

The DENSO logo is written in a bold, italicized, red sans-serif font.

**DENSO**

Crafting the Core

# D-Case記法を応用した 変更要求仕様書の提案

株式会社デンソークリエイト

柏原 一雄

松本 雄太

高田 修身

# 目次

- 1.はじめに
- 2.現状分析
- 3.課題提起
- 4.先行研究
- 5.解決策の提案
- 6.解決策の評価
- 7.おわりに

# 1. はじめに (1)

- 我々のプロジェクトでは、派生開発において、仕様の漏れ・誤りによる不具合の流出を防ぐため、USDmを導入した。
- 要求と仕様を階層構造で表現したことで、レビューで変更仕様の漏れ・誤りを検出できるようになった。
- しかし、レビューで変更仕様の漏れを見逃す事例を、完全になくすことはできなかった。
- 本発表では、変更仕様の漏れを検出しやすくするための、変更要求仕様書に対する“ひと工夫”を提案する。

# 1. はじめに (2)

## ■ USDM (Universal Specification Describing Manner)

[1] 清水吉男, 「要求仕様記述手法「USDM」ってどんなの? ~明日から使えるUSDMのエッセンス~」

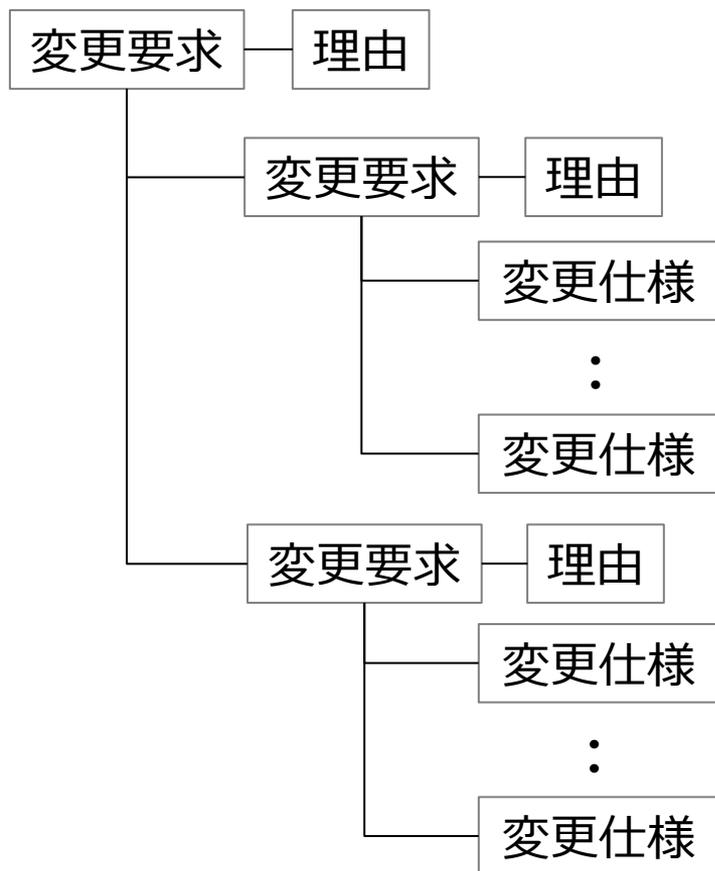
- USDMの特徴と派生開発での効果

USDMの特徴	派生開発での効果
要求と仕様の階層表現	変更の狙いや意図を「変更要求」で表現し, 具体的な仕様レベルの変更を「変更仕様」として階層で表現する. 当初の変更に対して影響を受けて <u>変更する箇所</u> も, その変更要求の下位層に表現することで, <u>関係がわかりやすくなる</u> .
要求と仕様を分けて表現	変更を仕様レベルで表現することで, <u>影響箇所に気付きやすくなる</u> .
要求に理由をつける	変更には必ず「理由」があり, それを明示することで <u>変更の意図</u> を正確に把握することができる.
Specifyを前提とする	変更には設計プロセスがないため, 「before / after」で変更要求と変更仕様を表現することで <u>変更の様子をイメージさせる</u> .

