

探索的テストの「気づき」に基づきプロダクトバックログ の書き方の改善点を導出する手法

2022年10月6日

○谷崎 浩一¹, 田上 諭¹, 森 龍二¹, 蛭田 恭章¹, 森崎 修司²

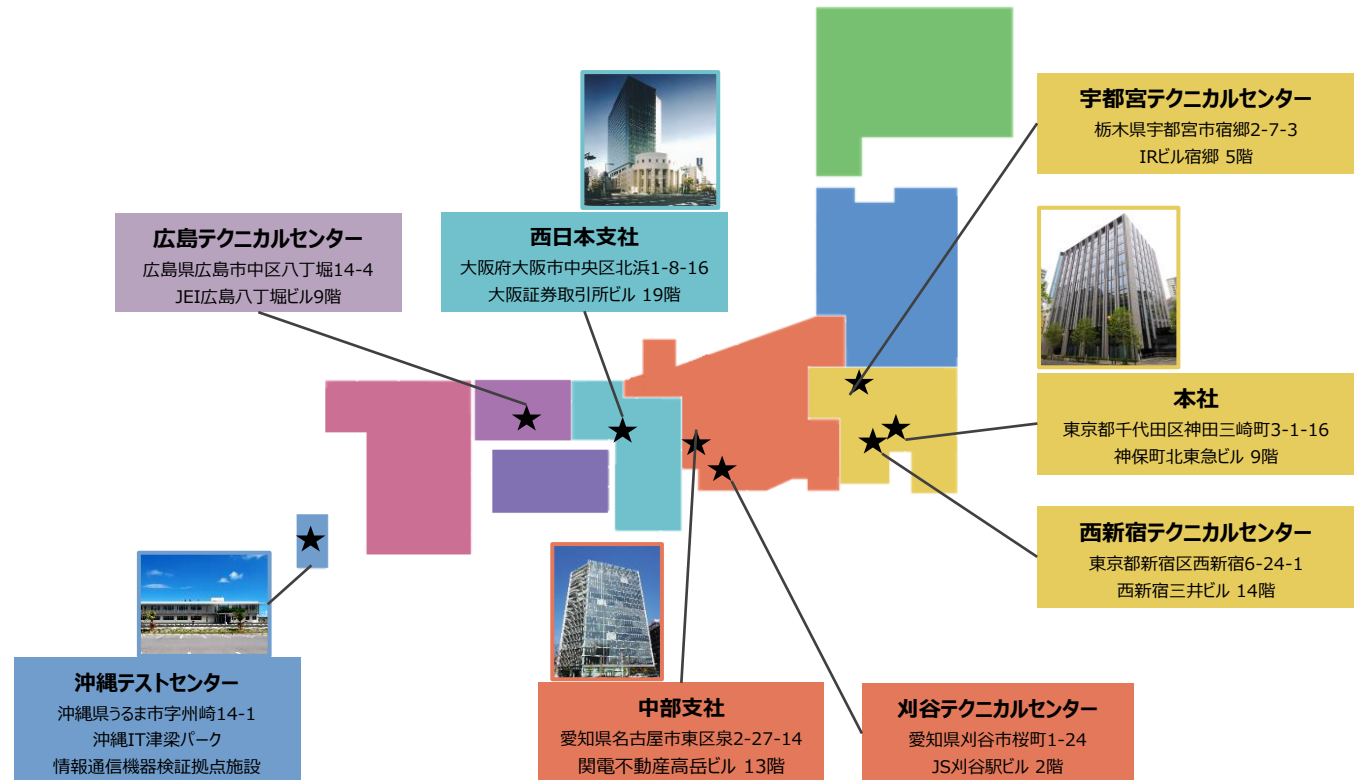
¹ 株式会社ベリサーブ, ² 名古屋大学大学院情報学研究科

- 背景
- 改善したいこと
- 改善策を導き出した経緯
- 改善策の内容
- 改善策の実現方法
- 改善による変化や効果
- まとめと今後の課題

➤ 約40年にわたりソフトウェア検証で品質向上に貢献

- 仕様などの要求事項が満たされているかを評価する **Verification** と、機能や性能が本来意図された用途や目的に合っているかを評価する **Validation**。
当社の社名にはこの **2つのV** を提供する (**Service**) という思いが込められています。

社 名	株式会社ベリサーブ (VeriServe Corporation)
設 立	2001年7月24日
代 表 者	代表取締役社長 新堀 義之
従業員数	1,452名（2022年3月31日現在 連結）
資 本 金	792百万円（2022年3月31日現在）
事業内容	1. 製品検証サービス 2. セキュリティ検証サービス 3. その他サービス
子 会 社	株式会社ベリサーブ沖縄テストセンター AIQVE ONE株式会社
拠 点	右記参照



