

ソフトウェアテストシンポジウム 2012 東京
(JaSST' 12 Tokyo)
セッション B5-1

**TOC思考プロセスを用いた
テスト工程の課題分析と改善
- テストにおけるトレーサビリティ確保の落とし穴 -**

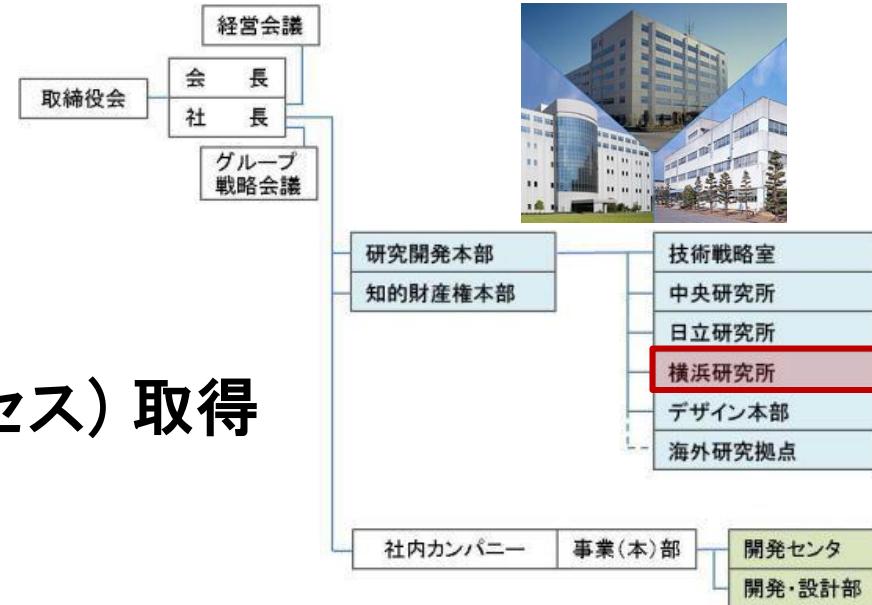
2012/1/26

株式会社 日立製作所 横浜研究所
システム開発研究部

○八木 将計, 小川 秀人

■ ハ木 将計(やぎ まさかず)

- (株)日立製作所 横浜研究所
システム開発研究部 研究員
- JSTQB Foundation Level 取得
- TOC-ICO登録 Jonah(思考プロセス) 取得



■ 所属部署のミッション

- 組込みソフトウェアの生産性・品質向上技術の研究開発
「ソフトウェアを品質良く簡単に作るにはどうすべきか？」

■ 研究内容

- ソフトウェア検証, SPL(Software Product Line),
モデル駆動開発, モデル検査, etc...

(テストプロセス上の)
「改善のための事象」

が、

「様々な理由」

により、

悪影響を及ぼす

ことがある

TOC思考プロセスを用いたテスト工程の課題分析と改善 - テストにおけるトレーサビリティ確保の落とし穴 -

Contents

- 1.背景と目的
- 2.TOC思考プロセスとは
- 3.TOC思考プロセスによる分析
- 4.改善策の策定
- 5.改善策の試行・評価
- 6.考察
- 7.まとめ

TOC思考プロセスを用いたテスト工程の課題分析と改善 - テストにおけるトレーサビリティ確保の落とし穴 -

Contents

► 1.背景と目的

2.TOC思考プロセスとは

3.TOC思考プロセスによる分析

4.改善策の策定

5.改善策の試行・評価

6.考察

7.まとめ

■ テストプロセス改善の対象組織

- 「ある組込み製品」を開発している事業部
- 5製品を開発・製造・販売している
- SEPG(Software Engineering Process Group)があり、5製品のソフト開発を管轄する
- 各製品にソフト設計部と品質保証部(QA)がある
- 「テスト」としては、
設計部によるテスト(Verification)と
QAによる妥当性確認(Validation)の二つがある
- 設計テストは、設計者自身が作成・実施している
#テスト専門部隊は置いていない

■ テストプロセス改善の目的

- QAの指摘不具合を削減すること
→ 設計内での抽出不具合を増加すること

■ 報告者の立ち位置

- SEPGの依頼で、コンサルタントとしてテストプロセス改善を支援

■ 進め方

- 各製品のソフト設計者とQAからなるワーキンググループを組織(計16名)
- 課題分析と改善策の検討・適用

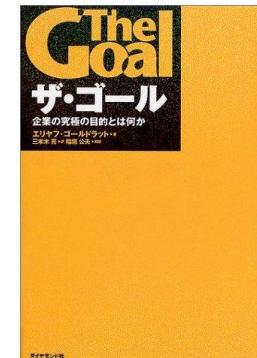
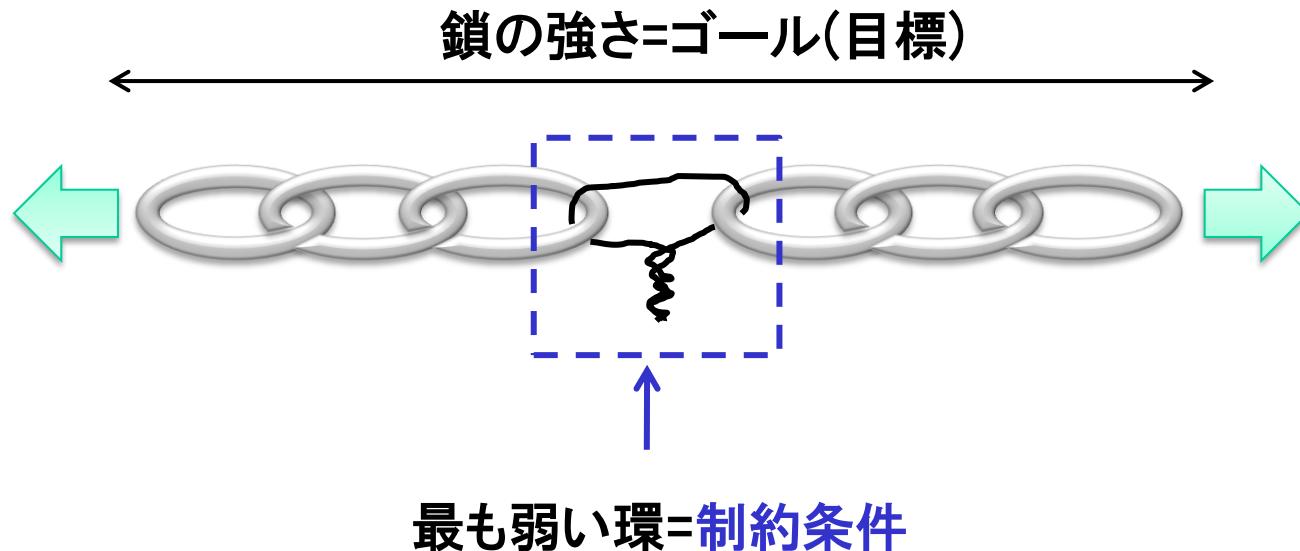
TOC思考プロセスを用いたテスト工程の課題分析と改善 - テストにおけるトレーサビリティ確保の落とし穴 -

Contents

- 1.背景と目的
- **2.TOC思考プロセスとは**
- 3.TOC思考プロセスによる分析
- 4.改善策の策定
- 5.改善策の試行・評価
- 6.考察
- 7.まとめ

■ TOC(Theory of Constraints(制約条件の理論))とは？

- イスラエル出身の物理学者エリヤフ・ゴールドラット博士が提唱した理論体系
- 組織の目標達成を決定づける制約条件を中心に改善を行い最小の努力で最大の効果をあげる経営理論



出典：エリヤフ・ゴールドラット著『ザ・ゴール ー企業の究極の目的とは何か』、ダイアモンド社、2001。

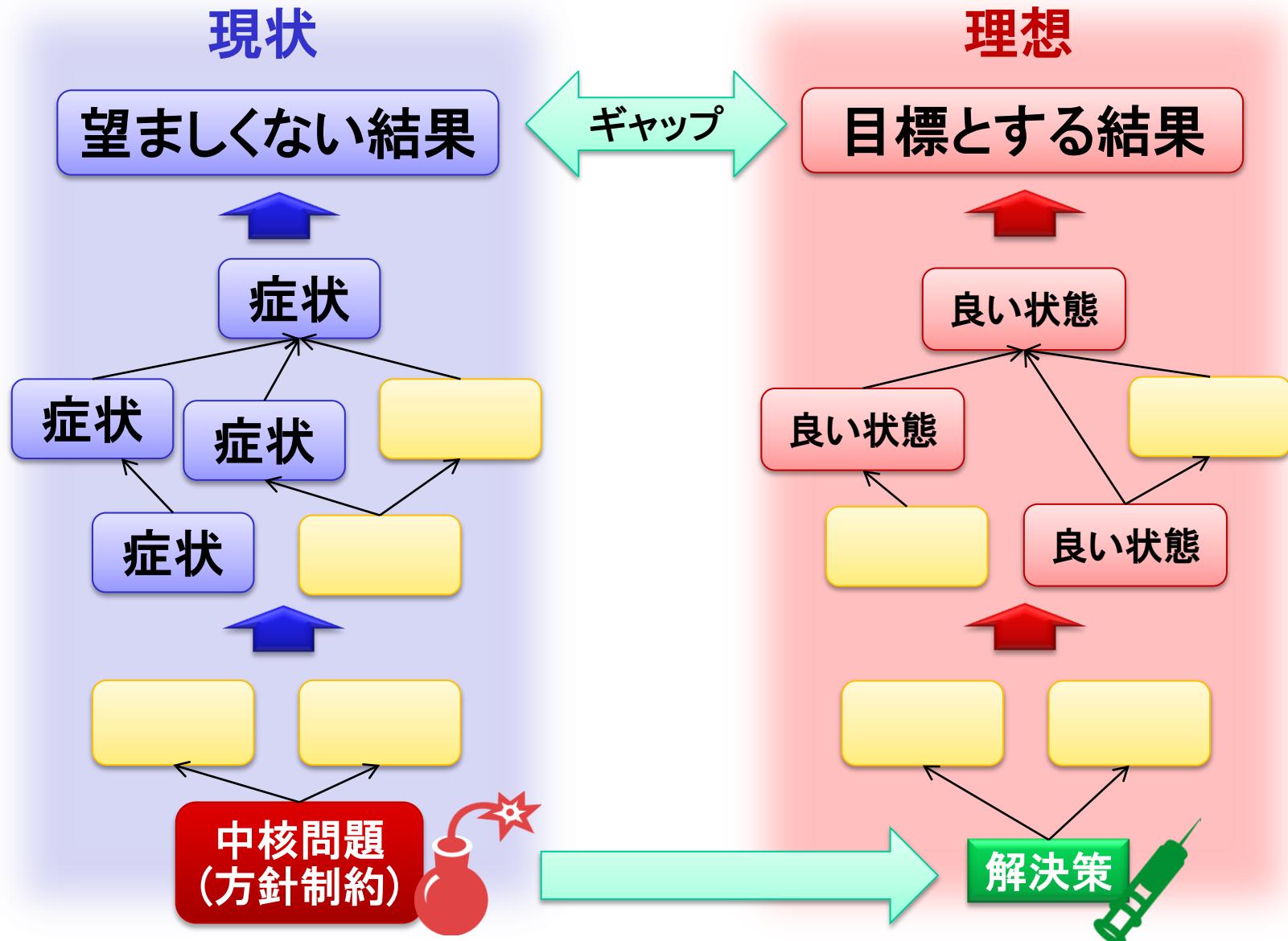
■ TOC思考プロセスとは？

- 組織の抱える個々の「問題」を一つ一つ対策するのではなく、全体を考慮して「中核問題」を改善することで、全体の成果を大きく向上する問題解決手法
- 「中核問題」=「方針制約」の改善に注力する

※ 方針制約：組織の目標達成を阻害している方針や文化、
ものの見方、価値観、評価基準など

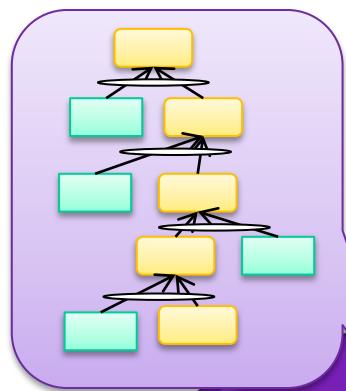


対象組織についても、テストプロセスにおける複数の問題が表出していたため、
TOC思考プロセスが有効であると判断した



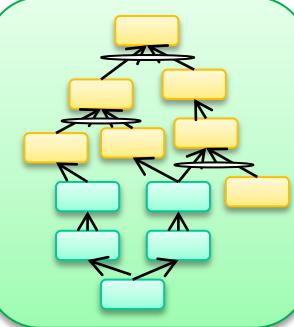
2-4 TOC思考プロセスの5つのツリー

移行ツリー

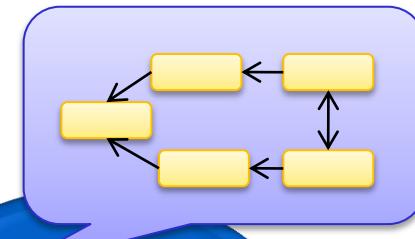


(1)何を変えるか?
What to change?

