

アジャイル開発「スクラム」における プロダクトオーナーの「勘所」

- QCD問題を察知するための「メトリクス」と「勘所性」の提言 -

2016/2/26

第1分科会 ソフトウェアプロセス評価・改善

主査: 三浦 邦彦(矢崎総業株式会社)

副主査: 中森 博晃(パナソニック ファクトリーソリューションズ株式会社)

山田 淳(株式会社東芝)

リーダー: 田中 桂三(オムロン株式会社) ※発表者

研究員: 磯貝 竜太(株式会社ソフトフロント)

内山 哲三(ビジネスキューブ・アンド・パートナーズ株式会社)

海野 和由(矢崎総業株式会社)

森内 真人(株式会社インテック)

山本 和紀(株式会社システムソフィア)



研究紹介の前に:「アジャイル開発」とは？

利用者の要求に素早く適応するための、軽量的開発手法群の総称

【アジャイル宣言】

4つの価値と12の原則

4つの価値

アジャイルソフトウェア開発宣言

私たちは、ソフトウェア開発の実践
あるいは実践を手助けをする活動を通じて、
よりよい開発方法を見つけだそうとしている。
この活動を通じて、私たちは以下の価値に至った。

プロセスやツールよりも**個人と対話**を、
包括的なドキュメントよりも**動くソフトウェア**を、
契約交渉よりも**顧客との協調**を、
計画に従うことよりも**変化への対応**を、

価値とする。すなわち、左記のことがらに価値があることを
認めながらも、私たちは右記のことがらにより価値をおく。

引用元 <http://www.agilemanifesto.org/iso/ja/>

12の原則

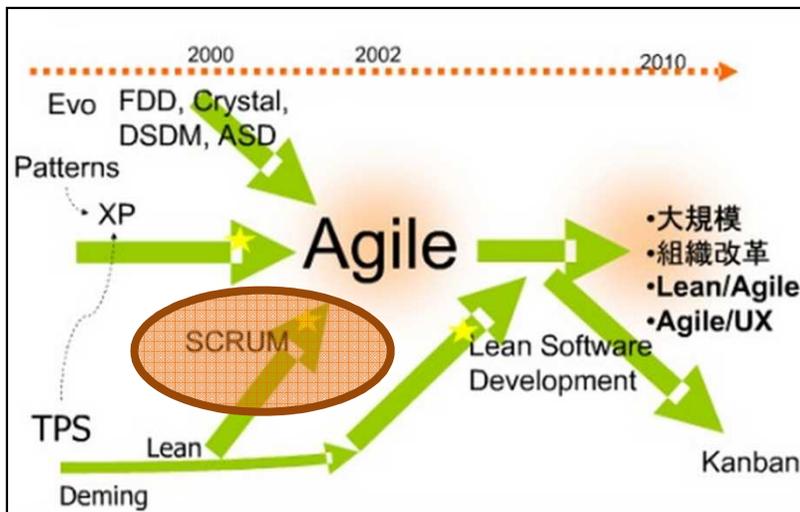
アジャイル宣言の背後にある原則

私たちは以下の原則に従う:

- 顧客満足を最優先し、
価値のあるソフトウェアを早く継続的に提供します。
- 要求の変更はたとえ開発の後期であっても歓迎します。
変化を味方につけることによって、お客様の競争力を引き上げます。
- 動くソフトウェアを、2-3週間から2-3ヶ月という
できるだけ短い時間間隔でリリースします。
- ビジネス側の人と開発者は、プロジェクトを通して
日々一緒に働かなければなりません。
- 意欲に満ちた人々を集めてプロジェクトを構成します。
環境と支援を与え仕事が無事終わるまで彼らを信頼します。
- 情報を伝えるもっとも効率的で効果的な方法は
フェイス・トゥ・フェイスで話をすることです。
- 動くソフトウェアこそが進捗の最も重要な尺度です。
- アジャイル・プロセスは持続可能な開発を促進します。
一定のペースを継続的に維持できるようにしなければなりません。
- 技術的卓越性と優れた設計に対する
不断の注意が機敏さを高めます。
- シンプルさ(ムダなく作れる量を最大限にすること)が本質です。
- 最良のアーキテクチャ・要求・設計は、
自己組織的なチームから生み出されます。
- チームがもっと効率を高めることができるかを定期的に振り返り、
それに基づいて自分たちのやり方を最適に調整します。

研究紹介の前に:「スクラム」とは？

アジャイル開発手法の一つ。
ラグビーのスクラム(SCRUM)に因んで名づけられたとおり、**開発チームが一体となり協力して開発を進めることに重点を置く。**



引用元 <http://www.publickey1.jp/blog/10/xpfes04.jpg>

こんな風に協力して開発する。

■「スクラム」での重要人物

- ・**プロダクトオーナー(PO)**:開発チームの作業とプロダクトの価値の最大化に責任を持つ(**開発責任者**)
- ・**スクラムマスター(SM)**:スクラムの理解と成立に責任を持つ人。

