アジャイル開発の見積りと契約モデルに関する研究

筑波大学審査学位論文（博士）

２０１５

吉　田　　知　加

筑波大学大学院
ビジネス科学研究科　企業科学専攻

論　文　概　要

博士（システムズ・マネジメント）

アジャイル開発の見積りと契約モデルに関する研究

ビジネス科学研究科

企業科学専攻

吉田知加

企業の情報システム環境は，ホストコンピューターによる集中処理の時代からクライアントサーバー分散処理を経て，現在ではクラウドコンピューティングの時代に変革している．そこでは，コンピューターを物理的に集中配置し，データ処理は複数のサーバーで行うといった集中と分散を組み合わせた処理形態をとる．またソフトウェア開発では，開発期間を短縮し早期に業務に適用することを目的に，ERP（Enterprise Resource Planning）の様なパッケージソリューションの適用やクラウドコンピューティングに代表される課金利用など，その利用形態も多様化している．しかし，情報システムの存在が企業の経営成果達成に不可欠な基盤となった今日，戦略の差別化をはかるためにカスタムメードのシステム構築はいまだ必要とされ続けている．

そして，そのソフトウェア開発モデルは規模や環境条件により，プロトタイプ型，スパイラル型，反復型，そしてアジャイル型と多様化しており，各モデルの適用率は国により異なる．日本ではウォーターフォール型開発が，変更対応の難しさ，納期遅れやコスト超過の多発といった多くの問題を抱えつつも適用率の首位を維持しているが，欧米でのソフトウェア開発モデルはウォーターフォール型開発からアジャイル型開発にシフトしている．2009 年の第 3 四半期に実施した調査で，約 45%の企業が，50%以上の開発プロジェクトでアジャイル型開発を採用しているという結果が示され，ウォーターフォール型開発を凌いで一般的開発工程とされている．ウォーターフォール型が主流の日本では，経営環境変化に柔軟に対応でき,迅速なシステム開発が可能なアジャイル開発が注目されているものの，いまだアジャイル開発を含む反復型開発を採用したプロジェクトは全体の 2.8%である．

この要因として，欧米での採用率の高さは，情報システム構築を自社で内製化する企業が多く，開発を外部に発注する企業が少ないこと，一方日本では，一括請負契約での開発

が多く，早期に要件や開発コストを確定しておきたいということが考えられる．そして開発中に環境変化が予想され，迅速な対応が必要なプロジェクトも，途中，幾度かのリリースが必要となるプロジェクトも，一括請負契約で進めている場合が想定される．その様なプロジェクトにアジャイル開発を適用するといった適用推進への貢献が，本研究の目的である．

はじめに，想定される日本でのソフトウェア開発の現状を確認するために，日本の IT 企業がアジャイル開発を適用するにあたっての阻害要因について，アンケートを用いて調査分析を行った．分析の結果，ウォーターフォール型開発について環境成熟度が高い傾向にあること，アジャイル開発については多様であることが明らかとなった．また，アジャイル開発の阻害要因として，次の要因が最も大きな比重を占めることがわかった．

- 受託契約締結の難しさ．
- 見積り方法の認知度の低さ．
- 顧客のアジャイル開発に対する認知度の低さ．

また，組織としてアジャイル開発を推奨しているか否かが，売上比率と深く関係することが認められた．この結果をもとに，本研究ではソフトウェアエンジニアリングの見地から，以下の 2 点に焦点をあて，テーマとした．

- アジャイル開発に適用できる実用的な見積り方式の有効性を確認し，提案する．
- アジャイル開発を受託開発に適用できる契約モデルを提案する．

第二に，その主たる阻害要因のひとつである，開発における見積り方式を検討した．ソフトウェア開発において，外部委託するためには，見積り精度が重要となる．一方で，アジャイル開発に新規に取り組む企業は，過去実績が存在せず，見積りの実施が難しいという課題がある．そこで，過去実績値が無くとも効果的な見積りを可能とするサンプリング見積り方式に注目し，実際のプロジェクトデータを用いて，この手法の有効性を検証した．さらに，アジャイル開発プロジェクトの実績データを用いることにより，過去実績からの回帰モデルによりこの見積り方式が有効性であることを確認した．そして，この見積り方式をアジャイル開発においてもプロジェクト初期段階で有効な全体の概算見積りがで

きる方式として提案した.

第三として，サンプリング見積り方式を用いて実際に受託契約が締結できるかの有用性を現行の契約様式にあてはめ確認し，提案した．手順としては，経済産業省および情報処理推進機構のモデル契約書の考え方，特徴を確認し，アジャイル開発において，受託契約が可能な契約モデルを考察し，それを提案した．また，それが実務レベルで利用できるかを検証するため，現在使われている「システム開発委任契約」の様式をもとに適用の可能性を確認した.

本研究では，日本においてアジャイル開発が普及していくことを目標に，アジャイル開発の適用阻害要因を調査し，一括受託開発契約に適用する場合の課題である見積り方式をとりあげ，実務的な見積り方式と，その契約モデルを提案した.

アジャイル開発はいまだわが国では適用率が低いが，今後は，アジャイル開発の定量的プロジェクトデータを基にした更なる調査研究とその分析結果から，日本の現状に適するより精度の高いモデルの考察を進めていきたい.

目次

第 1 章　緒言

企業の情報システムは，ホストコンピューターによる集中処理の時代からクライアントサーバー分散処理を経て，現在ではネットワークコンピューティングにその処理形態が多様化している．そしてユーザーには，開発期間を短縮し早期に業務に適用することを目的に，ERP（Enterprise Resource Planning）の様なパッケージソリューションの適用やクラウドコンピューティングに代表される汎用型アプリケーションを利用するといった選択肢も増えている．しかし，日本の場合，特殊な要件の無い場合においても，汎用的なアプリケーションを利用するのではなく，顧客の注文から独自に作成するカスタムメードのシステム構築の需要が高い割合にある[IPA 2010].

ソフトウェア開発モデルにはプロセスや環境条件により，多様なモデルが存在する．日本では，それらの中で，ウォーターフォール型開発[W. Royce 1970]が，最も多く利用されている[IPA 2010]．一方，欧米でのソフトウェア開発適用モデルはウォーターフォール型開発からアジャイル型開発にシフトしている．2009 年の第 3 四半期に実施した[Versionone 2010]の調査では，約 45%の企業が，50%以上の開発プロジェクトでアジャイル型開発を採用しているという結果が示されている．一方，日本では，アジャイル開発が注目されているものの，未だアジャイル開発を含む反復型開発を採用したプロジェクトは全体の僅か 2.8%である[Nishida 2009]．この要因として，欧米では，情報システム構築を自社で内製化する企業が多く，開発を外注する企業が少ないこと，一方日本では，一括請負契約での開発が多く，早期に要件や開発コストを確定しておきたい動機が存在することが考えられる[Umemoto 2012]．開発中の環境変化に柔軟かつ迅速な対応が必要なプロジェクトさえ，一括請負契約として進めたい場合も想定される．その様なプロジェクトにアジャイル開発を適用するといった適用推進が，本研究の目的である．

本研究では，アジャイル開発における，日本での適用率を低下させる原因を「阻害要因」と定義し，その調査を実施する．そしてその結果のなかで，ソフトウェアエンジニアリングの見地から見積り方式，プロジェクト契約モデルについて研究・分析を実施し，それらの有効な方式・モデルを提案する．

アジャイル開発におけるこれら 3 つの研究の位置づけを図 1-1 に示す.

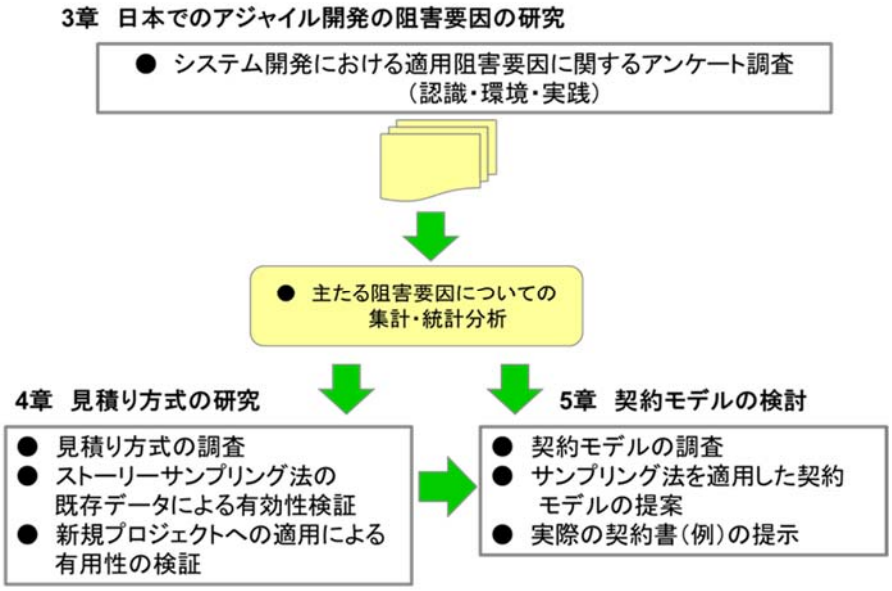

図 1-1　研究各章(3 章～5 章)の位置づけ

本論文は，以下に示すとおり 6 章で構成されている．

第 2 章では，アジャイル開発を含むソフトウェア開発の見積りに関わる先行研究をレビューすることで，本研究との立場と視点の違いを明確にする．

第 3 章では，先行研究から仮説を設定し，それに基づくアンケート調査をシステム開発に関わる IT 企業組織を対象として実施する．その集計・分析結果から，日本におけるアジャイル開発モデルの適用における阻害要因を明確にする．調査結果では，「顧客の認知度の低さ」「外注のしにくさ」「見積り手法が認知されていない」「プロジェクト経験者が少ない」が主たる要因であった．そのなかで本研究では，ソフトウェアエンジニアリングの視点から「見積り手法」「契約モデル」（外注のしにくさ）に注目する．

第 4 章では，日本におけるアジャイル開発の予算・実績データから，アジャイル開発での工数を求める見積りプロセスを明確化する．そして過去の実績データが存在しない場合においても，要件の規模見積りから効果的な工数見積りを可能とするサンプリング見積り方式の有用性を，誤差算出方式により検証する．また，統計分析から定量的にその見積りプロセスの有効性を検証する．

第 5 章では，経済産業省および情報処理推進機構のモデル契約書を概観し，その特徴を確認した上で，第 4 章での研究結果に基づいた見積り方式を適用することで，アジャイル開発の初期段階でのコスト見積りを実現する契約書を提案する．

第 6 章では，本研究の成果をまとめると共に，今後の取り組みについて述べる．

なお，本研究では，「アジャイルソフトウェア開発」を「アジャイル開発」と記載している．また，第 3 章を中心にウォーターフォール型開発と対比し議論している箇所については，「アジャイル型開発」とした．

第 2 章　ソフトウェア開発の現状と課題

本章では，本論文で取り上げる 3 つのアジャイル開発に関わる研究テーマについて関連研究の調査を行う．はじめに 2.1 節ではソフトウェア開発モデルのなかでウォーターフォールモデルとスパイラルモデル，そしてアジャイルモデルを取り上げ，それぞれの開発思想と特徴を明確にする．次に，2.2 節では，第 3 章で取り上げる「日本でのアジャイル開発モデルの適用における阻害要因の研究」に関するソフトウェア開発モデル適用率の研究を確認する．そして 2.3 節において，第 4 章で取り上げる「ストーリーによるサンプリング見積り方式の提案」に関するソフトウェア開発見積りと，定量的分析を伴うソフトウェア開発についてまとめる．2.4 節では，第 5 章で取り上げる「アジャイル開発のプロジェクト契約モデルの検討」に関連するソフトウェア契約モデルについて米国と我が国での現状を確認する．そして本研究での新しい視点を明確にする．最後にまとめとして，2.5 節でこれまで行なってきた先行研究の立場を整理し本研究の視点との違いを明確にする．

2.1　ソフトウェア開発モデル

2.1.1　ウォーターフォールモデル

ソフトウェアをどのような手順で開発するかについては，1960 年代から多くの議論がなされてきた．本論文で取り上げるアジャイル開発もその議論の流れの中にある．そこで，代表的な開発モデルとして，ウォーターフォールモデル，スパイラルモデルをとりあげ，そのうえでアジャイルモデルを確認する．

はじめに，ウォーターフォールモデルは，1970 年，W.W. Royce によって発表されたモデルである [W. Royce 1970]．このモデルは，図 2-1 のように，ソフトウェアの開発工程を何段階かに分割し，前工程の結果を次工程の入力として，順次作業を進めていく．あたかも，滝が段階的に落ちていく様をイメージしている．その後，多様なバリエーションを生みながら広く受け入れられ，現在でも多くのプロジェクトで採用されている．

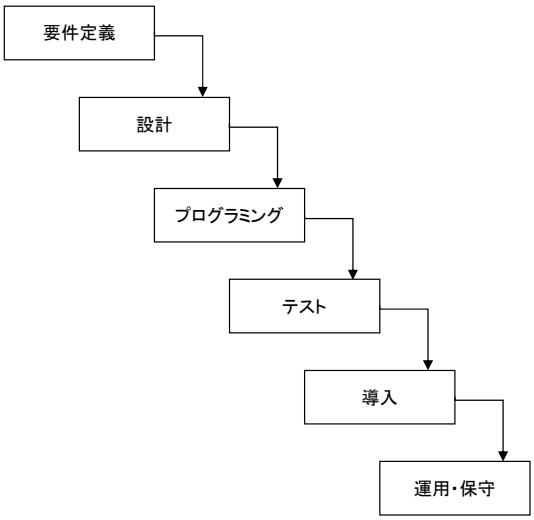

図 2-1　ウォーターフォールモデルの例

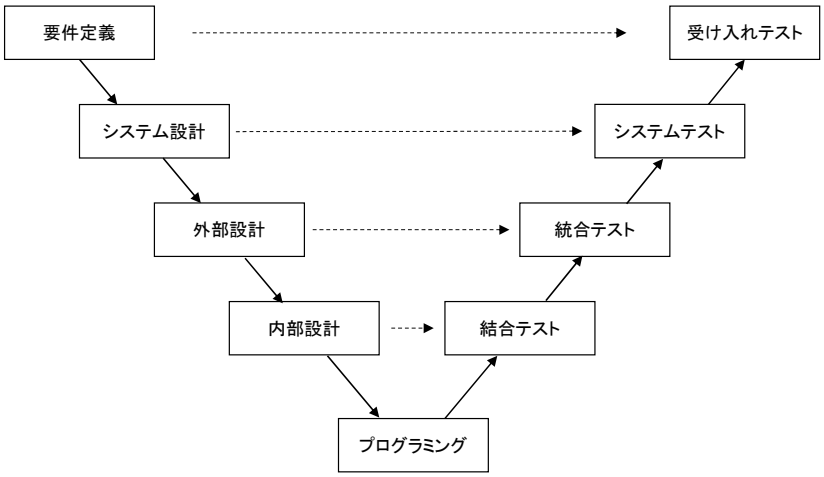

図 2-2　V 字モデルの例

図 2-2 は，V 字モデルと呼ばれ，ウォーターフォールモデルの表記を V 字型にしたものである．この図の例では，システム設計の成果物は，外部設計のインプットになることを表現しており，同時に，最終的なテストフェーズであるシステムテストのインプットになることを示している．このように，ウォーターフォールモデルは，単に上から下に流れているのではなく，設計のフェーズではユーザーに近いところから始まり，プログラミングに至り，テスト工程では単体のプログラムのテストから徐々に結合され，ユーザー要件に対する受け入れテストに戻っていくことを示している．

ウォーターフォールモデルは，一般的に広く受け入れられているものの，以下のような課題が指摘されている．

[ウォーターフォールモデルの課題]

- 要件を工程の早い段階で固定し，順次，次工程に進んでゆく．そのため，後から要件を変更することは，大きな手戻りを発生させてしまう．
- 要件は，ユーザーの意見を聞きながら作成されるが，実際に開発されたシステムを操作し，希望通りであるかを確認できるのは，システムテスト付近となる．そのため，誤解が生じたまま工程が進み，当初期待していたものと異なっているかもしれないというリスクを抱えたまま，開発を行うことになる．
- 開発範囲を早い段階で固定する．納期・費用も固定されるが，開発範囲の固定が優先される傾向にある．

2.1.2 スパイラルモデル

次にスパイラルモデルについて確認する．前述のウォーターフォールモデルは，図 2-2 を見て分かるように，要求定義から，設計，実装，テストと進んでゆく．そのため，要件を確認するテストは，工程の一番後ろ寄りになる．このことから，ユーザー要件をユーザーが確認できるのは，プロジェクトの最終段階であり，ユーザーが求めている要件を満たしていないかもしれないというリスクを抱えながら開発が進んでゆく．また，技術的課題に関しても，実際に実装をし，テストを行うまで確認がとれない．それらの課題点を克服するために，スパイラル型の開発モデルが 1986 年，C.W.Boehm によって提案された[Boehm 1986].

スパイラルモデルでは，計画や要件定義，設計といった開発の初期段階から，実際に「モノ」を作り，リスクを軽減させながら，開発工程を進める[Nunokawa 1997]．各工程で，プロトタイピングを行い，要求・設計・制作・評価のプロセスを繰り返す形態をとることから，後戻りを前提としないウォーターフォール型に対して，スパイラル型と言われる[Nunokawa 1997].

なお，スパイラルモデルにも多様なバリエーションが存在し，広い意味では，小さなウォーターフォールを繰り返す反復型開発も，スパイラルモデルに含まれる．また，ウォーターフォール型をはじめ，従来の開発手法では，各工程（フェーズ）で達成すべき成果物や品質基準が定められ，その基準を達成することによって次のフェーズに進む．スパイラルモデルにおいても，達成すべき品質を定め，フェーズを定義する場合がある．それ以外にも，場合によっては期間を優先し，スパイラルを回していくという考え方がある．これ

を，タイムボックス型[Boehm 1986]のプロセスモデルと呼ぶ場合がある．

後述するアジャイル開発モデルは，通常，タイムボックスの考え方を採用している．

2.1.3 アジャイル開発モデル

アジャイル開発の歴史は 1990 年代半ばに，「重量ソフトウェア開発手法」への反対運動の一部から発展して形成されてきたところに遡る．重量開発手法は，ウォーターフォール開発モデルを適用した場合に多くみられる．厳格な規律と統制を有する官僚的な特徴があり，そのためソフトウェア技術者が効果的に作業を進めるという観点と矛盾していたことが批判の根拠であった．今日で言うアジャイル開発手法は，以前は「軽量ソフトウェア開発手法」と呼ばれており，2001 年に「アジャイルソフトウェア開発手法」と呼称を変える．同年，アジャイル開発手法の先駆者 17 人がアメリカ合衆国ユタ州のスノーバードに会し，それぞれ別個に提唱していた開発手法の重要な部分を統合することを議論し,「アジャイルソフトウェア開発宣言」(Manifesto for Agile Software Development) という文書にまとめている[K. Beck 1999]．以下はその価値として表明された 4 行である．

- プロセスやツールより人と人同士の相互作用を重視する．
- 包括的なドキュメントより動作するソフトウェアを重視する．
- 契約上の交渉より顧客との協調を重視する．
- 計画に従うことより変化に対応することを重視する．

米国 CMU/SEI (Carnegie Mellon University Software Engineering Institute) ではアジャイル開発を，「ステークホルダーの変更要求を満たす，費用効率が高くタイムリーなソフトウェア構築手法で，自律的組織のチームによって協働作業方法で実行される反復型インクリメンタルソフトウェア開発アプローチ．」と定義している[Carnegie-Mellon 2010]．本研究では，アジャイル開発を，ウォーターフォールモデルと同様に，ソフトウェア開発プロセスモデル（Software Development Process Models）のひとつと位置づける．その他の開発プロセスモデルとしては，RAD(Rapid Application Development)，SLCP(Software LifeCycle Process)などがある[Satoh 2013]．アジャイル開発の手法（方法論）としては，Extreme Programming (XP)，Feature Driven Development (FDD)，SCRUM，Adaptive Software Development (ASD)などがあり，本論文では，「ソフトウェア開発手法」という呼び名で混

同されがちな構造化手法，オブジェクト指向，データ中心アプローチ等はあえて「方法論」として区別した．SCRUM(スクラム)は 1986 年に開発された．スクラムは，スプリント（イテレーション）と呼ばれる約 4 週間で測られる反復周期単位に基づいており，これを 1 回～数回の反復でリリースすることで，発注者のニーズに柔軟に対応しつつもより迅速に製品を納品することを実現する．XP はそれを基に 1996 年に開発され，その評価を確立した[K. Beck 1999]．図 2-3 にソフトウェア開発モデルとアジャイル開発手法を表示する．

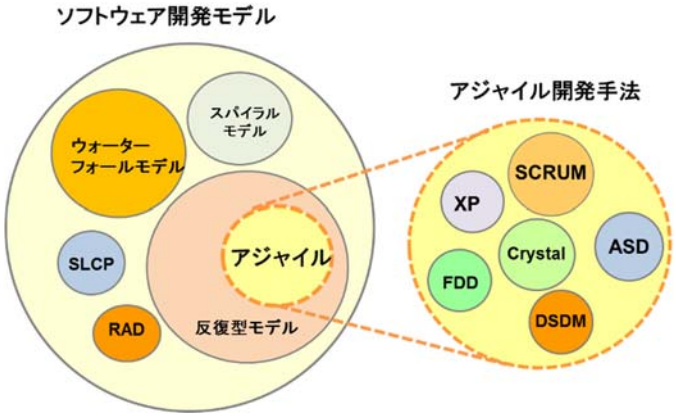

図 2-3　ソフトウェア開発モデルとアジャイル開発手法

アジャイル開発では，図 2-4 に示すとおりウォーターフォール型開発のように開発範囲が固定された不変要素としてプロジェクト開始時に決定し，納期，費用はプロジェクト開発の過程で変動するのではなく，納期，費用は概ね確定している．アジャイル開発では，開発範囲は開発過程で環境変化や顧客満足などにより柔軟に変更され，初期要件と変更要件がトレードオフされ進められる．この開発スタイルそのものがアジャイル開発の特徴といえる．

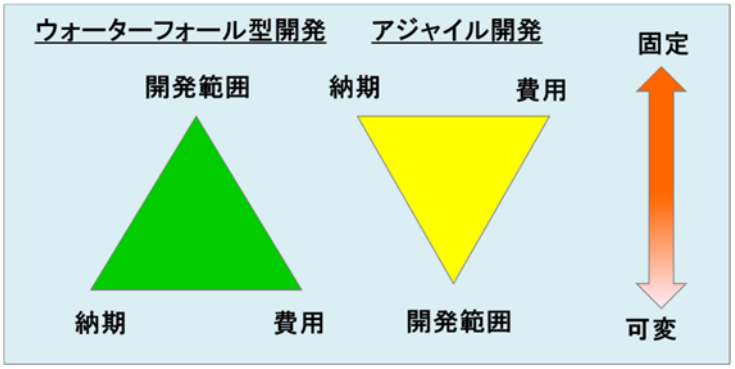

図 2-4　ウォーターフォール型開発とアジャイル開発の違い

2.2　アジャイル開発モデルの適用における阻害要因

2.2.1　ソフトウェア開発モデルに関する調査

第 3 章で取り上げる「アジャイル開発モデルの適用における阻害要因の研究」に関連して，ソフトウェア開発方法についてアンケート調査を行った研究を確認する．その上で，本研究での新規性を明確にする．

はじめに本項では，ソフトウェア開発プロセスモデルの適用比率に関する調査研究について確認する．マイケル・A・クスマノは，要件定義や設計を最初に確定しない開発方式は，欧米やインドにおいて普及していると述べている[M. Cusumano 2003]．この開発方式の適用率は，インドの 70%，欧州の 86%，米国の 55%であること，他方，日本では開発プロジェクトの 53%（30 件中 16 件）でウォーターフォールモデルが採用されていることが記されている．しかし，このデータは 10 年以上前のものであること，「要件定義や設計を最初に確定しない開発方式」の定義が曖昧なこと，調査手法が明示されていないこと等の課題がある．

[Ito 2009]は，静岡県内の企業を対象に実施したソフトウェア開発に関する調査で，開発方法論の認知（認知度），使用（使用実績），効果（効果の認識）に関するデータを紹介している．アジャイル型開発方法論は，認知（14.8%），使用（4.1%），効果（8.2%），ウォーターフォールモデルは，認知（57.4%），使用（50.0%），効果（23.8%）としている． ここでは，調査対象が静岡県内 IT 企業と限定しており，かつ調査アンケート集計結果を提示するにとどまっている．

[Ohsugi 2006]は，ソフトウェア開発プロセスモデルのみならず，ソフトウェア開発技術を API，Web 関連技術，プログラム言語，プロジェクト開発技法，見積り関連技法に分類

し，それぞれに対する興味の度合いを産官学でアンケート調査している．そのうちアジャイル開発については企業のソフトウェア技術者，ソフトウェア工学の研究者および大学生が同程度に強い興味を示したとしている．この研究では産官学に共通する，興味深い技術分野への関心の傾向や課題を分析している．

2.2.2 アジャイル開発モデルに関する調査

前項で，マイケル・A・クスマノがアジャイル開発を含む「要件定義や設計を最初に確定しない開発方式」の普及率調査を実施していることを紹介した[M. Cusumano 2003].

