

# 変更の影響範囲を特定するための 「標準調査プロセス」の提案

2014年ソフトウェア品質管理研究会  
[第6分科会Aグループ]

リーダー：宇田泰子（アンリツエンジニアリング株式会社）  
多田一成（アンリツエンジニアリング株式会社）  
川井めぐみ（サントリーシステムテクノロジー株式会社）  
伊藤友一（TIS株式会社）

# 1. 研究の動機

研究員の現場では、調査を行なっているにも関わらず、影響範囲の調査漏れによる不具合が後を絶たない

**なぜ変更の影響範囲を特定できないのか？**

## ■研究スタート当初、我々が思っていたこと

担当者がシステム全体を把握し切れていない

調査のやり方が、担当者の力量（経験）に依存

調査結果が担当者以外と共有できない

影響調査漏れ

どうにかして影響範囲の特定の抜け・漏れによる手戻りを防げないだろうか。。



## 2. 現状分析(1/2)

115件の不具合事例を分析

### ■不具合の内訳

- A. 変更要求
- B. 変更内容 (件数多)
- C. 調査方法 (手戻り大)
- D. 実装

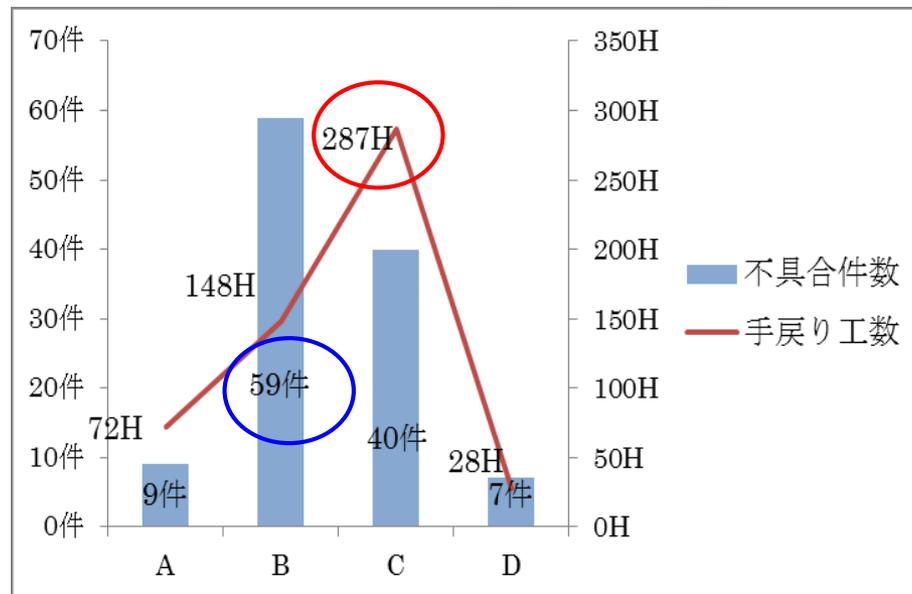


図1. 不具合件数と手戻り工数

### ■原因として考えられること

- A. 変更要求の理解不足 → XDDP(USDMM等)で改善
- B. 仕様変更の検討が不十分 → XDDP(変更要求仕様)で改善
- C. 調査方法・調査結果に不備 → 着目！
- D. 実装ミス → XDDP (レビュー強化等) で改善



