研究コース４（ヒゲメガネグループ）

アジャイル開発におけるスプリントプランニングの正当性検証と問題検出の早期化に向けた研究

Validation and early detection to defeat of splint planning in Agile development

研 究 員：阿部　健太郎（アンリツエンジニアリング株式会社）
　　中村　賢二（株式会社インテック）
主　査　：永田　敦（サイボウズ株式会社）
副 主 査：山口　鉄平（ヤフー株式会社/一般社団法人アジャイルチームを支える会）
アドバイザー：細谷　泰夫（三菱電機株式会社）

研究概要

スクラムにおけるプロダクトバックログアイテム（PBI）は「何を」実現するかが示される．スプリント計画（SP）時に「どのように」実現するかを考え，実際のタスクに分割し開発に着手する．スプリントの途中で考慮漏れのタスクが発覚した場合には，スプリントの PBI を見直すなど，アジャイル開発は柔軟な対応が可能である．しかし，スプリントの後期に大規模な考慮漏れの問題が発覚するとスプリントゴールの達成が困難になり，計画通りにリリースできないという大きな問題に発展することもある．

本研究では，スプリント初期の段階で実際の開発に先立ち，各 PBI の調査およびレビューをし，SP の正当性を確保，あるいは SP 時に認識できなかった考慮漏れの問題を早期に検知できる手法を提案する．実験の結果，提案手法は不確実な PBI を含んだスプリントでも，スプリント初期の段階で全容を把握できることが確認できた．

1.　はじめに

1.1　プロジェクト計画

ソフトウェア開発において,計画がプロジェクトの成功を左右するといっても過言ではない．しかし，実際の開発現場では，計画の段階で要求や仕様が不明確であったり，それを実現するための手段が技術的に開発チーム内で確立できていないことがある．この場合，開発と並行して技術調査を行わなければならない．このような開発状況では，タスクの考慮漏れが発生しやすく，それが発生することによって残業の増加や計画の見直しをしなければならない，という事態が起きる．

1.2　アジャイル開発における計画と問題

アジャイル開発の一つであるスクラム[1]においては，特に不確実な技術に対するリスクを未然に考慮して，SP の段階において，調査タスクとして切り出すことでバッファを設ける対策が取られている[2]．また，スプリントバックログ（SB）はスプリントゴールを達成するために選択した PBI と計画を合わせたものであり，スプリントのインクリメントに含まれる機能と，その完成に必要な作業について，開発チームが予想したものである．インクリメントに必要な新しい作業があれば，開発チームによって SB に追加される．不要となった場合には削除も行う．これは，開発チームが SP を実行する中で，スプリントゴールに必要な作業を学習することに起因する．このように，スクラムは従来のウォーターフォール型の開発よりも柔軟に対応することができる．

しかし，実際の開発ではプレスリリースした機能や顧客と取り決めたマイルストーンがあるなど，明確な納期が存在することがある．このような状況でスプリントの初期に考慮漏れのタスクが見つかった場合であれは，そのスプリントに当該スプリントバッグログア

イテム（SBI）を含めるかどうかなど柔軟に対応ができるが，考慮漏れのタスクがスプリントの後期に取り掛かったタスクで見つかった場合や，考慮漏れのタスクが芋づる式に見つかった場合には，スプリントゴールの達成に間に合わないということが起こり，プロダクトオーナー（PO）が目論んだ価値の提供がオンタイムにデリバリーできなくなる．あるいは，スケジュールに合わせるために残業がかさむような無理な開発をチームは強いられることになり，提供するプロダクトの品質が落ちてしまう可能性が増えるため問題である．

本研究では，SP の段階で明らかになっていない不明確なタスクへの対応に問題があるのではないかと仮定した．その問題を分析すると，次の３段階に分けられると考えた．

(1) PBI の作業内容・規模が不明確【前提】

(2) 作業内容・規模が判明するのが遅れている【事実＝着眼点】

(3) 問題発覚が遅れたことにより対策が打てない【結果＝問題】

この３点の中で，我々の研究グループでは２点目の「作業内容・規模が判明するのが遅れている」という現状・事実の部分を改善することにより，３点目にある問題が発生しなくなることを期待する．

