IPA Better Life with IT

アジャイル開発の導入に向けて

～ IPA/SECにおける取組みの概要 ～

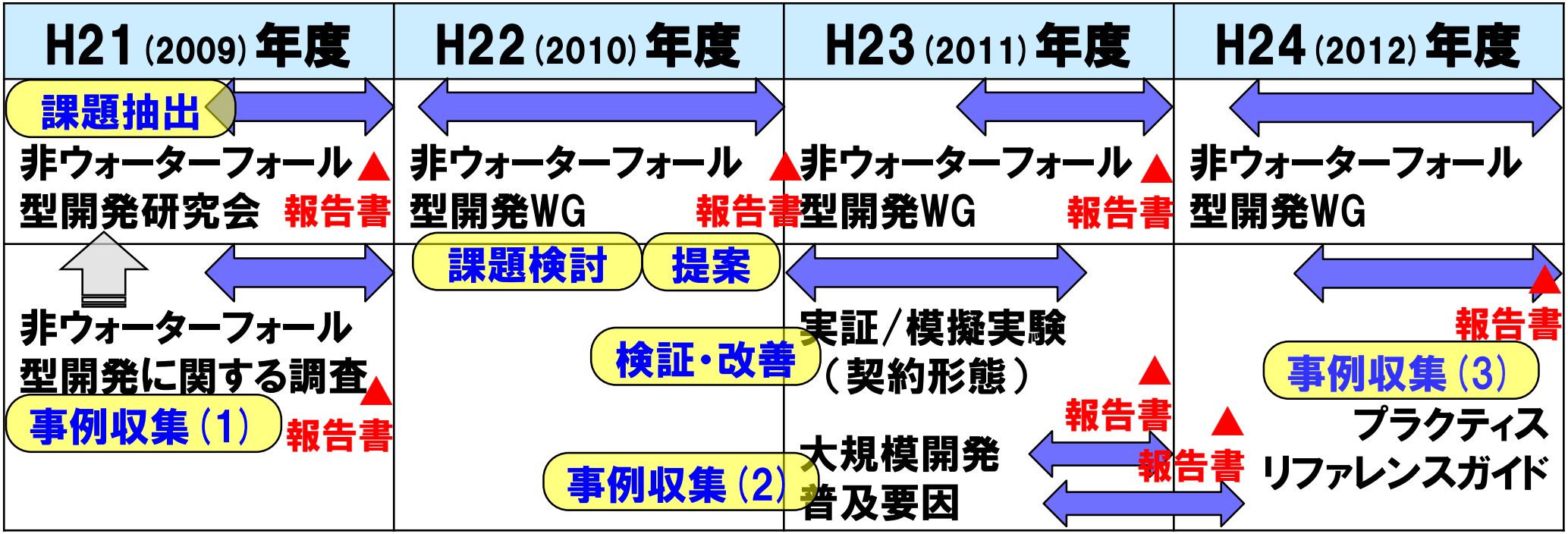
アジャイルジャパン2015　特別セッション団体交流会
2015年4月16日

独立行政法人情報処理推進機構（IPA）
技術本部 ソフトウェア高信頼化センター（SEC）
山　下　博　之

http://www.ipa.go.jp/sec/index.html

アジャイル開発に関するIPA/SECの取組み

SEC Software Reliability Enhancement Center



報告書（公開中）

H21年度版	***http://www.ipa.go.jp/sec/softwareengineering/reports/20100330a.html***
H22年度版	***http://www.ipa.go.jp/sec/softwareengineering/reports/20110407.html***
H23年度版	***http://www.ipa.go.jp/sec/softwareengineering/reports/20120326.html***
（大規模開発）	***http://www.ipa.go.jp/about/press/20120328.html***
（普及要因）	***http://www.ipa.go.jp/sec/softwareengineering/reports/20120611.html***
（プラクティス）	***http://www.ipa.go.jp/about/press/20130319.html***

（1）中・大規模開発への適用事例調査

中規模及び大規模の
非ウォーターフォール型開発プロジェクト
における課題と対策例

http://www.ipa.go.jp/about/press/20120328.html
http://www.ipa.go.jp/sec/reports/20120328.html