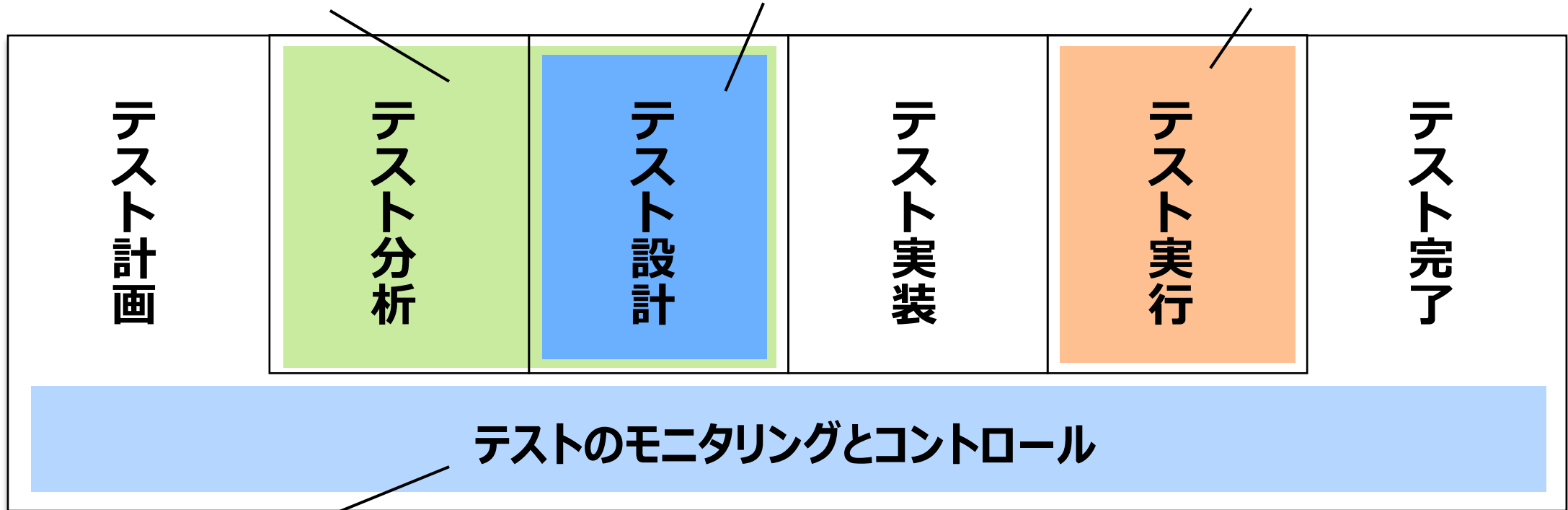
➤ テストプロセスのさまざまな工程を支援するプロダクトを開発・運用中

今回の取り組みで
活用！

 **TESTSTRUCTURE**
業界初のテスト分析/設計支援ツール

 **GIHOZ**
“今すぐ使える”テスト技法ツール

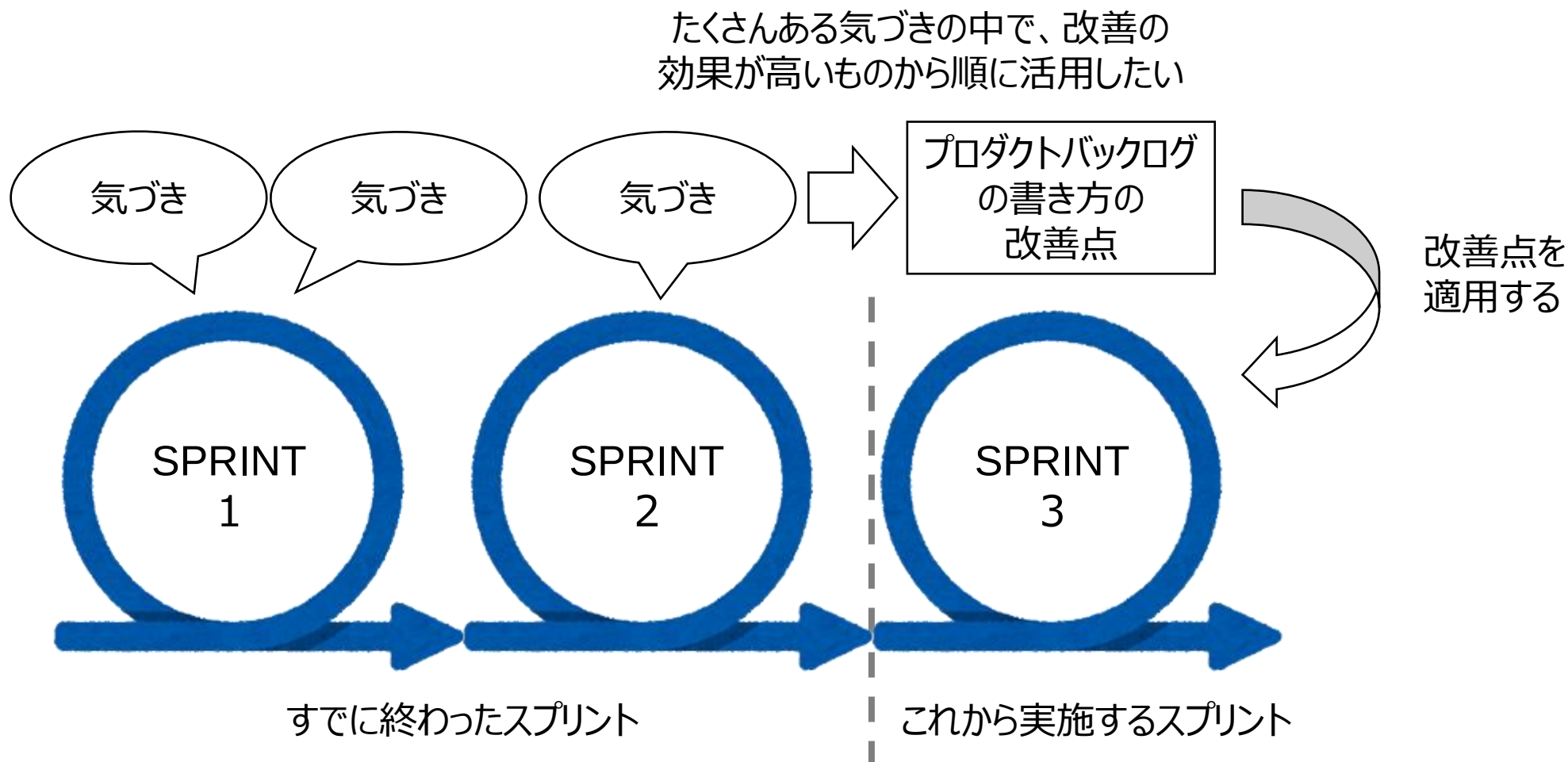
InsighTest
チームによる探索的テストを加速させるツール



 **QualityForward** テスト管理をシンプルに。快適、スケーラブルなテスト管理ツール

- スクラムでは開発する機能や実施するタスクや修正するバグなどを、プロダクトバックログとして列挙するが、その書き方には唯一のものがあるわけではない
 - ユーザストーリー形式での記述や、受け入れ基準を記載することは一般的に知られている
 - ◆ ユーザストーリー形式の例：<ユーザ>として<機能や性能>がほしい。なぜなら<理由>だから。
- プロダクトバックログで定義すべき項目に定義漏れがあった場合、実装後に修正点が見つかり、修正と再テストが発生し、プロダクトのユーザに対して価値を届けることが遅くなる
- 発表者が参画するプロジェクトでも実装後に修正と再テストが発生しており、社内のWGや名古屋大学との共同研究にてプロダクトバックログの書き方の改善点を導出する方法について検討した

- 過去のスプリントで得られた情報から、次以降のスプリントに向けてのプロダクトバックログの書き方の改善点を見つけて、実装完了後の修正・再テストにより発生する工数を減らしたい



あるクラウドサービスにおけるプロダクトバックログの例

ユーザストーリーや説明

強いパスワードを入力したい

添付 子課題を追加 課題をリンク ▼ ...

説明

ユーザとして強いパスワードを入力したい。なぜならパスワードを予測されるリスクを減らしたいからだ。

以下のようなインジケータを付けるとともに、入力可能なパスワードとして「数字とアルファベットを含む長さ 8 文字以上」という制限をかける。