**派生開発において, 仕様漏れなどの問題を解決できる有効な手法**

# 1. はじめに (3)

## ■ 変更要求仕様書の構造



要求	ID-XXX	
	理由	
	説明	
□□□	ID-XXX-xx	
□□□	ID-XXX-xx	
□□□	ID-XXX-xx	
要求	ID-XXX	
	理由	
	説明	
□□□	ID-XXX-xx	
□□□	ID-XXX-xx	
□□□	ID-XXX-xx	

## 2. 現状分析 (1)

### ■ 変更要求仕様書の導入による効果

- 変更要求仕様書に対するレビューで問題を検出できるようになった。ただし、問題の見逃しは、ゼロにはなっていない。
- レビューでの見逃しやすい問題

対象	問題のタイプ <sup>o</sup>	レビューでの問題の見逃し有無
変更要求	漏れ	なし
	誤解釈	なし
<b>変更仕様</b>	<b>漏れ</b>	<b>あり</b>
	誤解釈	あり※
	衝突	なし

※変更仕様の誤解釈については、以下の論文で解決策を提案

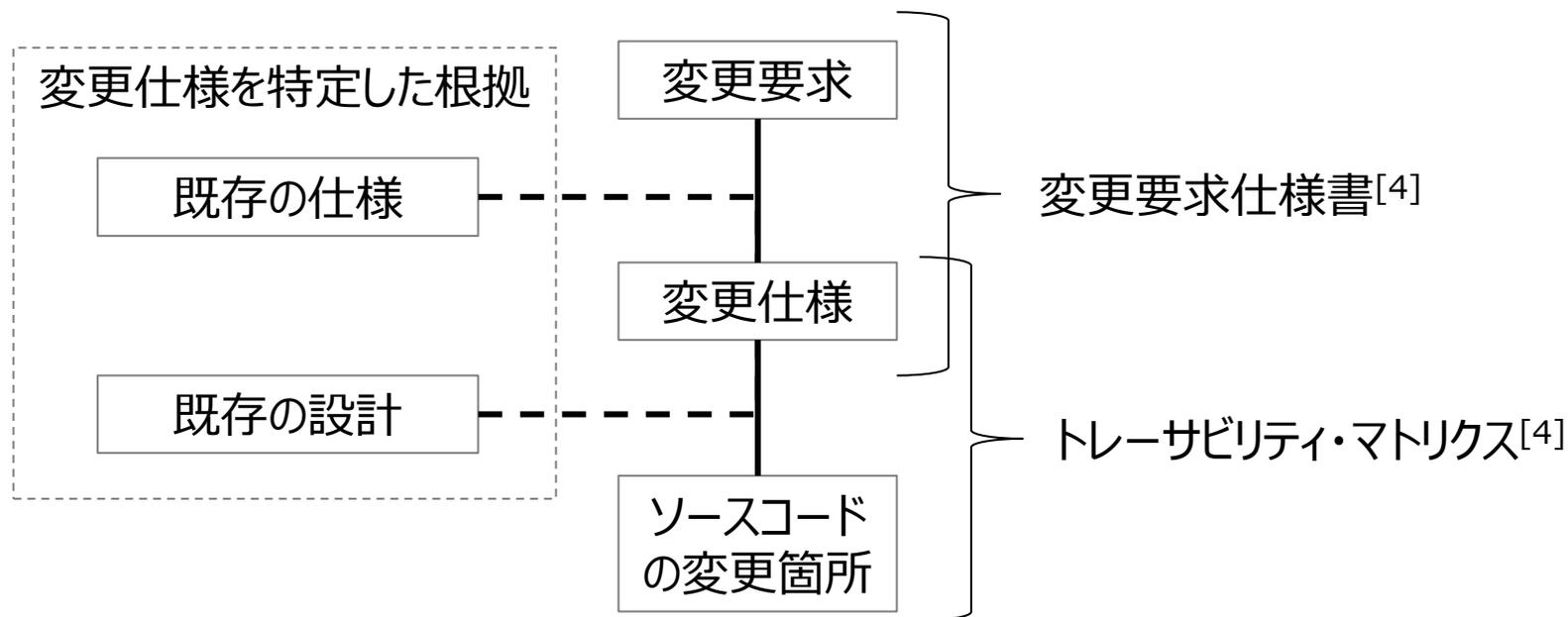
- 要求仕様の誤解釈を検出するDomain Word Modelingの提案[2]
- 要求仕様に対する形態素ベースレビューの提案[3]