中・大規模開発の事例一覧（H23年度調査）

SEC
Software Reliability Enhancement Center

No.	規模	部分適用	採用手法	対象システム種別	契約
1	大		独自	B2Cサービス（SNS）	無（自社内）
2	大		Scrum	B2Cサービス（ソーシャルゲーム）	無（自社内）
3	大	○	Scrum	ゲームソフト	受託（未公開）
4	大	○	Scrum+独自	基幹システム	受託（準委任）
5	中		Scrum	B2Cサービス（会員サービス）	無（自社内）
6	中		Scrum+XP	B2Cサービス（医療・健康）	無（自社内）+オフショア*
7	中		Scrum+XP	B2Cサービス（エンタテインメント）	無（自社内）+オフショア*
8	中		XP	B2Cサービス（会員サービス）	受託（請負）
9	中	○	XP	B2Cサービス（ECサイト）	受託（請負）
10	中	○	XP	B2Cサービス（会員サービス）	受託（準委任）

中規模：30～100名，大規模：100名以上
独自：特に手法を決めず自ら定義，Scrum+XP：両手法を組み合わせて実践

*：準委任

<出典> *http://www.ipa.go.jp/sec/softwareengineering/reports/20120328.html*

適用にあたっての主な工夫（一覧）

SEC Software Reliability Enhancement Center

中・大規模開発特有の工夫

■組織体制
- チーム間ローテーション

■コミュニケーション
- 段階的朝会
- チーム跨ぎのふりかえり

■展開
- 漸進的な人数増加
- 漸進的な展開
- 社内勉強会

■分散拠点開発
- 同一拠点から分散へ
- TV会議

■アーキテクチャ
- 組織の共通基盤アーキテクチャの利用
- アーキテクチャについての教育

■システム分割/インテグレーション
- 同じリズム

■品質
