

ソフトウェアテストにおける バグ推測によるテスト設計手法の提案

～過去バグ事例を活用した経験が浅いメンバーのバグ推測能力の向上～

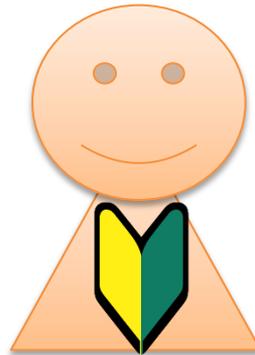
主査	: 奥村 有紀子 (有限会社デバッグ工学研究所)
副主査	: 秋山 浩一 (富士ゼロックス株式会社)
	喜多 義弘 (東京工科大学)
アドバイザー	: 堀田 文明 (有限会社デバッグ工学研究所)
研究員	: 川上 祐司 (ブラザー工業株式会社)
	大島 祥吾 (株式会社東海理化)
	榎本 和也 (株式会社神戸製鋼所)
	岩尾 隆弘 (株式会社日立製作所)

2016年2月26日

はじめに

このテスト設計手法により

新人のバグ推測をレベルアップすることが目的



新人:A君

目次

1. 背景
2. 現状調査
3. 提案手法
4. 具体例
5. まとめと課題

1.背景：最終確認テスト



不具合の市場流出
をなくしなさい

開発部門

開発



テスト



QA部門

ソフト
受入



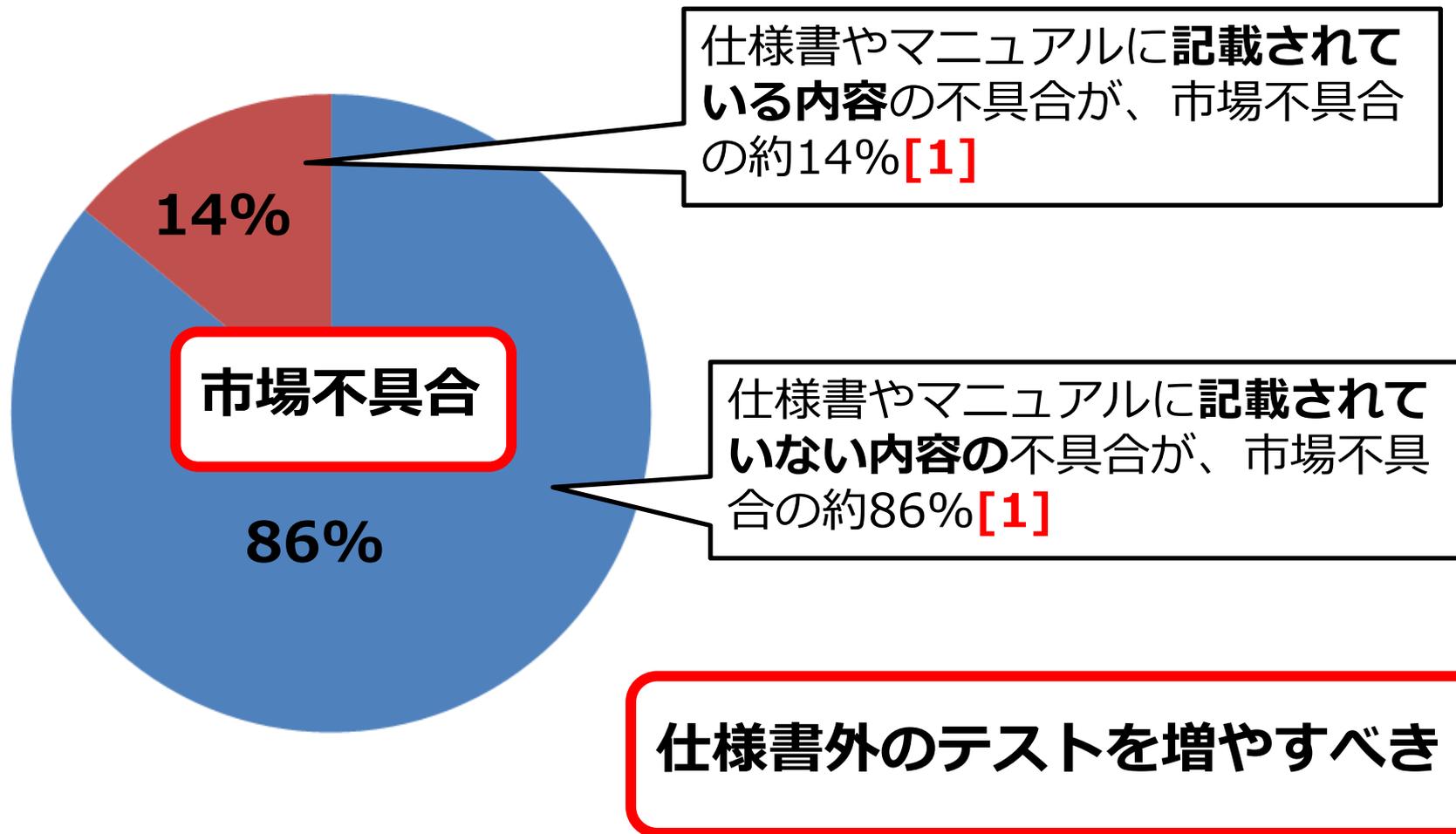
最終
確認
テスト



出荷

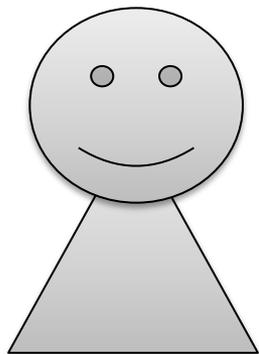
研究員の職場では仕様書に記載のない
市場条件でのテストを実施

1.背景：市場不具合状況



[1]<星名 卓郎 他, 製品検査における仕様書記載外の問題点摘出強化に向けたテスト観点拡充方法の検討と実践, ソフトウェア品質シンポジウム 2015, 2015>