日本では情報処理推進機構が，2011 年に，非ウォーターフォール型開発 WG（ワーキング・グループ）の活動報告として，海外（主として米国）との開発環境の比較から，米国におけるアジャイル開発の普及要因として以下の 3 点を指摘している[IPA 2012].

表2-1　日米の開発環境要因 [IPA 2012]

	海外（主に米国）	日本
1	同一組織（企業）内でのソフトウェア開発が多い.	顧客と開発チームの間に受託開発が多い. （受託契約が必要）
2	オフショアであっても顧客と開発チームがコミュニケーションをとりあえる環境を作っている.	契約の壁があったり物理的に離れていたりでコミュニケーションがとりにくい.
3	ユーザー企業にIT 技術者が多い. （内製化されている）	IT技術者はSIerやソフトハウスに所属している.

この調査では，海外との開発環境の比較で，アジャイル開発の普及要因は内製化に起因するところが多く,言い換えれば,日本での普及を阻害する要因は受託開発が中心の環境，さらに労働者派遣，または受託契約によるコミュニケーションの難しさに原因があるとみている．しかし，アジャイル開発で一括受託開発が難しいか否かは，ここでは明確にされていない．さらに，この調査には，以下の課題があると考えられる．

① 調査対象データの抽出基準が不明であること.

② データは集計に留まり統計分析していないこと.

③ 阻害要因に関わる調査がないこと.

そこで，今日，システム開発に携わる組織がアジャイル開発に取り組む上で抱える阻害要因を調査すべく本研究の実施に至った.

2.2.3　アジャイル開発モデルの適用における阻害要因調査のまとめ

以上，確認したソフトウェア開発プロセスの調査およびアジャイル開発モデルの調査と本研究における新たな視点を表 2-2 にまとめる．

表 2-2　阻害要因に関する文献サーベイ結果まとめ

	分野	研究テーマ	研究の視点
2.2.1	ソフトウェア開発プロセスの調査	・日本のソフトウェア産業の国際競争力に関する一考察 [Takahashi 2010] ・Business of Software [M. Cusumano 2004]	・日本のソフトウェア開発方法論の適用率 ・諸外国のソフトウェア開発方法論の適用率
		・ソフトウェア開発における工学的技術導入に関する考察 [Ito 2009]	・静岡県ソフトウェア開発企業へのアンケート調査結果
		・産官学連携における参加者の興味についての対応分析 [Ohsugi 2006]	・産官学によるソフトウェア技術についての興味に関する分析
2.2.2	アジャイル開発方法の調査	・非ウォーターフォール型開発の普及要因と適用領域の拡大に関する調査 [IPA 2012]	・非ウォーターフォール開発における普及要因の調査
	本研究	・アジャイル開発の適用の阻害要因の調査	＜新たな視点＞ ・適用阻害要因の調査実施 ・日本のIT企業システム開発組織を対象に調査 ・t検定，相関分析，重回帰分析による統計分析

ソフトウェア開発プロセス適用率については，海外（米国），日本ともに，調査実績がある[M. Cusumano 2003][Takahashi 2010]．また，アジャイル開発を含む非ウォーターフォール型開発における普及要因についても，日本において米国を中心としたアジャイル開発普及率の高い国に調査を実施している[IPA 2012]．ただ，阻害要因については調査されていない．本研究では，日本の IT 企業のシステム開発組織を対象とし，アジャイル開発の適用阻害要因を調査する．

2.3　見積りプロセスに関する研究調査

2.3.1　ソフトウェア開発の見積り手法の現状

ソフトウェア開発のプロセスまたはウォーターフォール型も含めたプロジェクトの見積りに関する現状を確認し，さらにアジャイル開発における見積り手法の現状と課題点を明確にする．

本項では，ソフトウェア開発の見積りを規模見積り，工数見積りに分け，それらの現状と課題点を明確にする．

(1) 規模見積り

ソフトウェア開発の規模を見積もるためには，初めに，開発されるソフトウェアの規模を見積もることが重要になる．規模を表す代表的な尺度として，ソースコード数（LOC: Lines of Code）とファンクションポイント(FP: Function Point)がある．LOC は，開発するソフトウェアのソースコードの行数を数えるものであることから，ある程度正確な見積りが行えるのはコーディングを行う開発工程以降である．なお，LOC については SLOC(Source Lines of Code)と表記された文献もあるが，同義のため，本論文では LOC に統一表記した．

ファンクションポイントは，Albrecht によって，1979 年に考案されたソフトウェアの規模を測定する尺度である[Albrecht 1979]．ファンクションポイント法は，ソフトウェアが持つ機能数や複雑さにより重みづけした点数（ファンクションポイント：FP）をつけ，そのソフトウェアにおける合計点数から開発工数を見積もる手法である[Hatsuda 2013]．FP 法は，①作り方（開発言語・方法論など）に左右されにくい，②ユーザー側とベンダー側の双方で見積り結果を共有できる，③見積りからプロジェクト完了まで測定値に一貫性がある，といった特長がある．しかし，正確な規模見積りを求める「計測法」が可能になるのはデータベース設計が完了し，互いに関係する複数の処理をまとめた単位であるトランザクションが明確になる段階である．概算・試算レベルでの「推測」も，限られたデータフロー図 (Data Flow Diagram) や画面数・帳票数をもとに見積りは可能だが，それぞれ IFPUG(International Function Point Users Group)法，NESMA（Netherlands Software Metrics Association）法といった見積り方式の知識習得が必要になる[Hatsuda 2013] ．つまり，FP 法では，提案依頼書（RFP: Request For Proposal）が提示されているのみで，実装すべき機能仕様が詳細に定まっていないような状況で見積りを行う場合は, FP を予測するのが難しいという課題がある．これは FP を正確に予測するためには，画面等のユーザーインターフェイスを介するデータ入出力処理や，論理データモデルがある程度明確になっている必要があるためである[Kurashige 2006]．FP 法での早期見積り技法の研究[Kurashige 2006] [Kurashige 2007]においても FP を計測できるのは基本設計が完了した詳細設計の開始段階であることは変わらない．

他にも FP 法の初期段階における効果的見積りモデルの研究はみられるが，対象プロジェクトの特殊性に基づく限られたプロジェクトへの適用研究に留まっている[Kaneko 2004][Hosotani 2008][Kimura 2009][Niwa 2005].

さらに，オブジェクト指向方法論で UML（Unified Model Language）が登場し，当該方

法論ではユースケースポイント法での見積りが一般的になっている. ユースケースは, RUP（Rational Unified Process）をはじめとするオブジェクト指向開発手法において，開発のごく初期段階で着手するアクティビティとして定着している[Kodama 2004]. ユースケースポイント法は，関係する複数のユースケースを重みづけし，直接ソフトウェア規模を見積もれる見積り手法として知られている．上流工程のモデルを前提にして見積りを行うため，開発の初期段階で計測できるメリットがあるが，計測するユースケースの粒度に厳密なルールがないため，計測結果が異なるデメリットがある[Hatsuda 2006].

ユースケースポイント法を用いた見積りの研究には，大学における PBL（Project Based Learning）でのプロジェクト実績による見積りの評価[Naito 2010]やテストケース数による規模見積り[Shimanaka 2004]，および，計測ツールによる規模見積り[Matsukawa 2004]の研究がある．これらはいずれもオブジェクト指向方法論を用いたものであり，プログラミング言語には Java，C++，または PHP（Hypertext Preprocessor）が用いられている．

(2) 工数見積り

工数見積り技法には, 過去の事例や経験から全体工数を算出する「類推（トップダウン）見積り」と，数式モデルを使って算出する「係数モデル見積り」，プロジェクトで実施すべき作業を洗い出して各工数を積み上げる「ボトムアップ見積り」がある[Hatsuda 2013].

係数モデルについては, 数式法の代表的例として Putnam モデル[Putnam 1978], COCOMO Ⅱモデル[Boehm 2007]があげられる.

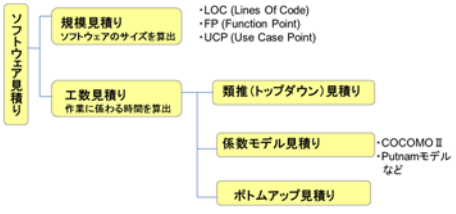

図 2-5 代表的な見積り手法

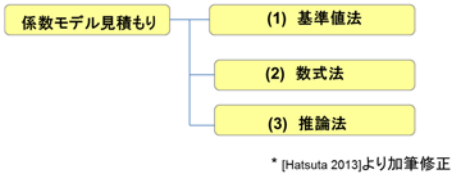

図 2-6 係数モデル見積りの種類

Putnam モデルは Putnam が Norden/Reylight のマンパワー理論式を活用して提案したモデルである[Yoshizaki 2012]. Putnam モデルの特徴は，他のモデルが工数予測を主体としているのに対して，適切な要員配置を示す「マンパワー曲線」をモデル化した点にある．要員

数は次式で求められる．

$$M(t) = \frac{2Kt}{td^2} \exp\left[-\frac{t^2}{2td^2}\right] \quad \text{・・・(式 2-1)}$$

ここで，M(t)は時刻 t における要員数，K は開発プロジェクト総工数，td は要員が最大になる時刻を表す．生産性を「P＝ 総 LOC/総工数」で表し，投入コストおよび要員計画から総 LOC を計算する[Ohfude 1992]．これは，要員配分モデルとして使えるものでもあり，規模見積りの基準は LOC である．

COCOMOⅡは，Barry Boehm により考案されたモデルで，原型は COCOMO81 である．COCOMOⅡは，クライアント・サーバーシステムや Web システムを対象としたスパイラル型開発，パッケージソフトを利用した開発などに対応するために考案され，以下の 3 種類のモデルから作られている．

① 開発初期のプロトタイプ用「アプリケーション組立モデル」

② アーキテクチャー決定用「初期設計モデル」

③ アーキテクチャー決定後開発開始時用の「ポストアーキテクチャーモデル」

このうち①は，開発初期にリスクを軽減する為に行う規模見積りで，オブジェクト・ポイント法 [Aoki 2003] [Hatsuda 2013]をベースとしている．②, ③は規模をスケール要因（E）でべき乗し，その結果に 2.94 と複数のコスト要因（EM）を乗算する以下の数式で工数を算出する．

$$\text{工数} = 2.94 \times \text{規模}^{E} \times EM_1 \times \cdots \times EM_{17} \quad \text{・・・(式 2-2)}$$

COCOMOⅡでは最初に規模算出から始めるが，規模については FP 数か LOC を用いるものの FP 数を使う場合は変換テーブルによって一旦 LOC に変換する [Hatsuda 2006]．つまり，COCOMOⅡは Web システムを対象とした開発手法に適用できるとされているが，LOC が基準であるためアジャイル開発のような規模見積りとして要件（ストーリーポイント：SP）を基準とする開発には適切であるとは言い難い．

コスト見積りに関わる研究分野では，CoBRA 法に基づく見積りモデルが，研究されており，工数予想モデルは以下の定式で示される[Makuta 2008].

Effort＝α×(Size)×(1+∑CO)　　・・・（式 2-3）

CO は Cost Overhead の略記であり,開発コストを低減させる要因は考慮せず,全ての変動要因を Overhead，すなわちコストを増加させる要因として定義する点に特徴がある．規模は FP または LOC での検証はされているがユースケースポイント法（UCP）など他の方法では検証されていない．

また見積りプロセスそのものについては，パーソナルソフトウェアプロセス[W. Humphrey 2009]においてその手順が検討されているが，生産性，工数の基準は LOC である．プロセスのトレーニングの視点での研究は日本でも紹介されている[Murakoshi 2009].

2.3.2　実績データによる定量的分析の現状

本項では，ソフトウェア開発プロジェクトでどのように実績データが使われ，それを見積りのためにどのように分析しているかを確認する．本研究の視点が日本のソフトウェア開発にあるため,日本でのプロジェクトデータの扱い方についての文献を中心に確認した.

ソフトウェア開発プロジェクトのデータは，ほとんどの場合，多岐にわたるデータ項目を実際の開発現場から人手で収集しなければならない点で，①データを収集しにくく，②欠損データが多い．また，③データの精度が低く，④基準が不明確な主観的データも多い[Furuyama 1994]．[Furuyama 2005]は，プロジェクトデータの特徴と統計手法の利用方法について説明している．[Fujinuki 2006]は，ソフトウェア開発プロジェクトで工数を適切に見積もるために，生産性データを蓄積，分析して次回の研究開発に利活用していくことの重要性とその生産性データの活用を推進するアプローチについて紹介している．また，[Kanzaki 2006]は，実績データ（Java による 100KLOC のデータがベース）を利用した自社のプロジェクト管理知識体系をモデルとする標準見積りプロセスを紹介している.

ソフトウェア開発データを使った研究には，品質およびリスクに関する研究がある[Hosokawa 2004][Hosokawa 2005][Kaneko 2004]. また企業における開発プロジェクトの規模と生産性の関連やプロジェクト管理に反映した研究も見られる[Hattori 2006][Murakoshi 2009].

2.3.3 アジャイル開発における見積り

ソフトウェア開発の見積りに関わる研究を前項で述べたが，本項ではアジャイル開発に関わる研究の現状を説明する．

[Boehm 2003]では，適応的つまりアジャイルな開発手法と計画重視の開発手法のどちらを選択すべきか，という問題に対する指針として，適応的開発，計画重視開発それぞれの得意分野を大きく分類し，表 2-3 のとおりとしている．

表 2-3　アジャイルな開発手法と計画重視の開発手法の得意分野

アジャイルな開発手法	計画重視の開発手法
・クリティカルではないシステム （顧客の業務に重大な支障をきたす可能性がなく、人命に関わらないシステム） ・熟練した開発者が参加するシステム ・開発中に頻繁に要件が変わるシステム ・開発者が少ない場合 ・混沌とした状況に意欲を持って取り組む組織文化	・クリティカルなシステム （顧客の業務に重大な支障をきたす可能性がある、あるいは人命に関わるシステム） ・経験の少ない開発者が多い場合 ・開発者が多い場合 ・秩序を重視する組織文化

（[Boehm 2003] より加筆修正．）

「アジャイルな開発手法」は，開発中に頻繁に要件が変わるようなシステムを熟練開発者が少ない人数で行う分野を得意とし，「計画重視の開発手法」は，クリティカルなシステムを多くの開発者で行う分野が得意とされる．また，[Boehm 2003]では，この様なそれぞれの得意分野の指針に照らして，これから取り組むプロジェクトを分析することで，主にそのリスクからアジャイル開発か計画重視開発かを選択する検討材料とすることができるとしている．

[Carnegie-Mellon 2010]のアジャイルに関わる研究資料は DOD（Department of Defense）向けの購買に関わる考察[FAR 2005]やアーキテクチャー，ドキュメンテーションに関わる研究レポート，ホワイトペーパーが中心である．日本のプロジェクトマネジメントへの適用の研究である[Kaneko 2009]は，アジャイル開発の留意点を PMBOK の知識エリアに準拠してとらえた研究であるが，見積りをはじめ手法およびその定量的評価は行っていない．

[Ichiyanagi 2006]では定量的計測があるもののリスクに焦点を絞っており，かつ取り上げた3つの反復型開発プロジェクトは2～3年がかりの大型プロジェクトの開発部分に反復を取り入れたもので「アジャイル開発」の定義とは異なる．

[Adachi 2011]では，アジャイル開発オフショアプロジェクトを独自メトリクスで評価し

ているが，品質管理にフォーカスしたもので，時間，コスト，見積りの分野は含まれていない．

2.3.4 アジャイル開発の見積りプロセスと実績データ分析の現状

[W. Humphrey 2009]は，ソフトウェア見積りプロセスについてのデータ採集の手順を「パーソナルソフトウェアプロセス」にまとめている．ただし，アジャイル開発に特化したものではないため規模計測の単位がソースコード行数（LOC）であるなど，規模を単位としないアジャイル開発に対しては適用が難しいところも多い．

アジャイルの見積り単位としては，ストーリーポイントでの規模見積りが基準とされているが[M. Cohn 2005]，工数はベロシティ（Velocity）と呼ばれる生産性を基準としており，アジャイル開発における実績と生産性が既に存在する前提での記述となっている．

アジャイル開発のサンプル調査[S. Keaveney 2006]では COCOMO モデルは使われておらず，40%が FP 法を使用していたと記されており，コスト見積りでは実績値から推測値を利用した調査結果を提示している．

また見積りのアルゴリズムについての研究[S. Bhalerao 2009][S. Bhalerao 2011]でもストーリーポイントによる規模見積りとベロシティを基にした方程式を紹介しており，これも実績値が必須となっている．これからアジャイル開発に取り組もうとする企業が適用するのは困難である．

定量的データ分析を含む学術論文では 87 件のアジャイル開発データのストーリーポイントとベロシティから所要期間を割り出すアルゴリズムが紹介されている[S. Bhalerao 2009][S. Bhalerao 2011]．ただし，これらのプロジェクトデータの採集項目が不明であり，イテレーション回数の決め方，工数に関わる基準値（平均作業時間など）の明記がないため，コスト算出に繋がるプロジェクト工数の算出には使用できない．また，プロジェクトデータの分類と説明が欠損しているため，プロジェクトの業態，技術的特徴も不明確である．日本でもプロジェクトアジリティ計測が研究されたが[Ikoma 2008]，アジャイル開発ではない．

2.3.5 アジャイル開発実績データ分析と考察のまとめ

ソフトウェア開発見積り，ソフトウェア開発のデータ分析，アジャイル開発，アジャイル開発の見積りとデータ分析と，本研究での新たな視点を以下の表に示す．

表 2-4. アジャイル開発見積りに関する文献サーベイ結果まとめ

	分野	研究テーマ	研究の視点
2.3.1	ソフトウェア開発見積り	・ソフトウェア開発における機能改良見積もり法 [Hatsuda 2006] ・業務ソフトウェア開発における早期見積技法の適用 [Kurashige 2006]	・FP法による規模・工数・コスト見積方法とプロジェクトでの見積りプロセスとタイミング ・FP法での早期見積方法の考案と適用の検証
		・ユースケースから導いたテストケース数による規模見積りに関する研究 [Shimanaka 2004]	・UCP法でテストケースによる規模見積方法の提案と検証
		・CoBRA法に基づくソフトウェア開発プロジェクトの見積りモデル構築手順の改善 [Makuta 2008]	・CoBRA法による見積手法と検証
2.3.2	ソフトウェア開発のデータ分析	・プロジェクトデータ分析の指針と分析事例 [Furuyama 2005]	・データの特徴と統計手法の利用方法
		・ソフトウェア開発プロジェクトにおける生産性データ活用についての一考察 [Fujinuki 2006]	・生産性データの見積りへの利用方法と検証
		・実績データの活用による見積りプロセスの改善 [Kanzaki 2006]	・Java言語でのプロジェクトデータによる見積りプロセスと利用方法の紹介
2.3.3	アジャイル開発	・アジャイル開発におけるプロジェクトマネジメント [Kaneko 2009]	・プロジェクトマネジメントPMBOKの適用方法
		・アジャイル開発における品質管理の取り組みと評価 [Adachi 2011]	・ウォーターフォール型モデルでの品質管理のアジャイル開発での適用
2.3.4	アジャイル開発の見積りとデータ分析	・Cost Estimation in Agile Development Projects [S.Keaveney 2006] ・Agile Estimation using CAEA [S.Bhalerao 2011]	・実績（経験）データにからの見積方法 ・ストーリーポイント（規模見積）とベロシティ（生産性）から期間を算出するアルゴリズムを実プロジェクトで検証
	本研究	・アジャイル開発の見積り方式の提案	＜新たな視点＞ ・実績値の有無別工数見積りプロセスの比較。 ・実績値の無いプロジェクトでのサンプル抽出見積プロセスの実査。 ・ストーリーサンプリング見積り方式の実績データによる統計的検証

ソフトウェア開発見積りについては，ファンクションポイント法，ユースケースポイント法ならびに CoBRA 法などの研究論文がある[Hatsuda 2006][Shimanaka 2004][Makuta 2008]．ソフトウェア開発で Java 言語でのプロジェクト実績データを活用した見積り方法をテーマとする文献もある[Kanzaki 2006]．海外ではアジャイル開発に関する見積りをテーマとする研究も行われている[S.Bhalerao 2011]．本研究ではストーリーによるサンプリング見積り方式による見積りという新しい方式に焦点をあてる．アジャイル開発での過去実績がない場合，実績データを使用する場合それぞれに調査を実施し，統計的に確認することを新たな視点とする．

2.4 ソフトウェア開発の契約モデルの調査

2.4.1 米国政府における契約種類

ソフトウェアモデルと米国での契約モデルの現状，および我が国での現状を調べる．本

項では，米国政府調達における契約規約から，米国でのシステム開発調達における契約の種類とその特徴をまとめ，わが国の契約との共通点，相違点を確認したうえで，アジャイル開発により適切なモデル契約を検討する.今回のモデル契約の請負契約,準委任契約は，日本の契約類型による区分である．米国では政府が調達における契約種類として発注者，受注者のリスク，責任，コスト・報酬の配分の点などから 13 の種類に区分している．以下の表 2-5 にその種類と特徴を示す.

