OMRON
Sensing tomorrow™

グレーボックステストによる効果的な品質確保の取り組み

ー無駄なテストを排除し、 影響範囲に焦点 をあてた、テスト設計技法の紹介ー

2013/1/30

オムロン株式会社
田中桂三

自己紹介

OMRON
Sensing tomorrow™

- 役割
  - 産業用電子機器プログラミングツール（Windows ソフトウェア）開発のテストリーダ
- 最近の注力活動
  - テスト設計プロセス改善
    信頼性分析、グレーボックステスト
  - 自動テスト普及・展開
- 社外活動
  - JaSST2012Tokyoにて論文発表
    “信頼性分析による品質確保の取り組み”
  - ソフトウェア品質シンポジウム2012にて論文発表
    “テスト種類に着目した最適な自動テスト支援ツールの選定方法と実践”

目次

OMRON
Sensing tomorrow™

- グレーボックステスト導入の目的
- グレーボックスによるテスト設計
- グレーボックステストの導入効果
- 今後の展開

グレーボックステスト導入の目的

OMRON
Sensing tomorrow™

お客様のご要望にタイムリーに応える為に、ソフトウェア開発の短縮化が必要。

目標値：開発全体期間：６ヵ月⇒３ヵ月　テスト実施期間半減：12週間⇒６週間

如何にテスト期間を短縮するか？

従来のテストを分析した結果、

内部構造を理解して効率よくテストを実施すべき！

そこで、

グレーボックステスト(※)を導入、実践！
※内部構造を理解した上で外部動作をテストする手法

導入・実践結果、

テスト期間：5週間を達成！（目標値クリア）

本発表では、グレーボックステストの導入、設計時の工夫内容、効果および今後の展開について説明します。

OMRON
Sensing tomorrow™

グレーボックスによるテスト設計

OMRON
Sensing tomorrow™

テストの課題設定

テスト期間がかかる理由 ～従来の派生型商品開発の分析結果～

課題1)
多大なテスト工数が必要

テスト工程で全機能を検証

↓

内部変更に影響ない機能までテストしている

↓

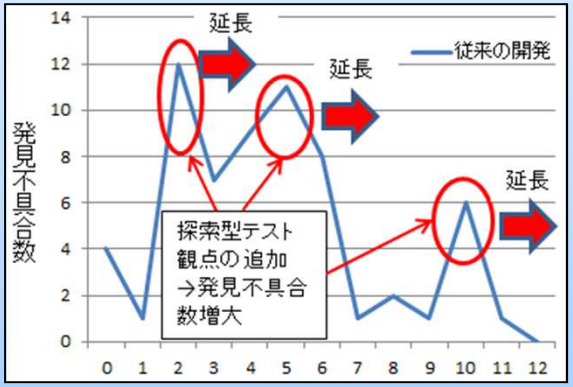
上流設計で品質保証できる範囲までもテスト実施

