

JaSST' 14 Tokyo
セッション C4-2
17:35~18:05(30分)

探索的テストを活用したシステム開発手法の提案

2014/3/7
株式会社NTTデータ
技術開発本部 プロアクティブ・テストイングCOE
熊川 一平

NTT DATA

- **熊川 一平 (くまがわ いっぺい)**

- **所属**

- **株式会社NTTデータ 技術開発本部**
プロジェクトマネジメント・イノベーションセンター
プロアクティブ・テストイングCOE 主任

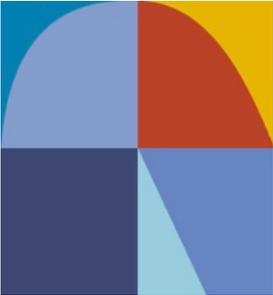
- テスト・品質保証に関する技術支援、研究開発
主としてテスト自動実行ツールの技術調査、普及活動に従事

- **活動**

- **執筆・講演歴**

- ITPro(日経BP社)
実践！テスト自動化の勘所～実装・実行の自動化
「キーワード駆動テストによるGUIテストの効率化」
<http://itpro.nikkeibp.co.jp/article/COLUMN/20121023/431821/>
- Micro Focus テストソリューション カンファレンス
テストツールを使ったプロセス改善のコツ、NTTデータが教えます
<http://special.nikkeibp.co.jp/ts/article/ac0g/147403/>
- Borlandソリューションカンファレンス
テスト自動化を成功させるには？ ～NTTデータの事例～
<http://special.nikkeibp.co.jp/ts/article/acab/158682/>

1. 探索的テストとは？
2. SI事業における探索的テスト
3. 探索的テスト活用の課題
4. 探索的テストの活用方法について



探索的テストとは？

テスト対象製品の学習、テスト設計、テストの実行を同時並行に行う技法。テスト担当者は製品を探索しながらテストの設計と実行を進めていく。



テストケースを事前に用意せず、テストをしながら怪しそうなところを探してテストしよう。

※知識ゼロから学ぶソフトウェアテスト【改訂版】、高橋寿一、翔泳社より

〇〇開発にて



テスト実施者

このテストの結果って
こんな感じで良かったん
でしたっけ？



機能設計者

...大丈夫。
でも、この機能って全角
の入力できたかなあ...？



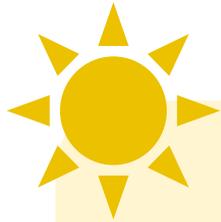
テスト実施者

心配なんで全角入力した
らバグが見つかったちゃい
ました。



機能設計者

やっぱりそうかー。
この部分の仕様、揉めた
もんなあ。



- テストケースを作成する従来のテストに比べバグの抽出効率が良い
- 使いやすさ、魅力といった評価しづらいテストを実行できる
- 弱点をピンポイントに攻めるテストができる



- 探索的テストだけではテストを完了できない
- テスト担当者にスキルが要求される
- 見つけたバグの再現性が低い



SI事業における探索的テスト

	利用者	責任	期間
SI	他人	大きい	長い
自社開発	自分/他人	大きい	長い
日曜プログラマ	自分	小さい	短い

SI事業者のテストでは テストに可監査性・客観性が求められる

テスト/バグ密度

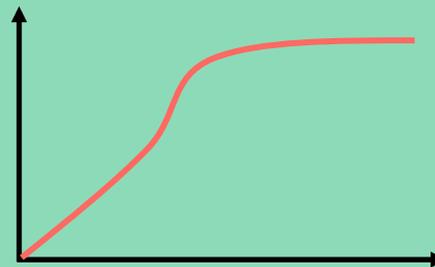
上限



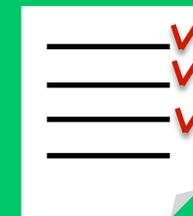
下限



信頼度成長曲線



テストケース表



公式なテスト

非公式なテスト

こっち



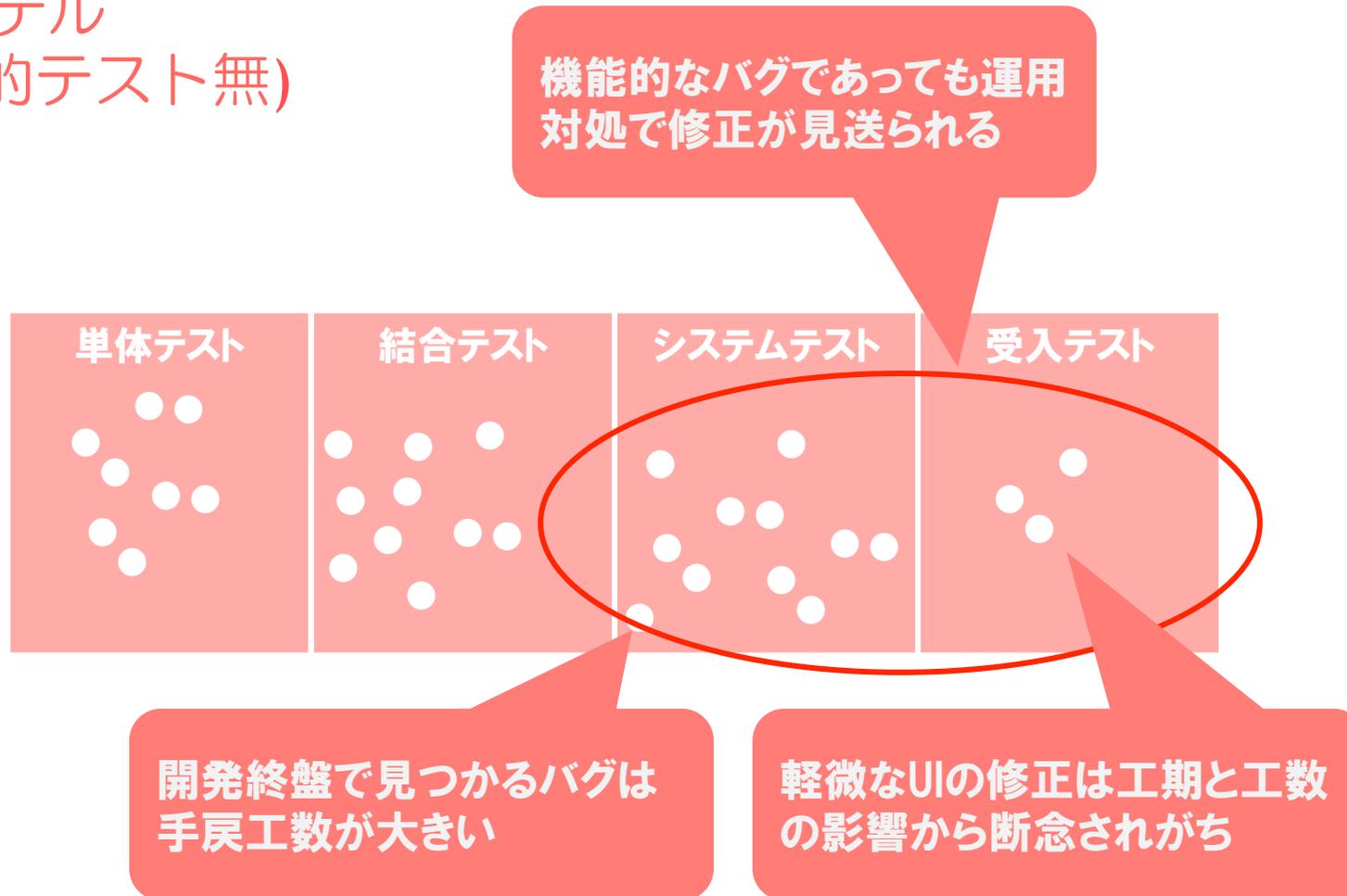
- 事前にテストケースを作成し、要件を順番通りに検証する
- 再現性／客観性／可監査性に**優れる**
- テスト1回あたりに必要な工数が**大きい**