必須 パスワード

①パスワードの強度が色で表示されます。安全に利用するために緑色にするのがお勧めです。

必須 確認用パスワード

もう一度入力してください

受け入れ基準や付随する情報

受入条件

デザインは  の案1を参照。

• 共通

- パスワードの強さのインジケータは5段階で表示される。強さの判定のアルゴリズムは○○○○に従う。
- 以下の文字を表示
 - 既定の文字数に満たない場合：短すぎます
 - パスワードの強度が最も弱い場合：弱い
 - パスワードの強度がやや弱い場合：やや弱い
 - パスワードの強度が中くらいの場合：普通
 - パスワードの強度がやや強い場合：やや強い
 - パスワードの強度が最も強い場合：強い

• アカウントの新規登録

- アカウントの新規登録画面に、パスワードの強さのインジケータが表示される。数字や文字を入力するたびに、インジケータの表示が更新される。
- パスワードの強度が「短すぎます」か「弱い」の場合はアカウントを作成できない
 - 「やや弱い」以上の場合、アカウント作成できる。

- アジャイル開発においては実装後のテストで探索的テストが多く活用されている
- 探索的テストでは、テスト実施者の経験・知識と照らし合わせて違和感のある挙動が、バグとは別に「気になった点」として報告されることがある
 - バックログで定義された挙動と、実際のプロダクトの挙動が異なる → バグ
 - バックログで明示的に定義されていないが、違和感ある挙動など → 気になった点
- 本発表では、「気になった点」のことを「気づき」と呼んでいる
 - 気づきには、プロダクトの使い勝手に関する違和感など、バックログには明示されていないが、プロダクトの価値の向上につながる改善点が含まれる
- 「気づきを分析することで、プロダクトバックログで定義が不十分な項目を明らかにできる」と仮説を立て、気づきに基づいてプロダクトバックログの書き方の改善点を導出する方法を考案した

