JaSST'22 Hokkaido

探索的テストの導入効果
および
実施方法の納得性を高める
ための取り組み事例

Quality Assurance Service
with high performance
and good agility

2022/7/15

日本ナレッジ株式会社
藤澤 敏浩

発表者の自己紹介

氏名：藤澤　敏浩（ふじさわ　としひろ）

所属：日本ナレッジ株式会社
検証事業本部　検証部
札幌検証センター　センター長

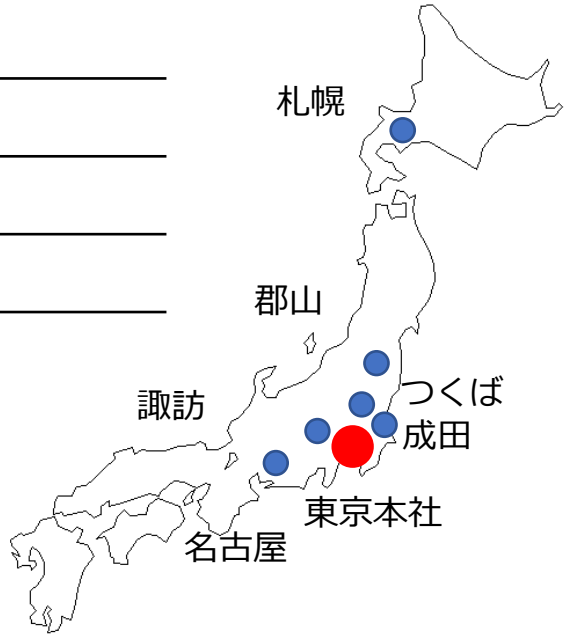
経歴：

- 1996年にIT業界へ就職。ガラケーおよびガラケーPFの検証業務や品質評価業務に携わる。
- 2013年に日本ナレッジへ転職。スマホ、IoT家電、業務系システム、Webシステムなど、様々な検証業務に携わる。
- IT検証技術者認定試験(IVEC) Lv.5取得後、2019年からIVEC試験開発委員として活動中。
- 冬の休日はスキーインストラクターとしても活動中。

所属企業のご紹介

日本ナレッジ株式会社

本社所在地	〒111-0042 東京都台東区寿3-19-5 JSビル9階
拠点	札幌事業所、郡山センター、つくばセンター、成田センター、諏訪センター、名古屋センター
設立	1985年10月
代表取締役	藤井 洋一
資本金	8,600万円
従業員数	330名 (2022年4月時点)
売上	32.6億円 (2021年度：第37期)予定