1.3 不明確なタスクへの対応

我々は本稿で，この具体的な対策としてスプリントの初期に，当該スプリントのスコープに入るすべての SBI について，次のステップで作業を実施することを提案する．

(1) SBI としてタスクに分割した作業内容に対して，具体的な作業予定をまとめる．

(2) この作業予定について実際の作業に入る前に開発チームによるレビューを実施する．

(3) レビューが済んだものから実際の作業に着手する．

さらに，(1)の作業予定については，障害対応や直前のリリースに対する緊急の修正などがない限りは，スプリントの始めに実施する．これにより，スプリントの初期段階で，SP の正当性が確保され，仮に予測に誤りがあった場合でも，スプリントの初期に対策が打てるため，本来アジャイル開発で備えていた柔軟性を享受できることを期待する．

2. 背景(問題提起)

2.1 柔軟性では補えない場合

アジャイル開発には 1.2 章で述べたような柔軟性があるが，見込んでいたよりもタスクの解決に時間がかかることがある．あるいは，開発チームが見えていなかったタスクがスプリント実行中に発覚することがある．これは PBI の段階では「どのように」ではなく「何を」として記述されているので，SP の段階で PBI からタスクに起こす際，具体的に「どのように」対処するかをタスクの担当者が考える必要があるからと考えられる．

このような事態を防止するためには，開発チームが PBI やプロジェクト，およびそのプロダクトに対して十分に理解しており,また SP を実施する前にリファインメントなどを行って PBI を事前に整理し,タスク化する際のイメージを明確化しておくことが必要である．またそのためには PO との連携も十分に取れているなど,チームとして自己組織化されている姿が理想像である.しかし,それらが出来ていない未成熟なスクラムチームの場合には，SP の段階でタスクの見通しが悪く，不明確なタスクも見込みとしてスケジュールに組み込まれることが多くなる．この不明確なタスクによって認識できていなかったタスクが芋づる式に発覚する事態にも展開することもある．さらにこのような認識できていなかったタスクはスプリントの後期になってスケジュールにインパクトを与える重大な考慮漏れになることもあり得る．発覚が遅れればスプリント計画通りに成果物を上げることが難しく，残業で対応したり，リリースのスコープを変更せざるを得なくなる．

2.2 スプリント計画について

このような問題に対する打開策としてまず挙げられることは，スプリント計画の段階で十分な調査を行い全てが決まってから，実際のスプリントに着手するということである．しかしながら，SP は実際に使えるタイムボックスは短く，30 日のスプリント計画でも最大

で８時間と言われており，もし，より小さなスプリントにすれば同じ比率で短くなるべきであると言われている[3]．実際の開発現場において，限られたタイムボックスの中で不明確なタスク内容を明らかにするための十分な調査を行うことは，先に述べたような未成熟なスクラムチームであるほど困難であることが想像に難くない．

2.3 未成熟なスクラムチームとは

本章で示した問題は，スクラムチームの成熟度も影響しており，未成熟なスクラムチームであるほど顕著に表れると考えられる．本稿で表現する未成熟なスクラムチームとは，以下のような特徴を持つスクラムチームを指す．

＜未成熟なスクラムチームの特徴＞

- アジャイル開発に不慣れなメンバーが多数いる．
- リファインメントで PBI の整理ができていないままスプリントに入れてしまう．
- PO との連携が不十分．
- 開発チームのベロシティが不安定．
- 計画の見通しが曖昧．

3. 提案

2 章で述べた問題に対する我々の対策として，不明確なタスク内容を明らかにする工程をスプリントの開発作業初期に実施する手法を提案する．これによって，スプリント計画時点では不明確だったタスクの規模や，タスク漏れをスプリントの初期に明らかにし，計画の是正が必要な場合に早期の対策を打てるようにする．ただしこの提案を実施するために，従来行ってきたタスク作業とは別に，追加の作業を発生させてしまうと開発チームのベロシティを下げる要因になり兼ねない．よって本研究では，これらのタスク作業を細分化し分割して開発する方法と，スプリントにおけるタスクの処理順序について着目した．

