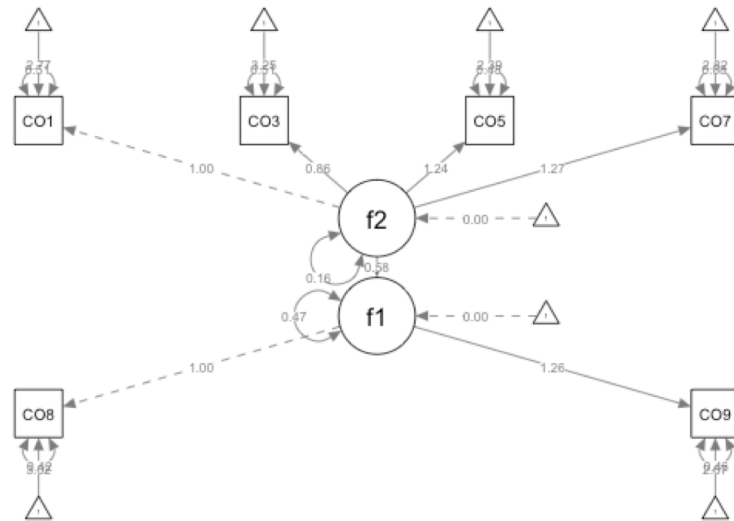


図 5-3 公務のパス図

4

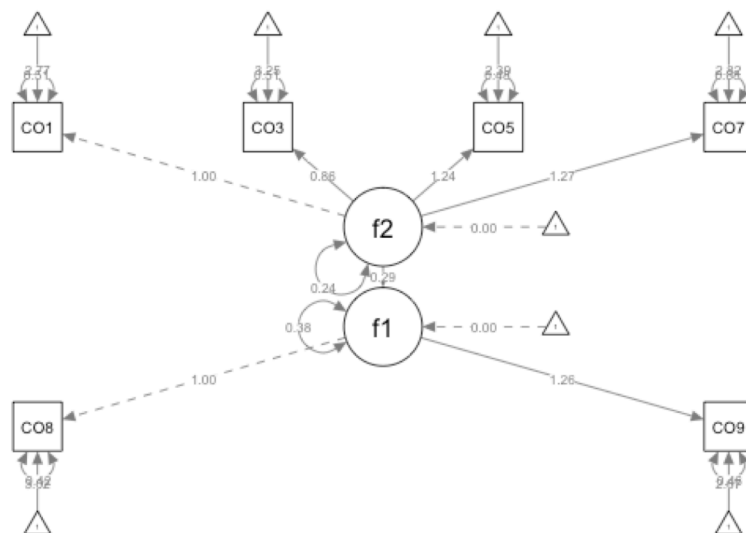


図 5-4 サービス業のパス図

5

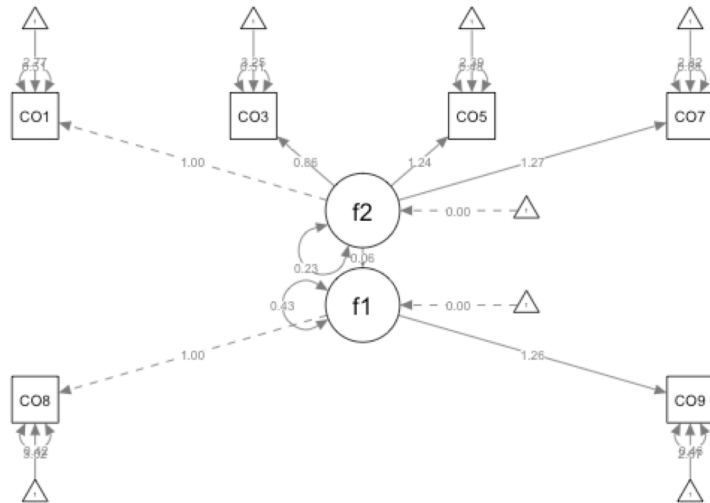


図 5-5 その他業種のパス図

6

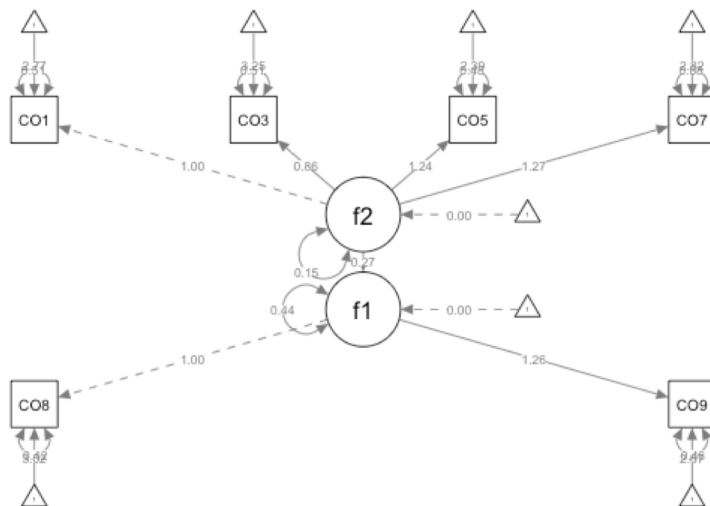


図 5-6 電気・ガス・熱供給・水道業のパス図

7

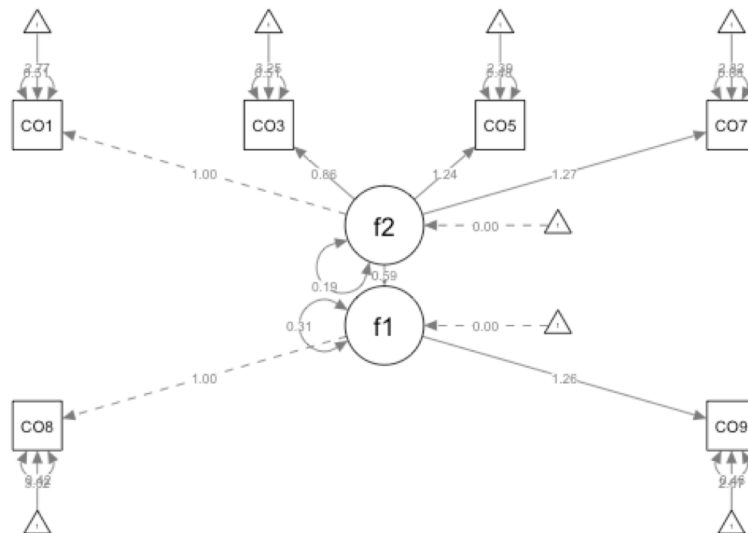


図 5-7 流通業のパス図

8

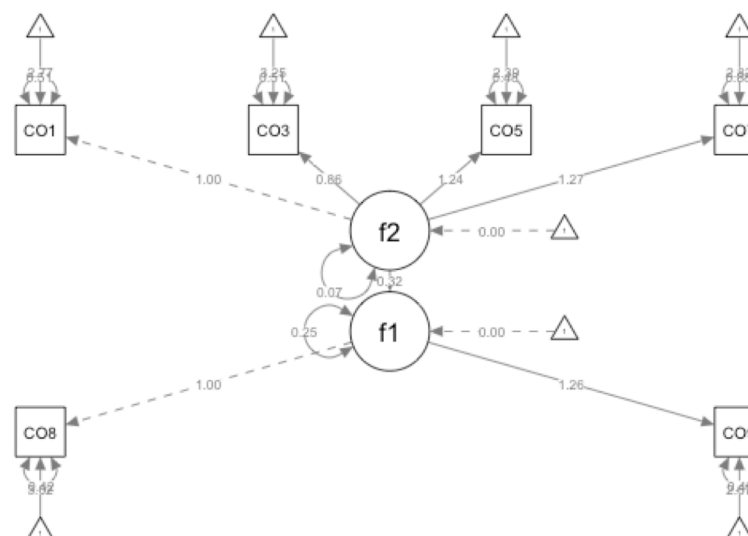


図 5-8 建設業のパス図

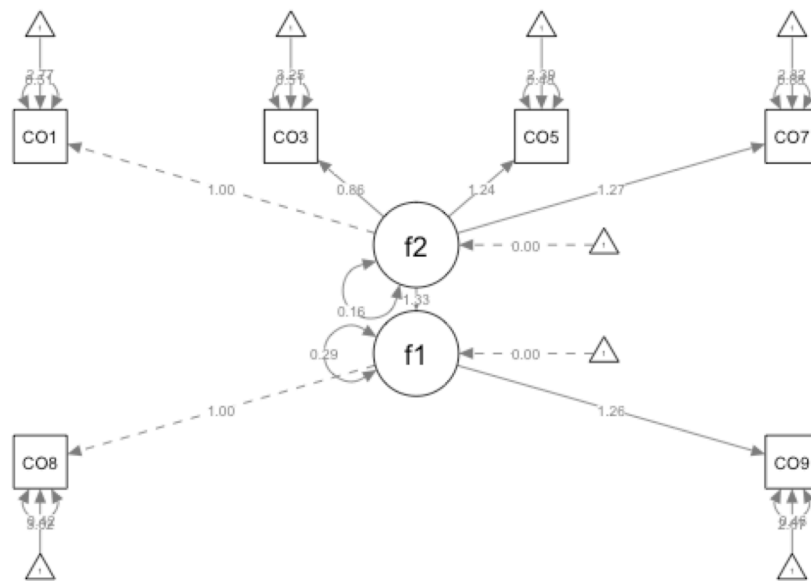


図 5-9 製造業のパス図

5.3.2 クラスタ分析

クラスタ分析とは異なる性質のものが混ざりあっている集団の中から互いに似たものを集めてクラスタを作り、対象を分類する方法である。クラスタ分析には階層クラスタ分析があり、個体間の類似度あるいは非類似度（距離）に基づいて、最も似ている個体から順次集めてクラスタを作っていく方法である。クラスタ分析の解析は以下の手順で実施する。

- ① それぞれの距離の近さを測るための距離、およびクラスタを併合する際の距離あるいは類似度を求める。
- ② それぞれの対象間の距離をすべて計算し、距離が最小となる対象を統合して最初のクラスタとする。
- ③ 新しく形成されたクラスタと対象間の距離をすべて計算し、対象間の距離を含めて最小のものを統合する。
- ④ これをすべてのクラスタで統合されるまで繰り返す。

変数が p 個の場合、 $(x_{i1}, x_{i2}, \dots, x_{ip})$, $(x_{j1}, x_{j2}, \dots, x_{jp})$ のユークリッド距離は以下の式 5.1 で表すことが出来る。