課題2)
不具合発見に期間がかかる

テストの中盤～後半に不具合発見

↓

複数条件で発生する不具合



↓

外部動作だけで不具合手順を探索

中味を知らないから・・・（テスタのつぶやき）

↓

中味を理解していないため、非効率な作業

解決策として、

中味を理解してテストをしよう！

グレーボックステストを導入！

グレーボックステストの導入問題

OMRON
Sensing tomorrow™

「グレーボックステスト」導入を決めたものの、問題に直面

ノウハウを持っていない！
効果的な事例や手法が見つからない！

そこで、

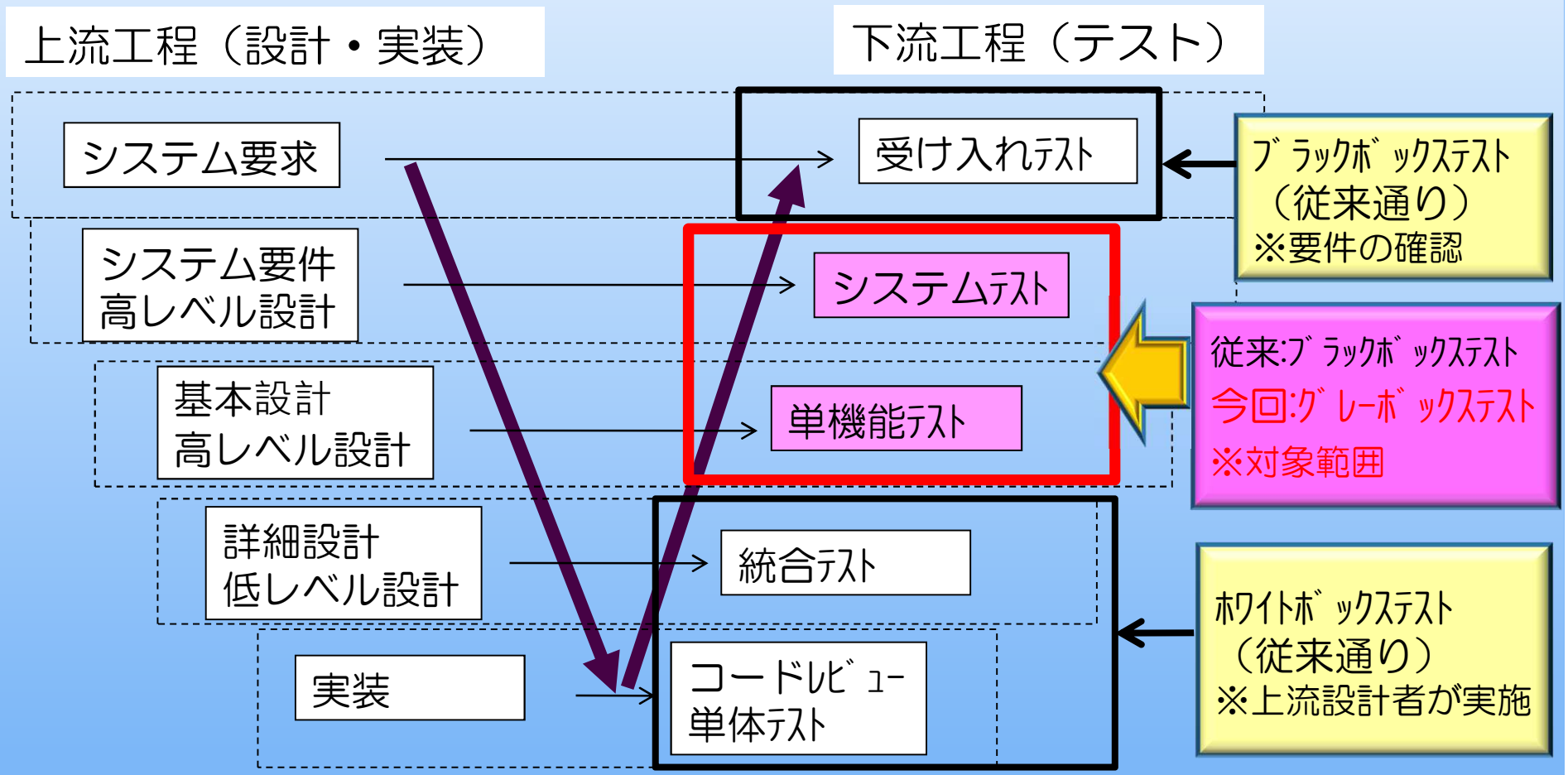
【解決策】

自分たちのプロジェクトに適合したやり方を、自分たちで創り出そう！

①グレーボックステストの対象範囲

OMRON
Sensing tomorrow™

システムテストと機能テストにグレーボックステストを導入。
【理由】期間がかかる＆探索テストの対象⇒導入効果があると判断！



②グレーボックステストのINPUT資料

OMRON
Sensing tomorrow™

グレーボックス導入における、前提条件

- 仕様書を増やさない！
- 上流設計者の負担を増やさない！

【理由】
仕様書メンテの工数増大防止、更新漏れによる品質低下防止

既存の構造仕様書を活用しよう！

そこで、

現開発プロジェクトで作っている
「UML」をINPUT資料とする！

③効果的なテスト設計方法

OMRON
Sensing tomorrow™

グレーボックステスト導入にあたり、テストの課題から３つの解決方針を設定

課題1)「多大なテスト工数が必要」の解決策として

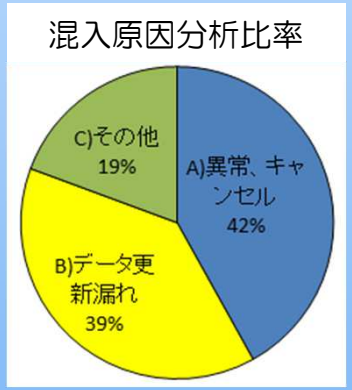
「とりあえず全機能実施」という考えは、止めよう！

【解決方針1)】
漏れなく内部変更情報を集めて、各機能への影響有無を特定

課題2)「不具合発見に期間がかかる」の解決策として

テストの中盤～後半に発見される不具合を早い段階でみつけよう！

該当不具合の混入原因分析⇒A)とB)が大半、ここに注力



A)異常、キャンセルケースの考慮漏れ

【解決方針2)】1)の機能で、
影響を及ぼす処理（操作）を特定

B)データ更新の考慮漏れ