**レビューで変更仕様の漏れを見逃す事例は、ゼロにはできていない**  
**本研究では、変更仕様の漏れを防止することを目的とする**

## 2. 現状分析（2）

### ■ 変更仕様の漏れの分類

- 「変更要求」に対する漏れ
- 「ソースコードの変更箇所」に対する漏れ
- 「既存の仕様・設計」に対する漏れ



**変更仕様を特定した根拠となる既存の仕様・設計は、  
変更要求仕様書とトレーサビリティ・マトリクスには明確に示されない**

## 2. 現状分析 (3)

### ■ レビューで変更仕様の漏れを見逃す事例

#### 【変更要求仕様書】

要求	ID-XXX	〇〇の送信処理が失敗したときに、△△の処理を中止する。
	理由	・・・
	説明	・・・
□□□	ID-XXX-xx	AAAの条件を満たす場合に、△△の処理を継続せず、キャンセル処理をする。
□□□	ID-XXX-xx	BBBの条件を満たす場合に、△△の処理を継続せず、キャンセル処理をする。



#### 【変更仕様を特定した根拠】

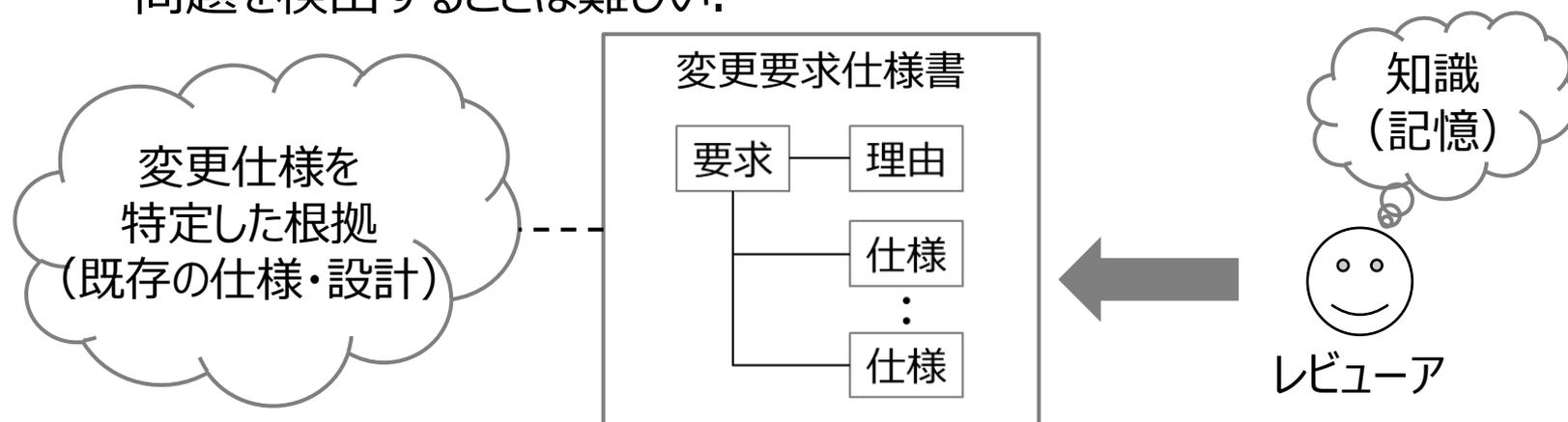
〇〇の送信処理が失敗する要因には、  
AAAとBBBがある。

**変更仕様を特定した根拠となる既存の仕様・設計を知らなければ、  
レビューは変更仕様の漏れを検出できない**

## 2. 現状分析（4）

### ■ 変更要求仕様書のレビュー

- 変更要求仕様書に示されている「理由」「要求」「仕様」の内容と関係に対して、問題を検出していた。  
変更要求仕様書に明確に示されていないこと（既存の仕様・設計との関係）に対して、問題を検出することは難しい。
- レビューアの知識（記憶）をもとに、問題を検出していた。  
レビューアに既存の仕様・設計に関する知識（記憶）がなければ、問題を検出することは難しい。