$$d_{ij} = \sqrt{\sum_{k=1}^p (x_{ik} - x_{jk})^2} \quad (\text{式 5.1})$$

そしてクラスタ間の距離にはいくつかの方法がある。以下、よく利用されるクラスタ間の方法について以下 a) から e) の 5 つ方法について述べる。

a) 最短距離法

2 つのクラスタに属する対象のうち、最も近い対象間の距離をクラスタ間の距離とする方法である。

b) 最長距離法

2 つのクラスタに属する対象のうち、最も遠い対象間の距離をクラスタ距離とする方法である。

c) 群平均法

2 つのクラスタに属する対象間のすべての組み合わせの距離を求め、

その平均値をクラスタ間の距離とする方法である.

d) 重心法

各クラスタの代表点を重心とし, 重心間の距離をクラスタ間の距離とする方法である.

e) ウォード法

新たに統合されるクラスタ内の平方和を最も小さくするという基準でクラスタを形成していく方法である. ウォード法はグループ内の分散が小さく, 且つグループ間の分散が大きい組み合わせでグループ分けを行う.

以下, 式 5.2 から式 5.5 はウォード法の平方和と増分を表した式である.

$$S_l = \sum_{i=1}^{n_l} \sum_{k=1}^p (x_{lik} - \bar{x}_{l \cdot k}) \quad (\text{式 5.2})$$

$$S_m = \sum_{i=1}^{n_m} \sum_{k=1}^p (x_{mik} - \bar{x}_{m \cdot k}) \quad (\text{式 5.3})$$

$$S_{lm} = S_l + S_m + \Delta S_{lm} \quad (\text{式 5.4})$$

$$\Delta S_{lm} = \frac{n_l n_m}{n_l + n_m} \sum_{k=1}^p (\bar{x}_{l \cdot k} - \bar{x}_{m \cdot k})^2 \quad (\text{式 5.5})$$

式 5.2 に従い, 距離を計算し, 次に統合した時のクラスター内平方和の増加分が最小のものを統合していく.