## 2. 現状分析(2/2)

### ■「変更の影響範囲を特定する調査」方法のヒヤリング

#### (1) やったこと

実態を把握するため、開発者27名にアンケート実施

#### (2) わかったこと

- ・ 「調査」の一括りで色々な作業をしている（「ムリ」）

例. 基本的な構造の理解、変更箇所の特定制、影響箇所の特定制 など

- ・ 調査のしかたが人それぞれ（「ムラ」）

例. 調査結果の残し方（文書化していない、メモ程度。残す内容も様々）

- ・ 調査に「ムダ」が多い

例. 変更が変わる度、もしくは同じ変更でも調査タイミングの度に、箇所を調査する



いつ不具合が起きても、おかしくない！

### 3. 解決策(1/6)

#### ■ 調査プロセスの比較(初心者、熟練者、XDDP)

調査の問題点を明らかにするため、比較検討を実施

項目	初心者	熟練者	XDDP
変更要求	理解するプロセスがない (言葉のままに調査を実施)	XDDPと同じ	理解するプロセスがある
調査目的	明確になっていない (複数の調査を一緒に実施)	XDDPと同じ	明確になっている (複数の調査を別々に実施)
ソースコードの探索ルート	関数を順に辿り処理の経路を調べる	XDDPと同じ	一つの関数を最後まで確認した後、次の関数を確認する
調査資料の残し方	メモ書き、もしくは、なし	メモ書き 構造図、チャート	構造図、PAD
ソースコードの理解の仕方	ソースコードを読みながら理解	初心者と同じ	構造図やPADなどに変換し書き出してから理解

変更による影響範囲の特定で、抜けや漏れに気付きにくい原因は、  
ソースコードを調べる際の**探索ルート**に問題あり

### 3. 解決策(2/6)

#### ■ 探索ルートの問題

探索ルートB: 関数を順に辿り処理の経路を調べる

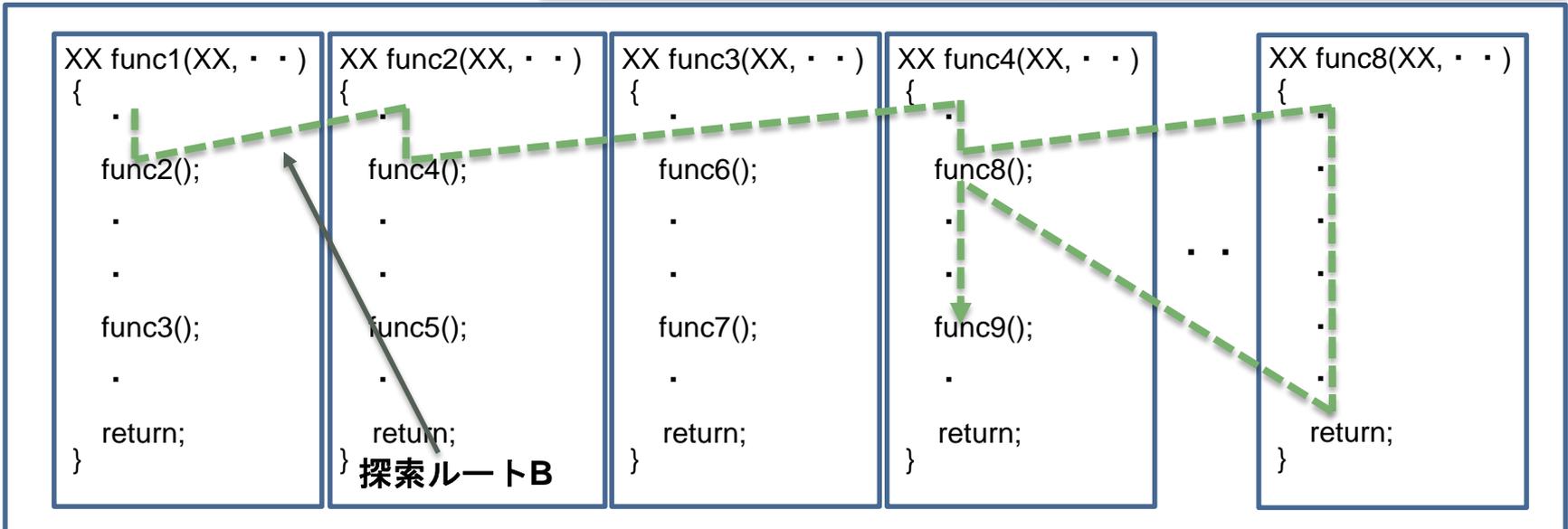


図2. ソースコードの探索

以下が内部データ保持の処理  
func1()  
// 受信データ設定  
func2()  
次に、各大項目レベルでの値格納処理に飛ぶ  
func4()  
.  
.