1.背景：実施メンバー



リーダー：B氏

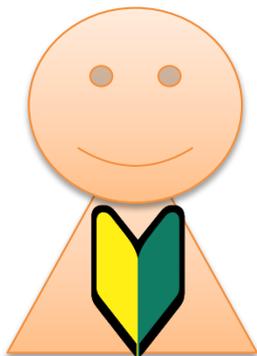
◆テスト経験が豊富なメンバー：

テスト設計を複数年実施した人

- ・既存機能におけるバグに関する情報を熟知
- ・バグの摘出を目的としたテスト設計をしている。



多忙や人数の関係により
テスト設計を依頼



新人：A君

◇テスト経験が浅いメンバー：

ソフト部門外から来た人

- ・製品の既存機能については理解している
- ・仕様書に記載されている範囲のテストしか実施できない

2.現状調査：経験豊富な考え方

知識ベース

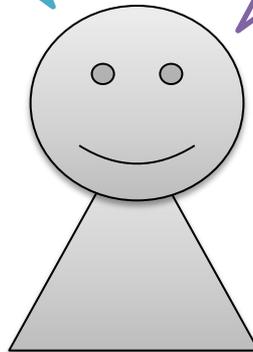
製品の位置づけや使われ方の知識や過去に抽出されたバグの知識からの情報

経験ベース

プログラミング経験からプログラム構造を意識した情報や出荷後のトラブル対応からの情報

設計仕様書

先行テストの情報



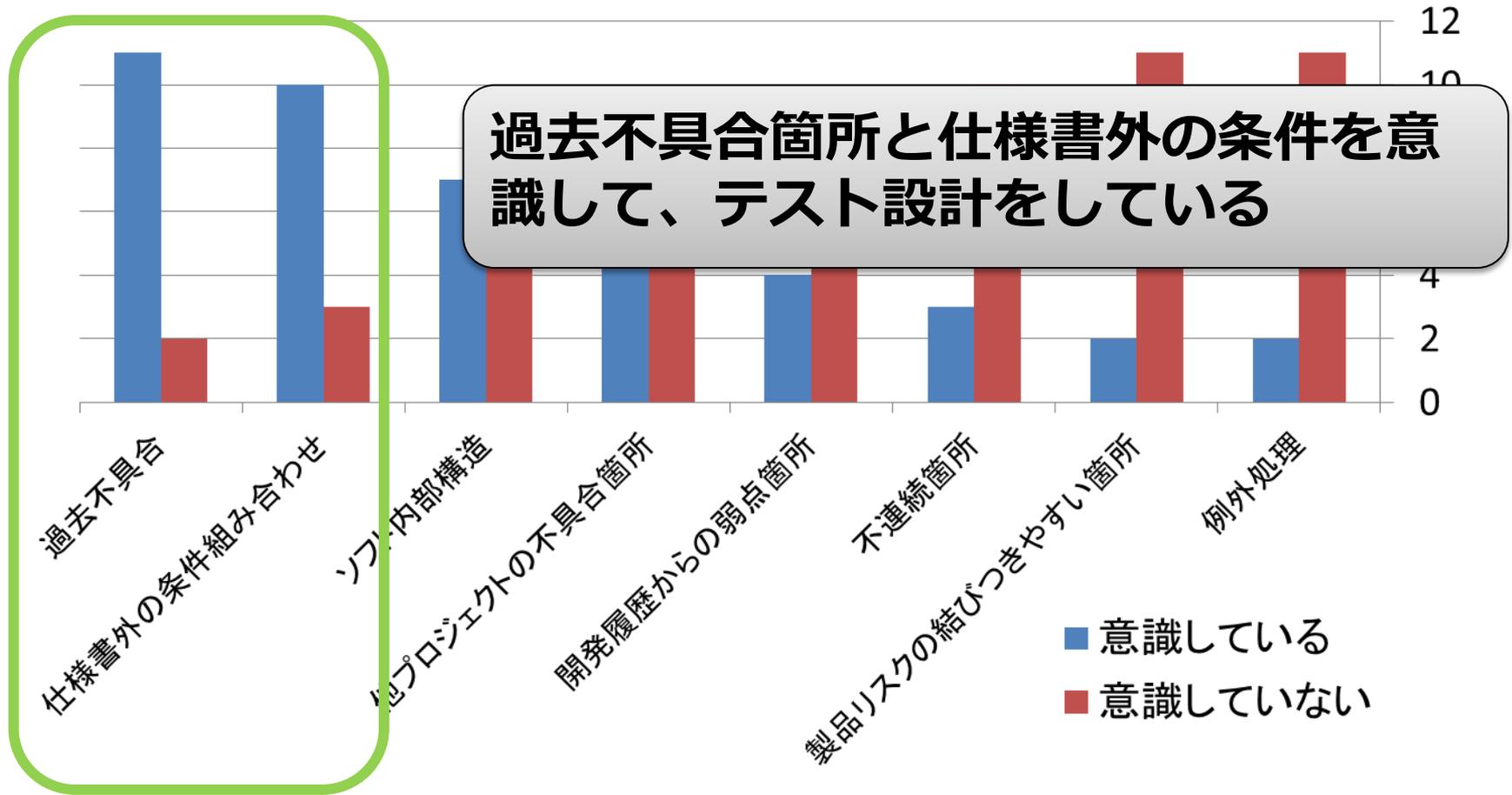
仕様書記載外のバグが見つかるテスト設計

リーダー：B氏

2.現状調査：アンケート

Q.テスト項目作成時に注目する箇所は？

(対象：経験豊富なメンバー 内容：選択式)



