

IPA『アジャイル型開発プラクティス・ リファレンスガイド』とパターン

2013年8月27日

合同会社カルチャーワークス

共同代表 本橋 正成

目的

- 目的
 - 独立行政法人情報処理推進機構(IPA)「アジャイル型開発におけるプラクティス活用事例調査」リファレンスガイドの構成を知り、活用できるようになる。
 - パターンと、パターンを採用した理由を知り、自らの仕事場やプロジェクトにおける次のステップへ進む示唆を得る。
- 対象者
 - アジャイル型開発の導入を検討しているソフトウェア開発に関係する方。
- 内容
 - 該当調査のリファレンスガイドの概要と活用方法を説明する。
 - パターンと、パターンを適用した理由を説明する。

本資料は、IPA「アジャイル型開発におけるプラクティス活用事例調査」を参照・引用しています。

アジャイル型開発における プラクティス活用事例調査とは

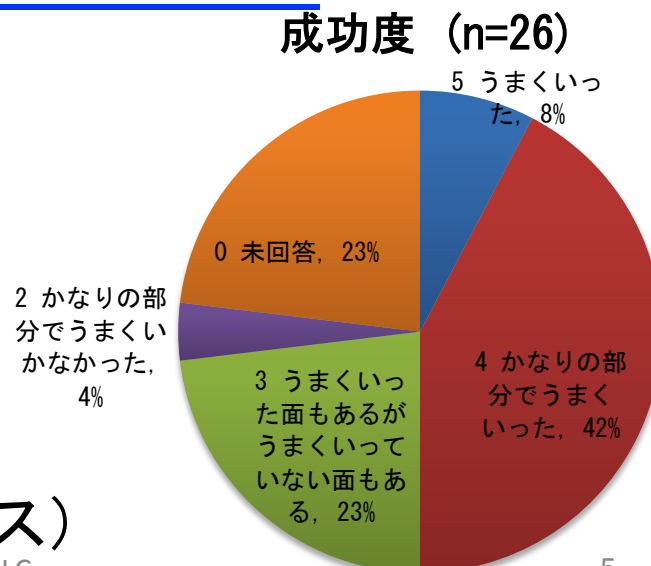
- 2013年3月公開

“国内でアジャイル型開発の活用が進む先駆的企業の59件のプラクティスの利用状況を調査、分類し、プラクティスのプロジェクト規模やアジャイル型開発の種類などによる適用状況の違い、適用する場合の工夫や留意点などについて、調査報告書とリファレンスガイド(以下本ガイド)としてまとめました”

<http://www.ipa.go.jp/sec/reports/20130319.html>

- 公開されている資料

- 調査概要報告書 (39ページ)
- 調査報告書 調査編 (49ページ)
- 調査報告書 ガイド編 (231ページ)
- リファレンスガイド (Word形式、
クリエイティブ・コモンズ・ライセンス)



調査概要報告書10ページより

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

- 2012年3月公開
“中規模及び大規模プロジェクトの事例を収集し、非ウォーターフォール型開発の大規模プロジェクトにおける課題の解決方法やその他の課題などに対する工夫等の調査を実施”

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

- 公開内容
 - 調査概要報告書
 - 調査報告書

調査概要報告書10ページより

非ウォーターフォール型開発の採用理由

SEC
Software Engineering
for Mo-No-Zu-Ku-Ri



アジャイル型開発におけるプラクティス活用事例調査

調査概要

リファレンスガイド目次(抜粋)

1. 目的
2. 使用方法
3. プラクティス解説
 1. プロセス・プロダクト
 2. 技術・ツール
 3. チーム運営・組織・チーム環境
 4. 発見されたプラクティス
 5. プラクティス適用時のよくある問題と対応策
4. 活用事例
5. 活用のポイント
6. 参考文献
7. 索引

付録1. 発見された工夫と課題

付録2. 事例とプラクティスの対応

参照:リファレンスガイド 目次

「目的」「背景」「課題」より

背景

- 世界的にアジャイル型開発が主流になりつつある
- 日本では普及が遅れている

課題

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

目的

- 現場導入のナレッジ収集と活用するためのTips集づくり
- 日本の現場での実践事例を広く紹介

参照：調査概要報告書 2-4ページ

「調査方法」より

調査
ポイント

- 調査対象プラクティスの策定
- 国内外のアジャイル型開発の調査

調査票

- コンテキスト情報の入手
- プラクティスの活用状況

ヒアリング

- コンテキストの確認
- 独自の工夫について詳細調査

分析

- パターンとして記述
- 集計

参照: 調査概要報告書 6ページ

「調査対象」より

- 26事例13社
 - できるだけ多様なプロジェクトの事例を収集
 - 中大規模適用プロジェクトを含む
 - 複数手法の組み合わせプロジェクトを含む
- 調査種別
 - システム種別 (B2C 61%、社内システム31%)
 - 契約 (自社開発34%、自社開発(一部オフショア)19%、受託開発(請負)35%)
 - 手法 (Scrum 35%, XP 19%, Scrum+XP 23%, Scrum+WF 15%)
 - 規模 (小規模77%、中規模15%、大規模8%)

