

# 派生開発での時間効率性劣化を 変更要求から検出する方法

## 第6分科会 Bグループ

小笠原 勝 (GEヘルスケア・ジャパン株式会社)  
島崎 稔史 (株式会社インテック)  
中島 秀人 (東京海上日動システムズ株式会社)  
中村 直人 (矢崎総業株式会社)  
中村 奈津子 (日本電子株式会社)

目次	1. 研究経緯
	2. 現状分析
	3. 解決策
	4. 検証
	5. まとめ

# 1. 研究経緯 (1)

---

## □ 派生開発の現場で困っていること (研究員各社)

**システムテストや納品後に見つかる  
応答時間や処理速度に関する不具合**

例)

「前よりもアプリの起動が遅くなって不便」

「前よりもデータ更新の待ち時間が増えた」

# 1. 研究経緯 (2)

---

□ 応答時間や処理速度とは

## 時間効率性

ISO/IEC 25010

例)

「前よりもアプリの起動が遅くなって不便」

「前よりもデータ更新の待ち時間が増えた」

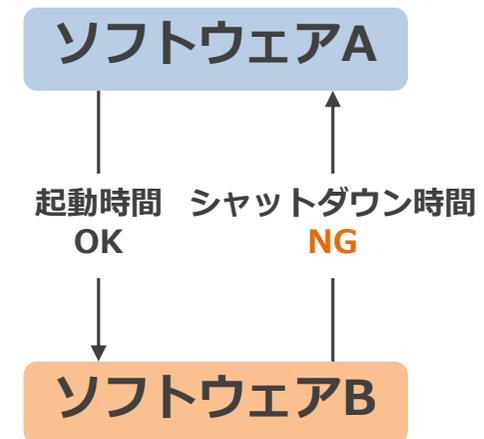


時間効率性の劣化

# 1. 研究経緯 (3)

□ 派生開発における時間効率性の劣化とは  
改修前と比較した場合の時間効率性の  
変化が**顧客の期待と乖離**すること

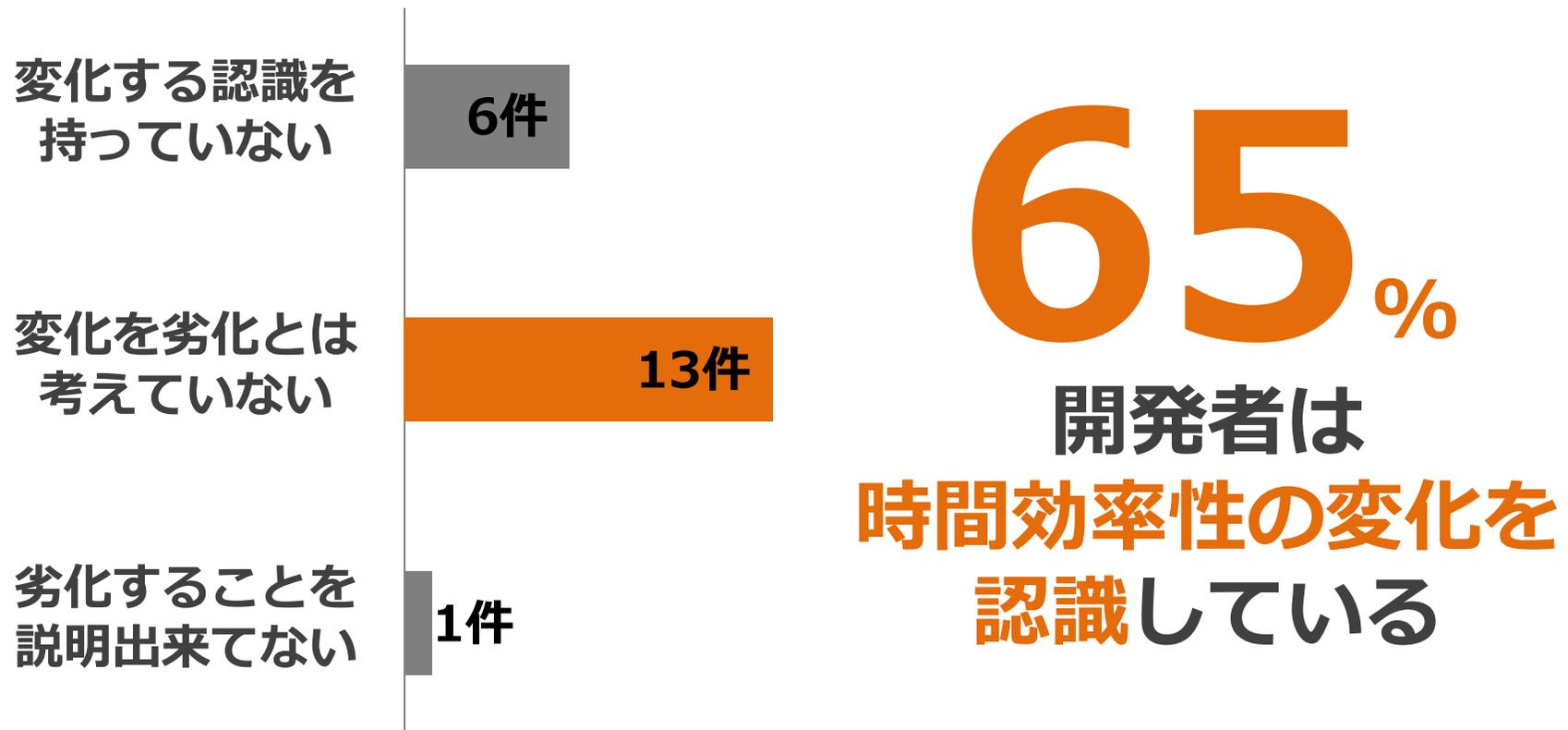
- 例) ①要求を実現するためソフトウェアBを追加  
↓  
②起動時間は要求通りの仕様を満たしている  
↓  
③シャットダウン時間が**前より遅くなった**…  
→要求してないところは前と同じにしてよ