**変更仕様を特定した根拠が明確に示されていないため、  
部分理解<sup>[4]</sup>の状態のレビューアでは、問題を検出することは難しい**

### 3. 課題提起

#### ■ 目的

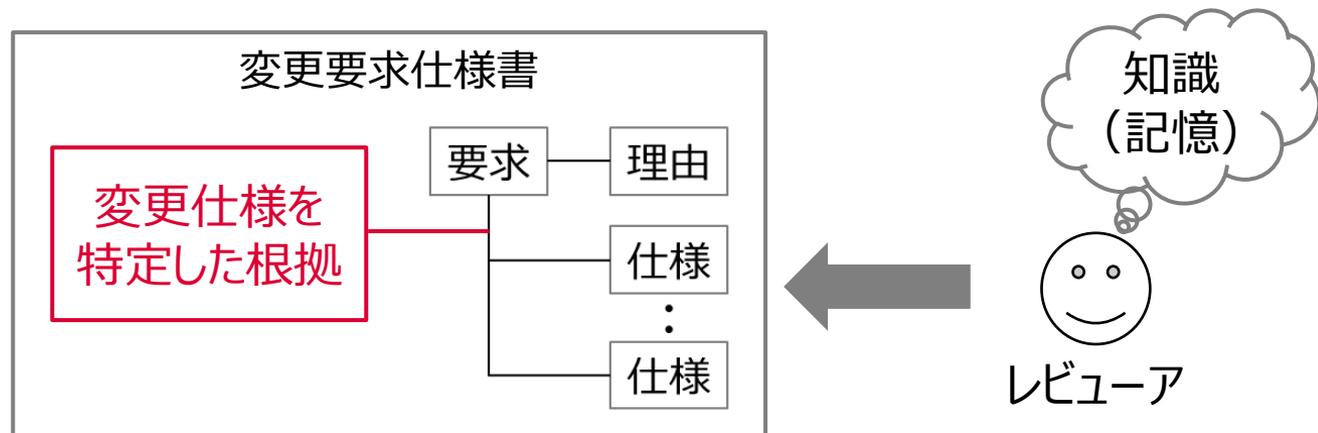
- 変更仕様の漏れをレビューで見逃すことを防ぐ。

#### ■ 課題

- レビューにおいて、「変更仕様を特定した根拠」が確認しやすくなるように、変更要求仕様書を改善する。

#### ■ 課題解決の前提

- レビューアも部分理解の状態である。
- 変更要求仕様書を導入済みのプロジェクトでの改善である。

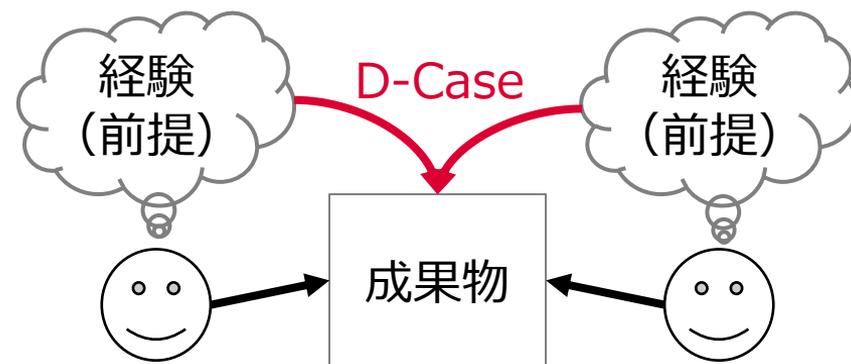


## 4. 先行研究 (1)

### ■ D-Caseを用いたレビューを見える化する方法の導入事例

[5] 小林展英, 「D-Caseを用いたレビューを見える化する方法の導入事例」, 12th WOCS2, 2015

- 上位者の経験が成果物に上手く反映できていない
  - 担当者は成果物の妥当性を論理的に説明できず, 上位者は明確な判断基準に基づいた指摘ができていない
- 経験を見える化して相手と共有する手段としてD-Caseを導入
  - 双方の経験 (常識, 語彙等) が揃えば相手と正しく情報を共有できる
