アジャイル型開発における
プラクティス活用事例調査

調査報告書

～調査編～

平成 25 年 3 月 19 日

独立行政法人情報処理推進機構
技術本部 ソフトウェア・エンジニアリング・センター

はじめに

独立行政法人情報処理推進機構 技術本部 ソフトウェア・エンジニアリング・センターでは、「日本のソフトウェア産業の競争力を強化すること」、「エンジニア一人ひとりが生き生きと働ける環境を作ること」を目指すべきゴールとして、「日本国内における非ウォーターフォール型開発の普及促進」と「わが国のソフトウェア産業の特性に照らした非ウォーターフォール型開発の課題解決」を目標に掲げ、平成 21 年度から調査検討に取り組んで参りました。その中で、開発手法の実践的な解説書と実例やテクニック等の工夫を取りまとめた Tips 集を、広く一般に公開することか求められていることが分かりました。本事業では、アジャイル型開発におけるプラクティス活用の事例調査を実施し、調査結果を報告書として取りまとめました。

報告書は以下の 2 部構成となっており、本報告書は「Ⅰ部 調査編」にあたります。

Ⅰ部 調査編
対象読者を今後の調査業務にあたる人物に設定して、調査全体の概要（調査の指針や手順）を簡潔にまとめました。

Ⅱ部 アジャイル型開発におけるプラクティス・リファレンス・ガイド
対象読者を「アジャイル型開発に関心を持ち始めて、自分で試してみようと思っている」初心者層に設定し、事例調査から得られた知見を元に、利用者の利用目的や課題解決等に役立つようプラクティスのレファレンスをまとめました。

本調査事業は、「アジャイル型開発におけるプラクティス活用事例調査事業」として、株式会社永和システムマネジメントに委託し、実施致しました。

アジャイル型開発におけるプラクティス活用事例調査
【調査報告書　調査編】
独立行政法人情報処理推進機構

目次

1. 背景と目的

1.1. 調査に至る背景

アジャイル型開発を中心とする非ウォーターフォール型開発は、従来のウォーターフォール型開発の課題を解決するソフトウェア開発手法として近年期待されている。また、ビジネス環境の変化に俊敏に対応でき、ソフトウェアの開発着手から市場投入までに要する期間を短縮する手法として注目され、Web アプリケーション開発等の分野で普及し始めている。

独立行政法人情報処理推進機構 技術本部 ソフトウェア・エンジニアリング・センター (IPA/SEC) では、「日本のソフトウェア産業の競争力を強化すること」、「エンジニア一人ひとりが生き生きと働ける環境を作ること」を目指すべきゴールとして、「日本国内における非ウォーターフォール型開発の普及促進」と、「わが国のソフトウェア産業の特性に照らした非ウォーターフォール型開発の課題解決」を目標に掲げ、平成 21 年度から調査検討に取り組んできた。

平成 21 年度・平成 22 年度の調査で、5 つの「重点課題」と 4 つの「試行領域」が明らかになった。

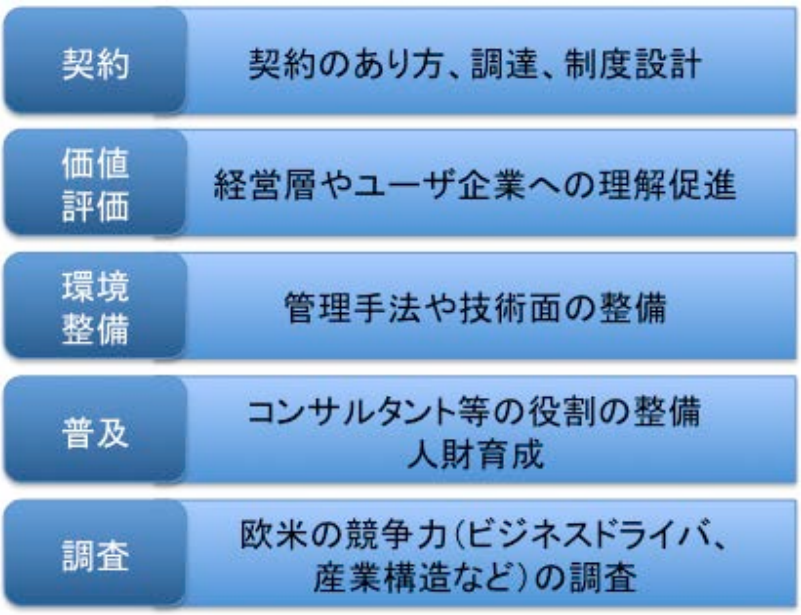

図 1-1 5 つの重点課題

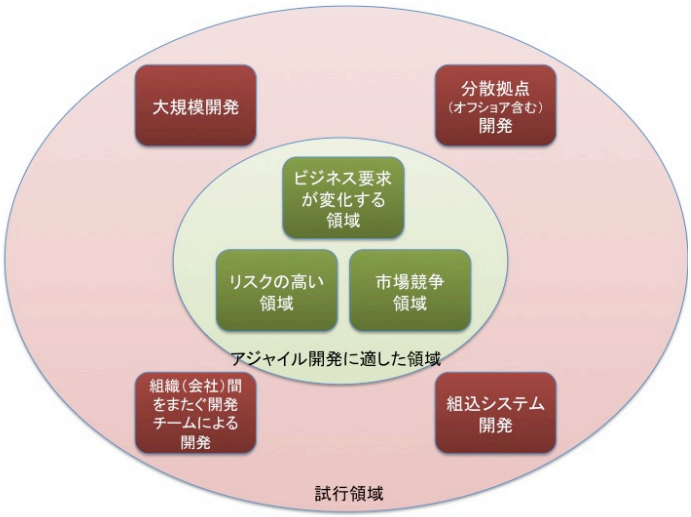

図 1-2 4 つの試行領域

重点課題のうち、以下の 2 点については、事例数が少なく十分な調査が行えなかった。

(1)「管理手法や技術面の整備」における、中規模、大規模開発プロジェクトに適用する開発技術及び管理技術の事例調査
(2)「欧米の競争力の調査」に関連した「普及要因」の調査

また、試行領域についても、非ウォーターフォール型開発による経験が十分には蓄積されておらず、具体的な施策の提示には至っていない。

平成 23 年度の調査では、残された重点課題の解決及び試行領域への適用拡大のため、以下の調査を実施した。

- 国内外の非ウォーターフォール開発を対象とした調査
- 中規模や大規模プロジェクトでの非ウォーターフォール型開発の適用状況や適用範囲を拡大するための課題についての調査

この調査結果を受けて、次のような課題が新たに提示された。

- 現場で活用できる具体的な事例の「レファレンス」がない
- 自プロジェクトの状況に適合する参考情報が提供されていない

この課題に対して、「現場導入のナレッジ収集と活用するための Tips 集作り」が提言されている。 すなわち、開発手法の実践的な解説書と実例やテクニック等の工夫を取りまとめた Tips 集を、広く一般に公開することが求められている。

1.2. 調査の目的

国内外の実際の非ウォーターフォール型開発の代表的な手法であるアジャイル型開発の適用プロジェクトにおける「プラクティス」（チーム運営からプログラミングまでの幅広い手法、活動など）の具体的な活用事例を幅広く収集し、開発対象ソフトウェアの特徴や開発組織・プロジェクトの状況（以下「コンテキスト」とする）の違いの観点で分類・整理した上で、ソフトウェア開発の現場で利用できるレファレンスのためのガイドとして取りまとめる。

ガイドのメインターゲットの読者層は、「アジャイル型開発に関心を持ち始めて、実際に自ら試してみようと思っている人」に定める。

海外書籍の翻訳に留まらず、日本の現場での実践事例を広く紹介する（特に、プロジェクト独自の工夫や日本国内でアジャイル型開発を実践する際に留意しなければならない点を紹介する）ことで、国内でのアジャイル型開発のより広い普及を目指す。

1.3. 国内のアジャイル型開発の背景・状況

近年、世界的にアジャイル型開発が主流になってきている。

Forrester2010 調査「メインストリームとなったアジャイル型開発（“AGILE DEVELOPMENT: MAINSTREAM ADOPTION HAS CHANGED AGILITY”）」によると、“2009 年の第 3 四半期に実施した調査で、約 35%の開発プロジェクトでアジャイル型開発手法を採用しているという結果が示されている … アジャイルと非アジャイルの技術とプラクティスを組み合わせて、より大きな組織にあうようにハイブリッドにすることに苦闘している。”と報じられている。
※ 2009 年の調査で、はじめてアジャイル型開発はウォーターフォール型開発を超えた。

こういった状況の中で、国際競争力の強化のため、世界的に主流になっているソフトウェア開発手法であるアジャイル型開発に日本でも取り組む必要がある。しかし、現状、日本では「普及が遅れており、ようやく認知されはじめた」段階であるとされている（“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」”）。

1.4. 国内のアジャイル型開発の課題

日本でアジャイル型開発の普及が阻害されている要因として、IPA の 2011 年度の非ウォーターフォール型開発の調査では、海外と国内では下記のような環境の違いがあると述べられている。

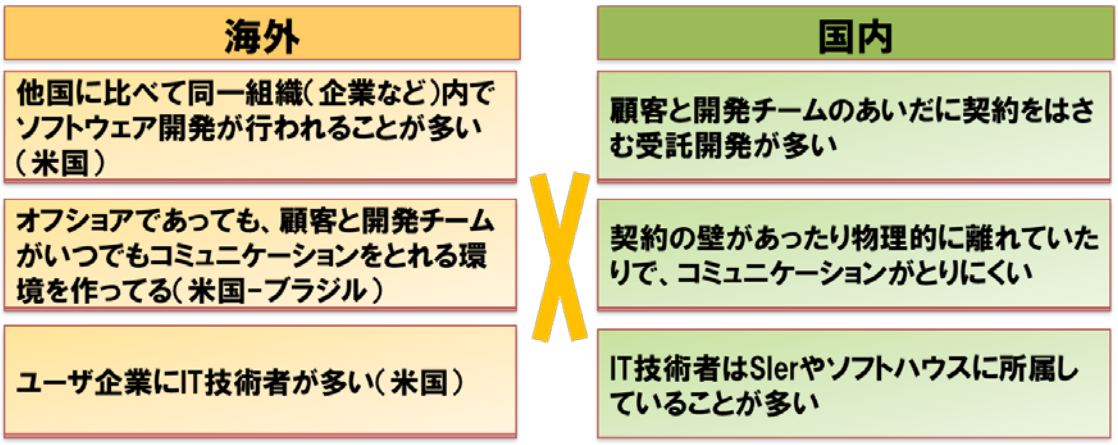

図 1-3 海外と国内の環境の違い

海外と国内でこのような環境の違いがあり、海外の書籍に書かれていることをそのまま実践しても、うまくいかないことがある。日本におけるアジャイル型開発の普及を図るため、本調査では、このような環境の違いを踏まえた国内のプラクティス利用例（独自の工夫・留意点）を紹介する。

2. 調査方法

2.1. 調査の流れ

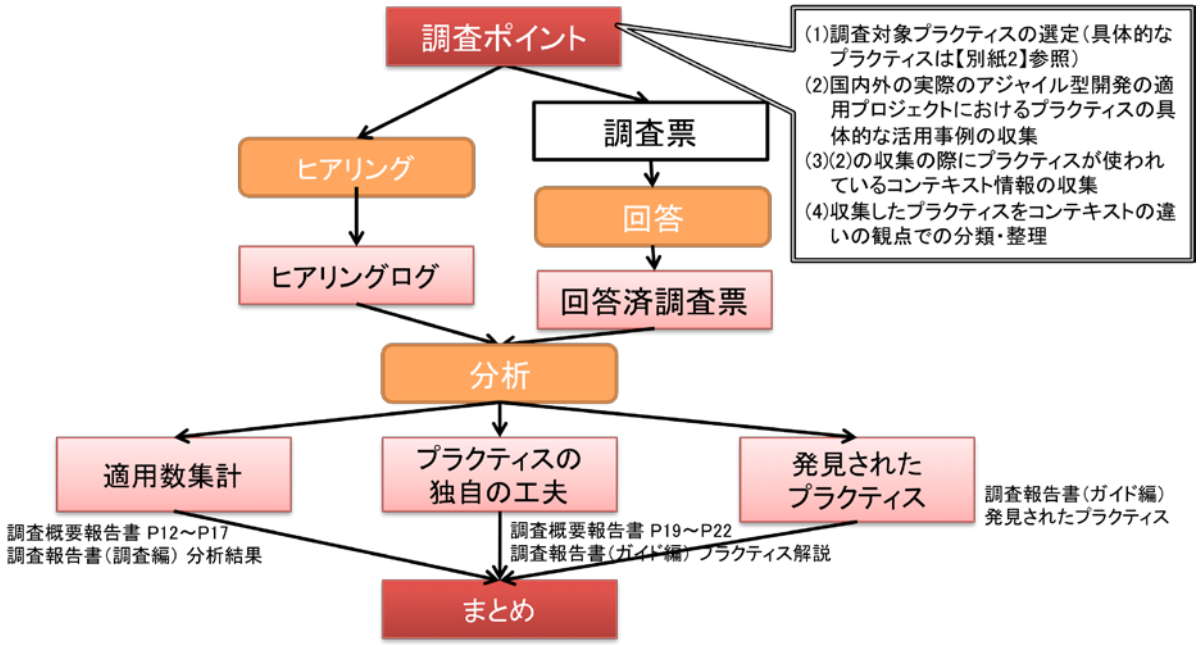

図 2-1 調査の流れ

本調査は、図 2-1 のような流れで各調査対象について実施したものである。以下にその流れを記述する。

1. 図 2-1 の通りに調査ポイントを設定した。
2. 調査に必要な項目を洗い出した上で 2 種類の調査票を作成し、ヒアリングの前に調査票を調査対象各社に送信した。
   - プラクティスを実践した事例のコンテキスト情報について、回答収集を行った。
   - プラクティス一覧に、適用度、評価、独自工夫や導入ポイントについて、回答収集を行った。