**時間効率性の劣化防止対策が必要**

## 2. 現状分析 (1)

### □ 時間効率性の不具合を分析 (N=20)



認識はあるが劣化とは思わず

## 2. 現状分析 (2)

---

### □ なぜ変化を劣化と考えないのか

- ✓ 不具合分析から判ったこと
  - 変化が生じる事は**漠然と認識**している
  - 具体的な**時間の変化を把握**出来ていない



- ✓ 開発者にヒアリングを実施
  - 時間の変化が**見える形**になっていない



**「時間の変化」の可視化が必要**

## 3. 解決策（1）

### □ 基本方針

手順	方針	解決策
1	時間効率性の <b>変化を認識し易くするための観点</b> を提供する	改訂版リソース 変化点確認表
2	時間効率性の <b>変化を定量的に示して変更箇所を可視化</b> する	時間効率性記入表
3	可視化した情報から <b>時間効率性の変化量を把握して、顧客との交渉や設計見直し</b> を行う	顧客交渉や 設計の見直し

Estimation Method Of Time behavior degradation

**EMOT（イーモット）と定義**

### 3. 解決策 (2)

---

#### □ 変化を劣化と認識し易くする理由

劣化に対する知見や経験がない開発者が影響を分析しても**考慮不足に陥る**ため



考慮すべき時間効率性の

劣化に繋がる観点を示せば良い

改訂版リソース変化点確認表

### 3. 解決策 (3)

#### □ 改訂版リソース変化点確認表

No	分類	リソース変化点 チェック項目	時間効率性の 劣化目安
1	変数・配列	配列のサイズに変化はないか	
2	メモリ	確保するメモリサイズに変化はないか	
3		同時に確保されるメモリサイズに変化はないか	
4		メモリへの書き込みサイズに変化はないか	

### 3. 解決策 (3)

#### □ 改訂版リソース変化点確認表

No	分類	リソース変化点 チェック項目	時間効率性の 劣化目安
		<b>ポイント：</b> 1) 劣化目安の値は推定ではなく <b>過去の派生開発実績を引用する</b> 2) プラス（遅くなること）だけで なく <b>マイナス（早くなること）</b> も考慮する	<b>±15ミリ秒</b>

劣化に繋がる観点を数値で示す

### 3. 解決策（4）

#### □ 変化を定量的に示して変更箇所を可視化

変更要求	顧客の操作	チェック項目No	現状の応答時間(ミリ秒)	予測時間(ミリ秒)	予測変化率(%増減)	実績(ミリ秒)
検索条件を3つから4つに変更してほしい	検索条件を選択する					
	検索ボタンをクリックする					

顧客の操作とリソース変化点の観点を紐付けて**具体的な影響**として検出する



**時間効率性記入表**

### 3. 解決策 (5)

#### □ 時間効率性記入表 (手順1)

変更要求	顧客の操作	チェック項目No	現状の応答時間 (ミリ秒)	予測時間 (ミリ秒)	予測変化率 (%増減)	実績 (ミリ秒)
検索条件を3つから4つに変更してほしい	検索条件を選択する					
	検索ボタンをクリックする					