- テストケースは事前に作成せず、検証対象を特に定めない
- 再現性／客観性／可監査性に**乏しい**
- テスト1回あたりに必要な工数が**小さい**



SI事業における探索的テスト活用案

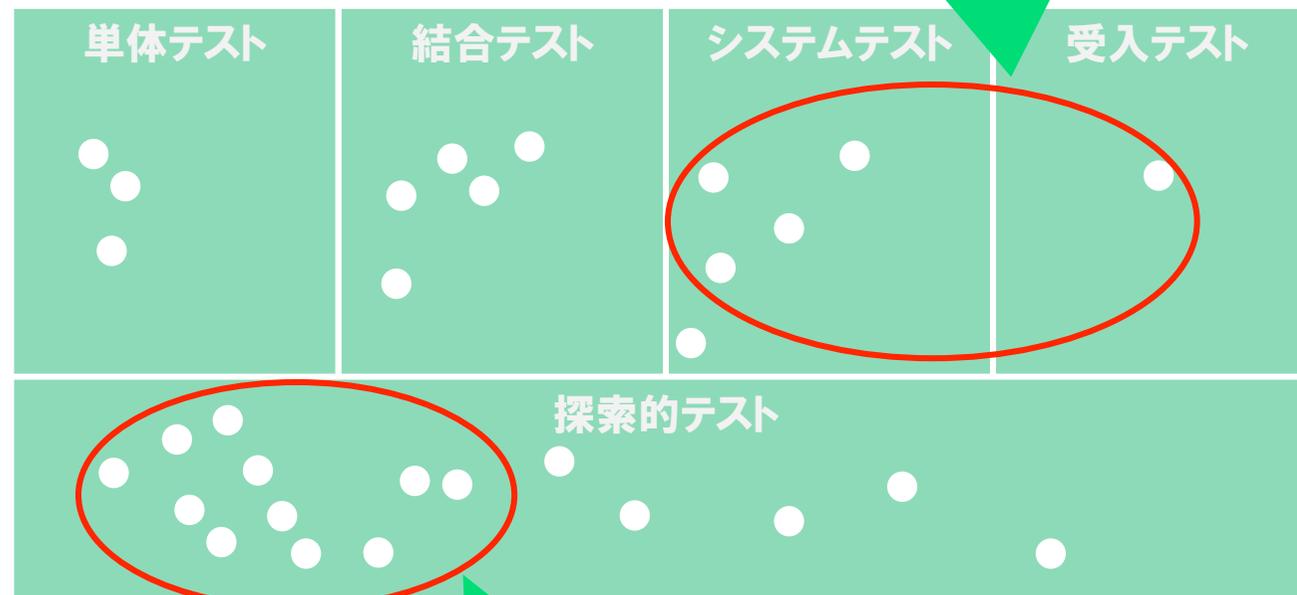
V字モデル
(探索的テスト無)