主な加盟団体

- SAJ　一般社団法人 ソフトウェア協会
- iVIA　一般社団法人 IT検証産業協会

目次

- ✓はじめに・背景
- ✓課題
- ✓課題解決に向けた取り組み
- ✓得られた結果
- ✓今後に向けた課題

✓はじめに・背景

✓課題

✓課題解決に向けた取り組み

✓得られた結果

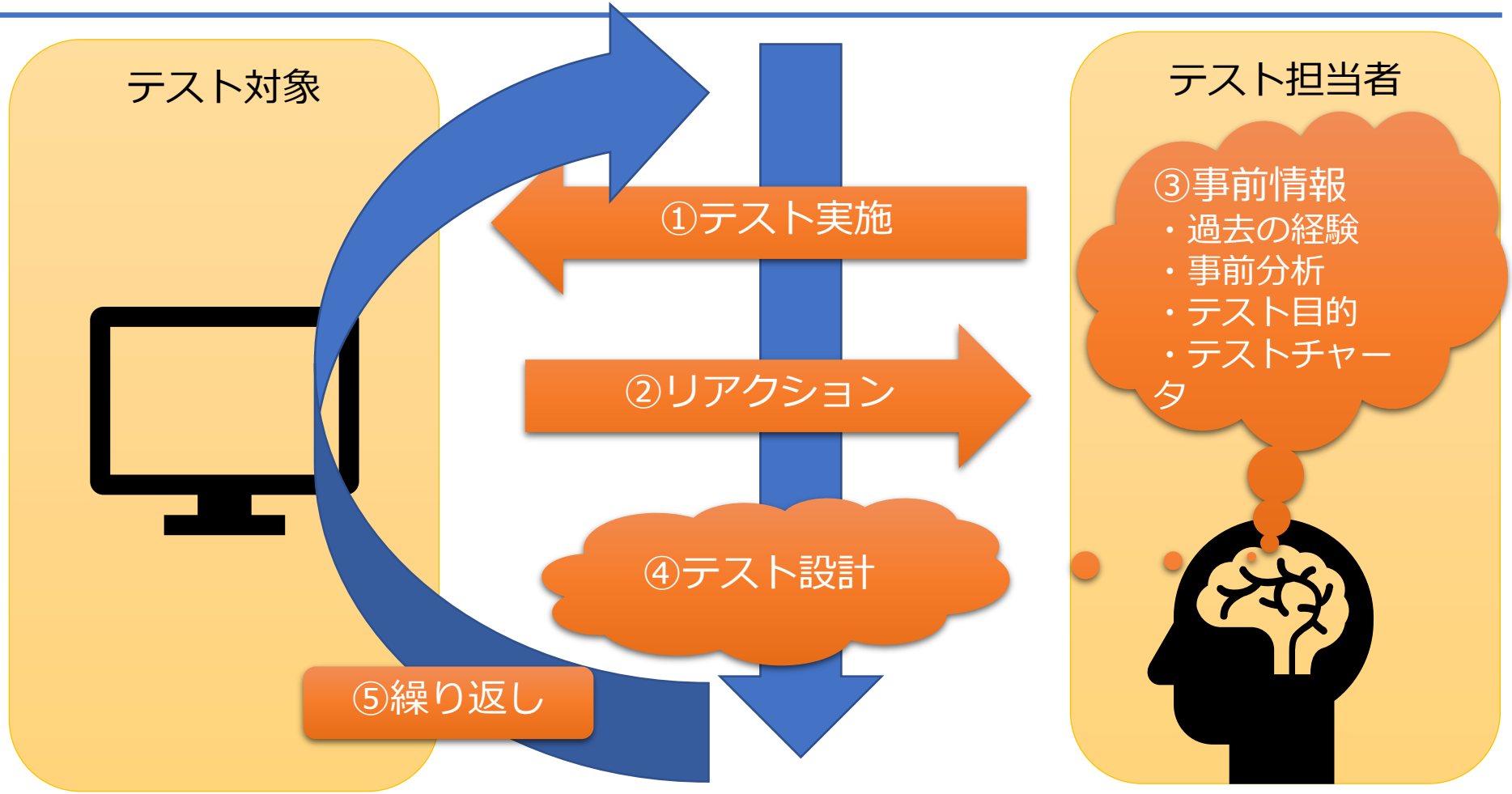
✓今後に向けた課題

探索的テストとは

ISTQBテスト技術者資格制度Advanced Level シラバス
テストアナリスト　Version 3.1.1.J03
3.3.3 探索的テスト
探索的テストには、テスト担当者がテスト対象とその欠陥についての学習、完了すべきテスト作業の計画、テストの設計と実行、および結果の報告を同時に行うという特徴がある。テスト担当者は、テスト実行時にテストのゴールを動的に調整し、軽量のドキュメントのみを準備する

ISTQB Glossary　※https://glossary.istqb.org/
テスト担当者がテストアイテムや以前のテストの結果の知識や調査情報を使用して、テストを動的に設計、および実行するテストアプローチ

探索的テストのイメージ



※テストチャータとは（ISTQB Glossaryより）
テストセッションのゴールや目的を記述したドキュメント

背景

とある業務系システムの開発プロジェクト。

過去実績を見ても、記述式テストで社内指標の試験密度、不具合密度を満たしているのに不具合は流出している。

そんな中、探索的テストというキーワードが目に止まる。

Web記事や書籍等で調べれば調べるほど、探索的テストは不具合流出対策として高い効果がありそうだし魅力的に映る。予算は限られているので、記述式テストの工数を削減して探索的テストの導入を推進したい。

ということで探索的テスト導入に向け具体検討を進めるのだが、導入効果や実施方法をプロジェクト関係者へ説明しても、なかなか納得してもらうことができない。

さて、どうしたものか、、、

納得してもらえないこと

① 探索的テストのメリットとデメリットがはっきりわからない

② 記述式テストの他に探索的テストを導入するとコスト増になるのではないか（予算は限られている）

③ 探索的テストの実施結果は再現性が保証できないのではないか

④ 担当者のスキル依存度が高い手法なので、担当範囲によって品質のバラつきが生じるのではないか

当発表で伝えたいこと

総合試験フェーズにおける探索的テストの導入に向けた取り組み事例である。

導入についてステークホルダーの理解を得られていない状況に対し、どのように課題を整理し、どのように納得性の高い効果測定方法とテスト実施方法を検討し、理解を得ていったかをお伝えしたい。