これらの方法からクラスタの統合過程を示すデンドログラムを描き, 適当な距離で切断されることによりいくつかのグループに分ける. そして各グループに含まれる対象を調べ, グループの特徴を把握する.

本実験では, 距離の指定にウォード法を指定しクラスタ分析を実施した. 尚, 実験で利用した統計解析ツールは R を使い, hclust 関数を使用し分析した. 実施した結果のデンドログラムを図 5-10 に示す. デンドログラムでは下方で結合

すればするほど、近い関係である．図 5-10 の場合，大きく 2 つのクラスタリングに別れた．

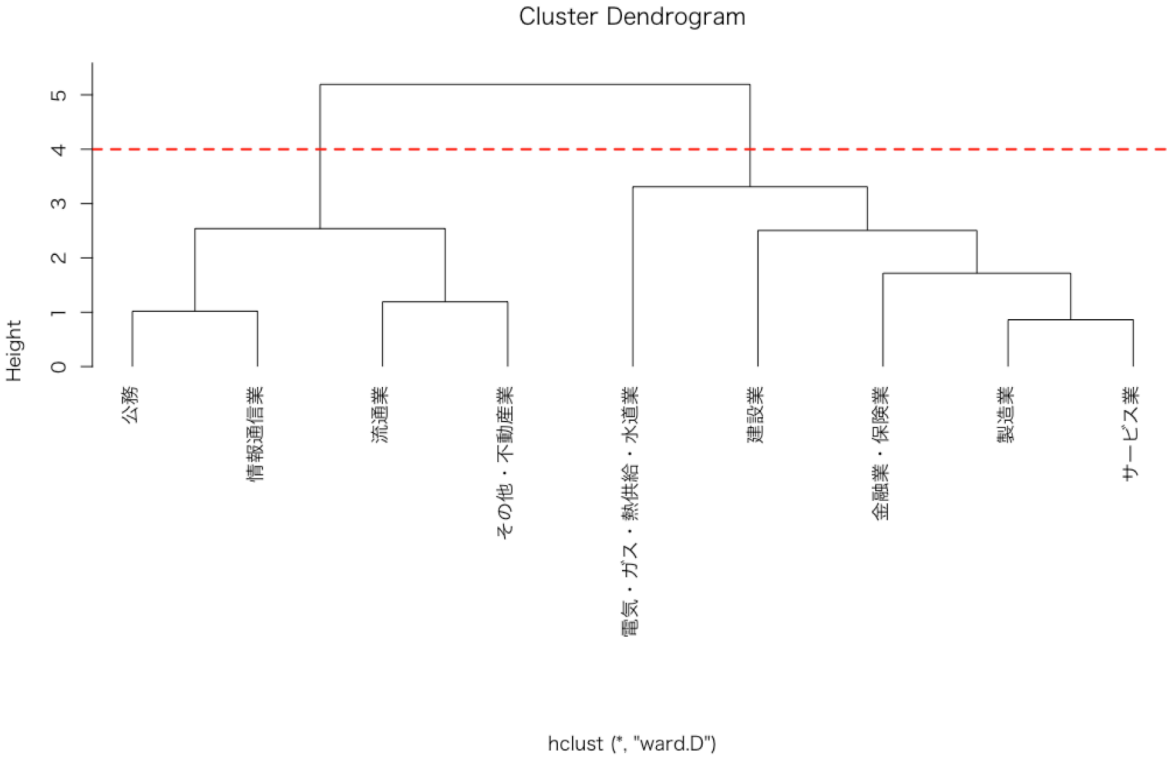


図 5-10 業種別デンドログラム

5.4 実験結果の考察

表 5-3 より，因子 1 の分散値は建設業が 0.25，製造業が 0.29，流通業が 0.31 となり，これら 3 業種は，その業種以外とで比較した場合，全体的に低いことが分かった．つまりこの 3 業種については，それ以外の業種と比べ，開発の容易性が統一されていると考えられる．一方，公務や電気・ガス・熱供給・水道業の 2 業種では分散値が高い．つまりプロジェクトによって開発の容易性が統一されていないと考えられる．これら要因の一つとして，公共機関のプロジェクトの場合，規模が大きいことで，複雑性が高いことが考えられる．

因子 2 の分散値全体では，因子 1 の分散値全体と比較し低いことが分かった．これはどの業種も上流工程の管理能力について，開発前の上流工程の段階でプロジェクト管理の経験と管理能力やアナリストの経験能力に差がないと考えられる．その中でも特に，建設業と金融業・保険業の 2 業種については，分散値が他業種と比べそれぞれ 0.07，0.12 と極端に低いことが分かった．つまりこの 2 業種は上流工程の管理能力がプロジェクトによる差は少ないと考えられる．特に建設業は因子 1 の開発の容易性の分散も低いことから，上流工程でも下流工程でも統一性が高い業種であると考えられる．また，金融関係の業種については，法律に則ったシステム構築が必要となるため，専門性の高いプロジェクト管理経験や能力，アナリストの経験能力が定着していると考えられる．一方，サービス業や情報通信業，その他・不動産業では分散が高いため，プロジェクト間での上流工程の管理能力の高低差がある業種であると考えられる．

相関係数を業種別に見た場合，建設業では因子 1 と因子 2 の因子間での相関係数が低い．つまり建設業の場合，上流工程での管理能力が高いからと言って，必ずしも開発の容易性との関連性が高くなるとは言えないことになる．一方，製造業の場合は相関係数が 0.7 と非常に高く，上流工程の管理能力が高い場合，開発の容易性も高くなる業種と考えられる．つまり製造業の場合，上流工程の管理能力の高いアナリストやプロジェクト管理者を担当させることで，開発の容易性を高めることが可能となる．

