



テストの質を上げるための要求仕様書

JaSST'07 講演資料

(株)システムクリエイツ

代表取締役 清水 吉男

URL=http://homepage3.nifty.com/koha_hp

shimz@mb.infoweb.ne.jp



agenda

1. 何がテスト作業を混乱させているのか
 - そこで書かれている要求仕様の出来映えに端を発している
 - 派生開発において変更の内容がテストに反映されていない
2. 要求仕様の表記法によってテスト作業が変わる
 - USDMによる要求仕様の表記
 - XDDPによる派生開発プロセスにおける「3点セット」

1.1 頻発する品質の問題

- **ソフトウェアのバグ**が元で**社会基盤をも揺るがす事態**が発生
 - 飛行機がとまる、電車がとまる、自動車をとまる
 - IP電話がとまる、ネットもとまる
 - 請求書の内容もいちいちチェックしないといけないのか???

毎日どこかでソフトウェアの品質に起因するトラブルが発生

- 企業……………損失の発生、利益の減少
- 消費者……………損失コストを負担させられている
- 開発現場………**過剰労働**、多重責務が発生
 - 有能な技術者のソフト産業からの離脱
 - **家庭の崩壊**、社会の崩壊へ

1.2 テストの3つの品質

- テストには3つの品質が求められる

テストケースの品質

- テストケースの出来映え、網羅性は適切か？
- 経済的合理性も無視できない

テスト作業の品質

- 手抜きなくテスト作業が行なわれたか？
- 無理な工数で作業の手抜きを引き起こしていないか？

テスト結果の判断の品質

- テスト結果の判断を間違えない仕組みになっているか？
- テストケースの有効性は？

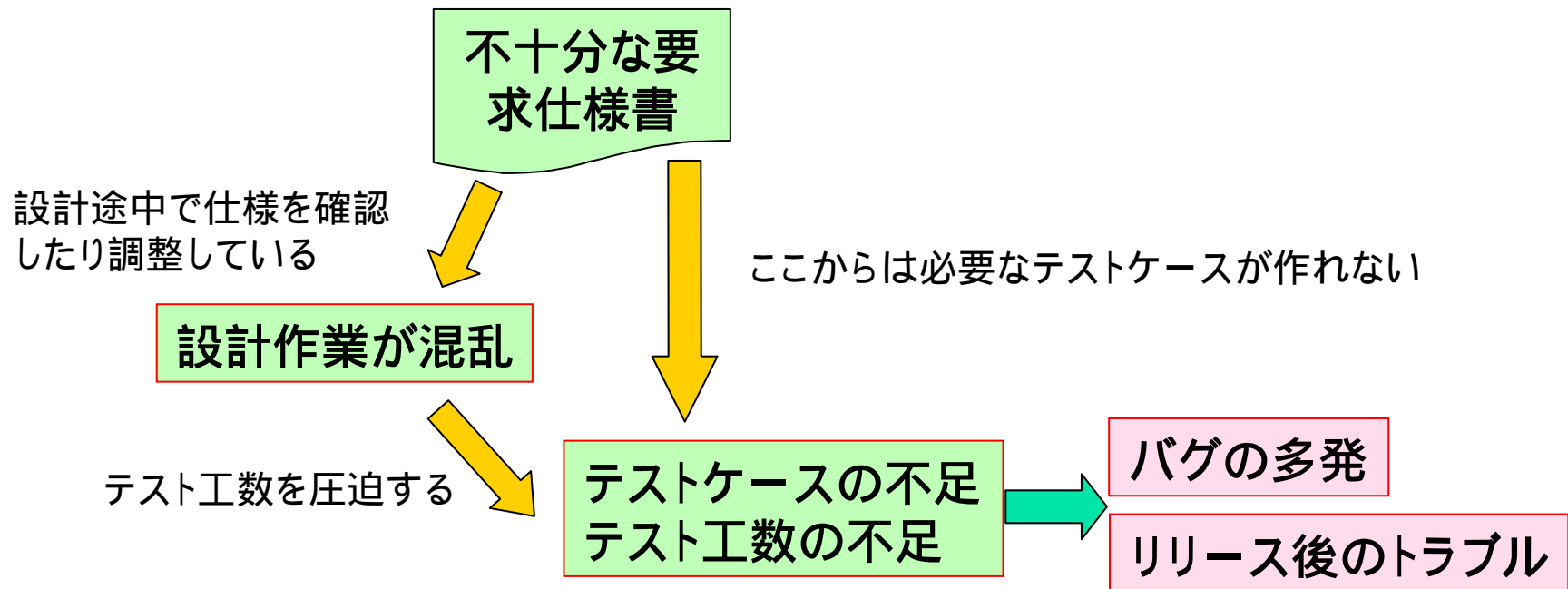
- 「テスト」は品質保証の**最終関門**
 - “ブランド”は品質保証の上に成立する