表 2-5　米国政府調達における契約の種類と特徴

契約タイプ	契約種類	特徴	リスク	
			発注者	受注者
定額型契約 ・SOWの基に計画通りに作業実施 ・リスクは受注者側	完全定額契約(FFP)	標準的定額型契約. SOWを定額で作業実施		○
	経済価格調整を含む定額契約(FPw/EPA)	賃金の変動など価格調整を含む定額契約		○
	再調整可能な定額契約(FP-R)	受注者の現在の業績で今後の価格調整可能		○
	インセンティブ付定額契約(FPI)	コスト超過の場合は受注者負担　未満の場合は発注者と受注者が契約時比率で分配		○
	報奨付定額契約(FPAF)	性能品質により追加報酬を受ける		○
実費償還型契約 ・コスト上昇のリスクは発注者が負う	確定報酬付実費償還契約（CPFF）	契約価格は確定報酬と受注者のコスト	○	
	報酬付実費償還型契約	予定コストと実際のコストの差分を分割負担	○	△
	成功報酬付実費償還型契約	CPFFに加えて成功報酬を受ける	○	
	Cost Contract	受注者は報酬ではなくコストのみを受け取る	○	
	Cost Sharing	受注者はコストの一部を受け取る	○	
間接作業型契約 ・受注者に作業の完遂をもとめない	T&M（Time & Material Level-of-Effort Term)	必要なスキルと作業時間が定義され各スキルと時間当たりの固定料金に対して価格設定	○	
	FFP-LET(Fixed Price Level of Effort Term)	一定期間（短期間）、一定作業に対する定額料金契約小規模R&Dプロジェクトなどに適用	○	
	Cost Plus Fee Term	FFP-LETに報酬が加わる	○	

*○：有り　　△：場合によって有り　*“SOW”Statement Of Work

[FAR 2005]より抜粋・加筆修正

日本のITプロジェクト契約では，請負または委任（準委任）あるいは派遣契約が主たる契約形態であり，ここに挙げた米国政府の調達における契約形態に直接対応付けることは現実的でない．しかしながら，発注者と受注者の状況にあわせたより適切な契約を締結するための参考になる.

米国政府調達における契約種類についての論文[Ohkubo 2002][Nishiyama 2004][Nozawa 1999]では，その特徴が分類されているが，日本の開発プロジェクト契約への具体的提案はない.

2.4.2　日本における契約の種類とモデル

情報システム開発等の委託に係る契約類型としては日本では民法で定められた契約種類として，請負・委任・労働者派遣がある．民法において，準委任は『委任』の規定を準用するとされており，法的効力は同等であるので，このうち本項では，請負，準委任によるシステム開発契約を検討する．両者はその目的の違いから，次のとおり定義される[Yoshida 2001].

・請負は，定められた仕事を完成させることを目的とする契約である．

・準委任は，代理人を頼むような法律行為以外の事務の遂行を目的とする契約である．

経済産業省，情報システムの信頼性向上のための取引慣行・契約に関する研究会では，この定義をもとに[METI 2012]が検討され，公表されている．そこではモデル契約の前提とする開発手法はウォーターフォール型であり，開発のフェーズ毎にユーザー・ベンダーの責任分担を契約書において詳細に規定し，契約を締結するように述べている[METI 2012].

（１）モデル契約の前提

① 契約当事者：対等に交渉力のあるユーザー・ベンダーを想定

(例)委託者（ユーザー）：民間大手企業，受託者（ベンダー）：情報サービス企業

② 開発手法：ウォーターフォールモデル

③ 対象システム：重要インフラ・企業基幹システムの受託開発，保守・運用

④ プロセス：共通フレーム 2007 による標準化されたシステム企画・開発・運用・保守プロセスによる．

一括発注の場合に加えて，マルチベンダー形態，工程分割発注に対応．つまり，アジャイル開発を含む反復型開発，パッケージ活用型，中小企業ユーザーとの取引等上記の前提と異なる場合の活用については，必ずしも適応しない．

（２）準委任と請負の相違

通常，委任契約は弁護士などの特定業務の場合のみ使われ，情報システムコンサルタント，アドバイザー等の専門サービスは準委任とされる．

① 仕事の完成義務：請負ではベンダーは仕事（受託業務）の完成の義務を負うのに対し，準委任ではベンダーは善良な管理者の注意をもって委任事務を処理する義務（善管注意義務）を負うものの，仕事の完成についての義務は負わない．

② 瑕疵担保責任：請負では，仕事の完成義務を負うので，例えばユーザーに完成されたものとして引き渡された成果物に瑕疵があった場合，無過失責任としての瑕疵担保責

任を負う．また，瑕疵により契約の目的を達成することができないときは契約を解除することができる．これに対して，準委任の場合，仕事の完成義務はないので，請負のような瑕疵担保責任を負うことはない．ただし，事務処理に関して善管注意義務違反があった場合には，債務不履行責任（例えば不完全な履行を完全なものにすることや損害賠償責任など）を負うこととなる．

[METI 2012]での理想的なユーザーとベンダーの役割分担としては，ユーザーが企画段階において業務要件およびシステム要件（外部設計に対するインプット）を主体的に決定・明確化し，その上で開発段階において，ベンダーが主体となって業務全体に対する利害関係者の要件のうち，システムに関する部分（システム要件）についての仕様化を行い，また，外部設計がユーザーによる承認を受けた後の要件追加，仕様変更，未決事項等は変更管理手続に則り，委託料・納期等の協議を実施する事が望ましいとされる．

つまり，システム開発工程では，要求定義（システム調査分析，外部仕様の決定），システム基本設計については準委任契約であり，それ以外の工程（システム詳細設計，プログラム設計，プログラミング，テスト）は請負ととらえられる[Yoshida 2001]．

また，モデル契約に関わる学術論文としては，日本のソフトウェア産業では，どのようなシステム開発においてどのような契約形態が望ましいか，ゲーム理論と契約理論により検討されている[Moriya 2009]．ここでは，前述の米国政府調達における契約において適用されているインセンティブ契約等の適用検討も含まれている．

2.4.3 アジャイル開発における契約モデル

情報処理推進機構ソフトウェア・エンジニアリング・センターでは「アジャイル開発向け契約モデル案」[Umemoto 2012]を公表している．

アジャイル開発の手法では，一般に初期計画時の見積りを実施するプロジェクトはあるものの，実際はイテレーションの都度行っているケースも多い．開発の中で変更を受け入れていくアジャイル開発の特質から契約はリリース単位毎に行い，受託側が完了責任を持たない準委任型が推奨されている[Umemoto 2012]．

従来のアジャイル開発のために検討・推奨された契約モデルでは，繰り返されるイテレーション群をそのリリース単位に個別契約を締結する形をとる．そこでは開発開始前の基本契約の他に，通常のリリース単位毎に開発内容と費用がある程度確定した段階で順次締結していくことが提案されている．各リリースでは，要件定義・開発・テストがイテレー

ションとして複数回反復される．

この契約モデルでは，プロジェクト全体に共通する事項を定めた基本契約を締結した上で，個別の機能内容について協議を行い，開発対象が確定し次第，（例えば 1 回にリリースされる開発対象機能群をひとまとめにして）順次，個別契約（請負契約あるいは準委任契約）を締結するとされている[Umemoto 2012].

2.4.4　契約モデルのまとめ

日本のソフトウェアプロジェクトの契約モデル，日本のアジャイル開発契約モデル，米国政府の調達における契約モデルと本研究での新たな視点を以下の表に示す．

表 2-6　契約モデルに関する研究の文献サーベイ結果のまとめ

	分野	研究テーマ	研究の視点
2.4.1	米国政府における契約モデル	・米国政府調達における契約種類とその実際[FAR 2005]	・調達行為に関わる種類の契約形態と対象取引
2.4.2	日本における契約の種類とモデル	・情報システムの信頼性向上のための取引慣行・契約に関する研究会 [METI 2012] ・ソフトウェア産業における契約について [Moriya 2009]	・ウォーターフォール型開発でのプロジェクトフェーズ毎のユーザ、ベンダの役割分担と主たる契約形態（請負または準委任） ・どのようなシステム開発においてどのような契約形態が望ましいかのゲーム理論と契約理論による位置づけ
2.4.3	日本のアジャイル開発における契約モデル	・アジャイル開発向けモデル案について [Umemoto 2012]	・アジャイルプロジェクトライフサイクルの中での契約形態と契約時期
	本研究	・アジャイル開発の契約モデルの提案	＜新たな視点＞ ・計画時のコスト見積りの実施 ・プロジェクト終盤の追加・変更開発のサービス契約化

海外諸国の契約事例については海外（米国）で Federal Acquisition Regulation がソフトウェアを含む調達に関する契約体系を公表している[FAR 2005]．日本のソフトウェア開発プロジェクトの契約モデルについては，経済産業省がモデル取引契約をまとめている[METI 2012]．また，アジャイル開発向け契約モデルについては，情報処理開発機構のソフトウェア・エンジニアリング・センターでの活動として，契約モデル（案）が紹介されている[Umemoto 2012].

2.5 文献サーベイのまとめと研究課題

2.5.1 文献サーベイのまとめ

アジャイル開発の見積りプロセスの検討では，ソフトウェア開発に関わる見積りの研究を規模見積り，工数見積り，コスト見積りの分野に分け調査した．そこでは FP 法に関するものが多く，特に初期段階での見積もり技法[Kurashige 2006][Kurashige 2007]，有効性向上[Hosotani 2008]，また，特殊プロジェクトでの適用の工夫[Kurashige 2006][Kaneko 2004][Kurashige 2007]について研究されている．

また，ユースケースポイント法での見積りについての論文では，見積り精度のプロジェクトでの検証[Naito 2010]，テストケースから規模見積りをする手法[Shimanaka 2004]など見積り精度に関わる研究が進められている．いずれもアジャイル開発に特化したものではない．工数の係数モデル見積りでは，早期見積りの係数モデルを使った検証[Ohfude 1992]や CoBRA 法の有効性検証[Makuta 2008]の研究があるが，これらも規模見積りはファンクションポイント法または LOC（Lines Of Code）である．日本のアジャイル開発の研究では，そのモデルにウォーターフォール型のプロジェクトマネジメントの適用を検討したもの[Kaneko 2009][Ichiyanagi 2006]，また品質管理に関する研究[Adachi 2011]はあるものの，見積り分野では見当たらない．

海外文献としてはいくつかアジャイル開発の見積りに関わる文献がみられる．アジャイル開発の実績値からコスト見積りを推測する研究[S. Keaveney 2006]，アジャイル開発プロジェクトでの所要時間を，生産性と規模見積りから計算するツールの検証[S. Bhalerao 2009][S. Bhalerao 2011]がある．いずれもアジャイル開発の見積りに関わる研究であり，生産性・実績データを必要とするが，分析でのデータがどの様に採集されたかを説明していない．加えて，アジャイル開発実績データによる定量的分析の分野においても，ソフトウェア開発プロジェクトの定量分析の研究について調査した．

プロジェクトデータ分析の難しさと統計手法の利用法[Furuyama 2005]，次プロジェクトに利用できる生産性を求めるためのデータを常に構築する必要性とそのアプローチ[Fujinuki 2006]，また実績データを利用した見積りプロセス[Kanzaki 2006]では，量的データからの生産性算出やその利用モデルを論じている．いずれも，アジャイル開発データを使ったものはない．アジャイル開発の研究に焦点を絞ると，先に述べた海外論文のアジャイル開発プロジェクトでの所要時間を，生産性と規模見積りから計算するツールの検証[S. Bhalerao 2009][S. Bhalerao 2011]がここでも対象範疇となるが，データ採集項目や工数に関

わる基準値が不明確でありデータの分類説明も欠落しているため，信頼性が得られていない．

アジャイル開発の契約モデルの提案では，まず日本のソフトウェア開発プロジェクトが準拠する契約モデルを経済産業省が[METI 2012]として公表している．そこで対象とする開発手法はウォーターフォールモデルであり，アジャイルを含む反復型モデルにはかならずしも適応していない．次にアジャイル開発契約モデルについては情報処理推進機構ソフトウェア・エンジニアリング・センターの[Umemoto 2012]を参考にした．どちらも請負契約・準委任契約をどのフェーズまたは単位に対していつ締結するかを考案している．

さらに，米国政府の調達における契約については，特徴を分類しまとめた論文[Ohkubo 2002][Nishiyama 2004]がある．それらの原本である政府調達規約 Federal Acquisition Regulations(FAR)は最新規約が 2005 年度にまとめられており[FAR 2005]，今回はそれを参考にしている．

2.5.2 問題点の総括と研究の視点

以上の調査を通じて得た課題点を以下のとおり設定し，それを解決するための研究を行った．

(1) アジャイル開発適用の日本での阻害要因

ソフトウェア開発方式の適用率の調査研究は海外，国内ともに実施されている．

アジャイル開発については，非ウォーターフォールモデルのひとつとして，主に普及要因について情報処理推進機構で調査されている．

第 3 章では，アジャイル開発の阻害要因について，システム開発組織への調査と統計分析からその明確化を行う．

(2) アジャイル開発の見積りプロセスの検討

日本のソフトウェア開発の研究分野でアジャイルに関わるものは少ない．理由として，データに基づく分析が必要だが現状ではそれが少ないことが考えられる．本研究では，見積りプロセスのみならず，採集データ項目とそのタイミング，および分析結果を明確にする．

アジャイル開発の定量的データをともなう海外の文献はあるが，そのデータの信頼性とプロジェクトデータの分類（業態，環境，技術レベルなど）が不明瞭である．

第 4 章の研究では，データ採集した上で統計分析により，それぞれの特徴と関係性をとら

える．さらに，アジャイル開発をこれから始める，まだ実績を持たない企業がデータを構築する手順と，初期の段階で「実績データ無しで正確な見積りができる手法」を提案し検証する．

(3) アジャイル開発の契約モデルの提案

現在公開されている日本のアジャイル開発契約モデル（案）は，アジャイル開発を推進する上では，契約開始時に全体のコストが見えず，各リリース単位における複数反復分の各フェーズの見積りはウォーターフォール型開発の見積りと同程度の時間と労力が必要である．それに適するモデル契約はまだみられない．第 5 章では，アジャイル開発のマニュフェストに準拠し，計画時の全体見積りを使った契約モデルを提案する．さらに，リリース後にサービスレベル契約による運用・保守に追加開発を含むモデルを提案する．

(4) 総括

文献サーベイから得た本研究の視点を表 2-7 にまとめる．

表 2-7　総括と研究の視点

アジャイル開発適用の日本での阻害要因						
研究テーマ	高橋	Cusumano	伊東	大杉他	IPA/SEC	本研究
ソフトウェア開発方式適用率（日本）	○					
ソフトウェア開発方式適用率（海外）		○				
ソフトウェア開発方式の使用・認知・効果の測定（静岡）			○	○		
ソフトウェア開発技術の興味の分類				○		
非ウォーターフォール型開発に関する調査					○	
＜新しい視点＞ 日本でのアジャイル開発適用の阻害要因の調査						◎

アジャイル開発の見積り方法								
研究テーマ	初田・島中・幕田	古山	藤貫	神崎	金子・一柳	Keaveney	Bhalerao	本研究
ソフトウェア見積法の改善案と評価	○							
プロジェクトデータ分析		○						
生産性データの活用モデル			○					
実績データの見積適用				○				
アジャイル・反復のPM適用					○			
アジャイル見積方法の定性的評価						○		
アジャイル見積方法の提案と評価							○	○
＜新しい視点＞ アジャイルの見積プロセスと新規ユーザへの適用モデル 実績データによるストーリーサンプリング方式の提案								◎

アジャイル開発の契約モデル				
研究テーマ	METI・森谷	IPA/SEC	西山・大久保	本研究
ソフトウェア開発の契約種類・時期の提案	○			
アジャイル開発の契約種類・時期の提案		○		○
米国政府の調達における契約種類の研究			○	
＜新しい視点＞ アジャイル開発の計画時コスト見積を含む契約の提案 リリース後の追加変更のサービス契約モデルの提案				◎

第 3 章　アジャイル開発モデルの適用における阻害要因の研究

「日本ではアジャイル開発が普及していないという実態」を前章の先行研究で確認した．本章では，アジャイル開発に関する適用阻害要因を調査することで「日本のシステム開発でアジャイル開発への移行を阻害する要因は何か」を確認する．そこで得られた調査結果に基づき，その解決策を第 4 章，第 5 章で検討，提案する．

3.1　研究のアプローチ

はじめに，阻害要因に関する仮説を設定した．仮説は先行研究[IPA 2012]と，実際のシステム開発に係わる技術者の方々のインタビュー結果から立案した．そして，それを確認するためにアンケート調査を実施した．図 3-1 に研究の全体像を示す．

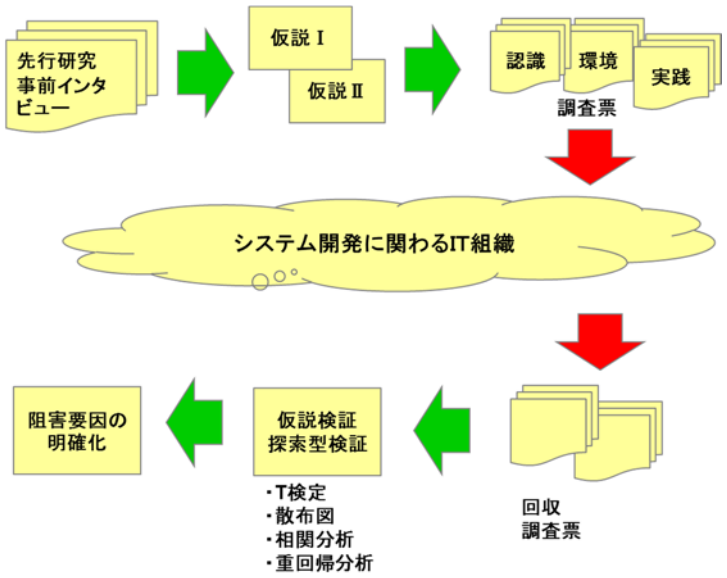

図 3-1　アジャイル開発モデルの適用における阻害要因

(1)　仮説立案

まず，アンケートで確認したいアジャイル開発の普及を妨げる理由を，次の立場から検討した．

Ⅰ．開発（受注）側の理由：開発環境状況（プロセス成熟度）に関する仮説

Ⅱ．顧客（発注）側の理由：ソフトウェア開発に関する顧客の認識の仮説

(2)　ステークホルダー関係構造図によるアンケート調査対象の明確化

次にステークホルダー関係構造図を基に，仮説を証明するためにアンケート対象を選択した． プロジェクトに関わる主要なステークホルダーを顧客（発注側），開発組織（受注側），受注者，外部委託先とし，オフショアは対象外とした．その関係を図 3-2 に示す．

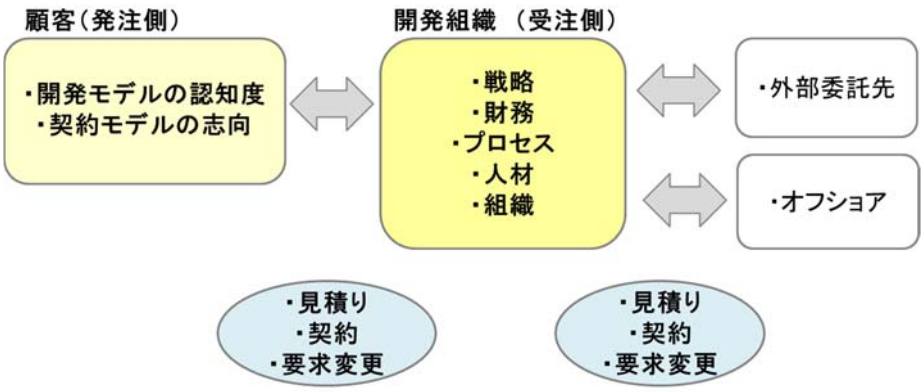

図 3-2　ステークホルダー関係構成図

(3) 質問のカテゴリーと質問項目

質問のカテゴリーと質問項目は先行研究を参考にした[Shibao 2004][W. Humphrey 2009]. そして環境，認識，実践の 3 つのカテゴリーに分けた 41 項目の質問項目から構成されるアンケートを作成した．アンケートは，対象として選択したシステム開発に関連する企業組織に対して実施した．

今日，日本で使われているソフトウェア開発プロセスにおいて，ウォーターフォール型開発が主流であるという事実は，[IPA2010]の調査により明らかになっている．アジャイル開発については，内製で実施している組織，受託開発でも取り入れている組織と適用するソフトウェア開発モデルは組織により多様である．そのため調査は，適用しているソフトウェア開発プロセスに関わらずソフトウェア開発を営む IT 企業全般を対象とした．また，調査の対象は組織であるため同一組織の人には配布せず，組織単位に配布した．

3.2　仮説設定

3.2.1　開発環境状況（プロセス成熟度）に関する仮説

仮説設定については，先行研究と調査対象として参加頂いたシステム開発を担当される技術者，プロジェクト管理者の意見から導いている．

現在，システム開発を営む IT 企業は，既にアジャイル型開発に取り組んでいる企業，ソフ

トウェア開発プロセスはウォーターフォール型のみを用いている企業，これからアジャイル型開発を導入すべく準備を進めている企業と多様である．

日本の企業ではウォーターフォール型を主たる開発モデルとしているところが多いと予想され，ウォーターフォール型開発の組織成熟度は高い傾向にあると考えられる．さらに，ウォーターフォール型開発に慣れ親しみ，高い環境成熟度を持つ組織は，アジャイル型開発に移行するメリットを見いだせず，アジャイル型開発については環境成熟度が低い傾向があることが予想される．一方，アジャイル型開発の成熟度については，アジャイル型開発のみを採用している企業も含まれることや，アジャイル型への取り組みにはレベル差があることから，高低分布のばらつきが予想される．さらにアジャイル型開発を積極的に推進している組織は，ウォーターフォール型開発の環境は未整備であり，アジャイル型開発への移行の垣根が低い企業であると予想される．そこで，仮説Ⅰを以下のように設定した．

