# 開発計画時における要件定義工程での開発体制と開発規模の関係分析

伏田 享平[1]　渡辺 絢子[1]　今井 二郎[1]　山中 啓之[1]　藤貫 美佐[1]　戸村 元久[1]

**概要**：IT システム開発においてプロジェクトが失敗する原因の一つとして，計画時やプロジェクトの早期に開発対象システムの仕様を十分に把握できていなかったことが挙げられる．開発対象の規模に応じた十分なプロジェクト体制を構築できていない場合，開発要員がシステム全体の仕様を十分把握することができず，結果としてプロジェクトの問題化につながる可能性がある．本稿では開発計画時の規模とプロジェクト体制に着目する．開発要員 1 人あたりが把握すべきシステムの規模を「仕様把握規模」と定義し，この規模とプロジェクトの成否との関係を定量的に分析した．分析にあたっては，要件定義工程での体制と開発規模との関係に着目した．分析の結果，仕様把握規模が 50KS/人以上であったプロジェクトは失敗する可能性があることがわかった．仕様把握規模を導入することで，開発体制の観点からプロジェクト計画の妥当性をチェックできる可能性がある．

## 1. はじめに

IT システム開発においてプロジェクトが失敗する原因はこれまでに多く調査，研究がなされている．たとえば，情報処理推進機構ソフトウェア・エンジニアリング・センター（IPA/SEC）では，107 件の大問題プロジェクトを対象に事例分析を行っている [11]．このような失敗原因の一つとして，計画時や要件定義工程で開発対象システムの規模に応じた十分な体制を構築できていなかったことが考えられる．プロジェクトの計画段階では仕様が不明確であったり，受発注者間で認識齟齬があったりすることが多い．要件定義を進める中でプロジェクトのスコープが変動することも多い．プロジェクトの計画時点では，これらの変動要素をあらかじめ認識した上で，規模，工数の見積もりを行い，リスクを洗い出した上で，十分な体制を計画することが求められる．

工数見積もりの結果は，最終的に工程別に期間と要員数の関係に分解される．これらの期間と要員数の関係を表すために，**図 1** のようなリソースヒストグラム（山積み表）や**図 2** のような体制図を作成することが多い．工数は要員数と開発期間との積で表される．しかし，これらの関係は可換ではない [8], [13]．開発期間を短縮する場合には，開発要員を増員すれば良いが，一般的に新規要員は教育コストなどがかかるため生産性が落ちる．また，ビジネス上の問題で納期（開発期間）は変更できない場合が多い．

本稿ではプロジェクトの計画妥当性を開発体制の観点から評価する手法について検討する．開発要員のスキルや経験などに依存せず，計画の妥当性を定量的に評価するため，本稿では開発対象の規模と開発体制との関係に着目した．

