

テストのリバースモデリングの事例



智美塾2号生
西 康晴(電気通信大学)

© NISHI, Yasuharu

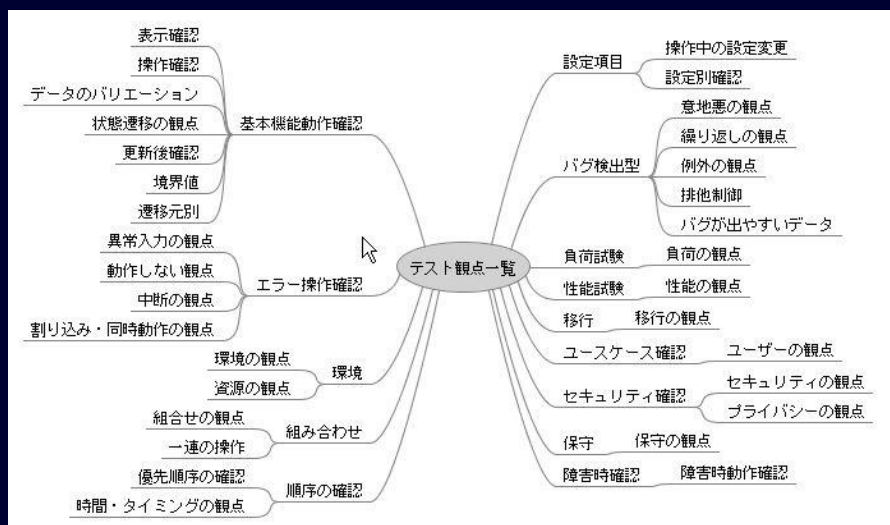
村上さんが改善後に気付いた問題点

- ① **テストでバグを出し尽くす**
→ テスト観点一覧などを用意して検討したので、今まで漏れていたバグが検出できた
- ② **レビュー容易性、保守性を高める**
→ テスト観点が軸になってしまい、全体の構成がわかりづらくなってしまった
- ③ **テストケースの作成を分担する**
→ テスト仕様を書いていない人が、現状のテスト仕様からテストケースを作成するのは困難
- ④ **テスト自動化も見据えて**
→ 遷移図、パラメータマトリクスなどを作成したが、わかりづらく今のままでは扱いづらそう。
- ⑤ **画面テスト**
→ 画面パーツという括りで分けたため、画面要素、画面毎の機能がわかりづらくなってしまった
- ⑥ **遷移テスト**
→ 機能結合テストになる想定だったので、機能ではなく、案件内の遷移にしてしまったので、機能を組み合わせってしまった
- ⑦ **機能結合テスト**
→ 検討が大変な割にはバグがあまり見つからなかった

あれれ？

- ①テストでバグを出し尽くす
→ テスト観点一覧などを用意して検討したので、今まで漏れていたバグが検出できた
- ②レビュー容易性、保守性を高める
→ テスト観点が軸になってしまい、全体の構成がわかりづらくなってしまった
- ③テストケースの作成を分担する
→ テスト仕様を書いていない人が、現状のテスト仕様からテストケースを作成するのは困難
- ④テスト自動化も見据えて
→ 遷移図、パラメータマトリクスなどを作成したが、わかりづらく今のままでは扱いづらそう。
- ⑤画面テスト
→ 画面パーツという括りで分けたため、画面要素、画面毎の機能がわかりづらくなってしまった
- ⑥遷移テスト
→ 機能結合テストになる想定だったので、機能ではなく、案件内の遷移にしてしまったので、機能を組み合わせてしまった
- ⑦機能結合テスト
→ 検討が大変な割にはバグがあまり見つからなかった

なんかそれっぽく見えます



なんかそれっぽく見えます

機能名	画面名	項目種類	観点	項目名	補足事項	パラメータ検討
アカウント新規登録	会員情報入力画面	画面レイアウト	ユーザーの観点	画面表示確認	パスワードはマスクされているか	-
			プライバシーの観点
		

しかし、本当にテストしたいことはなんでしょう？

機能名	画面名	項目種類	観点	項目名	補足事項	パラメータ検討
アカウント新規登録	会員情報入力画面	画面レイアウト	ユーザーの観点	画面表示確認	パスワードはマスクされているか	-
			プライバシーの観点
		

本当にテストしたいことが
「補足事項」の欄に書いてある

本当にテストしたいことが整理されていますか？

機能名	画面名	項目種類	観点	項目名	補足事項	パラメータ検討
アカウント新規登録	会員情報入力画面	画面レイアウト	ユーザーの観点	画面表示確認	パスワードはマスクされているか	-
			プライバシーの観点
		