● = バグ

探索的テストを
並行実施した場合

全てのバグが探索的テストで
見つかるわけではない



終盤に見つかるはずだったバグの一部が序盤で摘出される

手戻工数が少なく、UIの修正が行いやすく、運用対処にも逃げないで済む

● = バグ

探索的テストを
追加した場合

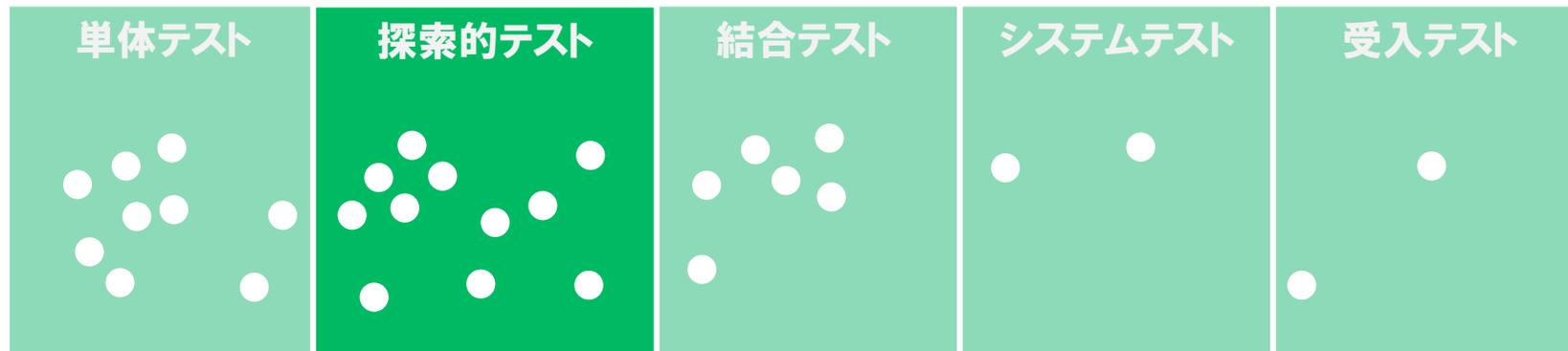


スモークテスト/ビッグバンテストに近い

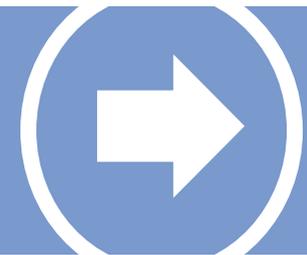
結合テスト初期/前に実施することでバグの発見を早める

● = バグ

バグは早く見つければ見つけるほど
QCDに良い影響が出る。

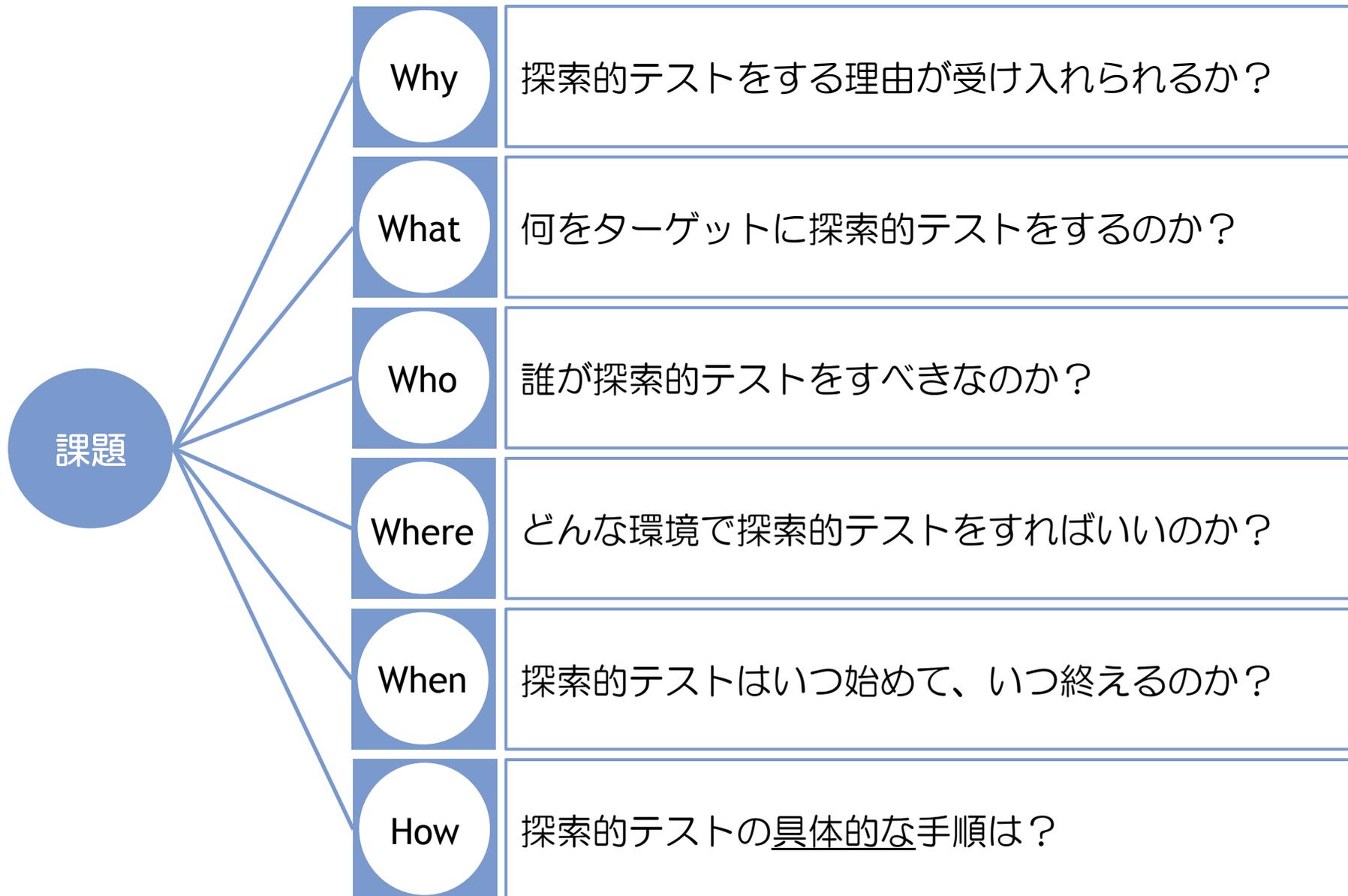


バグ抽出効率の良い探索的テストを活用して、バグ抽出工程のフロントローディングを目指す。





探索的テスト活用の課題



Why

探索的テストをする理由が（お客様に）受け入れられるか？

まずは内部のQCD向上施策として実施しよう。

Ans

探索的テストは **非公式なテスト**

現段階で見積もりに探索的テストを入れたり
探索的テストを理由に品質状況を説明することは難しい。

あなたがお客様だったら以下の報告は受け入れられますか？

- このプログラムはAさんが1週間探索的テストをしたからバッチリです