現状構造ツリー



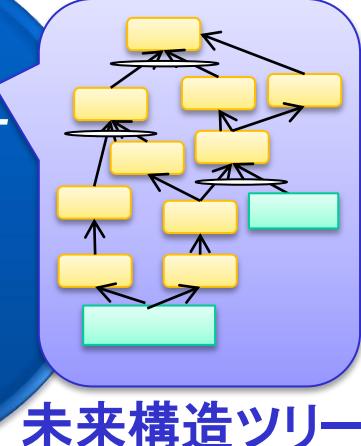
対立解消図



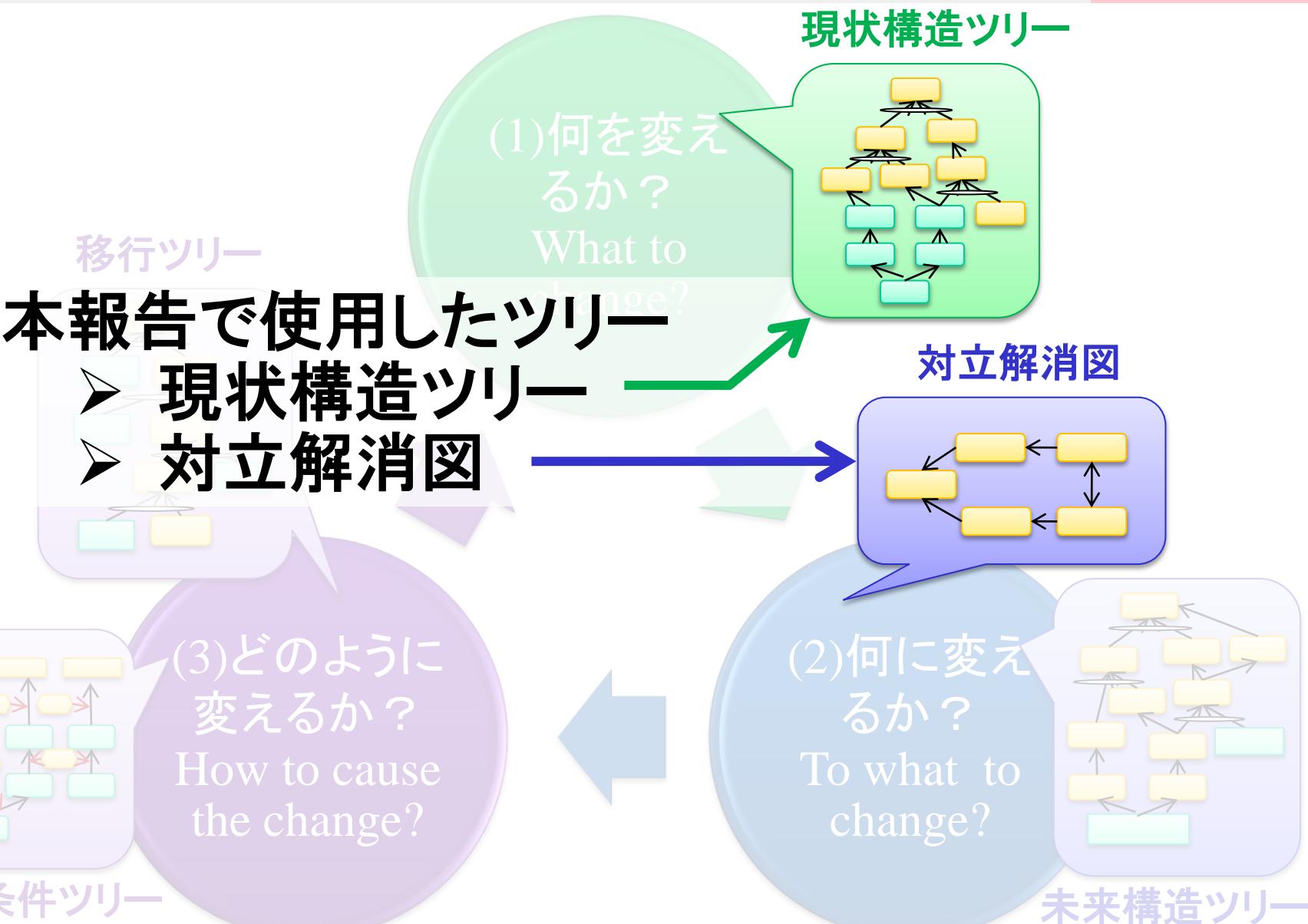
(3)どのように
変えるか?
How to cause
the change?

前提条件ツリー

(2)何に変えるか?
To what to
change?



2-4 TOC思考プロセスの5つのツリー



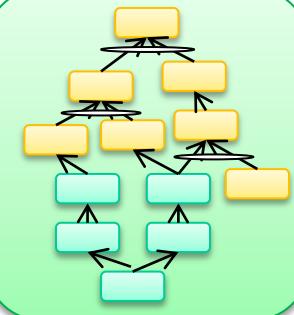
TOC思考プロセスを用いたテスト工程の課題分析と改善 - テストにおけるトレーサビリティ確保の落とし穴 -

Contents

- 1.背景と目的
- 2.TOC思考プロセスとは
- 3.TOC思考プロセスによる分析
- 4.改善策の策定
- 5.改善策の試行・評価
- 6.考察
- 7.まとめ

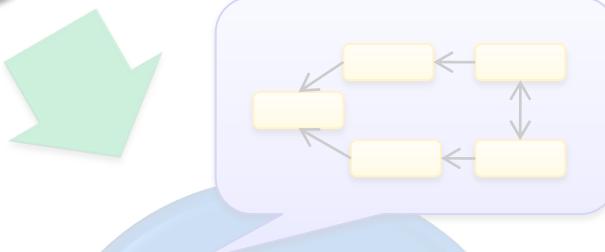
現状構造ツリー

(1)何を変えるか?
What to change?



対立解消図

(2)何に変えるか?
To what to change?



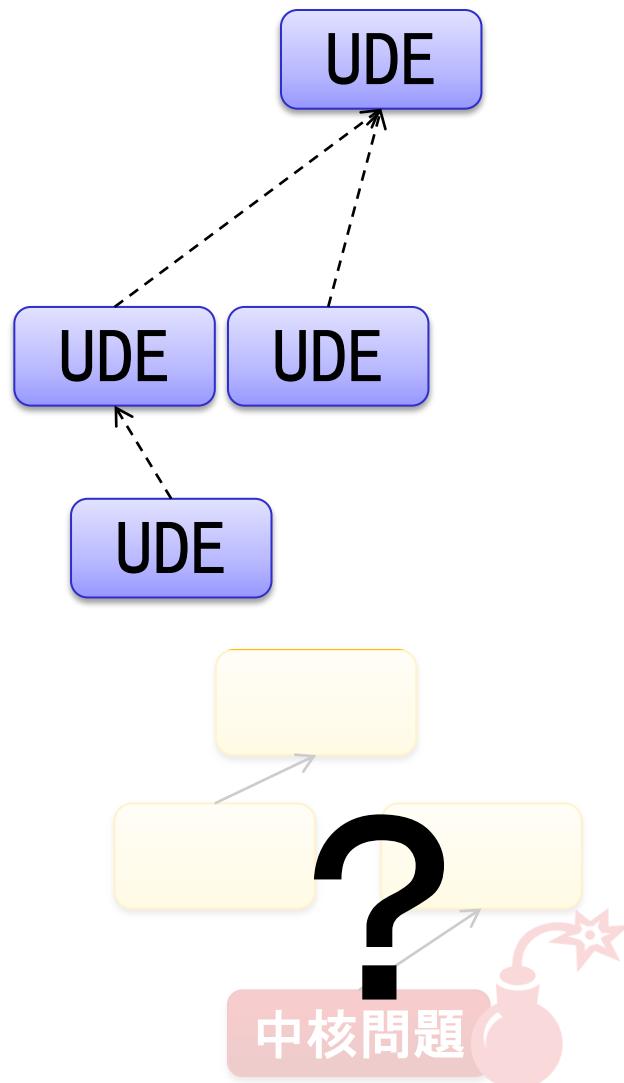
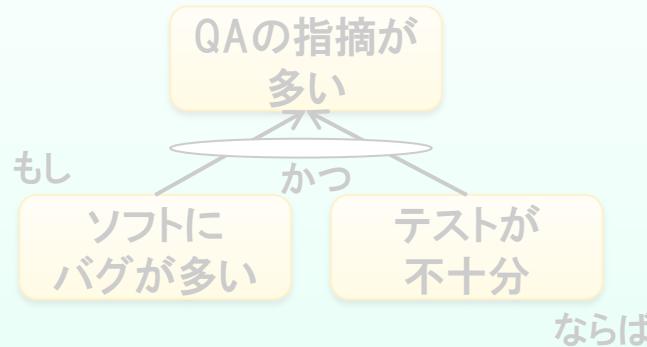
(3)どのように変えるか?
How to cause the change?