前提

◆テストチームＡ
⇒記述式テストのみ

◆テストチームＢ
⇒記述式テスト／探索的テスト併用
※ただし、一部機能に限定

テスト分析／テスト設計／テスト実行の
全フェーズにて、2チーム同時進行

✓はじめに・背景

✓課題

✓課題解決に向けた取り組み

✓得られた結果

✓今後に向けた課題

納得してもらえていないこと

課題

以下の4つの問題に対し、納得してもらうために何ができていないかを整理した。

①探索的テストのメリットとデメリットがわからない

②探索的テスト導入はコスト増になるのではないか

③探索的テストの実施結果は再現性が保証できないのではないか

④テスト担当者の担当範囲により品質がバラつくのではないか

②～④は探索的テストのデメリットと言われる部分

納得してもらえていないこと

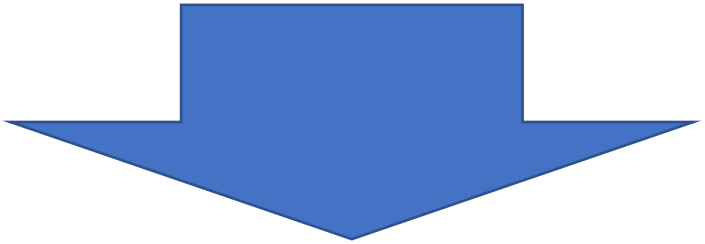
課題

①探索的テストのメリットとデメリットがわからない

・本当に品質があがって、不具合流出も減るのか

・記述式テストの工数を削ったら、網羅率が下がるはず
　網羅率が下がるなら、品質も下がるのではないか

・そもそも、探索的テストでテストの十分性を測れるのか



・探索的テストの実施効果、分析方法を明示的に説明できていない

・探索的テストのデメリットが大きな品質リスクにならない理由を説明できていない

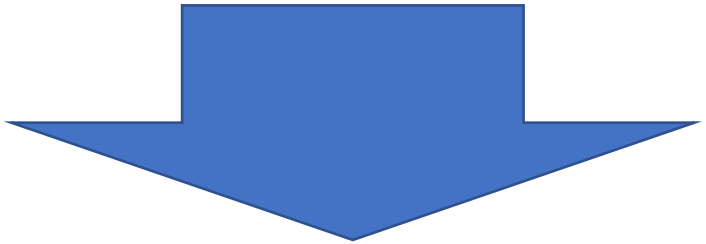
納得してもらえていないこと

課題

②探索的テスト導入によりコスト増になるのではないか

・記述式テストを削って不具合の見逃しが増えるなら、結果的にコスト増になるのではないか

・記述式テストと探索的テストの使い分け方がわからず、品質に影響しない記述式テスト工数の削り方もわからない



・記述式テストの工数を削っても不具合の見逃しが増えるわけではない理由を説明できていない

・記述式テストと探索的テストの棲み分け方を決める必要がある

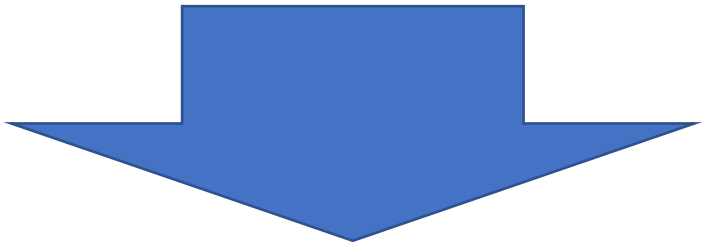
納得してもらえていないこと

課題

③探索的テストの実施結果は再現性が保証できないのではないか

・どのようなテストを行ったかのエビデンスが残らないと困る

・不具合の再現手順もあいまいになるのではないか



・探索的テストにおいて、どのようなエビデンスの残し方ができるか整理できていない

（エビデンスが残しづらい前提で担当者任せになっている）

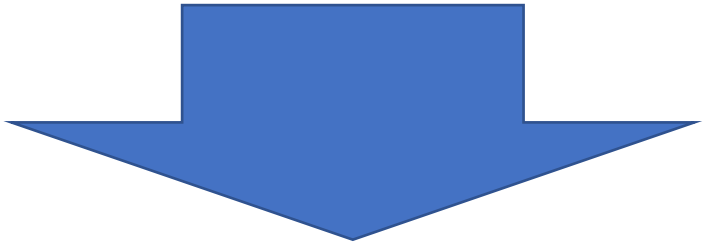
納得してもらえていないこと

課題

④テスト担当者の担当範囲により品質がバラつくのではないか

・スキルが高くない担当者は、テスト観点の漏れが多いのではないか

・担当範囲により、品質のバラつきが大きいのではないか



・探索的テストの準備プロセスが担当者任せになっている

・テスト観点、怪しい部分の勘所などが全て担当者スキル依存

・担当者依存の観点漏れを防ぐ仕掛けがない

✓はじめに・背景

✓課題

✓課題解決に向けた取り組み

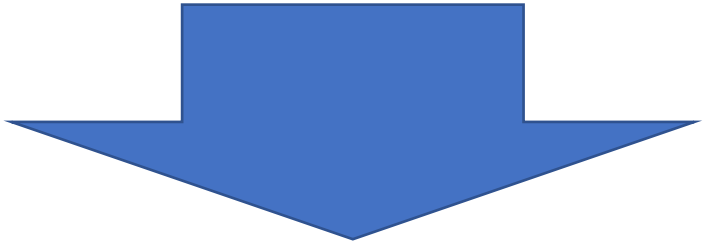
✓得られた結果

✓今後に向けた課題

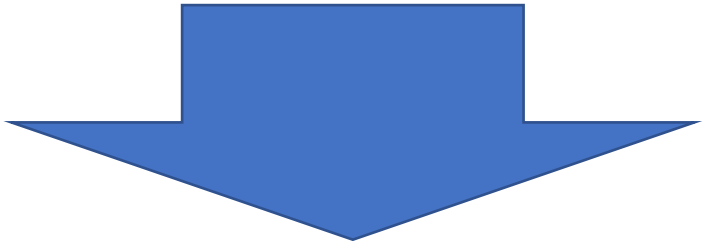
対策①

①-１）品質状況の実施効果や分析方法を明確化

探索的テストは体系化が難しい



テスト対象の分析、テストチャータ、テストエビデンス、不具合分析のいずれも、 JIS X 25010:2013の「SQuaRE品質モデル」にマッピングすることで体系立てた管理ができる