- D-Caseを用いたレビュー方法
  1. 前提となる文書を揃える
  2. 説明の構造を設計する
  3. 成果物を証拠に紐づける
  4. D-Caseで説明する



**経験と同じく「変更仕様を特定した根拠」は成果物に反映できていない  
「変更仕様を特定した根拠」を共有するために, D-Caseを応用する**

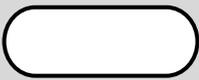
## 4. 先行研究 (2)

### ■ D-Case記法

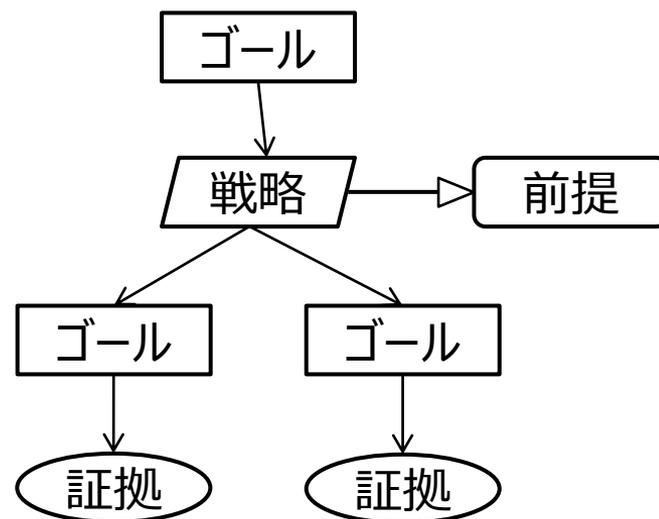
[6] 小林展英, 「D-Case入門」, Embedded Technology WEST 2015, 2015

[7] D-Case部会, 「はじめてのD-Case」, 一般社団法人 ディペンダビリティ技術推進協会, 2018

【GSNのノード】

ノード	記号	説明
ゴール		ステークホルダ間で合意したい主張
戦略		上位のゴールの分解の仕方を説明
前提		議論の前提となる情報
証拠		ゴールが達成できていることを示す証拠 テスト結果などのモノ, ドキュメント

【D-Caseの木構造】



- 「ゴール」は「前提」をもとに「戦略」に基づきサブゴールに分解する。
- 前提ノードがあることにより論証する対象や範囲, 前提条件や制約などを示し明確にすることができる。