仮説Ⅰ：ウォーターフォール型開発に慣れ親しみ，成熟度が高い企業は，アジャイル型開発に関する環境成熟度が低く，移行に躊躇する．

3.2.2 ソフトウェア開発に関する顧客認識の仮説

一括受託契約締結の際には，何を(Application Feature)，いつまでに（Delivery Date），いくらで (Cost based Price)といった情報が必須である．要件を確定し，それを基準に納期，コストを見積もる一括受託契約に慣れ親しできた顧客に，いくらで（工数）を固定し，何を（要件），いつまでに（納期），が流動的になるアジャイル型開発の基本的考え方が不安要因となり，適用の阻害要因になっている可能性が考えられる．そこで，以下のとおり仮説Ⅱを設定した．

仮説Ⅱ：契約タイプに関して，顧客は準委任契約より一括受託契約を望む傾向にある．

3.3 調査票設計

3.3.1 調査項目の分類

調査項目は仮説の検証を考慮し，仮説Ⅰに対する「環境編」，仮説Ⅱに対する「認識編」に分けて質問項目を設定した．さらに，探索的に阻害要因を分析するため，開発の現状を聞く「実践編」を準備した．

分類区分はバランススコアカード[Kaplan 1997]と「能力成熟度モデル統合（CMMI）」[W. Humphrey 1991]を参考に設定した．各編と分類区分の対応表を表3-1に示す．

表3-1　質問項目マッピング表

分類区分	環境編	認識編	実践編
顧客		顧客 （プロセス、受託契約、変更）	
戦略	戦略 （戦略、目標、仕組み）		推進計画 （アジャイルの推進、外部オフショア推進） 開発比率 （件数、売上、外部委託、オフショア）
財務 （含む契約・見積り）		財務 （見積り、売上、収益）	見積り方法 （対顧客、対外部委託、対オフショア） 契約形態 （契約形態、外部契約形態、オフショア契約形態） 委託形態 （実施有無、外部委託、規模）
プロセス	プロセス （プロセス、プラクティス、判断基準）		
組織	組織 （リソース配分、PMO）	組織 （方針、評価、PM）	
人材	人材 （教育、人事評価）	人材 （確保、理解、経験）	
その他		その他 （大規模、外注、オフショア）	

バランススコアカード[Kaplan 1997]は企業の業績評価システムである．従来の財務的指標中心の業績管理手法の欠点を補うものであり，戦略・ビジョンを4つの視点（財務の視点・顧客の視点・業務プロセスの視点・学習と成長の視点）で分類し評価する．

能力成熟度モデル統合（CMMI）[W. Humphrey 1991]は，プロセスの評価や改善をすすめるための枠組みで，組織がプロセスをより適切に管理できるようになることを目的として遵守するべき指針を体系化したものである．CMMIでは，段階表現と連続表現の2つの利用方法がある．段階表現は，組織の実施プロセスを評価し，レベル1からレベル5までの5段階に成熟度を評価する．代表的な評価対象領域としては，ソフトウェア開発，システム開発，プロジェクト管理，リスク管理等がある．連続表現は，個々のプロセス領域のプロセス能力を測定するために用いる．これらがIT企業のソフトウェア開発に関するプロセス成熟度と，企業の戦略，ビジョンがソフトウェア開発モデルを選定する上で大きく関わると考え，これらの分類区分を質問設計の基準とした．

3.3.2　質問項目

次に，環境編，認識編，実践編について述べる．環境編4要素についてはアジャイル型開発，ウォーターフォール型開発それぞれ10問，認識編5要素については各15問，さらに，実践編では委託，契約，見積りの3要素について各8問，開発比率，推進計画の2要素はアジャ

イル型開発に限定して8問の設問を定義した.

環境編は，主に能力成熟度モデル統合を参考にした．レベル1から5までの成熟度評価のレベル特性に準拠して全ての設問を5択で回答するように設計した.アジャイル型開発とウォーターフォール型開発について自組織をA.戦略，B.プロセス，C.組織，D.人材まで，レベル1からレベル5までの5段階の成熟度レベルで評価し回答頂いた.

表3-2 環境編　システム開発の組織環境成熟度に関するご質問

※各設問の答えとして該当する( )の番号をアジャイル開発、及びその他システム開発に対して各1つ選択の上ご記入ください。

カテゴリー	ご質問項目	ご回答欄	
		アジャイル型開発	ウォーターフォール型開発
A.戦略	1.　システム開発に係わる組織戦略が存在しますか？	1. の回答	
	(1)　存在しない。 (2)　存在するも共有されていない。 (3)　戦略は明確で具体的であるが行動に結びついていない。 (4)　戦略は明確でマネジメント層はその実現に向けて具体的アクションをとっている。 (5)　組織全員が共有された戦略のもとに戦略達成に向かい行動している。	(　)	(　)
	2.　1. の組織戦略を達成するための具体的なプラン(目標を含む)があり、それが実践されていますか。	2. の回答	
	(1)　存在しない。 (2)　存在するが、戦略との連携は曖昧である。 (3)　戦略達成のためのプランがあるが、実践されていない。 (4)　戦略達成のための明確なプランがあるが、実践は部分的である。 (5)　具体的なプランが立案され、達成度が定期的に計測されている。	(　)	(　)
	3. 貴組織では各プロジェクトと組織戦略を結びつけるプロセス(しくみ)が存在しますか。	3. の回答	
	(1)　プロセスは存在せず、プロジェクトの定期的レビューも行われていない。 (2)　プロセスは存在しないが、プロジェクトの定期的レビューがある。 (3)　プロセスは存在するが、戦略との連携が不明瞭である。 (4)　戦略と連携するプロセスが存在し、一部プロジェクトに適用されている。 (5)　プロセスは存在し、全プロジェクトが戦略との整合性をとり評価が実施されている。	(　)	(　)
B.プロセス	4. システム開発プロジェクトについて組織として一貫した開発プロセスがありますか？	4. の回答	
	(1)特にない。 個人の判断で適用している。 (2)(開発・テスト部分など)部分的なプロセスが存在し、ケースバイケースの適用。 (3)(開発・テスト部分など)部分的なプロセスが存在し、全プロジェクトに適用。 (4)一連のプロセスが存在し、特定のプロジェクトがそのプロセスで実行されている。 (5)一連のプロセスが存在し、全てのプロジェクトがそのプロセスを取り入れている。	(　)	(　)
	5. システム開発に係わるベストプラクティス(効率的な手法・技法等)の組織的共有プロセスはありますか？	5. の回答	
	(1)特にない。 個々の所有情報となっている。 (2)プロジェクトによって共有する場合もあるが、プロセスはない。 (3)一部のプラクティスを共有する仕組みがあり、部分的に活用されている。 (4)プラクティスがすべて組織として共有される仕組みがある。 (5)組織的にプロセスが共有されプラクティスは日々活用されている。	(　)	(　)
	6. システム開発プロジェクトの成功・不成功を含めた判断基準はありますか？	6. の回答	
	(1)存在しない。 (2)存在し、一部のプロジェクトに適用されている。 (3)存在し、全ての開発プロジェクトに適用されている。 (4)全ての開発モデルの評価基準が存在するも、特定のプロジェクトがその基準で評価される。 (5)全ての開発モデルの評価基準が存在し、全てのプロジェクトは決められた判断基準で評価される。	(　)	(　)
C. 組織	7. システム開発プロジェクトにむけて人材リソース配分ができる組織になっていますか？	7. の回答	
	(1)常に場当たり的である。 (2)人材リソースの配分のプロセスはあるが運用できていない。 (3)人材リソースの配分のプロセスはあり、一部での運用になっている。 (4)プロセスは存在しており、組織横断の人材リソースの配分がされている。 (5)外注・オフシェアを含めて人材リソースの配分が柔軟に行われるプロセスがある。	(　)	(　)
	8. システム開発プロジェクトをトータルに管理または指導する機能をもつ組織がありますか？	8. の回答	
	(1)存在しない。管理する人も特に決まっていない。 (2)存在しないが、特定の組織(または管理職など)がその機能を兼務している。 (3)存在するが、うまく機能していない。 (4)存在し、特定のプロジェクトを管理している。 (5)全プロジェクトに渡り機能する組織があり、効率的運営に貢献している。	(　)	(　)
D. 人材	9. システム開発プロジェクトに向けたスキル教育プログラムはありますか？	9. の回答	
	(1)存在しない。 OJTのみである。 (2)自社(組織)にはないが、必要な場合は外部ｺｰｽを受講させている。 (3)自社(組織)にはないが、プロジェクト向けに個々に教育を実施している。 (4)社内教育プログラムがあるが、技術スキル教育のみである。 (5)社内にあり、評価者・管理者向けの教育プログラムもある。	(　)	(　)
	10. システム開発プロジェクトでの評価を人事評価に反映するシステムがありますか。	10. の回答	
	(1)特にない。 (2)反映する場合もあるが、システム化されていない。 (3)開発プロジェクトでの評価は全て個人の評価に反映するが、システム化されていない。 (4)システムはあるが、運用は必ずしも全プロジェクトではない。 (5)全てのプロジェクトでの評価を個人の人事評価に常に反映する人事システムが存在する。	(　)	(　)

認識編では，A.顧客，B.人材，C.財務，D.組織，E.その他の分野からシステム開発組織の認識に関する質問に対して5段階評価による回答とし，“Yes”を5，”No”を1，「どちらでもない」を3，その間を2，4として採点依頼した．

表3-3　認識編　システム開発に関する認識についてのご質問

※ 各質問項目に対し、アジャイル型開発とウォーターフォール型開発について
該当するか否かを1点から5点の評価で採点をご記入ください。

カテゴリー	質問項目	アジャイル型開発	ウォーターフォール型開発
A.顧客	(1)顧客の認知度(顧客との共通フレーム)はある。	(　)	(　)
	(2)システム受託契約は締結し易い。	(　)	(　)
	(3)変更に対するコントロールがし易い。	(　)	(　)
B.人材	(4)技術者は常に確保できる。	(　)	(　)
	(5)管理者は開発手法・モデルを理解している。	(　)	(　)
	(6)プロジェクト経験者が豊富である。	(　)	(　)
C.財務	(7)共通な見積り手法は認知されている。	(　)	(　)
	(8)プロジェクトの早期に全体売上が見通せる。	(　)	(　)
	(9)収益算出が容易。	(　)	(　)
D.組織	(10)組織方針として推奨されている。	(　)	(　)
	(11)評価指標がある。	(　)	(　)
	(12)プロジェクトマネジメントがし易い。	(　)	(　)
E.その他	(13)大規模プロジェクトに対応できる。	(　)	(　)
	(14)外注が可能。	(　)	(　)
	(15)オフシェア開発が可能。	(　)	(　)

実践編ではアジャイル型，ウォーターフォール型の共通質問として見積り，契約，委託形態についての質問を用意し全て5択とした．アジャイル型開発に特化した開発比率，推進計画の質問項目では，比率を考慮した10択の質問を設定した．

表3-4 実践編（1）システム開発全般の実践に関するご質問

※各設問の答えとして該当する( )の番号をアジャイル開発及びその他システム開発についてそれぞれ選択の上、
回答欄にご記入ください。ご回答は**複数選択可能**です。**最も多い順に**記入願います。

カテゴリー	質問項目	ご回答欄	
		アジャイル型開発	ウォーターフォール型開発
A.委託形態	1. システム開発ではアウトソース(外部委託)されていますか？	1. の回答	
	(1) していない。全て自社リソースで開発。 (2) 一部アウトソース(国内ベンダ)に委託し実施。 (3) 全部アウトソース(国内ベンダ)に委託し実施。 (4) 一部オフショア(海外ベンダ)に委託し実施。 (5) 全部オフショア(海外ベンダ)に委託し実施。	 (　) (　)	 (　) (　)
	2. システム開発は主にどのようなプロジェクト規模で実践されていますか？	2. の回答	
	注: 小規模＜30人月　30＜＝中規模　＜100　100＜＝大規模　一人月＝160時間 (1) ほとんどが小規模。 (2) ほとんどが中・小規模。 一部自社開発では大規模開発も実施。 (3) ほとんどが中・小規模。一部受託開発で大規模にも適用。 (4) ほとんどが大規模受託開発、一部中・小規模も実施。 (5) その他 (ご説明:)	 (　) (　)	 (　) (　)
B.契約形態	3. 顧客とのシステム開発プロジェクトの契約はどのような形態で締結していますか？	3. の回答	
	(1) 準委任契約 (2) 請負契約 (3) 労働者派遣 (4) プロジェクト毎に異なるサービス契約として締結。 (5) その他 (ご説明:)	 (　) (　) (　) (　)	 (　) (　) (　) (　)
	4. 外部委託先(国内ベンダ)との契約はどのような形態で締結していますか？	4. の回答	
	(1) 準委任契約 (2) 請負契約 (3) 労働者派遣 (4) プロジェクト毎に異なるサービス契約として締結。 (5) その他 (ご説明:)	(　) (　) (　) (　) (　)	(　) (　) (　) (　) (　)
	5. オフショア委託先(海外ベンダ)との契約はどのような形態で締結していますか？	5. の回答	
	(1) 準委任契約 (2) 請負契約 (3) 労働者派遣 (4) プロジェクト毎に異なるサービス契約として締結。 (5) その他 (ご説明:)	(　) (　) (　) (　) (　)	(　) (　) (　) (　) (　)
C.見積方法	6. 受注の際の開発見積りはどの手法を使用していますか？	6. の回答	
	(1) ファンクションポイント(含むCOSMIC) (2) SLOC(ソースライン数) (3) ユースケースポイント法 (4) ストーリーポイント (5) その他 (ご説明:)	(　) (　) (　) (　) (　)	(　) (　) (　) (　) (　)
	7. 外部発注の際の開発見積りはどの手法を使用していますか？	7. の回答	
	(1) ファンクションポイント(含むCOSMIC) (2) SLOC(ソースライン数) (3) ユースケースポイント法 (4) ストーリーポイント (5) その他 (ご説明:)	(　) (　) (　) (　) (　)	(　) (　) (　) (　) (　)
	8. オフショア(海外ベンダ)発注の開発見積りはどの手法を使用していますか？	8. の回答	
	(1) ファンクションポイント(含むCOSMIC) (2) SLOC(ソースライン数) (3) ユースケースポイント法 (4) ストーリーポイント (5) その他 (ご説明:)	(　) (　) (　) (　) (　)	(　) (　) (　) (　) (　)

表3-5 実践編　(2) アジャイル型開発の実践に関するご質問

※各設問の答えとして該当する番号を選択の上、回答欄にご記入ください。
アジャイル開発を実践されていない組織の方もご回答お願いします。
設問 9.,15.,16.のご回答は**複数選択可能**です。**最も多い順に**記入願います。

カテゴリー	質問項目	ご回答欄
D.開発比率	9. 現在、貴組織ではアジャイル開発を実践していますか？ (複数選択可)	9. の回答
	(1)実践していない。 (2)自社システムと自社製品開発にのみ一部実施。 (3)自社システムと自社製品開発の全てに実施。 (4)受託にのみ一部に実施。 (5)受託を100%アジャイルで実施。	(　) (　)
	10. アジャイル開発プロジェクト件数は昨年度(2013年度)全プロジェクト件数の約何%を占めましたか？	10. の回答
	(1) 0%～10%　(6) 51%～60% (2) 11%～20%　(7) 61%～70% (3) 21%～30%　(8) 71%～80% (4) 31%～40%　(9) 81%～90% (5) 41%～50%　(10)91%～100%	(　)
	11. アジャイル開発は昨年度(2013年)全プロジェクト売上の約何%を占めましたか？(自社開発分は除く)	11. の回答
	(1) 0%～10%　(6) 51%～60% (2) 11%～20%　(7) 61%～70% (3) 21%～30%　(8) 71%～80% (4) 31%～40%　(9) 81%～90% (5) 41%～50%　(10)91%～100%	(　)
	12. アジャイル開発プロジェクトの外部委託(国内ベンダ)は昨年度(2013年度)コストの約何%を占めましたか？	12. の回答
	(1) 0%～10%　(6) 51%～60% (2) 11%～20%　(7) 61～70% (3) 21%～30%　(8) 71%～80% (4) 31%～40%　(9) 81%～90% (5) 41%～50%　(10)91%～100%	(　)
	13. アジャイル開発プロジェクトの内オフショア(海外ベンダ)は昨年度(2013年度)コストの何%を占めましたか？	13. の回答
	(1) 0%～10%　(6) 51～60% (2) 11%～20%　(7) 61～70% (3) 21%～30%　(8) 71%～80% (4) 31%～40%　(9) 81%～90% (5) 41%～50%　(10)91%～100%	(　)
E.推進計画	14. アジャイル開発をはじめて何年ですか？	14. の回答
	(1) 0年　(6) 5年未満　(11) 10年未満 (2) 1年未満　(7) 6年未満　(12) 10年以上 (3) 2年未満　(8) 7年未満 (4) 3年未満　(9) 8年未満 (5) 4年未満　(10) 9年未満	(　)
	15. アジャイル開発を今後積極的に推進していく予定ですか。 (複数選択可)	15. の回答
	(1)今のところ推進予定はない。 (2)計画はないが、状況(RFP,リソース等)により個別に対応する。 (3)自社システムと自社開発分は積極的に推進する。 (4)受託開発についても積極的に推進する。 (5)すでに積極的に推進している。	(　) (　)
	16. アジャイル開発を推進する上で、アウトソースをどう活用していく予定ですか。(複数選択可)	16. の回答
	(1)アジャイル開発を推進する予定はない。 (2)アジャイル開発を推進する。 その為に社内リソースを育成する。 (3)アジャイル開発を推進する。 その為にアウトソース(国内ベンダ)を活用する。 (4)アジャイル開発を推進する。 その為にオフショア(海外ベンダ)を活用する。 (5)すでに積極的に推進している。	(　) (　) (　)

3.4　調査の実施と分析

3.4.1　調査の実施

アンケートは，システム開発に係わるシステムインテグレーターとソフトウェアハウスに所属する 43 名（43 組織）に個人的関係を通じて依頼した．さらに，研究団体であるソフトウェア・エンジニアリング・マネジメント・ソサエティ（SEMS）の協力を得て依頼した IT 関係者 109 名，計 152 名に配布した．回収率は，個人的関係者 58%（25 名）研究団体 4%（5 名）平均 20%（30 名）であった．なお，回答者には組織的重複がないことを確認した．

3.4.2　分析

(1)　組織環境成熟度に関する分析

本調査は，IT 企業組織を対象としたものである．ウォーターフォール型開発に関する成熟度の高い企業とそうでない企業によって，アジャイル型開発への取り組みに違いがあるかどうかを確認するため，それら組織の開発モデルによる成熟度をレベルで分類した．組織環境に関する質問群から得られた各組織の数値を「組織環境成熟度」とした．そこで開発モデルの環境成熟度を以下の 4 つの領域に区分した．（図 3-3 参照）．

A:　ウォーターフォール型開発環境の成熟度は低いが，アジャイル型開発の環境成熟度が高い組織．

B:　ウォーターフォール型開発，アジャイル型開発共に環境成熟度が低い組織．

C:　ウォーターフォール型開発，アジャイル型開発共に環境成熟度が高い組織．

D:　ウォーターフォール型開発の環境成熟度が高く，アジャイル型開発の環境成熟度が低い組織．

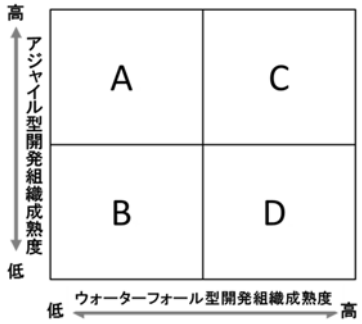

図3-3　組織の開発モデルに対する環境成熟度

環境編の 4 要素（戦略，プロセス，組織，人材）の全項目について，アジャイル型開発，ウォーターフォール型開発に分けて回答値の合計をとり，x 軸を，「ウォーターフォール型開発」，y 軸を「アジャイル型開発」として散布図を作成した．（図 3-4）

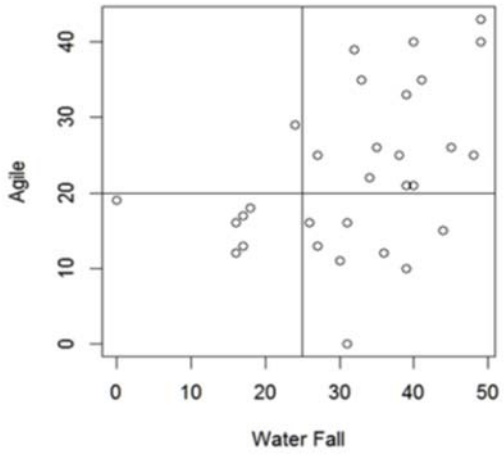

図3-4　環境成熟度　散布図

図3-4のとおり，回答はウォーターフォール型開発の環境成熟度の値が高い領域C，D，に偏り，わずかに領域Bにも分布が見られた．

図 3-4 の分布では，仮説 I 「ウォーターフォール型開発に慣れ親しみ，成熟度が高い企業は，アジャイル型開発に関する環境成熟度が低い.」という現象はみられない．この結果からは「ウォーターフォール型開発の環境成熟度の高い企業がほとんどであり，アジャイル型開発についての環境成熟度はばらつきのある状況である.」ことが観察された．

(2)　認識に関する分析

仮説Ⅱの検証のため，認識編の各項目についてt検定を実施した．その結果を表3-6に示す．平均値はアジャイル型，ウォーターフォール型開発それぞれに回答されたその項目の平均値(mean)である．p値は，その値が有意水準（1%）を下回る場合は，帰無仮説（アジャイル型開発とウォーターフォール型開発では差がない）を棄却することが妥当，つまり「差がある」と判断できる．　今回は，有意水準の値として，1%を基準とした．