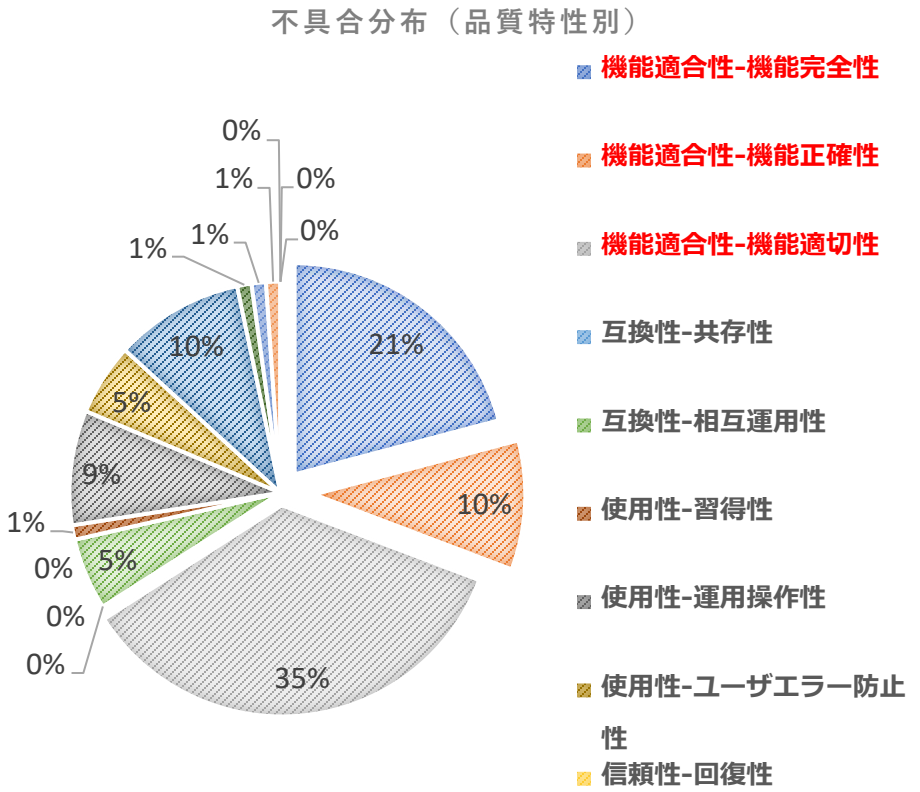


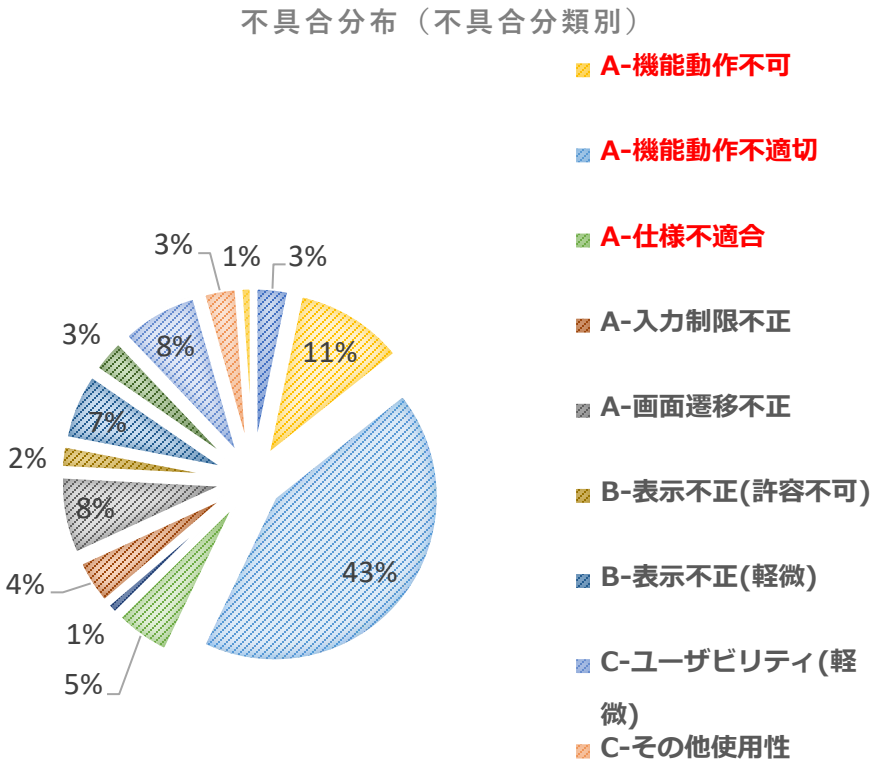
品質特性で分類化することで、客観的な品質分析が可能になる

対策①

①-１）品質状況の実施効果や分析方法を明確化

例）





他、機能分類など多角的に分析できるようにする

対策①

①-２）記述式テストと探索的テストの両手法を併用し、どのようにリスクを低減できるかを検討

◆品質に直結しそうなデメリット

・網羅性を求められない
　⇒削減した記述式テスト分の試験密度が下がる

記述式テスト／探索的テストの棲み分け範囲を決める際、
記述式テストの網羅性基準を取り決めることは可能

・テスト担当者により結果がバラつく
　⇒スキルがバラバラな担当者に任せっぱなしはバラつく

テストチャータや観点リストの作り方、探索テスト実行時の
観点共有のやり方次第で軽減は可能

対策①

①-２）記述式テストと探索的テストの両手法を併用し、どのようにリスクを低減できるかを検討

◆探索的テストのメリット

・テストケース作成工数がかからない
仕様変更が発生しても、メンテに追われない
⇒メンテ工数はテスト実行工数とトレードオフ

・不具合検出率が上がる
⇒不具合検出傾向を見ながら怪しい部分を狙って叩ける
テストケースに明記されていない狭間の部分など、
不具合の多くはテストケースの行間/周辺で発生しがち
⇒網羅性を上げるだけでは不具合流出は防げない