1.3 テストが不十分な状態でリリースしている

- 市場でトラブルとなるのは、事前の**テスト**で問題を発見できないままリリースしたことによる
- テストが不十分な状態に陥った**原因**は幾つもある
 - テストケースの**網羅性**に問題がある
 - 限界まで負荷をかけたテストや**イリーガルな場面**のテストが不足
 - 適切な**テストケース**を作るための**情報**が不足している
 - テストケースを設計できる**人材**がない
 - 最初からテスト**工数**が不足している
 - **バグが多すぎて**予定していたテスト工数でのテストができない
 - テスト**工数**が**上流の作業**に奪われている
 - テスト作業が**人手に依存しすぎて**いて、作業の品質を保てない
 - テスト作業者の**モチベーション**が低い

1.4 テストにしわ寄せされる状況

- **要求仕様書の出来映え**が、最終的にテスト作業にしわ寄せされる
 - テストケースの内容が不十分
 - テスト工数が消えていく



1.5 要求仕様書がテストケースの源泉になっていない

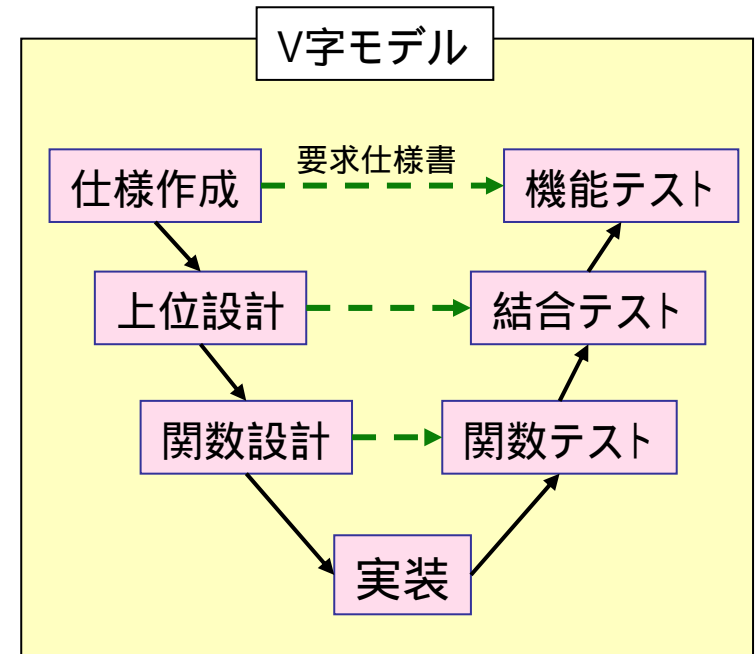
- 本来、テスト部門におけるテストは**要求仕様書に書かれていることが実現していることを確認すること**

現実とは…

- **振る舞いが不明確**
- 部分的な機能の仕様に終始
- 前提条件や細かな処理、終了時の状態などが不明確
- “**正常系**”の仕様が**中心**で、エラー条件やその時の対応が不十分