3. 事前に調査票への回答を可能な限り依頼した後に、先方を訪問し内容について個別ヒアリングを実施した。
   - 適用度や評価が高く、独自工夫があるものを中心に詳細のヒアリングを行った。
   - アンチパターンの収集のために、部分的に適用しているプラクティスや途中で中止したプラクティス、途中で方法を変更したプラクティスについてもヒアリングを行った。
   - ヒアリング結果から調査票の回答にフィードバックを実施した。
4. 調査票+ヒアリングの内容を元に分析を行った。
   - 各事例の結果を集計してのプラクティス適用数の傾向
   - 各事例から収集した個別の独自の工夫、アンチパターンのまとめ
   - 各事例から発見されたプラクティスの抽出
5. 上記の結果を踏まえてガイドにまとめた。

なお、作成した調査票は表 2-1 のような目的と内容になっている。調査票の具体的な内容については、「付録 1：調査票」を参照のこと。

調査票種類	質問内容等
プラクティス状況アンケート	アジャイルプラクティス活用事例の背景・コンテキストについての回答収集
プラクティス調査アンケート	アジャイルプラクティスの適用度・評価・独自の工夫や導入ポイントについての回答収集

表 2-1 調査票の種類と目的

2.2. 調査方法

本調査においては、時間を短縮することで被調査者の負担が軽減すること、また統計的な分析が可能であることから、主な手段としてアンケート（質問票）を用い、特に重点的な項目については対面によるインタビューを実施し、調査票に記述されている背景や独自の工夫を明確にした。

2.3. 調査対象の選定基準

方針 1：できるだけ多様な回答を収集するため、下記の特徴があるプロジェクトの事例を含めることとした。

方針 2：調査対象を国内の事例に絞り、アジャイル型開発実践に際しての工夫、留意点の収集を実施した。

施策 a：下記の特徴があるプロジェクトの事例を含める。

- 中大規模適用プロジェクト
本調査では、参加人数が 30 名以上のプロジェクトについて調査を実施する。
なお、「プロジェクトに参加した人数」はのべ人数ではなく、各自の参画した期間に係らず実人数を指すものとする。

- 複数手法の組み合わせプロジェクト
使用率が上位の代表的な手法である Scrum や XP だけでなく、Scrum+XP、Scrum+WF（ウォーターフォール）のように複数の手法を組み合わせたプロジェクトについて調査を実施する。

施策 b：なるべく多様なプロジェクトを調査するため、様々なシステム種別や契約形態も含めて幅広く収集対象とした。

2.4. 調査項目

2 種類の調査票のうち、プラクティス調査アンケートでは 2.4.1 対象プラクティスに挙げたプラクティスを調査対象とし、それぞれのプラクティスの適用度・評価・独自の工夫や活用ポイントについて回答を得た。
プラクティス状況アンケートでは 2.4.2 コンテキスト情報に挙げた情報を収集し、事例のコンテキストを把握した。

2.4.1. 対象プラクティス

下表の 45 のプラクティスを調査対象とする。なお、各プラクティスの詳細については、『ガイド編』を参照のこと。

調査対象のプラクティスを選定するにあたり、以下の資料を参照した。

- 『What Do We Know about Scientific Software Development's Agile Practices?』（http://www.computer.org/csdl/mags/cs/2012/02/mcs2012020024-abs.html）（Slethold, M.T. 他, 2012）
- 『CHAOS Manifesto』（Standish Group , 2011）
- 『非ウォーターフォール型開発に関する調査』（IPA, 2009）

上記の資料に掲載されているプラクティスをすべて洗い出し、重複しているものや類似のもの（例：XP における計画ゲーム、Scrum におけるスプリント計画）はまとめて扱うこととした。

また、これらのプラクティス以外に、新たに「インセプションデッキ」、「プロダクトオーナー」、「アジャイルコーチ」、「外部コンサルタント」を追加した。前年度の「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」の結果から導かれた提言にも「参加型意志決定手法の紹介・導入支援」が挙げられており、「インセプションデッキ」、「プロダクトオーナー」は調査に値するプラクティスであると判断した。また、同提言に「コンサルタント活用」が挙げられており、「アジャイルコーチ」、「外部コンサルタント」についても同様に調査対象プラクティスとした。

このようにして洗い出したプラクティスを 3 つのカテゴリと 10 のサブカテゴリに分類して、下表のように整理した。

カテゴリ	サブカテゴリ	日本語名	英語名
プロセス・プロダクト	プロセス	リリース計画ミーティング	Release planning to release product increments
		イテレーション計画ミーティング	The Planning Game
		イテレーション	Iteration
		プランニングポーカー	Planning poker to estimate tasks during sprint planning
		ベロシティ計測	The project velocity is measured
		日次ミーティング	Short daily meeting to resolve current issues
		ふりかえり	end-of-iteration retrospectives ／print retrospective to learn from previous sprint
		かんばん	Kanban
		スプリントレビュー	Sprint review meeting to present completed work
		タスクボード（タスクカード）	Task board (Task card)
		バーンダウンチャート	Burn down chart to monitor sprint progress
		柔軟なプロセス	Flexible process
	プロダクト	ユーザーストーリー	User stories are written
		スプリントバックログ	Mutual commitment to sprint backlog between product owner and team
		インセプションデッキ	Inception Deck
		プロダクトバックログ（優先順位付け）	Priorities (product backlog) maintained by a dedicated role (product owner)
	フィードバック	迅速なフィードバック	Rapid feedback

表 2-2 プラクティス一覧(1)

カテゴリ	サブカテゴリ	日本語名	英語名
技術・ツール	設計開発	ペアプログラミング	All production code is pair programmed
		自動化された回帰テスト	Automated regression test
		テスト駆動開発	Code the unit test first
		ユニットテストの自動化	Unit Test Automation
		受入テスト	Acceptance tests are run often and the score is published
		システムメタファ	System metaphor
		スパイク・ソリューション	Create spike solutions to reduce risk
		リファクタリング	Refactor whenever and wherever possible／Constant refactoring
		シンプルデザイン	Simplicity in design
		逐次の統合	Only one pair integrates code at a time
		継続的インテグレーション	Continuous Integration／Integrate often
		集団によるオーナーシップ	Use collective ownership
		コーディング規約	Code written to agreed standards
	障害対応	バグ時の再現テスト	When a bug is found, tests are created
	利用ツール	紙・手書きツール	Analog Tools

表 2-3 プラクティス一覧(2)

カテゴリ	サブカテゴリ	日本語名	英語名
チーム運営・組織・チーム環境	人	顧客プロキシ	Customer Proxy
		オンサイト顧客	The customer is always available／Constant user interaction
		プロダクトオーナー	Product Owner
		ファシリテータ（スクラムマスター)	Development process and practices facilitated by a dedicated role (Scrum master)
		アジャイルコーチ	Agile Coach
		自己組織化チーム	Team members volunteer for tasks (self-organizing team)
		ニコニコカレンダー	Niko-niko Calendar ／Smiley Calendar
	進め方	持続可能なペース	Set a sustainable pace
	組織導入	組織に合わせたアジャイルスタイル	Right agile style for their organization
	ファシリティ、ワークスペース	共通の部屋	Give the team a dedicated open work space
		チーム全体が一つに	One whole team
		人材のローテーション	Move people around
		インテグレーション専用マシン	Set up a dedicated integration computer

表 2-4 プラクティス一覧(3)

2.4.2. コンテキスト情報

以下のコンテキスト情報を収集した。
アンケート形式ですべてのコンテキスト情報を収集することは、調査対象企業の負担の面から困難であったため、ヒアリングによって補完した。

(1)調査対象プロジェクトの概要が分かる情報
①企業プロフィール（企業名、組織名等）
②アプリケーションシステム名または利用ソフトウェア名
③対象ドメイン（アプリケーションシステムの種別、優先した IT 戦略やアーキテクチャ等）
④使用した業務パッケージ等
(2)調査対象プロジェクトの推進体制が分かる情報
①プロジェクト概要（開発規模、チーム人数、開発期間、利用業務、言語等）
②開発プロジェクト内の責任分担
③プロジェクトの進め方
（ライフサイクルモデル、プロセス間の連携、ドキュメンテーション、プロジェクトマネージャの調整等）
④開発チーム編成
（メンバーのスキル、研修・訓練、ロケーション、コミュニケーション手段・頻度、顧客の関与度合い、コーチ/コンサルタントの有無を含む）
⑤契約形態
⑥開発時の課題と検討・実施された解決策
（アーキテクチャやデータモデルの変更のタイミングや対応策等）
(3)調査対象プロジェクトで使用した開発方法論やツール等に関する情報
①使用した開発手法（Scrum、XP 等）
②使用したツール群
③スコープ管理（機能分割の方法等）
(4)調査対象プロジェクトにおける下記の分野の手法や問題回避の工夫等に関する情報
①要求形成／追跡
②設計
③構築
④ドキュメンテーション（作成文書の種類、作成の責任、作成のタイミング等）
⑤テスティング／V&V（Verification and Validation）
⑥メンテナンス
⑦進捗管理（特に工事進行基準への対応）
⑧品質管理
⑨生産性向上
⑩システム監査

(5)調査対象プロジェクトの企画・結果等に関する情報

①アジャイル型開発の採用に至る経緯、判断ポイント

②成功（部分的成功を含む）／失敗の別と内容（低品質、コスト増、納期遅延等）と理由（成功の尺度も調査すること）

(6)他の開発手法と組み合わせに関する情報（該当するプロジェクトの場合のみ）

①複数手法を組み合わせたケース固有の特徴的な考え方

②複数手法を組み合わせた設計・管理方法、プロジェクト運営のテクニック等の開発技法

3. 調査結果

本節では、調査対象となった 13 社 26 事例の各事例について、その個別の情報を下表に示す。

調査先	事例番号	採用手法[*1]	特徴	システム種別	契約関係[*2]	開発言語
A 社	0	Scrum+XP		B2C サービス (広告配信)	自社開発	Java, PHP, Perl
	1	Scrum+XP		B2C サービス (広告配信)	自社開発	Ruby
B 社	2	Scrum+XP		B2C サービス (SNS)	自社開発	Java
	3	Scrum+XP		B2C サービス (メール配信)	自社開発	Java
C 社	4	XP+WF	中規模	B2C サービス (メール配信)	受託開発 (準委任)	Java
D 社	5	XP		B2C サービス (SNS)	自社開発	Java, PHP, Ruby
E 社	6	Scrum	初導入	社内システム	自社開発	C#
	7	Scrum+WF	中規模	社内システム	受託開発 (請負)	Java, COBOL
F 社	8	Scrum+WF	中規模	社内システム	自社開発	C#
G 社	9	Scrum+XP	初導入	社内システム	実証事業	Ruby
	10	Scrum+XP		社内システム	受託開発 (請負)	Ruby
H 社	11	Scrum		B2C サービス (音楽配信)	自社開発+オフショア(準委任)	Java, C#, Objective-C
	12	Scrum		B2C サービス (エンターテイメント)	自社開発+オフショア(準委任)	Java, C#, Objective-C
	13	Scrum		社内システム	自社開発+オフショア(準委任)	Java
	14	Scrum		B2C サービス (ヘルスケア)	自社開発+オフショア(準委任)	C#
I 社	15	Scrum	中規模 (組織導入)	B2C サービス (広告配信)	自社開発	Java, Objective-C

表 3-1 事例一覧(1)