・不具合が出ない部分のテストは打ち切る判断ができる
⇒その分、怪しい部分のテスト工数に廻せる

対策②

②-１）過去の実績を基に、探索的テストのメリット/デメリットと不具合検出効率が高い理由を示す

◆過去の実績データ

プロジェクト名称	探索的テスト適用	テストケース数(A)	不具合検出数(B)	テスト実施工数[人日](C)	ⅰ.テストケースに対する不具合の割合(B/A)	ⅱ.不具合検出に対する必要工数[人日] (C/B)
PJ-A	あり	1,358	328	205.0	24.15%	0.6
PJ-B	あり	1,430	230	92.6	16.08%	0.4
PJ-C	なし	2,675	48	65.0	1.79%	1.4
PJ-D	なし	1,631	48	64.6	2.94%	1.3

- ⅰの結果より、探索的テストを適用したプロジェクトは不具合の摘出効果が高い（不具合検出性の向上）
- ⅱの結果より、探索的テストを適用したプロジェクトは不具合検出に対する工数が少ない（工数の削減）。

対策②

②-１）過去の実績を基に、探索的テストのメリット/デメリットと不具合検出効率が高い理由を示す

◆メリット

・テストケース作成工数がかからない
⇒仕様変更があってもメンテに追われない

・不具合検出率が高い
⇒怪しい部分を狙って叩くので必然
不具合の多くはテストの行間で発生
⇒網羅性だけで不具合流出は防げない

・不具合が出ない部分はテストしない
⇒その分、怪しい部分のテスト工数に廻す

◆デメリット

・定量化しづらい
⇒探索的テストを測るメトリクスを用意
時間軸、不具合数を活用

・探索的テストの止め時がわからない
⇒品質分析方法と判断ルールを決める

・エビデンスが残らない
⇒最低限のエビデンスレベルをルール化する

・担当者により結果がバラつく
⇒観点リストや情報共有方法次第で軽減できる

対策②

②-２）それぞれのテストについて目的を明確にする

◆記述式テスト
- 単機能の基本部分を確認する
- 業務フローのメインパスを確認する
  ⇒根幹となる基本部分の品質を担保できる

記述式テストを削るのではなく、
担当レベルを基本レベルに留める

◆探索的テスト
- テストチャータに基づいて行間、枝葉の部分を狙う
- 不具合が多い部分、不具合が潜在していそうな部分を狙う
- 業務シチュエーション、ユーザーの使い方を想定する

品質状況を分析しながら実行し、怪しい機能や観点を
狙って叩くことで、集中的に不具合検出できる

対策③

③ 試験者を問わず同じ粒度のエビデンスを残すための記録ルールを作成する

◆探索的テストの実行エビデンス

No.	機能ID	観点No	機能（自動入力）	品質特性（自動入力）	日付	名前	結果	工数(h)	不具合No.	不具合概要	テスト概要	備
1	A001	1	機能A_サブ機能A	機能適合性 - 機能完全性	2022/7/15	担当者A	OK	0.50				
2	A001	2	機能A_サブ機能A	機能適合性 - 機能正確性	2022/7/15	担当者A	NG	2.00	No.1	××××	×××	
3	A002	2	_×××	機能適合性 - 機能正確性	2022/7/15	担当者B	NG	1.00	No.2	××××	×××	
4	A003	2	_×××	機能適合性 - 機能正確性	2022/7/15	担当者C	NG	3.00				
5	A003	3	_×××	機能適合性 - 機能適切性	2022/7/15	担当者C	OK	0.50				
6	A001	3	機能A_サブ機能A	機能適合性 - 機能適切性	2022/7/15	担当者C	NG	0.50				