最後に因子1，因子2，因子間相関係数を特徴量とし，総合的にどのように分類されるか確認するためにクラスタ分析を実施した．その結果が図5-10となった．結果的に，大きく2つのクラスタに分類された．金融業や電気・ガス・熱供給・水道業，建築業（以下グループ1）はBtoBの開発プロジェクト，流通業やその他・不動産（以下グループ2）ではBtoCの開発プロジェクトで分類出来ると考えた．また，グループ1では，サービス業と製造業が最も類似しており，そのクラスタ群と金融・保険業，建設業，電気・ガス・熱供給・水道業，建築業が順に類似している結果となった．グループ2では流通業とその他・不動産業（以下クラスター1），公務と情報通信業（以下クラスター2）の類似度が最も類似しており，続いて，クラスタ1群とクラスタ2群の類似していることが分かった．

ソフトウェア開発ベンダは多くの業種で見積もることがあり，過去の見積もりを参考に見積もることが多々ある．業種によっては過去に開発経験をしたことのない業種もあり，その場合，何を基準に見積もるのか．また，見積もる際は，なるべく多くの見積もりからそれに似通った見積もることもある．つまりこれらソフトウェア開発ベンダはソフトウェア見積もりをする際，類似している業種を参考に見積もりを行うことでより精度のよい見積もりができると考えられる．

5.5 結言

本章では、前章の確認的因子分析から抽出した2因子モデルを使い、業種別の相違性を分析した。まず5.2では本実験での説明をした。そして5.3では等値制約モデルからモデルの不変性の検証を行った。モデルの不変性では、5つのモデルを比較しモデルの指標結果から、最も当てはまりのよい全分母が等しいモデルを採用した。そして選択したモデルから、因子負荷量、観測変数の切片、観測変数の残差、潜在変数の平均をグループにし、多母集団同時分析を実施した。その結果、因子1と因子2の分散、因子間の相関係数を抽出した。そして、抽出した結果から業種間で比較し、業種ごとの特徴について抽出することが出来た。また、2つの因子と相関係数をクラスタリングし、大きく2つのグルーピングし、さらに距離の近い業種同士を抽出した。つまり業種間での相違性を示すことが出来た。ソフトウェア開発ベンダは決まった業種だけの見積もりを取ることは少なく、多くの業種の見積もりを任されることが多い。その場合、その業種の要因や特徴を意識することで見積もり精度を上げることができると考えられる。

第6章

結論

本論文では、経済調査会が大中企業から集めたアンケートデータを利用し、その中から見積もり工数、生産性変動要因を抽出し、見積もり工数に影響を与える生産性変動要因が何か。また業種ごとに影響を与える生産性変動要因は何かを見つけ出すため、統計手法を使いモデルを作成後、それぞれの作成したモデル影響を与える要因や因子について提案した。古くから研究されている見積もり技法では、変動要因から見積もりに影響がある変動要因は何かについて一般化されているものではなく、プロジェクト管理者は見積もりを作成する際に参考となる指標はなかった。そのため、多くの見積もり技法を使い、見積もりをした場合でも分散値が大きく、各企業がプロジェクトに合った見積もりをすることは難しかった。本研究ではこれらの問題を解決するために、見積もり工数や生産性変動要因を使い、業種全体もしくは業種ごとで影響する生産性変動要因を見つけ出すことが出来た。

本研究を行う動機は、筆者が15年以上ソフトウェア開発ベンダとして従事しており、日々、プロジェクト管理者としてソフトウェア見積もることを必要としている。ソフトウェア開発ベンダはRFPや要件定義をインプットとし、それを見積もり技法で見積もった結果をIT発注企業に提示してもなかなか理解してもらえず、お互いが納得しないままプロジェクトが開始されてしまうことがあった。それが原因でトラブルとなり、プロジェクトが途中で中断され、最悪の場合、裁判までにつながることもなりかねないことになる。その課題を少しでも解決できないものか。目に見えないソフトウェアの見積もりの誤差を少しでも縮めることができないか。そして、業種ごとで影響を与える要因があるのであればその要因は何か。そのための判断材料は何を用いるべきかなど問題を意識したことが大きい。

本論ではLOC法やFP法などの手法からは算出することができない問題点を解決するために、統計手法から見積もりに影響を与える要因を抽出する方法を考えた。しかしながら、生産性変動要因と見積もり工数から抽出しても、業種が異なればその影響する要因は異なってくるはずであると考えた。

そこで著者は先行研究を行い、その過程でソフトウェア見積もりに関して様々な解析や分析が行われ、数式モデルが作られてきており、COCOMO II 法やCoBRA 法では変動要因を変数パラメータとして計算する技法まで存在していた。ところがこれら技法でも、あくまでも生産性変動要因は補数としての扱いをしており、生産性変動要因のみの研究については、ほとんどされていないこと気づいた。また、企業では過去の見積もり工数についての情報を残す企業は多いが、筆者が所属しており、多くの企業が加盟している日本ファンクションポイントユーザ会(JFPUG: Japan Function Point Users Group)の各企業の担当者に対してヒアリングしても生産性変動要因の数値を企業に残している企業はほとんどいないとの結果を得ている。また、生産性変動要因がソフトウェア工数見積もりの大きな要因となっていることもヒアリング結果から得ている。このことから、生産性変動要因の多くが見積もり工数に影響を与えているということが本研究の核となった。そして見積もり工数と生産性変動要因との関係から見積もり工数に影響を与えている要因を抽出することが出来るのではないかという判断より、本研究に至った。

本論文の2章では、歴史の古い見積もり技法であるPMBOKで推奨されている類推法(トップダウン法)、ブレイクダウン法、統計手法そして、要因に関する研究について述べた。その中でもCOCOMO II 法やCoBRA 法では変動要因を利用して見積もり工数を算出する方法であった。そのうえで、第3章では、目的変数を見積もり工数、説明変数を生産性変動要因として、重回帰分析を実施し、見積もり工数に影響を与える生産性変動要因を抽出した。また、業種別でグルーピング後、同様に重回帰分析を実施し、業種ごとの見積もりモデルから、見積もり工数に影響がある生産性変動要因を抽出した。

第4章では、生産性変動要因を観測変数として、探索的因子分析を実施した。因子数を見つけるためにスクリープロットやカイザー・ガットマン基準から因子数が1から4を検証し、因子数が3以上のモデルでは不適会が出力され因子