- 気づきの情報を分析することで、プロダクトバックログの作成や検討の段階で定義可能だが定義できていない項目を明らかにし、その項目を定義できるようにするための改善点を導出する

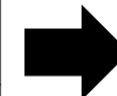
気づき
K
k さん
2022/05/26 17:50
機能A
xxxのパターンの際の動作が分かりにくいかもしれない

気づき
K
k さん
2022/05/26 17:50
機能A
英語メッセージの文言に違和感がある

FAIL
K
k さん
2022/05/26 17:50
機能A
ボタンAを押しても機能Aが動作しない



気づき	修正の工数	再テストの工数	合計工数
xxxのパターンの際の動作が分かりにくいかもしれない	1h	3h	4h
英語メッセージの文言に違和感がある	0.25h	0.25h	0.5h
.....	0.5h	1h	1.5h



バックログの書き方の改善点
yyyを受け入れ条件に定義する
zzzを記載できるフォーマットに変更する
.....

手順1：探索的テストの結果をバグと気づきに分けて記録する

手順2：気づきに対して、修正および再テストにかかった工数の情報を付与する

手順3：気づきの情報に基づいて、プロダクトバックログの書き方の改善点を導出する

➤ 探索的テストの結果記録用のツールを使って、バグと気づきに分けて結果を記録する

■ 表計算ソフトなどで記録しても良い

テスト実施

☰ トライーヴ >

新規トライを追加 +

⚙️ 対象機能一覧 >

テスト終了処理

✓ クローキング >

エクスポート

ショットをダウンロード ⬇

<

新規ショットを追加

テスト内容と結果 必須

簡潔に、チームメンバーにもわかりやすく記入しましょう

📋 ペーストでクリップボードの画像が添付できます

期待する動作 任意

特にFAILにする場合は期待値を記入しましょう

ステータス 必須

PASS FAIL 気づき

共有する

ショット一覧

個人 機能 全体

全て 4 件 PASS 1 件 FAIL 1 件 気づき 2 件

気づき K k さん

🕒 2022/05/26 17:37

テストカバレッジアイテムの名前を編集したい
カバレッジアイテムの名前が空のときはツールチップはなくて良さそう

気づき K k さん

🕒 2022/05/26 17:36

テストカバレッジアイテムの名前を編集したい
文字入力毎に自動保存が走る

FAIL K k さん

🕒 2022/05/26 17:36

テストカバレッジアイテムの名前を編集したい
過去に作成したデータを開くと、テストカバレッジアイテムの名前が表示されず、図の表示がずれる

PASS K k さん

🕒 2022/05/26 17:34

テストカバレッジアイテムの名前を編集したい

- 結果記録ツールから気づきの情報を抽出し、気づきの一覧表を表計算ソフトで作成する
- 開発チームのメンバ全員で話し合いながら修正や再テストにかかった工数の数値を入力する

	A	B	C	F	M	N	O
	No.	セッション	トライ: ターゲット機能	気づき: メモ	修正にかかった工数 (h)	再テストにかかった工数 (h)	未然に防げたかもしれない工数合計 (h)
1				各工数は0.25h単位で入力する			
74	73	70	TESプリント TEB-1206-デシジョンテーブル で複数の行を一気に追加したい	複数行選択した際も、英語版のメニューだと「Insert 1」になっているのが違和感あるかも	0.25	0.25	0.5
75	74	70	TESプリント TEB-1206-デシジョンテーブル で複数の行を一気に追加したい	条件記述部で親子を選んで行を挿入した際、子階層に行が追加されるのが、少し違和感あるかも？親階層に行が追加されるほうが自然かも。	1	2	3
76	75	70	TESプリント TES仕様一覧でテストモデル・ケース仕様を一括コピーしたい	コピーした先に同じファイル名がある場合の動作が "ファイル名(1)"になっている 普通に作成して保存した場合"ファイル名 (1)"	0.25	0.25	0.5
77	76	70	TESプリント TES仕様一覧でテストモデル・ケース仕様を一括コピーしたい	コピー作成のダイアログで、リポジトリ選択のプルダウンは、現在のリポジトリがデフォルトで選択されていると良いかも	0.25	0.25	0.5
78	77	70	TESプリント TES仕様一覧でテストモデル・ケース仕様を一括コピーしたい	削除済みのデータをコピーした場合、コピーに成功した旨のメッセージが表示されるのが、少し違和感ある（コピーダイアログを表示している間に、裏で別の人がそのデータを削除していた場合など）	0.75	0.5	1.25
87	86	72	TESプリント TEB-1237-ペアワイズ、 minimizeの黒ボチを自動でつくれるように	二つ目のクラフィケーションの下にクラスがない場合ペアワイズで自動組み合わせすると無限読み込み	0.25	1	1.25
88	87	72	TESプリント TEB-1237-ペアワイズ、 minimizeの黒ボチを自動でつくれるように	この状態でも組み合わせ生成は実行できても良い気はする（実装上の制約でなければ、あえてボタンをグレーアウトする意味はなさそう）	0.25	0.5	0.75
89	88	72	TESプリント TEB-1354-クラシフィケーションの下にクラシフィケーションを作れることに気づきやすくしたい	Rootはクラスを追加できないので、クラスを追加のメニューは非表示にしたほうが良さそう	0.25	0.25	0.5