- 第三者テスト

■部分適用
- 必要な部分のみ適用
- 疎結合なチーム
- 工程の見える化

小規模開発とは逆のアプローチをとる工夫

■アーキテクチャ
- 最初のアーキテクチャ構築
- アーキテクチャ専門チーム
- 運用保守チーム

■品質
- テスト・フェーズ

小規模開発と同様だが特に注意して実施する工夫

■コミュニケーション
- 完全透明性

■展開
- パイロット導入
- 認定研修・コンサルタントの利用

■分散拠点開発
- チケットシステム
- リアルタイムチャット

■アーキテクチャ
- アーキテクチャの改善

■システム分割/インテグレーション
- 疎結合で分割
- 早期からのインテグレーション
- 継続的インテグレーション

■品質
- 重視するビジネス価値
- ビジネス価値の変化
- タイムボックス優先の品質
- 自動単体テスト

(2) アジャイル開発の契約

日本におけるアジャイル開発にふさわしい契約モデルを提案

http://www.ipa.go.jp/sec/softwareengineering/reports/20120326.html

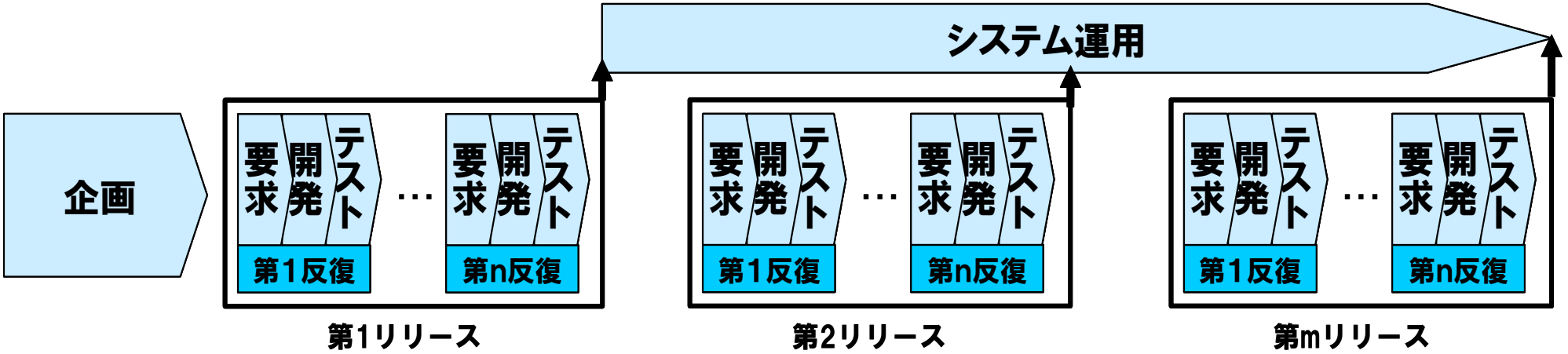
アジャイル開発には、どんな契約がふさわしいのか？

SEC Software Reliability Enhancement Center

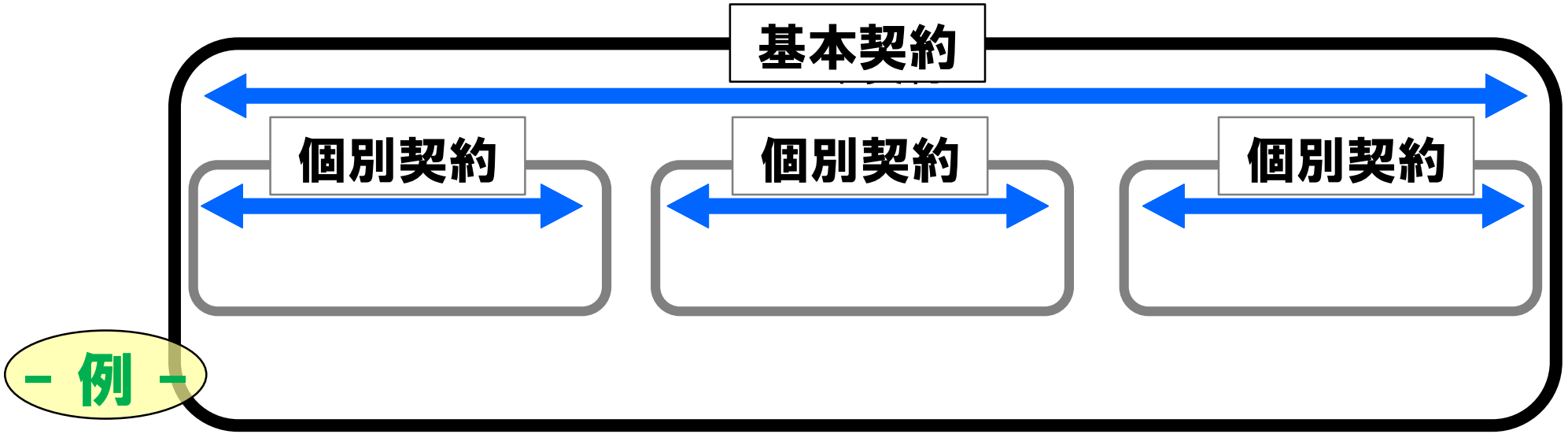
- 開発内容が決まっていない段階で、開発プロジェクト全体につき、一つの請負契約を結ぶのは適切ではない（何をいくらで完成させるか不明）。
- 他方、開発プロジェクト全体を準委任契約にすることは、ベンダが完成義務を負わない点で、ユーザ側に不安がある（たとえ成果物が完成しなくても、ユーザは対価を支払う必要）。
- また、アジャイル開発の特徴であるユーザとベンダの協働関係を、契約に取り入れる必要がある。