左右の欄の親子関係が整理されていない

どの欄に何を書くべきかを分かっていますか？

■ × ■ = ■

機能名	画面名	項目種類	観点	項目名	補足事項	パラメータ検討
アカウント新規登録	会員情報入力画面	画面レイアウト	ユーザーの観点	画面表示確認	パスワードはマスクされているか	-
			プライバシーの観点
		

役割の異なる3つのブロックが1つの表で同時に分解されている

問題は何か？

- 本当にテストしたいことが「補足事項」に書いてある
 - 本当にテストしたいことは、もっと基本的な欄に書き、そのように扱う
 - 補足事項や備考には可能な限り何も書かないようにする
- 左右の欄の親子関係が整理されていない
 - 親子関係をかなりしっかり整理する
- 役割の異なる3つのブロックが1つの表で同時に分解されている
 - それぞれのブロックの役割をきちんと検討する
 - 役割の異なるブロックは別の文書に分けて、それぞれ適切に分解する

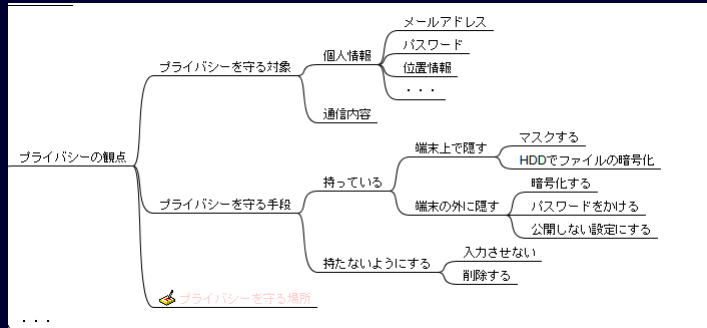
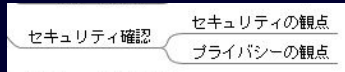
問題をきちんと見れば
自然に適切な改善案が浮かんでくる

本当にテストしたいことを具体的なテストケースから探す

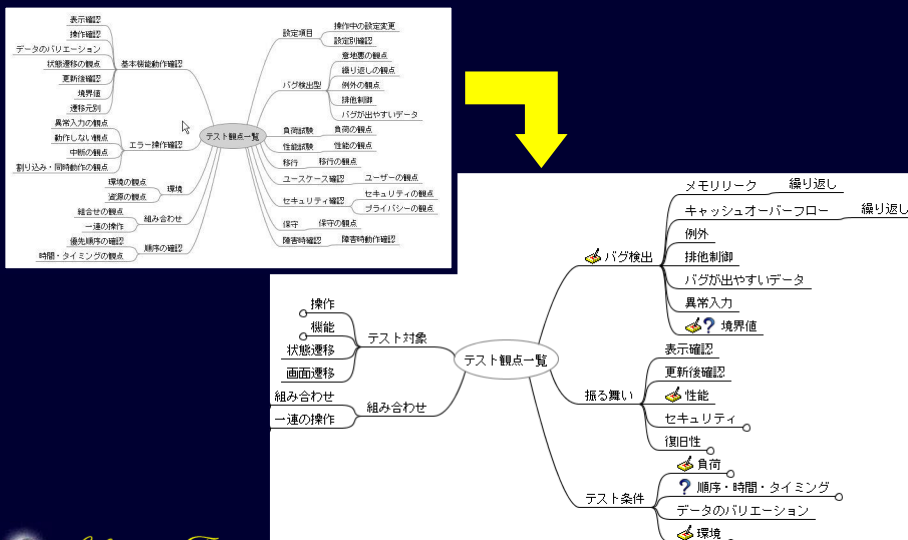
機能名	画面名	項目種類	観点	項目名	補足事項	パラメータ検討
アカウント新規登録	会員情報入力画面	画面レイアウト	ユーザーの観点	画面表示確認	パスワードはマスクされているか	-
			プライバシーの観点
		

「パスワードのマスク」を
テスト観点としてリバーズする

親子関係をかなりしっかり整理する



それぞれのブロックの役割をきちんと検討し分解する



テストフレームを組み上げる

- 複数のテスト観点からテストフレームを組み上げてテストケースを導きやすいようにする

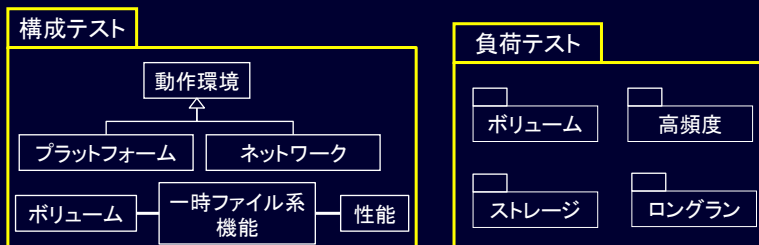
- テスト観点到それぞれ値(水準)を入れていくとテストケースになる
 - » テストフレームはテストケースのスケルトンと言ってもよい
- フレームにはフレーム名が必要だが、
- 整理されたテスト観点図は、テスト対象、テスト条件、振る舞い(期待結果)、狙いたいバグに分かれているので、テストフレームを組みやすい要求モデルになっている
- 今回はマインドマップのツールを使ってフレーミングした