その結果

- そこで分かることから動きを**想像**してテストするしかない
- テストケースの準備が不十分な状態に陥る



1.6 これって仕様ですか？

- テストの中で、要求仕様に書かれていない動きをして、**バグなのか判断できない状態**にしばしば遭遇している
- 設計部門の人に確認すると
 - 「それは仕様です」
 - 「その仕様には制限を付けました」



これって、仕様ですか？
バグですか？

「仕様は設計しながらでない」と抽出できない」という発想の下では、この種の事態は回避できない

1.7 テストを無視した設計が行なわれている

- **テストのしやすさを無視した設計**が行なわれているために、テストケースが対応しきれしていない
 - 複雑度を無視し、3000行を越えるような関数の中のアルゴリズムをモレなくテストすることに合理性はあるか？
 - グローバル変数を多用する状況では、耐性を確認するためのテストケースを準備しきれない

勝手気ままに設計・実装されたプログラムに潜むバグを、合理的な時間内に効果的に発見する方法などない

1.8 テストエンジニアが育っていない

- 長い間、テストは「**人海戦術**」で行なうものと考えられてきた
 - 製造工場での「テスト」と同一視されて来た？
 - テスト部門の人に**設計の経験が不足**

↓

“**テストエンジニアのキャリアパス**”がない

↓

- **設計部門のエンジニアと涉り合えるエンジニアがいない**
 - テスト技術の停滞
 - 膨張するテストケースをコントロールできない
 - テストの効率を上げるための設計方法を提案できない

参考:「AT&T のソフトウェアテスターのキャリアパス」IEEE Software Mar/Apr 2000

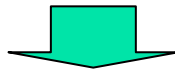
1.9 派生開発におけるテストの問題

組み込みシステムでは、ほとんどが「派生開発」である

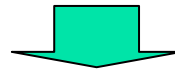
- 派生開発には2つの問題が起きている
 - A) 仕様変更に対応して**テストケースを更新**していない
 - 副次的に影響を受ける形で変更される仕様もある
 - B) ソースコードの**変更方法に狙いを定めたテスト**を行っていない
- ベースとなるテストケースが存在しない場合は、A)の対応はできないことがある
 - 今回の追加機能や変更に関連した部分だけでも、テストケースを作る方法はあるが...
- 「**品質保証**」のテストの他に、「**バグ出し**」のテストの認識がなければB)の対応は省かれる
 - そのためには、変更箇所、変更内容、変更方法の情報が必要

1.10 どこをどのように変更したのかわからない

- 現実に、ソースコード上で**該当箇所を見つけ次第に変更**しているために、**どこを、どのように**変更したという情報が残されていない
- 変更の担当者が**気づく範囲**で変更しているだけで、その範囲を**チェック**できていない



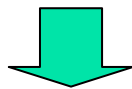
新規の機能が作動することは確認できたが、不正な変更を検出するためのテストができていない



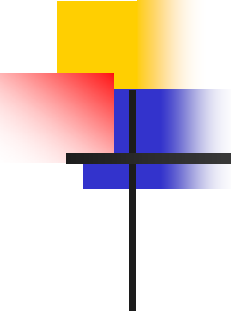
客先や市場でトラブルとなって表面化する

1.11 テスト工数が確保されない

- 派生開発では変更量に対して期間が設定される傾向があり、基本的に**開発期間は短く設定される**
 - 変更量は少なくとも、ソフトウェアの規模によっては**テスト工数は大幅には省けない**
 - 品質保証のためのテスト
 - 変更方法を狙い撃ちしたテスト
- そこで実施されているプロセスが「変更」要求にマッチしていないために、**テスト工数を浸食**してしまう
 - 残された期間でやれる事しかできない