参照: 調査概要報告書 7-10ページ

「ガイドの使い方」より

- 興味のあるカテゴリのプラクティスを知る
 - 「3.プラクティス解説」を参照のこと。
- プラクティス適用の際に発生している問題への対応策を探す
 - 「3.5.プラクティス適用時のよくある問題と対応策」を参照のこと。
- 特定の事例で活用されているプラクティスを探す
 - 「4.活用事例」を参照のこと。
- 読者の組織、プロジェクトの状況に対してのお勧めプラクティスを探す
 - 「5.活用のポイント」を参照のこと。
- プラクティス群活用に際しての留意点(アンチパターン等)を知る
 - 「3.プラクティス解説」の各プラクティスの留意点を参照
- 参考文献を探す
 - 各プラクティスは、各プラクティスの紹介ページ末
 - 複数のプラクティス、巻末の参考文献
- よく使われているプラクティスを知る。
 - 概要調査報告書の「適用種別」、調査報告書(調査編)の分析結果を参照。

参照:リファレンスガイド 4ページ

アジャイル型開発におけるプラクティス活用事例調査

プラクティスガイド プラクティス紹介

「プロセス・プロダクト」より

プラクティス名	問題
リリース計画ミーティング	いつどのような価値をユーザーに届けるのかという目標がなければ、ただイテレーションを繰り返しているだけになってしまう。
イテレーション計画ミーティング	リリース計画は中長期的な目標のため、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わからない。
イテレーション	新しいプロダクトの開発では、プロダクト及びプロジェクトについての知識が不足している。
プランニングポーカー	開発の初期ほど開発対象に対する知識が不足している。
ベロシティ計測	プロダクトの開発規模が見積もれたとしても、チームのイテレーションあたりの開発量がわからなければ、リリース日を見積もることができない。
日次ミーティング	情報の共有遅れが問題を大きくする。
ふりかえり	開発チームにとって、最初から最適な開発プロセスを実践することは困難である。
かんぱん	事前に計画を立てたいが、いつ、どのくらいの要件が発生するのが予測できない。
スプリントレビュー	イテレーションの成果を関係者で確認する機会がなければ、現在の状況を判断し、次のイテレーションの適切な計画を立てることができない。
タスクボード(タスクカード)	今、チームの抱えるタスクが、どれに取り掛かり中で、どのような状態になっているかわからない。
バーンダウンチャート	イテレーションをこなしているだけでは、プロジェクト全体が順調に進んでいるのか危険な状態にあるのかわからない。
柔軟なプロセス	アジャイル型開発をしているが、しっくりこない。組織の状況と目的に対して適切な状態になっていない。
ユーザーストーリー	要求を明確に伝えようと、ドキュメントを詳細に書いたとしても、ドキュメントを通してのコミュニケーションでは、憶測や誤解を招きやすい。
スプリントバックログ	そのイテレーションでどのプロダクトバックログアイテムを開発するのか、どのような手順で実現していくのか、必要な作業は何か明らかにしていない。
インセプションデッキ	プロジェクトについての認識がステークホルダーの間でそろっていない。
プロダクトバックログ(優先順位付け)	プロダクトを開発する上で、何から作業に取り掛かれればよいかかわからない。
迅速なフィードバック	あらゆるアウトプットについてフィードバックがないと効果や価値があるのかわからない。

「技術・ツール」より

プラクティス名	問題
ペアプログラミング	知識をより素早くチーム全体に広げたい。
自動化された回帰テスト	テストは何度も繰り返し実施しなければならないため、ミスなく、短時間で終わらせるようにする。
テスト駆動開発	新しくプロダクトコードを書いている最中も、今書いているコードが正しい仕様に基づいているのか、バグを組み込んでいないか不安である。
ユニットテストの自動化	ユニットテストを手動で実施するには、多くの問題がある。
受入テスト	ユーザーストーリーには顧客が受け入れることができる条件が不可欠である。
システムメタファ	システムの設計は、抽象的な思考だけで行われることが多い。システムを理解するために、開発者とユーザーの双方に通用する共通の用語が存在しない。
スパイク・ソリューション	情報が足りない中で推測を元に開発を進めることはリスクである。
リファクタリング	変更や修正を繰り返している中でコードがカオス状態になり、バグが入り込む余地ができてしまう。
シンプルデザイン	変更を予測して事前に設計を行うことは難しい。
逐次の統合	統合する際に、複数の修正が含まれていると、どこが問題であるかわからなくなる。
継続的インテグレーション	開発環境では動作していても、環境が変わったり、他人の変更と一緒にすると動作しない場合がある。
集団によるオーナーシップ	ソースコードや業務に関する知識が属人化されると、他の人がそのタスクを実施することができずに、作業を平準化することができず特定の人に作業が集中してしまう。
コーディング規約	チームで開発を行うためには、合意形成が大切になる。
バグ時の再現テスト	不具合が発見されたら、その原因をつきとめて修正し、再発を防がなければならない。
紙・手書きツール	同じ場所で作業をするチームの間で、情報共有やアイデア出しのためには、どんなツールを用いればよいのだろうか？