どのような観点で何を狙って
テストしたか、ざっくりしたテスト
目的レベルを記録

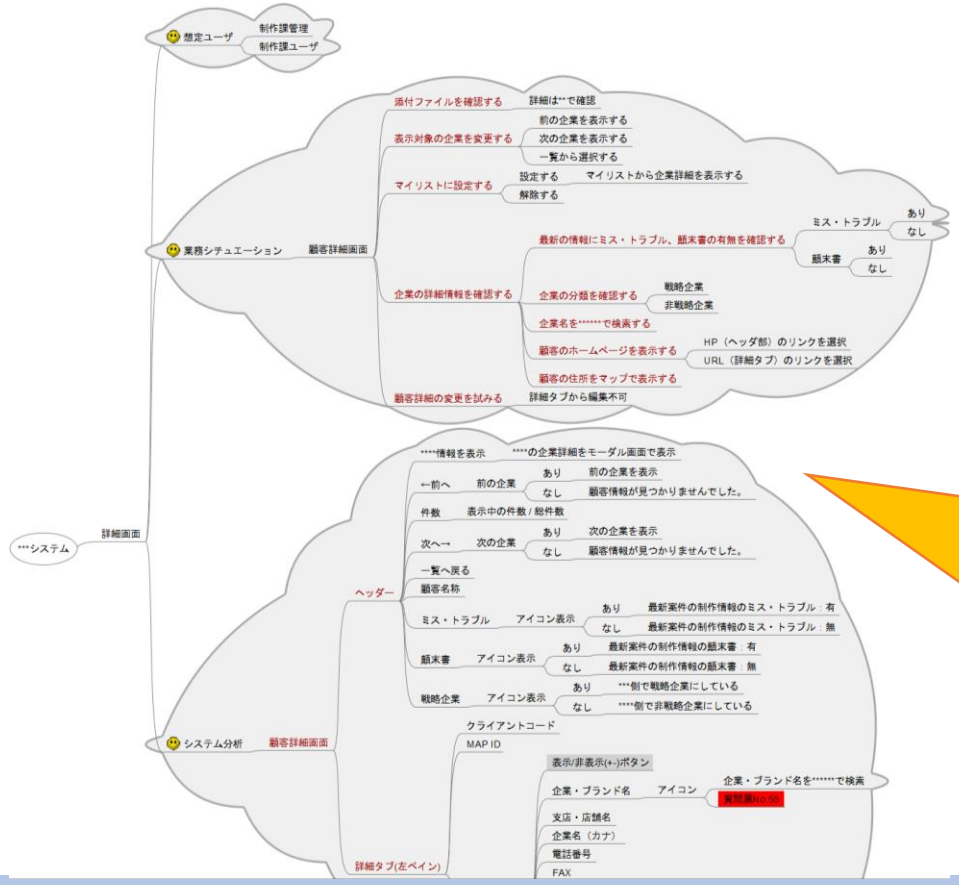
◆不具合エビデンス

No	不具合管理ステータス	起票者記入										
		起票日	起票者	テスト種別	項目番号	検証環境	品質特性（主特性）	品質特性（副特性）	不具合分類	機能分類	不具合概要	不具合詳細
1												■事象 ××××× ■テスト環境・データ 環境：××× アカウント：××× 対象データ：××× ■手順 ①××× ②××× ③××× ④××× ⑤××× ■動作状況 ××× ■期待動作

対策④

④-１）スキル依存度を低くするための分析方法とテストチャータの内容を検討する

◆どのような使われ方が想定されるかを分析しておく



さらに仕様分析の段階で、業務シチュエーションをリストアップしておく

すると、単なる機能の確認ではなく、使われ方を意識したテスト観点がイメージしやすくなる

対策④

④-１）スキル依存度を低くするための分析方法とテストチャータの内容を検討する

◆テストチャータは製品品質を軸として作成する

1	2	3	4	5	6	7	8	9	10	
機能適合性			性能効率性	使用性						
機能完全性	機能正確性	機能適切性	時間効率性	適切度認識性	習得性	運用操作性	ユーザエラー防止性	ユーザインタフェース快美性	アクセシビリティ	
要求仕様書に記載されているとおりの動作をするかを確認	入力に対して処理結果のデータが正確であるか	業務に対して的確な操作設計であるか	リクエストを開始後、そのレスポンスまでに要する時間が範囲内であるか	利用者がニーズを満たせるか判断できるか	利用方法を習得しやすいか	操作・表示の一貫性や、運用上の状態確認しやすいか	誤操作を防ぐ仕組みが実装されているか	画面デザインが見やすいか	幅広い範囲の人々が利用できるよう配慮されているか	

この粒度だとざっくりしすぎて、スキル依存を脱却できない

対策④

④-１）スキル依存度を低くするための分析方法とテストチャータの内容を検討する

◆品質特性ごとテスト対象にあわせたサブ観点を定義する

品質特性分類	主特性	副特性	説明・観点例
製品品質	機能適合性		・機能がステークホルダのニーズにどの程度うまく適合しているか
		機能完全性	・ステークホルダーが要求する機能がどれだけ備わっているか ・要求仕様書に記載されている通りに機能を実現しているか ・マニュアルに記載されている通りに機能を実現しているか
		機能正確性	・ステークホルダーが要求する機能が正確に提供されているか ・入力に対して処理結果のデータ出力が正確であるか(表示、連携機能先、外部出力) ・登録/編集/削除が正しくおこなえるか ・入力した検索内容で正しく検索されるか
		機能適切性	・ステークホルダーの目的達成を促進することができているか （単に機能を提供するだけではなく、複写機能や並べ替え機能など、目的達すい実装提供） ・業務に対して的確な操作設計であるか ・操作手順が多かったり、煩雑でないか
	性能効率性		
		時間効率性	・レスポンスタイムなどの動作速度が要求を満たしているか ・ストレスなくコンテンツを表示するか ・明らかな表示遅延等がないか
		資源効率性	・システム資源の量や種類は要求を満たしているか （システム実行中に利用されるメモリやディスクなど） ・ワークロードが高い場合でも、正常に動作するかを確認 ・複数人で同時アクセス可能か
		容量効率性	・データ量や通信容量、利用者数などのキャパシティが要求を満たしているか ・メモリ/ディスク/CPUの使用率、ネットワークトラフィックの計測結果が、許容範囲内であるか ・メモリ/ディスク/CPUの使用率が極端に増えることがないか

サブ観点をリストアップすることで、少し具体化できる

対策④

④-１）スキル依存度を低くするための分析方法とテストチャータの内容を検討する

◆探索的テストの傾向/メトリクスは対策③の書式から自動生成

◆不具合検出状況マトリクス（セッション辺りの不具合数）

観点No	1	2	3	4
	機能適合性			性能効率性
	機能完全性	機能正確性	機能適切性	時間効率性
	要求仕様書に記載されているとおりの動作をするかを確認	入力に対して処理結果のデータが正確であるか	業務に対して的確な操作設計であるか	リクエストを開始後、レスポンスまでに要するリソース時間が範囲内であるか