- ① 表出している症状
(UDE:UnDesirable Effect)
を洗い出して、仮に因果
関係をつなぐ

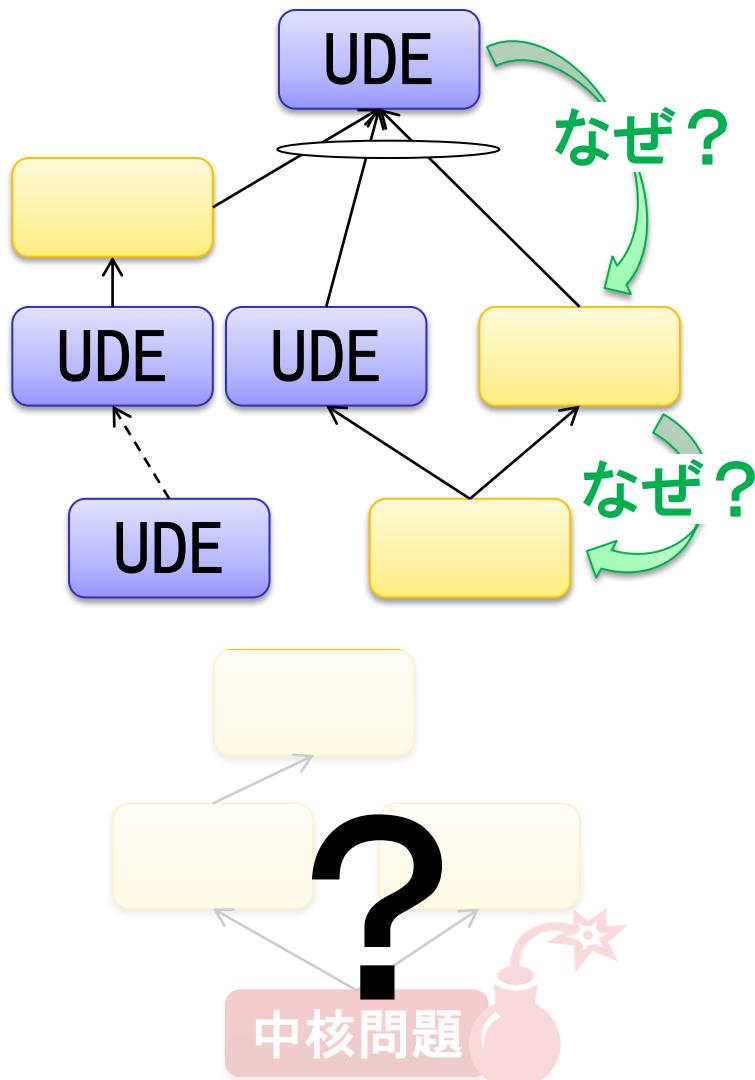
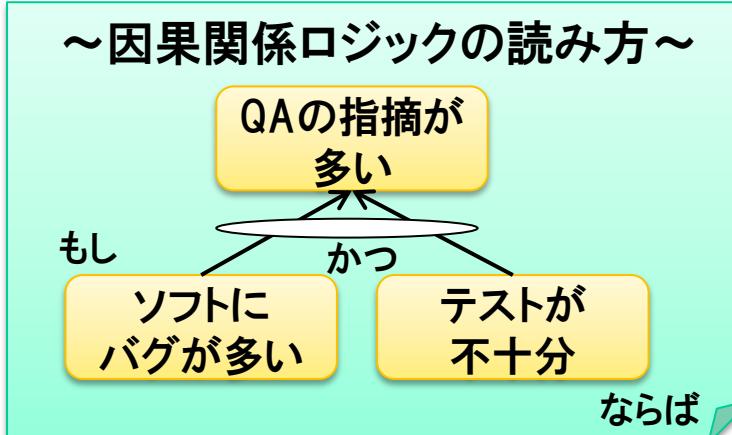
- ② UDEから因果関係を補完
しながらたどっていって…

～因果関係ロジックの読み方～



- ③ 「中核問題」を導出する

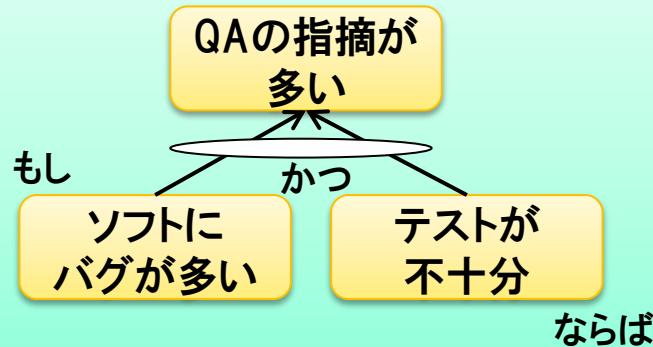
- ① 表出している症状
(UDE:UnDesirable Effect)
を洗い出して、仮に因果
関係をつなぐ
- ② UDEから因果関係を補完
しながらたどっていって…



- ③ 「中核問題」を導出する

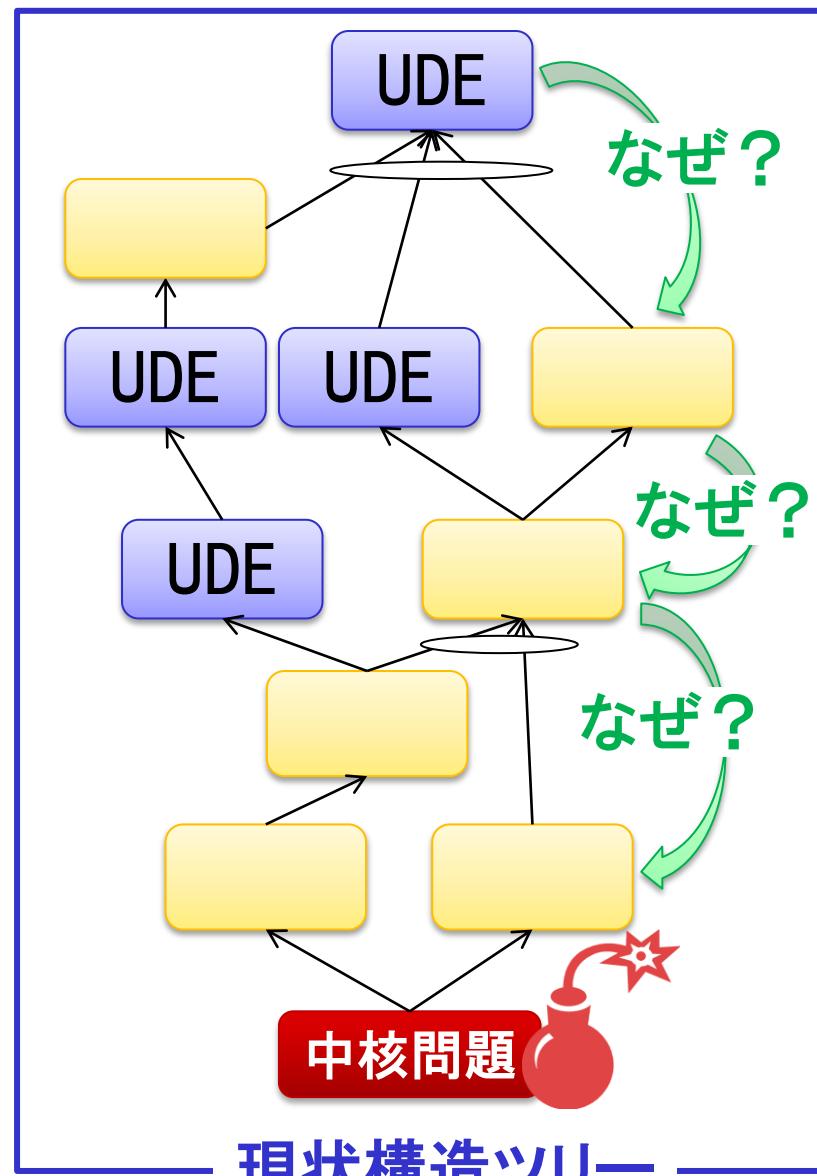
- ① 表出している症状
(UDE:UnDesirable Effect)
を洗い出して、仮に因果
関係をつなぐ
- ② UDEから因果関係を補完
しながらたどっていって…