数が1か2であることが分かった。そこで因子数1, 2それぞれの因子分析の実行結果から、適合評価を用いて最も良いモデルを抽出した。それが因子数2のモデルとなった。次にこの抽出した因子以外にも他に良いモデルがないかについて確認するため、構造化したモデルを作成し共分散構造分析を実施した。観測変数に影響を与える因子や因子間の関係性を考慮し、5つのモデルを定義した。その結果、探索的因子分析で抽出した因子が最も良いモデルとなった。この因子と因子負荷量から因子1は「上流工程での管理能力」、因子2は「開発のしやすさ」と因子の命名した。また、この2因子モデルのパラメータ検証を行い、2因子6観測変数であることを示した。

第5章では、4章から抽出した2因子の構造化モデルを使い、業種ごとによる相違性について検証した。はじめに、5つの制約条件をつけてそれぞれで多母集団同時分析を実行した。適合度指標より、制約条件の最も厳しい全母数が等しいモデルとなった。その抽出したモデルから、因子負荷量、観測変数の切片、観測変数の残差、潜在変数の平均をグループにし、多母集団同時分析を実施した。その結果、因子1, 2の分散値、因子相関を抽出した。因子の分散は小さい方がその因子に対する業種の中でのプロジェクトは一致していることになる。したがって、その業種は因子1が「建設業」、「製造業」、「流通業」であり、因子2が「建設業」「金融業・保険業」である。また、因子相関では因子相関が大きい場合、相関性が高くなるため、2つの因子に関係性が強いことになる。その業種が「製造業」となった。そして最後にこれら2つの因子と相関係数をクラスタ分析によりBtoBのグループ、BtoCでのグループ分類し、業種別の相違性についても述べた。

以上、本研究で提案する「ソフトウェア見積もり工数に影響を与える因子の研究」について報告した。

IT発注企業はサービスの拡大やより複雑なシステムを要求している中、ソフトウェア開発ベンダはIT発注企業の要望を実現するために、ソフトウェアの工数見積もりも複雑化している。したがってソフトウェア開発ベンダが蓄積してきたデータだけでは対応することは今後時代とともに一層難しくなってくる。そこで企業は自企業のみ蓄積してきた見積もりを参考にするのではなく、他の業種も含めて何が足りないのか把握する必要がある。

本論では、業種ごとの見積もりモデルや業種ごとの特徴を抽出した。したがってソフトウェア開発ベンダはこれらモデルや業種の特徴を把握することはとても有効である。ソフトウェア開発ベンダは見積もりをする上で、自ら、何が足りないのか把握し、それを IT 発注企業に伝えることで、お互いが納得のいく見積もりが出来ると考えている。

本研究ではエンタープライズでのウォーターフォール型モデルを中心に進めていった。今後はより幅を広げ、エンタープライズでのウォーターフォール型での開発のみだけでなく、アジャイル開発やスパイラル開発手法を使った見積もりのデータについても研究を進めていく。また、ソフトウェア業界のみでなく、他の業界での見積もりについても取り入れ、見積もり全体に対して発展させていく所存である。

謝辞

博士課程の在学中、公私にわたってたいへんお世話になった筑波大学大学院ビジネス科学研究科の津田 和彦 教授に深く感謝いたします。

副指導教官をお引き受け頂き、的確なアドバイスを頂戴した筑波大学大学院ビジネス科学研究科の吉田 健一 教授、倉橋 節也 教授に深く感謝いたします。

統計分析で悩んでいた時に、何度もお時間を割いて頂き、多くのアドバイスを頂いた椿 広計 筑波大学名誉教授に深く感謝しております。

発表会などで、様々なアドバイスやコメントを下された筑波大学大学院ビジネス科学研究科の教官の方々に、深く感謝いたします。

津田研究室の方々には、日頃より研究の進め方についての貴重な示唆やご意見を頂戴いたしました。また、ON と OFF のメリハリの効いたゼミのメンバと出会えたことは、私にとって貴重な財産となっております。メンバの皆様に対し、ここに深く感謝いたします。

ビジネス科学研究科の同期の方々は皆、とても優秀で刺激となりました。また、辛い時でも励まされ、今現在までやって来られたことに対し、ここに深く感謝いたします。

最後に、私が研究に没頭している中、なかなか遊びにも連れていけない中、いつも笑顔で素直に成長してくれた二人の娘たち、それから私が大学院に行こうか迷っていた時に次女が生まれ、そんな状況でも嫌な顔をせずに背中を押してくれ、毎日が子育てと仕事で慌ただしい中でも文句一つ言わず常に私を支えてくれた妻、志田 薫には本当に感謝します。皆の存在と励ましに深く感謝いたします。最大の感謝を述べ、この研究の成果を捧げ、深く感謝いたします。

お世話になりました皆さま、本当にありがとうございました。

参考文献

[Abdel 86]

Abdel-Hamid.T, and S. Madnick: “Impoact of Schedule Estimation on Software Project Behavior”, IEEE Software, 3,4, pp.70–75, (1986).

[Albrecht 79]

Albrecht Allan J: “Measuring Application Development Productivity”, Proceedings of the Joint SHARE/GUIDE/IBM Application Development Symposium, pp.83–92, (1979).

[Albrecht 83]

Albrecht Allan and J, Gaffney: “Software Function Source Lines of Code, and Development Effort Predition: A Software Science Validation”, IEEE Transactions on Software Engineering,SE–9(6) , pp.639–648, (1983).

[Armour 02]

Armour, Philip: “TEN Unmyths of Project Estimation”Communication of the ACM, pp.15–18, (2002).

[Baker 72]

Baker, F. Terry: “Chief Programmer Team Management of Production Programming”, IBM Systems Journal, vol11 no.1, pp.56–73, (1972).

[Bastani 87]

Bastani, Farokh and Sitharama Iyengar: “The Effect of Data Structures on the Logical COplexity of Programs”, Communications of the ACM vol.30, no.3, pp.250–259, (1987).

[Boehm 87]

Boehm, Barry W.: “Industrial software metrics top 10 list”, IEEE Software Engineering, vol.4, No9, pp84–85, (1987).