【解決方針3)】1)の機能で、
影響を及ぼすデータと更新タイミングを特定

UMLから
情報入手

1)各機能への影響有無特定

【解決方針1)より】

「コミュニケーション図」(※)の活用

※複数のオブジェクトが連携して目的を達成する振る舞いをモデリング

ところが、テスト設計の中で、

問題発生

そこで、

対策

テスタが理解できない

変更理由が理解できない！
オブジェクト名、インターフェース名だけでは理解できない ！

コミュニケーション図に注釈を追記（次頁）

内部変更の漏れが判断できない

テスタの誤り（仕様の誤解、反映漏れ）の有無が判断できない
変更記載漏れが判断できない

要件-機能マトリクスの作成（次頁）

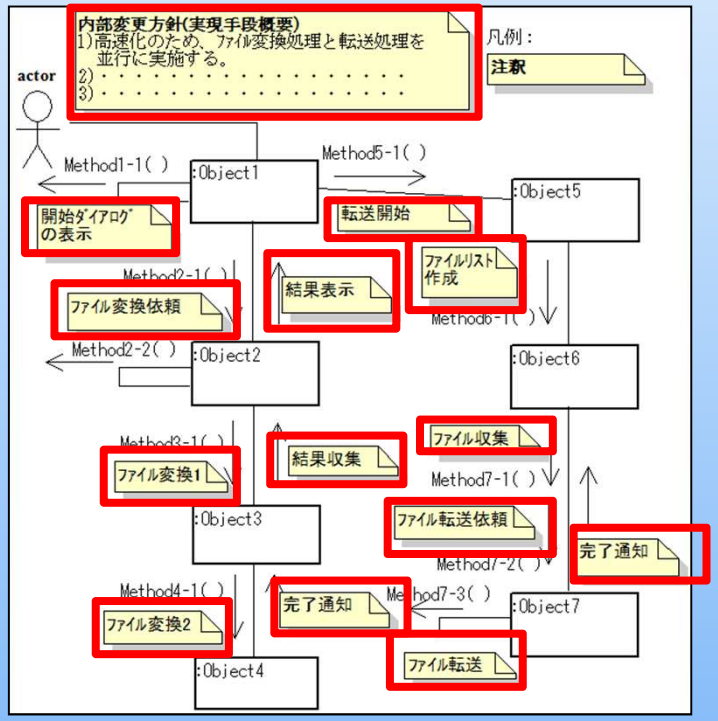
ソースコード変更履歴で確認

比較により、コミュニケーション図の記載漏れ有無を確認

コミュニケーション図、要件機能マトリクスでの対策

OMRON Sensing tomorrow™

コミュニケーション図に注釈を追記



- 内部変更方針（実現手段）
- 外部仕様の用語（機能名、データ名、処理名）

テスト設計者が要件-機能マトリクスを作成

要件＼機能	大機能1			大機能2			大機能3		
	小機能1-1	小機能1-2	.....	小機能2-1	小機能2-2	.....	小機能3-1	小機能3-2	.....
要件1	○	○		●	○		△	▲	
要件2	○	○		▲	△				
要件3	○	○		●	○		○	●	
要件4	○	○		▲	△		○	▲	
要件5	○	○					△	▲	
⋮									
要件16	▲	▲		●	●				
要件17	○	○		▲	○				
要件18				▲	△		○	●	
要件19	○	○		●	△		△	▲	
要件20				●	△		△	▲	

- 内部変更の影響度合により3レベル（5分類）に分解、テスト要否判断

●、○（※）：対象機能の内部変更あり
⇒上流設計者がホワイトボックステスト、テスタがグレーボックステストで検証。
▲、△（※）：対象機能の内部変更はないが、外部仕様に影響あり
⇒テスタがグレーボックステストで実施
空白：内部変更がなく、外部仕様にも影響がない。
⇒無駄なテストとして実施対象外
※黒印：該当要件に対し、重要度の高い不具合が発生する機能