～因果関係ロジックの読み方～



① 「中核問題」を導出する

※論文ではこの方法とは別の方法(3クラウド法)を使用

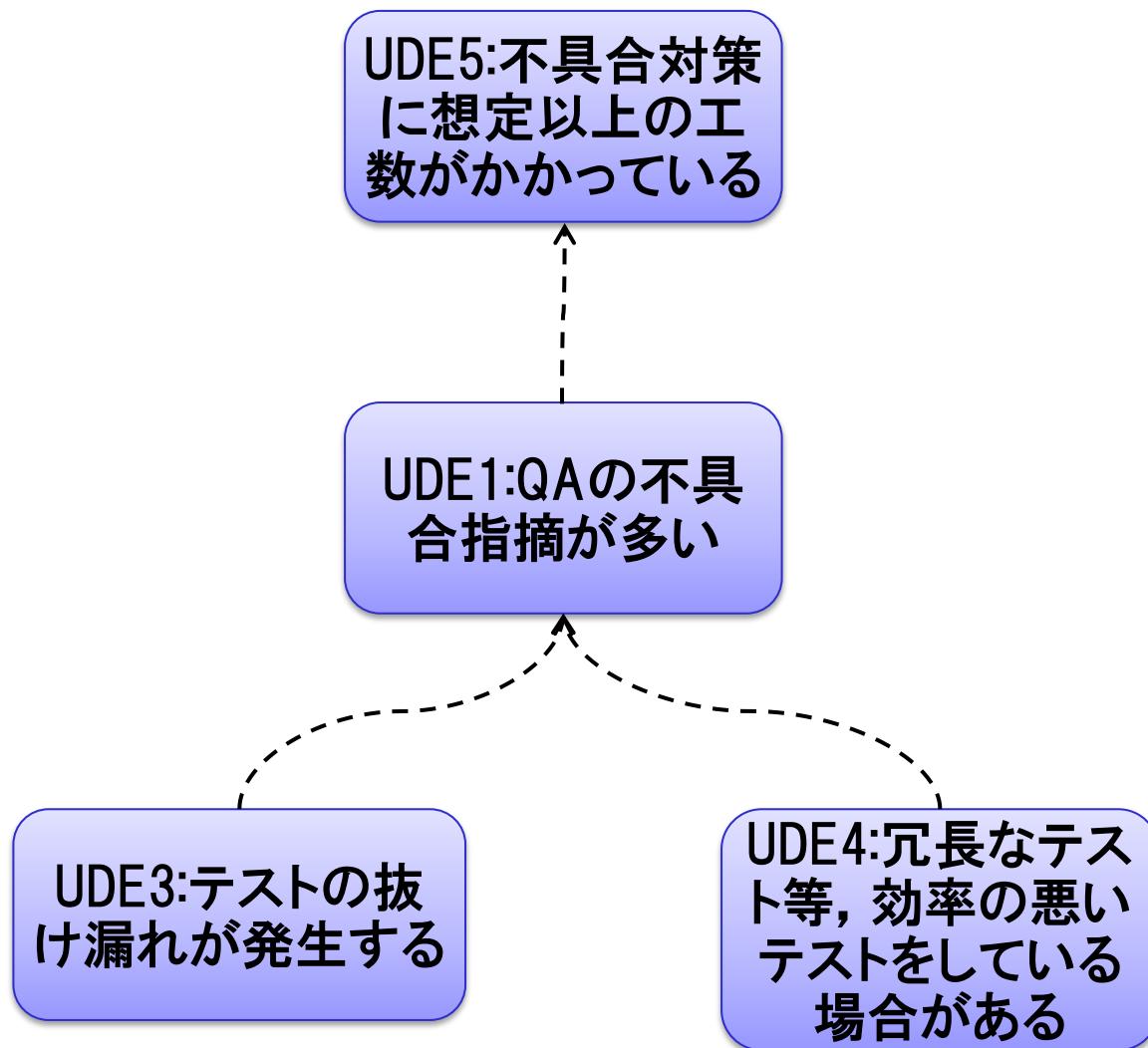


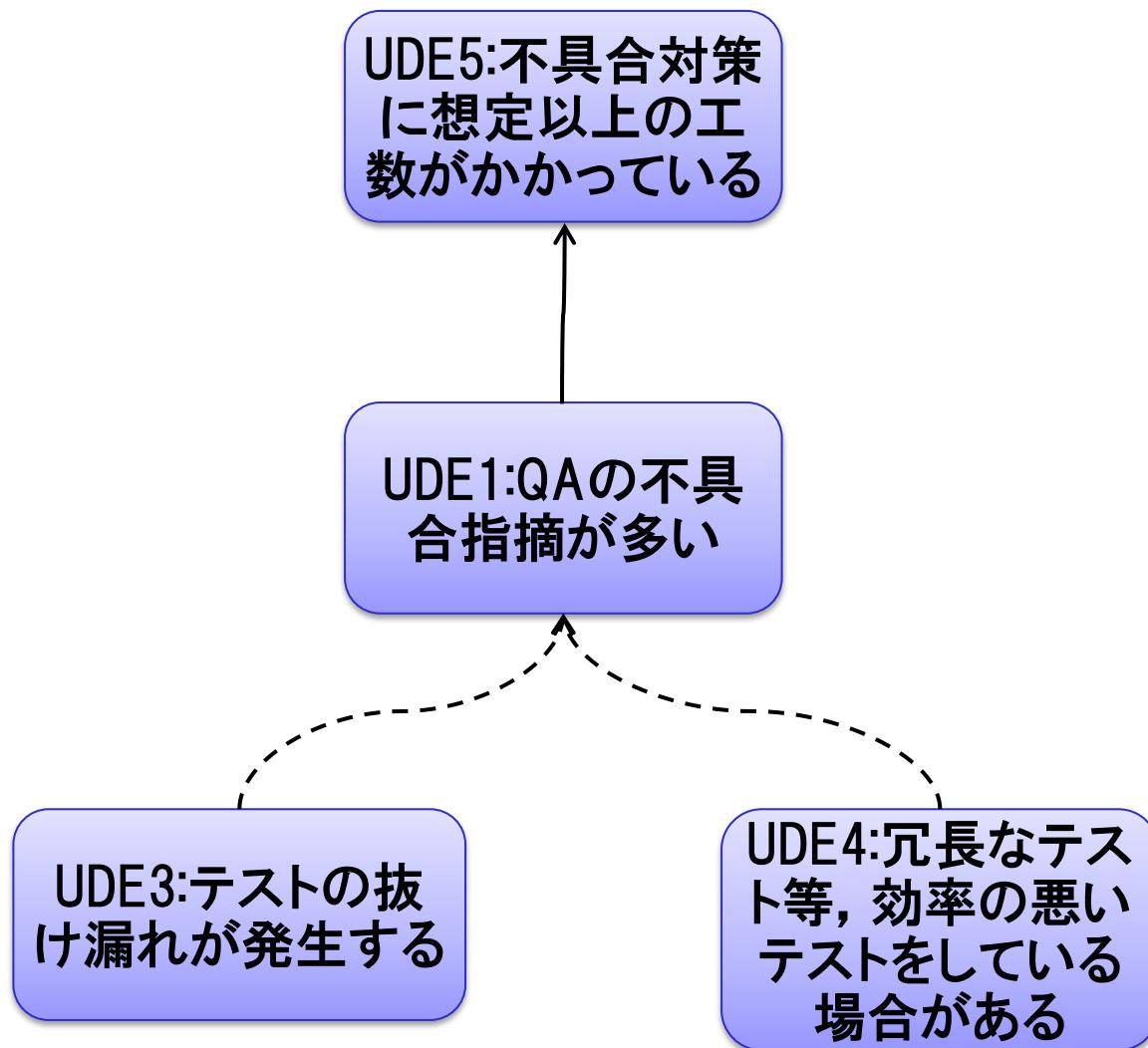
■ 対象組織のUDEs

- UDE1: QAの不具合指摘が多い
- UDE2: テストの品質(不具合抽出効果)がバラつく
- UDE3: テストに抜け漏れが発生している
- UDE4: 冗長なテスト等, 効率の悪いテストをしている場合がある
- UDE5: 不具合対策に想定以上の工数がかかっている
- UDE6: 不具合抽出に効果的なテストがわからない

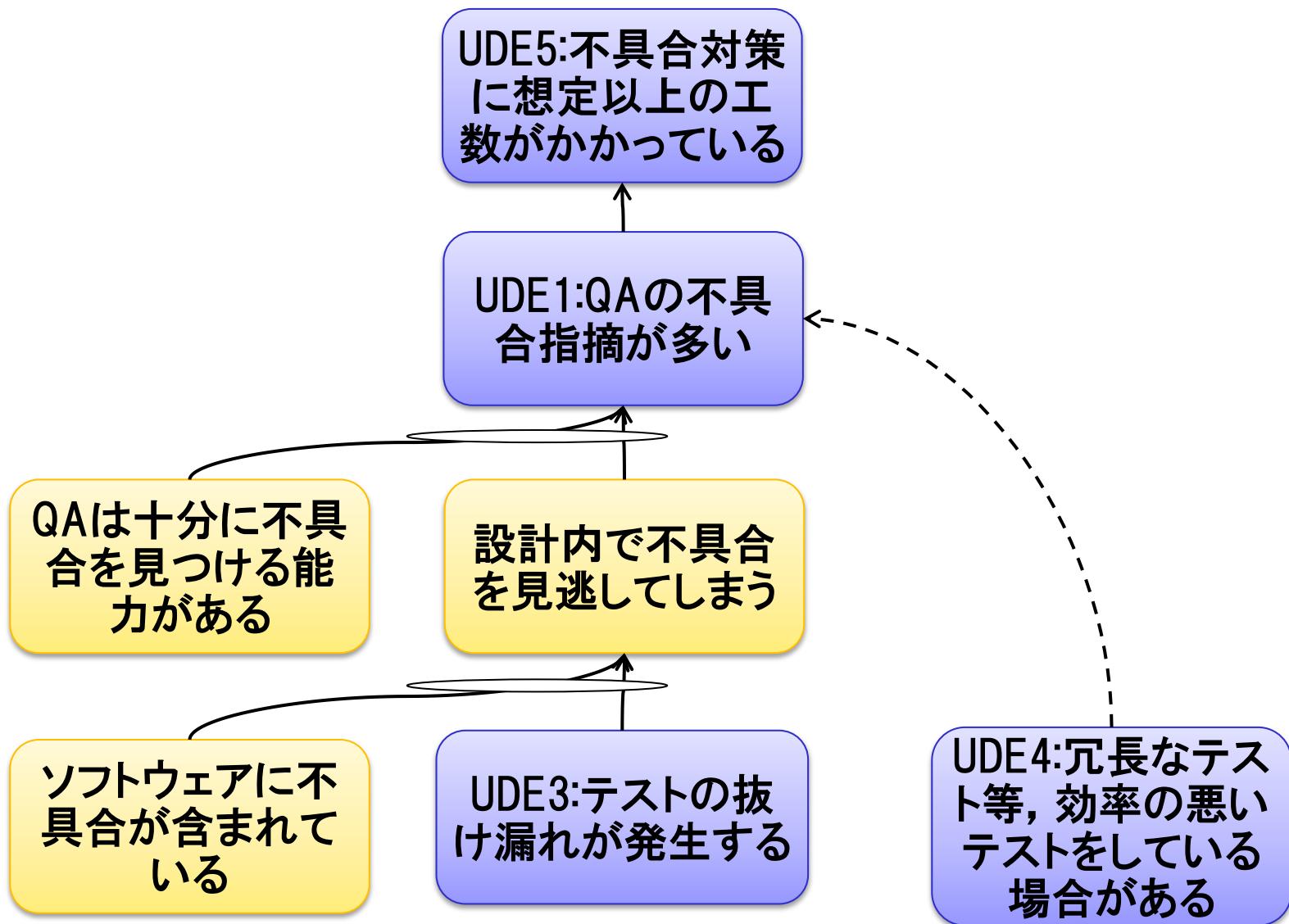
■ 対象組織のUDEs

- UDE1:QAの不具合指摘が多い
- UDE2:テストの品質(不具合抽出効果)がバラつく
- UDE3:テストに抜け漏れが発生している
- UDE4:冗長なテスト等, 効率の悪いテストをしている場合がある
- UDE5:不具合対策に想定以上の工数がかかっている
- UDE6:不具合抽出に効果的なテストがわからない