[Boehm 88]

Boehm, Barry W., and Philip N. Papaccio: “Understanding and Controlling Software Costs”, IEEE Transaction on Software Engineering, vol.14, No10, pp1462–1477, (1988).

[Boehm 95]

Boehm, Barry W.: “Software Engineering Economics Englewood Cliffs”, NJPrentice Hall, (1981).

[Boehm 00]

Boehm, B.W. et al.: “Software Cost Estimation with COCOMO II”, Center for Software Engineering, Englewood Cliffs, NJ: Prentice–Hall, (2000).

[Brooks 00]

Brooks, Jr Frederick P.: “人月の神話 – 狼人間を打つ銀の弾はない”, ピアソン・エデュケーション, (2002).

[Briand 98]

Briand, L. C., K. E. Emam and F. Bomarius: “COBRA: A Hybrid Method for Software Cost Estimation”, Benchmarking and Risk Assessment, Proceedings of the 20th International Conference on Softw Eng, pp. 390–399, (1998).

[Breiman 02]

Breiman, L.: “Random Forests Machine Learning”, vol45, pp. 5–32, (2002).

[Capers 98]

Capers Jones, "Estimating Software Costs", McGraw-Hill Companies, (1998).

[Card 87]

Card, David N.: "A Software Technology Evaluation Program", Information And Software Technology. V.29, No.6, (1987).

[Center 00]

Center for Software Engineering.: "COCOMO II Model Definition Manual The Center of Software ", USC, (2000).

[Cockburn 01]

Cockburn, Alistair.: "Agile Software Development", Boston, MA: Addison-Wesley, (2001).

[Conte 86]

Conste, S.D., H.E.Dunsmore, and V.Y.Shen, "Software Engineering Metrics and Models, Menlo Park", CA: Benjamin/Cummings, (1986).

[Curtis 86]

Curtis, Bill., et al.,: "Software Psychology, The Need for an Interdisciplinary Program," Proceedings of the IEEE, vol74, no.8, pp1092-1106, (1986).

[Cusumano 03]

Cusumano, Michael, and Richard W. Selby.: "Software Development Worldwide: The State of the Practice," IEEE Software, pp28-34, (2003).

[Ellis 95]

Ellis Horowitz, Chris Abts, A. Winsor Brown, Ray Madachy, Sunita Chulani, Don Reifer, Brad Clark, Bert Steece.: “COCOMO II Model Definition Manual The Center for Software”, Center for Software Engineering, (1995–2000).

[Fukawa 97]

布川薫他: “アプリケーション開発技術”, リックテレコム, (1997).

[Furuyana 07]

古山恒夫他: “ソフトウェア開発プロジェクトの遂行に影響を与える要因分析”, 情報処理学会論文誌 (2007).

[Garmus 12]

Garmus, David and David Herron.: “Function point Analysis: Measurement Practices for Successful Software Projects,”, Boston, Ma: Addison–Wesley, (2002).

[Gilb 88]

Gilb, Tom.: “Principles of Software Engineering Management”, Workingham, England: Addison–Wesley, (1988).

[Goldratt 97]

Goldratt, Eliyahu M.: “クリティカルチェーンー なぜプロジェクトは予定通りにすすまないのか?”, (1997).

[Gorla 90]

Gorla, N., A.C. Benander, and B. A. Benander.: “Debugging Effort Engineering”, vol16, no2 pp223–231, (1990).

[Grady 90]

Grady Robert B, and Deborah L.Caswell: “ソフトウェア・メロリクスー 生産性, 品質の計測方法からのその全社的展開まで”, 日経 BP, (1990).

[Hashimoto 05]

橋本真二, “オブジェクト指向ソフトウェアに対する機能量計測に関する研究”, 論文, (2005).

[Hatsuda 06]

初田賢司, “本当に使える見積もり技術”, 日経 SYSTEMS, (2006)

[Henry 84]

Henry, Sallie, and Dennis Kafura,: “The Evaluation of Software Systems Structure Using Quantitative Software Metrics”, Software-Practive and Experience, vol.14, no.6, pp561-573, (1984).

[Ito 16]

伊藤大幸,他, “一般小中学生における食行動異常の実態とメンタルヘルスおよび社会的不適応との関連”,教育心理学研究, pp170-183, (2016).

[IPA 05]

IPA 独立行政法人 情報処理推進機構: “見積もりの実情と課題“, “IT ユーザとベンダのための定量的見積もりの勧め”, (2005).

[IPA 12]

IPA 独立行政法人 情報処理推進機構(IPA): “ソフトウェア産業の実態把握に関する調査”, (2012).

[IPA 15]

IPA 独立行政法人 情報処理推進機構(IPA): “ソフトウェア開発データ白書 2014-2015”, (2015).

[IPA 16]

IPA 独立行政法人 情報処理推進機構(IPA): “ソフトウェア開発データ白書 2014-2015”, (2016).

[IPSEC 06]

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(IPASEC): “ソフトウェア開発見積もりガイドブック”, (2006).

[IPSEC 16]

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(IPASEC): “ソフトウェア開発見積もりガイドブック”, (2016-2017).

[Ishitani 06]

石谷靖.: “ハイブリッドなコスト見積もりモデルの反復的な構築方法について”, SEC journal, vol7, (2006).

[ITR 16]

ITR: “国内企業 IT 投資動向調査”,
<https://www.itr.co.jp/company/press/161019PR.html>

[JUAS 11]

日本情報システム・ユーザ協会(JUAS): “企業 IT 動向調査 2011”,
<http://www.juas.or.jp/cms/media/2017/02/11itdoukou.pdf>, (2011).

[JUAS 16]

日本情報システム・ユーザ協会(JUAS): “企業 IT 動向調査 2016”,
http://www.juas.or.jp/cms/media/2017/02/it16_ppt.pdf, (2016).

[JUAS 18]

日本情報システム・ユーザ協会(JUAS): “企業 IT 動向調査 2018”,
http://www.juas.or.jp/cms/media/2018/01/it18_yosan.pdf, (2018).

[Jones 97]

Jones, Capers, “Applied Software Measurement: Assuring Productivity and Quality, 2d ed”, NY: McGraw-Hill, (1997)

[Jorgensen 02]

Jorgensen, M., and D.I.K. Sjoberg,: “The Impact of Customer Expectation on Software Development Effort Estimates”, International Journal of Project Managemet, (2002).