3.1 タスクの分割

１つのタスクを次の 3 つのタスクグループに分割する．まず，作業予定，作業内容を明らかにするための「事前調査タスク」，次に，事前調査タスクに対してレビューを行う「事前調査レビュータスク」，最後に，明らかにした作業内容に従って作業を実施する「作業タスク」である．これらタスクグループの処理順序は次のようにする．(1) と (2) の具体的な内容については 3.3 章に後述する．

(1) 作業予定，作業内容を明らかにするための事前調査タスク
(2) (1) に対してレビューを行う事前調査レビュータスク
(3) 実際の作業タスク

従来法では未成熟なスクラムチームであった場合，スクラムチームもしくはタスクの担当者が SP 時に必要なタスクを見落としていた場合には,スプリントの各 PBI のタスクに着手した後に問題が初めて認知され，2.1 章で述べたような問題に発展する．本提案手法では，タスクの担当者が無意識的に行っていた事前調査のタスクと，実際の作業に着手する前でも確認できる作業自体の完全性や正当性といった観点によるレビューのタスクを，実際の作業タスクより前に実施するという点が従来法からの相違である.これにより SP 時に見えていなかった問題が潜在していた場合でも,「事前調査タスク」あるいは「事前調査レビュータスク」でタスクの見落としが発見でき，2.1 章で述べたような問題が起きることを未然に防げると期待する．分割した各タスクの処理順序の詳細は次項で論じる．

3.2 タスクの処理順序

次にスプリント全体でのタスクの処理順序については,3.1 章で行ったタスク分割で (1) に該当するすべてのタスクをスプリントの初期に行うこととする．従来法と比較すると，次の図 3.2-1 のようになる．

従来法ではスプリント中はPBI1を完了してからPBI2に着手，PBI2を完了してからPBI3に着手というように，PBIの優先順位でタスクを実施していく．PBI1>PBI2>PBI3と表記する．提案手法ではn番目のPBIであるPBInを，調査およびそのレビュー（PBIn’）と作業（PBIn’’)に分けて，SBのすべての調査とそのレビューを実際の作業に先行して実施する．
PBI1’>PBI2’>...>PBI1’’>PBI2’’>...
のような順序となる．
これにより，SP時には不明確であったタスクが，スプリントの初期にすべて明確になるため，スプリントの後期になり不測の事態に陥ることを軽減することができると考える．