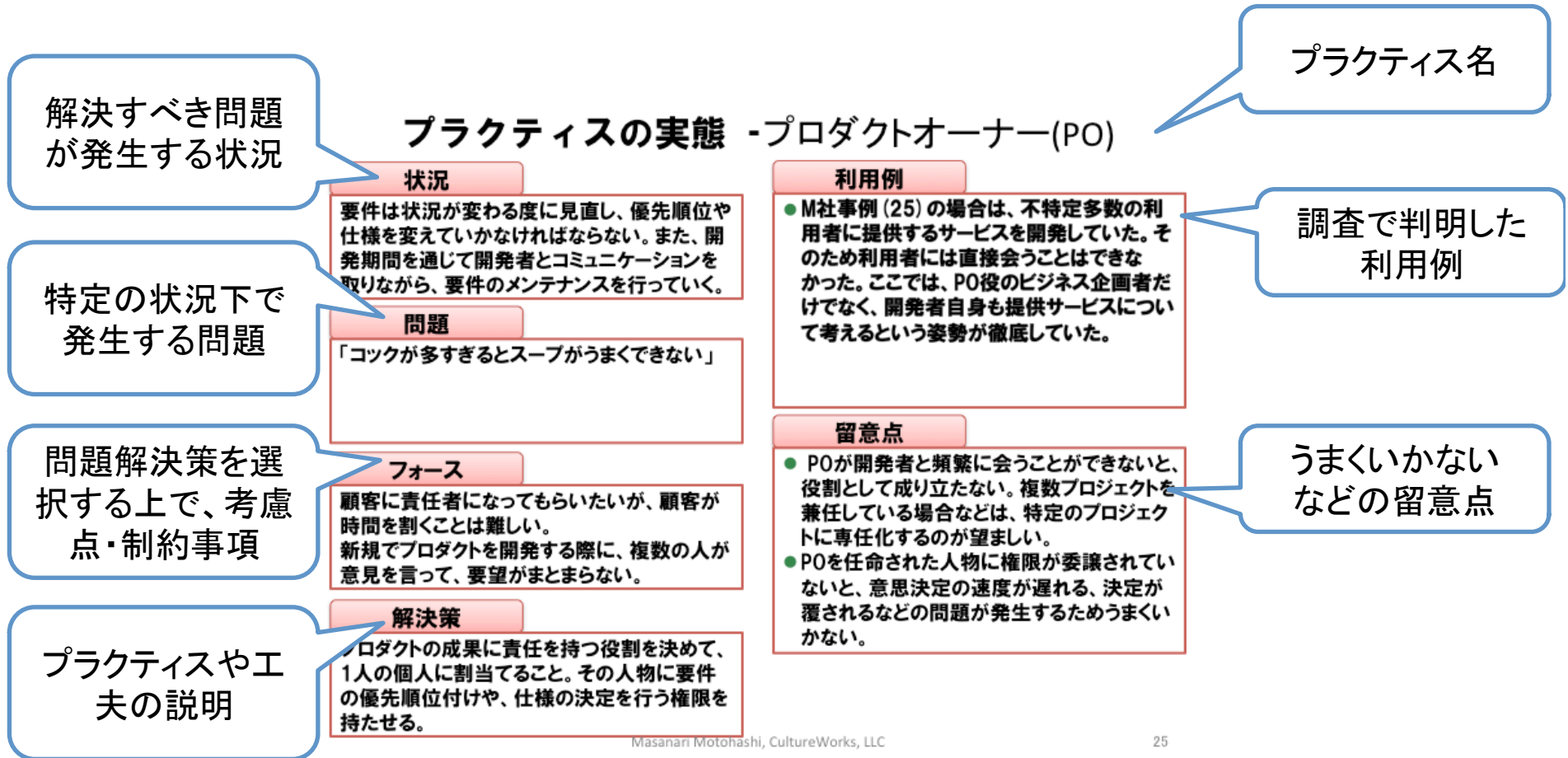
「チーム運営・組織・チーム環境」より

プラクティス名	問題
顧客プロキシ	顧客の声を聞いていないために、顧客を無視した設計、開発を行ってしまう。結果、顧客の期待するプロダクトが出来上がらない。
オンサイト顧客	文書やオンライン上のやりとりではコミュニケーションは充分とは言えない。
プロダクトオーナー	「コックが多すぎるとスープがうまくできない」日本語では「船頭多くして船山に登る」
ファシリテータ(スクラムマスター)	自らの状況を客観的に見ることができない組織では、問題を認識し、改善や運用が難しい。
アジャイルコーチ	アジャイル型開発は教科書通りに実施しても、簡単にはうまくいかない。
自己組織化チーム	複雑な領域の詳細までPMが把握することは困難で工数がかかる。
ニコニコカレンダー	メンバーに起きている異常に気付くことができない。
持続可能なペース	無理をしてでも、高い効果や生産性を維持したい。
組織に合わせたアジャイルスタイル	アジャイル型開発をしているが、しっくりこない。組織の状況と目的に対して適切な状態になっていない。
共通の部屋	チームや関係者のコミュニケーションがうまくいっていない。活気がない。
チーム全体が一つに	チームが一つになっていなければ、互いの作業、学習をサポートすることはなく、結果として成果が上がらない。
人材のローテーション	チームや組織内で専門知識や業務知識などの偏りが発生し、属人化している。
インテグレーション専用マシン	インテグレーション環境が、共用だったり、遅かったり、本番環境と異なっている場合は、インテグレーションの価値が下がってしまう。