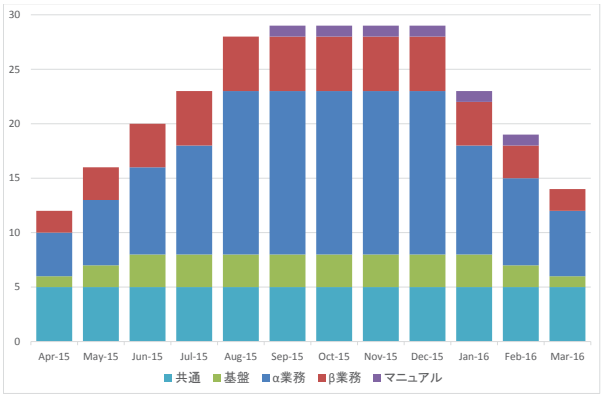



**図 1**　リソースヒストグラム（山積み表）の例

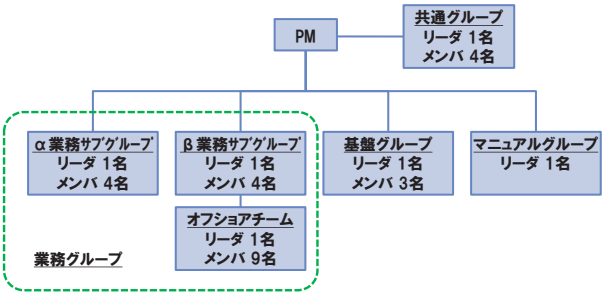



**図 2**　体制図の例

[1] 株式会社 NTT データ
NTT DATA Corporation

具体的には，下記に示す 2 つのリサーチクエスチョンを設定した．

**RQ1** 開発対象の規模に対して開発体制はどの程度の規模であったのか．

**RQ2** 開発要員一人あたりが担当することとなる規模は，プロジェクトの成否に影響を与えるのか．

実際に開発プロジェクトにおいて，開発対象の規模に対してどの程度の開発要員の配置が計画されているかを明らかにするために RQ1 を設定した．また RQ1 で明らかにした開発要員 1 人あたりの規模とプロジェクトの成否の関係を明らかにするため RQ2 を設定した．

上記のリサーチクエスチョンに対して，本研究では「仕様把握規模」という概念を導入し分析した．仕様把握規模は，開発要員 1 人あたりが把握すべき開発対象の規模を表す概念である．仕様把握規模とプロジェクトの成否の関係を分析することで，開発規模に応じた要員数を明らかにする．分析は株式会社エヌ・ティ・ティ・データ（以降，NTT データと略す）で収集・蓄積されているデータを対象に行った．特にプロジェクトの根本的な問題が混入しやすい，要件定義工程を対象に分析を行った．本研究の貢献は以下の 2 点である．

- 実際のプロジェクトデータを分析することで，開発対象の規模と開発要員の関係を明らかにした．
- プロジェクトの上流工程において，開発要員 1 人あたりの開発規模とプロジェクトの成否との関係を明らかにした．

## 2. キーアイデア

プロジェクトの成否を決定する要因の一つは，開発対象の規模に対して十分な体制を組めているかにあると考えられる．開発対象の規模に対して過小な開発体制であった場合，開発に必要な作業を実施することができず，結果として納期遅延や品質不良の多発を招く可能性がある．特に要件定義や設計といった上流工程において過小な体制であった場合には，開発対象のシステム仕様を十分に把握することができず，工程遅延や品質不良につながる可能性が高くなる．一方で，要員が多すぎるなど過剰な開発体制であった場合，開発要員間のコミュニケーションコストが増大し，結果としてコスト超過となる可能性がある．また，作業が発生していないにも関わらずプロジェクトに配置されているため，別プロジェクトに対して適切なタイミングで適切な要員を投入できないといった機会損失にもつながる．

図 2 に IT システム開発における典型的な開発プロジェクトの体制図を示す．Conway の法則 [3] として知られるように，開発プロジェクトの体制は開発システムのアーキテクチャに従うことが多い．IT システム開発では，システムのアプリケーション部分を開発する「業務グループ」とネットワークやサーバ構築などの基盤部分の構築を担当する「基盤グループ」，進捗や品質などの管理，共通業務を担当する「共通グループ」に大別されることが多い．各グループはさらに業務・機能ごとにサブグループに分割されることがある．各グループには「グループリーダー」と呼ばれる責任者が配置され，担当するグループの管理・統括を行う．「プロジェクトマネージャ（PM）」はプロジェクト全体，すなわち全グループを管理，統括する役目である．グループ内の要員構成は，工程が進むに従って変動するのが一般的である．たとえば，要件定義から設計，製造と工程が進むに従い，要員数は増加することが多い．

開発対象の規模を計量する尺度はこれまでに多く提案されてきている．代表的なものとしては，ソースコード行数（Lines of Code: LoC）やファンクションポイント（FP）が挙げられる．また，上流工程で見積もり，計測が可能な尺度として設計書のページ数やストーリーポイントなども挙げられる．LoC に関しては，どの範囲を測定対象として計上するかによってもその量が変動する．コメント行や空白行の取り扱い（計上するかしないか）といったものはその典型例である．また，既存のソースコードを流用して開発する場合には，流用元のソースコード行数を含めるか否かといった議論もある．