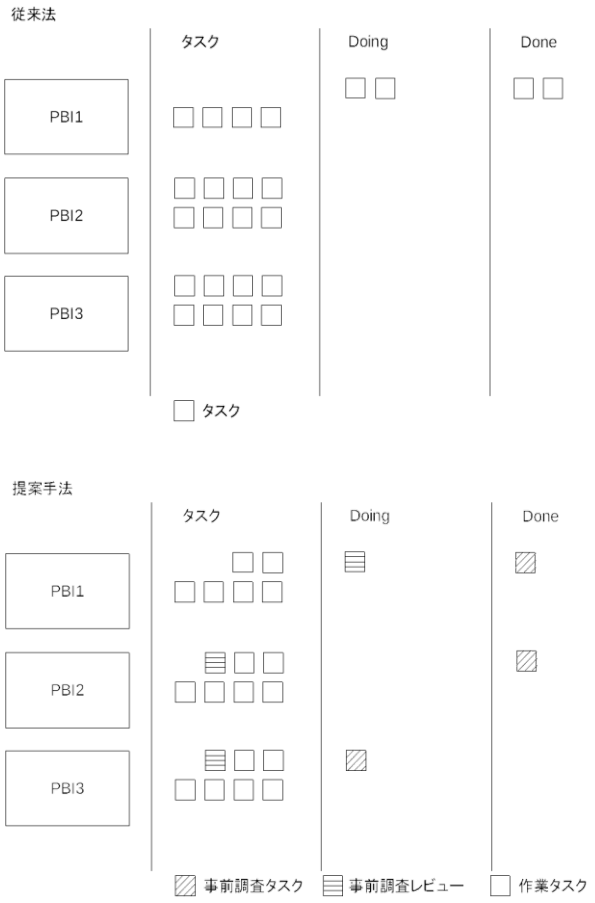

図 3.2-1 タスクの処理イメージ

また留意点としては、本提案はPBIの優先度を無視して各PBIの事前調査タスクのみを先に実施し，レビューを行っている．このためスプリント初期ではPBIの仕掛かりとなってしまう．これはPBIの仕掛かりを作らないように開発がしたいという主張とトレードオフになると考えられる．スプリントの初期段階ではPBIの仕掛かりが増えてしまうが，SPの時点で精度の悪い見積もりをしていたPBIに対してより正確な見通しが得られ，さらに大きな手戻りも未然に防ぐことができるため，スプリント全体でみれば適正なベロシティで開発が行えるということを事前にPOと合意を得なければならないことが考えられる．

3.3 具体的な方法

3.3.1 事前調査タスク

「事前調査タスク」では，対象のPBIをDONEにするために必要な作業内容を具体化し，「いつまでに」「何を」「どうするか」について明文化する．調査結果に応じて，タスク規模の見直しや，タスクを更に細分化して分割することも必要になる場合もある．事前調査タスクでまとめる具体的な内容を表3.3.1-1に示す．このとき事前調査タスクはあくまでも明文化するところまでが作業の内容であり，実際にソースの追加や変更，削除などの開発に着手してはならない．

3.3.2 事前調査レビュータスク

次に行う「事前調査レビュータスク」では「PBIをDONEにするために必要なタスクは洗い出せているか」（網羅性）と「各タスクの作業内容（実現手段）とDONEの定義が明確になっているか」（透明性）の２つの観点のみに絞ったレビューを行う．このレビューでの指摘により，漏れていたタスクの追加や，実現手段の再検討が必要になる場合もある．この時，予測していた作業規模を明らかに超えてしまうことが発覚した場合は，対象のレビュー指摘に対するフォローアップを一時中断して，3.3.3章に後述するSPの見直しを先行して実施するべきである．

尚，「実際の作業タスク」の成果物に対するレビューでは，上記の観点以外に主眼を置い

てレビューを行う．例えば内容の正確性や妥当性，整合性の確認などが該当する．

表 3.3.1-1 事前調査タスクの項目

項目	説明
作業タスク名	何の要求，機能，設計に対するものかを示す作業名． ※チケット駆動開発ではチケット名，あるいはメールの件名など，開発チーム内で共有できる表題．
前提条件の有無	タスクを Ready にするために先に Done にしておくべき別のタスクがあるか．
作業タスクの変更，追加内容の概要	どのようにして作業タスク対象を実現するのかの実現手段の概要．
作業タスク対象範囲	変更，追加する対象機能名の一覧，または変更，追加する対象ソースファイル名，クラス名の一覧などの対象範囲．
作業タスクの予定工数	SP 時に見積もった工数を再見積りした工数．

3.3.3 SP の見直し

「事前調査レビュータスク」までが完了した時点で,「事前調査タスク」に関連する PBI がスプリントで実施可能か,そうでないか,の見極めが SP 時点よりも正確になっていると想定されるため，この時点でスプリントにより本当に実施可能な PBI だけとなっているかを見直し，実施が困難になった PBI が存在するかを確認する．実施が困難になった PBI が存在する場合,その旨を PO に提言し,対象の PBI を分割する,代わりの PBI に入れ替える,などの方針を合意してから「作業タスク」に着手するものとする．

4. 実験と評価

4.1 実験対象

実験はスクラム開発を取り入れて間もない未成熟なスクラムチームを対象に，3 章で示した提案手法を適用した．適用の前後で開発進捗の安定性についてバーンダウンチャートを用いて分析する．尚，実際の開発時期は，適用前，適用後の順番で行っている．実験対象としたスクラムチーム構成は以下の通りである．

表 4.1-1 実験対象チーム構成

チーム人数	スクラム開発歴	1 スプリント期間
5 名	4 か月	2 週間

4.2 実験方法の詳細

3 章で示した提案手法を，要件定義，設計，実装，テストの各工程の内，設計と実装の範囲で適用させた．適用させた各工程にて行った事前調査タスクでは，調査に対応する作業タスクに対して表 3.3.1-1 のような内容を，各事前調査タスクの担当者が開発チームへレビューを依頼する．それに対して 3.3 章で示した観点（網羅性と透明性）を用い，開発チーム内でレビューを行った後,実際の作業タスクに着手するという手順で開発を行った．