ポイント：

記入する変更要求は**フィルター**に通して選別する

変更要求を記入する前の一仕事

### 3. 解決策（6）

---

#### □ フィルターとは？

EMOTを適用する

プロジェクト毎に重視する観点のこと

#### □ フィルターの効果

- ✓ 重視すべき変更要求を選別することが可能
- ✓ 過去に発生した問題と類似する問題の検出が可能

重視する変更要求だけ記入する

### 3. 解決策 (7)

#### □ 時間効率性記入表 (手順2)

変更要求	顧客の操作	チェック項目No	現状の応答時間 (ミリ秒)	予測時間 (ミリ秒)	予測変化率 (%増減)	実績 (ミリ秒)
検索条件を3つから4つに変更してほしい	検索条件を選択する					
	検索ボタンをクリックする					

ポイント：

変化を劣化と判断するのは顧客なので  
顧客目線で操作を分割する

顧客目線は開発者にもプラス

### 3. 解決策（8）

#### □ 時間効率性記入表（手順3／手順4）

変更要求	顧客の操作	チェック項目No	現状の応答時間(ミリ秒)	予測時間(ミリ秒)	予測変化率(%増減)	実績(ミリ秒)
検索条件を3つから4つに変更してほしい	検索条件を選択する					
	検索ボタンをクリックする					

ポイント：

時間効率性の劣化に繋がる**リソース変化点の有無**をチェック項目Noに記入する

**リソース変化点確認表を活用する**

### 3. 解決策 (9)

#### □ 時間効率性記入表 (手順5 / 手順6)

変更要求	顧客の操作	チェック項目No	現状の応答時間 (ミリ秒)	予測時間 (ミリ秒)	予測変化率 (%増減)	実績 (ミリ秒)
検索条件を3つから4つに変更してほしい	検索条件を選択する					
	検索ボタンをクリックする					

ポイント：

**変化時間が少ない & 変化率が高い**操作を検出  
すべく変化の**予測時間と予測変化率**を併記

**二つの観点で影響を正しく把握**

## 4. 検証（1）

---

### □ 検証方法

過去に発生した不具合7件でシミュレーション

### □ 検証内容

1. EMOTで時間効率性の変化を検出できるか
2. EMOTで導出した予測と実績の変化量の差
3. EMOTの適用に必要な作業工数



**EMOTの実力（精度）を検証**

## 4. 検証 (2)

### □ 検証結果

No	検出可否	予測時間 (ミリ秒)	実績 (ミリ秒)	予測時間と実績の差 (%)	検出の精度 (高・中・低)	作業工数 (h)
1	○	165,000	167,240	1.34	高	2.5
2	○	22,100	25,000	11.60	高	2.0
3	○	1,060	1,075	1.40	高	3.0
4	○	3,600	6,000	40.00	中	2.0
5	○	10,000	データ紛失	算出不可	中	3.0
6	○	9,803	20,000	50.99	低	0.5
7	○	14,400	80,000	82.00	低	3.0

7件中3件の事例で高精度に予測できた

時間効率性の変化は全事例で検出

## 4. 検証（3）

---

### □ 考察

- ✓ EMOTが**苦手とする事象**が明確になった
  - 待機時間の遅延やシステムリソース負荷など、**変化量が非線形に変動**するもの
- ✓ 開発経験が**2年程度**あれば**3時間程度**で**変化量の可視化が可能**
  - 不具合の改修工数と比較すると最も効果が出た事例で**9割工数を削減**できる計算

**苦手を克服し長所を伸ばす**

## 5. まとめ（1）

---

### □ 研究の成果

1. 時間効率性の劣化を数値で可視化する手法  
EMOT（イーモット）の提案
2. 過去の不具合事例 7 件全ての変化を可視化することができた
  - うち 3 件については具体的な影響を把握するに至った

**EMOTの有用性を確認できた**

## 5. まとめ (2)

---

### □ 今後の課題

1. EMOTの苦手とする部分の改善に努め適用可能な事象を拡大する
2. EMOTを稼動プロジェクトに適用して有用性を検証、運用上の問題を明らかにする

**EMOTをモットイーものにする**

EMOT

**ご静聴ありがとうございました**

2016年度ソフトウェア品質管理研究会(32SQiP)

第6分科会 Bグループ