マトリクス上の凡例
：実施実績あり
：実施実績あり→不具合検出

※当案件では、以下の品質特性は対象外とする
・性能効率性（資源効率性、容量満足性）
・互換性
・信頼性
・セキュリティ
・保守性
・移植性

機能ID	検証対象	機能	5	0	4	0	1
A001	機能A	サブ機能A	1	0	1	0	0
A002		ｘｘｘ	1	0	1		
A003		ｘｘｘ	2	0	2		
B001	機能B	ｘｘｘ	0	0	0	0	0
B002		ｘｘｘ	0		0		
B003		ｘｘｘ	0		0		
C001	機能C	ｘｘｘ	1	0	0	0	1
C002		ｘｘｘ	0		0	0	
C003		ｘｘｘ	0		0	0	

◆セッションタイム（セッション辺りの試験時間）

観点No	1	2	3	4
	機能適合性			性能効率性
	機能完全性	機能正確性	機能適切性	時間効率性
	要求仕様書に記載されているとおりの動作をするかを確認	入力に対して処理結果のデータが正確であるか	業務に対して的確な操作設計であるか	リクエストを開始後、レスポンスまでに要するリソース時間が範囲内であるか

マトリクス上の凡例
：実施実績あり

※当案件では、以下の品質特性は対象外とする
・性能効率性（資源効率性、容量満足性）
・互換性
・信頼性
・セキュリティ
・保守性
・移植性

機能ID	検証対象	機能	21.00	6.00	10.00	2.50	2.50
A001	機能A	サブ機能A	3.50	0.50	2.00	0.50	0.50
A002		ｘｘｘ	3.00	2.00	1.00	0.00	0.00
A003		ｘｘｘ	5.00	1.00	4.00	0.00	0.00
B001	機能B	ｘｘｘ	2.00	0.50	0.50	0.50	0.50
B002		ｘｘｘ	0.50	0.00	0.50	0.00	0.00
B003		ｘｘｘ	0.50	0.00	0.50	0.00	0.00
C001	機能C	ｘｘｘ	4.50	2.00	0.50	0.50	1.50
C002		ｘｘｘ	1.00	0.00	0.50	0.50	0.00
C003		ｘｘｘ	1.00	0.00	0.50	0.50	0.00

◆不具合検出効率（セッション別「不具合検出数／時間」）

観点No	1	2	3	4
	機能適合性			性能効率性
	機能完全性	機能正確性	機能適切性	時間効率性
	要求仕様書に記載されているとおりの動作をするかを確認	入力に対して処理結果のデータが正確であるか	業務に対して的確な操作設計であるか	リクエストを開始後、レスポンスまでに要するリソース時間が範囲内であるか

マトリクス上の凡例
：実施実績あり

※当案件では、以下の品質特性は対象外とする
・性能効率性（資源効率性、容量満足性）
・互換性
・信頼性
・セキュリティ
・保守性
・移植性

機能ID	検証対象	機能	2.67	0.00	2.00	0.00	0.67
A001	機能A	サブ機能A	0.50	0.00	0.50	0.00	0.00
A002		ｘｘｘ	1.00	0.00	1.00		
A003		ｘｘｘ	0.50	0.00	0.50		
B001	機能B	ｘｘｘ	0.00	0.00	0.00	0.00	0.00
B002		ｘｘｘ	0.00		0.00		
B003		ｘｘｘ	0.00		0.00		
C001	機能C	ｘｘｘ	0.67	0.00	0.00	0.00	0.67
C002		ｘｘｘ	0.00		0.00	0.00	
C003		ｘｘｘ	0.00		0.00	0.00	

不具合の分布状況
(ヒートマップで可視化)

かけた時間

検出効率

対策④

④-２）テスト実行時に不具合傾向や品質状況を踏まえた重要観点を共有する仕組みを検討する

◆日々2回(朝/午後一に30分程度)の情報共有Meetingを実施

- 進行状況
- 不具合検出状況
- 不具合が多い観点、少ない観点の共有
- 次の探索ポイントを議論

短いスパンで状況の認識合わせを実施。怪しい観点の共有、探索ポイントのジャッジ、観点変更をスピーディに展開。

フォーマットとルールを決めて、あとはよろしく、では担当者任せ（スキル依存）なやり方と大した変わらない
如何にテスト担当者間で発想を分かち合うかがポイント

対策④

④-２）テスト実行時に不具合傾向や品質状況を踏まえた重要観点を共有する仕組みを検討する

情報共有Meeting方針例

- **Daily 2回の打合せで、リアルタイムなテスト状況／品質状況を全員で共有する**
- **打合せでは、品質状況を踏まえ探索箇所を決め、柔軟にテスト方針を変更する**
- **記述式テストの担当者も打合せに入ることで、多角的な視点で方針を検討できる**

探索テストの1日の流れ		
9:00 12:00	朝ミーティング	前日のテスト状況／品質状況を共有し、その日のテスト方針、分担を決める
13:00 16:00 18:00	昼ミーティング 進捗集計	午前中のテスト状況／品質状況を共有し、テスト方針や分担の変更要否を確認する