**D-Caseの木構造を参考に, 変更要求仕様書の構造を見直す**

## 4. 先行研究 (3)

### ■ スペックアウト

[8] 清水吉男, 「派生開発」を成功させるプロセス改善の技術と極意, 技術評論社, 2007

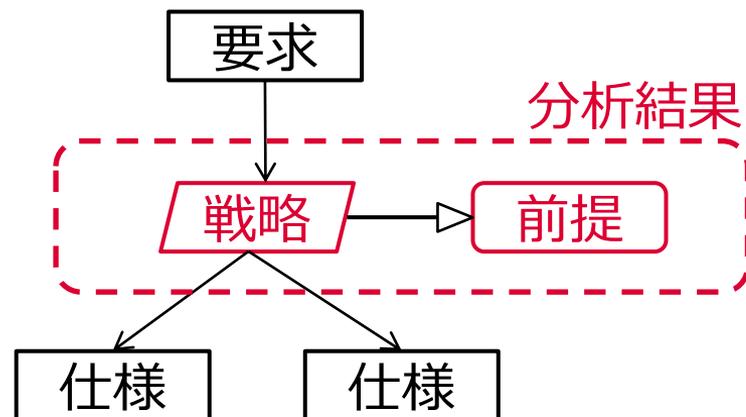
- ソースコードから必要な設計書を起こす行為を「スペックアウト」と呼ぶ.
- 変更要求に対して変更仕様を抽出する方法
  - ベースのソースコードに対応する機能仕様書や操作仕様書などの文書から変更箇所を拾う
  - 関係しそうな箇所のソースコードを読んで, 変更する箇所を拾う
  - ソースコードの関係箇所をスペックアウトしながら, 変更する箇所を拾う
- ソースコードから変更仕様を抽出する作業は, 必ず「変更要求単位」で行う. 「スペックアウト」によって作成された成果物は, 変更要求が所属する機能に対応した資料となる.
- 変更要求に対してスペックアウトで作成した成果物は, 公式にはレビュー対象になっていないが, このソフトウェアシステムを調査した担当者が認識を間違えたままではこのあとで正しい変更作業はできない.

**「スペックアウト」で作成した既存仕様・設計を示す成果物は,  
「変更仕様を特定した根拠」として変更要求に紐づけ, レビューする**

# 5. 解決策の提案（1）

## ■ 課題の解決方針

- 変更仕様が特定された根拠を示すために、D-Case記法を参考に、「戦略」と「前提」を要求と仕様の階層構造に要素として加える。
- 変更仕様が特定された根拠とは、「何をもとに、どのようにして、その変更仕様を特定したか」を示す情報であると捉える。
- 変更要求仕様書の要素とする「前提（何をもとに）」と「戦略（どのようにして）」を合わせて「分析結果」と呼ぶ。
- 新たな成果物を作成するのではなく、変更要求仕様書の構造を変更する方針とすることで、改善による工数の増加を最小限とする。



## 5. 解決策の提案（2）

### ■ 変更要求仕様書の形式の改善点

- 変更要求仕様書に分析結果欄を追加する。
- 分析結果欄には、「何をもとに（前提）, どのようにして（戦略）, その変更仕様を特定したか」を示す。
- 前提となる調査結果（「スペックアウト」で作成した成果物）などは、別ファイル・別シートに示してもよい。  
その場合、分析結果欄には、前提となる情報が示されたファイル・シートなどが特定できる情報を記載する。

要求	ID-XXX	
	理由	
	説明	
	分析結果	
□□□	ID-XXX-xx	
□□□	ID-XXX-xx	
□□□	ID-XXX-xx	

**変更要求に対して変更仕様の過不足がないと言える根拠を示す**

## 5. 解決策の提案（3）

### ■ 変更要求仕様書のイメージ

要求	ID-XXX	〇〇の送信処理が失敗したときに、△△の処理を中止する。
	理由	・・・
	説明	・・・
□□□	ID-XXX-xx	AAAの条件を満たす場合に、△△の処理を継続せず、キャンセル処理をする。
□□□	ID-XXX-xx	BBBの条件を満たす場合に、△△の処理を継続せず、キャンセル処理をする。