表 3-6　認識編項目による t 検定結果

設問	内容	AG 平均	WF 平均	p値	t値	DF 自由度
1	顧客の認知度がある	1.965517	4.275862	1.125e-08	-8.043	27
2	受託契約がし易い	2.034483	4.137931	8.322e-06	-5.484	27
3	変更管理がし易い	3.344828	2.517241	0.03153	2.2682	27
4	技術者確保がし易い	1.896552	3.413793	3.691e-05	-4.9285	27
5	管理者が手法を理解している	2.551724	4.275862	7.421e-06	-5.5269	27
6	プロジェクト経験者が豊富だ	2.137931	4.206897	2.416e-06	-5.9497	27
7	見積り手法が認知されている	1.793103	3.758621	1.98e-07	-6.9141	27
8	早期に売り上げが見通せる	2.206897	3.689655	2.814e-06	-5.8910	27
9	収益算出が容易である	2.241379	3.103448	0.009083	-2.811	27
10	組織方針として推奨されている	2.827586	3.655172	0.04659	-2.0856	27
11	評価指標がある	2.344828	3.655172	0.0007494	-3.8	27
12	プロジェクト管理がし易い	2.379310	3.517241	0.001311	-3.584	27
13	大規模プロジェクトに対応可能	2.344828	4.034483	7.281e-06	-5.5341	27
14	外注が可能	2.379310	4.310345	9.133e-08	-7.2207	27
15	オフショア開発が可能	2.413793	3.620690	0.0003748	-4.0627	27

表3-6では，p値が1%以下の項目が13項目ある中で，”0”に近い上位4項目を見ると，①顧客の認知度がある，②外注が可能，③見積り手法が認知されている，④プロジェクト経験者が豊富，⑤受託契約がし易い，である．

つまり，これらの項目に対して，回答者は「アジャイル型開発とウォーターフォール型開発では差がある．」と認識しているといえる．

このことから，ウォーターフォール型開発を主とする組織がアジャイル型開発に移行するのを阻害する主たる要因は，「契約」，「見積り」およびそれらの背後にある「顧客の認識」である，ということが確認できた．さらに「ウォーターフォール型開発との格差が2番目に大きかったプロジェクト経験者の少なさは,実績の少なさからのリスク回避として,アジャイル型開発を回避する傾向が見られた．

(3)　現状の実施状況による分析

企業のアジャイル型開発の浸透度合いを把握する上で，売上率は不可欠な要素である．またそれらともっとも深く関連すると思われる項目を探ることで，売上率を向上させることも可能と考えられる．

この調査では，アジャイル開発を実践している企業の回答データから，実践編質問項目である売上率について他の項目との相関をとり，重回帰分析を行った．分析手順は以下のとおりである．

① 全体集計データから，アジャイル開発を実践していない企業のデータを外す．
（実践編質問9，14でアジャイル開発を実践していない，はじめていないと回答した標本を対象外とした．）
② 売上比率と全質問項目との相関表を作成し相関係数を得る．
③ そのなかで相関係数が高い方から（今回の場合0.26以上のもの）を複数（3～4）選択する．
④ 売上率を目的変数，③で選んだ項目を説明変数として重回帰分析を行う．
⑤ 偏回帰係数，決定係数から，より相関の強い項目を探索する．

この手順に従い重回帰分析を実施した結果を以下の表に示す．

表3-7 売上比率と関連項目の分析結果

質問項番	相関の高い項目	相関係数	偏回帰係数
認識編(9)	収益算出が容易	0.30	-0.41
認識編(10)	組織方針として推奨されている	0.34	0.36
認識編(12)	プロジェクトマネジメントがし易い	0.37	0.88
認識編(13)	大規模プロジェクトに対応できる	0.26	1.43

切片(Intercept)・・・-2.96 決定係数(Multiple R-squared)・・・0.41

相関表から，「収益算出が容易」，「組織方針として推奨されている」，「プロジェクトマネジメントがし易い」，「大規模プロジェクトに対応できる」が売上率と相関があることが確認されたので，これらを重回帰分析の説明変数とした．重回帰分析の結果，偏回帰係数より，「大規模プロジェクトに対応できること」が，より影響を与えていることが推測できる．決定係数では，0.4程度の意味づけができることが確認された．

3.5 まとめ

本章では，IT企業がアジャイル型開発を適用するにあたってその阻害要因となる要素を，アンケート調査・分析を通じて，仮説型検証，探索型検証により定量的に明確化した．こ

れは，次の段階である阻害要因の解決策の考察のための方向性の検出を目的としている．

環境編の分析結果からは，日本のシステム開発に関わるIT組織では，ウォーターフォール型開発の環境成熟度の高い企業がほとんどであり，アジャイル型開発についての環境成熟度は，ばらつきのある状況であることが散布図より読み取れた．認識編の分析結果からは，アジャイル型開発推進の阻害要因として，受託契約締結の難しさ，並びに見積り方法の認知度の低さ，顧客のアジャイル型開発に対する認知度の低さが大きな比重を占めていることが明確化した．また，重回帰分析による探索型検証では，大規模プロジェクトに対応できることが売上比率と深く関係する傾向がみられた．

第 4 章　ストーリーによるサンプリング見積り方式の提案

システム開発プロジェクトの場合，日本では発注側にプロジェクト管理スキルが保持されず，さらに受託側が仕事の完了責任を負う受託契約で締結せざるおえない場合が多い．そこでは一般的にプロジェクト開始前に予算確保が必要になる．第2章で述べた通り，アジャイル開発は，要求が変動的であるとか，部分的にリリースを実施していくような開発プロジェクトに対して適用できる特徴を持つ．そのためプロジェクト初期段階で作業工数が見積りにくい．

そこで，プロジェクト初期段階であげられる全要件（ストーリー）からサンプル要件を抽出し，その工数を求め，全体に反映する見積り方式に着目した．本稿ではそれを「サンプリング方式」と呼ぶ．

本章では，この方式を，準委任契約で締結すべきと考えられているアジャイル開発が，受託契約で締結可能となる見積り方式として提案する．

4.1　研究のアプローチ

はじめに，アジャイル開発に適用可能な見積り方法を調査し，その問題点を確認した．アジャイル開発の見積り方法として，全タスク分解方式，サンプリング方式，実績方式の3つをとりあげ，その特徴や手順を明確にした．アジャイル開発では，要件に代えて，顧客視点でシステムの機能概要を記述したストーリーを使う[M.Cohn 2005][J. Rasmusson 2010]．ただし，アジャイル開発に慣れ親しんでいないユーザーの多くは要件，特にシステム要件とストーリーを区分し難いため，本章では要件（ストーリー）とした．全タスク分解方式は，全要件（ストーリー）をタスクレベルに分解し，そのタスク一つひとつに対する要員工数を見積り，その合計から総作業工数を求める方式である．従来のウォーターフォール型開発による経験が活かせるため見積りの精度は高い．その反面，見積り負荷は大きくなる．サンプリング方式は，アジャイル開発手法の一つであるスクラムの全要件（ストーリー）の中から一部を抽出し，その抽出した要件（ストーリー）をタスク分解して見積り，係数化し，そこから全体規模を類推する見積り方式である．見積り負荷を軽減する方式として，このサンプリング方式と過去データから類推する実績方式をとりあげる．サンプリング方式の特徴のひとつは，全タスク分解方式同様に，過去実績が必要ないことである．さらに，サンプリングの見積り工数を係数化し他の要件（ストーリー）の工数を見積もるため，見積り負荷の低減方法として有効であるが，全タスク分解方式に比べ，精度上の課

題が予想される．そのため，実際のプロジェクトに適用しそのデータを用いて有用性の検証を行った．また，サンプリング方式で見積りを実施したプロジェクト実績データを積み上げ，そこから回帰式によって新規プロジェクトの工数を類推する実績方式の有効性を検証した．そこではスクラムにおけるストーリーポイントによる見積り，実績の過去データをもとに，回帰式・相関による分析を行った．そしてその結果の有効性から見積り方式を提案する．本章の研究テーマとプロジェクトデータとの関連性を図4-1に示す．

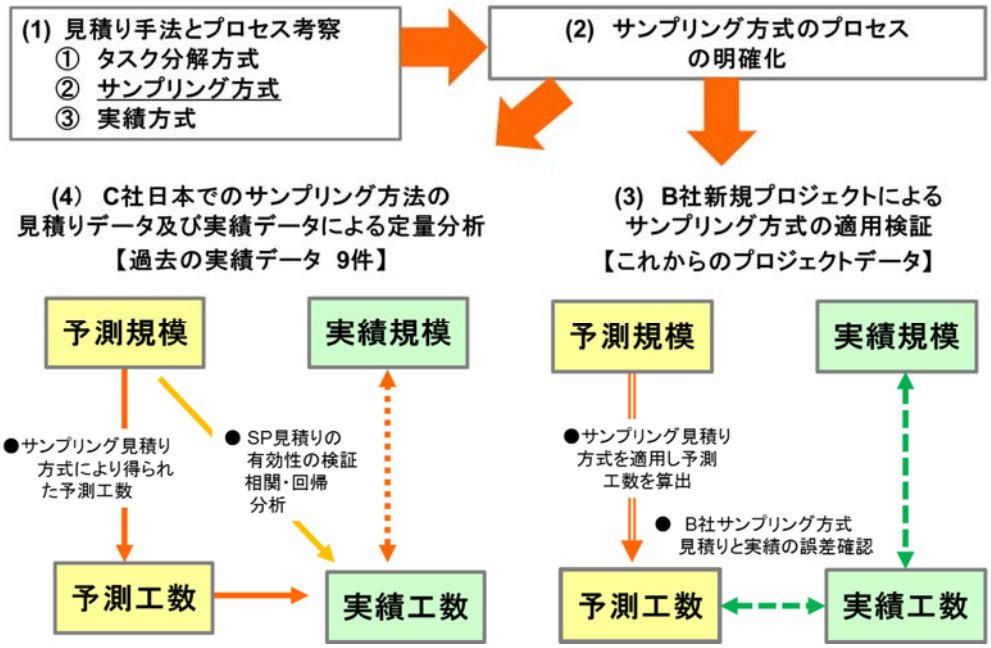

図 4-1　アジャイル開発の見積り方式

4.2　アジャイル開発工程

4.2.1　アジャイル開発の見積りプロセスと手法

アジャイル開発の手法のひとつであるスクラム[J. Sutherland 1995][M. Cohn 2005]では，スプリントと呼ばれる，およそ2~4週間単位に分割された反復周期単位（イテレーション）毎に，開発，テスト，リリースすることで，お客様のニーズに柔軟に対応しながら，迅速に製品を納品することを実現する．1986年に開発されたスクラムは，1993年ソフトウェア開発手法として反復型開発を取り入れたオブジェクト指向分析設計ツールの構築に使われた．当時は，素早い開発が求められており，要求仕様を簡単に動作するコードに変換する方法が求められていた．同じころ，ケン・シュエイバーが自社 (ADM) のソフトウェア開発にこの手法を用いた．これらは産業界での様々なベストプラクティスに基づいており，

それらがソフトウェア開発手法としてのスクラムの元となった．スクラムの開発工程は，「計画ミーティング」，「製品基準の調整・レビュー・配布」，「スプリント」，「スプリントレビュー」，「振り返り」，「クロージャ」6つの活動からなる．このうち，計画ミーティング・スプリント・スプリントレビュー・振り返りが繰り返し行われる．計画ミーティングでは，スプリントで実現させたい製品やサービスの要件リストであるプロダクトバックログを，優先順位を伴って確定させる．チームは，これまでのチームのパーフォーマンス(ベロシティ･前回スプリントで達成したプロダクトバックログの相対見積り合計)からプロダクトバックログの相対見積りを行う．この相対見積りは，ストーリーポイントを使って行われる．ストーリーポイントはスクラムチームが使っている任意の測定基準である.1つのストーリーを実装するのに必要な労力を測るために使われる.優先順位をつけ，パーフォーマンスに則した相対見積りを行う見積り手法は開発中の変更対応によるスケジュール管理への柔軟さが期待できる[J. Sutherland 2013]．スクラムはアジャイル開発で最も多く用いられている手法である[Versionone 2010].

本研究では，スクラムのストーリーポイントを規模指標とする見積り方法に着目する[J. Sutherland 2013]．以後「スプリント」は「イテレーション」に統一する．

4.2.2　アジャイル開発の工数見積り方式

アジャイル開発では，開発前に全体のスコープが見えないことが多くプロジェクト計画時に全体の工数を把握することが難しい.

[M. Corn 2005]は，見積りに利用できる過去実績がない場合，ストーリーポイント (SP)をボトムアップ法でタスク（チケット）に分解しその工数を見積もり，プロジェクト計画時また各イテレーション開始時に利用する方法を紹介している．それでもアジャイル開発では，プロジェクト計画時の全要件(ストーリー)のタスク分解は現実的には難しい．現在もプロジェクト開始時には予算と要員数による概算で各イテレーション開始時にのみ見積りを実行する企業や，全く見積りそのものを実施しない要員依存型の企業もある．計画時および各イテレーション開発前での全体規模・工数見積りの実施もまだ定着していない.

プロジェクト開始時にアジャイル開発で開発領域（スコープ）を明確化し必要な工数を把握するためには次の方式が考えられる.

(1)　全タスク分解方式：計画時の全要件（ストーリー）をボトムアップによりタスク分

解する．それらを積み上げ全体の工数を算出する[M. Cohn 2005].

(2) サンプリング方式 : 全要件（ストーリー）からランダムに 10%～20%のサンプル（標本）を抽出し，それらをタスク分解し，それらから 1 ストーリーポイント（以後 SP）当たりの工数を求めて全体工数を算出（類推）する．

(3) 実績方式 : 過去に実施した複数のプロジェクトの工数実績と SP 実績から 1SP 当たりの工数を求めて全体工数を算出する．

上記方式には，それぞれ適用する上での条件や利点，欠点がある．それらを以下の表 4-1 にまとめる．

表 4-1　アジャイル開発工数見積り方式

	方式	利点	欠点	条件
(1)	全タスク分解方式	・より正確な工数が算出可能 ・過去実績が不要	・見積り時間がかかる	・変更分はデルファイ法等別途考慮が必要
(2)	サンプリング方式	・比較的短時間で見積りができる ・過去実績が不要	・サンプル抽出時の基準（比率等）が未定	・約10%～20%のサンプル抽出が必要
(3)	実績方式	・見積り時間が短い ・変更分を反映した工数見積りが可能	・日本では実績データが少なく，有効性・信頼性が未証明	・数件以上の実績データが必要

数件以上の実績データがあれば「(3) 実績方式」を使った工数見積りはより正確で短い時間で可能だが，日本ではアジャイル開発の過去データを持つユーザーは少ない．

実績データの無い，これからアジャイル開発に取り組もうとする企業には，「(1) 全タスク分解方式」または，「(2) サンプリング方式」となる．迅速性を特徴とするアジャイル開発では，「(1) 全タスク分解方式」の適用は現実的には難しく，有効性を確認することで「(2) サンプリング方式」が現実的であると考えらえる．

4.3　提案手法の検証の枠組み

4.3.1　ストーリーポイントによる規模見積り

迅速さと変更対応を特徴とするアジャイル開発は特に日本のソフトウェア開発では適用し難いという課題がある．なぜならアジャイル開発のストーリー定義はプロジェクト初期のみではなく，各イテレーション開始時に再定義（主にトレードオフ）後の，再見積りが求められるからである．さらにその見積り方法は，開発者だけでなく顧客にも信頼され

る必要がある．

アジャイル開発ではソフトウェア要求を表す単位としてユーザーストーリーを使う．これは顧客がソフトウェアで実現したいと思っているフィーチャ（機能）の概要を記述したものである[M.Cohn 2005][J. Rasmusson 2010]．ストーリーポイントはユーザーストーリーの相対的な作業の大きさを測る単位である[J. Rasmusson 2010].

[M. Cohn 2005]は，ストーリーポイントによる 2 通りの規模見積りの方法を紹介している．ひとつはイテレーションで実現するプロダクトバックログの全ストーリーの中で，相対的に最も小さいものを基準とし 1 ポイントとする方法，もう一つは，相対的に中くらいの大きさと思えるストーリーを決めて，それに見積りで使う数値範囲の中央に近い値（例えば，”5”）をつける方法である．

また，[J. Rasmusson 2010]は，ストーリーポイントによる規模見積り手法として，三角測量(Triangulation)とプランニングポーカーを紹介している．三角測量は他のストーリーと比較できる代表的なストーリーをいくつか基準として選出して，残りのストーリーを基準となるストーリーとの相対サイズで見積もる方法である．プランニングポーカーは，初めに開発メンバー各人が各ストーリーを見積り，それらをメンバー全員で議論し再びチームとしての見積りを決定する方法である．

4.3.2 採集プロジェクトデータの状況

本研究では，2 企業よりデータを採集した．2 企業はアジャイル開発を実施する際にストーリーポイント(SP)による規模見積りを行い，その見積りと実績データを採集し，構築していた企業である．本研究では，企業名称を仮に B 社，C 社とする．

C 社では 9 件のプロジェクトの規模，工数について予測，実績の計測をしており，工数見積りは，サンプリング方式を採用していた．これは実績データが無くとも計測可能な手法である．

本研究では，この見積りプロセスを明確化し B 社の新規プロジェクトに適用し採集したデータで，その有用性を検証した（表 4-2，1 行目）．さらに C 社の複数のプロジェクトデータを基に，実績方式の有効性を検証した（表 4-2，2 行目）．

その見積り範囲とその実施状況を表 4-2 に示す．

表 4-2　採集プロジェクトデータと検証方法

団体	プロジェクト数	開発前全体見積り*(1)		各イテレーション開発前見積り		見積り方式	検証方法
		規模	工数（時間）	規模	工数（時間）		
B社	1	○	○	○	○	サンプリング方式	新規案件へサンプリング方式適用
C社	9	○	○	○	○	サンプリング方式	既存実績データによる回帰式の検証

『○』・・・採集データ有り『▲』・・・規模と同じ値で採集『－』・・・採集データ無し
*(1)　プロジェクト開始後、開発開始前に実施するプロジェクト全体の見積り

注：アジャイル開発の場合，要件定義は必ずしもプロジェクト初期段階のみならず，各イテレーション開始前にも実施される．

4.3.3　サンプリング方式の見積りプロセスとデータ採集ポイント

サンプリング方式は，アジャイル開発の工数見積り方法のひとつとして，4.2.3 項で説明した．C 社では全てのプロジェクトにプロジェクト開始前から標準的な見積りプロセスを適用している．ここではデータ採集ポイントを明確にするため C 社の見積りプロセスを図 4-2 に示す．

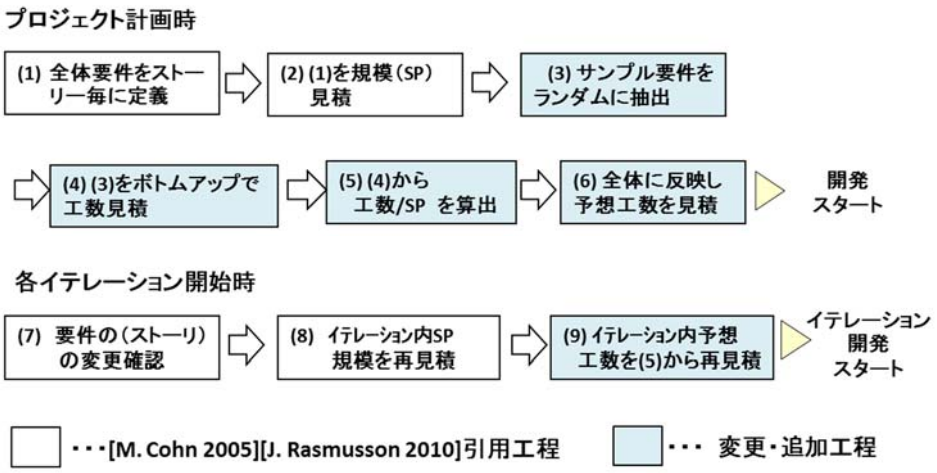

図 4-2　サンプリング方式による見積りプロセス

プロジェクト開始時には以下のプロセスで規模・工数を予測する．以下に説明する図 4-2 の(3)から(6)および(9)がサンプリング方式として追加した工程になる．

(1) 要件（ストーリー）定義

全体要件（ストーリー）を洗出しプロダクトバックログ（要件リスト）を作成する．それらを実現するための開発側から見た調査，環境構築要件も加える．

(2) ストーリーポイント(SP)による規模見積り

SP は要件（ストーリー）定義後に各要件（ストーリー）の規模を把握するために行う．（C 社では『フィボナッチ数列を用いた方法：ストーリーを 1, 2, 3, 5, 8…で分類しサイジングする.』を採用）全体規模を累計して見積もる．方法は三角測量を基準とする．それらはタスク分解のベースになる．

(3) サンプル要件（ストーリー）抽出

工数見積りの為に，全体の 10%～20%の要件（ストーリー）をランダムに抽出する．

(4) サンプル要件（ストーリー）のボトムアップ工数見積り

抽出した要件（ストーリー）サンプルをタスク分解しその工数を見積り，要件（ストーリー）毎に累計する．

(5) SP 当たりの平均工数算出

サンプル要件（ストーリー）の SP 数と(4)で見積もった工数から 1SP 当たりの平均工数を係数として求める．

(6) 全体の各要件（ストーリー）工数見積り

(2)の SP と(5)で得られた係数によりプロジェクト全体の工数を要件（ストーリー）毎に求める．また，各イテレーションの開始時には以降のプロセスで規模・工数の予測値を再度調整する．

(7) 要件（ストーリー）の変更確認

各イテレーション開始時に開発進捗や顧客との調整により要件（ストーリー）の優先度を確認し，要件（ストーリー）修正を行う．（基本的に同規模要件のトレードオフ）

(8) イテレーション SP 数の再見積り

(7)で修正した要件（ストーリー）に対して，再度フィボナッチ数列を用いた方法で SP を見積もる．

(9) イテレーション内予想工数を再見積り

(8)の値と(5)で得られた係数により工数を再見積りする．

4.4　検証と分析

4.4.1　過去開発データの有無による違い

4.2.3 項，表 4-1 で述べたアジャイル開発工数見積り方法のうち，全タスク分解方式は，ファンクションポイント法をはじめとするソフトウェア開発で広く利用されている．しかし，全タスクを分解し見積もる負荷から，本研究のねらいとするコスト低減にはそぐわない．そのためここでは，サンプリング方式とその実績方式の検証分析に焦点を絞った．

B 社による検証では，B 社をアジャイル開発で新規に見積りに取り組む初期段階（ステージ 1）のユーザーとして，新規のアジャイル開発プロジェクトで複数のパターンでのサンプリングを使った見積りをし，誤差を検証することでサンプリングそのものの有用性検証を目的とした．この有用性が確認できれば，それを使った実績構築をする次の段階（ステージ 2）に進める．また C 社の実績データ検証は，C 社を既にサンプリング方式による実績を持つステージ 2 のユーザーとして，それら実績データから回帰線を得る実績方式でその見積りの有効性を検証することを目的とした．この有効性が確認できれば，この回帰線から得られる式での見積りも可能になる．これら 2 社での検証の違いを表 4-3 に示す．