#### 探索ルートBの悪い点

- (1) 範囲が狭い (線で捉えている)
- (2) 目的/ステージを意識していない
- (3) 構造を把握できない
- (4) 活用できる調査結果が残らない

図3. 探索ルートBの成果物(メモ)

### 3. 解決策(3/6)

#### ■ 探索ルートの問題

探索ルートA: 個々の関数ごとに最後まで調べる(XDDP推奨)

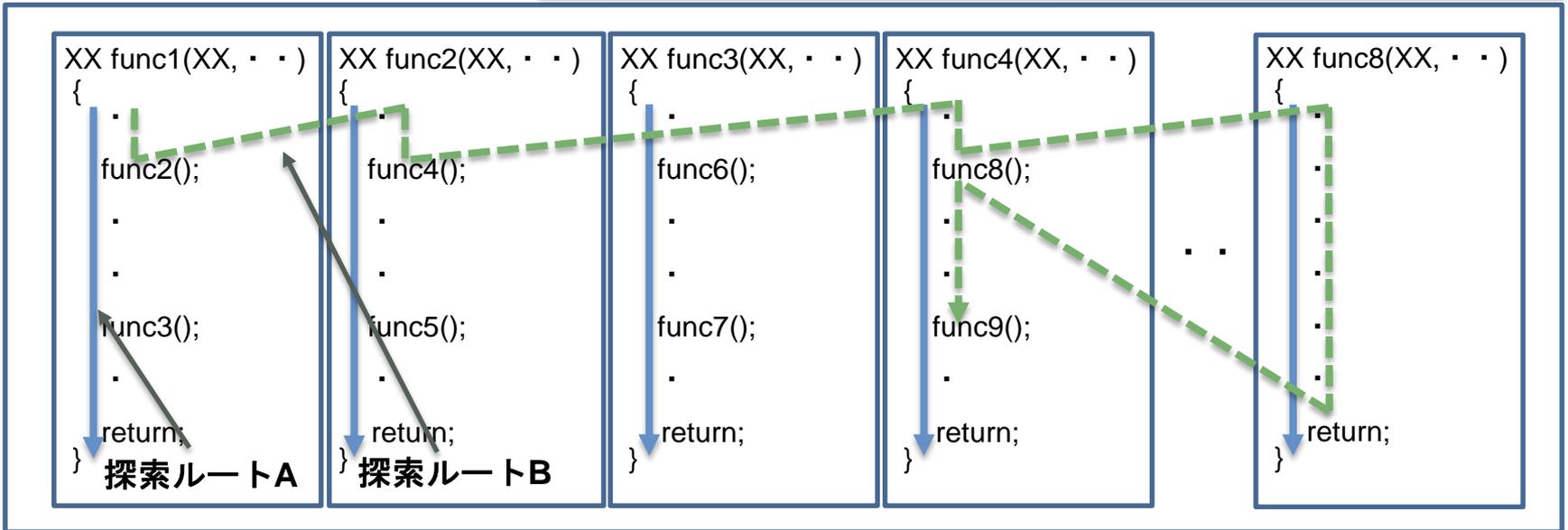


図2. ソースコードの探索

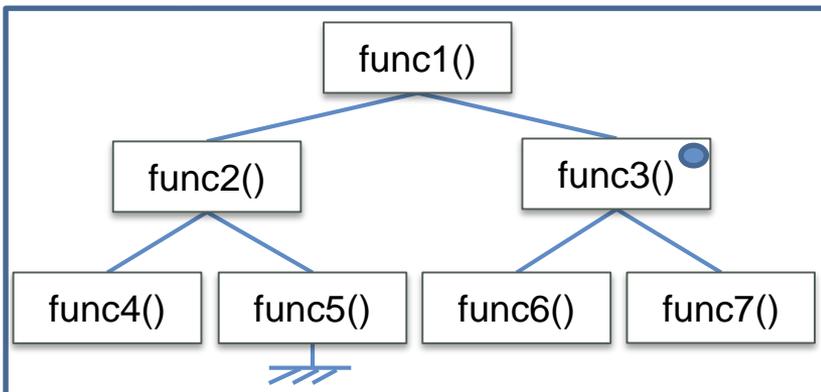


図4. 探索ルートAの成果物(構造図)