[Jones 98]

Jones Capers,: “Estimationg Software Costs”, NY: McGraw-Hill, (1998).

[Kakimoto 06]

柿元 健・角田雅照・大杉直樹・門田暁人・松本健一,「協調フィルタリングに基づく工数見積り手法のデータの欠損に対するロバスト性の評価」, 電子情報通信学会論文誌. D, 情報・システム J89-D(12), pp.2602-2611, (2006).

[Keung 08]

Keung, J., Kitchenham, B., and Jeffery, R: “Analogy-X: Providing Statistical Inference to Analogy-Based Software Cost Estimation”, IEEE Trans. on Software Eng., Vol.34, No.4, pp.471-484, (2008).

[Keung 99]

Keung, J., Kitchenham, B., and Jeffery, R: “An Empirical Study of Analogy-based Software Effort Estimation, Empirical Software Engineering”, IEEE Trans. on Software Eng., Vol.4, Issue 4, pp.135–158, (1999).

[Keisan 10]

情報システムソフトウェア取引高度化コンソーシアム編

http://www.meti.go.jp/policy/it_policy/softseibi/trouble%20cases.pdf

[Keizai 15]

一般財団法人 経済調査会: “ソフトウェア開発データリポジトリの分析 2015”, (2015).

http://www.meti.go.jp/policy/it_policy/softseibi/trouble%20cases.pdf

[Kermerer 87]

Kermerer, C. F.: “An Emprical Validation of Software Cost Estimation Models”, Communications of the ACM, pp.416–429, (1987).

[Kitchenham 08]

Kitchenham, B., Pfleeger, S., McColl,B. and Eagan, S., : “An empirical study of maintenance and development estimation accuracy”, Journal of Systems and Software, Vol.64, No.1, pp.57–77, (2002).

[Kogure 00]

木暮仁, “情報化投資の費用対効果に関する考察”, 論文, (2000)

[Konishi 12]

小西文章・内田眞司・戸田航史・門田暁人: “説明変数のランダムサンプリングを用いた対数重回帰モデルによるソフトウェア開発工数予測の試み”, 情報処理学会論文誌, Vol.2012-SE-175 No2, (2012).

[Kotani 98]

小谷義明, “ファンクションポイント法によるソフトウェア開発規模・工数見積の現状”, 関西情報センター, (1998).

[Kolenikov 2010]

S.Kolenikov, K.A. Bolleny.: Testing Negative Error Variances: Is a Heywood Case a Symptom of Misspecification, (2010).

[Kubo 14]

久保拓弥: “データ解析のための統計モデリング入門”, (2014).

[Kurihara 10]

栗原瑛・東基衛・野中誠.: “コスト誘因の評価実績に基づいた COCOMO II ソフトウェア開発工数見積もりの補正手法”, 情報処理学会論文誌, Vol.2010-SE-167 No2 (2010).

[Laranjeira 10]

Laranjeira, Luiz.: “Software Size Estimation of Object-Oriented System”, IEEE Transaction on Software Engineering, (1990).

[Lawlis 95]

Lawlis, Dr.patricia K., Capt. Rovert M/Flowe, and Capt.James B. Thordahl: “A Correlational Study of CMM and Software Development Performance”, Crosstalk, (1995).

[McCabe 76]

McCabe, Tom: “A Complexity Measure”, IEEE Transactions on Software Engineering, volSE-2, no12, pp308-320, (1976).

[McConnell 04]

McConnell, Steve : “ソフトウェア開発プロフェッショナル”, 日経 BP, (2004).

[McGraw 04]

McGarry, John, et al., : “Practical Software Measurement:Objective Information for Decision Makers”,Boston,MA:Addison-Wesley, (2004).

[Monden 10]

門田暁人, 小林健一., : “線形回帰モデルを用いたソフトウェア開発予測における対数変換効果”,コンピュータソフトウェア,vol27, No4, pp234-239, (2010).

[Mittas 10]

Mittas, N., Athanasiades, M and Angelis, L: “Improving analogy -base software cost estimation by a resampling method”, Information and Software Technology,vol50, No3, pp221-230, (2002).

[M & t Consulting 02]

m & t Consulting, “WBS とマイルストーン”, Project Management OS 3, (2002).

[Miyazaki 94]

Miyazaki, Y, Terakado, M, Ozaki, K, and Nozaki, H.: “Robust Regression for Developing Software Estimation Models”, Journal of Systems and Software“, vol.27, No.1, pp.3-16, (1994).

[Michael 84]

Michael.: “Program Complexity and Programming Style”, Proc.1st Int. Conf.Data Engineering New York, NY:IEEE Computer Society Press, pp.537-541, (1984).

[Niessink 97]

Niessink, F., and H/van Vliet: “Predicting Maintenance Effort With Function Point”, IEEE Computer Society, pp32–39, (1997).

[Nikkei 18]

日経 XTECH [特報] 27 億円の賠償巡り新たな IT 裁判始まる, 文化シャッタが提訴 <http://tech.nikkeibp.co.jp/atcl/nxt/column/18/00001/00014/>

[Oosugi 04]

大杉直樹・門田暁人・森崎修司・松本健一: “強調フィルタリングに基づくソフトウェア機能推薦システム”, 情報処理学会論文誌, Vo45. No1, (2004).

[Okamura 09]

岡村正司: “プロジェクトコスト見積もり入門~ファンクションポイント, COCOMO II, WBS によるソフトウェア開発コストの導き方”, (2009).

[Ohsugi 2006]

大杉 直樹・角田雅照・門田暁人, 「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り」, SEC journal 2(1), pp16–25, (2006).

[Park 96]

Park, R. E., : “A Manager’s Checklist for Validating Software Cost and Schedule Estimates”, American Programmer, pp30–35, (1996).

[Putnam 99]

Putnam, Lawrence H., and Ware Myers.,: “Get the Estimate Right”, American Programmer, pp4–12, (1999).

[Putnam 05]

Putnam, Lawrence H., and Ware Myers.,: “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, (2005).