本稿では**仕様把握規模**という概念を導入する．仕様把握規模は開発体制規模に対する開発規模を表し，次式で定義される．

$$\text{仕様把握規模} := \frac{\text{開発規模}}{\text{開発体制規模}}$$

ここで開発規模は LoC や FP などの尺度を用いて計量された値，開発体制規模は開発プロジェクトにおける平均要員数などが該当する．仕様把握規模は直観的には「当該プロジェクトにおいて開発要員 1 人が最低限把握しておくべき仕様の量」を表している．また，開発規模は開発対象のシステムで実現すべき仕様の量を表すものと考える．

仕様把握規模を用いることで，開発規模に対して十分な体制となっているかを検証することが可能となる．たとえば，水準値を設定することで，「規模に対して要員数が不足している」といった判断を下すことが可能となる．なお，適切な体制となっているかは，要員数以外に開発対象の業務知識，実装能力など，要員の能力に依存する面もある．しかし，開発対象の規模，仕様の複雑さに応じた体制となっているかを定量的に判断するために，仕様把握規模の導入には意義があるものと考える．

次節以降では，実際のプロプライエタリな開発プロジェクトのデータを対象に，仕様把握規模に関する定量的評価を行う．

## 3. 分析に用いたデータ

本稿では NTT データで 2009 年から 2014 年までに完了したプロジェクトのデータを分析対象とした．NTT データ

ではプロジェクト実施前に計画が妥当なものとなっているか審査を行っている．審査にあたり，プロジェクトは規模や工数などの見積もり結果，および開発体制やスケジュールなどの資料（以降，これらの資料を「計画時データ」と呼ぶ）を提出しなくてはならない．組織長などの意思決定者や PMO（プロジェクトマネジメントオフィス）はこれらの資料をもとに計画内容の妥当性をチェックし，最終的な意思決定（プロジェクトの開始判断）を行う．また，プロジェクトの完了時には，規模や工数，品質に関する実績値（以降，これらのデータを「完了時データ」と呼ぶ）を報告することも社内ルールとして義務づけられている．本稿ではこれらの計画時データおよび完了時データを分析した．以下，仕様把握規模算出に必要な開発規模および開発体制規模の定義について述べる．あわせて，プロジェクトの成功・失敗の定義も与える．

### 3.1 開発規模

開発規模に関するデータとして，完了時データの LoC の値を用いた．実際に仕様把握規模を用いた計画妥当性を検証する場合，規模としては見積もり値が採用されることとなる．ただし，今回の分析では，実際に開発，納品したシステムの規模に対して必要な体制（要員数）が計画されていたかを分析するため，開発規模は実績値を採用する．

LoC の計測範囲として下記に示す箇所の行数を計上する．以降，この規模を「プログラム開発規模」と呼ぶ．

- 新規に作成したモジュールの行数
- 既存モジュールを修正した際の追加行数
- 再利用した（一部手を加えた）既存モジュールのうち修正を加えていない箇所の行数

なお，仕様の大きさを表す規模としては，FP や設計書のページ数なども考えられる．一方で LoC も仕様に基づきコーディングした規模を表しており，仕様の大きさをある程度表現できると考えられる．本来は FP など他の指標についても検証すべきだが，LoC 以外の指標で規模が報告されたデータ件数が少ないため，今回の分析では LoC のみを採用する．

### 3.2 開発体制規模

開発体制規模に関するデータとして，計画時データにおける要員数を採用した．要員数の実績値（プロジェクト終了時の値）を採用した場合，その値はプロジェクトが問題化していると，是正措置（要員投入）後の値となってしまう．本稿では計画時における体制がプロジェクトの成否に影響を与えているかを分析したいため，計画時の要員数を採用した．また，本稿では特に上流工程での問題抑止を目的とするため，要件定義工程の要員数に着目した．