打合せでのフィードバック情報例	探索観点、探索箇所の決め方例
・不具合傾向から、同じ観点で他の機能を確認したい ・不具合の検出数は多くないが、〇〇の観点も怪しい印象がある ・〇〇の機能は重要性が高いのでもう少し深掘りして確認したい	・機能〇〇を重視して確認する ・△△やXXの観点で他機能も確認する ・機能〇〇と類似した機能の経験があるAさんをアサインする ・△△やXXの観点で不具合を検出しているBさんに同じ観点で他機能をアサインする

探索①～④に共通すること

それぞれの対策を検討して提示するだけでなく、準備段階からステークホルダーへの説明、段階的議論、認識合わせを実施。

疑問点や課題の解消など、密なコミュニケーションを積み重ね、具体的な進め方や、それによる効果の説明などを一つずつ丁寧に積み重ねることで、段階的にメリットの理解、課題の解決など。ステークホルダーの不安を緩和していくことができた。

なお、テスト実施の段階においては、ステークホルダーにもテスト実施や分析作業に参加してもらい、効果的に不具合検出を進めるアプローチを肌で感じてもらうことで、品質面の不安が低減されることを理解してもらうことができた。

✓はじめに・背景

✓課題

✓課題解決に向けた取り組み

✓得られた結果

✓今後に向けた課題

実績

得られた結果

◆テスト実績
- 記述式テスト、探索的テスト同等レベルの工数を投入
- 探索的テストの不具合検出効率は記述式テストの2.8倍

◆テスト終了判定の結果
- 探索的テストは期間を通じてコンスタントに不具合を検出
  ⇒不具合がありそうな部分を叩くので想定通り
  信頼度成長曲線は右肩あがりになるが、致命的不具合の終息傾向を以て、総合テスト終了と判断

◆不具合傾向
- 記述式テスト：機能適合性が100%
- 探索的テスト：機能適合性が50%、使用性が50%

探索的テストは、効率よく不具合検出できる

効果

得られた結果

① 探索的テストのメリットとデメリットを認識いただけた
そのうえで、記述式テスト／探索的テストの併用により、
大きな品質リスクが発生しないことを示せた
加えて、**テストチームA（記述式テストのみ）で検出できなかった仕様に影響がある不具合を検出**することができた

② 従来と同水準のコスト増で実現可能であることを示せた

③ エビデンス粒度による手戻り、バグピンポン、ロス工数が
発生しないことを示せた

④ 担当者のスキル依存度が軽減する仕組みにより、
スキル差分由来の大きな問題は発生しないことを示せた

チーム間の情報公開制限により、実績値の比較はできず
（流出有無は継続的なウォッチが必要）

副次的効果

とにかく、コミュニケーションが円滑になった印象。

単なる報告だけではなく、全員参加の議論により、技術面を含めたスキルアップにつながった。

さらにコミュニケーション量に比例し、日常的なコミュニケーションも良好に。
複数拠点、在宅勤務を問わず、すぐにWeb会議するシーンも。

信頼関係も構築でき、今後の業務にも期待できる。（と思う）

テストエンジニアは、コミュニケーション力が重要
それを活かせる風土も重要

✓はじめに・背景
✓課題
✓課題解決に向けた取り組み
✓得られた結果
✓今後に向けた課題

課題

◆メトリクスについて
- 今回のような探索的テスト併用時、記述式テストの試験密度(↓)、不具合密度(↑)に影響が生じる
  その場合の試験密度の妥当性について検討が必要
- 探索的テストのメトリクスが十分であるか検討が必要

◆探索的テスト導入時のテストプロセス
- 担当者依存のテストプロセスにならないよう、今回の実績を踏まえたガイドライン化、標準フォーマットを定義する必要がある

テストプロセスに基づいた運用実績を積み上げ、
検証を繰り返すことが必要

参考文献

SQiPシンポジウム発表論文

短納期型開発プロジェクトのためのテスト手法「FaRSeT（ Flexible and Rapid Software Test）」の適用と効果

https://www.juse.jp/sqip/symposium/archive/2018/day1/files/A2-2_ronbun.pdf

JaSST'15 Hokkaido基調講演「探索的テスト - アジャイル時代の効率的なテストを考える」

https://www.jasst.jp/symposium/jasst15hokkaido/pdf/S1.pdf

JaSST'20 Hokkaido基調講演

「なぜ大規模SIerで探索的テストを推進しているのか？～NTTデータが目指すソフトウェアテストの世界～」

https://jasst.jp/symposium/jasst20hokkaido/pdf/S2-1.pdf

Microsoft Doc「Exploratory Software Testing (探索的ソフトウェア テスト)」

https://docs.microsoft.com/ja-jp/previous-versions/jj620911(v=vs.120)?redirectedfrom=MSDN

仙台ソフトウェアテスト勉強会ブログ「ソフトウェアテスト勉強会 探索的テスト入門②」

https://ameblo.jp/sendai-test/entry-11919265075.html

はてなブログ「シナリオベースの探索的テストとは何か」

https://www.kzsuzuki.com/entry/2014/03/26/091300

同じような課題をお持ちの方を含め、
一つでも参考になる点があれば幸いです。

探索的テストをよりロジカルに、
さらに納得性の高い説明と効果的な活用ができるよう
引き続き、模索していきたいと思います。

ご清聴、ありがとうございました。