表 4-3　2 社の検証の違い

会社	アジャイル開発の取組みステージ	検証方法	目的	次ステージ
B社	ステージ1（新規の取組み）	誤差検証	サンプリング方式の有用性検証	実績を構築しステージ2へ
C社	ステージ2（実績データ有り）	実績データ検証	実績データから回帰線を得る方式で有効性検証	回帰線からの見積り実施

4.4.2　過去の開発データを使わないプロジェクトの見積りの検証

サンプリング方式の有用性を検証する為に，B 社のプロジェクトデータを採集した．Ruby の開発事業を戦略のひとつとする B 社では，3 年前よりアジャイルによる受託開発を開始している．検証の対象とするプロジェクトでは，見積りはイテレーション開始時に行う工数見積りのみで，規模および計画時の見積りは実施していなかった．そこで開発中のアジャイル開発プロジェクトに，C 社で実施のサンプリング方式を遡り適用し，その有用性を検証した．

(1)　B 社の研究試料管理システムプロジェクト

B社では文教系プロジェクトとしてO大学の既存システムである地質学研究試料管理シ

ステムの機能拡張プロジェクトをアジャイル開発で実施している．

Ruby言語によるアジャイル開発で，3カ月で既存機能更新と機能追加を行うプロジェクトである．従来実施していなかった規模・工数見積りについてC社よりサンプリング抽出方式のトレーニングを受講し遡って適用した．トレーニング以前からの顧客との取り決めで以下の成果物を作成している．

① ユースケース図 （全体モデル図）
② 機能リスト
③ 機能別画面一覧
④ 画面遷移

設計・開発は機能別に実施され，イテレーションは4週間（約一カ月）単位であった．B社では，顧客との契約時に初期計画時に規模・工数の見積りはなく，顧客の予算と希望納期により可能な要員数から準委任契約を締結している．この解決手段として，計画時に信頼性のある共通な基準によるシステム全体の規模・工数見積りが必要と考えた．以下の表4-4に採集データ項目とB社の採集データ項目を示す．

表4-4 採集データ項目とデータ内容

区分	項目	説明	B社新規システム
F-1	予測イテレーション数	開発前に見積もったイテレーション総数 納期，SP・要員数から調整し予測する	3 （3か月間4週間毎のイテレーション）
F-2	予測ストーリー数	開発前に見積もったストーリーの総数 （除く変更リスク）	18
★F-3	予測ストーリーポイント数 （初期規模見積）	開発前に見積もったストーリーポイント数 （フィボナッチ数列(1,2,3,5,8…)等で見積る プロジェクト全体の予想規模のSP数）	ストーリー毎の予測SP数と合計SP数 （表4-5参照）
F-4	予測開発工数 （初期工数見積）	開発前に見積もったプロジェクト開発工数	ストーリー毎の予測開発工数と合計工数 （表4-5照）
A-1	実績イテレーション数	開発後のプロジェクトのイテレーション数	3 （3か月間4週間毎のイテレーション）
A-2	実績ストーリー数	開発後のストーリー数	18
A-3	実績ストーリーポイント数 （開発後規模）	開発後のストーリーポイント数	ストーリー毎の実績SP数と合計SP数 （表4-5参照）
★A-4	実績開発工数 （開発後工数）	開発後のプロジェクト開発工数	ストーリー毎の実績開発工数と合計工数 （表4-5参照）

区分 ★・・・検証で使用した項目
F: Forecast 開発開始前見積り値　　A: Actual 開発完了後実績値

採集データ項目は，パーソナルソフトウェアプロセス技法[W. Humphrey 2009] （以降”

PSP”とする）を参考にした．ただし，PSP の場合，規模測定の標準を LOC (Lines Of Code) とするため，成果をソースコード数としないアジャイル開発手法にそぐわない．そこで規模測定単位をストーリーポイント(SP)とし，F(Forecast): 開発開始前見積り値，A(Actual): 開発完了後実績値の区分でデータを採集した．

(2) プロジェクトのシステム要件比率

検証対象プロジェクトは O 大学地質学資材の管理システムで「（１）既存システムのインフラアップグレード」と「（２）追加機能拡張」の二つを目的としている．

表 4-5 に全要件（ストーリー）一覧を示す．

ユーザーからの要件は基より，それらを開発する上で必要なシステム調査，環境構築，データ移行等のシステム要件は，予め見積りから要件（ストーリー）として追加した．

システム要件が全要件(ストーリー)に対する比率(% ,分数にて以下に表示)は,「（１）既存システム部分（1~13)）では，初期規模見積りで 28%（55/196)，初期工数見積りで 39%（63/160）だった．開発後工数実績での数値は 40%（63.5/159)と初期見積りと近い値だった．「（２）追加機能拡張部分（14~17)」でも，システム要件は初期工数見積りで 29%（19.5/65.5)，開発後工数実績では 27%（18/65）と僅差であった．

この，ほぼ 1~2%の誤差の理由としては，今回の分析対象期間が 3 か月の短期間であったことが理由と考えられるが，要件の実現に必要なシステム要件（全体の 30%~40%程度）を洗い出し，見積りに加えたことが大きく影響していると考えられる．

表 4-5　プロジェクト要件一覧

No.	要件（ユーザーストーリー）	初期規模見積（SP）	初期工数見積*(2)（人日）			開発後工数（人日）
			(A)	(B)	(C)	
1	既存システム調査	13.0	7.7	9.2	10.2	12.0
2	DB構築・データ移行	21.0	12.4	14.9	16.4	27.5
3	共通画面のヘッダー・検索機能作成	13.0	7.7	9.2	5.0	6.0
4	全試料一覧の表示機能作成	8.0	4.7	5.7	6.3	8.0
5	stone　一覧画面表示・個別検索・　明細表示	21.0	12.4	14.9	16.4	18.5
6	box　一覧画面表示・個別検索・　明細表示	13.0	7.7	10.0	10.2	9.0
7	place　一覧画面表示・個別検索・　明細表示	13.0	7.7	9.2	10.2	11.5
8	analysis　一覧画面表示・個別検索・　明細表示	13.0	7.7	9.2	10.2	4.5
9	bib　一覧画面表示・個別検索・　明細表示	13.0	7.7	9.2	10.2	11.5
10	file　一覧画面表示・個別検索・　明細表示	13.0	5.0	9.2	10.2	6.0
11	マスタ管理	21.0	12.4	14.9	12.0	12.0
12	帳票出力機能	13.0	7.7	8.0	10.2	8.5
13	納品準備・ドキュメント作成	21.0	12.4	14.9	16.4	24.0
14	外部インターフェイスの調査・実装・結合資料作成	13.0	7.7	9.2	17.0	15.5
15-(1) *(1)	クライアント端末用AP 調査・環境構築	21.0	12.4	12.0	16.4	12.0
15-(2) *(1)	クライアント端末用AP実装・結合テスト・資料作成	34.0	12.0	24.1	26.6	12.0
16	アクセス管理	8.0	4.7	9.0	9.0	12.0
17	運用要件　データ保護	8.0	15.5	5.7	6.3	13.5
	合計	280.0	165.5	198.5	219.2	224.0

□　サンプル抽出ストーリー
*(1)　15-(1),15-(2)は,同じイテレーション内に完了できないため分割したもの
*(2)　初期工数見積の値　①サンプル項目はタスク分解により算出
②サンプル外項目はサンプル項目の初期規模見積と工数見積の比率による案分で算出

(3)　サンプリング方式の適用

サンプリング方式による工数見積りを実施するにあたりサンプルとして選ぶ要件（ストーリー）として次の 3 点を留意した.

① 全要件（ストーリー）の基準とするため，代表的な要件（ストーリー）とする[J. Rasmusson 2010] .

② 全体要件（ストーリー）の初期規模見積りで得られた SP 数合計の約 20%を目安とする.

③ 大・中・小とサイズの異なる要件（ストーリー）を選択する[J. Rasmusson 2010] .

なお，①の「代表的な」の基準を，今回は以下 2 点に該当する要件（ストーリー）とした.

・ 論理的なグループ分けが出来そうなものであること.

・ プロジェクトを象徴する要件（ストーリー）であること.

上記の留意点から 表 4-5 における(1)，(2)，(13)は抽出対象外としている.

また，③では見積り技法として三角測量[J. Rasmusson 2010]を適用し，代表的な要件（ストーリー）（上記①参照）をいくつかの基準として選択して，残りの要件（ストーリー）を，基準となるストーリーとの相対サイズを SP で見積もる方法を適用した .

以上の条件を基に(A)(B)(C)三つのサンプリングを実施し，それぞれの実績値との誤差を求めた．表 4-5 はそのプロセスを，表 4-6 はその結果を示す．また，誤差を算出するまでの手順についてサンプリング対象(A)を例に以下に解説する．（表 4-5　参照）

① 全体の約 20%のサンプル要件（ストーリー）を抽出する．(No.10,15-(2),17)

② ①の各サンプル要件（ストーリー）をタスク分解し，工数を見積もる．(5.0,12.0,15.5)

③ ②の初期工数見積の合計(5.0+12.0+15.5 = 32.5)を，①の各サンプル要件（ストーリー）の初期規模見積の合計(13.0+34.0+8.0 = 55.0)で割り，サンプル要件（ストーリー）以外の要件（ストーリー）に対する係数を求める．(32.5/55.0　=　0.591)

④ サンプル以外の全 15 要件（ストーリー）(No.1~9,No.11~14,　No.15-(1),No.16)の初期規模見積の値 (13.0, 21.0, 13.0, 8.0, 21.0, 13.0, 13.0, 13.0, 13.0, 21.0, 13.0, 21.0, 13.0, 21.0, 8.0) それぞれに③の係数(0.591)をかけて，初期工数見積を求める．(7.7, 12.4, 7.7, 4.7, 12.4, 7.7, 7.7, 7.7, 7.7, 12.4, 7.7, 12.4 7.7, 12.4, 4.7 = 133.0)

⑤ ④に②のサンプル要件（ストーリー）の工数見積値を合わせ，全要件（ストーリー）の初期工数見積を求める．(32.5 + 133.0 = 165.5)

⑥ 誤差を以下の計算式によって求める.

誤差 = 1 - 初期工数見積りの合計（No.1～No.17 の合計) /

開発後工数の合計（No.1～No.17 の合計）

= 1 - (165.5 / 224.0) = 0.261 = 26.1%

⑦ サンプリング対象(B)(C)については，上記①の抽出対象をそれぞれ『No.6,12,15-(1),16』『No.3,11,14,16』に置き換えて算出すればよい.

表 4-6　サンプリング結果一覧

No.	サンプリング対象*(1)	サンプリング　ストーリー	サンプリング比率%*(2)	サンプル外項目に対する乗数*(3)	誤差%　*(4)
1	代表的大中小（A）	10,15-(2),17	19.6	0.591	26.1%
2	代表的大中小（B）	6,12,15-(1),16	19.6	0.709	11.4%
3	代表的大中小（C）	3,11,14,16	19.6	0.782	2.3%

*(1)　表 4-5 の初期工数見積 (A),(B),(C)参照

*(2)　サンプリング比率 = サンプルの初期規模見積合計 ÷ 初期規模見積の合計

*(3)　サンプリング外項目に対する乗数

　　= サンプル項目の初期工数見積の合計 ÷ サンプル項目の初期規模見積りの合計

*(4)　誤差 = 1 - 初期工数見積の合計（表4のNo.1～No.17の合計）÷開発後工数の合計（表4のNo.1～No.17の合計）

代表的大中小約 20%の要件（ストーリー）を抽出したサンプルの誤差はいずれも 2.3%～26.1%の範囲内にあった．この結果からアジャイル開発の実績が少ないか，または無くとも，サンプリング方式により実務上耐えうる誤差の範囲内でアジャイル開発を見積もることが可能であることが確認できた．これにより開発計画時の契約に於いてもこの見積手法が有効であると考えられる．

4.4.3　実績データによる見積りの検証

本項では，既に実績値としてサンプリング方式での見積りを実施し，完了したプロジェクト実績と見積り値の相関から有効性を確認する．

C 社はブラジルに本社をもち，米国，中国，英国，そして日本で金融，消費財，インターネット業界の大企業・中堅企業向けにアジャイル開発サービスを提供し，2010 年以降，100%アジャイル開発で受託事業を推進している IT 企業である．しかし，実際の契約では初期計画時の見積りは実施しているものの，開発のなかで変更を受け入れて行くアジャイル開発ではその信頼性の論理的証明が難しく，準委任契約の形態をとる場合が多い．

本章では，各指標の計測を実施した日本企業向けのプロジェクトデータを基に相関および回帰分析を行い[Yoshida 2013]，その検証結果による受託事業の推進を目的とした．

(1) C 社プロジェクトの採集データ

本研究では C 社が日本市場で実施したアジャイル開発プロジェクト 9 件（37 イテレーション）のデータ項目を基に，相関・回帰分析を実施し考察した．採集したプロジェクトデータ項目は B 社と同様の表 4-4 に示す項目を採集した．

(2) プロジェクト全体の相関分析

過去プロジェクトの実績値を基に新規プロジェクの見積りを実施するために，表 4-4 の

採集データより規模予測データである「F-3 予測 SP 数」と，実際に開発作業に要した工数を表す「A-4 実績作業時間」の相関と回帰を，R を用いて検証した． 回帰直線をひいた散布図を図 4-3 に示す．

散布図中の横軸（x 軸）は予想 SP 数，縦軸（y 軸）は実績開発工数を表している．

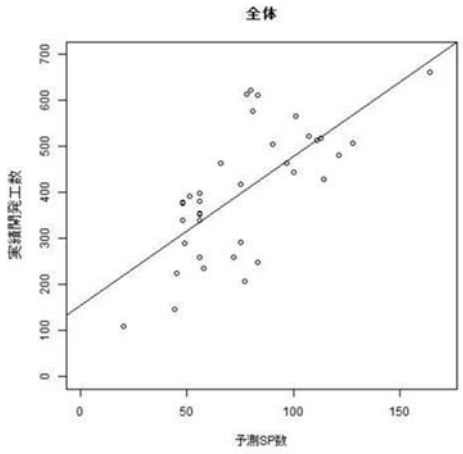

図 4-3　相関係数による散布図と回帰直線グラフ

ドット数は，全対象プロジェクトのイテレーション数の合計である．相関係数は 0.69 であり 1%の水準で有意（両側）になっており，相関があることがわかる．

(3) フェーズ分割での相関

アジャイル開発は短期間の開発を完遂に向かい繰り返していく反復開発が基本である．そこでプロジェクト全体の見積り精度と初期フェーズの見積り精度，さらに終期フェーズのそれでは差異があるのではないかと考え，プロジェクトデータをイテレーション毎に前期・中期・後期の 3 フェーズに分割しそれぞれの相関を求めた．イテレーションが均等に 3 分割できない場合は中期のデータで調整している．例えば，全 5 イテレーションのプロジェクトの場合は，前期を 2 イテレーション，中期を 1 イテレーション，後期を 2 イテレーションとした．また，イテレーション数が 3 未満の場合は，対象外とした．表 4-7 は各期の相関係数，回帰係数を示す．なお，図 4-4 のドット数は前期・中期・後期に分配したイテレーション数を意味する．

表 4-7　開発期別相関係数　「予測 SP 数 vs.実績開発工数」

開発フェーズ	イテレーション数＊(1)	ピアソン相関係数	回帰係数（a）	回帰定数（b）
開発前期	12	0.417	1.636	289.912
開発中期	10	0.638	3.143	168.132
開発後期	12	0.704	3.118	173.621

*(1)・・・グラフ中のドット数 / イテレーション数3未満のプロジェクトは対象外.

分析結果より，開発前期から中期，後期とフェーズが進むに従い相関係数も徐々に高くなり，後期では 0.70 を越える相関があるといえる．このことから開発が進むにつれ見積り精度が安定してゆく傾向が伺える．また，前期・中期・後期のそれぞれにおいて回帰直線をひいた散布図を図 4-4 に示す.

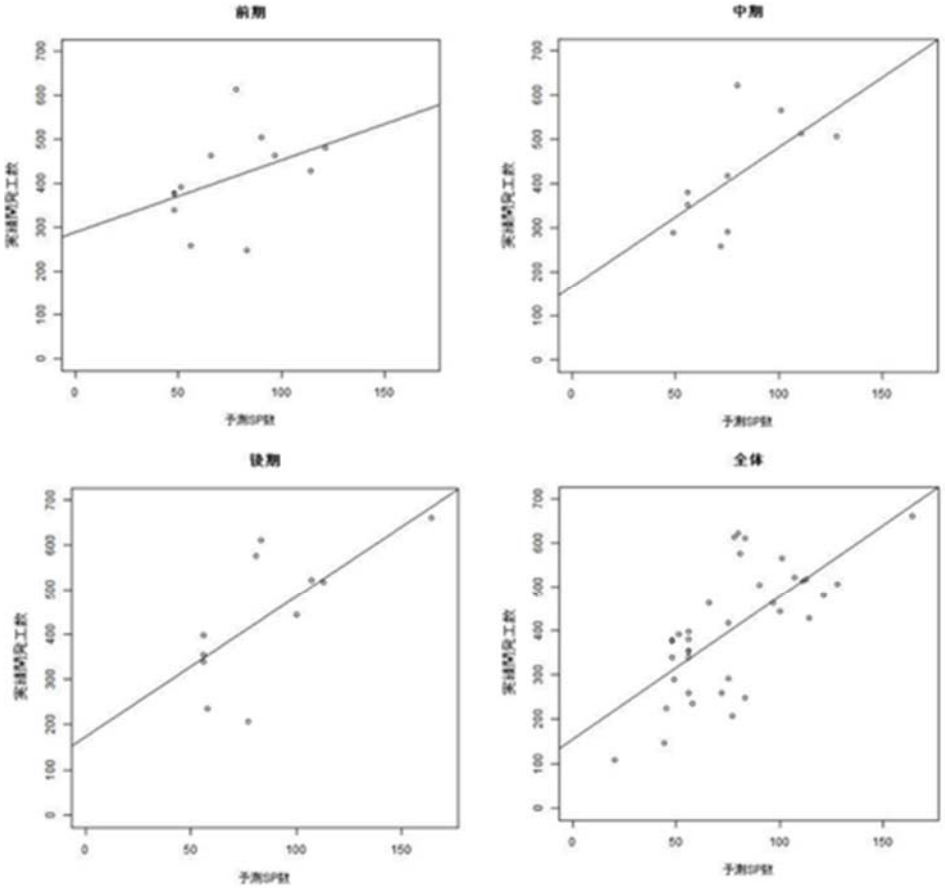

図 4-4 期別相関係数による散布図と回帰直線グラフ

新規プロジェクトの見積りで，その技術，業務の習熟度により，未経験要素の多いプロジェクトには前期データでの係数，類似プロジェクト経験があるプロジェクトには中期以降のデータを利用するといった使い方も可能になると考える．

4.5 まとめ

本章では，サンプリング方式の有効性を検証することを目的にした．そして 2 つの企業 B 社，C 社のデータに基づいて検証を行った．

B 社プロジェクトでは, 過去のプロジェクト実績データを使わないサンプリング方式を，C 社の複数の実績データ分析からは，実績方式を検証した．B 社プロジェクトでは，実績データのない企業でも要件（ストーリー）サンプリングの条件によりストーリーポイントでの見積りが有効であることが確認できた．C 社のデータ分析では，予想と実績の間に相関がみられ，ストーリーポイント数を用いた工数の予測の可能性を認めた．この結果よりアジャイル開発におけるサンプリング方式は有効な見積り方式であると考えられる．

第 5 章　アジャイル開発のプロジェクト契約モデルの検討

第1章に述べた日本でのアジャイル開発の採用率の低さには，開発中の要件変更に対応するために初期段階で見積りが確定し難いという課題がある．このことを，第3章の阻害要因の調査で確認し，第4章でサンプリングによる見積り方式を提案した．さらに，第3章の調査では，発注者，開発者間での要件合意の難しさとそれに関連する契約管理の難しさも理由として確認された．また，2.2.2項のとおり，欧米でのアジャイル型開発の採用率の高さは，情報システム構築を自社で内製化する企業が多く，開発を発注する企業が少ないことが大きな理由のひとつとなっている[IPA 2012]．日本の場合は，顧客が開発を発注しベンダーが開発を請け負う受託開発が多い．その受発注の遂行のために契約締結が不可欠になっている．

契約はIT（Information Technology）システム構築プロジェクトの成功にとって大きな役割を担う．日本のシステム開発契約では請負，準委任，および労働者派遣という三つの契約類型がある．

本章ではシステム開発委任契約に関わる契約形態として請負，準委任を対象にアジャイル開発を請負契約で遂行できるモデルを提案する．

5.1　研究のアプローチ

はじめに，経済産業省の「情報システム・モデル取引契約書」，情報処理推進機構の「アジャイル開発のための契約モデル（案）」に注目し，アジャイル開発に適用する上での課題を明確にする．さらに，アジャイル開発を実務に適応するために契約書上の条項を提案する．提案する契約モデルでは，第4章での見積り方式を適用しプロジェクトの工数算出を行う．

本章の研究テーマとプロジェクトデータとの関連性を図5-1に示す．

図5-1 アジャイル開発の契約モデル

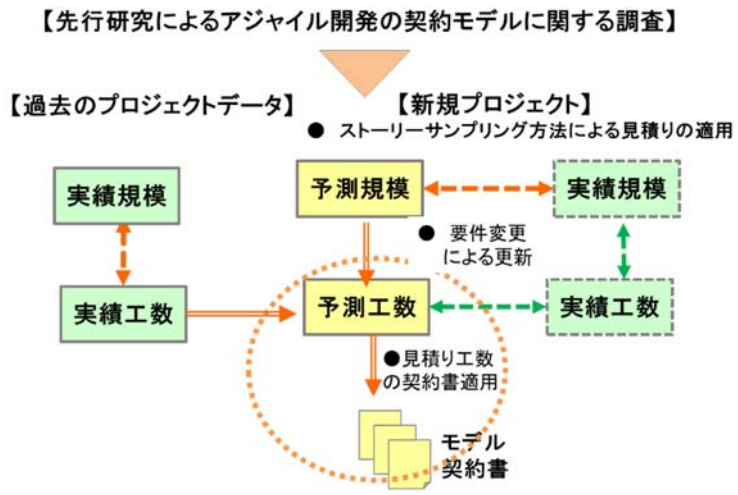

図5-1 では，ストーリーによるサンプリング方式から見積もった予測工数をもとに，コストを算出し，ソフトウェア開発委託契約書の条項とするプロセスを示す.