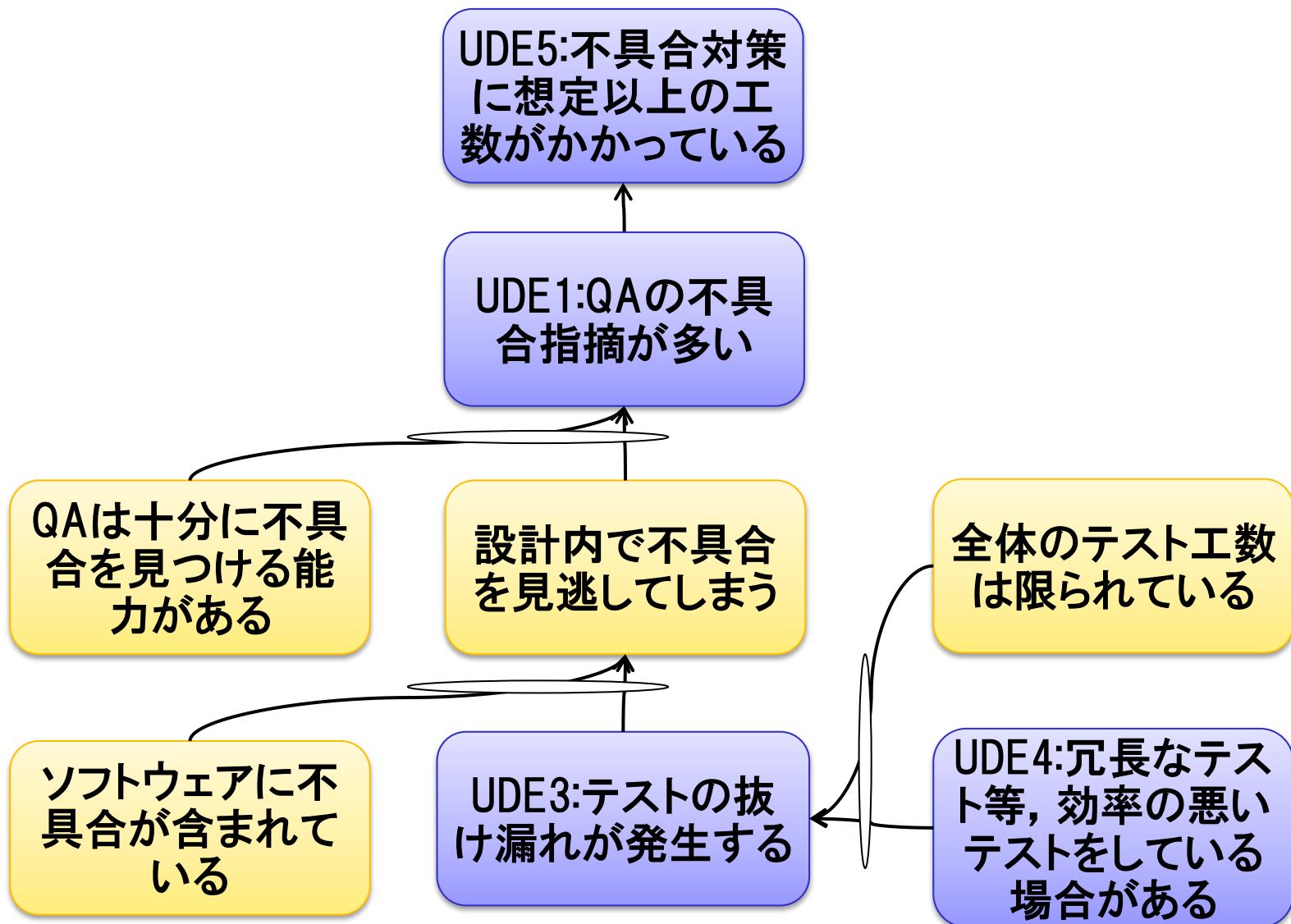




3-3 現状構造ツリーの作成例 3/5

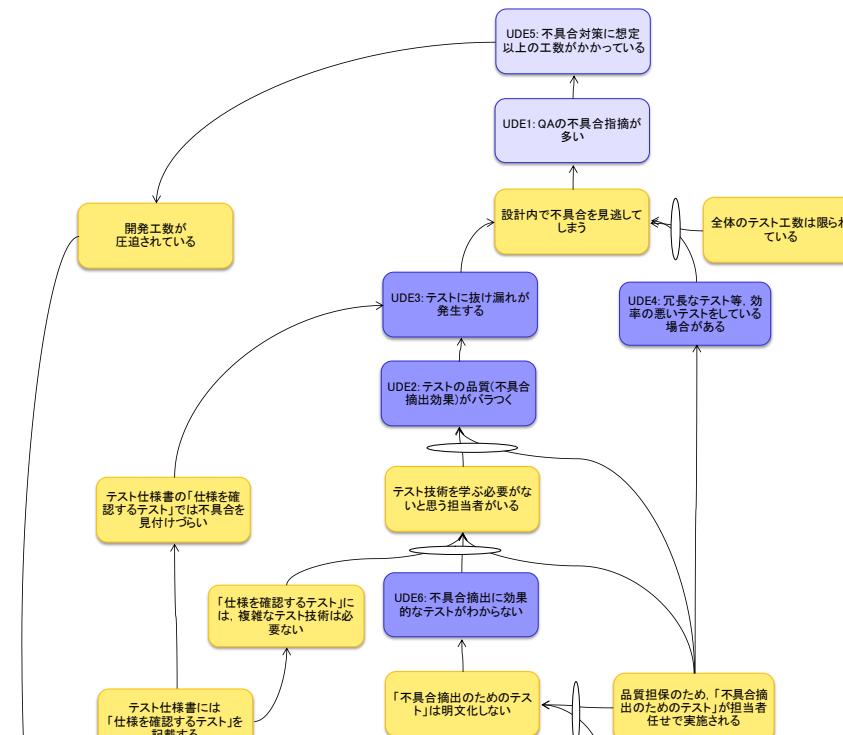


3-3 現状構造ツリーの作成例 3/5

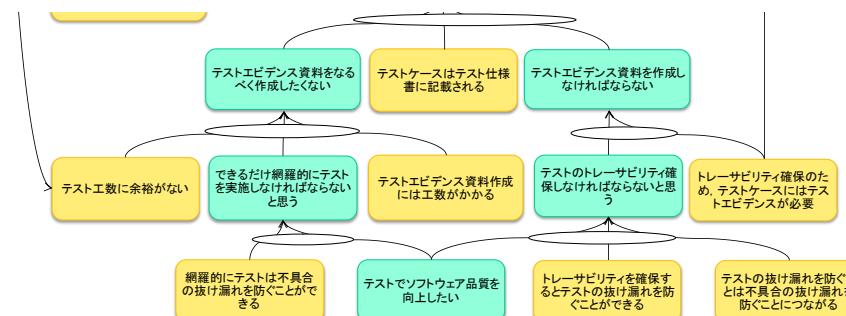


3-3 現状構造ツリーの作成例 5/5

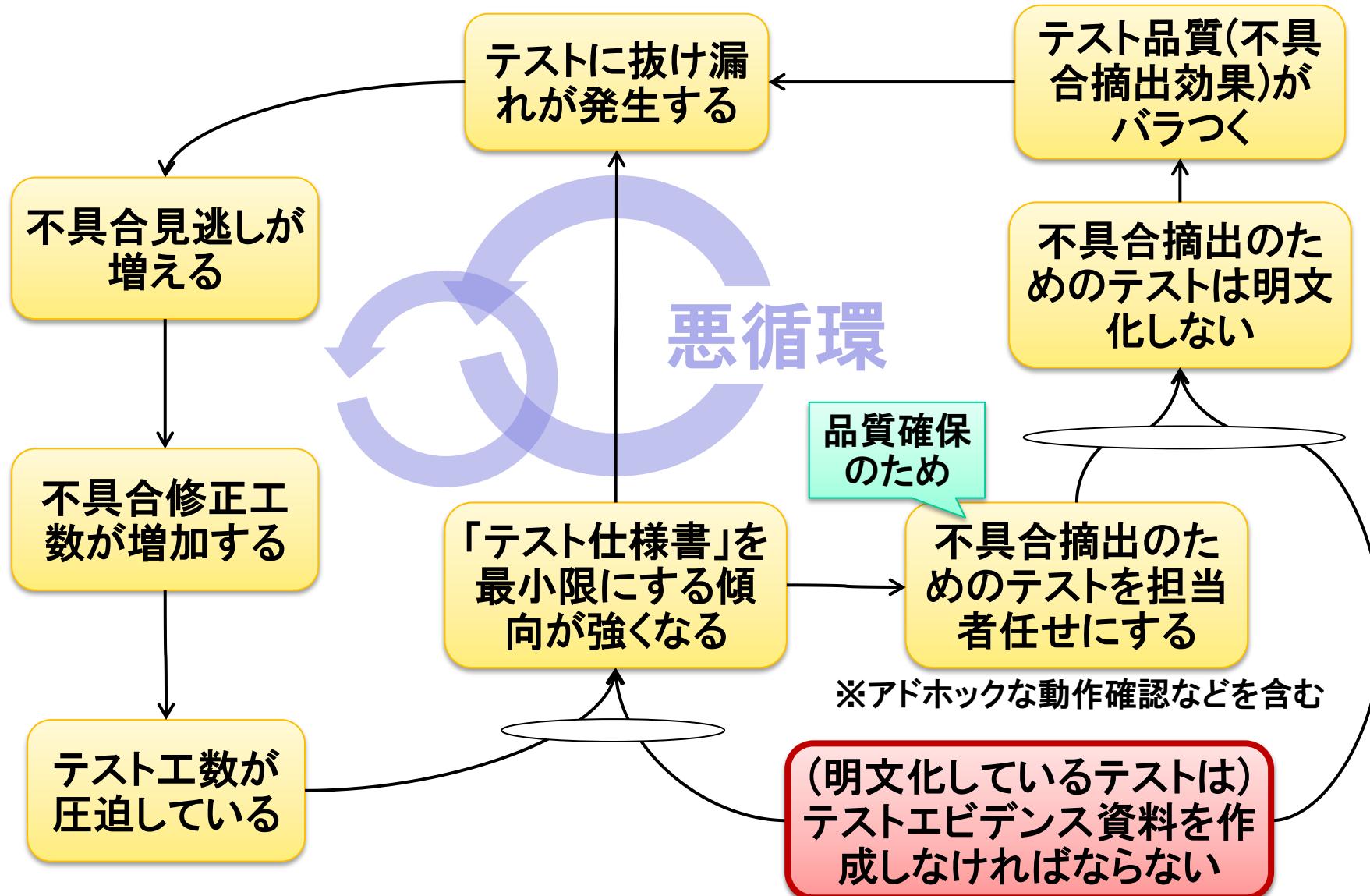
最終的に. . .



これだとわからんので. . .



3-4 現状構造ツリーによる分析結果(概要)



■ 対象組織では. . .

- 「トレーサビリティ」を内部規定としてる
- 「テスト」も同様

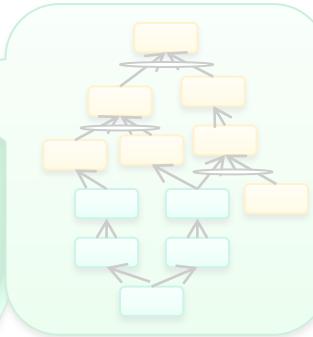
■ 「テストにおけるトレーサビリティ」とは？

- 「要求・仕様～テストケース～テスト結果」の追跡性
 - ✓ 要求項目・仕様項目に対応してテストケースを作成する
 - ✓ テスト結果「テストエビデンス」を資料に纏める
 - ※ この資料を「テストエビデンス資料」と呼ぶ
- 目的は
 - ✓ 要求項目・仕様項目が適切にテストされているかを示す
 - ✓ 不具合が発生した場合、迅速に要因を特定する

↓
テスト工数が
圧迫している

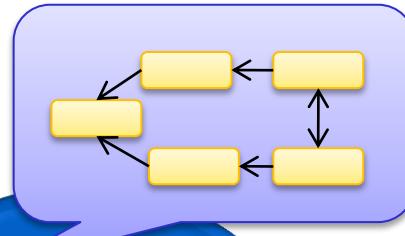
(明文化しているテストは)
テストエビデンス資料を作成しなければならない

現状構造ツリー



(1)何を変えるか?
What to change?

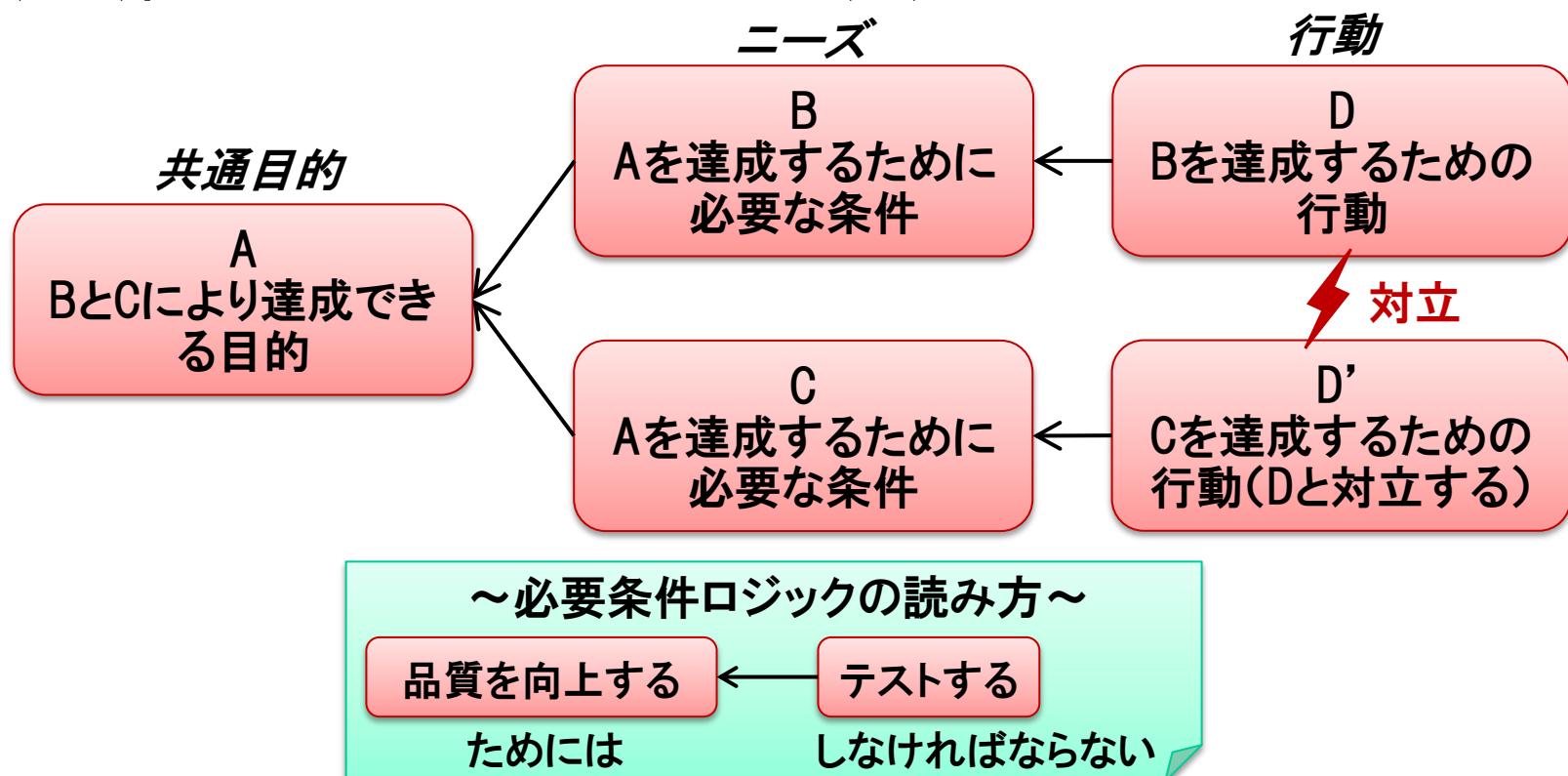
対立解消図



(3)どのように
変えるか?
How to cause
the change?