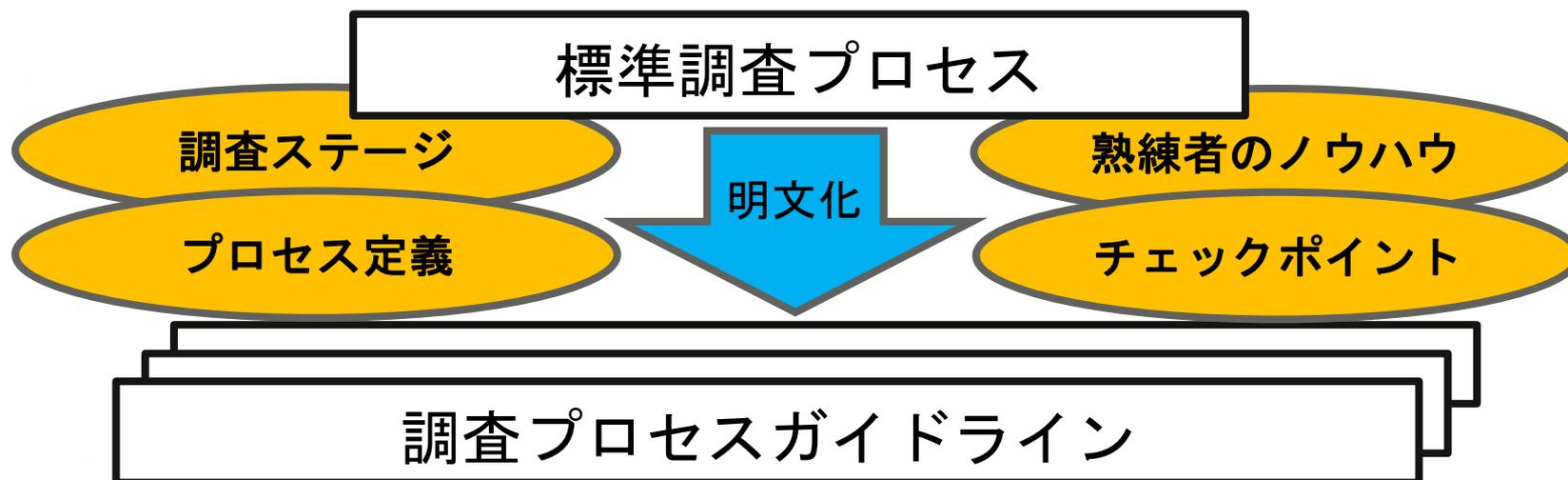
#### 探索ルートAの良い点

- (1) 漏れを起こす状況を改善できる
- (2) 調べる時間を短縮できる  
※1関数1分など、見積もりが可能となる
- (3) 全体の構造を把握できる
- (4) 調査結果を活用・再利用できる

### 3. 解決策(4/6)

#### ■ 問題解決のため「標準調査プロセス」を定義

プロセスの統一および調査結果記録のテンプレート作成



- ・変更による影響範囲の特定漏れを低減できる
- ・現場で標準調査プロセスを効果的に活用できる
- ・初心者の調査を支援できる
- ・初心者の育成に効果が期待できる