■「スクラム」での重要キーワード

- ・**スプリント**:スクラムでの開発単位。「完成」したインクリメントを作るための、1か月以下のタイム。

※引用元:Ken Schwaber and Jeff Sutherland, スクラムガイド スクラム完全ガイド :ゲームのルール, 2013年7月,

<<http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-JA.pdf>>

※その他の用語は、論文の付録③「スクラム用語集」参照。

本研究のあらすじを、少し柔らかかめに、紹介します。

スクラムでは**プロジェクトの後半にQCD問題が発覚**することがよくある。

スクラムを森に例えると、

森:プロジェクト全体の状況 木:スプリントの状況

問題1:開発責任者のPOが**「木」**の
出来栄を見ていない。

1.技術面
の問題

問題2:**「木を見て森を見ず」**になりがち。

2.管理面
の問題

POの想い

「木を見て、森を見たい！」

～**「木」**の状況を見て、**「森」**の問題を察知したい！～

「勘所」を掴みたい！



有効そうなメトリクスを集めて**「スクラム用PO勘所一覧表」**を作ってみた。

有効かどうか、各社でアンケートをした。その際、**「勘所性」**を定義し**分析**した。



「勘所一覧表」の全メトリクスが**「勘所性あり」**と確認した。

森(プロジェクト)



「スクラム」の問題点

POが、スプリントレビューで、SMのデモンストレーションにより、判断するが、

問題1:「木」自身の出来栄を見ていない。

問題2:「木を見て森を見ず」になりがち。



具体的
には

1. ソフトウェア構造の乱れにより、品質が低下しがちだが、それを確認できていない。
2. スプリント内で発生した問題の全体への影響具合を確認できていない。



結果

プロジェクトの後半にQCD問題として表面化し、後半のスプリントで作業負荷が増加する。

「スクラム用PO勘所一覧表」のポイント

- QCD問題が察知ができるような、「**メトリクス**」を体系立てて抽出
- アジャイル開発の「**迅速・軽量さ**」を守るため、必要最小限に絞る(20個程度)
- POがスプリントレビューで確認しやすいように、メトリクスを集めて、「**一覧表**」にする。