調査先	事例番号	採用手法[*1]	特徴	システム種別	契約関係[*2]	開発言語
J 社	16	XP		B2C サービス(スマートフォンアプリ)	受託開発(請負)	Java
	17	XP		B2C サービス(クラウド基盤)	受託開発(請負)	Java
	18	XP		B2C サービス(クラウド基盤)	受託開発(請負)	Java
	19	XP		B2C サービス(PaaS)	受託開発(請負)	Java
K 社	20	Scrum		B2C サービス(EC サイト)	受託開発(請負)	PHP
L 社	21	Scrum+UP		社内システム	受託開発(請負)	Java
	22	Scrum+WF	大規模	社内システム	受託開発(準委任)	Java
	23	Scrum+WF		技術評価	受託開発(請負)	Java
	24	Scrum		パッケージ	自社開発+オフショア(請負)	C#
M 社	25	Scrum	大規模(組織導入)	B2C サービス(ソーシャルゲーム)	自社開発	Perl

表 3-2 事例一覧(2)

※1：XP：エクストリームプログラミング、Scrum：スクラム、
WF：ウォーターフォール、UP：統一プロセス、
もしくは、これらの手法の組み合わせ

※2：自社開発 → 自社組織内に開発部隊あり、一部パートナー(派遣)
受託開発 → 自社組織内に開発部隊なし、外部ベンダに発注している

3.1. 調査対象の特性

3.1.1. システム種別

事例をシステムの種別によって下表のように分類した。

B2C サービス	一般に広く公開して利用されるシステム
社内システム	企業内もしくは企業間に限定して利用されるシステム
パッケージ	特定業務用途向けに汎用的に利用されるシステム
技術評価	システムの利用ではなく、システム開発過程における技術面の評価を目的としたシステム

表 3-3 システム種別分類

調査対象とした事例をシステム種別で分類したところ、下図の通りとなった。
今回、調査対象とした事例の約 60%がインターネット上の B2C サービスの開発にアジャイル型開発を適用しているのに対して、社内システムは 30%弱と少ないことが分かった。
社内システムの場合は、ある程度要件を固めた上で開発を進めることができるが、B2C サービスの場合、リリースして市場やユーザーの反応を見ながら開発を進める必要がある。よって、このことから、B2C サービスのほうがアジャイル型開発のメリットをより享受できるため、自然とアジャイル型開発の採用率が高くなる傾向にあると言える。

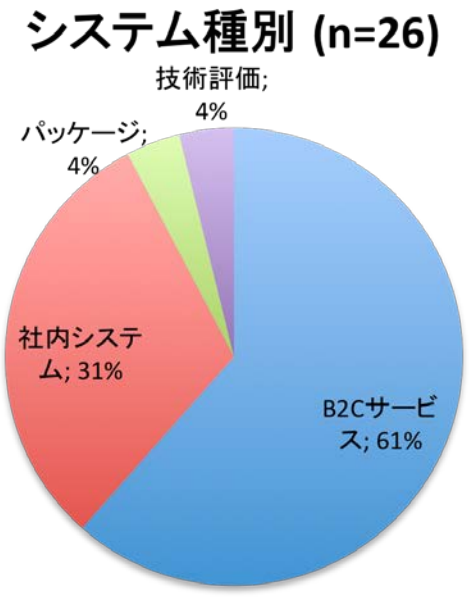

図 3-1 事例分類（システム種別）

3.1.2. 規模

事例をシステムの規模によって下表のように分類した。
規模の分類については、2011 年度に IPA が実施した「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」で用いた定義に合わせた。
規模を表す基準としては、人月やファンクションポイントなどがあるが、今回の調査では人数が増えることによりコミュニケーション面に及ぼす影響を明らかにするため、規模を表す基準として、チームの人数を採用した。
なお、チームの人数には、顧客やプロダクトオーナーといった開発メンバー以外の人数も含めている。

小規模	30 名未満のプロジェクト
中規模	30 名以上～100 名未満のプロジェクト
大規模	100 名以上のプロジェクト

表 3-4 規模別分類

調査対象とした事例をシステムの規模で分類したところ、下図の通りとなった。
30 名以下の小規模プロジェクトが 80%近くを占め、内最も人数が少ないプロジェクトが 3 名、最も多いプロジェクトが 400 名となった。

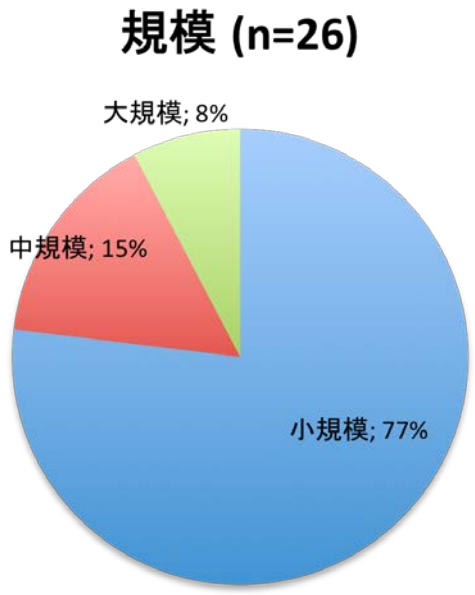

図 3-2 事例分類（規模別）

3.1.3. 手法

事例を手法によって下表のように分類した。

Scrum	主に Scrum を使ったプロジェクト
XP	主に XP(エクストリームプログラミング)を使ったプロジェクト
Scrum + XP	Scrum と XP のそれぞれの手法に対するプラクティスを組み合わせて実践しているプロジェクト
Scrum + WF	一部を Scrum、一部をウォーターフォールというように、対象としているサブシステムや機能によって使い分けているプロジェクト
Scrum + UP	全体としては UP（Unified Process）を採用しつつ、推敲フェーズ・作成フェーズで Scrum を取り入れたプロジェクト
XP + WF	一部を XP、一部をウォーターフォールというように、対象としているサブシステムや機能によって使い分けているプロジェクト

表 3-5 手法別分類

調査対象とした事例を採用されている手法で分類したところ、下図の通りとなった。
どの手法を採用しているかについては、調査先企業の申告制とした。アジャイル型開発を導入した経緯や導入を担当した外部コーチの影響によって、調査先企業毎にそれぞれこだわりのポイントがあり、どの手法を採用するかに影響を与えている。
Scrum の採用率が、Scrum と他の手法と組み合わせている事例も合わせると、4 分の 3 を超えていた。
なお、社内で様々な手法が乱立しているという例はあまりなく、会社全体で手法の統一がされている例が多く見受けられた。

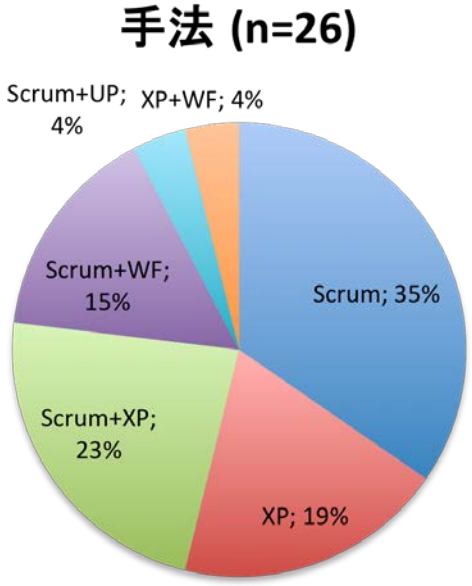

図 3-3 事例分類（手法別）

なお、本調査において調査対象事例として、Scrum と XP を採用している事例を中心に取り上げることにした理由としては、2011 年に VersionOne 社が行ったワールドワイドのアジャイル型開発実態調査で、最もよく用いられる手法として、Scrum 及び Scrum+XP の組み合わせが挙げられている（合わせて全体の 66%）ことがある。

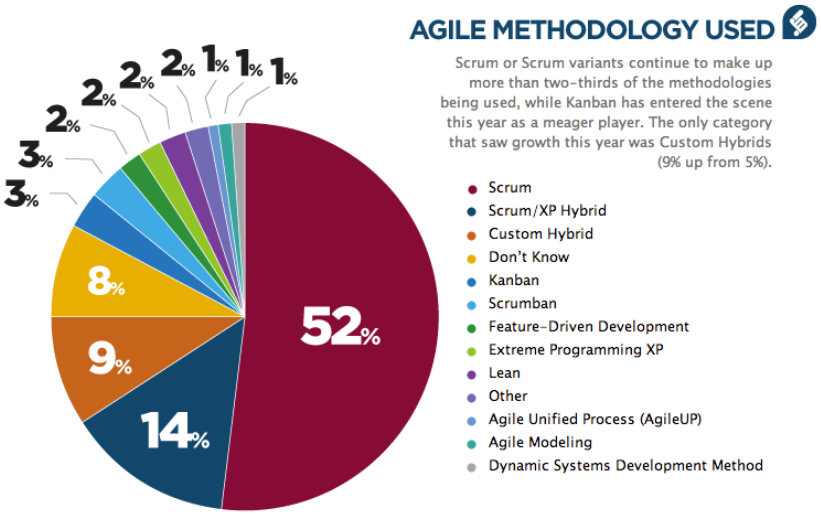

図 3-4 アジャイル型開発手法の適用率
出典：State of Agile Development Survey 2011（VersionOne 社）

両アジャイル開発手法の特徴と代表的なプラクティスを図 3-5 に示す。

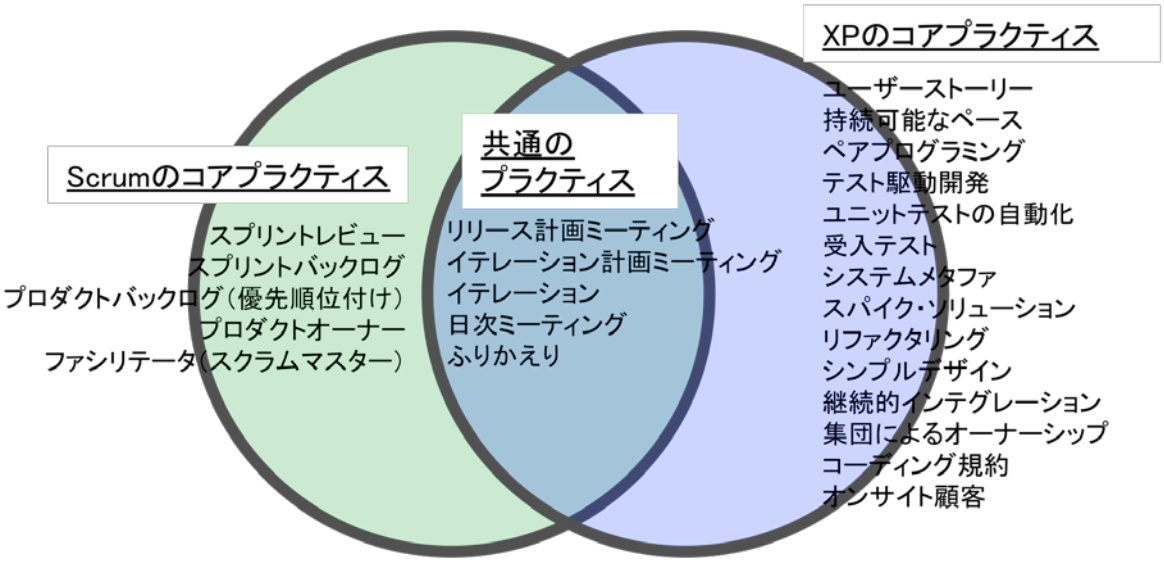

図 3-5 アジャイル型開発手法の適用率

- Scrum とは、マネジメント・プロセスをシンプルなフレームワークで提供する軽量開発手法である。
- XP とは、エンジニアリング・プロセスとマネジメント・プロセスをバランスよく価値／原則／プラクティスというフォーマットで提供する開発手法である。

3.1.4. 契約

事例を契約形態によって下表のように分類した。

自社開発	自社組織内に開発部隊を抱える形態、及び事業者と開発者の間に契約関係がない形態
自社開発（一部オフショア）	自社組織内に開発部隊を抱えているが、一部を海外のパートナー会社にオフショアしている 事業者とオフショア先のパートナー会社の間の契約は準委任契約である
受託開発（請負）	事業者と開発者の間の契約形態が請負契約
受託開発（準委任）	事業者と開発者の間の契約形態が準委任契約
実証事業	官公庁や地方自治体が資金を提供する事業

表 3-3 契約別分類

今回の調査では、開発に使用したプラクティスの事例を収集する目的に伴い、受託開発の事例では開発者側（受注者側）の企業から調査票の回答を得た結果、半数強が自社組織内に開発部隊を抱える自社開発（一部オフショアも含む）を採用していることが分かった。