注）ユーザ側での労働者派遣契約に基づく開発チーム構成には，契約上の問題は特にない．
（内製傾向の高まりに伴い増加？）

基本/個別契約モデルの概要



- n=1のケースもあり。



契約

(3) アジャイル開発
プラクティス活用リファレンスガイド

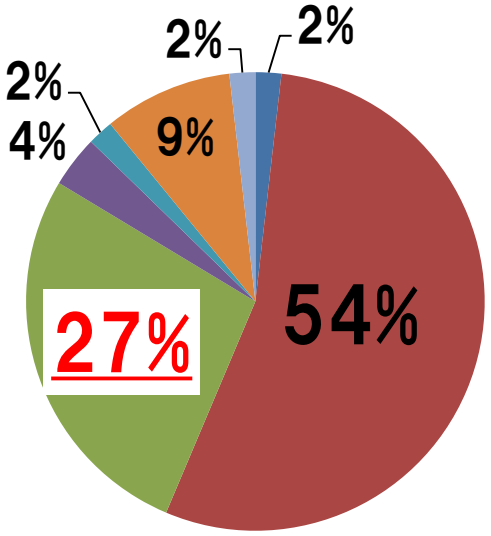
※プラクティス：アジャイル開発を実践する活動項目

http://www.ipa.go.jp/about/press/20130319.html
http://www.ipa.go.jp/sec/softwareengineering/reports/20130319.html

アジャイル開発の適用状況－国内－

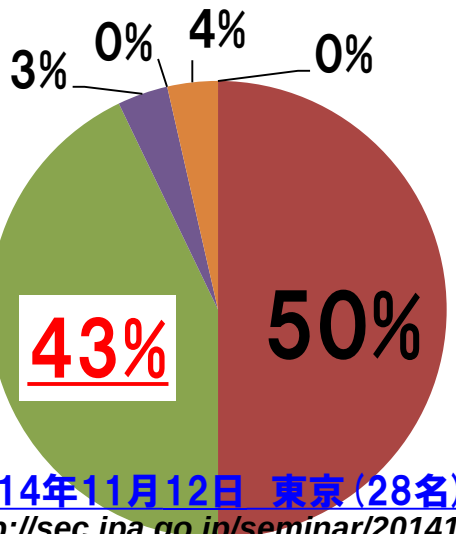
SEC Software Reliability Enhancement Center

アジャイル開発を使っていますか？

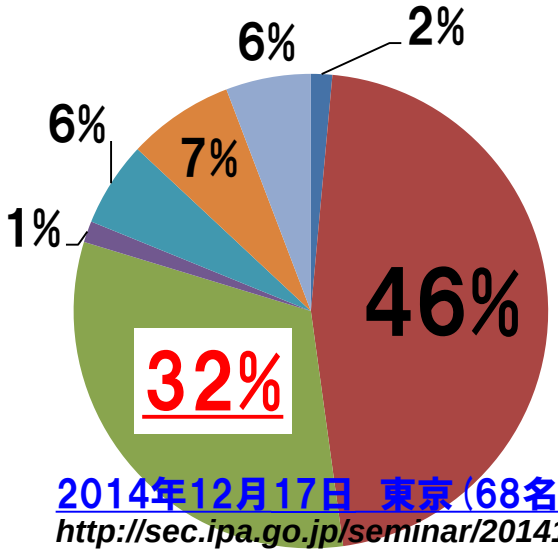


- 多くのプロジェクトで使っている
- 一部のプロジェクトで使っている
- 使いたいと考えているが実現していない
- 使う予定はない
- よく分からない／関係ない
- その他
- 無回答

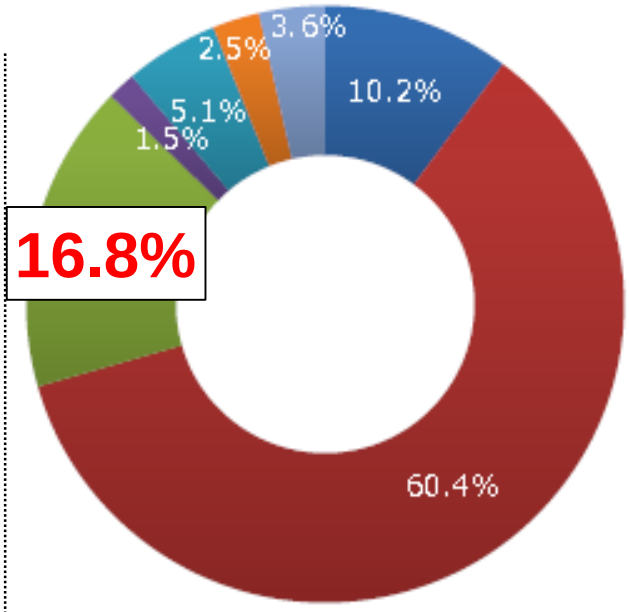
2013年10月30日　東京（55名）
http://sec.ipa.go.jp/seminar/20131030.html