「発見されたプラクティス」より

プラクティス名	問題
ユーザーストーリーマッピング	ユーザーストーリーやプロダクトバックログだけでは、プロダクトの全体像は把握しづらい。全体像を把握した上でスコープを決めないとプロダクトの価値提供がうまくいかない。
「完了」の定義	チームの中で「完了」についての認識が揃っているだろうか？
楽しい工夫	普段の仕事の中で、ちょっとした違和感を見つけたり、心理的なストレスの解消法を模索したり、工夫できる場所を見つけたりしているが、なかなか改善に結びつかない。
組織構造のバウンダリをゆるめる	組織やプロセスの構造が生み出すボトルネックがある。

パターンのフォーム(形式)



参照: 調査概要報告書 18ページ

プラクティスの実態 - プロダクトオーナー(PO)

状況

要件は状況が変わる度に見直し、優先順位や仕様を変えていかなければならない。また、開発期間を通じて開発者とコミュニケーションを取りながら、要件のメンテナンスを行っていく。

問題

「コックが多すぎるとスープがうまくできない」

フォース

顧客に責任者になってもらいたいが、顧客が時間を割くことは難しい。
新規でプロダクトを開発する際に、複数の人が意見を言って、要望がまとまらない。

解決策

プロダクトの成果に責任を持つ役割を決めて、1人の個人に割り当てること。その人物に要件の優先順位付けや、仕様の決定を行う権限を持たせる。

利用例

- M社事例(25)の場合は、不特定多数の利用者に提供するサービスを開発していた。そのため利用者には直接会うことはできなかった。ここでは、PO役のビジネス企画者だけでなく、開発者自身も提供サービスについて考えるという姿勢が徹底していた。

留意点

- POが開発者と頻繁に会うことができないと、役割として成り立たない。複数プロジェクトを兼任している場合などは、特定のプロジェクトに専任化するのが望ましい。
- POを任命された人物に権限が委譲されていないと、意思決定の速度が遅れる、決定が覆されるなどの問題が発生するためうまくいかない。

参照:リファレンスガイド 85-87ページ

プラクティスの実態 - 楽しい工夫

状況

職場環境がギスギスした雰囲気であったり、プレッシャーが高い状況であったり、良くない状態である。

問題

普段の仕事の中で、ちょっとした違和感を見つけたり、心理的なストレスの解消法を模索したり、工夫できるところを見つけたりしているが、なかなか改善に結びつかない。

フォース

トラブルに対処する一番の方法は、そのトラブルにまっすぐ対応することではあるが、その状況のプレッシャーから、気疲れしてしまう

解決策

自分たちで工夫を楽しもう。また、その楽しい工夫を受け入れよう。単に楽しむだけでなく、「楽しく」かつ「役に立つ」工夫を発見しよう。

利用例

- B社事例(3)では、バグが発生した場合に、バグという言葉から受けるネガティブなイメージで開発者はストレスフルな状況であった。そこでチームはバグというネガティブな表現ではなく、「ラーメンの具」を追加するというポジティブなイメージでバグの記録を付けるようにした。

留意点

- 複雑さや困難に打ち勝つためには、ルールに従うだけでは難しい場合がある。状況をよく把握し、いろいろ試す環境を用意する。
- このような取り組みを批判し、止めさせることは簡単だ。しかし、試したいと思ったことを試せる環境が、次のイノベーションに結びつく。

参照:リファレンスガイド 115-116ページ

プラクティスの実態 - 組織構造のバウンダリをゆるめる

状況

プロダクトには、様々な組織が関わっている。チームもひとつの組織であるし、その周囲にも様々な組織がある。

問題

組織やプロセスの構造が生み出すボトルネックがある。

フォース

バウンダリを明確に切り分けようと思っても、複雑で現実的ではないが、その周辺こそ創造性を発揮しやすいところである。

解決策

組織的な構造が生み出すバウンダリをゆるめよう。お互いに役割を越えてチームとして助けあおう。

利用例

- D社事例(5)では、スクラムにおいてプロダクトオーナーと開発チームにわかれている状況であったが、プロダクトオーナーは、ビジネス企画がつまらなくなり、ボトルネックになることもあった。そのため、プロダクトオーナーも開発について知り、開発チームもプロダクト企画に関わり、より一体になることを目指した。

留意点

- 構造を変更させた結果、また新たな視点に基づいた問題に行き当たることがある。適切なフィードバックをしながら検討する。
- 第三者機関などの必要性については、経営者や必要な各部門などと調整する必要がある。

参照:リファレンスガイド 117-118ページ

「プラクティス適用時の よくある問題と対応策」より

1. プロダクトオーナーがボトルネック
2. 分散拠点で円滑なコミュニケーションがとれない
3. テストの自動化が後回しになってしまう
4. リファクタリングができない
5. 外から進行状況がわかりづらい
6. 真の顧客からのインプットやフィードバックがない