- テスト設計者の誤解防止のため、上流設計者の全員レビュー

OMRON
Sensing tomorrow™

2)影響を及ぼす処理（操作）を特定

【解決方針2) より】

「アクティビティー図」（※）の活用

※一連の処理を構成する動作に着目。実行順序や条件、制御などの依存関係を示す。システムの振る舞いやワークフローなどを表現できる。

ところが、テスト設計の中で、

問題発生

キャンセル、異常ケースが記載されていない。

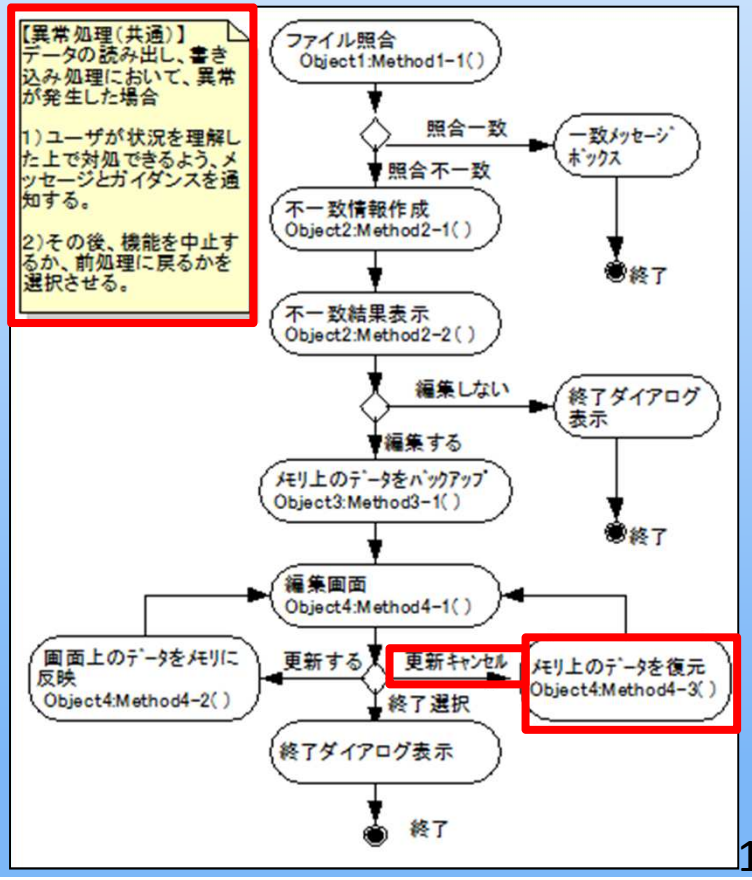
そこで、

対策

途中でキャンセルした処理を、アクティビティー図に追記

異常処理は、フロー図に入れず、注釈として追記

【注釈にした理由】
- フローに記載すると図が複雑になる
- 大部分が共通の処理、注釈で十分



OMRON
Sensing tomorrow™

3)影響を及ぼすデータを特定

【解決方針３）より】

「データフロー図」（※）の作成

※データの流れを表現する図。データ処理の可視化に使われる。