※現段階では、処理の流れなどの静的な構造の理解を目的としており、動的な振る舞い  
(例：状態遷移、タイミングチャート)の理解には対応していない

### 3. 解決策(5/6)

#### ■ 調査ステージの定義

No	調査ステージ	説明
1	事前調査	依頼内容を理解するためや、予め知っておくべきシステム構造や動作を把握するために実施する調査 ・調査範囲を事前に決める ・把握した構造などを成果物として残す ・変更箇所は探さない
2	変更箇所調査	変更要求を実現するためや、システムの変更箇所を特定するための調査 ※XDDPでいう「スペックアウト」に相当する ・ソースコードを理解しながら、変更箇所を探す ・ひとつおとり変更仕様が抽出できたら調査を終了する
3	影響調査	システムの変更によって、影響が及ぶ箇所を特定するための調査 ※XDDPでいう「スペックアウト」に相当する ・変更方法や変更箇所により影響を受ける箇所が変わるため、変更箇所調査が完了した後に実施する ・変更箇所があるとは限らない

調査ステージを定義することで、  
変更による影響範囲の特定漏れを低減できる

### 3. 解決策(6/6)

#### ■ 調査プロセスのステップ

##### ●ステップ1

基本パターン1か2を決定する

【基本パターン1】「事前調査」「変更箇所調査」「影響調査」を実施

【基本パターン2】「変更箇所調査」「影響調査」を実施

- ・変更箇所を特定できる
- ・構造や処理の流れを理解している

基本パターン1

基本パターン2

##### ●ステップ2

調査プロセスのテンプレート(調査テーマ記入シート)を記載する  
事前調査を実施する

事前調査

##### ●ステップ3

プロセスチェックポイントと調査実施事例(探索ルートと成果物)を参考にし、変更箇所調査と影響調査を実施する。

変更箇所調査

影響調査

Yes

No

- ・ソースコードを効率よく漏れなく調査
- ・変更による特定漏れを低減
- ・有用な資料が残せる

## 4. 解決策の評価(1/3)

---

### ■検証方法

「標準調査プロセス」と「調査プロセスガイドライン」の有用性を判断するために、不具合事例を用いてシミュレーションを実施

### ■検証内容

- (1) 調査のステージを意識して調査を行なうことができるか
- (2) 不具合事例の原因となった影響を受ける範囲や変更箇所を特定できるか
- (3) メモ書きではない調査結果が残せるか

## 4. 解決策の評価(2/3)

### ■結果

不具合事例	変更の影響範囲	調査のステージ	結果	成果物	調査工数	構造図作成工数
事例1	無知見	事前調査	特定できた	構造図	5時間	5時間 (64関数)
事例2	無知見	事前調査	特定できた	構造図	4時間	1.4時間 (70関数)
事例3	知見	事前調査 変更箇所調査	特定できた	構造図 フローチャート	3時間	1.8時間 (104関数)
事例4	知見	変更箇所調査	特定できた	フローチャート	2時間	作成していない
事例5	知見	事前調査 変更箇所調査	特定できなかった	構造図 フローチャート	1時間	0.5時間 (28関数)

- ・調査ステージを意識して調査ができた！
- ・5件中4件特定できた！
- ・構造図・フローチャートを残せた！

## 4. 解決策の評価(3/3)

---

### ■検証の考察

- (1) 調査のステージの意識づけで変更による影響範囲の特定漏れを低減できる！
- (2) 調査結果が残せ、レビューのインプットとなる！

### ■課題

- ・ 動的な振る舞い(例：状態遷移、タイミングチャート)は別の調査方法が必要となる

## 5. まとめ

---

### ■研究の成果

「標準調査プロセス」と「調査プロセスガイドライン」を作成した。その結果、以下の成果が得られた

- ・ 適切な調査のステージで適切な調査を行なえる
- ・ 処理の流れなど静的なケースにおける  
変更による影響範囲の特定漏れを低減できる
- ・ レビューのインプットになる調査結果を残せる

### ■今後の課題

動的な振る舞いの理解を目的とする調査方法についても着目し、「標準調査プロセス」と

「調査プロセスガイドライン」を改良していく

ご清聴ありがとうございました

変更の影響範囲を特定するための  
「標準調査プロセス」の提案

Unified investigating process to Detect the range Affected  
by specification changes

2014年ソフトウェア品質管理研究会  
(30SQiP-A)