本ポイントを受けて、

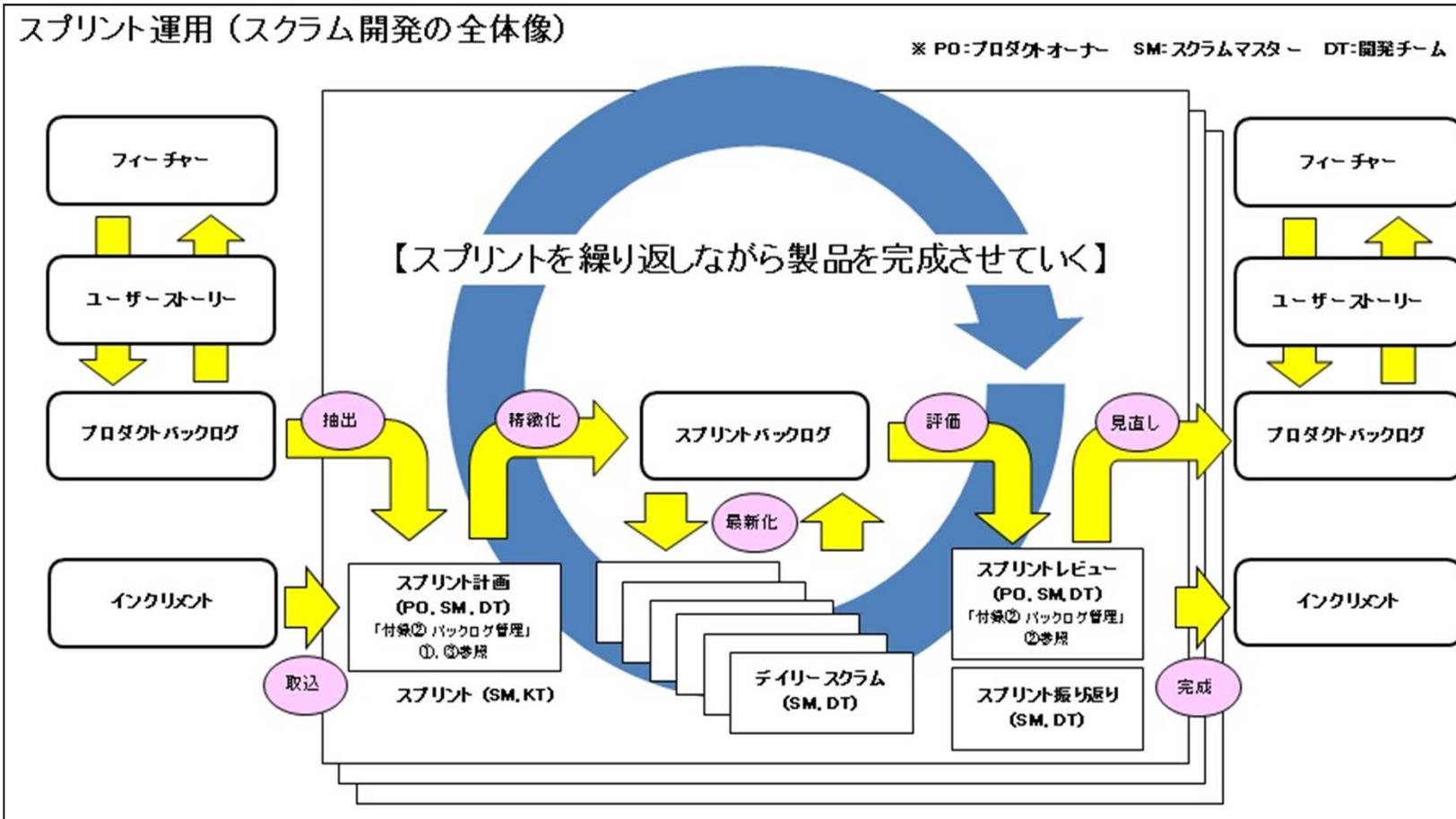
「スクラム用PO勘所一覧表」作成プロセス へ

「スクラム用PO勘所一覧表」作成プロセス

1. スクラム開発の認識共有
「スクラム用PO 勘所一覧表」の作成にあたり、今回の研究で対象とする**スクラムのイメージを整理**。 ⇒ 「スクラム開発の全体像」のページ
2. メトリクス情報の収集
「問題1(技術面の問題)」に対するメトリクス情報は、「**ISO/IEC25010**」の**品質特性**に従い整理
「問題2(管理面の問題)」に対しては、**PMBOKの知識エリア**に従い、プロジェクト管理のメトリクスから抽出。
3. メトリクスの定義
GQM法を元に「メトリクス定義フレームワーク」を作成、これを用いて、収集した情報からメトリクスを定義。 ⇒ 「メトリクス定義フレームワーク」のページ
4. メトリクスの絞り込み
過去の経験より、各スプリントで評価するメトリクス数を20項目以下に絞り込む。
⇒ 「絞り込み後のメトリクス(17項目)」のページ
5. スクラム用PO勘所一覧表の作成
POが手軽に運用できるように、メトリクスの解説も含め「スクラム用PO勘所一覧表」に整理。 ⇒ 「スクラム用PO勘所一覧表」のページ

スクラム開発の全体像

各研究員の認識合わせのため、スクラム開発のイメージを整理



メトリクス定義フレームワーク

「GQM法(Goal, Question, Metrics)」で、収集したメトリクス情報を分析、整理

どのようなスキル、バックボーンのPOでも、理解可能な指標を目指して汎化した

【Goal】

測定を行う目的の定義。測定の対象，理由，観点，およびコンテキストを定めるもの。

カテゴリー

目的

【Question】

目的の達成のために評価すべき質問の定義。
ゴールが達成されたか否かを判断する具体的な基準。

質問項目

【Metric】

前提となるモデルに基づいて測定可能なメトリクス。
上記の質問に対する答えを提供するのに必要な情報を提供する。

導出尺度

基礎尺度

前提条件

判定基準値

判定時期

収集時期

前提条件

入力時期

絞り込み後のメトリクス(17項目)

過去の経験より、各スプリントで評価するメトリクス数を20項目以下に絞り込んだ。

分類	ID	メトリクス名	カテゴリ
技術編	DMs1	要件実装率	機能適合性
	DMs2	仕様変更の発生度	機能適合性
	DMs3	スループット	性能効率性
	DMs4	メモリ利用率	性能効率性
	DMs5	CPU利用率	性能効率性
	DMs6	インタフェース実装率	互換性
	DMs7	テストカバレッジ	信頼性
	DMs8	バグ密度	信頼性
	DMs9	スタビリティインデックス “Si”	保守性
	DMs10	サイクロマティック複雑度 “v (G) ”	保守性
	DMs11	コールする関数数 “CALLING”	保守性
	DMs12	コールされる関数数 “CALLS”	保守性
管理編	DMp1	生産性(ペロシティー)	タイム・マネジメント
	DMp2	総ストーリーポイント	タイム・マネジメント
	DMp3	プロダクトの完成率	タイム・マネジメント
	DMp4	見積りの乖離	コスト・マネジメント
	DMp5	追加作業の有無	コスト・マネジメント