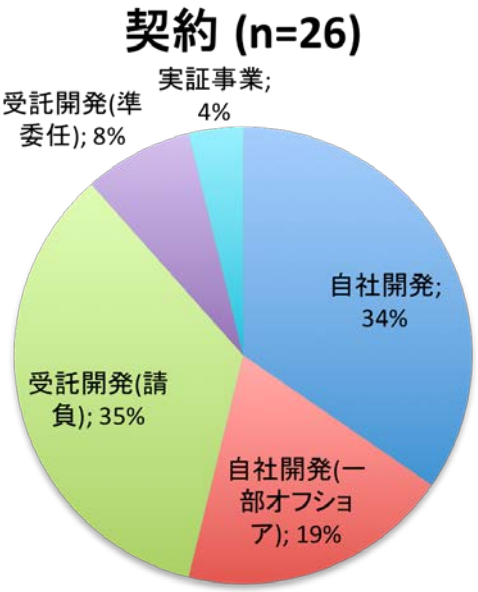

図 3-6 事例分類（契約別）

3.2. 調査対象の成功度

調査対象となるプロジェクトの成功度について、以下の 5 段階で回答を得た。

- ●5：うまくいった
- ●4：かなりの部分でうまくいった
- ●3：うまくいった面もあるがうまくいっていない面もある
- ●2：かなりの部分でうまくいかなかった
- ●1：うまくいかなかった

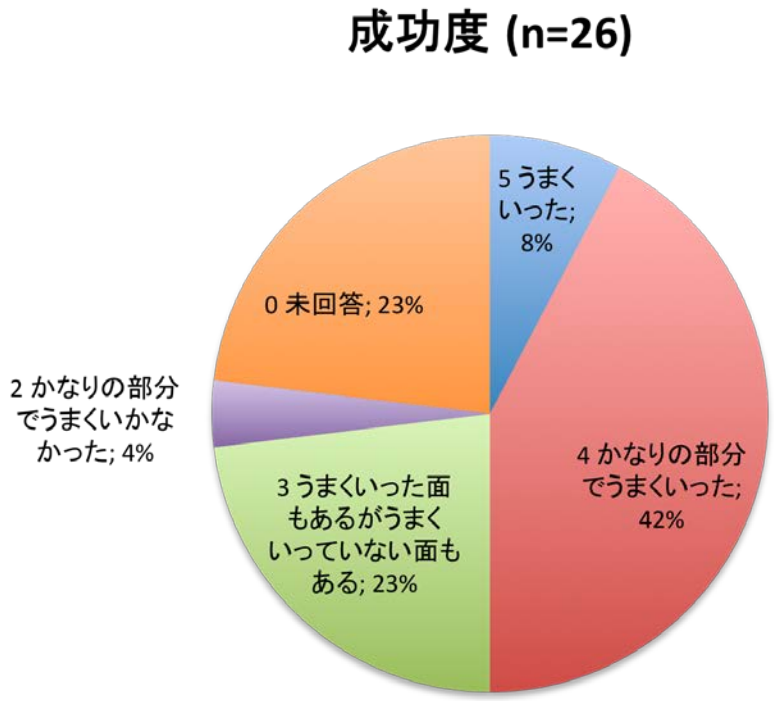

図 3-7 調査対象事例の成功度

「うまくいった」もしくは「かなりの部分でうまくいった」と回答した半数を占めるプロジェクトにおいては、主にプラクティスの有効な利用例を収集した。また、「うまくいった面もあるがうまくいっていない面もある」、「かなりの部分でうまくいかなかった」に該当する約 1/4 の回答からは、主にプラクティスを活用する際の留意点を収集した。

なお、成功の尺度としては以下の観点を挙げる企業が多かった。

- ●顧客にとってのビジネス価値・利便性の高いシステムを実現できたか？
- ●顧客からの要求の充足度が高かったか？顧客の望むシステムが実現できたか？
- ●費用対効果が高く、ムダのない開発ができたか？
- ●顧客満足度・開発者満足度が高かったか？
- ●顧客の参画度が高かったか？顧客からの信頼を獲得できたか？
- ●メンテナンスしやすく、変更容易性の高いシステムが構築できたか？
- ●品質・納期・コストは守れたか。生産性は向上したか？

4. 分析結果

本章では事例の調査結果を元に定量的な分析を行った。

なお、定性的な分析についてはガイド編の「3.5 プラクティス適用時のよくある問題と対応策」「5. 活用のポイント」を参照されたい。

4.1. プラクティス適用数

4.1.1. 全体

調査先企業より回答を得た「プラクティス調査アンケート」（付録 1：調査票）から、プラクティス毎に「◯ ： 適用」の事例を 1 件、「△ ： 部分的に適用」の事例を 0.5 件として適用数を集計した。

プラクティス毎の適用数は下図の通りとなった。

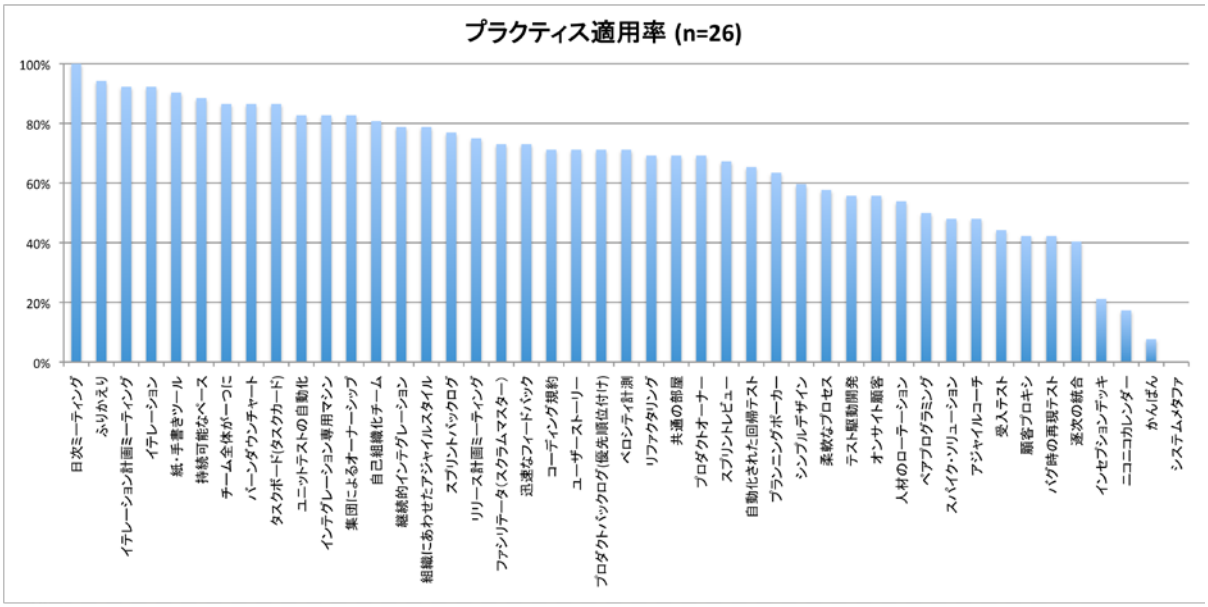

図 4-1 プラクティス適用数

ほとんどの事例（80%以上の事例）で使われていたプラクティス

- 日次ミーティング
- ふりかえり
- イテレーション計画ミーティング
- イテレーション
- 紙・手書きツール
- 持続可能なペース
- チーム全体が一つに
- バーンダウンチャート
- タスクボード（タスクカード）
- ユニットテストの自動化
- インテグレーション専用マシン
- 集団によるオーナーシップ
- 自己組織化チーム

多くの事例（50%以上の事例）で使われていたプラクティス

- 継続的インテグレーション
- 組織に合わせたアジャイルスタイル
- スプリントバックログ
- リリース計画ミーティング
- ファシリテータ（スクラムマスター）
- 迅速なフィードバック
- コーディング規約
- ユーザーストーリー
- プロダクトバックログ（優先順位付け）
- ベロシティ計測
- リファクタリング
- 共通の部屋
- プロダクトオーナー
- スプリントレビュー
- 自動化された回帰テスト
- プランニングポーカー
- シンプルデザイン
- 柔軟なプロセス
- テスト駆動開発
- オンサイト顧客
- 人材のローテーション
- ペアプログラミング

あまり使われていないプラクティス（適用事例が 50%未満）

- スパイク・ソリューション
- アジャイルコーチ
- 受入テスト
- 顧客プロキシ
- バグ時の再現テスト
- 逐次の統合
- インセプションデッキ

めったに使われていないプラクティス（適用事例が 20%未満）

- ニコニコカレンダー
- かんばん
- システムメタファ

システムメタファは国内の 26 事例の中で活用されている事例はなかった。『ガイド編 プラクティス解説』では、海外の事例を調査した。

プラクティスを 2.4.1 対象プラクティスで定義した 3 つのカテゴリに分類し、カテゴリ毎にプラクティス適用数の平均値を算出した。

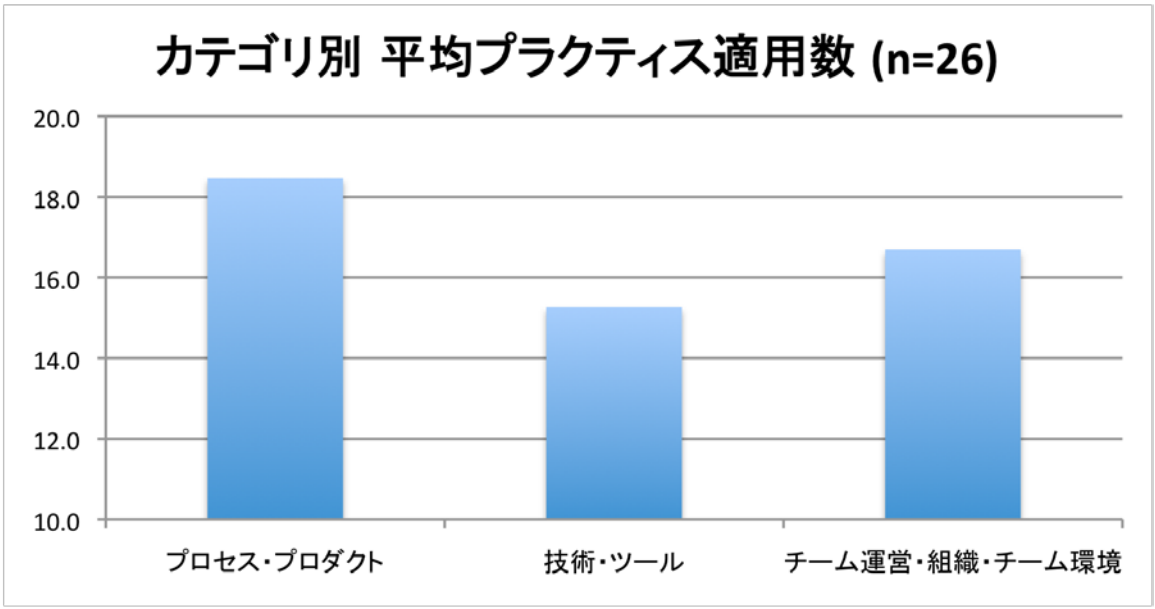

図 4-2 カテゴリ別平均プラクティス数

「プロセス・プロダクト」カテゴリのプラクティスの適用数が多く、「技術・ツール」カテゴリのプラクティスの適用数が少ないという傾向が窺えた。
「プロセス・プロダクト」カテゴリの中でも、日次ミーティングやふりかえりのような日々の改善系のプラクティスが上位に入っており、テスト駆動開発やリファクタリングのような「技術・ツール」カテゴリのプラクティスは多くの事例で使われているものの、ほとんどの事例で採用されるまでには至っていない。

4.1.2. 過年度調査との比較

ここ数年間でのプラクティス適用状況の変化を分析するため、2009 年度の IPA「非ウォーターフォール型開発に関する調査」のプラクティス適用数の調査結果との比較を行う。

注）本調査と 2009 年度の調査では、調査対象とした事例の特性（システム種別、規模、手法、契約）が異なるため単純に比較することはできないが、2009 年度からの変化の傾向を示すために本節を設けた。

本調査と 2009 年度の調査では、名称に差異があるものがあるため、以下のようにマッピングし、2012 年度調査欄（下表）の名称を使って説明することとした。

2009 年度調査	2012 年度調査
反復型計画	タイムボックス、頻繁なリリース、スクラムのスプリントと統合し、イテレーションとした ※2009 年度調査での「反復型計画」の適用数を、本調査の「イテレーション」の適用数と比較した
チーム全体が一つに	チーム全体が一つに
頻繁なふりかえり	ふりかえり
計画ゲーム	イテレーション計画ミーティング
日次のスタンドアップミーティング（朝会）	日次ミーティング
継続的インテグレーション	継続的インテグレーション
ペアプログラミング	ペアプログラミング
バーンダウンチャート	バーンダウンチャート
リファクタリング	リファクタリング
テスト駆動開発	テスト駆動開発
コードの共同所有	集団によるオーナーシップ