参照:リファレンスガイド 119-128ページ

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 ガイド編

活用のポイント紹介

「活用のポイント」より

1. 短納期、開発期間が短い
2. スコープの変動が激しい
3. 求められる品質が高い
4. コスト要求が厳しい
5. チームメンバーのスキルが未成熟
6. チームにとって初めての技術領域や業務知識を扱う
7. 初めてチームを組むメンバーが多い
8. オフショアなど分散開発を行う
9. 初めてアジャイル型開発に取り組む

参照:リファレンスガイド 182ページ

「活用のポイント」より

短納期、開発期間が短い

バーンダウンチャートを作成し、ベロシティ計測とで開発速度を計測する。ステークホルダーとの調整が大切なので、プロダクトオーナーによるステークホルダーとの調整が大切になる。

スコープの変動が激しい

プロダクトオーナーは、プロダクトバックログで要求全体とその順序を管理する。イテレーションで実施するスプリントバックログを作成する。

参照:リファレンスガイド 183ページ

アジャイル型開発におけるプラクティス活用事例調査

付録1. 発見された工夫と課題

「ふりかえり」より

名前	混成チームでふりかえり
状況	顧客のオフィスは離れた場所になるが、週に2～3日は開発ルームを訪れる。開発チームでふりかえりを実施していた。ふりかえりの初心者が多かった。プロジェクトは複数チームで構成されている。
問題	他チームへの悪口が増えてしまい改善が行われない。
制約事項	自分達で自身の仕事の進め方を改善するという、ふりかえりの趣旨が理解されていない。
解決策	参加メンバーの一部を他チームとメンバーと入れ替えたり、複数チーム合同でふりかえりをやることにした。
結果	手本となるチームの振る舞いや行動を学び、改善が行われるようになった。

名前	インフォーマルなふりかえり
状況	ふりかえりをおこなっている。
問題	ふりかえりを実施しても、なかなか本音で話すことは困難である。
制約事項	緊張している雰囲気の中では話しにくい。
解決策	定期的にインフォーマルに集まれる場所を用意し、ざっくばらんにビジネス状況の話をしてふりかえる。
結果	-

名前	褒めるふりかえり
状況	ふりかえりをおこなっている。
問題	発言者に対し批判的な意見を言う人がいると、自分の発言も批判されるのではないかと思い、様々な意見が言いにくく、発言しにくい雰囲気になってしまう。
制約事項	-
解決策	ふりかえりの中でメンバー同士を褒め合った。
結果	

アジャイル型開発におけるプラクティス活用事例調査
 アジャイル型開発におけるプラクティス活用リファレンスガイド
<http://www.ipa.go.jp/sec/reports/20130319.html>

「コミュニケーション」より

名前	オンライン・チャットによるコミュニケーション
状況	顧客のオフィスは離れた場所になるが、週に2~3日は開発ルームを訪れる。
問題	物理的な距離があるため、確認事項がある場合にもすぐに聞くことができない。
制約事項	コミュニケーション頻度を上げたいが、顧客もしくは開発メンバーが間の場所に常駐することは困難である。
解決策	チャットシステムを使って顧客とリアルタイムに連絡がとれるようにした。
結果	頻繁にコミュニケーションをとれるようになった。
課題	顧客の作業時間を奪ってしまう。

名前	メンバー固定
状況	オフショアも含めた中長期(数年スパン)のプロジェクトである。
問題	開発チームのメンバーが変わると、新たに教育する必要がある。
制約事項	メンバーは減ることがあっても増えることがないプロジェクト状況である。
解決策	開発チームのメンバーの入れ替えが起こらないように手配した。
結果	開発や業務に関するノウハウが成熟した。

名前	係のローテーション
状況	チームにはメインの開発以外にも色々な仕事がある。(例:バーンダウンチャートのプロット)
問題	開発以外の仕事も、誰かがやらなければならない。
制約事項	リーダーが指示することはない。
解決策	XXX係を任命して、一週間単位で回すようにした。
結果	全員が責任をもって開発以外の仕事をできるようになった。

アジャイル型開発におけるプラクティス活用事例調査
 アジャイル型開発におけるプラクティス活用リファレンスガイド
<http://www.ipa.go.jp/sec/reports/20130319.html>

「まとめ」と「提言」より

- 三大適用プラクティス
 - 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
- プラクティス適用増加
 - Scrumの認知向上
- 規模の違いが適用するプラクティスに及ぼす影響あり
 - 契約やシステム種別は、あまり影響がない。
- 事例(コンテキスト)にあわせた工夫
 - 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」
 - 180以上の独自の工夫
- 特に重要なプラクティス
 - ファシリテータ(スクラムマスター)
 - 顧客プロキシ
 - 共通の部屋