3.提案手法：仮説

知識ベース

今までに蓄積されてきた
機能、観点、事例のデータ
がある。

観点

過去
事例

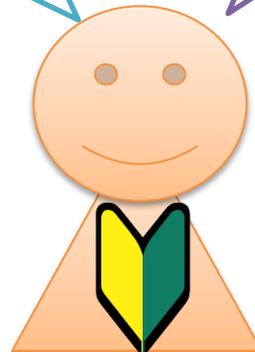
既存
機能

経験ベース

プログラミングやトラブル
対応経験があまりない。

設計仕様書

先行テストの情報

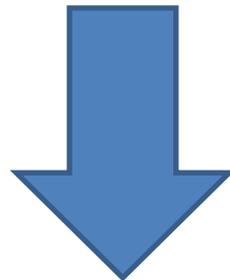


新人:A君

**知識ベースに蓄積されている
データ同士を関連付けた情報
があれば、バグを推測したテ
スト設計ができる。**

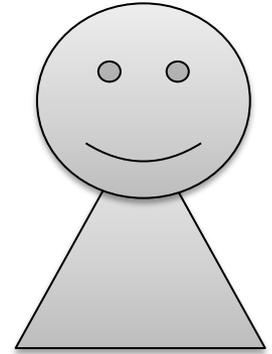
3.提案手法：提案

既存機能と過去のバグ抽出時の**観点**から過去事例が結びつくようなマトリクス表の作成



経験豊富なメンバーの考え方を明示することで、新人でもバグを推測したテスト設計が可能。

4.具体例：作成手順



リーダー：B氏

①テスト対象の既存機能の一覧作成

②バグ事例から、テストケースを
抽象化したテスト観点一覧作成

③ **既存機能**と**過去のバグ抽出時の観点**
のマトリクス表を作成する。

④バグ抽出事例帳票を作成、帳票番号
を交点に書込みリンク付

4.具体例：既存機能

1

過去のバグ摘出時の観点

既存
機能

『既存機能』

それぞれの製品に対応した機能に分割する。

機能A

機能B

機能C

機能D

信
奇

資源
競合

・
・
・

30,45

14

4

55

22

65

4.具体例：観点

2

過去のバグ抽出時の観点

既存
機能

中止後の
操作

素早い
操作

同時

初期化

冗長化
構成

通信
負荷

資源
競合

・
・
・

『過去のバグ抽出時の観点』

テスト観点は分かりやすく分類、
製品の特성에応じてカスタマイズ

機能A

機能B

機能C

機能D

4.具体例：バグ事例

3

マトリクス表の作成

既存機能	過去のバグ抽出時の観点							
	中止後の操作	素早い操作	同時	初期化	冗長化構成	通信負荷	資源競合	・・・
機能A			<u>4</u> <u>11</u>	<u>3,13,61</u>	<u>1</u>			
機能B	<u>15,28</u>	<u>19</u>	<u>12,51</u>				<u>30,45</u>	
機能C			<u>14</u>			<u>4</u>		<u>55</u>
機能D						<u>22</u>		<u>65</u>

※青数字下線部は「バグ事例の帳票番号」を表す。

4. 具体例：バグ事例

3

マトリクス表の作成