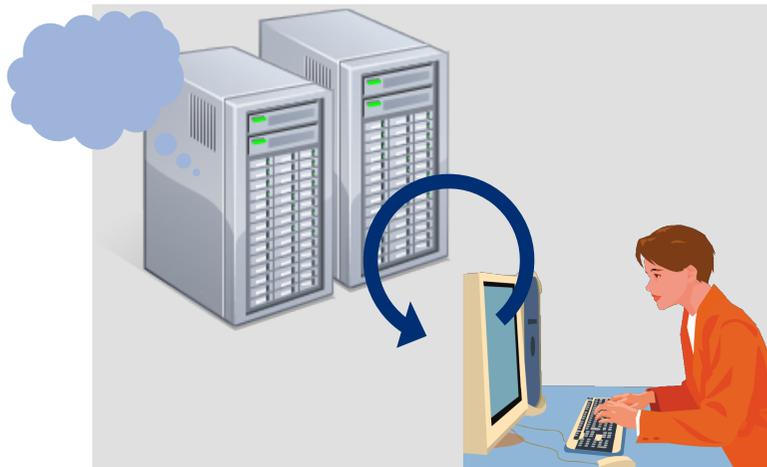
あくまでも **非公式なテスト** として始めることをおすすめします。

What

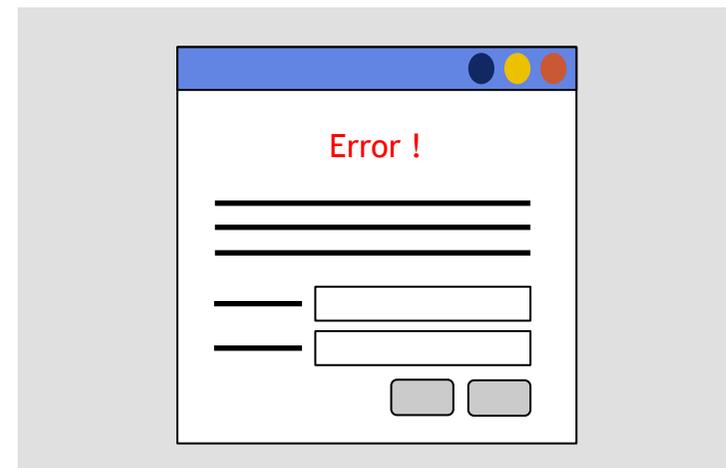
何をターゲットに探索的テストをするのか？

異常が見た目に現れやすいソフトウェアがおすすめ。

Ans



バッチ系や基盤系のシステムは、テスト結果やソフトウェアの状態が確認しづらく、探索しにくいことがある。



UIのあるシステムの方が、実行結果や状態を確認しやすく、次のテスト観点を探索しやすい

Who

誰が探索的テストをすべきなのか？

バグの探索が出来るように業務ノウハウがある人物が望ましい。探索の偏りを避けるため、テスト経験はバリエーションを取り揃えたほうがいい。

Ans

事例

開発経験の浅い新人と、開発経験の多いベテランで探索的テストを実施して結果を比較。（探索的テストの実施方法は後述）



Where

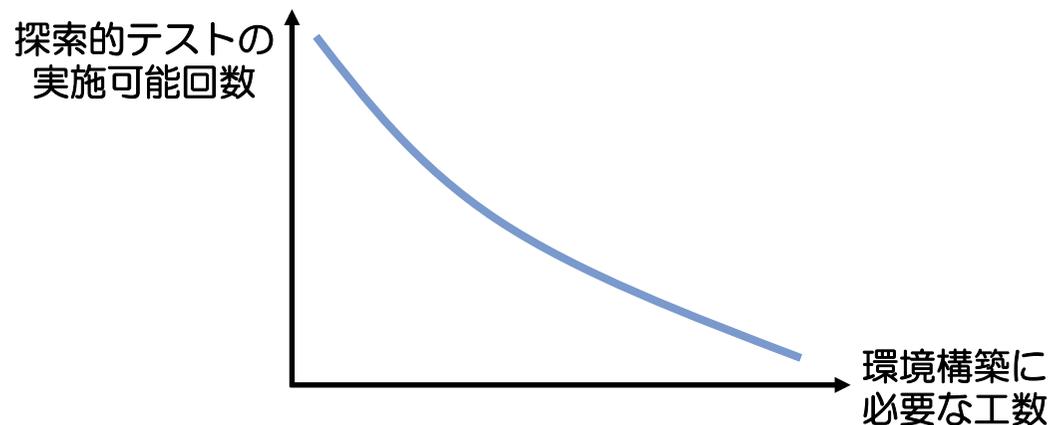
どんな環境で探索的テストをすればいいのか？

見つけたいバグによるが、構築に手間の掛からない環境が望ましい。

Ans

探索的テストはUIなどの使用性を除く、非機能要件のテストにあまり向かない(※)
また、環境構築に手間がかかると、その分実行できるテスト回数が減少してしまうので、あまり構築に手間の掛からない開発環境などが良いと思われる。