- 開発チームのメンバ全員で、手順2で作成した一覧表を見ながら、修正および再テストにかかる工数
 が大きい気づきを中心に、プロダクトバックログの書き方の改善点を話し合う

工数の列でソートしても良い

	A	B	C	F	M	N	O
1	No.	セッション	トライ：ターゲット機能	気づき：メモ	修正にかかった工数 (h)	再テストにかかった工数 (h)	未然に防げたかもしれない 工数合計 (h)
74	73	70	TESスプリント TEB-1206-デシジョンテーブル で複数の行を一気に追加したい	複数行選択した際も、英語版のメニューだと「Insert 1」になっているのが違和感あるかも	0.25	0.25	0.5
75	74	70	TESスプリント TEB-1206-デシジョンテーブル で複数の行を一気に追加したい	条件記述部で親子を選んで行を挿入した際、子階層に行が追加されるのが、少し違和感あるかも？親階層に行が追加されるほうが自然かも。	1	2	3
76	75	70	TESスプリント TEB-1156-テストモデル・ケース 仕様一覧でテストモデル・ケース仕様を一括コピーしたい	コピーした先に同じファイル名がある場合の動作が 「ファイル名(1)」になっている 普通に作成して保存した場合「ファイル名 (1)」	0.25	0.25	0.5
77	76	70	TESスプリント TEB-1156-テストモデル・ケース 仕様一覧でテストモデル・ケース仕様を一括コピーしたい	コピー作成のダイアログで、リポジトリ選択のプルダウンは、現在のリポジトリがデフォルトで選択されていると良いかも	0.25	0.25	0.5
78	77	70	TESスプリント TEB-1156-テストモデル・ケース 仕様一覧でテストモデル・ケース仕様を一括コピーしたい	削除済みのデータをコピーした場合、コピーに成功した旨のメッセージが表示されるのが、少し違和感ある（コピーダイアログを表示している間に、裏で別の人がそのデータを削除していた場合など）	0.75	0.5	1.25
87	86	72	TESスプリント TEB-1237-ペアワイズ、minimizeの黒ガチを自動でつくれるように	二つ目のクラフィケーションの下にクラスがない場合ペアワイズで自動組み合わせると無限読み込み	0.25	1	1.25
88	87	72	TESスプリント TEB-1237-ペアワイズ、minimizeの黒ガチを自動でつくれるように	この状態でも組み合わせ生成は実行できても良い気はする（実装上の制約でなければ、あえてボタンをグレーアウトする意味はなさそう）	0.25	0.5	0.75
89	88	72	TESスプリント TEB-1354-クラシフィケーションの下にクラシフィケーションを作ることに気づきやすくない	Rootはクラスを追加できないので、クラスを追加のメニューは非表示にしたほうが良さそう	0.25	0.25	0.5

手順2で作成した一覧表

UI上の文言の
精査があると
良さそう（日
本語・英語）

デザイン段階で
デザイナーとP
Oで文言とか動
作をすり合わせ
すると良さそう

操作のパターンや
入力のパターンが
多いところは漏れ
がち

事前にパターン整
理しておいて、開
発段階でバグつぶ
しておけば、再テ
ストで無駄になる
工数は減らせそう

付箋などで改善点を
書き出したもの

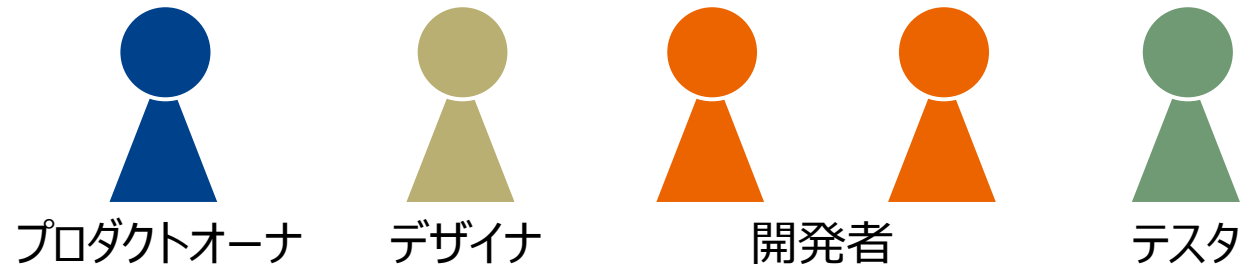
- あるプロジェクトにて、以下の3つの方法でプロダクトバックログの書き方の改善点を洗い出し、改善の効果（実装後の修正・再テストの工数をどのくらい削減できそうか）を試算した