(2)何に変えるか?
To what to
change?

- 「問題」とは「対立」であり、対立が根本原因を招く
 - この対立構造を示したものが「対立解消図」
 - 対立する行動(DD'), 各行動により満されるニーズ(BC), 両ニーズの共通目的(A)を表す。



3-6 中核問題の対立解消図

なぜならば(A-B)
網羅的なテストは不具合の抜け漏れを防ぐ

なぜならば(B-D)
テスト工数に余裕がなく、テストエビデンスに工数をかけると全てのテストが終了しない

A
テストでソフトウェア品質を向上する

B
できるだけ網羅的にテストを実施する

D
テストエビデンス資料を作成しない

C
テストのトレーサビリティを確保する

D'
テストエビデンス資料を作成する

なぜならば(A-C)
・トレーサビリティを確保しているテストは要求項目の確認を漏らさない
・不具合発生時に迅速に対応できる

なぜならば(C-D')
・テストエビデンスはトレーサビリティ確保の一つ
・全てのテストケースにテストエビデンスが必要

3-6 中核問題の対立解消図

なぜならば(A-B)
網羅的なテストは不具合の抜け漏れを防ぐ

なぜならば(B-D)
テスト工数に余裕がなく、テストエビデンスに工数をかけると全てのテストが終了しない

A
テストでソフトウェア品質を向上する

B
できるだけ網羅的にテストを実施する

C
テストのトレーサビリティを確保する

D
テストエビデンスを確保する

この2つのニーズを満足する方法を検討する

なぜならば(A-C)
・トレーサビリティを確保しているテストは要求項目の確認を漏らさない
・不具合発生時に迅速に対応できる

なぜならば(C-D')
・テストエビデンスはトレーサビリティ確保の一つ
・全てのテストケースにテストエビデンスが必要

3-6 中核問題の対立解消図

なぜならば(A-B)
網羅的なテストは不具合の抜け漏れを防ぐ

なぜならば(B-D)
テスト工数に余裕がなく、テストエビデンスに工数をかけると全てのテストが終了しない

A
テストでソフトウェア品質を向上する

そのためにこれら
「仮定」を検証する

B
抜け網羅的

D
テストエビデンス
資料を作成しない

対立

D'
テストエビデンス
資料を作成する

なぜならば(A-C)
・トレーサビリティを確保しているテストは要求項目の確認を漏らさない
・不具合発生時に迅速に対応できる

なぜならば(C-D')
・テストエビデンスはトレーサビリティ確保の一つ
・全てのテストケースにテストエビデンスが必要

なぜならば(A-B)
網羅的なテストは不具合の抜け漏れを防ぐ

なぜならば(B-D)
テスト工数に余裕がなく、テストエビデンスに工数をかけると全てのテストが終了しない

B
できるだけ網羅的

D
テストエビデンス

A
テストでソフトウェア品質を向上する

この「仮定」に開発者の思い込みがある!!
→ 全てのテストにテストエビデンスが必要なわけではない!!

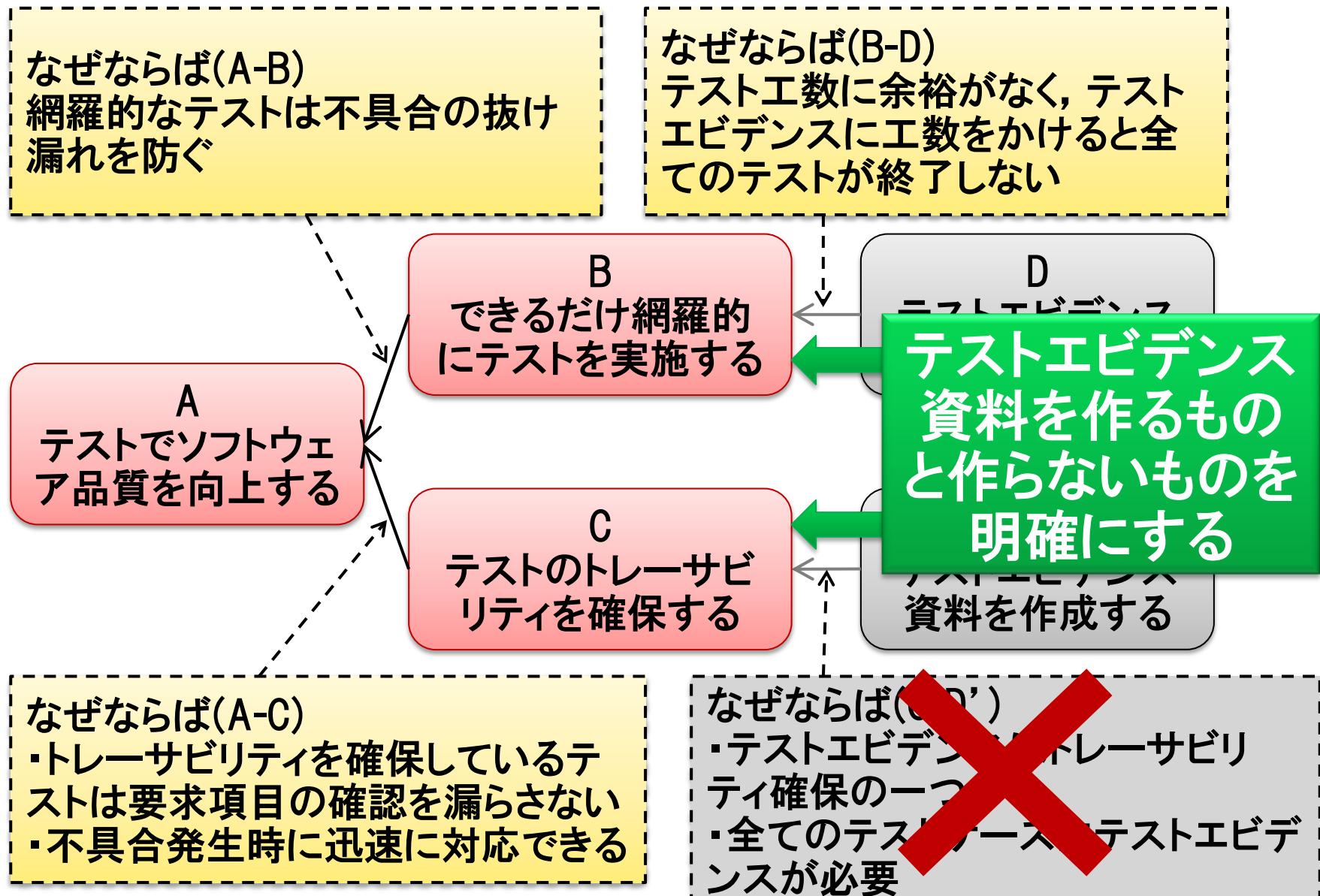
テストのトレーサビリティを確保する

テストエビデンスを作成する

なぜならば(A-C)
・トレーサビリティを確保しているテストは要求項目の確認を漏らさない
・不具合発生時に迅速に対応できる

なぜならば(C-D')
・テストエビデンスはトレーサビリティ確保の一つ
・全てのテストケースにテストエビデンスが必要

3-8 中核問題の対立解消図による改善策



TOC思考プロセスを用いたテスト工程の課題分析と改善 - テストにおけるトレーサビリティ確保の落とし穴 -

Contents

- 1.背景と目的
- 2.TOC思考プロセスとは
- 3.TOC思考プロセスによる分析
- ▶ **4.改善策の策定**
- 5.改善策の試行・評価
- 6.考察
- 7.まとめ

■ 改善策概要

① 「テスト」を定義:

テストエビデンス資料を作成しない不具合抽出のための「テスト」を明確に定義

② 「テスト」のレビュー:

不具合抽出のための「テスト」を明文化, レビューすることで, 担当者によるテスト品質のバラツキを防止

■ 改善策概要

- ▶ ① 「テスト」を定義:
テストエビデンス資料を作成しない不具合抽出のための「テスト」を明確に定義

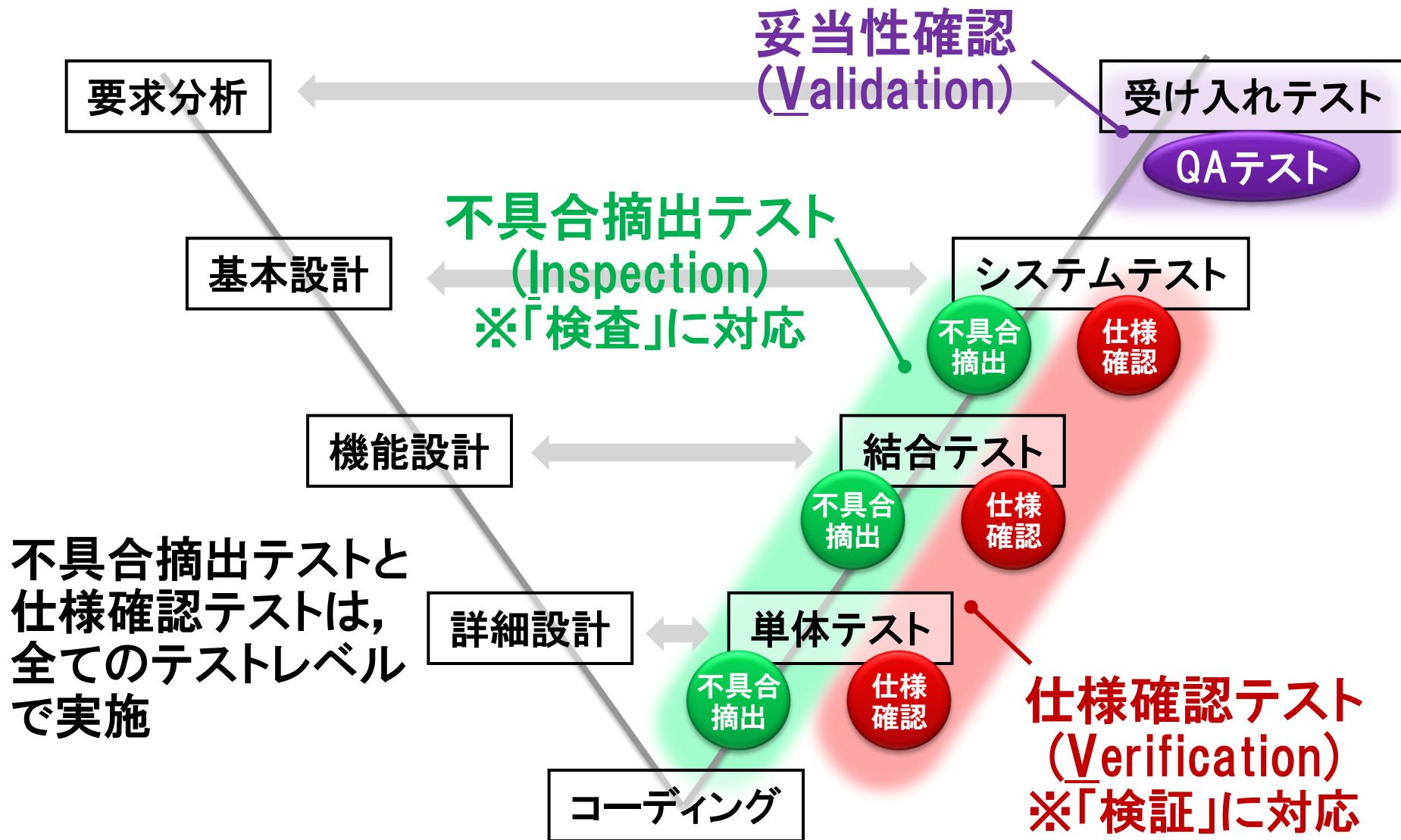
- ② 「テスト」のレビュー:
不具合抽出のための「テスト」を明文化, レビューすることで, 担当者によるテスト品質のバラツキを防止