スクラム用PO勘所一覧表

技術編

問題1⇒「技術編」で解決 問題2⇒「管理編」で解決

ID	メトリクス名	目的	質問項目	前提条件	導出尺度	判定基準値	収集時期	判定時期	基礎尺度
◆機能適合性 - 機能完全性・機能適切性									
DMs1	要件実装率	当該スプリントで実装すべき要件が、全て（欠落なく・正しく）実装されていることの確認。	当該スプリントで実装すべき要件が、全て実装されたか？	当該スプリントで実装すべき要件がスプリントバックログとしてリスト化されていること。	要件実装率 [%] 「当該スプリントに割り当てられたスプリントバックログ数」のうち「当該スプリントで実装完了したスプリントバックログ数」	<100%> 当該スプリント内で実装できなかった要件がある場合は・・・POはプロダクトバックログ及び次のスプリントバックログを修正する	各スプリントバックログの実装完了時	スプリントレビュー時	BMs1
DMs2	仕様変更の発生度	当該機能の要件が曖昧でないことの確認。 (その1) 「必要なユーザー要求を漏れなく捕捉できているか。」	当該スプリントで完了した機能について、要件定義作業以降の仕様変更はどれくらい発生したか？	仕様変更の管理方法が確立していること。 当該機能に対する仕様履歴の管理が正しく運用されていること。 (例・スプリントバックログに関連する仕様IDを設定したタスクを追加)	機能毎の仕様変更の発生件数	<基準値はP J計画時に設定> 基準値を超える仕様変更がある機能については・・・POがその仕様見直しの要否を決定し、プロダクトバックログ及び次のスプリントバックログを修正する	各スプリントバックログの実装完了時	スプリントレビュー時	BMs2

ID	メトリクス名	目的	質問項目	前提条件	導出尺度	判定基準値	収集時期	判定時期	基礎尺度
◆性能効率性 - 時間効率性・資源利用率									
DMs3	スループット	当該機能の処理能力が、非機能要件を満たしていることの確認。 (単位時間当たりの処理量)	当該機能は単位時間あたり、どれくらいのタスク（作業）を処理できるか？	評価対象機能毎に期待する目標値（目標レベル）が明らかになっていること。	スループット・目標レベル スループットの測定値の平均をもとに、スループットの余裕率を6段階のレベルで整理する	<目標レベルはP J計画時に設定> 目標レベルに満たない機能については・・・POがその性能改善の要否を決定し、プロダクトバックログ及び次のスプリントバックログを修正する	対象機能の機能テスト完了時	スプリントレビュー時	BMs7

管理編

ID	メトリクス名	目的	質問項目	前提条件	導出尺度	判定基準値	収集時期	判定時期	基礎尺度		
◆プロジェクト・タイム・マネジメント (スケジュール・コントロール)											
DMp1	生産性 (ベロシティー)	計画時に設定した生産性が出ているか確認する。	当該スプリントで予測していたベロシティーが出ているか？ (質問事項詳細①)	・ベロシティーの予測 (計画) があること。 ・スプリントバーンダウンチャートを、日々、記入していること。	ベロシティーの予実差： スプリントバーンダウンチャートの傾きの予定と実績の差異	<許容範囲はP J計画時に設定する> 予実差異が許容範囲を超えた場合は・・・ベロシティーの基準を見直し、次回以降のスプリントに割り当てるスプリントバックログの量を調整する。	各スプリントバックログの実装完了時	スプリントレビュー時	BMp3		
DMs4	メモリ利用率	当該機能が要件を満たしているか確認する。	総ストーリーポイント	計画時に設定した開発ボリュームが大きく変わっていないかを確認する。	プロダクトバーンダウンチャートの高さ、計画との差異はないか？ (質問事項詳細②)	・当該スプリントに対する要件の追加/削除/変更が発生した場合、その内容をタイムリーにスプリントバックログに反映する運用が確立されている事。（ストーリーポイントの変更等） ・スプリントバックログの変更をプロダクトバックログ（プロダクトバーンダウンチャート）に反映する運用が確立されている事。	総ストーリーポイントの予実差： 「プロダクトバーンダウンチャート（縦軸はストーリーポイント）の高さ」の予定と現時点の最終見通しとの差	<総ストーリーポイントの予実差が、計画時にバックアップとして見込んでいるストーリーポイントより小さいこと> 総ストーリーポイントの予実差が、許容範囲を超えている場合は・・・スコープの調整をする。 (当該プロジェクトで対応しないプロダクトバックログのユーザーストーリーを選択)	各スプリントバックログの実装完了時	スプリントレビュー時	BMp1, BMp2, BMp3, BMp4
DMs5	CPU利用率	当該機能が要件を満たしているか確認する。	プロダクトの完成率	プロダクトの完成状況を定量的に把握する。	プロダクト全体の機能のうち、どれくらいが完了したか？ 計画との差異はないか？ (質問事項詳細③)	・SMがスプリント計画時にスプリントバックログとプロダクトバックログをリンクしておく。 プロダクトの完成率 [%] 「完成したプロダクトバックログの機能数」/「プロダクトバックログの全機能数」	<各スプリント終了時点での完成率の計画値> 当該スプリントまでの完成率の計画値を達成できなかった場合は・・・POはスコープの変更を検討する	各スプリントバックログの実装完了時	スプリントレビュー時	BMp4	