- 各方法で30～60分のオンラインミーティングを設け、チームメンバ全員が参加して実施した

方法	概要	詳細
①	従来手法を利用	振り返りのフレームワークの1つである KPT を用いてプロダクトバックログの書き方の改善点を洗い出す ※KPT：Keep・Problem・Tryを振り返る方法
②	提案手法の一部を利用	探索的テストの 気づきを記録された順にすべて確認してから プロダクトバックログの書き方の改善点を洗い出す
③	提案手法を利用	探索的テストの気づきを記録された順にすべて確認しながら 修正・再テストにかかった工数のデータを付与し、その上で工数の大きかった気づきから プロダクトバックログの書き方の改善点を洗い出す

【仮説】客観的なデータに基づき優先順位付けし、改善点を検討することで、改善の効果が高くなる

➤ Webアプリケーションの開発を行う5名のアジャイル開発チーム



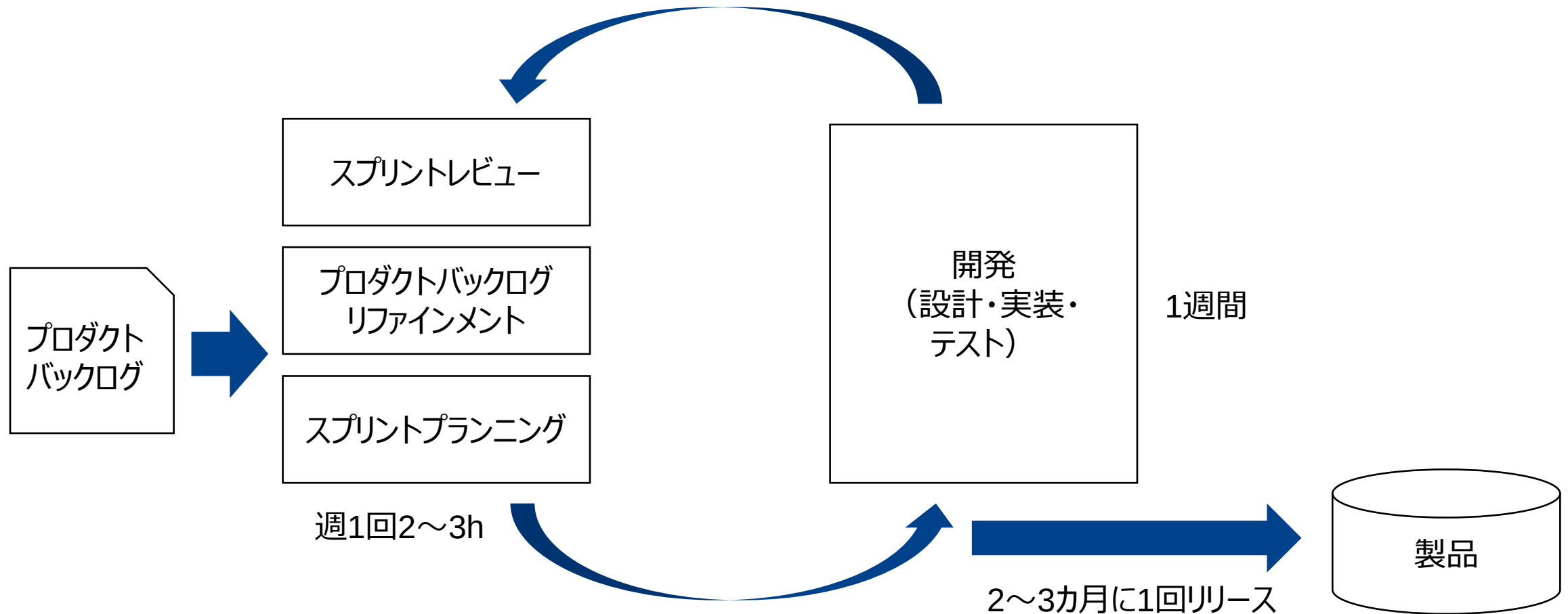
➤ プロジェクトの開始から約3年経過、現在のメンバ構成になってから約1.5年経過

➤ 開発者による実装後、主にプロダクトオーナー（PO）とテストが、探索的テストでバックログの受け入れ確認を実施している

- このチームでは、実装後のプロダクトを触ってバックログを完了（Done）にしてよいか判定することを、「受け入れ確認」と呼んでいる
- 最終的な完了の判断はPOが行うが、この確認の際にPOとテストがバグや気づきを出している

対象プロジェクトの開発の進め方と対象にしたスプリント

- スクラムを参考にし、1週間を1スプリントとして開発を行っている
- 当該プロジェクトの1カ月分（4スプリント）を対象に、先述の①～③の方法でプロダクトバックログの書き方の改善点を導出した。①～③の各方法で、それぞれ異なる4スプリントを対象とした