要求	ID-XXX	〇〇の送信処理が失敗したときに、△△の処理を中止する。
	理由	・・・
	説明	・・・
	分析結果	ソースコードから送信処理が失敗する条件を漏れなく洗い出し、その条件全てでキャンセル処理をする。 【前提】 送信処理が失敗する条件には以下の2つがある。 ・ AAA ・ BBB ※詳細は「送信処理失敗要因調査結果.xls」に示す。
□□□	ID-XXX-xx	AAAの条件を満たす場合に、△△の処理を継続せず、キャンセル処理をする。
□□□	ID-XXX-xx	BBBの条件を満たす場合に、△△の処理を継続せず、キャンセル処理をする。

# 6. 解決策の評価（1）

## ■ 評価方法

- 改善した変更要求仕様書を，実開発（4案件）に導入し，評価する。
- 評価観点
  - 分析結果欄に「変更仕様を特定した根拠」として必要なことが記載できるか
  - レビューで「変更仕様を特定した根拠」に対する問題が検出できるか
- 集計データ ※集計はレビュー後の成果物に対して実施

データ名	説明
要求数	変更要求の数
仕様数	変更仕様の数
分析結果数	分析結果が記載された変更要求の数
有効な分析結果数	「変更仕様を特定した根拠」にあたる情報が記載された分析結果の数
レビュー指摘有無	レビューでの分析結果に対する指摘の有無

- 前提
  - レビューアは，以下の条件を満たす
    - D-Caseのトレーニングを受講し，実開発でD-Caseを利用した経験あり
    - ベースソフトの開発に携わっていない（部分理解の状態）

## 6. 解決策の評価（2）

### ■ 評価結果（1）

- 案件毎にデータ集計の結果を示す。

案件 No	要求数	仕様数	分析結果数	有効な分析結果数	レビュー指摘有無
1	10	25	12	12	あり
2	12	23	14	13	あり
3	7	9	7	2	あり
4	3	8	3	3	あり

- 全ての変更要求に対して、分析結果が記載された。
- 全ての案件で、分析結果欄には、「変更仕様を特定した根拠」にあたる情報が記載された。  
ただし、「変更仕様を特定した根拠」に位置付けられない情報が記載されているケースもあった。（案件2：1/14件，案件3：5/7件）
- 全ての案件で、分析結果に対するレビュー指摘があった。

## 6. 解決策の評価 (3)

### ■ 評価結果 (2)

- 分析結果をパターン毎に分類し、件数を集計した結果を示す。

変更要求の種類	分析結果のパターン	件数
機能の変更	既存のソフトでの動作の確認※	6
機能の変更	複数の実現案の比較	4
機能の変更	既存の仕様の調査	12
機能の変更	既存の設計の調査	6
性能の改善	性能要求未達箇所の調査	1
結合部品の変更	結合部品の変化点及びその影響の調査	2
不具合の修正	不具合の原因の調査	5

※既存のソフトでの動作の確認・・・既存のソフトで要求を満たすことができないかを確認する

- 「既存のソフトでの動作の確認」は、変更要求を定義する前に実施しておくべきことであり、「変更仕様を特定した根拠」として分析結果欄に記載すべき情報ではないと考えられる。

## 6. 解決策の評価（4）

### ■ 結果の考察

- 変更仕様を特定した根拠を示すために、分析結果欄を活用できた。
  - ただし、変更要求仕様書に分析結果を示す必要がない変更要求も存在していた。変更要求仕様書のルールとして、「分析結果を記載不要と判断する条件」を明確にすることが求められる。
- 変更要求仕様書に分析結果欄があることで、レビューで「変更仕様を特定した根拠が“ない”または“不適切”」であることに気づきやすくなった。
  - ただし、変更要求仕様書に示される「戦略（どのようにして変更仕様を特定するか）」は、分析対象とする変更要求によって変わる。レビューアには、最適な「戦略」であるかを判断することが求められる。
- 変更仕様が特定された根拠が残っていることで、変更仕様の漏れによる不具合の発生時に、変更仕様が漏れた原因を特定しやすくなった。
- レビューアは、D-Caseのトレーニングを受講し、実開発でD-Caseを利用したことがあった。D-Caseに関する知識・経験がないメンバがレビューアの場合に、効果が得られるかは確認できていない。