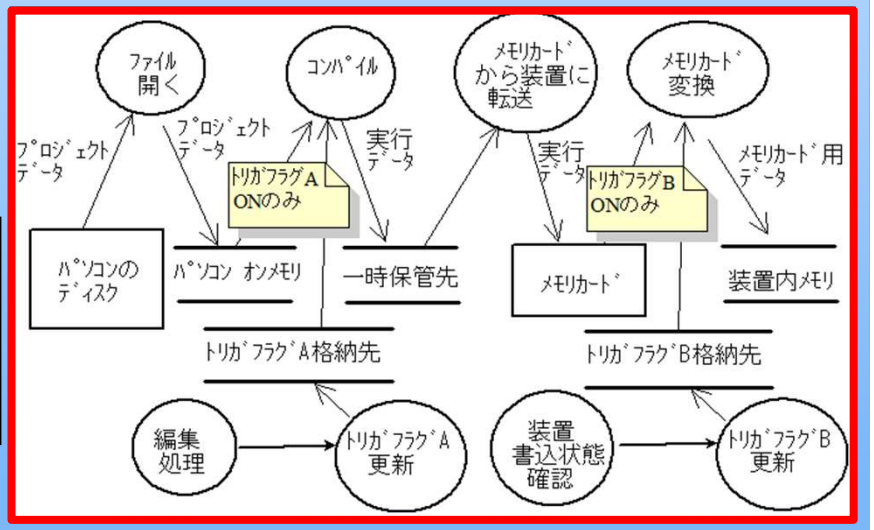
当初、コミュニケーション図で、データを特定しようとしていた。
⇒機能に影響を及ぼすデータはわかったものの、

問題発生　「どのタイミングでデータが更新されるのか？」がわからない

そこで、

対策　データフロー図（※）を作成

- UMLではないが、問題解決に最適と判断し導入
- テスト設計者の誤解防止のため、上流設計者の全員レビュー

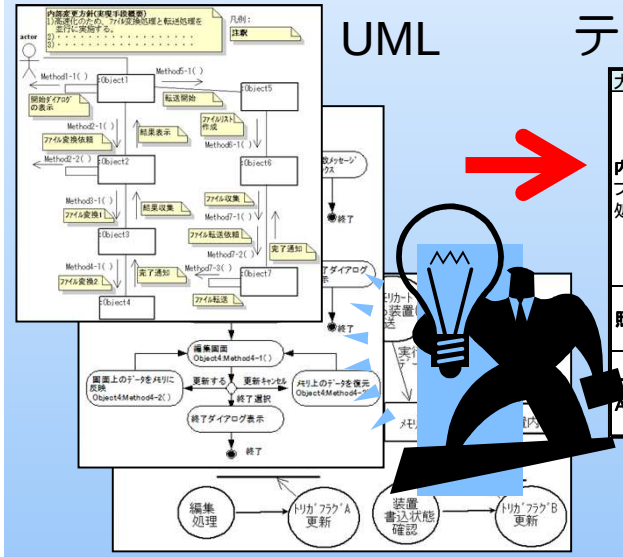


OMRON Sensing tomorrow™

④テスト設計⇒テスト実装への落とし込み

テスト設計者が、UML＋仕様書類から「テスト観点シート」に落とし、テスト実装者へINPUT

UML



テスト観点シート（観点）

大項目	テスト観点
内部変更方針 プログラム変換処理と転送処理を並行に実施する。	プログラム変更と転送の組み合わせ
	複数ノードからのアクセス
	：
照合不一致判定	不一致判定後の編集
	：
コンパイルフラグ（トリガAフラグ）の追加	メモリカード書き込みフラグ
	：
	：