市場でのトラブルとなって表面化する

- 
1. 何がテスト作業を混乱させているのか
 - そこで書かれている要求仕様の出来映えに端を発している
 - 派生開発において変更の内容がテストに反映されていない
 2. 要求仕様の表記法によってテスト作業が変わる
 - USDMによる要求仕様の表記
 - XDDPによる派生開発プロセスにおける「3点セット」の効果

2.1 USDMによる要求仕様書の特徴

- 要求を“**振る舞い**”で表現する
 - 「**理由**」をつけて要求を補強する
- 要求を**分割・階層化**して要求の範囲を制御する
 - 4つの分割基準を提供する
- 要求と仕様を**階層構造**で表現する
- 範囲が制御された**要求の中で仕様を抽出**する
- 仕様が発散しないように**<グループ>**でまとめる

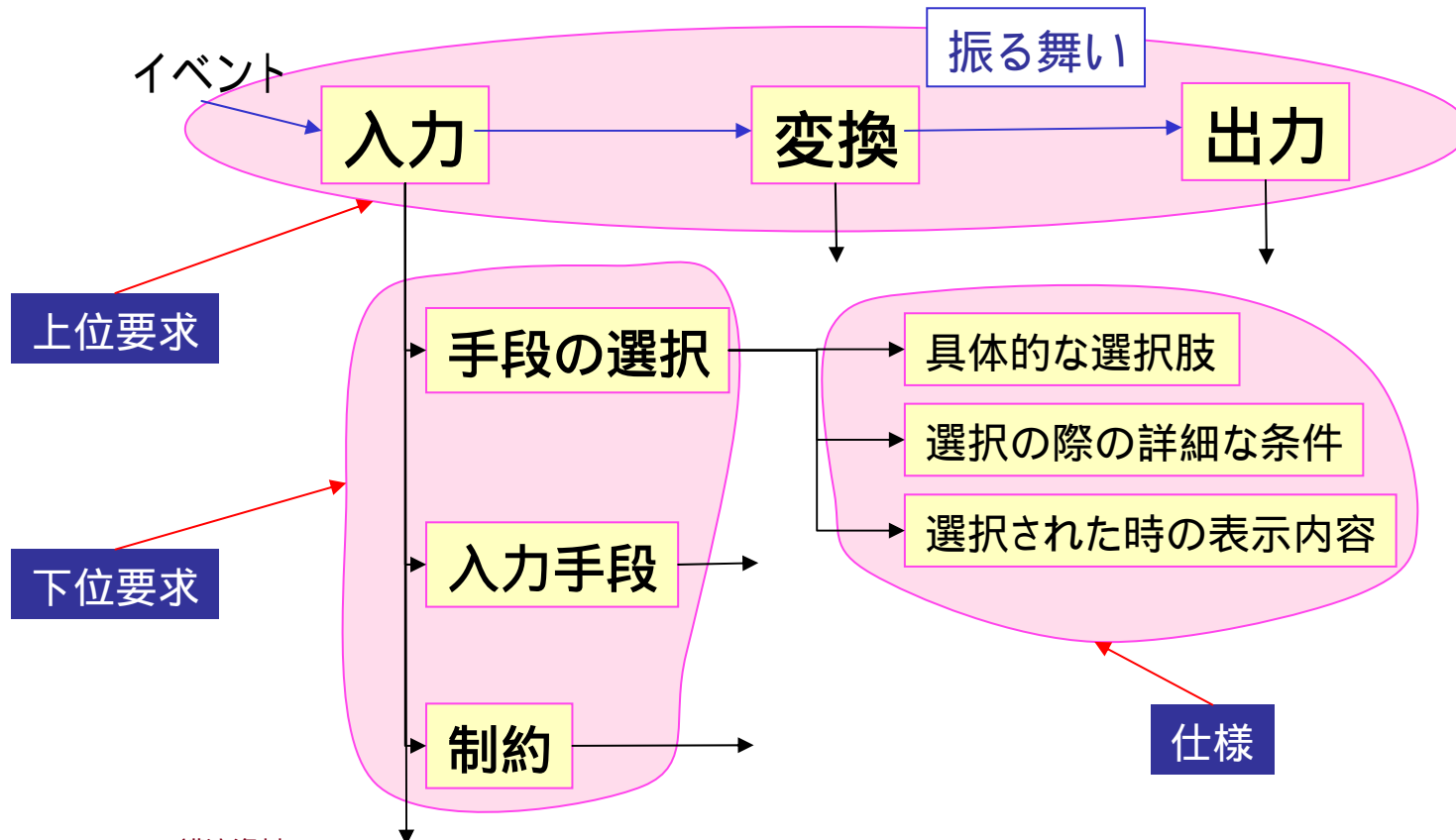
要求仕様書とは、「要求」すなわち実現して欲しいこと (Requirement) について、“作ることの関係者”が認識を特定 (Specify) できている文書。このとき、「関係者」も特定されている。