テストコンテナにまとめる(今後の話)

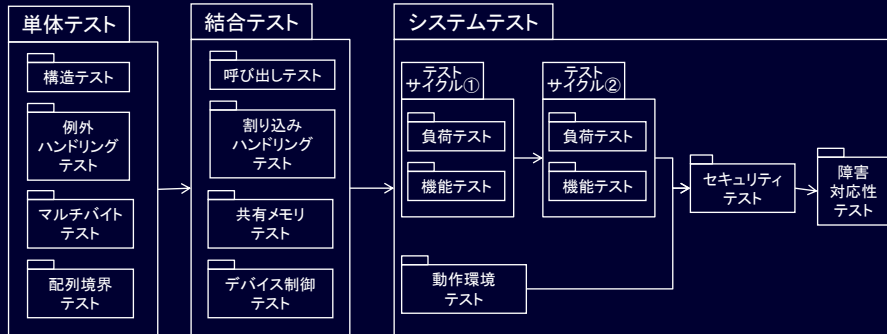
- テスト観点やテストフレームをテストコンテナにまとめる

- テストコンテナとは、テストタイプやテストサイクル、テストレベルのこと
 - » テストコンテナは他のテストコンテナを含んでも構わない
- うまくまとめると全体像を把握しやすくなる
 - » うまくまとめないとテスト設計がぐちゃぐちゃになる
 - » 最初によく検討せずに画面テスト・遷移テスト・機能結合テスト・非機能テストに分けたのが実は敗因である
- テストスイートの品質特性や設計原則を理解する必要がある
 - » 保守性や自動化容易性、結合度と凝集度など

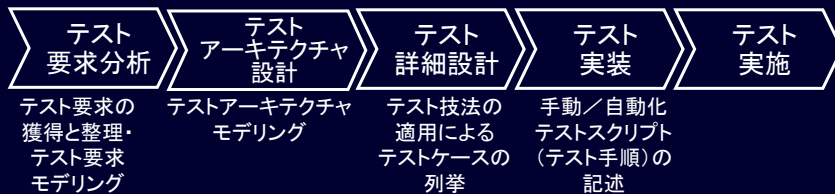


テストアーキテクチャを設計する(今後の話)

- テストの全体像(テストアーキテクチャ)を
テストコンテナ(やテスト観点)の粒度で表し、関係をつなぐ
 - 社内標準にあるテストアーキテクチャはほとんど粗すぎるので、
現実のテストアーキテクチャをリバースして
その意味や趣旨をきちんと理解することから始めるとよい



プロセスの実施順序と改善順序は同じとは限らない



改善順② 改善順③ 改善順①

まとめ

- **テストの趣旨とか意味とかをとにかく書く・描く**
 - 何を使って書く・描くかは大きな問題ではない
 - フォームを埋めようとしてはいけない
- **何が問題なのか、をとことん考える**
 - 自分たちの疑問や違和感を大事にする
 - » 「まあいいや」「こんなもんか」ではなく、書いてあることの意味をちゃんと理解し、同じグループかどうかや、つながりなどにしっかりこだわる
 - 何が問題なのか、をきちんと理解する前に方法論を導入してはいけない
 - » 改善できない組織に限って、外部の権威や方法論に頼ってしまう
- **今までのやり方はある程度尊重しつつ、それでいて今までのやり方に囚われないように改善する**
 - 何が問題なのか、が分かっているれば改善の方針は明確になっている
 - テスト技術者が考えることを分類するのが、プロセスを定義することである
 - 道具にはその道具の本分を果たしてもらい、何でも担わせようとしな
 - » No Excel, No Confusion!
 - そう変えて結局何が嬉しいのか、をいつも意識する
 - » 定量化方針を決めてから何が嬉しいかを考えるのではなく、何が嬉しいのかをきちんと実感できたら定量化方針は自然に決まる



Software Testing

17

© NISHI, Yasuharu

現場・現実・現物・原理・原則！



智美塾2号生 西 康晴(電気通信大学)

Yasuharu.Nishi@uec.ac.jp

© NISHI, Yasuharu