4.3 実験結果

本稿提案手法の適用後に実施した事前調査レビューで挙がった指摘の原因分析を表 4.3-1 に示す．各指摘の原因は，IEEE 830-1998 の品質特性[4]に当て嵌め，後戻りのリスクに順位付けを行った．品質特性の「完全性」が損なわれるということは、仕様や実装内容に抜け漏れがあることを意味する．また「正当性」が損なわれるということは，仕様や実装内容に誤りがあることを意味する．事前調査レビューを行わなかった場合、後者は誤った実装を修正するだけで済むが、前者は更に実装の調査からやり直さなければならないため、各特性は「完全性」＞「正当性」＞「その他」の順番で後戻りのリスクが高くなると考えられる[5]．

表 4.3-1 事前調査レビュー指摘分類

原因分類		内容説明	後戻りリスク	件数
要件定義漏れ	-	要求仕様書（上位文書）で項目の抜け漏れにより完全性に欠陥がある.	高	0
要件定義誤り	-	要求仕様書（上位文書）で項目の誤りにより正当性に欠陥がある.	中	1
内容不明確	検討不足	検討不足により調査結果の正当性に欠陥がある.	中	5
	曖昧記載	調査結果内容の非曖昧性に欠陥がある.	低	4
スコープ誤り	-	調査結果でピックアップした対象範囲の完全性に欠陥がある.	高	2
考慮漏れ	仕様不足	設計時に必要な仕様の抜け漏れにより完全性に欠陥がある.	高	10
	機能不足	実装時に必要な機能の抜け漏れにより完全性に欠陥がある.	高	4

実際には表 4.3-1 の通り「完全性」に関する指摘が 16 件,「正当性」に関する指摘が 6 件それぞれ挙がった. これにより本稿提案手法の適用後は大きな手戻りが少なくなることが分かった. これは実際にソースの追加や変更, 削除などの開発に着手する前の段階であっても, 事前調査レビューの観点によって多くの後戻りのリスクを検知することができると示せたと言える.

次に, 本稿提案手法の適用前と適用後のバーンダウンチャートを図 4.3-1 と図 4.3-2 に示す. 縦軸はスプリントで実施するタスク残工数 (H), 横軸はスプリントの経過日数(日)を表す. また, 適用前と適用後の計画工数に対する結果を表 4.3-2 に示す. 計画工数とは当該スプリントで開発チームが PO と DONE することを確約した PBI の全タスクの工数の合算値とし, 未消化工数とはスプリント終了時点で未完了だったタスクの工数の合算値を表す.また計画工数消化率を(計画工数 ― 未消化工数)／計画工数 の計算式から算出する.

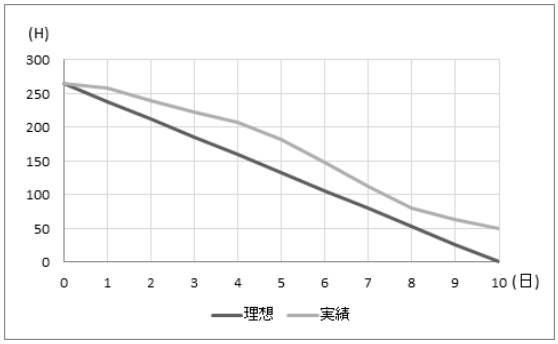

図 4.3-1 適用前バーンダウンチャート

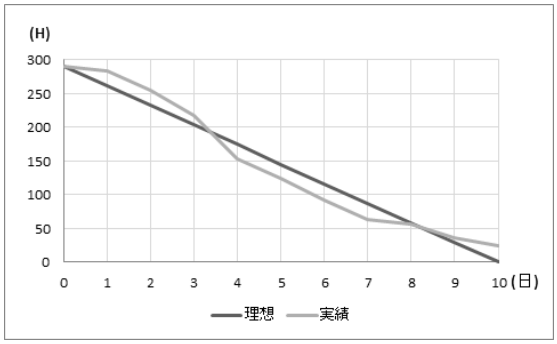

図 4.3-2 適用後バーンダウンチャート

表 4.3-2 適用前と適用後の工数比較

適用前後	計画工数	未消化工数	計画工数消化率
適用前	265(H)	49.25(H)	82(%)
適用後	290.5(H)	24.25(H)	92(%)