2014年11月12日　東京（28名）
http://sec.ipa.go.jp/seminar/20141112.html



2014年12月17日　東京（68名）
http://sec.ipa.go.jp/seminar/20141217.html



- 多くのプロジェクトで使っている
- 一部のプロジェクトで使っている
- 使いたいと考えているが実現していない
- 使う予定はない
- よく分からない／関係ない
- その他
- 無回答

アジャイルジャパン2014
（2014.6.27，東京）
参加者アンケート（265名）

ガイドの特徴

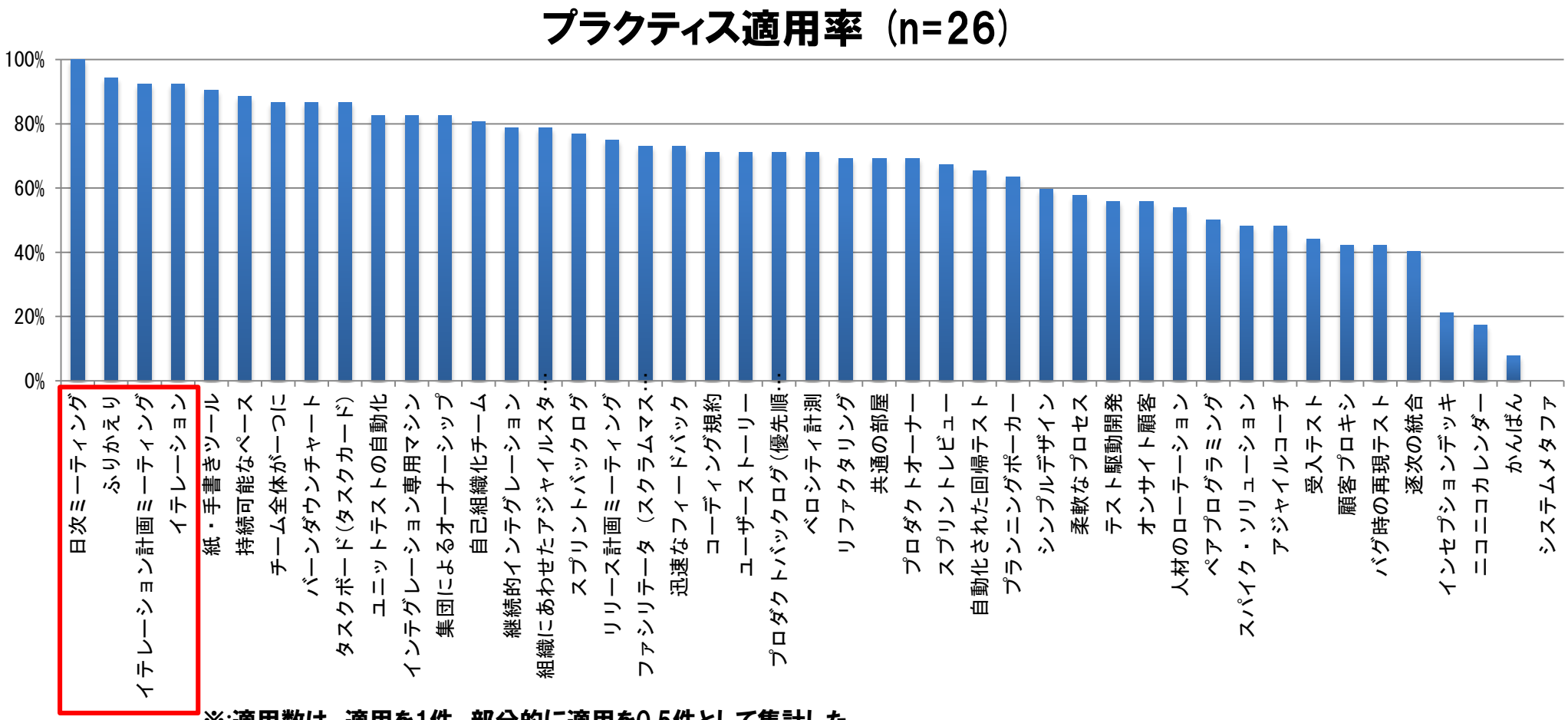
アジャイル開発を実践する活動項目

- 55個*のプラクティス，26個の事例，9つの活用ポイント　計224ページ
- 日本国内の開発現場からのヒアリングにより収集した知見を，パターン記述形式で取りまとめ
- MS-Wordファイルを公開し，クリエイティブ・コモンズ・ライセンスの下に，改変自由・営利目的利用可で使用許諾

＊ 類似のものを統合し，最終的には45個

適用プラクティス（全体）

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。



※:適用数は、適用を1件、部分的に適用を0.5件として集計した。
※ 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。

プラクティス例概要 - 日次ミーティング

SEC Software Reliability Enhancement Center

状況

チームは、プロジェクトのタスクをこなすためにほとんどの時間を使い、状況や情報の共有のために取れる時間がほとんどない。

問題

情報の共有遅れが問題を大きくする。
情報共有の時間が取れないまま、状況認識と問題対処への判断が遅れると、問題が大きくなるなど、より深刻な状況を招いてしまう。

フォース

関係者が多忙なため、情報共有のための時間が取れない。
情報共有の間隔が空いてしまうと、情報量が増え、共有に必要な時間が余分にかかる。

解決策

必要な情報を短い時間で毎日共有する。