テストケース

大項目	中項目	テストケース
内部変更方針 プログラム変換処理と転送処理を並行に実施する。	プログラム変更と転送の組み合わせ	プログラム変更と転送を繰り返すケース
		プログラム変更後、転送をキャンセルするケース
		：
	複数ノードからのアクセス	転送中に他ノードからプログラム変更をかけるケース
	：	：
照合不一致判定	不一致判定後の編集	編集後に更新・キャンセルを繰り返すケース
	：	：
コンパイルフラグ（トリガAフラグ）の追加	メモリカード書き込みフラグ	新規作成後に、編集しコンパイル
	：	ファイルオープン後に、編集しコンパイル
	：	：

観点,実施範囲をINPUT

＋

仕様書類
・システム仕様書
・機能仕様書

＋

要件-機能マトリクス（実施範囲）

機能 / 要件	大機能1 小機能1-1	大機能1 小機能1-2	……	大機能2 小機能2-1	大機能2 小機能2-2	……	大機能3 小機能3-1	大機能3 小機能3-2	……
要件1	○	○		●	○		△	▲	
要件2	○	○		▲	△				
要件3	○	○		●	○		○	●	
要件4	○	○		▲	△		○	▲	
要件5	○	○					△	▲	
：									
要件16	▲	▲		●	●				
要件17	○	○		▲	○				
要件18				▲	△		○	●	
要件19	○	○		●	△		△	▲	
要件20				●	△		△	▲	

テスト手順作成（テスト実装）

テスト要項書（テストケース毎の操作手順）

レビュー

OMRON
Sensing tomorrow™

グレーボックステストの導入効果

無駄テストの削除

OMRON
Sensing tomorrow™

課題1)「多大なテスト工数が必要」に対する効果

「要件-機能マトリクス」で空白の箇所

↓

内部変更による影響がない
ことを上流設計で確認できている

機能 / 要件	大機能1		.....	大機能2		.....	大機能3		.....
	小機能1-1	小機能1-2		小機能2-1	小機能2-2		小機能3-1	小機能3-2	
要件1	○	○		●	○		△	▲	
要件2	○	○		▲	△				
要件3	○	○		●	○		○	●	
要件4	○	○		▲	△		○	▲	
要件5	○	○					△	▲	
：									
要件16	▲	▲		●	●				
要件17	○	○		▲	○				
要件18				▲	△		○	●	
要件19	○	○		●	△		△	▲	
要件20				●	△		△	▲	

「無駄なテスト」
＝システムテスト実施対象から外した。(※)

※ただし、構成管理の問題による不具合流出防止の為
テスト自動化による回帰テストにて検証。
※実績：外した部分の不具合は構成管理の問題以外、
後工程で、見つかっていない。