バーンダウンチャートを見ると,適用前に比べて適用後は SP の理想線に沿っての開発が進められていることがわかる. これは事前調査と事前調査レビューのそれぞれのタスクに明確なゴールが決められたためタスクそのものを効率的に処理できたと考えられる. さらにスプリントの初期で実装内容が明確となったことで SP の後半でも同様のベロシティが

維持できたと考えられる．これによって計画工数および計画工数消化率についても，適用後は適用前よりも計画工数が多かったにも関わらず未消化工数が少なく，結果として計画工数消化率も高い数値を示している．これは事前調査および事前調査レビューによってスプリントの初期に後戻りのリスクを察知し，早々に対策をとることができた結果だと考えられる．

4.3.1　全体的な考察

4.3.1.1　適用前

適用前バーンダウンチャートでは，スプリントの丁度中間辺りにこぶができており，そこから徐々にスプリントゴールの達成へ向けてタスク残工数の消化速度が向上し始めている．これは Kane Mar 氏が提示するバーンダウンチャートカテゴリの「中間学習者」[6]の特徴に酷似している．ただし，後半にまたタスク残工数の消化速度の減退が見られ，計画の約 18%のタスクが消化できなかった．

4.3.1.2　適用後

適用後バーンダウンチャートでは，実績線が理想線を縫うように引かれている．特に大きな問題が発生している気配も見られず，スプリント計画に沿った開発が進められているようだった．これは Hiren Doshi 氏が提示する自己組織化されたスクラムチームのバーンダウンチャートの特徴[6]に酷似している．この結果は前述でも示したように，事前調査レビューの実施が安定した開発状況を生み出せたと考えられる．反対に事前調査レビューを実施せずに後戻りのリスクを潜在させたまま進めていた適用前では，最後までそれを解消できずに計画に従った開発ができなかったとも言える．そのため，本提案手法の適用により，開発状況の改善に効果があったものと考えられる．ただし，こちらも適用前と同様に，後半にタスク残工数の消化速度の減退が見られ，結果として全タスク消化は未達となっている．これには 2 つの原因が考えられる．1 つ目は事前にタスクの調査を行っていたとしても全ての問題を事前に察知することができず，スプリントでこなせるタスク量を超過してしまったため．2 つ目はスプリント後半にある必須イベントの対応により実質的な作業時間確保ができなかったため．今回の実験対象としたスクラムチームでは，スプリント最終日は午後にスプリントレビューとレトロスペクティブを実施するサイクルでスプリントを回していた．そのため最終日は実質的に午前中(0.5 日分)の作業時間しか確保できていなかった．よって今回の実験例では，スクラムチームが未成熟であったため，実質的にスプリントに確保できる作業時間に対して PBI を選択することができていなかったためと考えられる．

4.3.2　スプリント前期(1~3 日目)の比較

適用前では，実績線が理想線の傾きよりも横ばいの状態になっており，タスクの消化が追いついていない．これは，タスクに着手はしたがタスク作業の調査も並行して行ってしまっているため，タスクの作業完了までに時間が掛かっていると考えられる．

適用後では，初日は適用前同様，ほぼタスクが消化できていないが，その後は理想線の傾きに並行する形で進んでいることがわかる．このスプリント前期のタスク残工数の消化率の高さは，「事前調査タスク」を優先して作業していることが要因と考えられる．「事前調査タスク」は 1 タスクの作業規模が小さく，実施する内容が明確であるということで計画工数消化率の増加に起因している．

4.3.3　スプリント中期(4~6 日目)の比較

適用前では，実績線が理想線を上回った状態が続いており，前半遅延分のリカバリが追い付いていない．遅延中タスクへの対策が追い付いていない状況と取れる．

適用後では，スプリント前期からのタスク残工数の消化速度をそのまま維持しており，理想線を下回った状態で開発が順調に進んでいることが伺える．

4.3.4　スプリント後期(7~10 日目)の比較

適用前では，全タスク消化に向かっているが，前半遅延分のリカバリが間に合わずに未

消化のタスクが多く残ってしまっている.