「機能仕様書」と違う

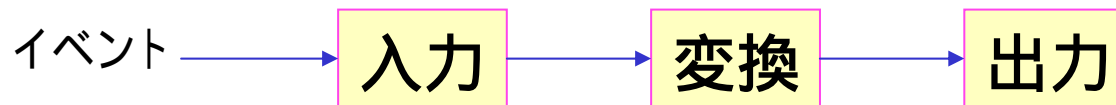
2.2 “要求”を階層構造で表現する

- 要求と仕様を階層構造で表現することで、一般の人にもイメージしやすいものとなる



2.3 上位要求は“振る舞い”で表現する

- 上位の要求を“振る舞い”で表現することで、テストでの**操作がイメージしやすくなる**
 - “振る舞い”とは、
 - 「イベント」に始まって「入力」→「変換」→「出力」の「シナリオ」の性質をもつ一連の動作



要求	RSV01	予約操作を終了した時は、予約を確定させて登録されているメールアドレスに予約結果を送信する。
----	-------	---

- “**イベント**”とは、振る舞いを開始する刺激(ただし、「要求」の中に表現するとは限らない)
 - ボタンの押下などの事象の検出、データの到着、タイムアウト など

2.4 “シナリオ”が動きのイメージを想像させる

- **上位要求では時系列を持った“シナリオ”を表現する**
 - 下位要求では、個々の「シーン」での**実現内容**にフォーカスする
 - 「理由」が要求のブレを矯正する効果を発揮する

要求	MAL01	事前に指定された受信および送信した電子メールをキーワードで検索してメーラー上で再利用したい	
	理由	メールが多くて、関連するメールを探せない	
	説明		
	要求	MAL01-01	検索対象のメールボックスを指定してグループ化できる
		理由	無関係なメールで振り回されたくない
	要求	MAL01-02	いくつかのキーワードを組み合わせて検索できる
		理由	可能性のあるキーワードで探したい
	要求	MAL01-03	検索結果から絞り込むための検索ができる
		理由	件数が多いときに目的のメールを探すのに手間取るから
	要求	MAL01-04	検索されたメールを選択し内容を表示する
		理由	目的のメールを絞り込めるような操作がしたい