OMRON
Sensing tomorrow™

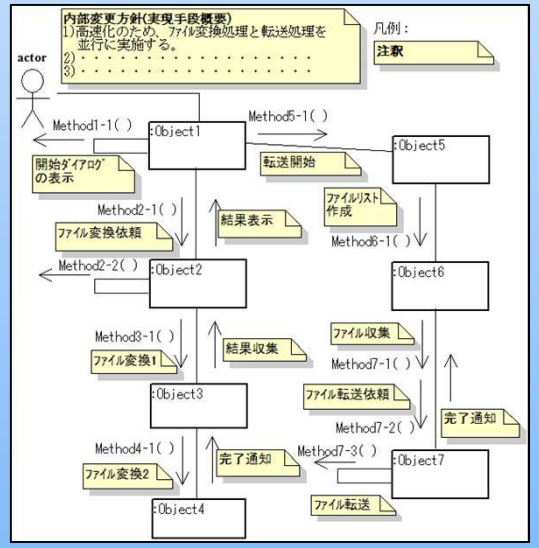
探索型テストでの不具合の早期発見

課題2)「不具合発見に期間がかかる」に対する効果

複数条件起因の不具合を短期間で発見する為、
３つのUMLから探索型テスト観点を抽出

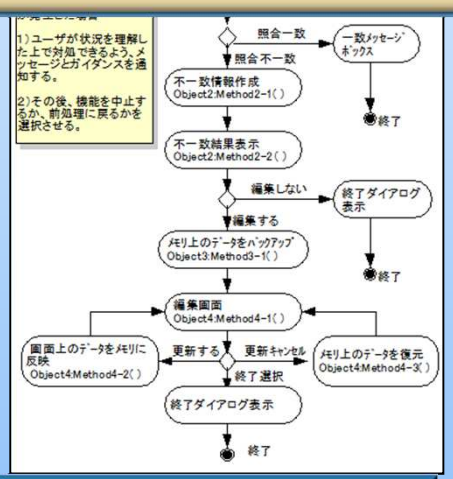
コミュニケーション図

影響機能を特定



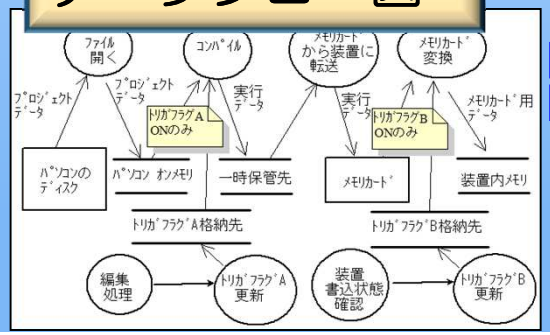
＋

アクティビティー図



＋

データフロー図



＝

機能＋処理（操作）の特定による観点抽出

【探索型テスト観点】
- ワークフローの条件分岐
- ループ処理、
- キャンセル処理
- 異常処理の条件

＝

機能＋データの特定による観点抽出

【探索型テスト観点】
- データ変換のトリガ操作
- コンポーネント間の排他処理とタイミング

グレーボックス導入効果

OMRON
Sensing tomorrow™

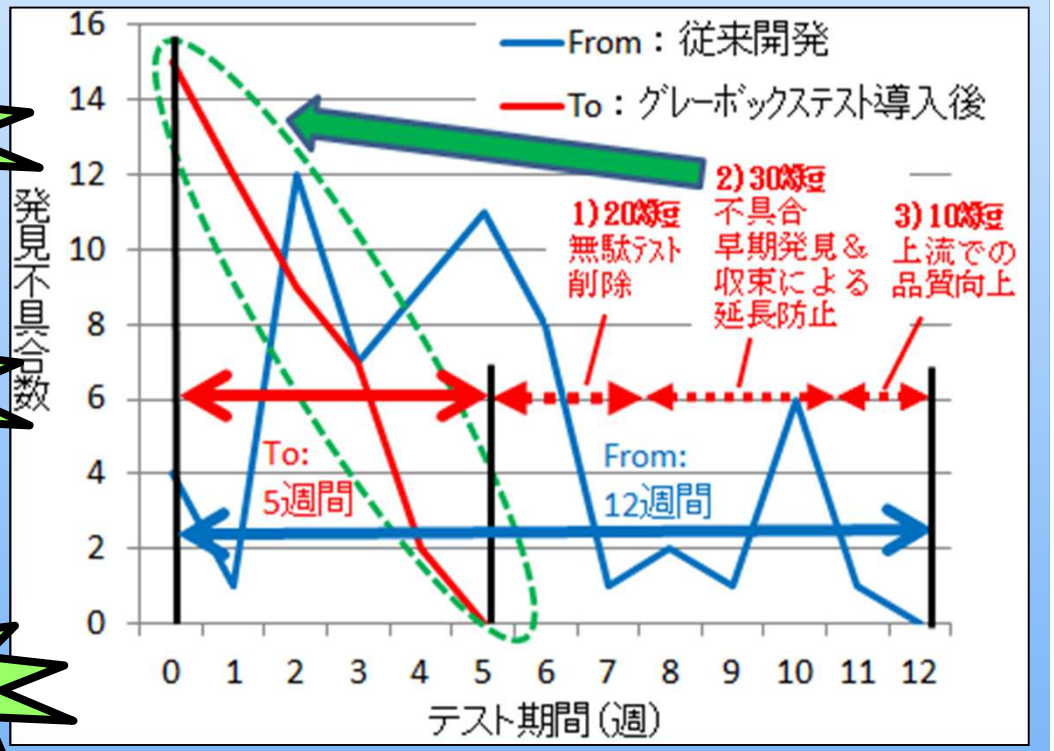
3点の効果により、目標（テスト期間半減：６週間）をクリア！
従来同規模派生型開発プロジェクトで、
「12週間」から「5週間」に短縮！