図 2 に示したように，プロジェクトはそれぞれの役割に応じてグループを構成する．グループの中で実際にアプリケーションを開発するのは，「業務グループ」と呼ばれるグループの要員である．そのため，要員数は業務グループの要員数にのみ着目し，共通グループや基盤グループなど，アプリケーション開発に関わらない要員，および全体を統括するプロジェクトマネージャは考慮しないこととする．

**表 1** 開発規模および体制規模の統計値

| 指標 | データ数 | 最小値 | 中央値 | 平均値 | 最大値 |
|---|---|---|---|---|---|
| 開発規模 | 19 | 2.99 | 171.90 | 581.40 | 4081.00 |
| 体制規模 | 19 | 1.00 | 10.00 | 19.58 | 103.50 |

要員数の計上にあたっては，計画時データに含まれる要件定義工程の体制図もしくはリソースヒストグラムから算出した．また，リーダなどは複数のグループのリーダを兼務している場合がある．そのような場合には，兼務の数で割った数を計上している．たとえば，1 人が業務グループと基盤グループのリーダを兼任している場合は，業務グループのリーダは 0.5 人，基盤グループのリーダは 0.5 人と計上する．計上にあたっては工程ごとの要員が明確にわかるプロジェクトのみを対象としている．たとえば，一部の工程のみ参画するメンバであっても全て記載したような体制図しか記録されていないプロジェクトについては，分析対象から除外した．

### 3.3 プロジェクトの成否

プロジェクトの成否は，QCD（品質・コスト・納期）の観点での評価や，納品後の重大欠陥の発生数，発注者（顧客）が満足したかなど，様々な定義が考えられる．今回分析対象としたプロジェクトは，いずれもプロジェクト開始時に難易度が高い [*1] と判定されたプロジェクトのデータである．本稿では，このプロジェクトデータのうち，開発計画時の予算をプロジェクト終了の段階で超過したプロジェクトを失敗，それ以外のプロジェクトは成功と定義する．

## 4. 仕様把握規模の分析

本稿では前節で定義した仕様把握規模に関して，プロジェクトの成否を判別する水準値を明らかにするため分析を行った．以降，分析結果と考察を述べる．

### 4.1 分析に用いたデータの傾向

分析対象としたプロジェクトは 19 件である．分析対象プロジェクトは，いずれもアプリケーション部分の開発が実際に行われたものである．**図 3** にプログラム開発規模（KS），**図 4** に要員数（人）の分布を示す．また**表 1** にはプログラム開発規模および要員数の統計値を示す．

規模としては 100KS から 400KS 程度のプロジェクトが多いが，中には 1000KS を超える大規模なプロジェクトも

[*1] 契約形態や利用技術など，NTT データで定められた基準に基づき判断されている．具体的な判断基準については，社外秘情報のため非公開とする．

表 2　仕様把握規模の統計値

| データ数 | 失敗プロジェクト数 | 中央値 | 平均値 |
|---|---|---|---|
| 19 | 5 | 17.02 | 49.16 |

いくつか存在している．要員数は実際に要件定義工程に関わる開発要員数であり，おおよそ 5 人から 20 人程度で実施されている．ただし，中には 100 人を超える要員が参画しているプロジェクトも存在する．

### 4.2　仕様把握規模の傾向

**図 5** および**表 2** に仕様把握規模の分布および統計値を示す．実際に仕様把握規模を用いて体制面の観点から計画の妥当性を検証する場合には，何らかの水準値を設定する必要がある．水準値を設定することで「仕様把握規模が水準値を超えたプロジェクトはプロジェクト実行中，重点的に管理を行う」などの対処が可能となる．本稿では表 2 に示した仕様把握規模の中央値および平均値より，20KS/人，50KS/人，100KS/人の 3 種類の水準値を設定し，以降の分析を進める．