表 4-1 過年度とのプラクティスの対応(1)

2009 年度調査	2012 年度調査
かんばん	タスクカードと統合し、タスクボード（タスクカード）とした ※2009 年度調査での「かんばん」の適用数を、本調査の「タスクボード（タスクカード）」の適用数と比較した ※本調査ではかんばんとタスクボードを区別しており、2009 年度調査での「かんばん」は、本調査におけるタスクボードに相当する
自動化された回帰テスト	自動化された回帰テスト
ニコニコカレンダー	ニコニコカレンダー
顧客プロキシ	顧客プロキシ
タスクカード	かんばんと統合し、タスクボード（タスクカード）とした ※2009 年度調査での「かんばん」の適用数を、本調査の「タスクボード（タスクカード）」の適用数と比較した
ポストイット	紙・手書きツール
タイムボックス	反復型計画、頻繁なリリース、スクラムのスプリントと統合し、イテレーションとした ※2009 年度調査での「反復型計画」の適用数を、本調査の「イテレーション」の適用数と比較した
頻繁なリリース	反復型計画、タイムボックス、スクラムのスプリントと統合し、イテレーションとした ※2009 年度調査での「反復型計画」の適用数を、本調査の「イテレーション」の適用数と比較した
コーディング規約	コーディング規約
ストーリーカード	該当なし
単体テストの自動化	ユニットテストの自動化
スクラムのスプリント	反復型計画、タイムボックス、頻繁なリリースと統合し、イテレーションとした ※2009 年度調査での「反復型計画」の適用数を、本調査の「イテレーション」の適用数と比較した
スプリントバックログ	スプリントバックログ

表 4-2 過年度とのプラクティスの対応(2)

上記のプラクティスの適用率は下図のようになった。

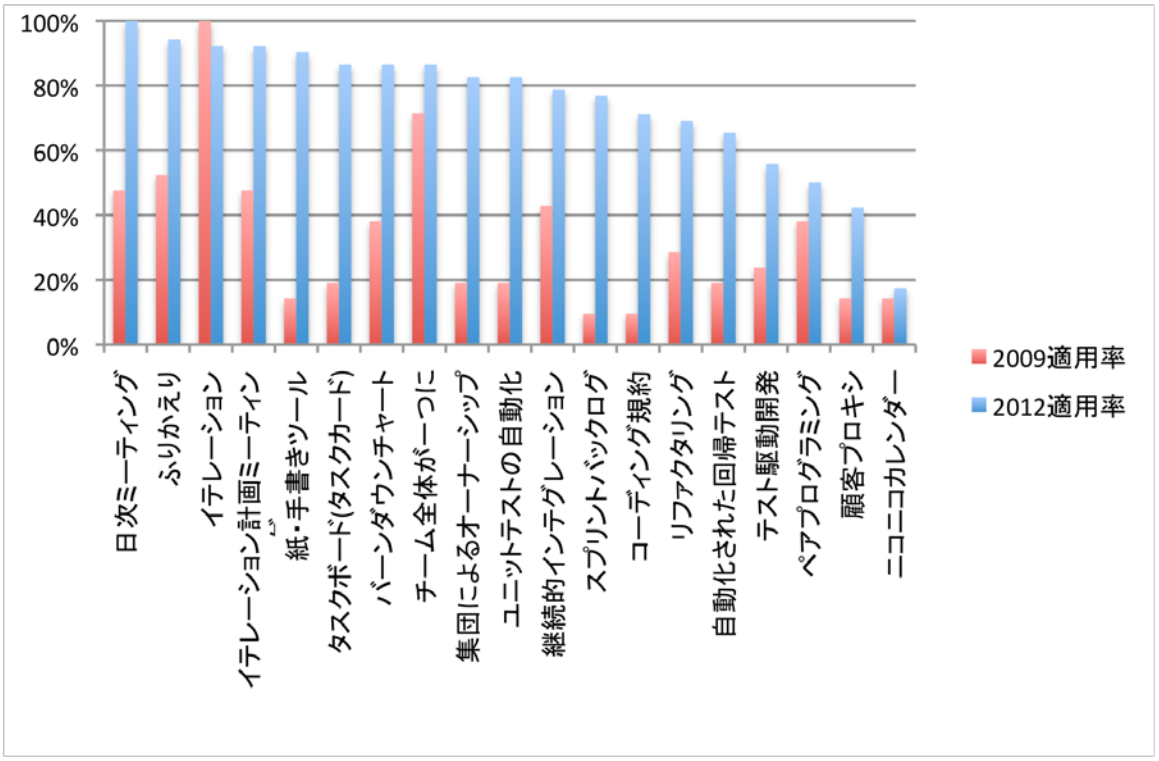

図 4-3 過年度とのプラクティス適用率の比較

下記のプラクティスでは、本調査と 2009 年度の調査結果と比較して適用率に大差が見られた。

プラクティス	2009 年度適用率	2012 年度適用率	倍率
スプリントバックログ	9.5%	76.9%	808%
コーディング規約	9.5%	71.2%	747%
紙・手書きツール	14.3%	90.4%	633%
タスクボード（タスクカード）	19.0%	86.5%	454%
ユニットテストの自動化	19.0%	82.7%	434%

表 4-3 過年度とのプラクティス適用率の比較（倍率上位 5 プラクティス）

スプリントバックログは Scrum のプラクティスであるが、2009 年度から適用率が 8 倍以上上昇している。これは、2009 年当時は Scrum 自体を採用している事例が少なかったが、その後、認定スクラムマスター研修が国内で開催されるようになったことにより、Scrum が普及したためであると思われる。
紙・手書きツールやタスクボード・タスクカードについても適用率が上昇していることについては、アジャイル型開発の認知度が高まり、社内にこれらのアナログツールを設置する障壁が減ったことや、事業者と開発者間の密なコミュニケーションの必要性が認識されたことにより、事業者と開発者が同一拠点で開発に取り組む割合が多くなり（調査対象 26 事例中

18 事例が事業者の代表であるプロダクトオーナーと開発チームが同一拠点で開発）、アナログツールが活用される場面が増えたためであると思われる。
ユニットテストの自動化につても、2009 年度から適用率が 4 倍以上、上昇している。これは、アジャイル型開発が一般的になるにつれ、プロセスの側面だけでなく、技術プラクティスを適切に実践することの重要性が認識されたためであると思われる。

反対に、本調査と 2009 年度の適用率にあまり差が出なかったのは下記のプラクティスである。

プラクティス	2009 年度適用率	2012 年度適用率	倍率
ふりかえり	52.4%	94.2%	180%
ペアプログラミング	38.1%	50.0%	131%
チーム全体が一つに	71.4%	86.5%	121%
ニコニコカレンダー	14.3%	17.3%	121%
イテレーション	100.0%	92.3%	92%

表 4-4 過年度とのプラクティス適用率の比較（倍率下位 5 プラクティス）

ふりかえりやペアプログラミングは、アジャイル型開発が国内で紹介されはじめた当時から現場で適用されてきたプラクティスである。
イテレーションは唯一 2009 年度の調査から適用率が減少したプラクティスであるが、2009 年度当時はすべての事例で適用されていた。今回の調査では適用していない事例が 2 事例あったことに対しては、明確なイテレーション（タイムボックス）を設けずにタスクの総量を決め、タスクが一つ完了したら新しいタスクを追加するやり方（カンバン方式）を採用していたことが理由としてある。

4.2. コンテキストと使用プラクティスの関係

どのコンテキストで、どのようなプラクティスが使われているかの分析は非常に重要である。本節ではコンテキスト情報に加えて、フォース（その場において働いている力、また制約）に着目して、コンテキスト+フォースによって、どのプラクティスが利用されているかの関係を示す。

4.2.1. システム種類別

B2C サービスと社内システムの間で、適用されているプラクティスの違いを比較した。

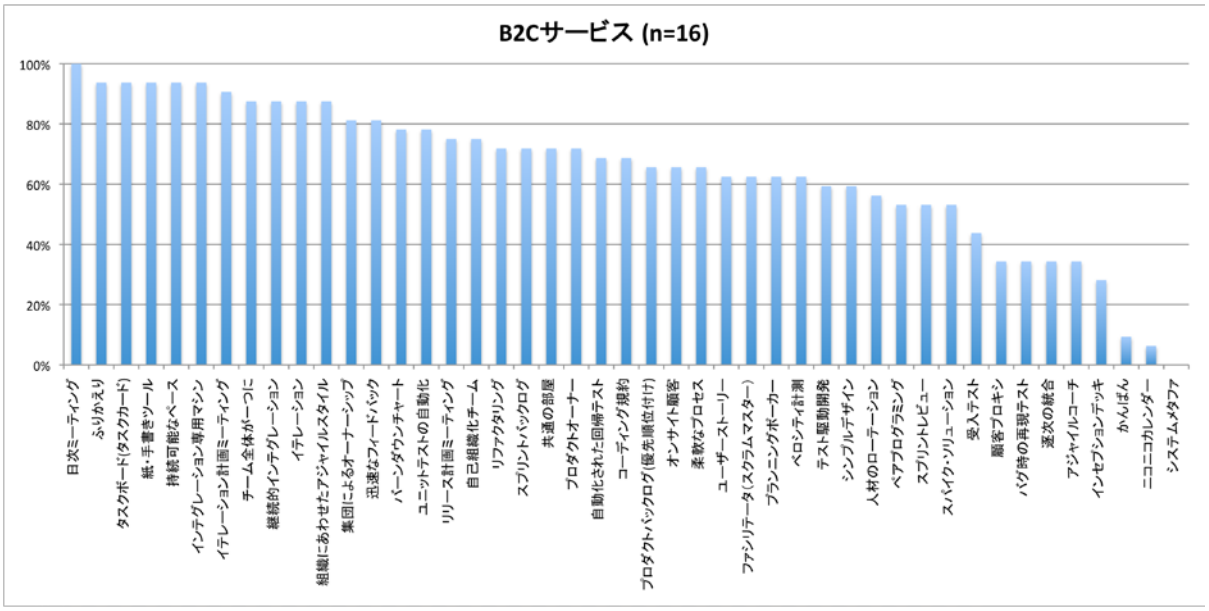

図 4-4 プラクティス適用率（B2C サービス）

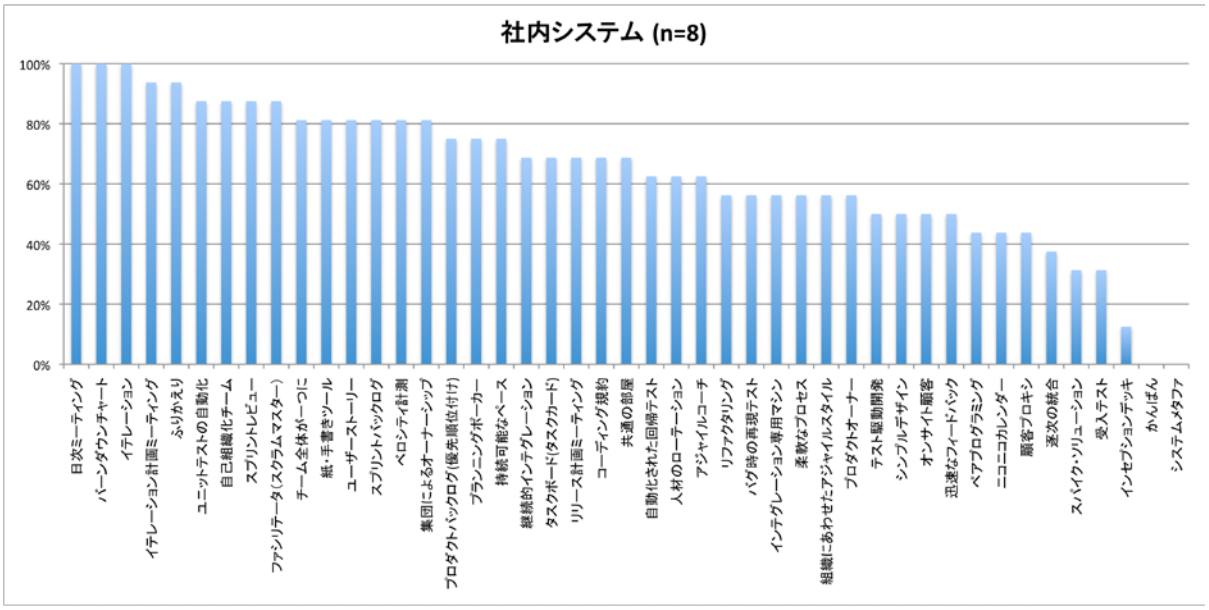

図 4-5 プラクティス適用率（社内システム）

特に、適用率の差が大きいのは以下のプラクティスである。

プラクティス	B2C サービス	社内システム	適用率差
ニコニコカレンダー	6.3%	43.8%	37.5%
インテグレーション専用マシン	93.8%	56.3%	37.5%
スプリントレビュー	53.1%	87.5%	34.4%
組織に合わせたアジャイルスタイル	87.5%	56.3%	31.3%
迅速なフィードバック	81.3%	50.0%	31.3%