関係者が長時間、時間を取れないようであれば、短い時間（15分を目安に）で済むように、共有を必要な情報に絞る。

利用例

- **事例**（9）：遠隔地にいるメンバーも日次ミーティングに参加するため、チャットツールや電話会議システムを利用した。
- **事例**（17）：1日3回（朝、昼、夕）10分程度のミーティングを実施。問題を報告/解決するためのリズムが開発メンバー全員に浸透して、短期での問題提起ができている。

留意点

- 必ずしも朝の時間帯にこだわらず、関係者が集まりやすい時間帯に開催する（例えば、終業近い時間帯に開催する夕会）。

プラクティス例概要 - ふりかえり

SEC Software Reliability Enhancement Center

状況

イテレーション毎に、チームは動くソフトウェアとして成果を出そうとしている。イテレーションを繰り返して、チームはソフトウェアを開発していく。

問題

開発チームは、そこに集まったメンバーにとって最適な開発プロセスを、最初から実践することはできない。

フォース

イテレーションでの開発はうまくいくこともあるが、うまくいかないこともある

解決策

反復内で実施したことを、反復の最後にチームでふりかえり、開発プロセス、コミュニケーション、その他様々な活動をよりよくする改善案をチームで考え実施する機会を設ける。

※1 メンバー全員で、Keep（よかったこと・続けたいこと）、Problem（問題・困っていること）、Try（改善したいこと・チャレンジしたいこと）を出し合い、チームの改善を促す手法。

利用例

- **事例（25）**：当初はKPT[※1]を用いてふりかえりを行っていたが、ファシリテータの技量にふりかえりの質が依存してしまう、声の大きいメンバーに影響を受けてしまうことに気づいた。そのため、意見を集めるやり方として、555（Triple Nickels）[※2]を用いることにした。

留意点

- ふりかえりにチームが慣れていない場合は、進行で各人の意見をうまく引出すようにしないとうまくいかない。
- 問題点を糾弾する場にしてしまうと、改善すべき点を積極的に話し合えなくなってしまう。
- 改善案を出しても、実際に実行可能なレベルの具体的なアクションになっていないと実施されない。