5.2 アジャイル開発とわが国の開発モデル契約

5.2.1 アジャイル開発の特徴

アジャイル開発とウォーターフォール型開発モデルはその開発方法に違いが多いため頻繁に比較される[Matsushima 2009][Ikoma 2008].

ウォーターフォール型開発では開発範囲をプロジェクト開始時に決定し，納期，費用はプロジェクト開発の過程で変動する可能性がある．アジャイル開発では，納期，費用は概ね確定しているものの，開発範囲は開発過程で環境変化や顧客満足などにより変更され，初期の要件は変更要件とトレードオフされ進められる． そのアジャイル開発の特徴を時間・コスト・その他に分類し表5-1にまとめる.

表5-1　アジャイル開発のメリット・デメリット

	メリット	デメリット
時間	・開発期間の短縮できる ・効果を早期に実現 ・環境リスクの削減 ・利益を早期にもたらす[Matsushima 2010]	・見積り工数が膨らんだ場合の調整が困難 ・終わりが見えない
コスト	・文書作成コストの大幅削減できる ・過剰な調整コストの削減できる	・熟練技術者が求められ人件費@が高い
その他	・発注単位を小さくでき、進捗に応じて明確になった分だけ発注できる ・発注側は停止や縮小などの多様なオプションを持てる ・金融工学の手法で金額換算を可能にする[Matsushima 2010]	・大規模開発に向かない。 ・オフショア開発に適さない。 ・チームメンバーの都合のよい解釈で進められがちで，思わぬ方向に進む場合がある

アジャイル開発モデルの特徴として，時間の観点では「開発期間が短縮できる」が最大の利点とされる半面，開発範囲が不明確なことからいつ完結するか「終わりが見えない」ことが欠点とされる．またリリースが，イテレーションの都度実施される点で「利益を早期にもたらす」[Matsushima 2010]ことが挙げられる．また，コストの観点では，「過剰な文書作成，管理調整に係わるコストが削減される」点が利点としてあげられ，デメリットとしては「熟練技術者の人件費の高さ」が挙げられる．

その他としては，「発注側が縮小，中止等の選択肢を持つことができる」こと，また「金融工学の手法で金額（収益）換算が可能であること」[Matsushima 2010]がメリットとされる．一方，「大規模開発に向かない」「オフショア開発に向かない」などがデメリットである．

5.2.2　準委任と請負の相違

日本のソフトウェアに関する契約類型のひとつであるシステム開発契約書では，請負，委任に準じた準委任，および労働者派遣法に基づく派遣契約がある．このうち労働者派遣は労働者を派遣して派遣先の指揮監督のもとに派遣先の業務に従事させる契約であるため，本項では，請負，準委任によるシステム開発契約を検討する．両者はその目的の違いから，次のとおり定義される[Yoshida 2001].

・請負は，定められた仕事を完成させることを目的とする契約である．

・準委任は，代理人を頼むような法律行為以外の事務の遂行を目的とする契約である．

通常，委任契約は弁護士などの特定業務の場合のみに適用され，通常の情報システムコンサルタント，アドバイザー等の専門サービスには準委任が適用される．

システム開発での請負契約と準委任契約は，仕事の完成義務と瑕疵担保責任から大きく区分される．その違いを表 5-2 に示す．

表　5-2　請負契約と準委任契約の違い

	仕事の完成義務	瑕疵担保責任
準委任契約	受託者は仕事の完成について一切責任を持たない	受託者は仕事の完成義務はないので，請負のような瑕疵担保責任を負うことはない．
請負契約	受託者は仕事の完成について一切の責任を持つ	受託者は委託者に完成されたものとして引き渡された成果物に瑕疵があった場合，無過失責任としての瑕疵担保責任を負う．

① 仕事の完成義務：請負ではベンダーは仕事（受託業務）の完成の義務を負うのに対し，準委任ではベンダーは善良な管理者の注意をもって委任事務を処理する義務を負うものの，仕事の完成についての義務は負わない．

② 瑕疵担保責任：請負では，仕事の完成義務を負うので，例えばユーザーに完成されたものとして引き渡された成果物に瑕疵があった場合，無過失責任としての瑕疵担保責任を負う．このような場合，民法（第 634 条乃至第 640 条）によれば，注文者であるユーザーは修補や損害賠償を請求することができる．また，瑕疵により契約の目的を達成することができないときは契約を解除することができる．これに対して委任に準ずる準委任の場合は，仕事の完成義務はないので，請負のような瑕疵担保責任を負うことは無い．このように，請負と準委任には法律上の重要な相違点がある．但し，民法の強行法規（強行規定）を除く任意法規（任意規定）に置いては契約書において，具体的な取引・契約プロセスに合致するように条件を軽減することが可能である [Minpou 2011].

さらに，アジャイル開発の場合は常にユーザーとベンダー間で要件の調整が必要になるため，請負は適切なスタイルとは言い難い．

実際の契約において，準委任型とするか，請負型とするかは，成果物についての当事者同士の経験や役割分担の遂行能力等に基づく．これは成果物についての共通理解が事前に

十分に成立していることが前提となるが，アジャイル開発の契約で計画時に請負型とした場合は，ユーザーのシステム化計画や要件定義における調整責任が曖昧になり，プロジェクトへの介入が希薄になるリスクが考えられる．その結果，要件定義上の見落としやすれ違いも生じ易くなることでソフトウェア品質にも影響を及ぼすこととなる．

5.2.3 経済産業省「モデル取引・契約書」との関連

経済産業省「モデル取引・契約書」の対象となるプロジェクトの前提は次のとおりである．

- 契約当事者：対等に交渉力のあるユーザー・ベンダーを想定
  (例)委託者（ユーザー）：民間大手企業，受託者（ベンダー）：情報サービス企業
- 開発手法：ウォーターフォールモデル
- 対象システム：重要インフラ・企業基幹システムの受託開発，保守・運用
- プロセス：共通フレーム 2007 による標準化されたシステム企画・開発・運用・保守プロセスによる一括発注の場合に加えて，マルチベンダー形態，工程分割発注に対応．
  パッケージ，SaaS/ASP については，＜追補版＞として，2008 年 4 月に別途公表された[METI 2008].

以上に述べた「モデル取引・契約書　第一版」とパッケージ＜追補版＞の前提の違いを表 5-3 にまとめる．

表 5-3　「モデル取引・契約書　第一版」とパッケージ＜追補版＞の前提の相違

	モデル取引・契約書 ＜第一版＞（平成19年4月公表）	パッケージ,SaaS/ASP活用,保守・運用 ＜追補版＞（平成20年4月公表）
契約 当事者	・対等に交渉力のあるユーザ・ベンダ	ITの専門知識を有しないユーザと業として情報サービスを提供するベンダ
対象モデル	・ウォーターフォールモデル	パッケージ，SaaS，ASP
対象システム	・重要インフラ・企業基幹システムの一般業務システム、受託開発（一部企画を含む）、保守・運用	一般業務システム
特長	・ユーザ・ベンダ双方が議論の上策定 ・フェーズごとのユーザ・ベンダ間の責任の明確化（準委任・請負） ・SLCP2007準拠（共通プロセス） ・仕様の変更管理手続の明確化 ・マルチベンダ・工程分割発注への対応	・重要事項説明書を用いた契約合意 ・ITコーディネータや中小企業診断士をはじめとする外部専門家やコンサルタントの参画を前提 ・システム構築後のプロセスの重視（保守，運用等） ・パッケージソフトウェアの取扱いについてのベンダの責任明確化 ・著作権のベンダへの帰属 ※上記以外の点について第一版の特徴は原則，追補版でも踏襲

パッケージ，SaaS/ASP を対象とする追補版での前提は，契約当事者は IT の専門知識を有しないユーザーと業として情報サービスを提供するベンダーとされている．対象モデルはパッケージ，SaaS，ASP となる．そして IT コーディネーターや中小企業診断士をはじめとする外部専門家，コンサル担当の参画が前提とされることが条件とされる．また，著作権のベンダー帰属が前提とされることも異なる．この追補版は，アジャイルを含む反復型開発など上記と異なる場合の活用については，必ずしも適用しない．

先に述べた前提事項と民法による契約類型「請負」，「委任」に準ずる「準委任」の適用から，モデル取引・契約書での理想的なユーザーとベンダーの役割分担としては以下が望ましいとされている．

- 企画段階：ユーザーが業務要件およびシステム要件（外部設計に対するインプット）を主体的に決定・明確化する．
- 開発段階：ベンダーが主体となって業務全体に対する利害関係者の要件のうち，システムに関する部分（システム要件）について仕様化を行う．
- 要件追加，仕様変更，未決事項：外部設計がユーザーによる承認を受けた後は変更管理手続に則り委託料・納期等の協議を実施する．

経済産業省では，このモデルを実現するために下記の契約類型を基本としながら各フェーズにおけるユーザー・ベンダーの責任分担を契約書において詳細に規定することを推奨している．ただし，実際の契約において，請負型とするか，準委任型とするかは，成果物の特定についての当事者同士の経験や役割分担の遂行能力等に基づき，成果物についての共通理解が事前に十分に成立しているかに依存する．

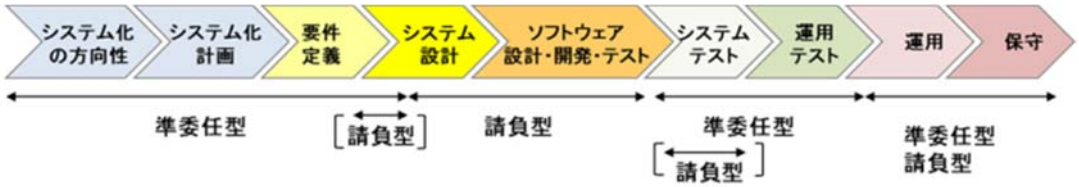

図 5-2　ウォーターフォール型開発モデル契約

（ [METI 2012]を参考に加筆修正．）

5.2.4　情報処理推進機構「アジャイル開発向け契約モデル案」の課題

2.2.1項でも述べたとおり，現在日本で定着し成熟した市場を持つウォーターフォール型

の開発工程は，すべての機能を開発するまで本番運用ができず，稼動開始までに長期間を要する．さらに，完成時にはニーズとかけ離れたシステムになるリスクがある．プロトタイピングでスモールスタートを図る場合は，あくまで全体開発の試作目的であるため，それのみを単独でリリースするケースは現実的に契約上合意も難しく，極めて稀である．これに対し，アジャイル開発工程はまさにスモールスタートをするために形成されてきたといえる．早期に本番運用を開始でき，徐々に拡張していくインクリメンタル型の開発である．それは，早期の投資回収を可能とし，システム開発の投資効率を改善できる．また，ユーザーニーズを吸い上げながら徐々に拡張することによってユーザーにとって価値のある機能を作り込むことができる利点がある．これらはすぐにシステム化することでビジネスに繋げたい中小企業等の新規サービスモデルの構築などには大変適したモデルといえる．

アジャイル開発手法の，ひとつであるスクラムでは，一般に初期計画時の見積りを実施しているプロジェクトもあるが，実際はイテレーションのはじめにその都度行っているケースが多い．

開発の中で変更を受け入れて行くアジャイル開発の特質から契約はリリース単位毎に行い，受託側が完了責任を持たない準委任型が一般的である．図5-3に情報処理推進機構が推奨する開発モデル契約[Umemoto 2012]を示す．

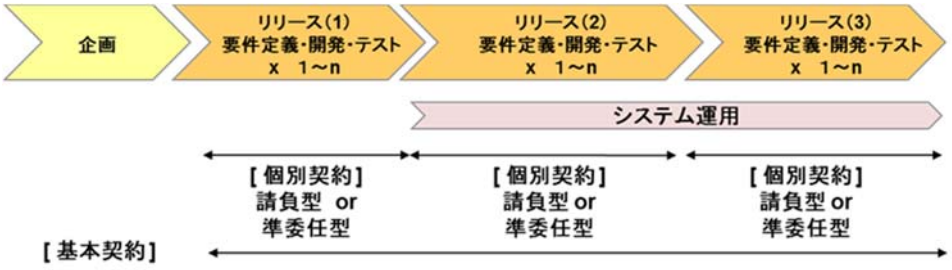

図5-3　情報処理推進機構　アジャイル開発モデル契約

（情報処理推進機構「アジャイル開発向けモデル契約案について」

[Umemoto 2012]を参考に加筆修正．）

上記モデル契約は，アジャイル開発契約のために検討・推奨されるモデルである．そこでは企画フェーズ，つまり開発前の基本契約と，リリース単位毎に開発内容と費用が確定した段階で締結される個別契約が提案されている．

各リリースでは，要件定義・開発・テストがイテレーションとして 1～n 回反復される．

基本契約では，プロジェクト全体に共通する事項を定め，それが締結された上で，個別の機能内容について協議を行い，開発対象が確定し次第，（例えば 1 回にリリースされる開発対象機能群をひとまとめにして）順次，個別契約（請負契約/準委任契約）を締結する．

基本契約では，システム開発に関わる方針や環境，期間など概要が定められるが，全体の要件は変更を前提として概要のみを定義しているので，規模・工数・コストに関わる部分は未定になる．また，実際のアジャイル開発は，中・小規模の開発案件が多く，1～2 週間のイテレーションを数回反復しリリース，または途中でのリリースが行われるものも多い．リリース単位に複数反復分の要件定義から開発・テストまでの工程を請負契約が可能となる高い精度で見積もることは，ウォーターフォール型開発の見積りと同程度の時間，労力が必要と考えられる．

これら基本契約でコスト算出ができない，各イテレーション開始時に見積り負荷がかかる，の 2 点から現実の運用では適応し難いプロジェクトが多く出てくると想定される．

5.3 新アジャイル開発契約モデルの提案と実務へのアプローチ

5.3.1 新アジャイル開発の契約モデルの提案

アジャイル開発プロジェクトで計画時に信頼性のある共通な基準によるシステム全体の規模・工数見積りを行うことができれば，より合理的な委託・受託側の協議が可能になる．新モデルでは，プロジェクト計画時に全体の規模・工数見積りを実施することを前提に，図 5-4 のとおりフェーズ分けした開発ライフサイクルのサービス内容を発注側と受注側が協議の上サービスレベル契約を締結しプロジェクトを遂行する．これは，情報処理推進機構のアジャイル開発モデル契約をもとに，そこでの課題である次の点を解決すべく提案するものである．

① プロジェクト計画時に全体の工数・コスト概算見積りによる開発フェーズの請負契約の実現．
② リリース単位毎の見積りによる負荷の削減．

運用・保守のフェーズでは，計画時と終盤でのスコープ乖離や環境・経営変化にともなう要件変更・追加に対応する開発を含めた運用に対する契約を行う．そのライフサイクルとモデル契約を図 5-4 に，そのプロジェクトフェーズと作業内容を表 5-4 に示す．

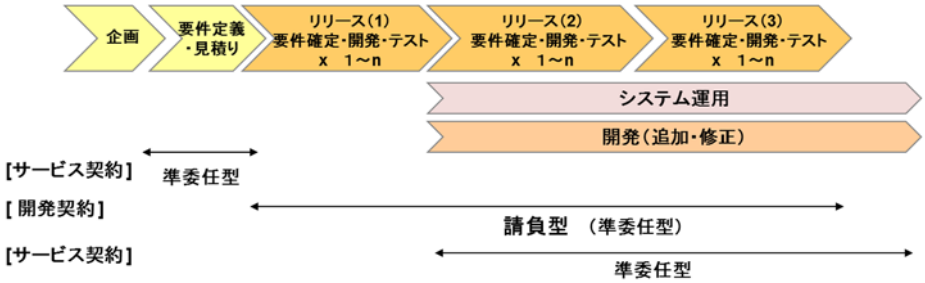

図 5-4　アジャイル開発のライフサイクルとモデル契約

表5-4　新アジャイル開発フェーズと作業内容

フェーズ	作業内容	契約	備考
企画	システム化計画 目的・組織・予算・期間・環境の明文化	準委任契約	通常発注側作業
計画	①全体の要件(ストーリー)定義 ②ストーリー ポイントによる規模見積り ③タスク分解、または係数による工数見積り	準委任契約	
開発	①要件(ストーリー)確定 ②開発 ③テスト	**請負契約** (準委任契約)	
運用・保守	①運用 ②補修・追加開発 ③保守	準委任契約	

5.2.4項の情報処理推進機構「アジャイル開発向け契約モデル案」との違いは，第一に計画フェーズにおいて明確になっている要件を洗い出し，全体の見積りを計画段階で実施することである．図5-3と比較すると，企画フェーズ後に要件定義・見積りフェーズがある．その時点で，明らかになっている全体要件についての見積りを実施する．

アジャイル開発の実績を持たない，または少ない企業では，実績データから得られる見積りの算出根拠となる基準値が構築されていない．そのため開発要件が明確化するまではイテレーション毎の開発直前にタスク分解しボトムアップで見積りを実施することになる．そこで見積り負荷の解決策として，計画時と各イテレーション（スプリント）開始時の見積り手順を図5-5のとおり提案する．

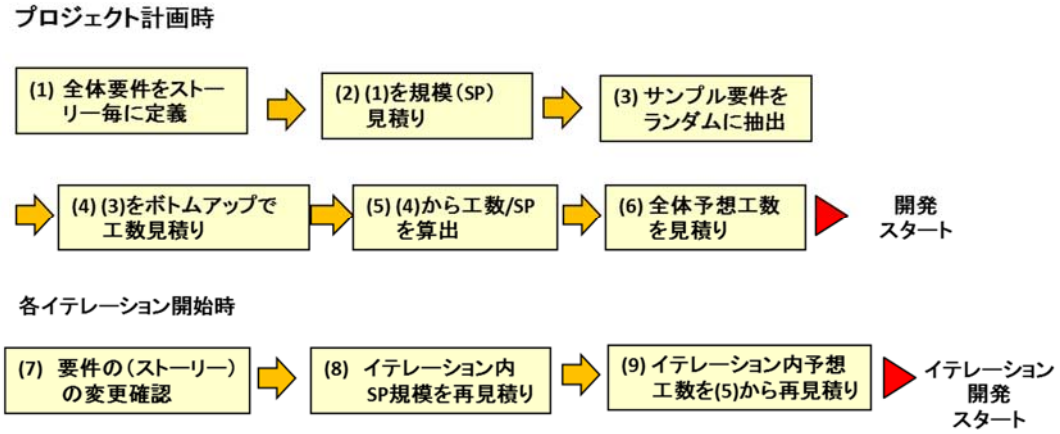

図5-5 アジャイル開発の見積り手順

これで全体の開発規模・工数を把握することができるため，開発前の契約では概算コストを入れた開発契約が可能である．第二に運用・保守フェーズでは，開発中の要求変更率（1－計画時の要件の最終残存率）より稼働後に発生することが想定される追加・修正要件に対応する開発工数を別途，保守と同様に運用フェーズに含め，契約を締結することもできる．

5.3.2 新モデルの適用により予想される効果

アジャイル開発を請負契約で実施することは，受託者側の見積り精度に信頼性と論理的な基準が求められる．さらに，この新しいモデルを適用することによる効果とその理由を以下に述べる．

（１）リリース毎の契約業務の煩雑さを軽減する

従来モデルでのリリース毎の見積りと契約の実施は，小規模プロジェクトの場合，期間が大変短くなり，かなり煩雑な実務を伴う．本提案モデルでは，企画フェーズ，開発フェーズ，運用・保守フェーズの 3 回とすることで実務を軽減する．

（２）請負契約で実施できる

プロジェクト全体を準委任契約にすることは，ベンダーが完全責任を負わない点で，たとえ成果物が完成しなくても契約上対価を支払わなければならないことはユーザー側のリスクとなる．本提案モデルで請負契約することでこのリスクは軽減できる．

（３）「サービスレベルアグリーメント(SLA)」の締結が可能になる

現在アジャイル開発向けに検討されているモデルでは，イテレーション毎の個別契約

締結が推奨されている．本提案モデルでは，開発，テスト，オペレーション，保守といったフェーズ毎に提供するサービス種類によって契約を分け，委託者・受託者間の協議の中でサービス内容の詳細を取り決めることが可能になる．これによりサービスレベルアグリーメント（SLA）を締結することができる．また，ユーザー，ベンダー間のみならず外部のサービス提供者との契約であるアンダーピンニングアグリーメント（UC）もこの SLA が基準となる．

5.3.3 実務契約へのアプローチ

先に述べた 5.2.4 項, 図 5-3 の情報処理推進機構のモデルではアジャイル開発の契約において，基本契約と個別契約が締結される．基本契約書では開発概算が見積もれないため委託料の条項はなく，開発委託料は個別契約で複数回締結される．一方，本研究で提案する新モデルでは，ウォーターフォールモデルと同様に開発フェーズでは一回のシステム開発委託契約で締結することが可能となる．それを確認するために，図 5-4 のモデル契約を用いたアジャイル開発において，既存のウォーターフォール型開発を前提とした標準様式である「システム開発委託契約書」を使い契約締結ができるか確認した．

必要とされる契約条項にあてはめたところ，以下の条件を満たせば，適用できることが確認できた．その条件は以下のとおりである．

(1) 委託料は 5.3.1 項の手順で得られた見積りを基に算出する．
ただし，フェーズ毎に，準委任契約，請負契約と契約類型が異なる場合があるため，図 5-6 の様な条項（例）を契約に追加することが望ましい．なお，ここでは甲は顧客（委託者，発注者）で，乙はシステム開発会社（受託者，受注者）となる．

(2) 企画フェーズ，計画フェーズは，準委任契約に准じ，必要なスキルと作業を定義する. そのスキルと時間当たりの単価に対して価格を設定するか, ある一定期間, 一定作業時間に対する料金を委託料として契約する．

(3) 開発フェーズで，請負契約を締結する場合には，要件変更による超過規模分（計画時見積りからの超過分）については委託料変更がともなうことを示す条項を契約書に追加する．その条項例を図 5-7 に示す．

委任料および支払方法の条項例

甲は，乙に対し，本件業務の対価として別紙委託料目録記載の委託料及び付帯費用を同目録記載の方法で現金にて支払う.