テスト期間短縮

1）無駄テストの削除 → 20%減

2）探索型テストによる不具合早期発見
⇒早期不具合収束 → 30%減

3）UMLレビュー&改善による上流工程での品質が向上
⇒テスト工程での不具合数が30%削減、不具合修正確認時間が短縮 → 10%減



目標を上回った主要因

OMRON
Sensing tomorrow™

今後の展開

最適なインプット情報の検証

OMRON
Sensing tomorrow™

前提条件　既存の仕様書を活用しよう！

結果

○　上流設計で作成した既存のUML「コミュニケーション図」と「アクティビティー図」を活用

×　「要件−機能マトリクス」と「データフロー図」を新規にテスト設計者が作成

+2人月

今後に向けて

「最適な設計ドキュメントは何か？」を調査検討

他のUML、構造設計ドキュメントの特徴を調査分析
与件　・テスト設計工数を増やさない
　　・設計、テスト両方で活用できる

上流工程での品質確保の取り組み

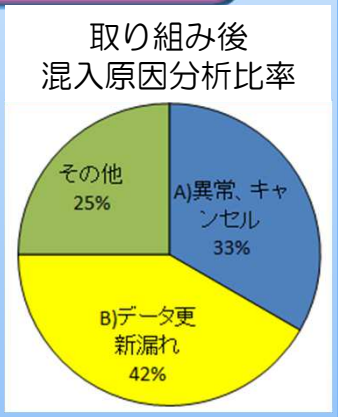
OMRON
Sensing tomorrow™

○ 上流工程での品質が向上し、テスト実施工程での
不具合数が30%削減

△ 異常、キャンセル、データ更新漏れ起因
の不具合比率が依然高い（＝設計工程からの流出）

振り返り

グレーボックスのテスト設計が構造設計工程で
完了できず、実装工程と並行して実施した。
⇒テスト設計を早く完了すれば上流設計の品質が上がり、
テスト工程の不具合が減る！



今後に向けて

不具合数半減！

「Wモデル」実践によりテスト設計完了を早める

- 上流設計とテスト設計を並行して進める。
- テスト設計を構造設計工程期間内で完了させる
- 上流設計者、マネージャへの理解・協力に取り組む

プロダクト品質向上への貢献度考察

OMRON
Sensing tomorrow™

？ プロダクト品質向上への貢献については、十分深堀りできていない。

× 受け入れテスト（後工程）でも不具合が見つかっている。⇒テスト観点漏れがある。

今後に向けて

- 受け入れテスト（後工程）の発見不具合分析
  ⇒ 本取り組みの流出分析、改善
- テクニカルテストアナリスト育成、観点強化
- グレーボックステスト導入の開発プロジェクト数を増やし、導入前後での品質状況観測

OMRON
Sensing tomorrow™

以上で発表を終わります。
ご清聴、ありがとうございました！