**図 6** は横軸に要員数，縦軸にプログラム開発規模を取ったグラフである．なお，要員数の具体的なスケールに関しては社外秘情報のため非公開とする．図 6 でプロットされた点は，青い丸は成功プロジェクト，赤い三角は失敗プロジェクトをそれぞれ表している．またあわせて傾きが 20KS/人（青色），50KS/人（オレンジ色），100KS/人（赤色）の 3 種類の直線を描いている．これらの直線は，それぞれ先に設定した仕様把握規模の水準値を表している．水準値を使った計画の妥当性の検証方法として，これらの直線の上部にあるプロジェクトを体制面でリスクのあるプロジェクトと判定する，といった運用が考えられる．

## 5.　考察

### 5.1　リサーチクエスチョンに対する回答

前節での分析結果を踏まえて，1 節で挙げたリサーチクエスチョンに対する回答を述べる．

**開発要員 1 人あたりが担当することとなる規模**

表 2 に示すように，開発要員 1 人あたりが担当すること


図 3　規模の分布

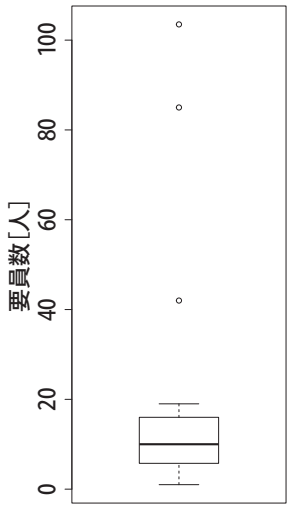



図 4　要員数の分布

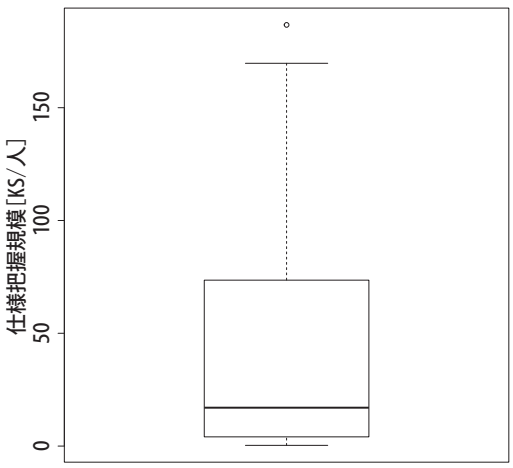
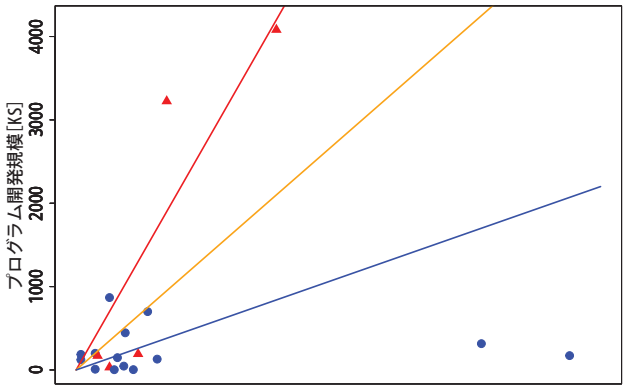
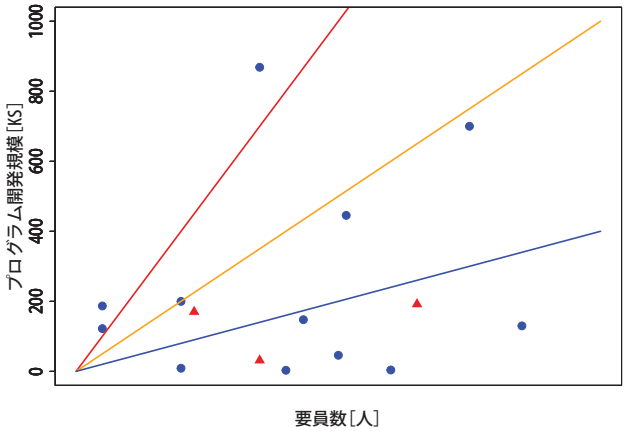