委任料目録の例

委任料目録
1. コンサルテーション料
プロジェクト企画時のシステム化計画
① XXXX 円/人時
概算見積額 XXXXXXXX 円
② 支払方法
毎月 XX日締め 翌日 XX日 払い
2. システム設計料
プロジェクト計画時の全体要件定義と概算見積額
① XXXX 円/人時
概算見積額 XXXXXXXX 円
② 支払方法
毎月 XX日締め 翌日 XX日 払い
3. プログラム作成料 （成果物作成料を含む）
開発に関わる要件（ストーリー）確定，開発，テスト
① 概算見積額 XXXXXXXX 円
② 支払方法
契約締結時 XXXXXXXX 円
最終リリース後検査合格後 XX 日以内
4. 付帯費用
① 項目 旅費・交通費
通信費
② 支払方法
毎月 XX日締め 翌日 XX日 払い

図 5-6 委託料および支払方法の条項と委託料目録の例

（[Yoshida 2001]を参考に加筆修正.）

委託料変更の条項（例）

1. 次の各号に該当する場合は，乙は再見積を行って甲に対し，委託料および支払方法の変更を請求することができる.

① 本件システムの要件が変更され，計画時要件との交換，調整を行っても，その規模，工数が増加される場合
② 成果物の納入期限が変更される場合
③ 原始資料その他甲の提供すべき資料、情報の遅延，誤りにより乙の費用が増加した場合
④ イテレーション開始時の再見積もりの結果，計画時のそれ以降の原見積額が不適当であると判明した場合

2. 甲は，前項④に基づく乙の再見積により要件の交換・調整を行っても計画時のそれ以降の原見積額よりXX % 以上増加した場合，前項の請求を受けた日から ○○日以内に乙に書面で通知することにより本契約の未了部分の解約をすることができる.
この場合甲または乙はこの解約に対して相手方に対し，損害賠償その他一切の請求をなすことはできない.

図 5-7 委託料変更の条項の例

（[Yoshida 2001]を参考に加筆修正.）

アジャイル開発では，イテレーション毎に開発前に実施される要件を確認決定する．その際計画時の要件との差異は交換（トレードオフ）によって調整され開発規模，工数を予測値に納める．図 5-7 委託料変更の条項例 1.①は，その変更，追加要件が調整しても計画値を超え，規模，工数が増加する場合の対応としての条項である．同図 1.①，②，③についてはシステム開発委任契約において本来要件変更によって契約変更，委託料の変更をすべきところ，現実には委託者（乙）が顧客（甲）より委託料の変更なく，仕様の変更を要求される場合の対応として置かれる条項である．同図 1．④は開発を定額受託で実施した場合，各イテレーション開始時に行われる要件確定，再見積りによってそれ以降の工程での見積りが不適当である場合への対応である．単価方式（準委任契約）の場合は見積りと実績に差が生じても清算できるので必要ない．

これらの条件を満たし，契約締結することで，情報処理推進機構のアジャイル開発モデル契約の課題であるリリース毎の契約による実務の負荷は軽減されると考えられる．

5.4 まとめ

本章の視点は，アジャイル開発の契約はどのように締結されるべきかにあった．そこで経済産業省の開発モデル契約，情報処理推進機構のアジャイル開発のモデル契約を検討した．

そしてアジャイル開発では困難とされていた計画時の規模見積り・工数見積りを実現することで，プロジェクト計画時に全体の締結できる契約モデルを提案した．また，本研究で，日本におけるアジャイル開発にも委託・受託での請負契約を適用できる可能性があることを確認した．

アジャイル開発はいまだ日本では適用率が低く，その契約実態明確ではない．今後は，アジャイル開発の契約実態を調査し，さらにその分析結果からさらに日本の現状に適したモデルの考察を進めていきたい．また，本契約モデルの評価を実務者または専門家から得ることを次の課題と考えている．

第 6 章　結言

本研究では，日本でのアジャイル開発の適用施策に焦点をあてた．そして，主にソフトウェア開発は受託開発で遂行され，開発プロセスとしてはウォーターフォール型が主流とされる日本の IT 業界でのソフトウェア開発に関する実態調査を実施した．その上で，「アジャイル型開発プロジェクトを受託開発で推進するための解決策」として，見積り方式と契約モデルを提案し，その有効性を検証した．

研究の目的は以下のとおりであった．

（1）日本でのアジャイル開発の阻害要因を明確にする．
（2）アジャイル開発での計画時における現実的な見積り方式を提案する．
（3）アジャイル開発において委託側と受託側が合意できる契約モデルを提案する．

本論文では，第 1 章で研究全体の方向性と各研究の関連について述べた．第 2 章では，関連する既存の研究について調査し，本研究の視点を明確にした．

「上記（1）」に関する研究成果の学術的貢献としては，第3章において，IT企業がアジャイル開発を適用する上で阻害要因となる要素を，アンケート調査し，仮説型検証，探索型検証により分析した．その結果，分析結果からは，IT企業組織ではウォーターフォール型開発について環境成熟度が高い傾向にあること，アジャイル開発については多様であることが明らかになった．また，アジャイル開発の阻害要因として，受託契約締結の難しさ，並びに見積り方法の認知度の低さ，顧客のアジャイル開発に対する認知度の低さが主な要因であることが明確化した．さらに，重回帰分析からは，組織としてアジャイル開発を推奨されているか否かが，売上比率と深く関係することを確認した．このことから実務への示唆として，アジャイル開発の売上比率向上のための次の知見を示した．

- 組織としてアジャイル型開発を推奨すること
- 共通の見積り手法により受託契約締結を可能にすること
- 顧客のアジャイル型開発に対する認知度向上させること

引き続く章の研究では，ここで明確化した課題のうち，ソフトウェアエンジニアリング

の見地から見積り方式と契約モデルによる解決策を提案した.

課題としては，対象とする組織の標本数を増やし調査すること，回答データに対して因子分析など他の切り口で分析を行うことを挙げる.

「上記（２）」に関する研究成果の学術的貢献としては，第 4 章において，アジャイル開発の見積りの種類とプロセスを明確にした．そして新規受託開発プロジェクトにおける予測と実績の誤差の測定からストーリーサンプリング見積り方式の有用性を証明した．また，アジャイル開発プロジェクトの過去実績データを相関分析することでストーリーサンプリング見積り方式の有効性を検証した．実務への示唆としては，アジャイル開発にこれから取り組む企業に対して，ストーリーサンプリング見積り方式を受託開発において有用性のある見積り方法として提示した．さらに，今後アジャイル開発の実績値を構築することで，より効率的で有効性のある見積りが可能であることを提示した.

今後の課題としては，複数の企業データによるより精度の高い分析を検討している.

さらに，アジャイル開発プロジェクトにおいてプロセス以外に影響する顧客とのコミュニケーションやチームのスキルレベルの，見積り精度や生産性，さらに変更率・製品品質への係わり方を更なる研究としたい.

「上記（３）」に関する研究成果の学術的貢献としては，第 5 章において，「アジャイル開発の契約はどのように締結されるべきか」の視点から，経済産業省の開発モデル契約，情報処理推進機構のアジャイル開発のモデル契約（案）の適用課題を明確にした.

そしてアジャイル開発で計画時の規模見積り・工数見積りを実現することにより契約を締結するための概算見積を実現するモデルを提案した．さらに，実務への示唆として，実際の「システム開発委託契約書」の委託料条項例を提案することで具体的解決法を提示した.

次の課題としては，本契約モデルの評価を実務者または専門家から得ることと，日本でのアジャイル開発における契約実態調査の実施を検討している．アジャイル開発はいまだわが国ではシェアが少ないが，アジャイル開発の契約実態を調査した上で，その分析結果から日本の現状に適するモデルの考察を進めたい.

謝辞

本論文の作成にあたっては，多くの方々にご支援，ご指導を賜りました．研究面でのご指導とともに論文作成にあたって懇切なご指導を頂いた筑波大学大学院　ビジネス科学研究科の木野泰伸准教授に深く感謝申し上げます．

また，副指導教官をお引き受け頂き，適格なアドバイスを頂戴しました筑波大学大学院ビジネス科学研究科の津田和彦教授，吉田健一教授，放送大学の中谷多哉子教授に深く感謝申し上げます．

そして，プロジェクトデータの提供と，検証へのご協力いただいたシー・アイ・アンド・ティー・パシフィック株式会社の上田様，河村様，株式会社プロビズモ女鹿田様，大國様，松川様に心より感謝申し上げます．

加えて，アンケート調査にご協力いただいた IT 企業の皆様にも厚くお礼申し上げます．

学位論文を構成したジャーナル論文は，情報処理学会に投稿採録された論文を加筆修正，または，プロジェクトマネジメント学会の国際会議 ProMAC で採択され発表した論文を和訳の上，加筆修正し書き直したものであります．特に情報処理学会において懇切な助言を頂いた査読委員各位に心から感謝申し上げます．

あらためて，お世話になりました皆様に深く感謝申し上げます．

参考文献

[Aoki 2003] 青木孝則, 高橋宗雄, 下村隆夫, プロジェクトマネジメント学会研究発表大会予稿集 2003（秋季）, pp.102-106, 2003.

[Adachi 2011] 足達直, 平野建一, 疋田久子, 山中敦, 田中恵美, アジャイル開発における品質管理の取り組みと評価, プロジェクトマネジメント学会誌, Vol.13, No.2, pp.24-29, 2011.

[Aiiso 2011] 相磯正司, 湯浅耕季, 鈴木圭一, ソフトウェア開発における統計的プロジェクト管理手法の導入と実践, 富士フイルム研究報告, No.56, pp. 31-34, 2011.

[Albrecht 1979] A. J. Albrecht, “Measuring Application Development Productivity,” Proceedings of the Joint SHARE, GUIDE, and IBM Application Development Symposium, Monterey, California, October 14-17, IBM Corporation, pp. 83-92, 1979.

[Boehm 1986] Barry Boehm, A spiral model of software development and enhancement, ACM SIGSOFT Software Engineering Notes, Volume 11 Issue 4, pp.14-24, 1986.

[Boehm 2003] Barry Boehm, Richard Turner, Balancing Agility and Discipline, Addison-Wesley Professional, 2003.

[Boehm 2007] Barry Boehm, Software Engineering, John Wiley & Sons, 2007.

[Carnegie-Mellon 2010] Carnegie Mellon Software Engineering Institute, Consideration for Using Agile in DoD Acquisition, Carnegie Mellon Software Engineering Institute, 2010.

[Endo 2005] 遠藤雄一, XP 開発手法におけるプロジェクトマネジメント, プロジェクトマネジメント学会誌, Vol.7,No.3, pp.58-63, 2005.

[FAR 2005] General Services Administration Department of Defense National Aeronautics and Space Administration, Federal Acquisition Regulation, https://www.acquisition.gov/sites/default/files/current/far/pdf/FAR.pdf

[Forester 2010] Forester, メインストリームとなったアジャイル型開発, Forrster2010 調査, 2010.

[Fujinuki 2006] 藤貫美佐, 西尾敏郎, 端山毅, ソフトウェア開発プロジェクトにおける生産性データ活用についての一考察, プロジェクトマネジメント学会誌, Vol.8, No.4, pp.3-6, 2006.

[Furuyama 1994] 古山恒夫, ソフトウェア開発データと統計分析, 日本計算機統計学会大

会論文集, Vol.8, pp.19-22, 1994.

[Furuyama 2005] 古山恒夫, プロジェクトデータ分析の指針と分析事例, 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 Journal , Vol.1, No.3, pp.6-13, 2005.

[Hatsuda 2006] 初田賢司, ソフトウェア開発における機能改良プロジェクトの見積もり方法,プロジェクトマネジメント学会誌, Vol.8, No.4, pp. 21-24, 2006.

[Hatsuda 2013] 初田賢司, 本当に見える見積もり技術, 日経 BP 社, 第 2 版, 2013.

[Hattori 2006] 服部昇, 福島千鶴, 山本修一郎, ソフトウェア開発プロジェクトの大きさと生産性の関係に関する考察, プロジェクトマネジメント学会誌, Vol.8, No.4, pp.13-16, 2006.

[Hosokawa 2004] 細川宣啓, 上流フェーズでの品質メトリクス・データ測定, プロジェクトマネジメント学会誌, Vol.6, No.6, pp.16-21, 2004.

[Hosokawa 2005] 細川宣啓, プロジェクトの健全性を測定するープロジェクト上流フェーズにおける 測定ベースの方針決定, プロジェクトマネジメント学会研究発表大会予稿集, 2005（春季）, pp.331-336, 2005.

[Hosotani 2008] 細谷和伸, ファンクションポイント法有効性検証と精度の向上に関する考察, プロジェクトマネジメント学会研究発表会予稿集, 2008（春季）, pp.107-110, 2008.

[Ichiyanagi 2006] 一柳晶子, 反復型ソフトウェア開発へのプロジェクトマネジメント手法の適用 : リスク軽減と二次リスク, プロジェクトマネジメント学会研究発表会予稿集, 2006（春季）, pp.143-148, 2006.

[Ikoma 2008] 居駒幹夫, 大島真幸, 谷田耕救, 大場美智子, 酒井三四郎, ソフトウェア開発プロジェクト,開発組織のアジリティ計測, 情報処理学会研究報告ソフトウェア工学研究会報告, 2008(93), pp.17-24, 2008.

[Ito 2009] 伊東暁人, ソフトウェア開発における工学的技法導入に関する考察, 経営情報学会全国研究発表大会要旨集, 2009.

[IPA 2010] 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 監修, ソフトウェアデータ白書 2010-2011, 日経 BP 社, 2010.

[IPA 2012] 情報処理推進機構ソフトウェア・エンジニアリング・センター監修, 非ウォーターフォール型開発の普及要因と適用領域の拡大に関する調査, 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, 2012.

[J. Sutherland 1995] Jeff Sutherland, Ken. Business object design and implementation, OOPSLA '95 Workshop Proceedings. The University of Michigan. p.118.ISBN3-540-76096-2, 1995.

[J. Sutherland 2013] Jeff Sutherland, Ken Schwaber, Scrum Guides, ScrumGuides.org., 2013.

[J. Rasmusson 2010] Jonathan. Rasmusson, The Agile Samurai, O'reilly, 2010.

[Kaneko 2009] 金子美和, 適応型反復開発におけるプロジェクトマネジメント, プロジェクトマネジメント学会誌, Vol.11, No.5, pp.22-27, 2009.

[Kaneko 2004] 金子由紀, 細川宣啓, 品質メトリクス測定とリスク・マネジメントの連動に関する一考察, プロジェクトマネジメント学会研究発表大会予稿集, 2004（春季）, pp.199-203, 2004.

[Kanzaki 2006] 神崎光司, 田村良二, 山本良子, 実績データ活用による見積プロセスの改善, プロジェクトマネジメント学会 Vol.8, No.4, pp.7-12, 2006.

[Kaplan 1997] Kaplan, R.S. and Norton, D.P 著, 古川武男 訳, バランススコアカード - 新しい経営指標による企業改革 - , 生産性出版, 1997.

[K. Beck 1999] Kent Beck, Extreme Programming Explained: Embrace Change. Addison-Wesley, 1999.

[Kimura 2009] 木村信男, ROI の考え方に基づく開発規模管理, プロジェクトマネジメント学会誌, Vol.11, No.3, pp.9-12, 2009.

[Kodama 2004] 児玉公信, UML モデリングの本質, 日経 BP 社, 2004.

[Kodama 2008] 児玉公信, ユースケースの使い方に関する提案（一般）,情報処理学会研究報告・情報システムと社会環境研究報告, 2008(81), pp.39-44, 2008.

[Kurashige 2006] 倉重誠, 原田晃, 業務ソフトウェア開発における早期見積技法の適用, プロジェクトマネジメント学会発表大会予稿集, 2006（春季）, pp.184-186, 2006.

[Kurashige 2007] 倉重 誠, 原田 晃, 三島紀子, 竹田佳史, 中野和哉, 業務ソフトウェア開発におけるファンクションポイント早期見積技法, プロジェクトマネジメント学会誌, Vol.9, No.3, pp.7-10, 2007.

[Manifesto 2001] Agile Software Manifest, 2001. http://agilemanifesto.org/iso/en/

[Makuta 2008] 幕田行雄, 福地 豊, 谷川晃一, 江口良二, 渡辺嘉也, 村瀬武志, 石谷 靖, 塩田英雄, CoBRA 法に基づくソフトウェア開発プロジェクトの見積りモデル構築手順の改善, プロジェクトマネジメント学会誌, Vol.10, No.6, pp.25-30, 2008.

[Matsukawa 2004] 松川文一, 楠本真二, 井上克郎, 英繁雄, 前川佑介, ユースケースポイント計測支援ツールの実装とその適用, 情報処理学会研究報告, 2004(30), pp. 91-98, 2004.

[Matsushima 2010] 松島桂樹, IT 経営におけるアジャイル開発手法の価値, 研究年報経済学/

東北大学, 通号 254, pp.35-48, 2010.

[Matsushima 2009] 松島桂樹, IT 投資マネジメントの変革, 白桃書房, 2013.

[M. Cusumano 2003] Michael A. Cusumano, Synchronize-and-stabilize ～ソフトウェア開発の成功例に学ぶ～, CSK EXPRESS, vol.112, pp.23-39, 2003.

[M. Cohn 2005] Mike Cohn, Agile Estimation and Planning, Pearson Edition, 2005.

[METI 2012] 経済産業省商務情報政策局,情報システムの信頼性向上のための取引慣行・契約に関する研究会～情報システム・モデル取引契約書～, 2012.

http://www.meti.go.jp/policy/it_policy/keiyaku/model_keiyakusyo_gaiyou.pdf,

[METI 2008] 経済産業省, 情報システムの信頼性向上のための取引慣行・契約に関する研究会 ～情報システム・モデル取引契約書～（パッケージ，SaaS/ASP 活用，運用・保守＜追補版＞, 2008.

[Minpou 2011] 民法条文解説，民法第 91 条,

http://www.minnpou-sousoku.com/category/article/5/91.html

[Moriya 2009] 森谷文利, 土橋俊寛, 西村健, 高橋 茂, 中村宏美, ソフトウェア産業における契約について, SEC journal, Vol,5,No.6, pp.362-367, 2009.

[Murakoshi 2009] 村越英樹, PSP を基盤とするソフトウェア開発プロセス改善トレーニング実施について, 産業技術大学院大学紀要, 第 3 巻 pp.157-162, 2009.

[Naito 2010] 内藤正樹, 菊池純男, 駒谷昇一, 田中二郎, 学生によるシステム開発プロジェクトにおけるユースケースポイント法を用いた見積もりと実績の評価, 情報処理学会
第 72 回全国大会, 2010(1), pp.519-520, 2010.

[Nakamura 2011] 中村忠之, ネットビジネス進化論, 中央経済社, 2011.

[Nishida 2009] 西田宗千佳, クラウドコンピューティングウェブ 2.0 の先にくるもの, 朝日新書, 2009.

[Nishiyama 2004] 西山泰男, 米国政府調達における動機づけ契約の特徴と分類の考察 －契約形態の整理と考慮－, プロジェクトマネジメント学会予稿集 2004（春）, pp.118-123, 2004.

[Niwa 2005] 丹羽展男, IT 開発プロジェクトにおける正確な開発規模見積り手法, プロジェクトマネジメント学会研究発表会予稿集, 2005（秋季）, pp.218-223, 2005.

[Nozawa 1999] 能澤徹, インセンティブ付固定価格契約(FPI)の仕組みについて, プロジェクトマネジメント学会予稿集 1999（秋）, pp.52-55, 1999.

[Nunokawa 1997] 布川薫他著，アプリケーション開発技術，リックテレコム，1997.

[Ohfude 1992] 大筆 豊，ソフトウェアのコスト見積り技術,情報処理，Vol.33，No.8，pp.906-911, 1992.

[Ohkubo 2002] 大久保隆， 持原眞理子，米国政府調達における契約の種類とその実際 －コントラクト・マネジメントの考慮点－，プロジェクトマネジメント学会誌，Vol.4，No.2，pp.3-7, 2002.

[Ohsugi 2006] 大杉 直樹，大平 雅雄，松村 知子，森崎 修司，玉田 春昭，松本 健一， 産官学連携における参加者の興味についての対応分析，情報科学技術レターズ 5, pp.313-315, 2006.

[Putnam 1978] Putnam, H. Lawrence, "A General Empirical Solution to the Macro Software Sizing and Estimating Problem". IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, SE-4(4), pp.345-361, 1978.

[Satoh 2013] 佐藤 創，システム開発現場のプロジェクトマネジメント教科書 2013 年版 -情報処理技術者試験 プロジェクトマネージャ試験範囲全網羅テキスト-，みんなの SE 創研, 2013.

[S. Bhalerao 2009] S. Bhalerao, Incorporating Vital Factors in Agile Estimation through Algomic Method, International Journal of Computer Science and Application (IIJCSA), 6(1), pp.85-97, 2009.

[S. Bhalerao 2011] S. Bhalerao, Agile Estimation using CAEA: A Comparative Study of Agile Projects, International Conference on Computer Engineering and Application (IPCSIVol.2, pp.76-82, 2011.

[Shibao 2004] 芝尾芳昭，越智 匡，プロジェクトマネジメント成熟度評価 法の研究について，プロジェクトマネジメント学会誌，Vol.7, No.6, pp.34-46, 2005.

[Shimanaka 2004] 島中一俊，古賀順二，寺澤啓司，ユースケースから導いたテストケース数による規模見積もりに関する研究，プロジェクトマネジメント学会研究発表会予稿集，2004（春季）, pp.366-370, 2004.

[S. Keaveney 2006] S. Keaveney, Cost Estimation in Agile Development Projects, ECIS, 2006.

[Takahashi 2010] 高橋信弘，日本のソフトウェア産業の国際競争力に関する一考察，経営研究,Vol.60, No.4, 151-167, 2010.

[Umemoto 2012] 梅本大祐，アジャイル開発向けモデル契約案について 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, 2012.
http://www.ipa.go.jp/files/000005404.pdf
[Versionone 2010] Versionone, “State of Agile Survey 2009”, 2010.
http://www.versionone.com/pdf/2009_State_of_Agile_Development_Survey_Results.pdf
[W. Humphrey 1991] W. Humphrey 著，藤野喜一 監訳，ソフトウェアプロセス成熟度の改善，日科技連出版社, 1991.
[W. Humphrey 2009] W. Humphrey 著，松本正雄 監訳，パーソナルソフトウェアプロセス技法，共立出版, 2009.
[W. Royce 1970] W. W. Royce, "Managing the Development of Large Software Systems", Proceedings of IEEE WESCON 26 (August), pp.328-338, 1970.
[Yoshizaki 2012] 吉崎浩二，ソフトウェア開発マネジメントに関する考察，上武大学ビジネス情報学部紀要，第 11 巻第 2 号, 2012.
[Yoshida 2001] 吉田正夫，ソフトウェア取引の契約ハンドブック，共立出版株式会社, 2001.
[Yoshida 2013] 吉田知加，木野泰伸，上田善行，アジャイル開発の見積プロセスの検討，プロジェクトマネジメント学会研究発表大会予稿集 2013（秋季）, pp.240-245, 2013.