ID	メトリクス名	目的	質問項目	前提条件	導出尺度	判定基準値	収集時期	判定時期	基礎尺度
◆プロジェクト・コスト・マネジメント (コスト・コントロール)									
DMp4	見積りの乖離	計画時に行った見積りが大きくふれていないか、その要素が発生していないかを確認する。	当該スプリントでストーリーポイントが大きく変わったスプリントバックログはあるか？ (質問事項詳細④)	・スプリントバックログの追加、変更、削除が発生した場合、（ストーリーポイントを含め）その内容をタイムリーに反映する運用が確立されている事。	スプリントバックログのストーリーポイントの予実の差	<許容範囲はP J計画時に設定する> ・許容範囲を超えるスプリントバックログがあった場合・・・そのストーリーポイントの増減理由が明確である、他のプロダクトバックログを洗い出し、ストーリーポイントを見直す。	各スプリントバックログの実装完了時	スプリントレビュー時	BMp1, BMp2, BMp3, BMp4
DMp5	追加作業の有無	計画時に洗い出した作業に漏れが無い、その要素が発生していないかを確認する。	当該スプリントの実施途中で追加した（新たに発生した）スプリントバックログはあるか？ (質問事項詳細⑤)	・スプリントバックログの追加、変更、削除が発生した場合、（ストーリーポイントを含め）その内容をタイムリーに反映する運用が確立されている事。	追加したスプリントバックログ	<追加作業なし> ・追加したスプリントバックログがあった場合・・・その作業が今後も発生する可能性があるならば、当該作業を今後のプロダクトバックログに追加するか、または新たな作業の発生に備えダミー・プロダクトバックログのストーリーポイントを増加する。	各スプリントバックログの実装完了時	スプリントレビュー時	BMp5

本一覧表は、論文の付録⑦、付録⑧に掲載。

効果検証手法(アンケート実施)

【目的】

定義した**メトリクス**が、
スクラムにおける**QCD問題**
の予兆を察知できる
「勘所となるか？」
を判断する

- ***勘所性**:本研究で作成した、**オリジナル**の判断基準
メトリクスがQCD問題の予兆を察知できる勘所となるか否か？
を判断する。
- ・「勘所性あり」⇒ 勘所となる
 - ・「勘所性なし」⇒ 勘所とならない

【検証手順】

アンケート作成



アンケート実施



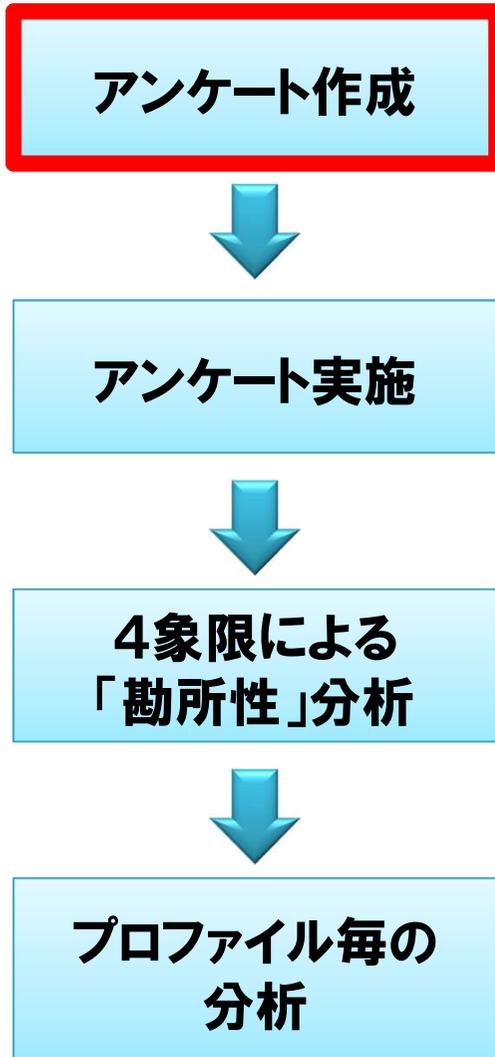
4象限による
「勘所性*」分析



プロフィール毎の
分析

アンケート作成

【検証手順】



【各メトリクスに対する評価】

観点	評価	選択肢
有効性	低い評価	1. 全く有益でない
		2. あまり有益でない
	高い評価	3. ある程度有益である
		4. とても有益である
実現可能性	低い評価	1. 実現不可能
		2. 実現が困難
	高い評価	3. 実現できる
		4. 容易に実現できる

評価を明確にするため中間値を無くし4択

【アンケート対象者のプロフィール(一般質問)】

No.	質問項目	回答選択肢
1	現在の役職	担当者, リーダー職, 管理職, 役員, その他
2	開発経験	5年未満, 5年以上10年未満, 10年以上15年未満, 15年以上
3	管理経験	5年未満, 5年以上10年未満, 10年以上15年未満, 15年以上
4	開発分野	エンタープライズ, 組込み系
5	アジャイル開発	開発している, 開発していない
6	役割	PO, SM, DT, その他