図 5　仕様把握規模の分布




(a) 全体




(b) (a) の左下の領域を拡大

図 6　開発規模と要員数の関係

となる規模の中央値は 17KS であった．Robles らの研究で扱っている OpenStack の開発では，開発規模がおおよそ 1650KS であるのに対し，開発者は 1410 名であったことが報告されている [9]．McConell の書籍では，35KS〜100KS 程度のプロジェクトではチームの人数が 7 人を超えてはいけないとの言及がある [5]．これらの開発規模と仕様把握規模は，厳密には概念は異なる．また，プロプライエタリな IT システムとオープンソースソフトウェアではその開発形態は異なる．ただ，これらの文献で言及されている 1 人あたりの開発規模と，今回の仕様把握規模では，その値が大きく異ならないことがわかる．

**仕様把握規模がプロジェクトの成否に与える影響**

図 6 に示すように，失敗プロジェクトのうち 2 件は仕様把握規模が 50KS/人以上となっている．このようなプロジェクトに対しては，プロジェクト計画時に仕様把握規模を用いることで，体制面での妥当性，すなわち開発規模に対して要員配置が十分であるかを確認できる．また，設定した水準値を上回るプロジェクトに対して個別に体制面での問題が無いか詳細に計画をチェックするなど，是正措置を執るきっかけとしても利用することができる．

一方で，20KS/人を下回っているにも関わらず，失敗プロジェクトとなっているプロジェクトも 2 件存在する．このようなプロジェクトについては，体制面以外での要因により失敗したと考えられる．そのため体制面とプロジェクトの成否との因果関係はさらに分析する必要がある．

### 5.2 分析結果に対する妥当性の検証

プロジェクトの成功・失敗要因は複数存在する．そのため，実際には体制には問題が無く，他の要因で失敗したプロジェクトが混在している可能性もある．実際に図 6 が示すように失敗プロジェクトのうち開発規模が大きなものが 2 件あることから，開発規模の大きさが失敗の主要因である可能性もある．また，納期の制約のような開発期間についても本研究では考慮していない．ただし，失敗プロジェクトの完了後におけるプロジェクトマネージャの所見を確認すると，失敗要因の 1 つとして体制面での弱さを挙げているものが存在した．そのため，失敗要因の一つとして体制面の問題があるとみなすことは妥当であると考える．

開発規模に関するデータとして，今回の分析では LoC を採用した．ただし，仕様の大きさを表す量として LoC を採用することは必ずしも妥当とは言えないケースがある．LoC は仕様把握規模が想定している上流工程では見積値となる．そのため，開発が進むにつれ LoC の値が大きく変動する可能性がある．また，コード自動生成ツールを用いている場合やコンポーネントウェアを利用している場合は，実際の機能数や仕様の数と比較して増減が発生する可能性がある．そのため，上流工程で値を確定することができる，より妥当な規模尺度を採用する方が良い可能性がある．たとえば，仕様の大きさを表す設計書ページ数や FP は，仕様把握規模の概念により適した規模尺度であると考えられる．しかし，今回分析対象としたデータでは，FP や設計書ページ数による規模計測が行われているものが少なかったため分析を行っていない．

今回分析に使用した要員数のデータは第 1，第 2，第 3 著者が，リソースヒストグラム，体制図を目視で確認し，手作業で要員数を計上したものである．そのため，要員数の計上誤りがある可能性がある．ただし，要員数計上にあたっては，計上に関するルールをあらかじめ定めた．また，計測したものを一部抜き出して相互レビューを行い，誤りがないことを確認している．そのため，計測データは一定の品質が確保されていると考える．