■ 不具合抽出テストと仕様確認テスト

| | 不具合抽出テスト | 仕様確認テスト |
|---------------|--|---|
| テストの目的 | 不具合の抽出 | 仕様を満足していることの確認 |
| ISO9000との対応 | 「検査(Inspection)」 必要に応じて測定, 試験又はゲージ合せを伴う, 観察及び判定による適合性評価 | 「検証(Verification)」 客観的証拠(テストエビデンス)を提示することによって, 規定要求事項が満たされていることを確認すること |
| テストケースのフォーマット | <u>フリーフォーマット</u> | 内部規定のテスト仕様書 |
| テストエビデンスの必要性 | <u>なし</u> | あり |
| テストケースの相互関係 | 不具合抽出 テスト | 仕様確認テスト 性能テスト, メモリリーク対策確認, など |

■ 不具合抽出テストと仕様確認テスト

| | 不具合抽出テスト | 仕様確認テスト |
|--------------|--|----------------|
| テストの目的 | 不具合の抽出 | 仕様を満足していることの確認 |
| ISO9000との対応 | 「検査(Inspection)」 | |
| テストケースフォーマット | 要は「テストの目的」を分離している | |
| テストエビデンスの必要性 | <ul style="list-style-type: none">不具合抽出仕様確認 ▶ 欠陥を抽出する。 ▶ 対象ソフトウェアの品質レベルが十分であることを確認する。 ▶ 意志決定のための情報を示す。 ▶ 欠陥の作りこみを防ぐ。 | |
| テストケース相互関係 | (JSTQB FL シラバス 日本語版 Ver. 2011.J01 1.2「テストとは何か?」より) | |



■ 改善策概要

- ① 「テスト」を定義:
テストエビデンス資料を作成しない不具合抽出のための「テスト」を明確に定義
- ▶ ② 「テスト」のレビュー:
不具合抽出のための「テスト」を明文化, レビューすることで, 担当者によるテスト品質のバラツキを防止

■ テストレビューの目的

- 不具合抽出テストの担当者によるバラツキを防止する

不具合抽出に「効果的」かつ「効率的」な
テストになるようにレビューしたい!!

■ 「効果的」工夫

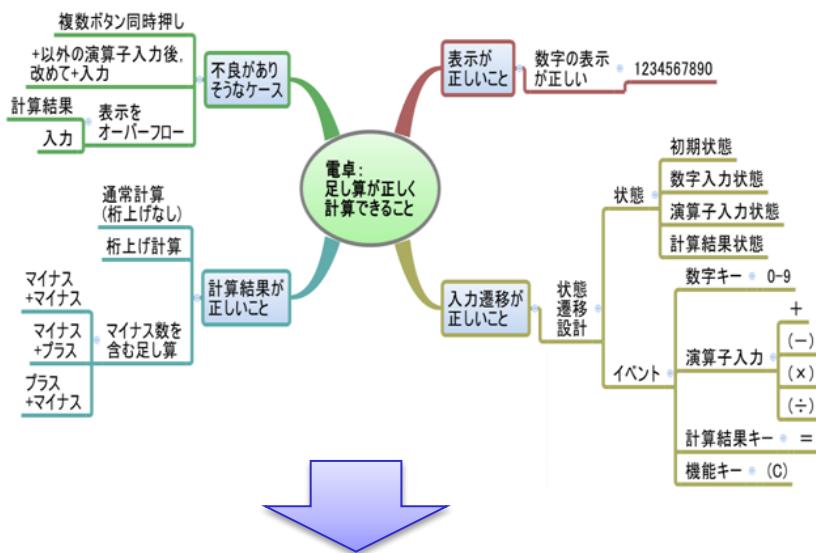
- マインドマップの使用
- 「テスト実施範囲の決定」の分離

■ 「効率的」工夫

- マインドマップツールの使用
- 不具合抽出テストのテストケース作成省略

■ マインドマップの使用

- 不具合抽出テストのテストレビュー対象物として、「**テスト設計のマインドマップ**」を用いる



『テスト観点レビュー』
と命名

【理由1】

「方針制約」に縛られていた開発者に「不具合抽出」という目的で、発散的に発想を広げてもらうため

【理由2】

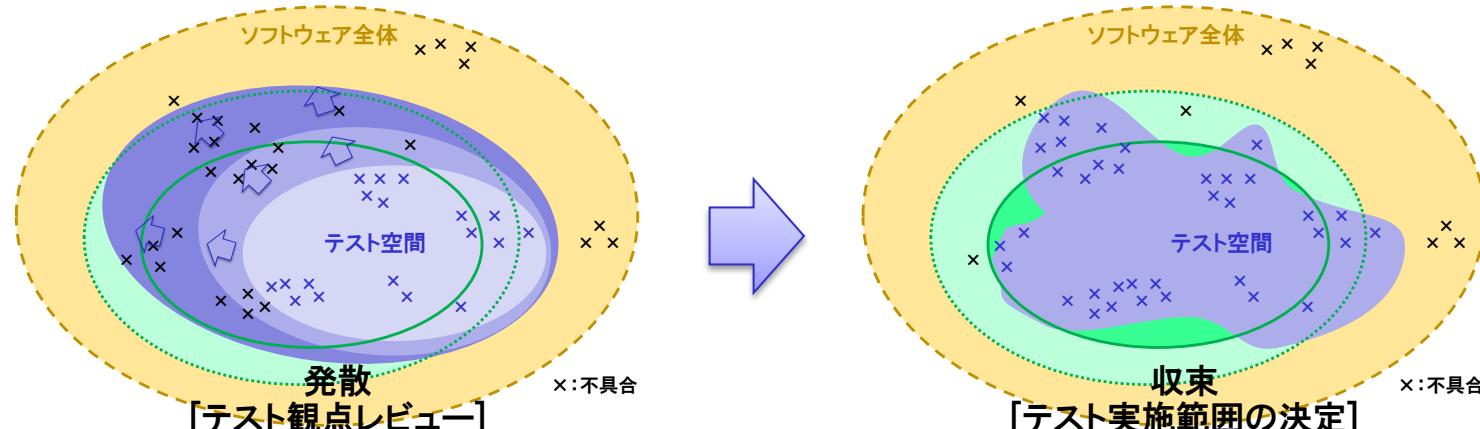
テストケースのレビューは「**テスト設計の意図(テスト観点)**」が抜けっていて、テストの抜け漏れを発見しにくい

■「テスト実施範囲の決定」の分離

- 「テスト観点レビュー」と「テスト実施範囲の決定」を分離する

【理由】

テスト観点レビューで、多くのテスト観点を叩き出せるが、多すぎてテスト実施困難である。一方、テスト観点レビュー中にテスト実施を意識すると発想が偏り、テスト観点の抜け漏れが発生すると考えられるため

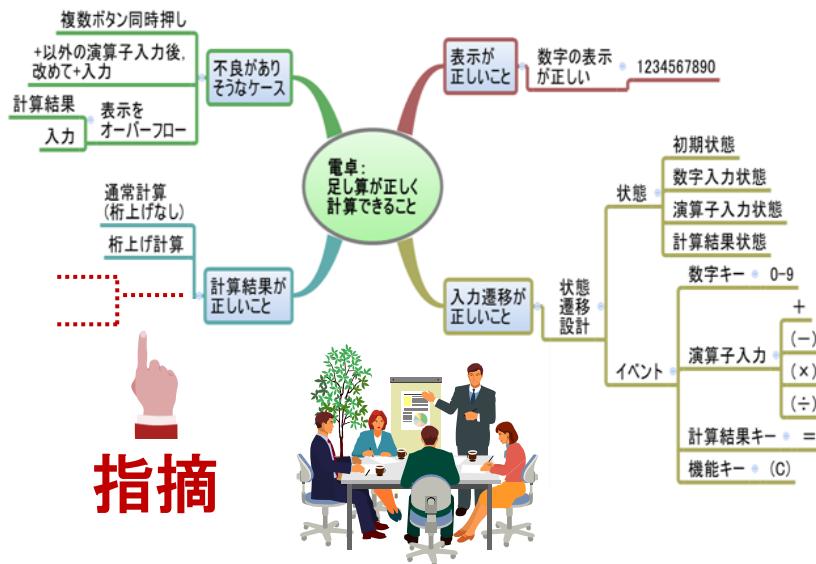


■ マインドマップツールの使用

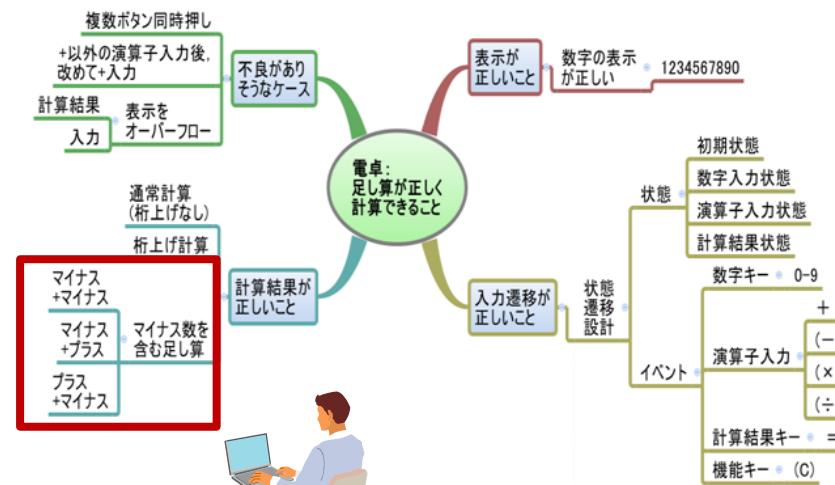
➤ テスト観点レビューを効率的に実施するために、マインドマップツール(XMind)を用いる