参照: 調査概要報告書 25-27ページ

アジャイル型開発におけるプラクティス活用事例調査

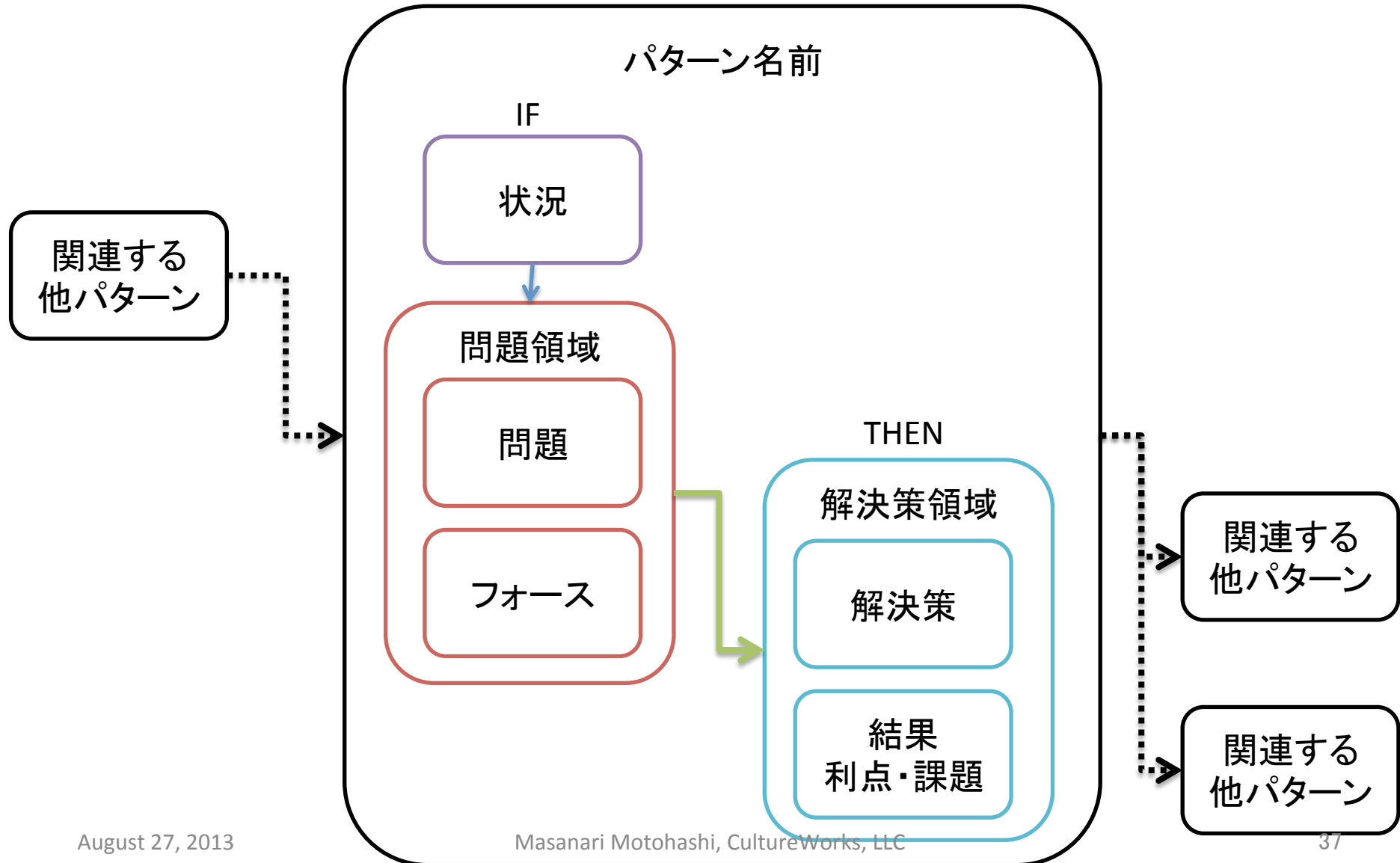
パターンについて

パターンとは？

アレグザンダーの定義

“各パターンが、環境の中で繰り返し繰り返し発生する問題を記述し、その問題に対する解決策の要点を記述します。その解決策は、全く同じことを繰り返すことなく、何百万回も適用可能であるようにします。”

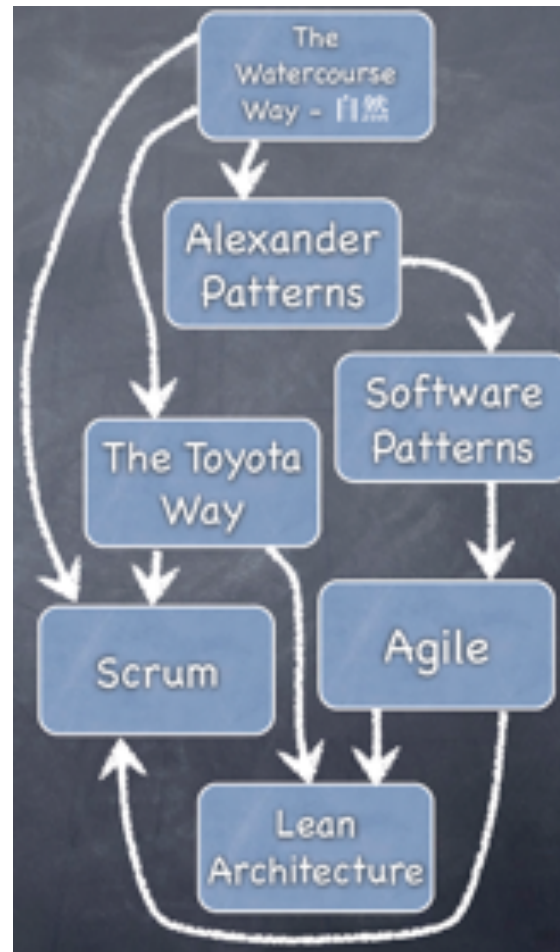
パターンの構造



なぜパターンなのか

- 状況と問題に適した解決策を適用できる。
- アジャイル型開発は、パターンと親和性が高い。
- ソフトウェアやITに関する分野では、パターンが十分に利用されている。
- パターンは、複数の解決策を組み合わせることができる。
- 当事者や利用者自らが、じぶんたちのパターンを作り、共通言語として利用することができる。