適用後では，最終日がやや失速したことにより全タスクの消化は未達成とはなっているが，適用前と比較して計画工数の消化率は上回っている.

4.3.5 提案手法に対する考察

提案手法は無意識に行っていた事前調査を意識的に行い，また，実際の作業タスクに着手する前に実施可能なレビューを先に実施することは，開発チームにとって負担が少ないものであった．事前調査レビューによって検出した指摘は，従来法では作業後に手戻りとなっていた事柄だったはずだが,開発者はその指摘を事前に認知することが可能になった．開発チームにとって導入コストが少なく,得られる効果の高い手法であることが分かった.

5. おわりに

本稿では，すべての PBI に対して「事前調査タスク」,「事前調査レビュータスク」そして実際の「作業タスク」にタスクを分割し，すべての「事前調査タスク」と「事前調査レビュータスク」をスプリントの初期に実施する手法を提案した．提案手法によって，スプリント計画（SP）時点で不明確であったタスクの規模や，タスク漏れをスプリントの初期で明らかにすることができた．4.3 章で定義した計画工数消化率により，提案手法の適用前より適用後で開発チームのベロシティを一定以上に保てていることも確認できた．さらに，不明確でないタスクを含めすべてのタスクに対して事前調査とレビューを行ったことで作業内容が明確になり，作業着手した際の作業効率の向上に寄与した.

今後，少なくとも次の 3 点について検討が必要であると考える．まず実験対象の開発チームではタスク管理にチケットを利用していたが，事前調査タスクにおいて「件名の事柄について実装する」という事柄しか書かれていないことがあった．本研究の趣旨を共有しきれておらず，再度詳細に記載してもらわなければならなかった．こちらに対しては開発チーム内で記載内容が統一できるような雛形を作ると改善が見込めると考えている．次に本研究では開発チームのみで,PBI を DONE にするために必要なタスクがすべて揃っているか，また各タスクの作業内容が明確になっているかという観点でレビューを行った．今回は起こらなかったが，お客様の要求が PBI で表せていないのではないかという観点もレビューには必要であったと思う．最後に本研究ではスプリントの期間が 10 営業日に対して SP を半日で行い，今回の提案内容の事前調査タスクに 3 日掛かり，事前調査レビュータスクに 1 日ほど使った．結果として事前調査とそのレビューにスプリント期間の半分程度のタイムボックスを利用していた．このような時間の使い方でも良いのか，あるいはそれだけの時間を確保する必要があるのならば，PBI から調査タスクを分割するのではなく，始めから別の PBI として管理した方が良いのか,について検討する必要があると考えている.

参考文献

[1] Ken Schwaber and Jeff Sutherland, “Scrum Guide”, ---, ---, ---, 2017

[2] 独立行政法人 情報処理推進機構(IPA), “アジャイル開発の進め方”, ---, ---, p8, 2018, https://www.ipa.go.jp/files/000065606.pdf

[3] Michael James and Luke Walter, “スクラムリファレンスカード”, ---, ---, p2, 2018, http://scrumreferencecard.com/ScrumReferenceCard_v1_3_jp.pdf

[4] IEEE-SA Standards Board, ”IEEE Recommended Practice for Software Requirement Specifications (IEEE Std 830-1998)”, ---, ---, p4-9, 1998

[5] 大立薫, 千葉美千代, “設計者十進による設計品質作り込み -テストエンジニア視点の活かし方-”, SQiP 研究会, ---, p3, 2009

[6] Vikas Hazrati(翻訳：南 伸二), “バーンダウンチャートを解読する”, InfoQ, ---, ---, 2010, https://www.infoq.com/jp/news/2010/02/deciphering-burndown-charts

（凡例)[番号] 著者名，論文名，雑誌名，巻号，ページ，発行年