【理由】

レビュー中に指摘を反映していきたい
手書きのマインドマップでは、柔軟な変更が困難



反映



■ テストケース作成の省略

- マインドマップそのものを用いて、担当者がテスト実施できる場合に限り、テストケース作成を省略して、よいものとする

【理由】

- ・そもそも不具合抽出テストの目的は、不具合抽出であり、テストケースを作成せずに、テスト実施できるのであればそれでよいため
- ・テストケース作成が方針制約になることを防ぐため

※ただし、
マインドマップを用いて消化率等の
進捗データを収集し、管理する

TOC思考プロセスを用いたテスト工程の課題分析と改善 - テストにおけるトレーサビリティ確保の落とし穴 -

Contents

- 1.背景と目的
- 2.TOC思考プロセスとは
- 3.TOC思考プロセスによる分析
- 4.改善策の策定
- ▶ **5.改善策の試行・評価**

- 6.考察
- 7.まとめ

■ パイロットプロジェクトの内容

- 複数のパイロットプロジェクトで、提案手法を試行し、評価を実施した。

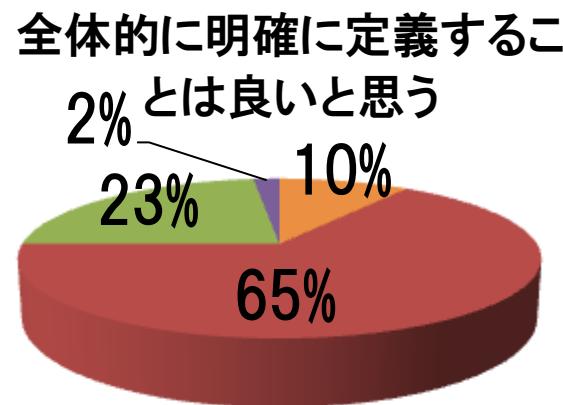
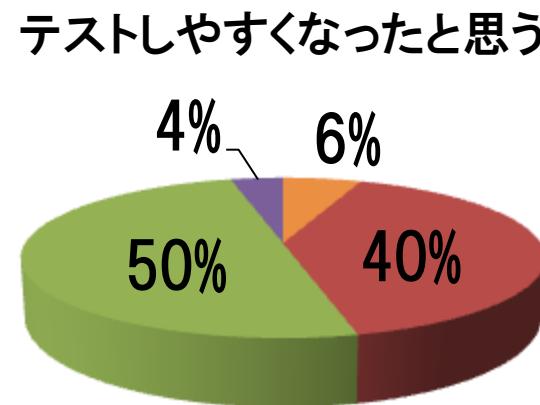
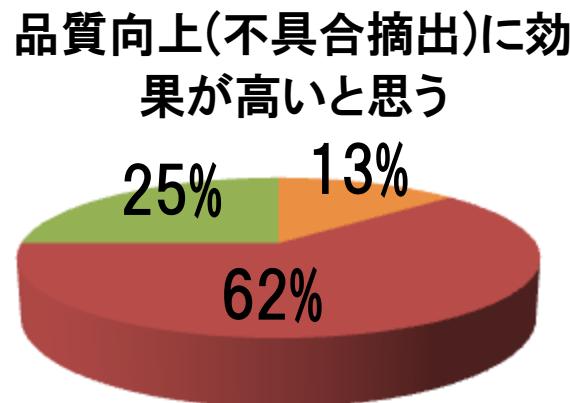
■ 評価方法

- 定性的評価：パイロットプロジェクトメンバーへのヒアリングとアンケート
- 定量的評価：パイロットプロジェクトでの不具合摘出テストのテスト観点レビューによる追加テストケース数と不具合数の分析

■ヒアリングで得た主な意見

| 効果の実感 | 検討事項 |
|---|---|
| <p>■不具合抽出テストについて</p> <ul style="list-style-type: none"> ➤ テストの定義で不具合を抽出しやすくなった ➤ テスト観点を挙げやすくなった ➤ テストで不具合を抽出する意識が高まった <p>■マインドマップについて</p> <ul style="list-style-type: none"> ➤ テスト観点の整理・洗い出しに良い ➤ テスト観点を残せるのが良い <p>■テスト観点レビューについて</p> <ul style="list-style-type: none"> ➤ テスト仕様書よりレビューしやすい ➤ 重大な抜け漏れを発見できた ➤ 例外処理などを多く叩き出すことができる ➤ 設計の詰めが甘いところを見付けられた ➤ 若手への教育的効果もありそう ➤ 協力会社のテストレビューに使うと良さそう | <p>■マインドマップについて</p> <ul style="list-style-type: none"> ➤ 性能系など適していないテストもある ➤ 大きな機能の場合、表示等に工夫が必要 ➤ 単語のみで作成すると意図がわからなくなることがある <p>■テスト観点レビューについて</p> <ul style="list-style-type: none"> ➤ レビューアにはドメイン知識が必要 ➤ レビューの人数・時間の目安がわからない ➤ レビュー後のマインドマップの整理が大変 <p>■その他</p> <ul style="list-style-type: none"> ➤ マインドマップからテストケースが生成できることよい |

■ 不具合抽出テストと仕様確認テストの定義について



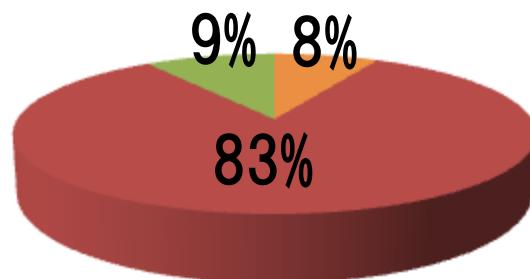
■ 色の説明
■ 強く同意する ■ 同意する ■ どちらとも言えない ■ 同意しない ■ 強く同意しない

※アンケート回答者:52名

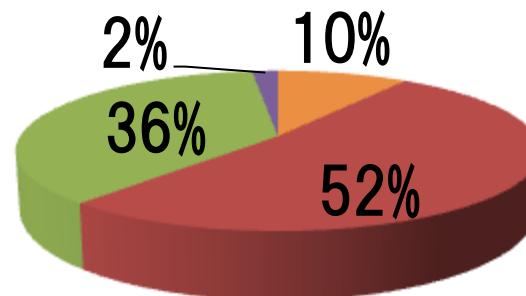
5-4 定性的データによる評価:アンケート結果

■ テスト観点レビューについて

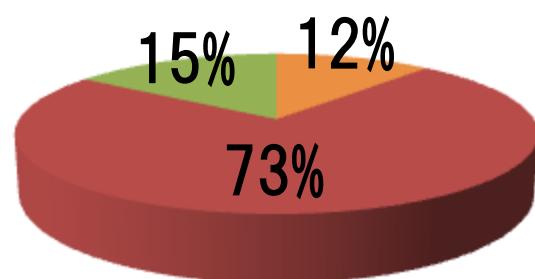
品質向上(不具合抽出)に効果が高いと思う



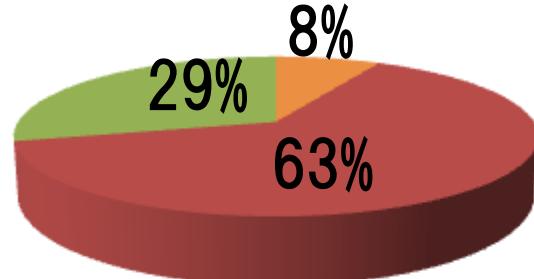
レビューしやすいと思う



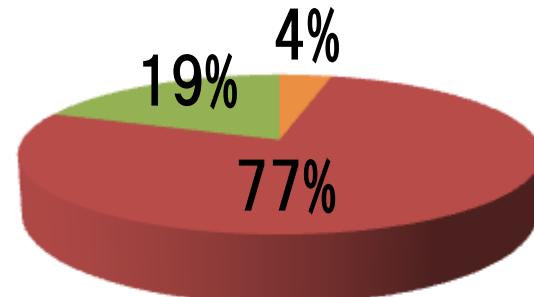
テストの抜け漏れを発見しやすいと思う



今後もやっていきたいと思っている



全体的に良い方法だと思う



■ 評価指標

- 強く同意する
- 同意する
- どちらとも言えない
- 同意しない
- 強く同意しない

※アンケート回答者:52名

■ テスト観点レビューで追加となったテストケース数とそのテストによる抽出不具合数

| PJ | テストケース数[件] | 抽出不具合数[件] | 開発規模 [KLOC] | QA指摘[件] |
|----|------------|-----------|-------------|---------|
| A | 22(144) | 1(1) | 3.5 | 1 |
| B | 36(160) | 2(3) | 1.0 | 1 |
| C | 31(48) | 2(4) | 1.5 | 0 |
| D | 29(300) | 0(5) | 4.1 | 0 |
| 合計 | 118(652) | 5(13) | 10.1 | 2 |

※カッコ内は、全体での数

■ 評価結果まとめ

- QA指摘密度(QA指摘数/対象規模)
 - ✓ 同時期の他プロジェクトの1/3程度
→ 妥当性確認時の品質向上を確認
- 不具合見逃し率(QA指摘数/全不具合数)
 - ✓ テスト観点レビュー実施: 13.3%
 - ✓ テスト観点レビュー未実施(見積り): 46.7%
→ 不具合見逃し低減効果を確認
- 不具合抽出効率(抽出不具合数/テストケース数)
 - ✓ レビュー追加分のテスト観点: 0.042件/テストケース
 - ✓ レビュー指摘以外のテスト観点: 0.015件/テストケース
→ テスト観点レビューが不具合抽出に効果が高いことを確認

- 不具合摘出テストの定義
- テスト観点レビュー

は「定性的」にも「定量的」にもテストエビデンスに関する中核問題(方針制約)を解消し、設計内での摘出不具合の増加に効果があるといえる。

TOC思考プロセスを用いたテスト工程の課題分析と改善 - テストにおけるトレーサビリティ確保の落とし穴 -

Contents

- 1.背景と目的
- 2.TOC思考プロセスとは
- 3.TOC思考プロセスによる分析
- 4.改善策の策定
- 5.改善策の試行・評価
- ▶ **6.考察**
- 7.まとめ

(テストプロセス上の)

「改善のための事象」

が、

「様々な理由」

により、

悪影響を及ぼす

ことがある

- 昔の基準・ルール
- 改善活動
ex) テストツール導入, テスト手法導入, など

- 目的・思想が失われる/歪曲される
- 環境・状況が変わる

「方針制約」となる!!!

TOC思考プロセスはそれを見つけ出し、
解消するのに効果がある

TOC思考プロセスを用いたテスト工程の課題分析と改善 - テストにおけるトレーサビリティ確保の落とし穴 -

Contents

- 1.背景と目的
- 2.TOC思考プロセスとは
- 3.TOC思考プロセスによる分析
- 4.改善策の策定
- 5.改善策の試行・評価
- 6.考察
- ▶ **7.まとめ**

- ある組込み製品事業部において, TOC思考プロセスを用いて, テストのトレーサビリティの一つであるテストエビデンスが, 不具合摘出を阻害していることを明確にした.
- この問題の背景に「全てのテストにテストエビデンスが必要」という誤った思い込みがあることを明らかにした.
→ TOC思考プロセスは, このような「方針制約」をみつけるのに有効である.
- ①不具合摘出テストの定義, ②テスト観点レビュー, を行うことにより, この問題を解消できることを示した.

ご清聴ありがとうございました。

END

TOC思考プロセスを用いたテスト工程の課題分析と改善
- テストにおけるトレーサビリティ確保の落とし穴 -

2012/1/26

株式会社 日立製作所 横浜研究所
システム開発研究部

○八木 将計, 小川 秀人

HITACHI
Inspire the Next[®]