- ②（提案手法の一部を利用）の対象としたスプリントでは探索的テストによる気づきが20個あった
- ③（提案手法を利用）の対象としたスプリントでは探索的テストによる気づきが24個あった

	A	B	C	F
1	No.	セッション	トライ:ターゲット機能	気づき: メモ
131	130	83	TEB-1200-テストモデル・ケース 仕様にメモやコメントを記載したい	返信元のコメントを削除すると返信した内容も消えるが削除確認の際に返信した内容が表示されない
132	131	83	TEB-1200-テストモデル・ケース 仕様にメモやコメントを記載したい	公開リポジトリに作成したテストケースを未ログインや招待されていないアカウントで表示した場合にコメントが表示されない・コメントが送信できそうでできていない
133	132	83	TEB-1200-テストモデル・ケース 仕様にメモやコメントを記載したい	【修正確認済】保存ボタンの左のアイコンを押してコメントを表示したとき、コン
134	133	83	TEB-1200-テストモデル・ケース 仕様にメモやコメントを記載したい	ヘッドの一番上が一番古い内容 いう順番が違和感あるかも
135	134	83	TEB-1200-テストモデル・ケース 仕様にメモやコメントを記載したい	目のコメントの間にあるのが 言アイコンがある。
136	135	83	TEB-1200-テストモデル・ケース 仕様にメモやコメントを記載したい	が表示されていない
137	136	83	TEB-1200-テストモデル・ケース 仕様にメモやコメントを記載したい	【修正済み】コメントを削除の確認ダイアログで、横スクロールできる
138	137	83	TEB-1200-テストモデル・ケース 仕様にメモやコメントを記載したい	v-dividerの先の太さがデザインと違う？
139	138	83	TEB-1200-テストモデル・ケース 仕様にメモやコメントを記載したい	公開リポジトリのテストケースでコメントがない場合に未ログイン・参加していない ユーザーでもコメントするボタンが表示されるログインしている場合はコメント投稿 画面になりサーバーエラーが表示される
140	139	83	TEB-1200-テストモデル・ケース 仕様にメモやコメントを記載したい	公開リポジトリのテストケースでコメントがあり、参加していないメンバーで表示 した場合にコメントボタンが表示されてしまう

②の対象：20個の気づき

	A	B	C	F
1	No.	セッション	トライ:ターゲット機能	気づき: メモ
74	73	70	TESスプリント TEB-1206-デシジョンテーブルで 複数の行を一気に追加したい	複数行選択した際も、英語版のメニューだと「Insert 1」になっているのが違和感あ るかも
75	74	70	TESスプリント TEB-1206-デシジョンテーブルで 複数の行を一気に追加したい	条件記述部で親子を選んで行を挿入した際、子階層に行が追加されるのが、少し違和 感あるかも？親階層に行が追加されるほうが自然かも。
76	75	70	TESスプリント TEB-1156-テストモデル・ケース 仕様一覧でテストモデル・ケース 仕様を一括コピーしたい	コピーした先に同じファイル名がある場合の動作が "ファイル名(1)"になっている 普通に作成して保存した場合"ファイル名(1)"
77	76	70	TESスプリント TEB-1156-テストモデル・ケース 仕様一覧で 仕様を一括コピーしたい	現在のリポジトリがデ
78	77	70	TESスプリント TEB-1156-テストモデル・ケース 仕様一覧で 仕様を一括コピーしたい	のメッセージが表示される 別に、裏で別の人がその
86	85	72	TESスプリント TEB-1237- の黒ボチを	
87	86	72	TESスプリント TEB-1237- の黒ボチを自動でつくれるように	すると無限読み込み
88	87	72	TESスプリント TEB-1237-ペアウィズ、minimize の黒ボチを自動でつくれるように	この状態でも組み合わせ生成は実行できても良い気はする（実装上の制約でなけれ ば、あえてボタンをグレイアウトする意味はなさそう）
89	88	72	TESスプリント TEB-1354-クラシフィケーション の下にクラシフィケーションを作 れることに気づきやすきたい	Rootはクラスを追加できないので、クラスを追加のメニューは非表示にしたほうが 良さそう
90	89	72	TESスプリント TEB-1329-ツリーを拡大・縮小し て全体が一目で見えるようにした い	状態遷移図の拡大縮小と操作が違うので、どちらかに統一したほうが良い（状態遷移 図はマウスホイールの回転のみで拡大縮小できるが、クラシフィケーションツリーは Ctrlキーを押しながらマウスホイールを回転する必要がある）
			TESスプリント TEB-1329-ツリーを拡大・縮小し て全体が一目で見えるようにした い	マウスホイールの動作はマウスホイールの押し込みで切り替えられるようです

③の対象：24個の気づき

➤ 以下の式で算出する

$$\text{改善による工数削減効果} = \text{修正にかかった工数} + \text{再テストにかかった工数} - \text{未然防止にかかる工数}$$