イメージ図



※知識ゼロから学ぶソフトウェアテスト【改訂版】、高橋寿一、翔泳社より

When

探索的テストはいつ始めて、いつ終わるのか？

継続課題。信頼度成長曲線との相性がいいと思われる。

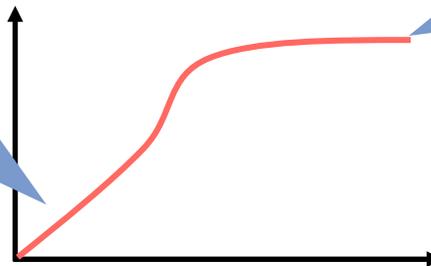
Ans

公式なテストにおける信頼度成長曲線の扱いは難しい。

（テストケースの消化速度、テスト順序の入れ替えによる信頼度のねつ造など）
探索的テストにおいては信頼度成長曲線との相性が良いと考えられる。

よくある曲線

開始時期は任意。
だが、単体テストが終わっていないと
フィードバックの得やすいUIからのテスト
実行が難しいケースが多い。



探索的テストにおいては
・探索するところが無くなってきた
・探索してもバグが出なくなってきた
を意味する。

How

探索的テストの具体的な手順は？

次頁参照。

Ans

システムの適切な振る舞いについての仮説（メンタルモデル）を作る。

仮説を反証するようなテストを1つまたは複数個、設計する。

仮説に反する結果を評価する。

仮説が証明されるか反証されるまでこの手順を繰り返す。

一般的な探索的テストの実施手順は左の通り。でもこれだと具体的に何をすればいいのかわからない。⇒ 属人性が極めて高い。

スキルも経験も異なる様々な人的リソースを活用しなければならないSI事業では、もう少し具体的な手順が必要。

私が実際に探索的テストを実施した際の手順を紹介します。



1：テスト環境の構築

任意の環境に、フィードバックの得やすいテスト環境を構築する。

2：ユーティリティの準備

テストの環境構築や、事前条件を整えるために必要な作業を補助するユーティリティを用意する。

3：参照ドキュメントの準備

探索的テストの指針や、探索補助のためのドキュメント類を用意する。

4：スモークテスト

スモークテストとして正常系代表テストを実施する。このテストが通過するまで以降のプロセスは実施しない。

5：バグの探索

スモークテストor前回のテストを通じてバグの潜んでいそうな箇所を観察する。抽出した観点をマインドマップに記述する。

6：テスト実行

マインドマップからテストする観点を選択し、テストを実行する。結果は簡単に記録・報告する。テスト実施後、「5：バグの探索」に戻る。

7：対処と振り返り

開発者はバグが報告されるたび、修正要否を判断して対処を行う。修正後、回帰テストと再テストを行う。

8：計画反映

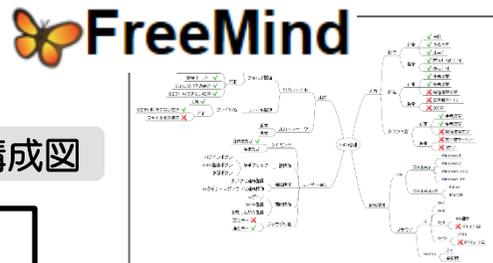
探索的テストの結果から品質状況を分析し、今後のテスト計画を見直す。

3：参照ドキュメントの準備

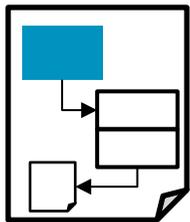
探索的テストの指針や、探索補助のためのドキュメント類を用意する。

どこを攻めるか

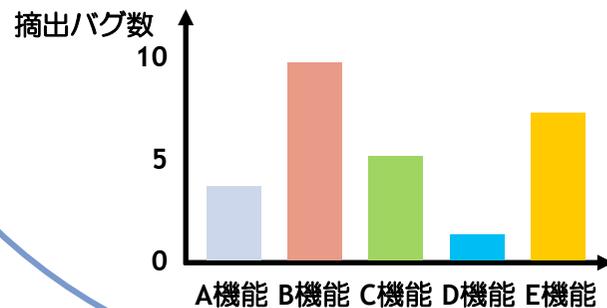
マインドマップ



システム構成図



前工程の品質状況

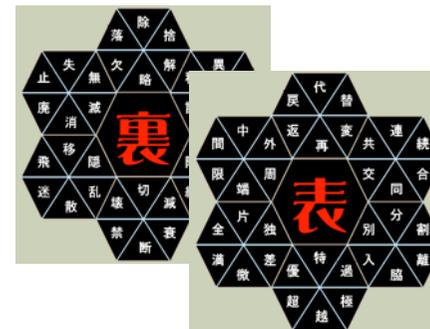


チェックリスト



どう攻めるか

意地悪漢字(※)