表 4-5 システム種類別プラクティス適用率の比較（適用率差上位 5 プラクティス）

B2C サービスでは社内システムに比べて、インテグレーション専用マシンや迅速なフィードバックの適用率が高い。これは、B2C サービスのほうが社内システムに比べてリリース後に変更が入る率が高く、迅速なフィードバックが求められるためであると思われる。
また B2C サービスでは、社内システムに比べてスプリントレビューの適用率が低い。これは、B2C サービスではプロダクトオーナーによるレビューが行われないというわけではなく、より迅速なフィードバックが求められるため、スプリントより短い周期でプロダクトオーナーによるレビューを行っていることが多いためである。

4.2.2. 規模別

小規模開発と中大規模開発との間で、適用されているプラクティスの違いを比較した。

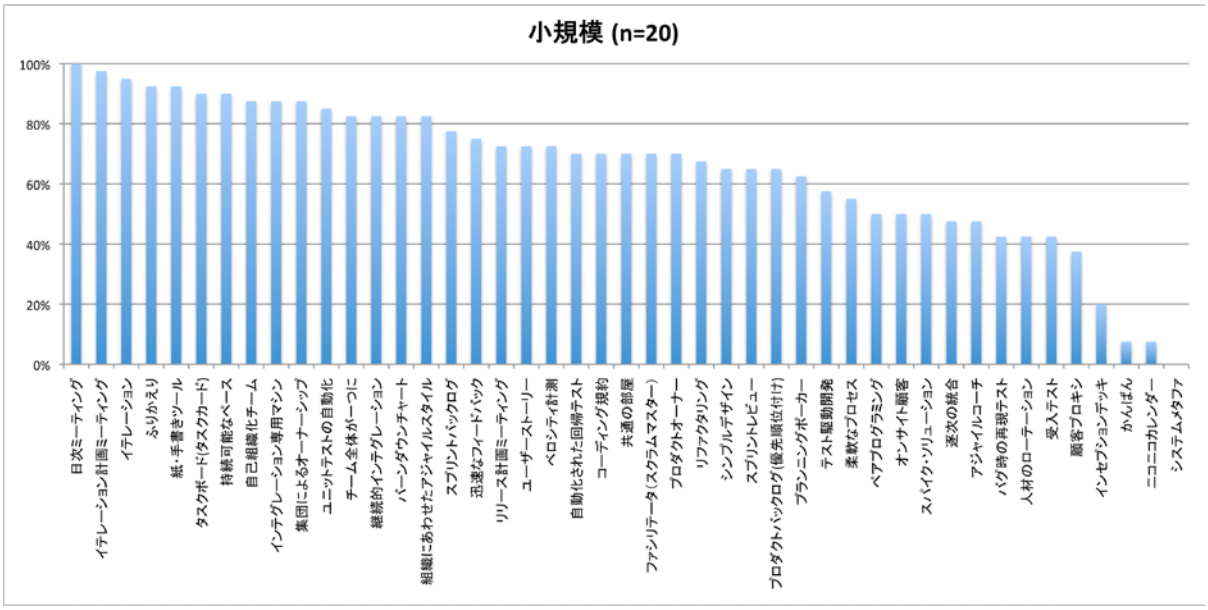

図 4-6 プラクティス適用率（小規模）

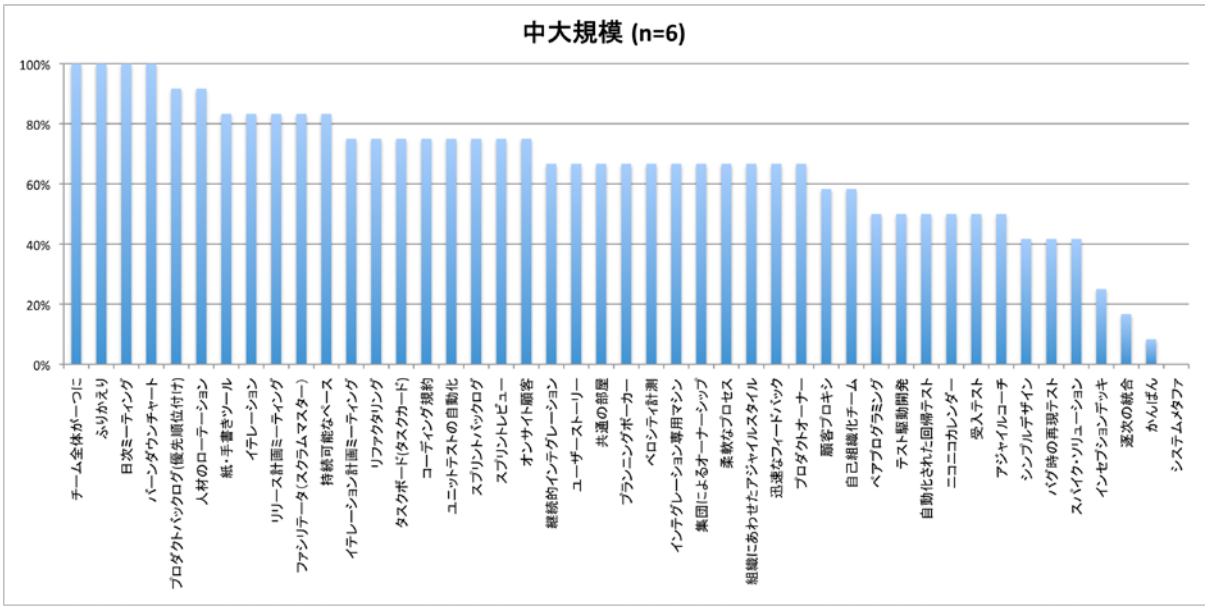

図 4-7 プラクティス適用率（中大規模）

小規模ではイテレーション計画ミーティングがリリース計画ミーティングよりも適用率が高いのに対して、中大規模ではこれが逆転している。このことから、中大規模では中長期の計画がより重視されていることが分かる。また、大規模ではオンサイト顧客が上位になっており、顧客と開発者のコミュニケーションをいかに円滑に行うかがポイントであることが分かる。

特に、適用率の差が大きいのは以下のプラクティスである。

プラクティス	小規模	中大規模	適用率差
人材のローテーション	42.5%	91.7%	49.2%
ニコニコカレンダー	7.5%	50.0%	42.5%
逐次の統合	47.5%	16.7%	30.8%
自己組織化チーム	87.5%	58.3%	29.2%
プロダクトバックログ（優先順位付け）	65.0%	91.7%	26.7%

表 4-6 規模別プラクティス適用率の比較（適用率差上位 5 プラクティス）

中大規模では人材のローテーションが適用される率が高い。これは、中大規模プロジェクトにおいて、人材を流動的にして、ナレッジやノウハウをいかに流通させるかが課題であるためであると思われる。
一方、中大規模では逐次の統合の適用率が低い。これは、中大規模プロジェクトではビッグバン統合になりがちで、逐次の統合をやりたくてもできないという事情があるためであると思われる。したがって、いかに大きな規模のシステムを大きく見せないか、分割して取り扱えるかがポイントであると言える。
自己組織化チームについても、中大規模での適用率が低くなっている。中大規模プロジェクトでは関わる人数が多いため、自己組織化よりも明文化された規範が重視される傾向にあるためであると思われる。

4.2.3. 手法別

Scrum と XP との間で、適用されているプラクティスの違いを比較した。

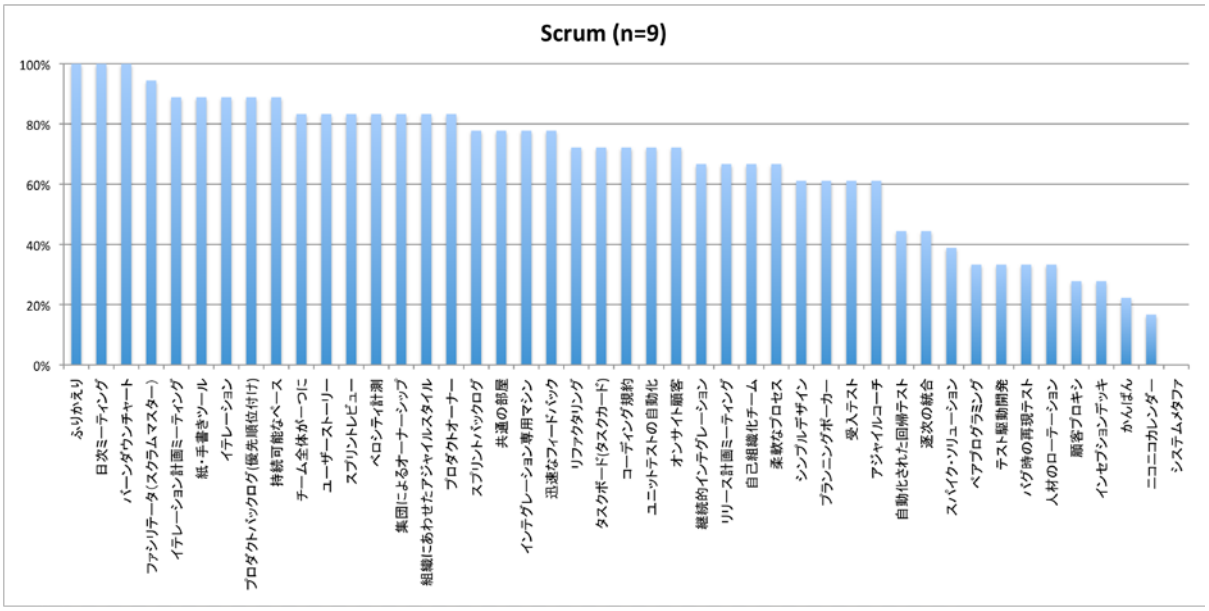

図 4-8 プラクティス適用率（Scrum）

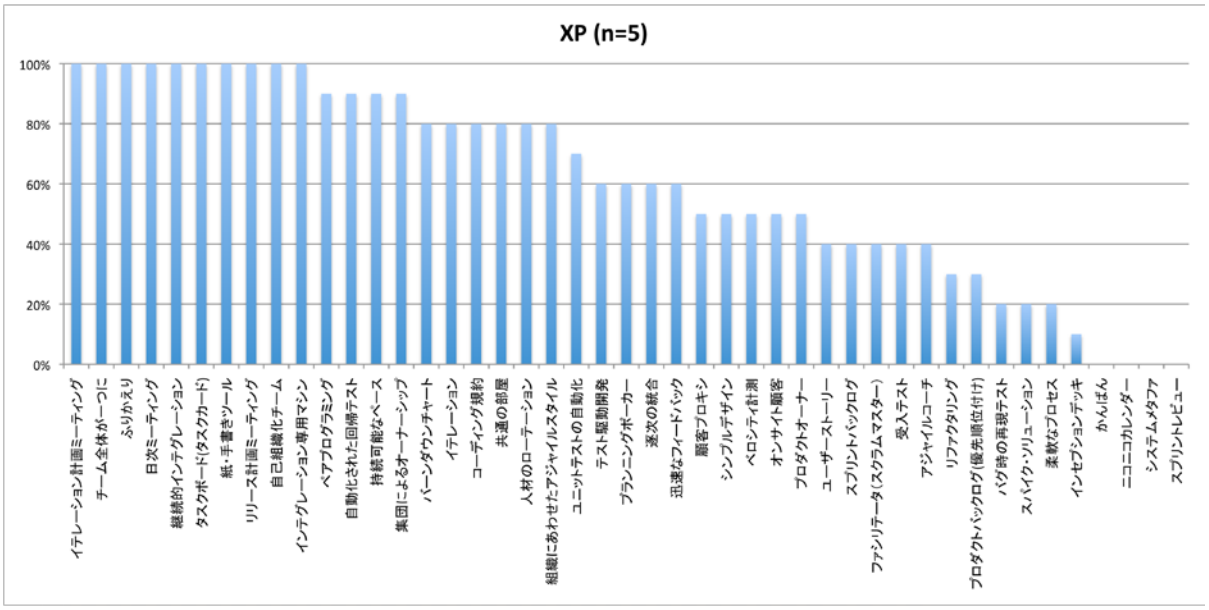

図 4-9 プラクティス適用率（XP）

特に、適用率の差が大きいのは以下のプラクティスである。

プラクティス	Scrum	XP	適用率差
スプリントレビュー	83.3%	0.0%	83.3%
プロダクトバックログ（優先順位付け）	88.9%	30.0%	58.9%
ペアプログラミング	33.3%	90.0%	56.7%
ファシリテータ（スクラムマスター）	94.4%	40.0%	54.4%
柔軟なプロセス	66.7%	20.0%	46.7%