本稿で示している仕様把握規模の水準値は，今回分析対象としたデータから求められたものである．開発組織や環境，開発規模の計測ルールによって水準値は変動し，他組織では結果の傾向が異なるおそれがある．ただし，本稿の立場は開発規模に対する体制面の妥当性検証に利用できる指標として「仕様把握規模」を導入したものである．よって，今回の分析結果で示した水準値は，他組織でも採用されうるものという立場は取らない．実際に仕様把握規模によるプロジェクト計画の妥当性確認を行う場合には，自組織において本稿で示したものと同様の分析を行い，組織内で水準値の設定を行う必要がある．

## 6. 関連研究

本研究で着目している開発プロジェクトの体制に関しては，これまでに多くの研究がされている．これらの研究は，規模や要員数が工数や品質に与える影響を分析したものと，プロジェクトの体制，組織構造に着目したものの 2 つに大別できる．ここではこれらの関連研究について紹介する．

要員数はプロジェクトの生産性と関連が強いことがこれまでに指摘されている．角田らの研究 [14] では，最大開発要員数を FP の実績値で割ったものを「開発要員規模比」と定義して分析している．この研究では，分析の結果，開発要員規模比が高いプロジェクトほど生産性が低くなっていることが示されている．Brooks の法則 [2] として知られるように，開発要員が増加するほどコミュニケーションパスが指数的に増加し，結果として生産性が低下することが知られている．この研究はその事実を定量的に示したものと言える．これに対し本研究は，プロジェクトの計画妥当性を検証するため，要件定義工程に着目して分析を行っている点が異なる．

開発プロジェクトの要員数や体制が品質に与える影響分析もこれまでに行われている．Ali Babar らの研究では，プロジェクトの要員数に応じてプロダクトの品質に差異が発生するかを実証的に確かめている [1]．その結果，要員数による品質の差は見られず，プロダクトの品質と個人の

スキル，経験とがトレードオフの関係にあると結論づけている．古山らの研究では，テスト体制がコスト，納期，品質に与える影響について分析している [12]．この分析の結果，テスト要員数が充足していると，納期遅延の防止に効果が見られることが示されている．Nagappan らの研究 [7] や Meneely らの研究 [6] は，開発プロジェクトの組織構造を定量化したメトリクスを用いることで品質に与える影響を分析している．いずれの研究も組織構造を定量化したメトリクスは欠陥予測に対して有用であることが示されている．これらの研究に対し，本研究では規模と体制の関係に着目し，コスト超過の観点で分析を行っている．

開発チームのサイズ，すなわち要員数に関しては，アジャイルソフトウェア開発においていくつか経験的に言われていることがある．Coplien らの書籍では，大規模なソフトウェア（25KS 以上）は，開発チームが大きすぎたり小さすぎたりすると期日内に予算内でリリースできないとの言及がある [4]．また，Schwaber らによる Scrum のガイドでは，メンバが 3 人未満もしくは 9 人を超えた場合にはプロジェクトの生産性向上につながらないとの言及がある [10]．ただし，これらはいずれも経験則に基づき導かれているものであり，その妥当性について定量的な評価は行われていない．本研究は開発対象の規模に応じた体制，要員数を検討するため仕様把握規模を導入している．仕様把握規模を組織内で運用することで，体制の妥当性に関して定量的な根拠を与えることが可能となる．

## 7. おわりに

本稿では開発規模とプロジェクトの体制との関係を定量的に明らかにすることを目的として分析を行った．分析にあたっては，開発要員 1 人あたりの開発規模を表す指標である「仕様把握規模」を導入し，NTT データでの開発データを利用して分析を行った．分析の結果，分析対象において仕様把握規模の中央値は 17KS/人であることが明らかになった．また，仕様把握規模の統計値をもとに水準値を設定し，プロジェクトの成否を分析した．その結果，開発規模に対する開発体制の妥当性を，仕様把握規模を用いて確認できる可能性があることを示した．