過去のトラブル



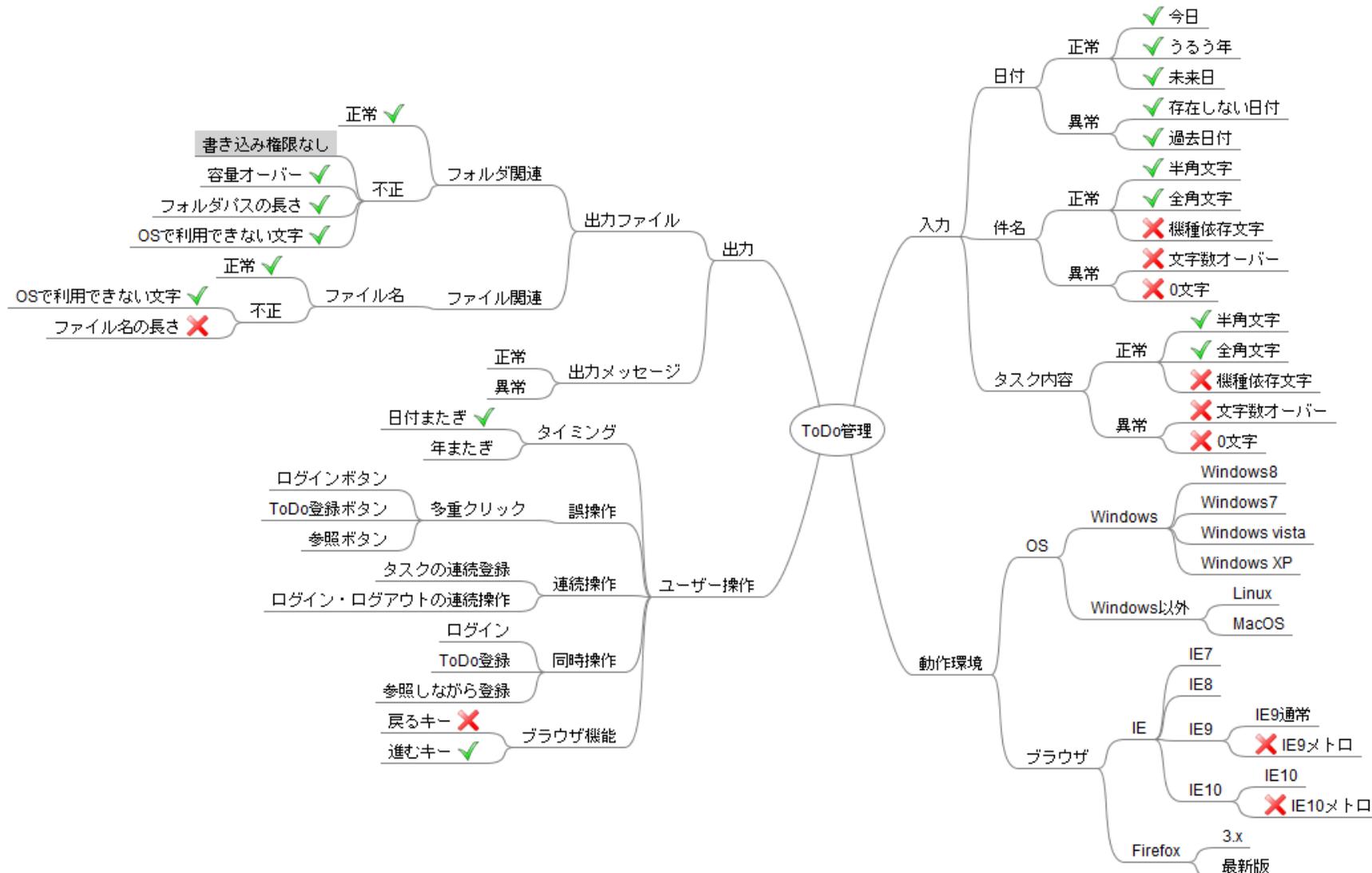
エラー推測観点集

うるう日
Null
全角/半角
ゼロ除算
...

※ 意地悪漢字” JaSST2010 Hokkaido, 鈴木 三紀夫様

5：バグの探索

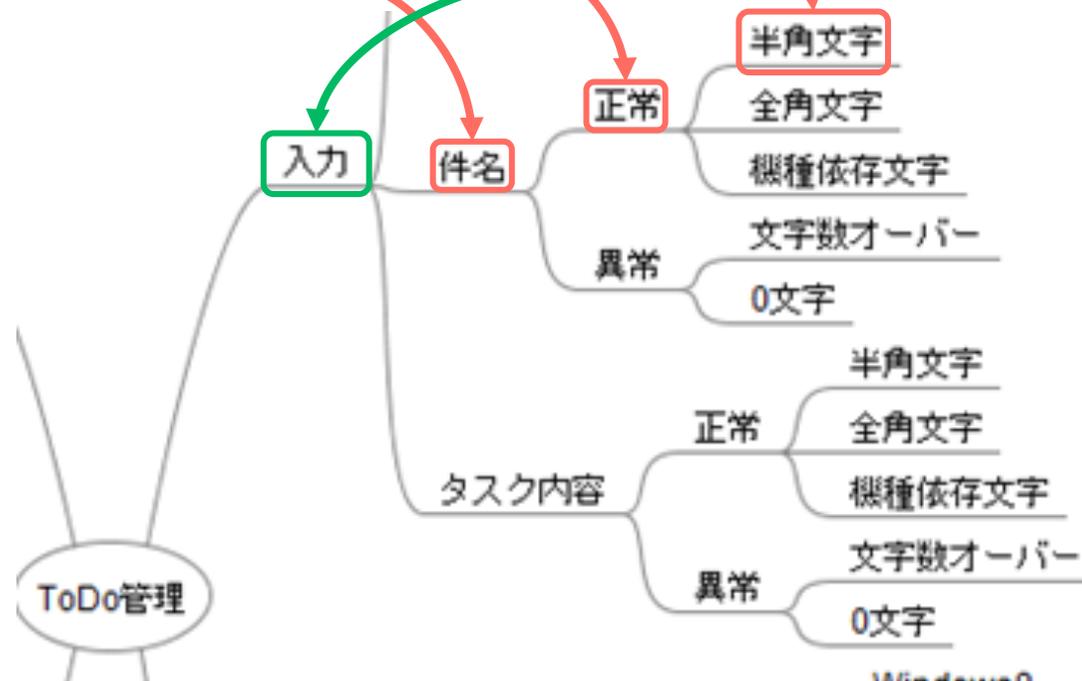