※2 アクションや提案に対するアイデアを出すための手法。5人程度のグループで、各人が5分間ブレインストーミングをしてアイデアを書き出す。5分経過したら紙を隣の人にまわし、新しいアイデアを書き加える。

プラクティス例概要 - イテレーション計画ミーティング

状況

開発を一定期間のサイクル（イテレーション）で繰り返し行っている。
プロダクトバックログの内容を、チームとプロダクトオーナーの間で合意している。

問題

リリース計画は遠い未来の目標のため、それだけではイテレーションで何をどのように開発すれば良いか分からない。

フォース

ユーザーストーリーのまま、イテレーションの詳細な計画を立て、開発を進めていくのは難しい。

解決策

イテレーションで開発するユーザーストーリーと、その完了までに必要なタスクおよびタスクの見積りを洗い出すミーティングを開く。

利用例

- **G社事例（9）**：ペーパープロトタイピング[※1]を用いたUIデザインの共有と受入れ条件の確認をイテレーション計画ミーティングで行っていた。そのため、計画にはかなり時間を要していたが、見積りの前提が具体的になったため、見積り作業時間の削減に繋がった。

留意点

- 見積りに関してチームが水増しする懸念を持つかもしれないが、チームを信じるべきである。プロジェクトの目的を理解したチームは、見積りが大きく外れるようであれば、自らその原因を分析し、次の見積りに活かすはずである。

※1 紙などを使った試作品でユーザビリティテストを行うこと。

事例概要 <<中大規模適用プロジェクトの事例>> 事例(4) C社

プロフィール

既存のサービスのリプレイス開発。単純なサービスのリプレイスではなく、新しい要件も加えながら開発したいとの要望があり、C社から顧客にアジャイル開発を提案して開始した。
リプレイスといいながらも、顧客から要件を聞き出しながら開発を進めていった。要件が固められない部分のみアジャイル開発を行い、要件が明らかな部分についてはウォーターフォール型開発を実施した。

システム種別	B2Cサービス
規模	中規模 開発者 32名 インフラ 4名 管理その他 23名 計 59名
手法	XP
契約	準委任契約（四半期毎に更新）
期間	2年
開発拠点	東京、地方を含めた3拠点

特徴的なプラクティス

- **日次ミーティング**: 複数のチームが存在したため、二段階の構成で実施していた。(チーム間→チーム毎)。
- **ふりかえり**: チーム毎に実施した場合には、他のチームへの不満などばかりになってしまい機能しなかった。そのために、複数チームの混成で実施することにより、問題へ集中するように意識を変えさせた。また、反復毎のふりかえりとは別に、四半期単位でも実施して大きな改善点について話しあった。
- **顧客プロキシ**: 当初は顧客に要件管理をしてもらっていたが、機能しなくなったため、C社の社員が顧客の会社へ出向して顧客プロキシとなり全面的に支援した。

活用のポイント (1)

SEC Software Reliability Enhancement Center

(1) 短納期、開発期間が短い

開発対象のボリュームに比して、開発期間が短い場合、チームの開発速度を計測し、そのスピード感で、予定している開発量が期限内に完了するのか、常に点検する必要があるため、「**ベロシティ計測**」と、「**バーンダウンチャート**」を活用する。
ベロシティ計測は、関係者であるプロダクトオーナーが理解できる基準で計測する必要がある (H社事例 (11))。**バーンダウンチャート**は、関係者と定期的に共有する機会を設けることが活用のポイントである (B社事例 (2)、J社事例 (17) (18))。

(2) スコープの変動が激しい

開発中に要求の変更が頻繁に発生することが予想されるプロジェクトでは、チームが扱う要求の全体像と状態、直近のイテレーションで何を開発するかが分かっており、柔軟に優先順位を変えられる必要があるため、「**プロダクトバックログ(優先順位付け)**」、「**スプリントバックログ**」および「**プロダクトオーナー**」を活用する。
プロダクトバックログ(優先順位付け)は、イテレーション毎に整理を行い、チーム全員で優先順位と内容を合意すると良い (B社事例 (2))。
プロダクトオーナーは、業務や全社的に全体最適となる判断を行うこと (G社事例 (10))。