➤ 具体的には以下の手順を実施した

- ②と③で対象としたスプリントの気づきに対して、改善点を事前に実施した場合に未然防止できたかどうかを確認
- 未然防止できたものについて、修正・再テストにかかった工数の合計をチームメンバーで話し合っ算出
- 未然防止にかかる工数をプロダクトオーナーが見積もり
- 上記の式で工数削減効果を算出

➤ 以下の改善点が導出された

方法	導出された改善点
①	プロダクトオーナーが英語版の画面デザインを作ってみる ※英語版のUIの文言定義がされていないことが多いというProblemに対する改善点 バックログのテンプレートを、<Who> <What> <Why> <How> を分けて書けるような形式にする ※「バックログの背景を忘れがち」「ユーザストーリーの形式だと日本語だと読みにくい」というProblemに対する改善点
②	ログイン時（権限あり・なし）と未ログイン時で画面表示したときのデザインをそれぞれ検討する z-indexを定義する（WebアプリケーションのUIのどの要素が前面に表示されるべきかの定義） （特にAPIの開発時に）受け入れ条件で、入出力のパターンを定義する。実際のリクエスト・レスポンス形式を実装前に検討する。
③	画面デザインの段階でデザイナーとプロダクトオーナーで文言や動作について擦り合わせし、日本語・英語ともにUI上の文言の精査を行う 操作のパターンや入力のパターンが多いところは漏れがちなため、バックログの段階でパターンを整理する

➤ ③の方法が、改善による工数削減効果がもっとも高い結果となった

- 客観的なデータに基づき優先順位をつけて改善点を検討したことで、改善の効果が高くなった
- ただし、この結果は対象としたスプリントに依存している可能性はある

		未然防止できるコスト		
		②の対象	③の対象	
方法	かかったコスト	スプリント	スプリント	平均
①	$0.5h \times 5名 = 2.5h$	0h	0.75h	0.375h
②	$1h \times 5名 = 5h$	1.75h	0h	0.875h
③	$1h \times 5名 = 5h$	1h	6.25h	3.625h

- 探索的テストの気づきの情報を用いることで、対象としたプロジェクトに固有の具体的なプロダクトバックログの書き方の改善点を導出可能なことが確認できた。
 - ①の方法ではバックログのテンプレートの改善など汎用的な改善点も導出されたが、②の方法では現状のバックログで具体的に定義できていなかった項目に対する改善点を導出できている
 - さらに③の方法では、修正・再テストという実装後の後戻りを防ぐことにフォーカスした改善点を導出できている

- 提案手法を利用した③の場合にもっとも工数削減効果の大きい改善点を導出できたことが分かった
 - ただし、新たなスプリントで改善案を適用してどの程度の工数削減効果があるかは確認できていない
 - ◆ アジャイル開発ではスプリントによって開発する内容が変わるため、同じ改善点がすべてのスプリントで有効とは限らない懸念がある
 - 1スプリントでは工数削減効果が小さくても、スプリントを何度も繰り返せば累計では効果が大きくなると考えられるため、改善点を取り入れてスプリントを繰り返した場合の効果の測定が今後必要

- チームメンバへのヒアリングから、あいまいな部分が減ることによる見積もり精度向上や、経験や技術レベルに依存せずチームの状況に即した改善内容を導出可能であるという効果も見えた

次回スプリント以降のプロダクトバックログの
見積もりの精度は変わるか？

見積もりの精度は良くなると思う

提案手法で導出された改善内容はメンバの
経験や技術レベルに依存するか？

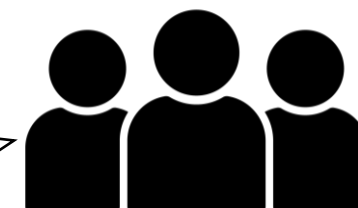
経験や技術レベルにはあまり依存しないと思う

改善点を導出する際のチームの議論のやり方に、
従来のKPTとは異なる部分があったか？

議論のやり方には印象に残るような変化はなかった



発表者



チーム
メンバ

➤ まとめ

- 探索的テストの気づきからプロダクトバックログの書き方の改善点を導出する方法を考案した
- 実際にその方法で改善点を導出できることと、提案手法を使った場合に工数削減効果の高い改善点を導出できることを確認した
- アジャイル開発では開発の進め方やプロダクトバックログの書き方はチームごとに議論してやりやすいように改善していくというスタイルが一般的であり、提案手法はアジャイル開発のスタイルにも合致する方法と考えている

➤ 今後の課題

- 提案手法で導出したプロダクトバックログの書き方の改善点を実行して、スプリントを繰り返した場合の効果の測定
- 他のアジャイル開発チームで提案手法を適用した場合の効果の検証や、提案手法をうまく活用するための探索的テストのやり方の研究

ご清聴ありがとうございました。
本発表に関するお問い合わせは下記までご連絡ください。

株式会社ベリサーブ

ベリサーブ

検索

研究企画開発部 谷崎 浩一

<kouichi.tanizaki@veriserve.co.jp>

TEL : 03-6629-8540

東京都千代田区神田三崎町3-1-16 神保町北東急ビル 9F