アンケート実施

【検証手順】

アンケート作成



アンケート実施



4象限による
「勘所性」分析



プロフィール毎の
分析

【アンケート対象者】

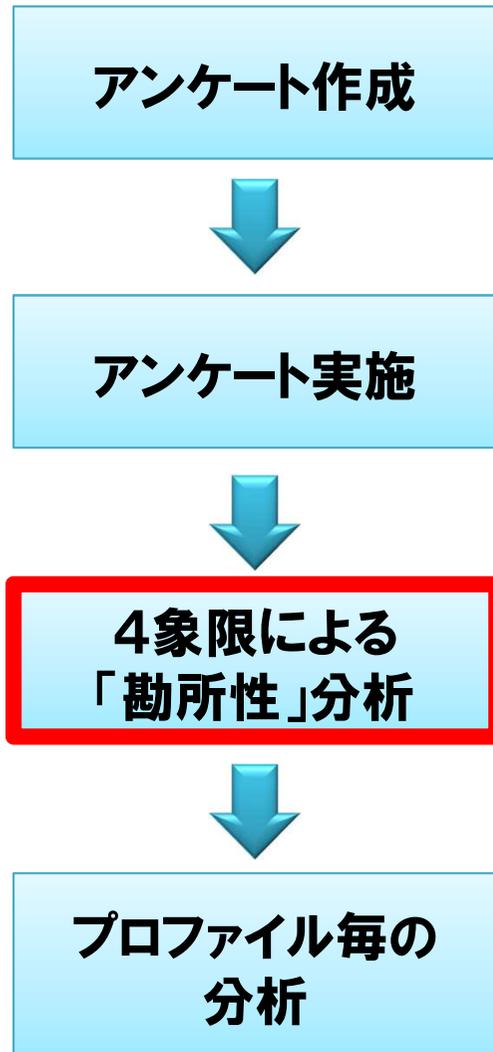
研究員所属各社のソフトウェア関係者で、以下の何れかを満たしている人

- ・スクラムを実践している人
- ・スクラムを理解している人

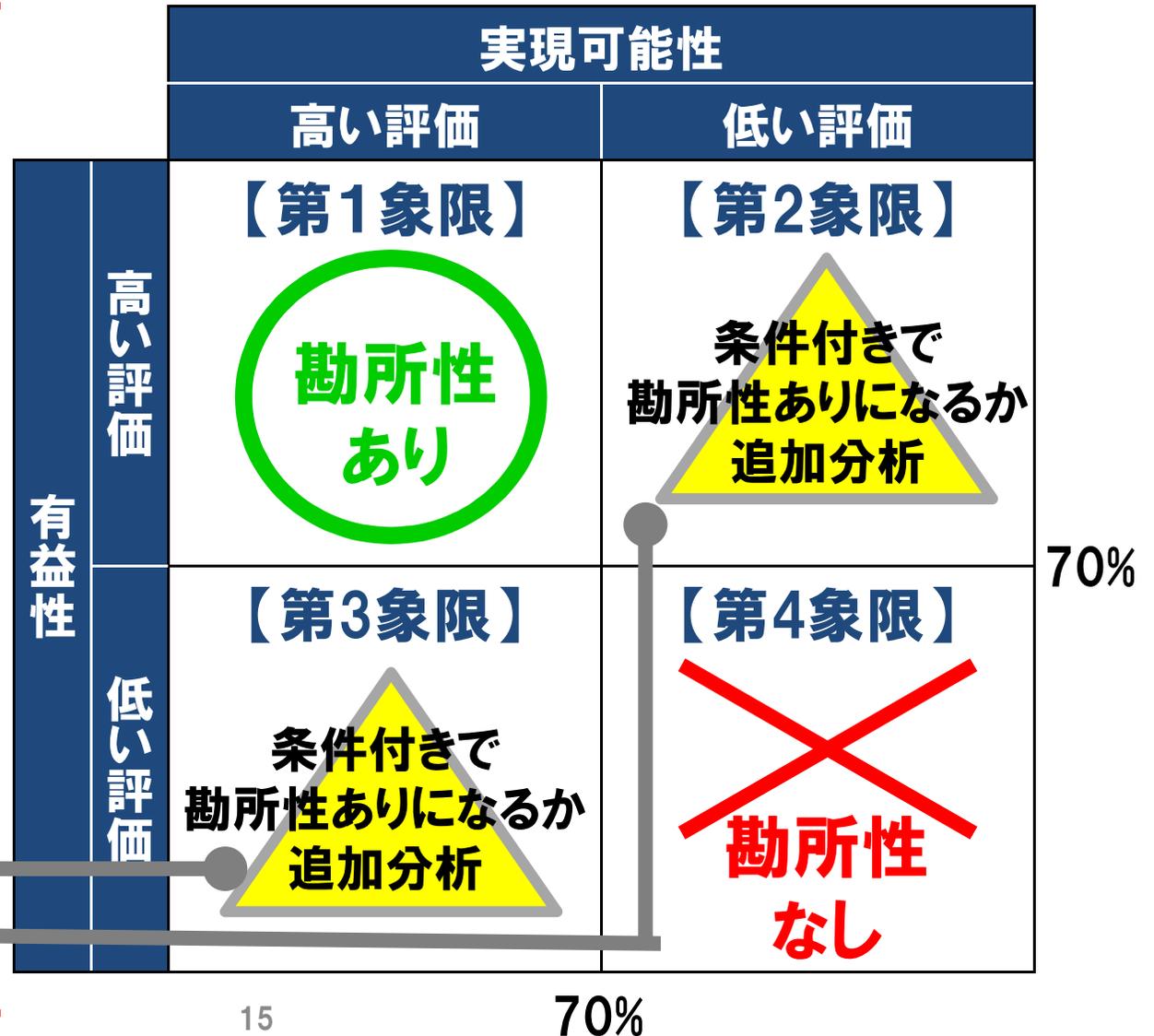
(対象数:計7社31名)