スモークテストor前回のテストを通じてバグの潜んでいそうな箇所を観察する。抽出した観点をマインドマップに記述する。



5：バグの探索

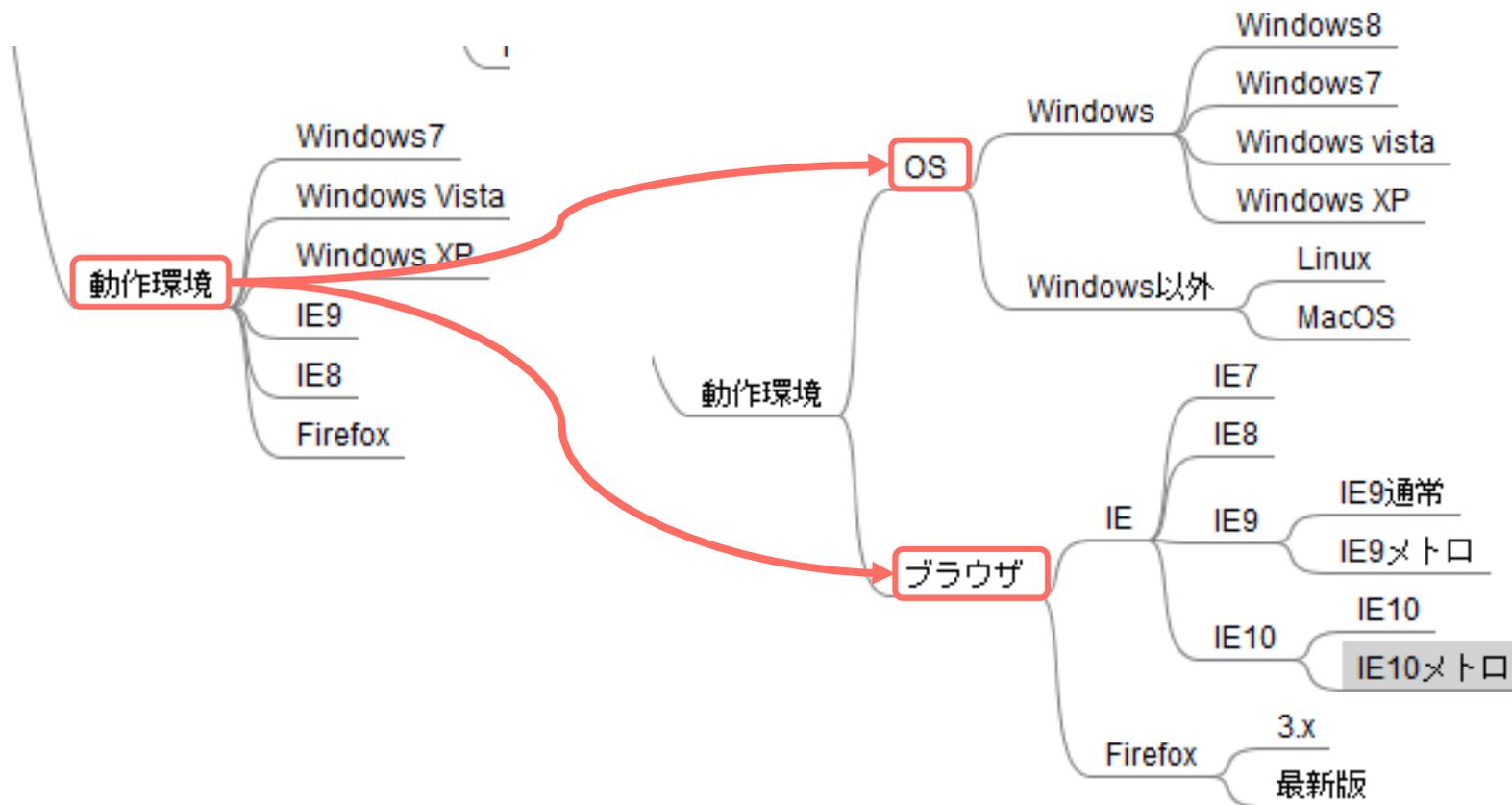
スモークテストor前回のテストを通じてバグの潜んでいそうな箇所を観察する。抽出した観点をマインドマップに記述する。