アジャイル型開発と親和性が高い。



James O. Coplien 2010 "From Patters: Eastward to Lean, Westward to true object"

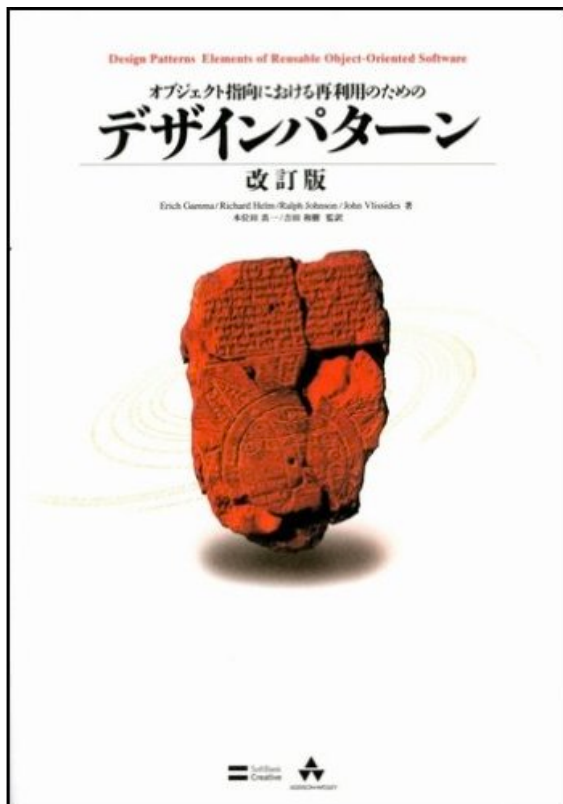
August 27, 2013

Mayanari Motohashi, CultureWorks, LLC

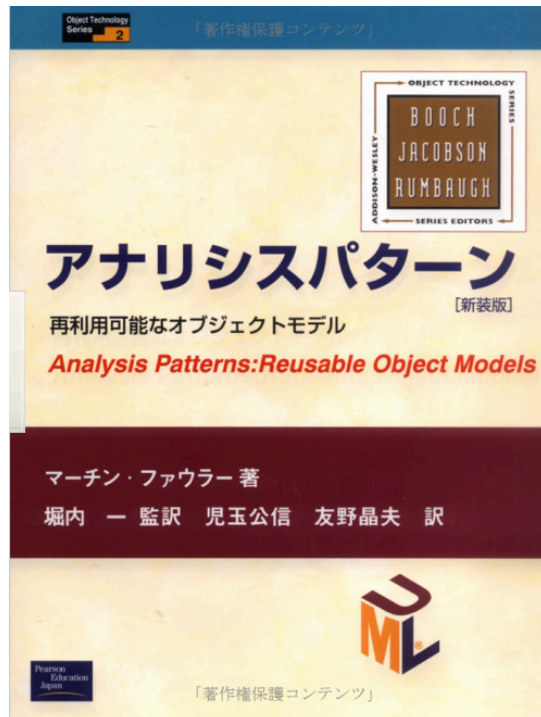
<http://www.infoq.com/jp/articles/AlexanderFestaReport>