表 4-7 手法別プラクティス適用率の比較（適用率差上位 5 プラクティス）

Scrum ではスプリントレビューやファシリテータ（スクラムマスター）、プロダクトバックログ（優先順位付け）といった Scrum 発祥のプラクティスの適用率が高くなっている。一方、XP 発祥のプラクティスであるペアプログラミングは XP プロジェクトでの適用率が高くなっている。
また、柔軟なプロセスの適用率が Scrum で高く XP で低くなっているのは、XP のほうがより規律的であるためであると言える
リファクタリングは XP 発祥のプラクティスにも関わらず、Scrum プロジェクトのほうが、適用率が高いという結果が出ている（Scrum：72%、XP：30%）。これは、「技術・ツール」系のプラクティスが定義されていない Scrum 手法を使うプロジェクトが XP のプラクティスの利点を取り込んでいるためであると考えられる。

4.2.4. 契約別

自社開発と受託開発との間で適用されているプラクティスの違いを比較した。

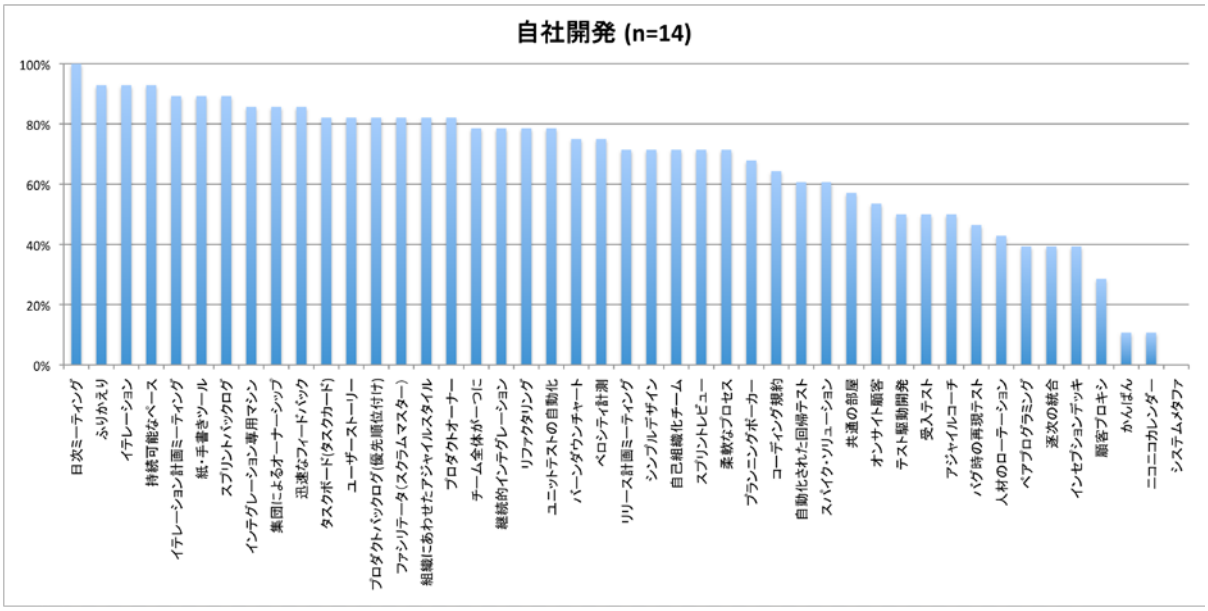

図 4-10 プラクティス適用率（自社開発）

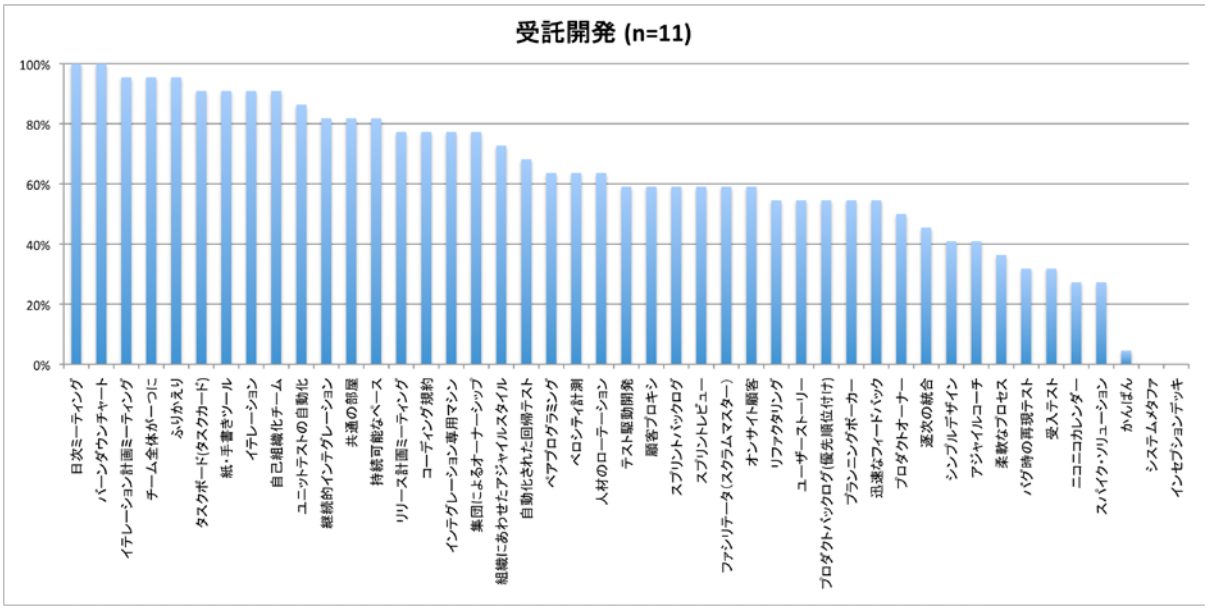

図 4-11 プラクティス適用率（受託開発）

特に、適用率の差が大きいのは以下のプラクティスである。

プラクティス	自社開発	受託開発	適用率差
インセプションデッキ	39.3%	0.0%	39.3%
柔軟なプロセス	71.4%	36.4%	35.1%
スパイク・ソリューション	60.7%	27.3%	33.4%
プロダクトオーナー	82.1%	50.0%	32.1%
迅速なフィードバック	85.7%	54.6%	31.2%

表 4-8 契約別プラクティス適用率の比較（適用率差上位 5 プラクティス）

自社開発と受託開発では、他の特性（システム種別、規模、手法）ほど適用率の差がなかった。 このことから、アジャイル型開発を適用するという前提に立った上では、契約形態の違いは、適用されるプラクティスにはあまり影響を及ぼさないと言うことが窺える。
受託開発では 10 の質問によりプロジェクトのビジョンやゴールを明らかにするインセプションデッキを使っている事例が皆無であったことに関して、受託開発では、インセプションデッキのような手法は開発者から事業者への 押し売りになりやすく、理解のある事業者の下でしか実践できないという傾向があるものと思われる。

5. まとめ

今回は、アジャイル型開発の事例を広く調査し、開発手法の実践的な解説書と実例やテクニック等の工夫を Tips 集として取りまとめた。特に、日本国内での実践事例を調査することにより、これからアジャイル型開発を導入しようというプロジェクトに向けて、独自の工夫ポイントを示した。

プラクティス状況アンケートとプラクティス調査アンケートを回収し、各事例の結果を集計してプラクティス適用数の傾向を分析した。また、調査先の各企業にヒアリングを行い、各事例から収集した独自の工夫や、発見された個別のプラクティスを抽出した。

調査対象とする事例は、プロジェクト特性（システム種別、規模、手法、契約）の観点から見たときに、できるだけ偏りが出ないように調整した。

5.1. 総括

日次ミーティング、ふりかえり、イテレーション計画ミーティング、イテレーションの順に適用率が高く、これらはアジャイル型開発を行う上でのほぼ必須のプラクティスであると言える。これらのプラクティスは Scrum と XP に共通するプラクティスである。

2009 年度の調査結果と比較すると、スプリントバックログ・紙・手書きツール・タスクボード（タスクカード）・ユニットテストの自動化適用率が 4 倍以上上昇している。これに関しては以下に示すように、ここ数年の日本国内でのアジャイルの動向を如実に表している。

- 認定スクラムマスター研修が国内で開催されるようになったことにより、Scrum の認知度が向上した。
- 事業者と開発者での密なコミュニケーションの必要性が認識されたことにより、事業者と開発者が同一拠点で開発に取り組む割合が多く（調査対象 26 事例中 18 事例が事業者の代表であるプロダクトオーナーと開発チームが同一拠点で開発）、紙・手書きツールやタスクボードなどのアナログのツールが活用される場面が増えた。
- プロセスの側面だけでなく、「技術・ツール」カテゴリのプラクティスを適切に実践することの重要性が認識された。

2011 年度の IPA 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」では、現状、日本のアジャイル型開発は「普及が遅れており、ようやく認知されはじめた」段階とされているが、2009 年度の調査と比較して、企業におけるプラクティスの適用率は総じて上昇の傾向にあった。また、今回調査対象とした企業においては、プラクティスを取り入れ、実践するまでに至っていた。
しかし、アジャイル型開発の認知度が高まるのに伴い、一方ではアジャイル型開発を導入しさえすれば「生産性、品質、コスト」に関する問題が解決するという誤った認識も生まれ始めていることから、プラクティスへの正しい理解を広めると共に、「アジャイルコーチ」

（外部からの変化）と「組織に合わせたアジャイルスタイル」（内部からの変化）をバランス良く取り入れながら、「正しい普及」活動を進めてゆくことが今後の課題となる。

プロジェクト特性（システム種別、規模、手法、契約）の中でも特に規模の違いが活用するプラクティスに及ぼす影響は大きい。反面、契約やシステム種別の違いは他のプロジェクト特性に比べると、あまりプラクティスに影響を及ぼさないことが分かった。
規模別（小規模と中大規模）で適用率に最も大きな差があったプラクティス（人材のローテーション）はその差が 49.5%、契約別（自社開発と受託開発）で適用率に最も大きな差があったプラクティス（インセプションデッキ）では 39.3%、システム種別（B2C サービスと社内システム）で適用率に最も大きな差があったプラクティス（ニコニコカレンダーとインテグレーション専用マシン）では 37.5%であった。

実際、契約の違いはあまりプラクティスに影響を与えていないが、海外と国内の環境の違い（1.4 国内のアジャイル型開発の課題）を踏まえると、日本の環境では特に以下のプラクティスを活用する必要がある。

- ファシリテータ（スクラムマスター）：契約絡みの問題のために、アジャイル型開発が進めにくくならないように、顧客や開発者とは別の組織に属する第三者を立てる。
- 顧客プロキシ：顧客と直接対話するのが難しい場合は顧客プロキシを立てる。
- 共通の部屋：物理的に近くで仕事をする、できない場合は電話会議ツールなどを活用する。

調査対象の事例それぞれの状況（コンテキスト）に合わせた工夫があることが分かった。実際に、本調査で独自の工夫を収集したところ、26 事例で 180 以上の工夫が見つかった（リファレンス・ガイド 付録を参照）。 現場での独自の工夫ができる余地があるところがアジャイル型開発の利点であり、本調査報告書のプラクティスを適用する状況・問題を参考にし、「なぜそのプラクティスを適用するのか」を考えて、それぞれの現場にあわせてカスタマイズして活用すべきである。

5.2. 提言

日本でアジャイル開発をより広く普及させる上での施策を示す。

1. ファシリテータ・顧客プロキシができる人材の育成
   「顧客と開発チームの間に契約をはさむ受託開発が多い」「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」といった環境では、ファシリテータ（スクラムマスター）や顧客プロキシといった、顧客と開発チームの間に立って両者の調整をする役割が非常に重要になる。

2. プラクティスを現場から発信できる場づくり
   今回は調査としてプラクティスを収集したが、時間的制約のため、現場の状況情報などの吸い上げが充分ではなかった。日本独自の工夫を収集するには、トップダウンによる調査だけでなく、もっと日常的に現場の開発者が自分達の工夫を言葉にして、それを社内・社外関わらず共有しながら、現場の知を交流させて、新しい知を生成していく必要がある。そうすることによって、海外からの輸入されたプラクティスではなく、日本、あるいは個々の業界独自のプラクティスが生まれて流通することで、国内産業の競争力の向上に寄与できると考える。