[Philippe 00]

Philippe Kruchten: “The Rational Unified Process”, Addison Wesley Publishing Company, (2000).

[Sackman 68]

Sackman, H., Erikson, W.J., Grant, E.E., : “Exploratory Experimental Studies Comparing Online and Offline Programming Performance”,Communications of the ACM v.11, no.1,,pp3–11, (1968).

[Schwaber 02]

Schwaber, Ken and Mike Beedle, : “Agile Software Development with Scrum, Englewood Cliffs”,NJ:Prentice Hall, (2002).

[Smith 99]

Smith, John, : “The Estimation of Effort Based on Use Cases”,Rational Software Whitepaer TP–171, (1999).

[Strike 01]

Strike, K., El Eman, K., and Madhavji, N.,: “Software Cost Estimation with Incomplete Data”, IEEE Trans. on Soft.Eng., vol.27, no.10, pp.890–908, (2001).

[Shulani 99]

Shulani, S., Boehm, B. and Steece, B.: “Bayesian analysis of empirical software engineering cost models”, IEEE Transaction on Software Engineering, vol.25, No.4, pp573–583, (1999).

[Srinivasan 95]

Srinivasan, K. and Fisher, D.: “Machine learning approaches to estimation of software development effort”, IEEE Transaction on Software Engineering, vol.21, No.2, pp126–137, (1995).

[Steve McConnell 06]

Steve McConnell, “ソフトウェア見積もり –人月の暗黙知を解き明かす–”, 日経BP 社, (2006)

[Takahashi 01]

TAKAHASHI M, AOKI T, SHIMOMURA Takao: “Applicability of Object Point Analysis to Software Sizing”, Proc.15th Asia Quality Symposium, pp228–231, (2001).

[Tamura 04]

田村晃一・柿元健・戸田航史・角田雅照・門田暁人・松本健一・大杉直樹: “工数予測における類似性に基づく欠損地補間法の実験的評価”, 情報処理学会論文誌, Vo45, No.1, (2004).

[Toda 05]

戸田航史, 他: “工数見積もりモデルで予測できないソフトウェアプロジェクトの特徴分析” 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, SS2005–74, pp67–72, (2005).

[Toda 09]

戸田航史, 他: “ソフトウェア開発工数予測のためのフィットデータ選定方法” 情報処理学会論文誌, vol50, no11, pp2699–2709, (2009).

[Toyoda 12–1]

豊田秀樹: “因子分析入門”, “東京図書”, (2012).

[Toyoda 14]

豊田秀樹: “共分散構造分析”, “東京図書”, (2014).

[Toyoda 98]

豊田秀樹: “共分散構造分析 [入門編]”, “朝倉書店”, (1998).

[Toyoda 12-2]

豊田秀樹: “共分散構造分析 [数理編]”, “朝倉書店”, (2012).

[Tsunoda 05]

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 46(5), pp1155-1164, (2005).

[Tsunoda 09]

角田雅照・門田暁人・松本健一・高橋明彦, “生産性要因に基づいて層別されたソフトウェア開発工数見積りモデル”, SEC Journal Vol5, No1 (2009), pp1155-1164, (2005).

[Tsunoda 09]

Tsunoda Masateru・Monden Akito・Matsumoto Ken-ichi・Takahashi Akihiko: “Software Development Effort Estimation Models Stratified by Productivity Factors”, Sec journal. No5 No.1 Mar, (2009).

[Tsunoda 13]

角田雅照・大杉直樹・門田暁人・松本健一: “開発要員数とその誤差を考慮した工数見積り方法”, 情報処理学会. Vol, 2013-se181 No.7, (2013).

[University 97]

University of Southern California, : “Cocomo2Model Definition Manual”, v1.4 undated, (1997).

[Van 91]

Van Genuchten, Michiel, "Why Is Software Late? An empirical Study of Reasons for Delay in Software Development", IEEE Transactions on Software Engineering SE-17 no.6(June), pp582-590, (1991).

[Valett 89]

Valett, J., and F/E/McGarry, "A Summary of Software Measurement Experiences in the Software Engineering Laboratory", Journal of Systems and Software, pp137-148, (1989).

[Watts 89]

Watts, S. Humphrey: "Managing the Software Process", Addison Wesley Publishing Company, (1989).

[Weinberg 94]

Weinberg, Gerald M., "Quality Software Management", vol3, Congruent Action, NY: Dorset House, (1994).

[Wiegers 03]

Wiegers, Kari, "Software Requirements", WA: Microsoft Press, (2003).

[Watanabe 87]

渡辺純一 : "品質と生産性を重視したソフトウェア開発プロジェクト開発プロジェクト技法", 近代科学者, (1987).

[Watanabe 07]

渡辺瑞穂 , 他: "ソフトウェア開発工数予測における特異プロジェクト除去の効果", 日本ソフトウェア科学会論文集, (2007).

[Yanai 90]

柳生晴夫，他: “因子分析 ～その理論と方法～”, 日本ソフトウェア科学会論文
集, 朝倉書店, (1990).

関連業績リスト

- [1] 志田剛，押野智樹，津田和彦：“ソフトウェア開発見積もりに影響する因子の抽出”，電気学会誌，(2017)，137 号，No7，P.960-986（査読付き学術論文）

- [2] Tsuyoshi Shida, Kazuhiko Tsuda: “A Study on Factors Causing Cost Variation of Project Delivery Estimation for Software Development” , The Fifth Asian Conference on Information Systems, (2016) (査読付き国際会議)

- [3] Tsuyoshi Shida, Kazuhiko Tsuda: “ A Study of Software Estimation Factors Extracted Using Covariance Structure Analysis” , 2017 KES 21th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, pp.1378-1387. (2017) (査読付き国際会議)

- [4] 志田剛，高橋雅和，津田和彦：“共分散構造分析から抽出したソフトウェア見積もり因子の研究”，電気学会研究会資料，IS 2017(1-5・7-25)，pp.45-50.（口頭発表）

- [5] Tsuyoshi Shida, Kazuhiko Tsuda: “A Study of Differences by Industry Using Factor Models Influencing Software Development Estimates” , Artificial Intelligence Research, 2018, Vol. 7, No. 2 (2018)（査読付き学術論文）