帳票番号	3、13、16
発生日時	2015/xx/xx
プロジェクト名	〇〇SWモジュール
現象	SW AをON後に信号Aの出力を確認し、電源をOFF/ONしてから再度SW AをONしたら、信号Aが出力されなかった。
発生メカニズム	初期化処理の不整合
対策	SW AをONし、電源ON→OFF後にSW AがONされることを確認する。

原
合

・
・
・

45

55

**既存機能、観点、バグ事例が紐付き
バグ推測が可能となる。**

4.具体例：バグ推測プロセス

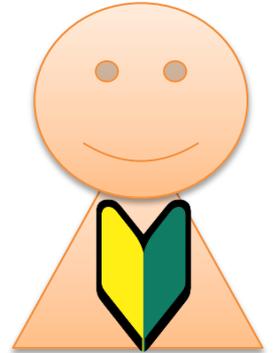
①仕様変更情報の把握

②マトリクス表からテスト対象機能の
選択

③リンク箇所のテスト観点確認

④テスト観点からバグの推測

⑤バグ事例を参照し、バグ推測の拡充



新人:A君

4. 具体例：マトリクス使用例

1 仕様変更情報の把握

既存機能	3 去のバグ抽出時の観点					
	中止後の操作	素早い操作	同時	4 バグの推測	資源競合	...
機能A			<u>1</u>	<u>3,13,61</u>	<u>1</u>	
機能B	<u>15,28</u>	<u>1</u>	<u>12,51</u>			<u>30,45</u>
機能C			<u>14</u>		<u>4</u>	<u>55</u>
機能D					<u>22</u>	<u>65</u>

テスト対象「機能C」とテスト観点「同時」から14を選択

4.具体例：バグ推測例

帳票番号	14
発生日時	2015/xx/xx
プロジェクト名	〇〇SWモジュール
現象	SW AとSW Bを同時にONしたらSW Bの出力情報しか表示されない。
発生メカニズム	SW Bの更新処理がSW BのRAM領域を上書きするプログラムとなっていた。
対策	SW Bの更新処理をSW Aで上書きしないように、論理和に修正する。

メモリー
ーク

[55](#)
[65](#)

検本機能

29

[29](#)

[85](#)

A君が推測したバグ

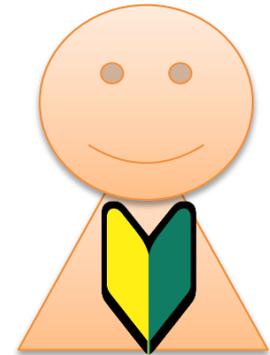
複数の入力のタイミングによって、データがクリアされる

A,B 同時条件成立時にCが動作する

5.まとめ、課題

■まとめ

この提案から、新人でも経験豊富なメンバーのようなバグ推測のレベルアップをすることができるようになる。



■今後の課題

実際のテスト現場で検証ができておらず、有効性の評価まで至らなかった。各社で、マトリクス表を作成したバグ推測の検証を行う。

御清聴ありがとうございました。

～SQiP'15 第5分科会 Bグループ～