分析アプローチ(4象限による「**勘所性**」の判断)

【検証手順】



【各メトリクスの「**勘所性**」の判断】



分析アプローチ(プロフィール毎の分析)

【検証手順】

アンケート作成



アンケート実施



4象限による
「勘所性」分析



プロフィール毎の
分析

【分析のアプローチ】

第2、第3象限に対して、

1. プロフィールとの相関分析

ピアソンのカイ2乗検定
(P 値 ≤ 0.5)

2. ヒストグラムでの傾向分析

有益性、実現可能性ともに「高い
評価」のプロフィールを特定

3. アンケート対象者に追加ヒアリング

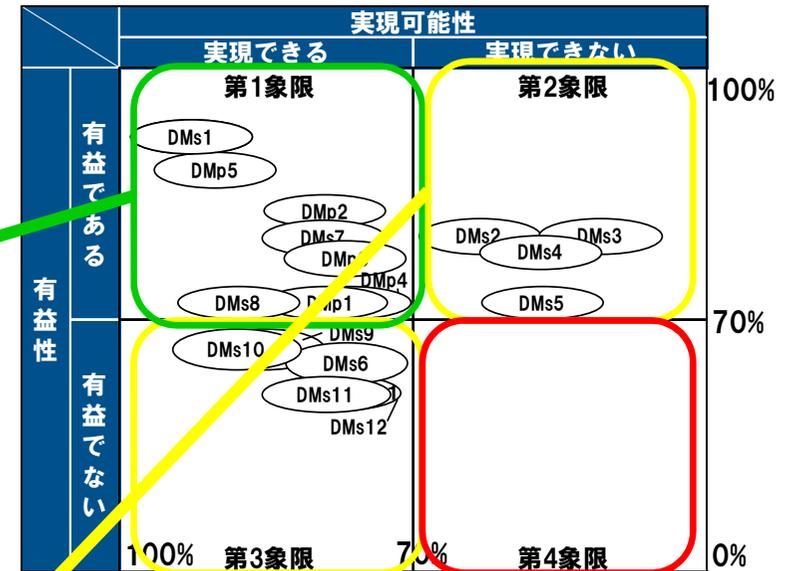
「低い評価」の回答理由を分析

分析結果～各メトリクスの「勘所性」評価結果～



17項目中8項目は「勘所性あり」。9項目は追加調査へ。「勘所性なし」は0項目。

各メトリクスのマッピング



勘所性なし=0!

ID	メトリクス名	カテゴリ	有益性	実現可能性
DMs1	要件実装率	機能適合性	93.55%	100.00%
DMs7	テストカバレッジ	信頼性	80.65%	74.19%
DMs8	バグ密度	信頼性	70.97%	80.65%
DMp1	生産性（ペロシティー）	タイムマネジメント	70.97%	74.19%
DMp2	総ストーリーポイント	タイムマネジメント	83.87%	74.19%
DMp3	プロダクトの完成率	タイムマネジメント	74.19%	70.97%
DMp4	見積りの乖離	コストマネジメント	70.97%	70.97%
DMp5	追加作業の有無	コストマネジメント	90.32%	90.32%

勘所性あり!

ID	メトリクス名	カテゴリ	有益性	実現可能性
DMs2	仕様変更の発生度	機能適合性	80.65%	64.52%
DMs3	スループット	性能効率性	80.00%	43.33%
DMs4	メモリ利用率	性能効率性	77.42%	58.06%
DMs5	CPU利用率	性能効率性	70.97%	45.16%

実現性?



WHY??
「プロファイル毎の分析」実施

ID	メトリクス名	カテゴリ	有益性	実現可能性
DMs6	インタフェース実装率	互換性	61.29%	70.97%
DMs9	スタビリティインデックス "Si"	信頼性	63.33%	78.57%
DMs10	サイクロマティック複雑度 "v(G)"	保守性	63.33%	82.14%
DMs11	コールする関数数 "CALLING"	保守性	53.33%	75.00%
DMs12	コールされる関数数 "CALLS"	保守性	53.33%	75.00%

有益性?