ToDo管理システムの **件名** に **正常** な **半角文字** を **入力する** テストをしよう。



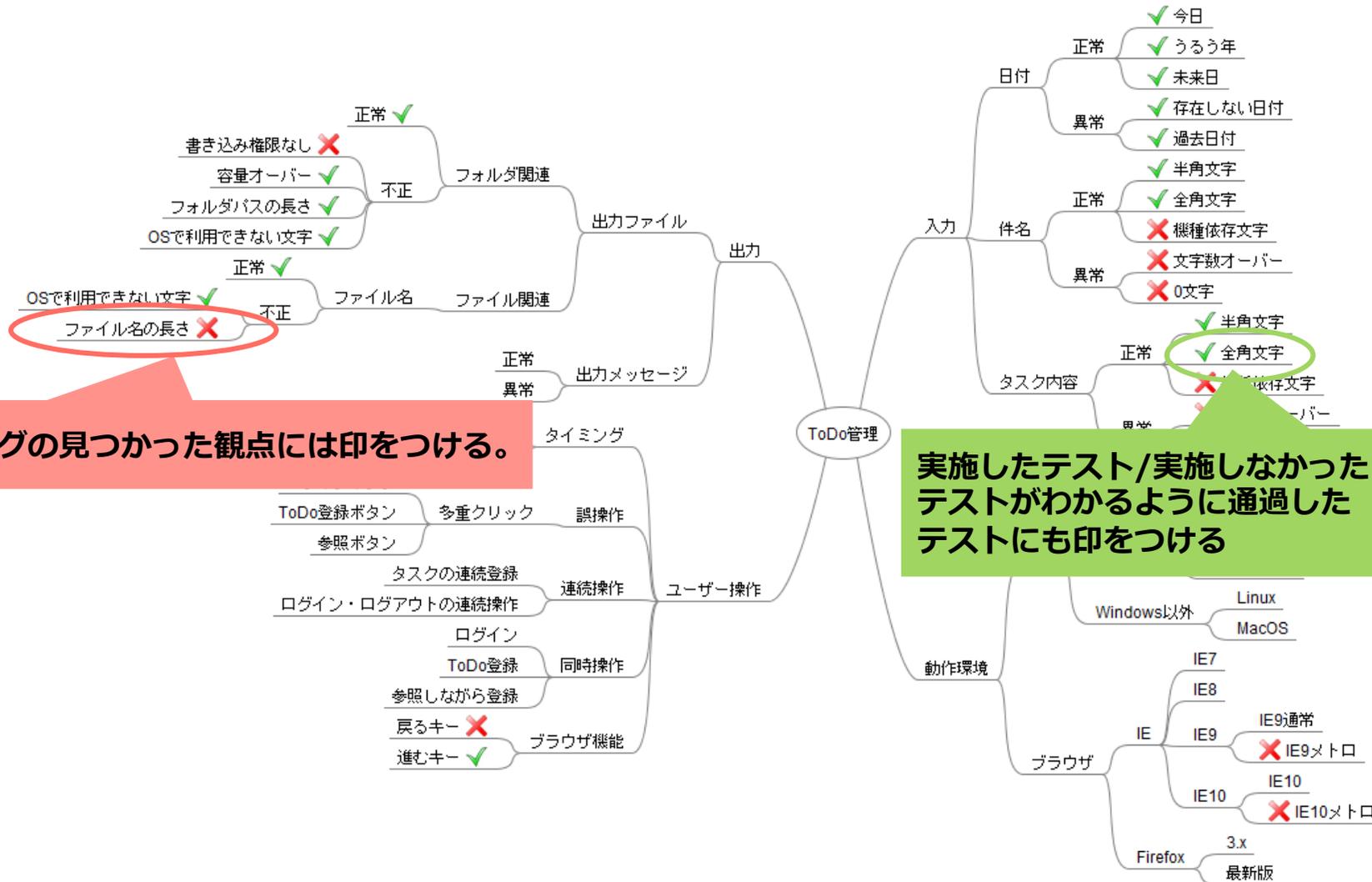
5：バグの探索

スモークテストor前回のテストを通じてバグの潜んでいそうな箇所を観察する。抽出した観点をマインドマップに記述する。



6：テスト実行

マインドマップからテストする観点を選択し、テストを実行する。
結果は簡単に記録・報告する。テスト実施後、「5：バグの探索」に戻る。

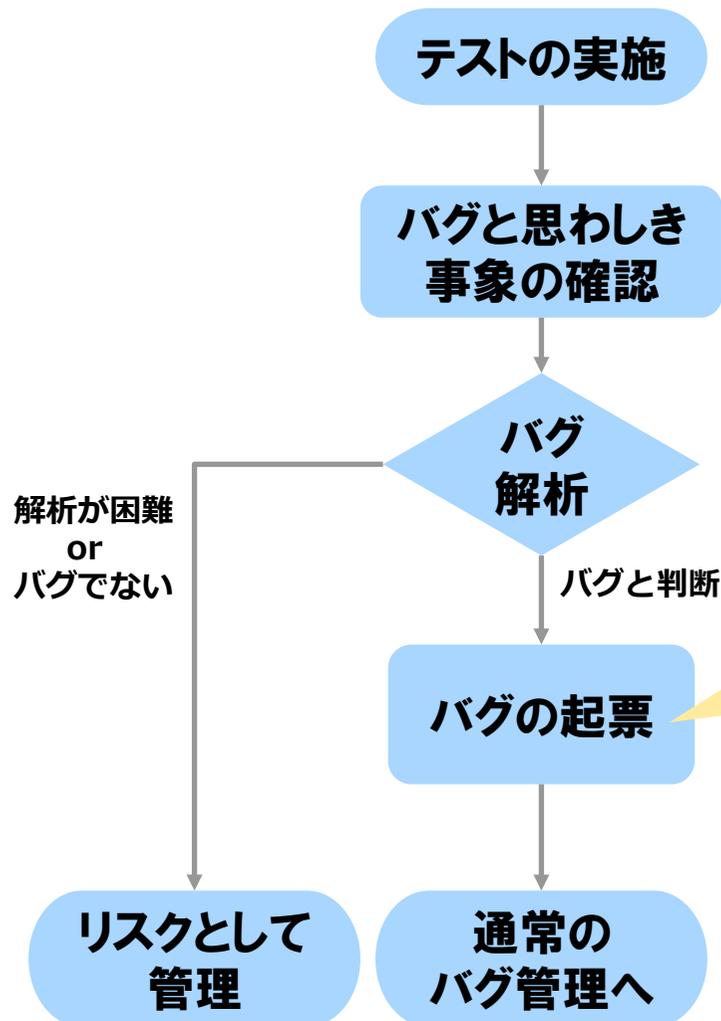


バグの見つかった観点には印をつける。

実施したテスト/実施しなかったテストがわかるように通過したテストにも印をつける

7：対処と振り返り

開発者はバグが報告されるたび、修正要否を判断して対処を行う。
修正後、回帰テストと再テストを行う。



- 探索的テストは非公式なテストなので、全てのバグに完全な対応をしようとは考えない。
- 解析が困難な場合や、バグでないと判断される場合は、以降の工程に備えてリスクとして管理する。
- バグと判断できた場合は、本来どのテストフェーズで摘出すべきだったバグかを確認する。
- 本来摘出すべきフェーズでの品質評価時に、探索的テストで摘出したバグの件数をカウントする場合/しない場合の評価を行う。

社内開発において、探索的テストの早期実施によるバグ抽出工程の前倒しを検証

結果

- ベテランと新人の2名による探索的テストで、**12件のバグを発見**
- バグ抽出工程が前倒しになることで、回帰テスト工数が削減され、**10%の開発工数が削減**。（テストだけでみると**25%の削減**）
- また、公式なテストだけではシステムテストでないと発見できなかった**バグを早期に見つけることで修正を実施できた**。
（システムテストで見つかっていたら運用対処であった。）

気付き

- テスター2名のマインドマップを比べると以下のような特性があった。
 - ベテランは経験に基づいてバグを探索する。
過去のトラブル事例や、これまでの品質傾向を重視する。
 - 新人は観点リストなどに基づいてバグを探索する。
エラー推測観点や、チェックリスト、また異常操作を重視する。
- 公式なテストに追加する形で、非公式なテストをすることに対して否定的な意見や拒否感があった。



探索的テストの活用方法について

SI事業でも探索的テストは活用できる

- 可監査性・客観性が求められるSI事業では、探索的テストだけでテストを完遂することはできない。公式なテストと非公式なテストをうまく組み合わせることが重要。（並行実施、先行実施、弱点強化など）
- テスターは業務ノウハウがある人物を中心に、様々な属性を持つ人物をアサインするのが良い。ただ、SI事業では限りあるリソースの中、スキル・経験・文化も異なるメンバーで開発するため、一定の手順を定めたほうが良い。特に見つかったバグの対処や管理方法などは重要。
- マインドマップや観点リストなど、テスターに気付きを与える様々なアイテムを用意しておく、テスト経験のない人物であっても探索がしやすくなる。

今後に向けた課題

- 探索的テストの終了条件は明確に定義できませんでした。実施する際は他のスケジュールの兼ね合いなどから、関係者で終了日や実施日数を決めると良い。
- 非公式なテストに対する拒否感は必ずある。事前に十分に説明をおこなったり事例を重ねていく必要がある。

活用ポイント

工数削減したい

ここで実施



- 結合テストの前に探索的テストでバグを出すことで、後工程のテストで見つけるよりも少ない工数でバグを修正できる。
- 運用対応やUI改善が実施でき、顧客満足度が向上する。
- UT以降継続して実施しても良い。

商用バグのすり抜けを減らしたい

ここで実施



- STまですり抜けてしまったバグを探索的テストで抽出できる。
- 設計書からテストを実施しないため、設計書からテストケースを作成する従来のテストとは違う観点のテストができる。

効果的な受入テストがしたい

ここで実施



- 受入テストの時点で、後工程で見つかるバグが抽出できる。
- 請負ベンダと違う観点で効果的な受入テストができる。

※上記はITまで委託した場合

活用ポイント

工数削減したい

商用バグのすり抜けを減らしたい

ここで実施

探索的テストの活用事例は
どんどん共有しましょう！

•

• 従来のテストと
異なる視点
が得られる。

• UT以降継続して実施
しても良い。

従来のテストと
異なる視点のテストが
できる。

異なる視点
で効果的な受入テスト
ができる。

※上記はITまで委託した場合



NTT DATA

変える力を、ともに生み出す。