今後の課題として以下の 3 点を挙げる．

**生産性や品質への影響に関する分析**　6 節でも示したとおり，開発体制は開発プロジェクトの生産性や品質に影響を与えることが示唆されている．今回導入した仕様把握規模から，開発体制が生産性や品質に与える影響を定量的に評価することが重要であると考える．

**アプリケーション部分以外の妥当性検証方法**　基盤グループや管理グループの体制を検証するにあたっては，これらに対応する妥当な規模尺度を採用し，分析を行う必要があると考える．

**組織の構造に着目した分析**　今回の分析では業務グループ内のサブグループの構成については考慮していない．実際のプロジェクトでは，複数の会社が共同で開発を行うことも多い．このようなプロジェクトの構成も考慮した分析もあわせて行う必要があると考える．

### 参考文献

[1] Babar, M. A. and Kitchenham, B.: The Impact of Group Size on Software Architecture Evaluation: A Controlled Experiment, *Proceedings of the First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*, pp. 420–429 (2007).

[2] Brooks, Jr., F. P.: *The Mythical Man-month (Anniversary Ed.)*, Addison-Wesley Longman Publishing Co., Inc. (1995). （滝沢 徹，牧野 祐子，富澤 昇訳：人月の神話，丸善 (2014)）.

[3] Conway, M. E.: How do committees invent?, *Datamation*, Vol. 14, No. 4, pp. 28–31 (1968).

[4] Coplien, J. O. and Harrison, N. B.: *Organizational Patterns of Agile Software Development*, Prentice-Hall, Inc. (2004). （和智 右桂訳：組織パターン チームの成長によりアジャイルソフトウェア開発の変革を促す，翔泳社 (2013)）.

[5] McConnell, S.: *Software Estimation: Demystifying the Black Art*, Microsoft Corporation, Redmond, Washington, U.S.A. (2006). （田沢 恵，溝口 真理子訳，久手堅 憲之監修：ソフトウェア見積り 人月の暗黙知を解き明かす，日経 BP ソフトプレス (2006)）.

[6] Meneely, A., Williams, L., Snipes, W. and Osborne, J.: Predicting Failures with Developer Networks and Social Network Analysis, *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2008)*, pp. 13–23 (2008).

[7] Nagappan, N., Murphy, B. and Basili, V.: The Influence of Organizational Structure on Software Quality: An Empirical Case Study, *Proceedings of the 30th International Conference on Software Engineering (ICSE '08)*, pp. 521–530 (2008).

[8] Putnam, L.: A General Empirical Solution to the Macro Software Sizing and Estimating Problem, *IEEE Transactions on Software Engineering*, Vol. SE-4, No. 4, pp. 345–361 (1978).

[9] Robles, G., González-Barahona, J. M., Cervigón, C., Capiluppi, A. and Izquierdo-Cortázar, D.: Estimating Development Effort in Free/Open Source Software Projects by Mining Software Repositories: A Case Study of OpenStack, *Proceedings of the 11th Working Conference on Mining Software Repositories (MSR 2014)*, pp. 222–231 (2014).

[10] Schwaber, K. and Sutherlan, J.: Scrum Guide 2013, ScrumGuides.org (online), available from ⟨http://www.scrumguides.org/⟩ (accessed 2015-4-30).

[11] 独立行政法人 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：IT プロジェクトの「見える化」総集編，日経 BP (2008).

[12] 古山恒夫，菊地奈穂美，安田　守，鶴保征城：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の分析，情報処理学会論文誌，Vol. 48, No. 8, pp. 2608–2619 (2007).

[13] 柿元　健，門田暁人，角田雅照，松本健一，菊地奈穂美：規模・工期・要員数・工数の関係の定量的導出，*SEC journal*, No. 14, pp. 6–11 (2008).

[14] 角田雅照，門田暁人，松本健一：ソフトウェア開発工数積算のための生産性分析，経済調査研究レビュー，No. 1, pp. 114–119 (2007).