参考文献

[1] Sletholt, M.T., Hannay, J.E., Pfahl, D., Langtangen, H.P. (2012). What Do We Know about Scientific Software Development's Agile Practices?: Computing in Science & Engineering

[2] Standish Group (2011). CHAOS Manifesto

[3] IPA (2009). 非ウォーターフォール型開発に関する調査

付録 1:調査票

今回の調査で調査先各社に回答を求めた調査票の雛形を添付する。
プラクティスが実践された事例のコンテキストを調査する調査票と、プラクティスそのものの実施状況を調査する調査票を用意した。

- プラクティス状況アンケート
- プラクティス調査アンケート

■ プラクティス状況アンケート

プラクティスが実践された事例のコンテキストを調査するために、プロジェクトの背景等を中心に以降の質問を行った。

【プラクティス状況アンケート 項目】

項目			注釈
プロジェクトの概要	企業プロフィール	企業名	
		組織名	
	開発対象のアプリケーションシステムの種類		開発対象のソフトウェア／サービスの種類・用途
	対象ドメイン	アプリケーションシステムの種別	顧客問題解決ソフトウェア（顧客の要望を受けて開発するソフトウェア）であるか、ソフトウェアプロダクト（ソフトウェアフレームワーク、プラットフォーム、パッケージソフトウェア、組込みソフトウェア等）であるか
		優先した IT 戦略やアーキテクチャ	システムアーキテクチャの設計、構築、変更について実施した工夫、参考にした手法などを回答ください。
	使用した業務パッケージ		アプリケーションを構築する上で利用している業務パッケージについて
プロジェクトの推進体制	プロジェクト概要	開発規模	人月、ユースケースの数、ユーザーストーリー数、ステップ数等の指標を用いて記述
		チーム人数	チーム数とチームあたりの平均人数をご回答ください。正確な数字がある場合は、それぞれのチームの人数をご記述ください。
		開発期間	プロジェクトの開始から終了までの期間
		利用業務	
		開発言語	Java、C、Ruby、等
	開発プロジェクト内の責任分担		事業責任、ガバナンス、組織の歴史・環境・文化という視点から、分担関係を記述 （システムオーナとは、開発対象システムの企画・投資に関して責任を持つ職責者を指す）
	プロジェクトの進め方	ライフサイクルモデル	各プロセスにかけた期間や人月、イテレーションを行ったプロセス・回数・期間、等を記述
		プロセス間の連携	プロセス間で連携するために実施したこと、利用したツール、等を記述
		ドキュメンテーション	どのような情報を文書化したのか、逆にしなかったのか、その範囲と、工夫した点についてご回答ください。
		プロジェクトマネージャの調整方法	たとえば、ステークホルダーとの関係性やスケジュールの調整方法などプロジェクトマネージャとしての調整方法をご記入ください。
	開発チー	メンバーのスキル	「別表 1」ワークシート記載の表におけるレベ

	ム編成		ルをそれぞれ記述する
		研修・訓練	研修や訓練方法についてご記入ください。
		ロケーション	拠点の場所と、参加メンバーについてご記述ください。同一拠点で同フロアに集っていた、同一拠点だがフロアが異なっていた、全く別の拠点に分散していた、など。(場所もお願いします)
		コミュニケーション手段・頻度	会議や電話会議システムなどについて
		顧客の関与度合い	顧客とどの程度、開発に関与するか度合いを記入
		コーチ／コンサルタントの有無	社内や社外のコーチやコンサルタントの活用
	契約形態		プロジェクトに参画した企業と、その間の契約形態についてご回答ください。(顧客との間、開発企業間、コンサルティング会社、など)
	開発時の課題と検討・解決策		アーキテクチャやデータモデルの変更のタイミングや対応策等、開発時の課題と、それを検討した内容と、実施した解決策をご記述ください。
開発方法論やツール	使用した開発手法		Scrum や XP、CMMI など使用した開発手法についてご記入ください。
	使用したツール群		Eclipse や Jenkins など使用したツール群についてご記入ください。
	スコープ管理(機能分割の方法等)		プロジェクトを成功のうちに完了するために、プロジェクトに必要とされるすべての作業を含み、かつ、必要とされる作業のみを含んでいることを確実にするために必要なプロセス。 スコープ計画、スコープ定義、WBS 作成、スコープ検証、スコープ・コントロールのプロセスからなる。
手法や問題回避の工夫	要求形成／追跡		オブジェクト指向分析やトレーサービリティ技術の利用等、要求形成／追跡に関して行ったこと、適用した手法、等を記述
	設計		コンポーネント利用設計やデザインパターンの利用等、設計に関して行ったこと、適用した手法、等を記述
	構築		テスト駆動開発や自動コード生成ツールの利用等、実装に関して行ったこと、適用した手法、等を記述
	ドキュメンテーション(作成文書の種類、作成の責任、作成のタイミング等)		同左
	テスティング／V&V(Verification and Validation)		リグレッションテストツールの利用やブラックボックステストの実行等、テスティング／V&V に関して行ったこと、適用した手法、等を記述

	メンテナンス		開発したアプリケーションシステムに対して行ったこと、適用した手法、体制、ドキュメント等を記述 特に、開発を委託／受注している場合は、瑕疵担保責任期間内とその後についてどのように対応したかを記述。
	進捗管理(特に工事進行基準への対応)		開発時の進捗管理手法
	品質管理		プロジェクトが意図するニーズを満足させることを確実にするために必要なプロセス。 品質計画、品質保証、品質管理のプロセスからなる。
	生産性向上		生産性の管理と向上についての取り組みがあれば、ご記入ください。
	システム監査		該当プロジェクトで必要なシステム監査の方法や頻度についてご記入ください。 特に、システム監査で必要となる証跡（設計文書やテストの記録）への対応など
プロジェクトの企画・結果	アジャイル型開発の採用	経緯	アジャイル型開発を採用することになった経緯や問題をご記入ください。
		判断ポイント	アジャイル型開発を採用することになった判断ポイント（享受できるメリットなど）をご記入ください。
	成功・失敗	成功度合い	プロジェクトの成功度合いを表現すると、5段階でいくつになるでしょうか？その他、経緯や考慮点などを踏まえて、5段階でお答えください。
		成功・失敗の理由	上記の「成功度」の尺度や結果についての理由についご記入ください。
	稼働後の品質状況		稼働後6ヶ月のバグ密度など、具体的な数値があれば記入ください。
他の開発手法と組み合わせ	特徴的な考え方		複数手法を組み合わせた場合、ケース固有の特徴的な考え方についてご記入ください。
	開発技法		複数手法を組み合わせた設計・管理方法、プロジェクト運営のテクニック等の開発技法についてご記入ください。

■ プラクティス調査アンケート

プラクティスを実践していた事例の代表者に対して、プラクティスそのものの実施状況を調査するための質問を行った。

なお、カテゴリの分類やプラクティス名については、調査後に名前や分類の変更及びプラクティスの集約を行ったため、表 2-2 プラクティス一覧 のカテゴリ・プラクティスとは必ずしも一致していない。

① 適用度

プラクティスを適用したかどうか。

◯適用

△部分的に／カスタマイズして適用

×未適用

？検討中

プラクティス適用数の分析に使用した。

② 評価

プラクティスに対する評価。

◯（効果があった等）評価が高い

△評価が高いとも低いとも言えない

×（デメリットが多い等）評価が低い

(空欄)未評価

評価が△や×のプラクティスについては、留意点が導き出せる可能性があるため、重点的にヒアリングを行った。

③ 独自の工夫や導入ポイント

【プラクティス調査アンケート 項目】

カテゴリ		プラクティス	説明
プロセス	リリース計画作り	リリース計画ミーティング	プロダクトリリースのためのリリース計画ミーティング
	反復計画づくり	反復型プロセス	ゴールや結果にアプローチするプロセスをくり返すこと。
		反復毎の計画ミーティング(計画ゲーム)	イテレーション（スプリント）ごとのリリース計画やアクティビティなどを計画するミーティング。
		反復毎の計画ミーティング(スプリント計画)	スプリントバックログを作るためのスプリント計画
	イテレーション	タイムボックス	提供可能なアウトプットを作るためのタイムボックスを用いている
		スクラムのスプリント	上記、タイムボックスと同義
	見積り手法	プランニングポーカー	スプリント計画時のタスクを見積もるためのプランニングポーカー
		ベロシティ計測	プロジェクトベロシティの計測

	デイリーミーティング	日次ミーティング（朝会／デイリースクラム）	現在の問題を解決するための短いデイリーミーティング
	出荷	ステッピングストーン（踏み石）	フィーチャを提供可能な状態で小さく分けること
	点検	ふりかえり	前のスプリント（イテレーション）から学ぶためにふりかえる
	進行管理	バーンダウンチャート	スプリント進捗をモニターするためのバーンダウンチャート
	ファシリティ	かんばん	ジャストインタイムの継続的なデリバリを強調した管理手法
		タスク（カード）	ボードに貼られたメンバーが継続的に更新するタスク
		スプリントレビュー	完了した仕事を表明するスプリントレビューミーティング
	その他	柔軟なプロセス	状況や対応に対応できる柔軟なプロセスにしている、もしくは、プロセスを柔軟に変更している。
プロダクト	要件管理	ユーザーストーリー	要求についての会話を行うときの開発チームとプロダクトオーナーの間の合意事項
		スプリントバックログ	プロダクトオーナーとチーム間でのスプリントバックログへの相互コミットメント
		インセプションデッキ	10 の質問によりプロジェクトの属性を明らかする
	順序付け	優先順位（プロダクトバックログ）の管理	プロダクトオーナーによる優先順位（プロダクトバックログ）の管理
人	顧客役	顧客プロキシ	要件や仕様をまとめるために顧客の業務に精通した顧客プロキシの設置
		オンサイト顧客	顧客といつでも／定期的にやりとりが可能である
		プロダクトオーナー	プロダクトオーナー役の設置
	リーダーシップ	ファシリテータ（スクラムマスター）	スクラムマスターによる開発プロセスとプラクティスのファシリテート
	支援者	アジャイルコーチ	アジャイルコーチがプロジェクトに参加している
		外部コンサルタント	外部コンサルタントがプロジェクトに参加している
		自己組織化チーム	チームメンバーがタスクに志願するなど自律的なチームになっている
	状況把握	ニコニコカレンダー	ニコニコカレンダーを用いてメンバーの気持ちを見える化している
		チーム全体が一つ	チーム全員が一つのゴールに向かうよ

		に	うな取り組みを行っている
		共通の部屋	オープンスペースがチームに与えられている
		人材のローテーション	多能工の育成などのため人材のローテーションを行っている
	フィードバック	迅速なフィードバック	迅速なフィードバックを得られるような取り組みを行っている
	組織導入	組織に合わせたアジャイルスタイル	組織にあった適切なアジャイルスタイルを用いるようにしている
	進め方	継続的なペース	継続的なペースで開発している
設計開発	開発戦略	ペアプログラミング	すべての製品コードはペアプロで開発している
	テスト	自動化された回帰テスト	自動化された回帰テストを行っている
		テスト駆動開発	ユニットテストを書き、そのテストを通るようなコードを実装する
		ユニットテストの自動化	ユニットテストの自動化し、開発者がいつでも実施して結果が分かるようにしておく
		全コードのユニットテスト	全コードがユニットテストされている
		バグ時の再現テスト作成	バグが見つかったとき、その再現テストが最初に作られる
		リリース前の全コードユニットテスト	リリース前の全コードがユニットテストを通す
		頻繁な受け入れテスト実施とスコア公開	受け入れテストの実施と、その結果を公開している
	アーキテクチャ	システムのメタファ	関係者全員が、そのシステムがどのように動くかについて伝えることができるメタファ(比喩)を使ってアーキテクチャの共通認識を作る。
		スパイク・ソリューション	リスクを軽減するために、かくれた問題を探索するための簡単なプログラム（スパイク・ソリューション）の試作
	品質作り込み	リファクタリング	定常的なリファクタリング
	モデリング	CRC カードの使用	設計セッションのための CRC(class-responsibility-collaboration) カードの使用
	設計	シンプルデザイン	設計をシンプルに保つ
	結合戦略	継続的インテグレーション	継続的イテレーション、または頻繁なインテグレーション
		逐次の統合	一度に統合するコードは一つだけとする
	コードの共同所有	集団によるオーナーシップ	全員がすべてのコードに対して責任を持つ

	コーディング規約	コーディング規約	同意された標準のためのコーディング規約
	その他	フィーチャ・パイプライン	プロジェクトコストを見積もる代わりに機能のパイプラインを作ること
利用ツール		紙・手書きツール	付箋紙の使用
		インテグレーション専用マシン	特定のインテグレーション用コンピュータ
その他		上記以外のプラクティス	上記以外のプラクティスがございましたらご記入ください