分析結果 ～第2象限、第3象限の追加調査～



1. プロファイルとの相関分析

2. ヒストグラムでの傾向分析

3. アンケート対象者へのヒアリング

ID	メトリクス名	分析対象	1. 相関分析	2. 傾向分析	3. ヒアリング
DMs2	仕様変更の発生度	実現可能性	—	現場は「実現可」	前提条件を詳細化すれば「勘所性あり」につながる
DMs3	スループット	実現可能性	組込み系は「実現可」	—	
DMs4	メモリ利用率	実現可能性	組込み系は「実現可」	組込み系or管理経験5年は「実現可」	
DMs5	CPU利用率	実現可能性	—	—	前提条件と尺度を詳細化すれば「勘所性あり」につながる
DMs6	インタフェース実装率	有益性	—	現場は「有益だ」	尺度を詳細化すれば「勘所性あり」につながる
DMs9	スタビリティインデックス "Si"	有益性	—	組込み系or現場は「有益だ」	・尺度を詳細化すれば「勘所性あり」につながる ・「機能満たしていればいいじゃん！レビューしてるし。」
DMs10	サイクロマティック複雑度 "v(G)"	有益性	—	組込み系or現場は「有益だ」	
DMs11	コールする関数数 "CALLING"	有益性	アジャイラーは「有益だ」	—	⇒保守性の意識づけ、リファクタリング作業をプロセス化するなどの対策が必要
DMs12	コールされる関数数 "CALLS"	有益性	—	—	

分析まとめ

アンケート分析結果、

『一部条件付きだが、全メトリクスが「勘所性あり」!』

- ・17項目中8項目は「勘所性あり」
- ・9項目は条件付きで「勘所性あり」
- ・「勘所性なし」は、0項目

⇒全メトリクスが、有効な指標になり得る



「スクラム用PO勘所一覧表」が、
『「木」の状況を見て、「森」の問題を察知できる*』
(「勘所性」につながるメトリクスを創出できた)



*森:プロジェクト全体の状況 木:スプリントの状況



プロダクトオーナー(PO)の「勘所」として活用しよう!

課題と今後の展望

目標は達成したが、課題有り。
 本研究成果を育て、**実運用で効果**につなげていきたい！



課題	説明	今後の展望
アジャイル、スクラム独自のメトリクス	「 迅速・軽量 」につながるメトリクスは未抽出	左記観点でメトリクスを抽出 (例:スプリント内で変更に対応できるコードとは何か?)
実運用での各メトリクスの勘所性検証	各メトリクスを、 実運用 で使っていない。	実際の開発で使ってみて、勘所性を評価
異なるスタイルの適応	例えば、 ハイブリッドアジャイル では、各メトリクスの勘所性の評価が変わる?	異なるスタイルの開発プロジェクトで実運用、分析

参考文献

- [1] Ken Schwaber and Jeff Sutherland, スクラムガイド スクラム完全ガイド :ゲームのルール, 2013年7月,
<<http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-JA.pdf>>
- [2] 株式会社アイネット, ユーザー・ストーリー, <<http://www.e-ainet.com/UserStory.html>>
- [3] 貝瀬岳志・原田勝信・和島史典・栗林健太郎・柴田博志・家永英治, スクラム実践入門 成果を生み出すアジャイルな開発プロセス, 技術評論社, 2015
- [4] Janet Gregory・Lisa Crispin他, 実践アジャイルテスト テスターとアジャイルチームのための実践ガイド, 翔泳社, 2009
- [5] 長瀬嘉秀・英繁雄・奈加健次・平岡嗣晃・前川祐介, ハイブリッドアジャイルの実践, リックテレコム, 2013
- [6] 鷺崎弘宜, メトリクスによるプロダクトの品質把握と改善 - Goal-Question-Metric (GQM) 法のコツ, および, 組織目標との整合, 早稲田大学グローバルソフトウェアエンジニアリング研究所, 2013
- [7] Project Management Institute, プロジェクトマネジメント知識体系ガイド第5版, 2013
- [8] 日本工業規格, JIS X25010 システム及びソフトウェア製品の品質要求及び評価 (SQuaRE) - システム及びソフトウェア品質モデル, 2013
(ISO/IEC 25010 Systems and Software Engineering Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models, 2011) 日本工業標準調査会,
<<http://www.jisc.go.jp/app/JPS/JPS00020.html>>
- [9] 小池利和, ソフトウェアメトリクス統計分析入門, 日科技連出版社, 2015
- [10] NECマネジメントパートナー, 相関分析 <https://www.neclearning.jp/sample_text/DB101-1.pdf>
- [11] 経済産業省, 情報システム/ソフトウェアの品質メトリクスセット, 2011,
<http://www.meti.go.jp/policy/mono_info_service/joho/cloud/2011/11_03.pdf>
- [12] Kuder Helmar, HIS Source Code Metrics, HIS (Herstellerinitiative Software), 2008,
<http://portal.automotive-his.de/images/pdf/SoftwareTest/his-sc-metriken.1.3.1_e.pdf>
- [13] Scott W. Ambler・Mark Lines, ディシプリンド・アジャイル・デリバリー エンタープライズ・アジャイル実践ガイド, 翔泳社, 2013

**以上で発表を終わります。
ご清聴、ありがとうございました！**