**残課題はあるが、実開発において考案手法の効果が確認できた**

# 7. おわりに (1)

## ■ まとめ

- 変更要求仕様書に、変更仕様が特定された根拠が示されておらず、レビューで変更仕様の十分性を確認しにくい状態であった。
- D-Case記法を参考に、要求と仕様の階層構造に、「戦略」と「前提」を要素として加える方針で、問題の解決を図った。
- 変更要求仕様書に分析結果欄を設け、「何をもとに、どのようにして、その変更仕様が特定したか」を示すルールとした。
- 変更要求仕様書の形式を改善したことで、レビューで「変更仕様が特定した根拠が“ない”または“不適切”」という問題に気づきやすくなった。
- 考案手法により、変更仕様の漏れを防止する効果が高まることが期待される。

**成果物の構造を少し変えただけで、レビューが変わる  
見えていなかったことを見えるようにすると、レビューが変わる**

## 7. おわりに (2)

### ■ 今後の課題

- 実開発への考案手法の適用を継続し、不具合の流出を防止する効果を確認する。また、不具合の原因分析の効率化という視点での効果も確認する。
- 変更要求仕様書の分析結果欄を無駄に埋める作業を防ぐために、分析結果を記載不要と判断する条件を定義する。
- 問題を検出しやすくするために、評価で明らかにした分析結果のパターンをもとに、自由記述形式である分析結果欄の記述ルールを検討する。 [8][9][10]
- 変更要求仕様書のレビューアに必要なスキルとトレーニングを明らかにする。

# 参考文献

- [1] 清水吉男, 「要求仕様記述手法「USDM」ってどんなの? ~明日から使えるUSDMのエッセンス~, [https://swquality.jp/temp/nagasakiqdg15\\_usdm.pdf](https://swquality.jp/temp/nagasakiqdg15_usdm.pdf)
- [2] 柏原一雄, 長井亘, 不破慎之介, 林香織, 石川冬樹, 栗田太郎, 「要求仕様の誤解釈を検出する Domain Word Modeling の提案」, ソフトウェア・シンポジウム2020, 2020
- [3] 柏原一雄, 新留光治, 李路雲, 鈴木淳, 松原潤弥, 不破慎之介, 「要求仕様に対する形態素ベースレビューの提案」, ソフトウェア・シンポジウム2021, 2021
- [4] 清水吉男, 「[改訂第2版] [入門+実践]要求を仕様化する技術・表現する技術 ~仕様が書けていますか?」, 技術評論社, 2010
- [5] 小林展英, 「D-Caseを用いたレビューを見える化する方法の導入事例」, 12th WOCS2, 2015
- [6] 小林展英, 「D-Case入門」, Embedded Technology WEST 2015, 2015
- [7] D-Case部会, 「はじめてのD-Case」, 一般社団法人 ディペンダビリティ技術推進協会, 2018
- [8] 清水吉男, 「「派生開発」を成功させるプロセス改善の技術と極意」, 技術評論社, 2007
- [9] 山本稔郎, 中島毅, 「USDM形式要求仕様記述パターンの提案とその評価」, 第81回全国大会講演論文集 2019巻 1号, p.261-262, 2019
- [10] 南伸二, 宮田朋人, 山本稔郎, 中島毅, 「USDM形式の要求記述の曖昧さを軽減する記述構文パターンの提案とその評価」, ソフトウェアエンジニアリングシンポジウム2019論文集, p.145-152, 2019
- [11] 山本修一郎, 猿渡卓也, 「議論パターンによるシステム高信頼性保証知識の再利用」, 人工知能学会第二種研究会資料 2014巻 (2014) KSN-014号, 2014

***DENSO***

Crafting the Core