(3) 求められる品質が高い

品質要求が高いプロジェクトでは、テストに関するプラクティスである「**自動化された回帰テスト**」、「**ユニットテストの自動化**」を活用する。
自動化された回帰テストや**ユニットテストの自動化**は、プロジェクトの初期段階で、実施有無、実施のための取決め、使用ツールを検討しておくことがポイントである。これを後回しにすると、必ず機能開発が優先され、自動化にたどりつかない (B社事例 (2))。

活用のポイント (2)

SEC Software Reliability Enhancement Center

(4) コスト要求が厳しい

必要のないものを作るムダをなくし、必要なものをより素早く提供することがROI(費用対効果)の向上につながり、コスト要求に応えることができる。そのためには、的確に顧客の要求を把握し、認識の相違をなくす必要があるため、「**プロダクトバックログ(優先順位付け)**」を活用する。
また、開発機能がプロダクトオーナーの意図通りになっているか否かの検証のために、「**受入テスト**」を活用する。「**オンサイト顧客**」には、優先順位や仕様の確認がその場で確認することができ、迅速に方針を決められるというメリットがある(K社事例(20))。

(5) チームメンバーのスキルが未成熟

スキル的に未成熟なメンバーが成長していく機会として、プロジェクトを計画する必要があるため、「**ペアプログラミング**」と「**ふりかえり**」を活用する。
ペアプログラミングは、ベテランとメンバーが一緒に仕事をすることで、技術的な指導を行うのに適したプラクティスである(C社事例(4))。
ふりかえりは、メンバーの成長の機会として捉えることができる。ふりかえりのやり方自体も見直しながらチームに適したやり方を模索すると良い(E社事例(6))。

(6) チームにとって初めての技術領域や業務知識を扱う

プロダクトの背景にある業界の知識や、要求の理解と実装に必要な業務知識の獲得が必要となるため、「**スパイク・ソリューション**」と「**システムメタファ**」を活用する。
スパイク・ソリューションを適用することは、リスクとなりそうな技術課題について、プロジェクトの初期段階で実験的に小さく試しておくことであり、チームとプロジェクトを後々助けることに繋がる(C社事例(4))。**システムメタファ**は、開発者にとって、なじみの薄い業務知識を理解する手段として、有効と考えられる。

活用のポイント (3)

SEC Software Reliability Enhancement Center

(7) 初めてチームを組むメンバーが多い

初めてチームを組むメンバーが多い場合、チームが向かう方向を明確にすることと、チームビルディングが必要となるため、「インセプションデッキ」や「ニコニコカレンダー」を活用する。
インセプションデッキは、作成を通じて、プロジェクトの目的や目標が明らかとなる(B社事例(1))。
ニコニコカレンダーは、メンバーの感情や状況を可視化し、チームメンバーのことを知ることがポイントになる(E社事例(6))。

(8) オフショアなど分散開発を行う

プロダクトオーナーと開発チームが別の拠点にいる場合、オンラインでのコミュニケーション手段を検討し、頻繁にコミュニケーションが取れるようにする必要があるため、「日次ミーティング」や「顧客プロキシ」を活用する。
TV会議システムを使った日次ミーティングは、離れた者同士が毎日顔を合わせる機会として、ぜひ活用するべきである(G社事例(9))。顧客プロキシは、分散した環境下でも、迅速なフィードバックが得られる工夫をしなければならない。

(9) 初めてアジャイル開発に取り組む

初めてアジャイル開発に取り組む際には、書籍や文書だけではなく人から人にやり方を伝えることが有効であるため、社内にアジャイル開発に取り組んだ経験のある人がいる場合はその人に、社内にない場合は、社外からアジャイルコーチを頼んで導入の手伝いをしてもらうのがよい。初めて取り組む場合は、イテレーション期間を短くした上で、ふりかえりの中で改善点をチームで考え実行していくことが不可欠となる。

ご清聴，ありがとうございました

アジャイル開発に関するIPA/SECの検討結果等は：
http://www.ipa.go.jp/sec/softwareengineering/std/ent02-c.html