40

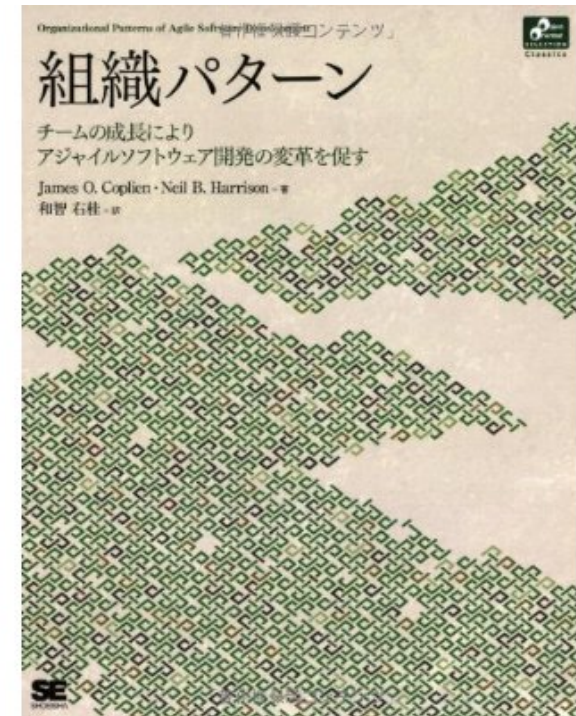
ITソフトウェア分野での応用



August 27, 2013



Masanari Motohashi, CultureWorks, LLC



書影はamazon.co.jpより

41

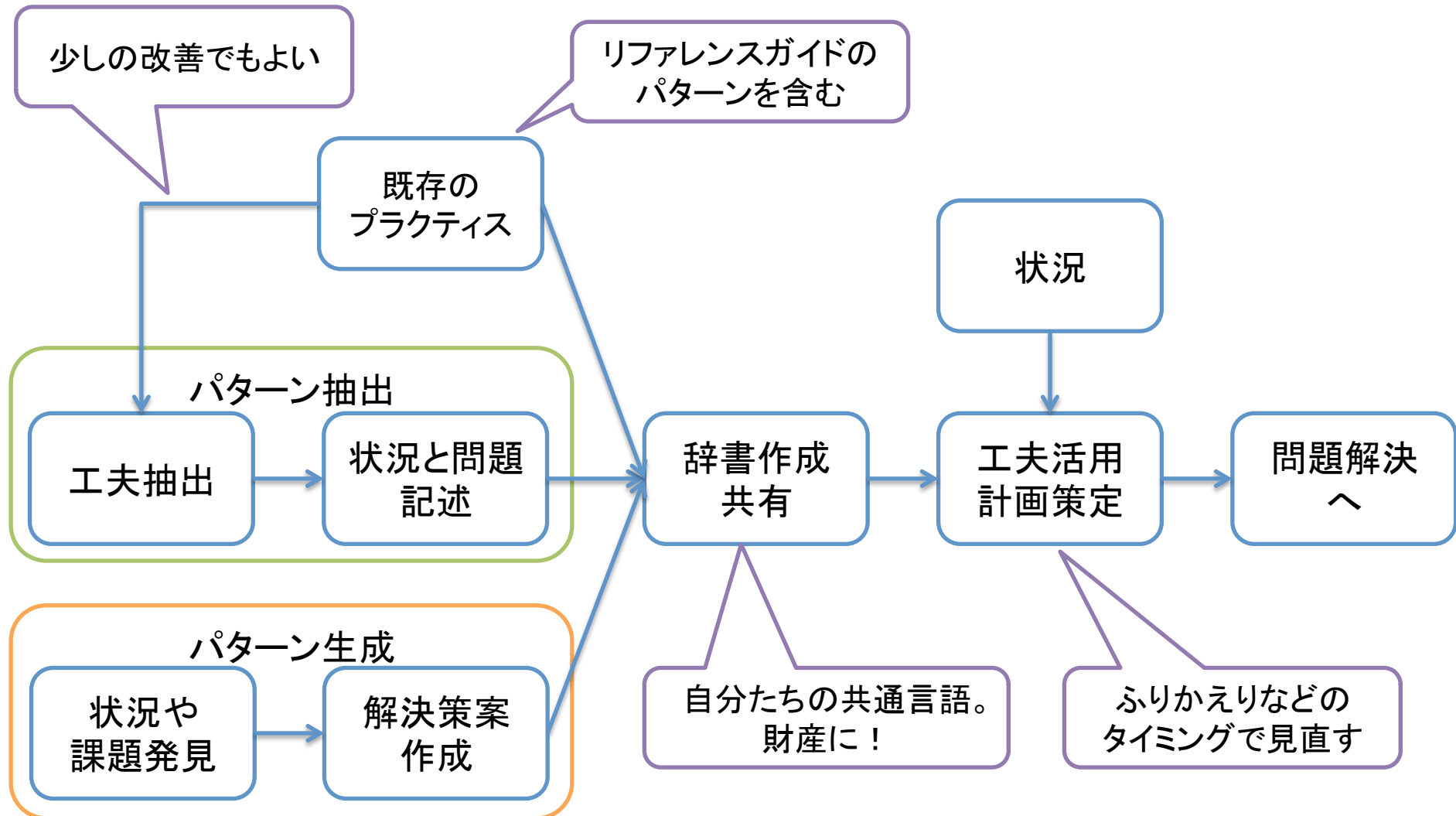
複数の解決策を 組み合わせることができる。

短納期、開発期間が短い場合：

活用するプラクティスは、ベロシティ計測と、バーンダウンチャートが挙げられる。ベロシティについては、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。また、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

パターンを単語として、パターンランゲージやプロジェクトランゲージを生成する。

自分たちの言葉と未来を 自分たちでつくる



参考資料

- 「アジャイル型開発におけるプラクティス活用事例調査」の報告書とリファレンスガイド
<http://www.ipa.go.jp/sec/reports/20130319.html>
- パタンの書き方
<http://www.cultureworks.jp/blog/?p=44>
- パターンライティングワークショップ Agile Japan 2013
<http://www.agilejapan.org/2013/program.html#06>
- ソフトウェアパターン-時を超えるソフトウェアの道-:4.これからのみんなのことば, みんなのかたち-パターンランゲージからプロジェクトランゲージへ
https://ipsj.ixsq.nii.ac.jp/ej/index.php?active_action=repository_view_main_item_detail&item_id=75809&item_no=1&page_id=13&block_id=8

どうもありがとうございました

- まとめ
 - IPA「アジャイル型開発におけるプラクティス活用事例調査」リファレンスガイドの概要を説明しました。
 - パターンと、パターンを適用した理由を説明しました。
- 本橋 正成(もとはし まさなり)
 - motohasi@cultureworks.jp
 - facebook.com/motohasi