2.5 すべての仕様は要求の下に表現される

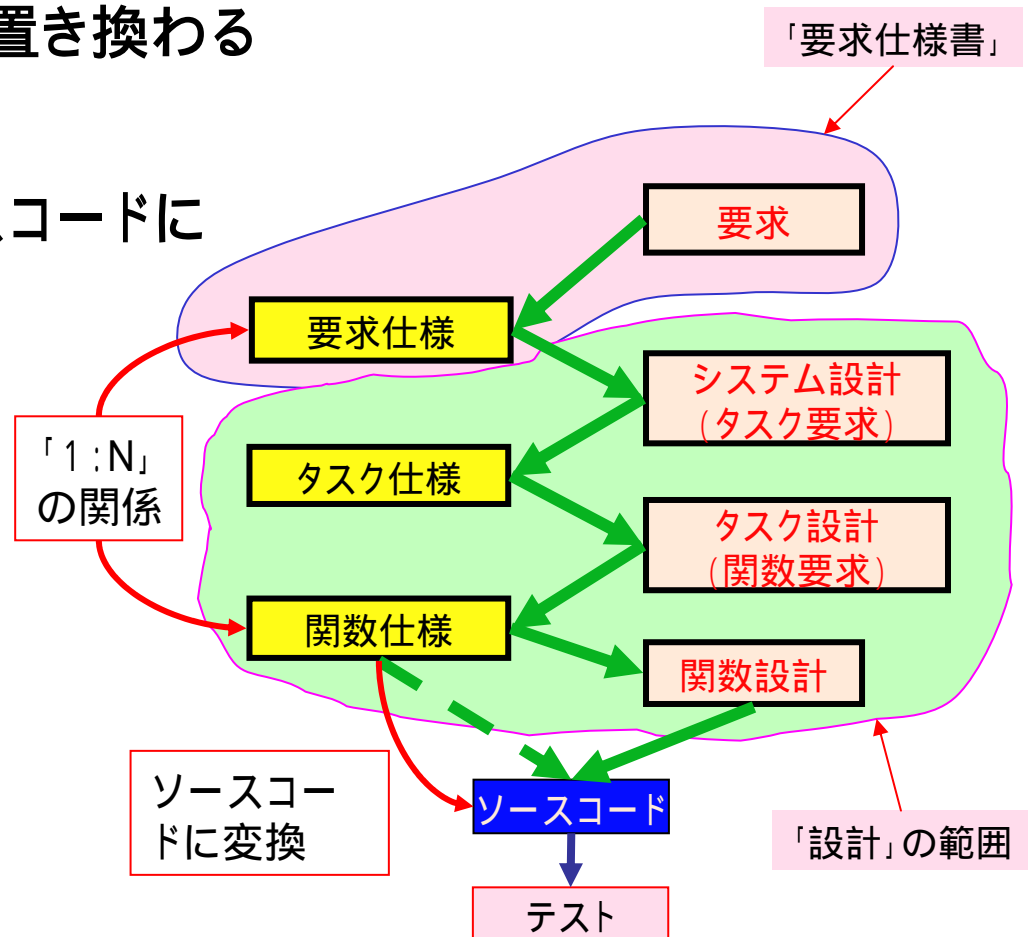
- 範囲が制限された要求の中で仕様を抽出することで
 - 仕様を抽出しやすい
 - 仕様の漏れに気づきやすい
 - 適切に仕様化されることで、他の要求の仕様と衝突していることにも気づきやすい

	要求	MAL01-02	検索のためのいくつかのキーワードを組み合わせて入力でき
		理由	可能性のあるキーワードで探したい
		MAL01-02-1	検索したいキーワードを入力できる
		MAL01-02-2	キーワードの文字数は最大31文字(英数字、記号)とする
		MAL01-02-3	複数のキーワードを「AND」と「OR」で繋ぐことができる [説明] キーワード自身は”一致”するものとして扱う
		MAL01-02-4	キーワードは最大8個まで指定できる

仕様が明らかなので、この場面でのテストケースを作りやすい

2.6 要求仕様とソースコードの関係

- 要求仕様書に書かれている仕様は、最終的に関数仕様置き換わる
 - [1:N]の関係
- 関数仕様は、ソースコードに置き換わる
- テスト作業は、要求仕様の内容が確実にソースコードに変換されていることを確認すること



2.7 テストケースの網羅性を見る方法

- 要求仕様と**テストマトリクス**を繋ぐことで、個々の仕様に対してどのような観点のテストを実施するかを**表現**できる

要求			同値テスト	境界値テスト	負荷テスト	状態遷移テスト
			●			
				●		●
				●		

- テストケース番号
- 関連する状態遷移番号
- 「直交表」番号
- 他の組み合わせを表すマトリクス番号



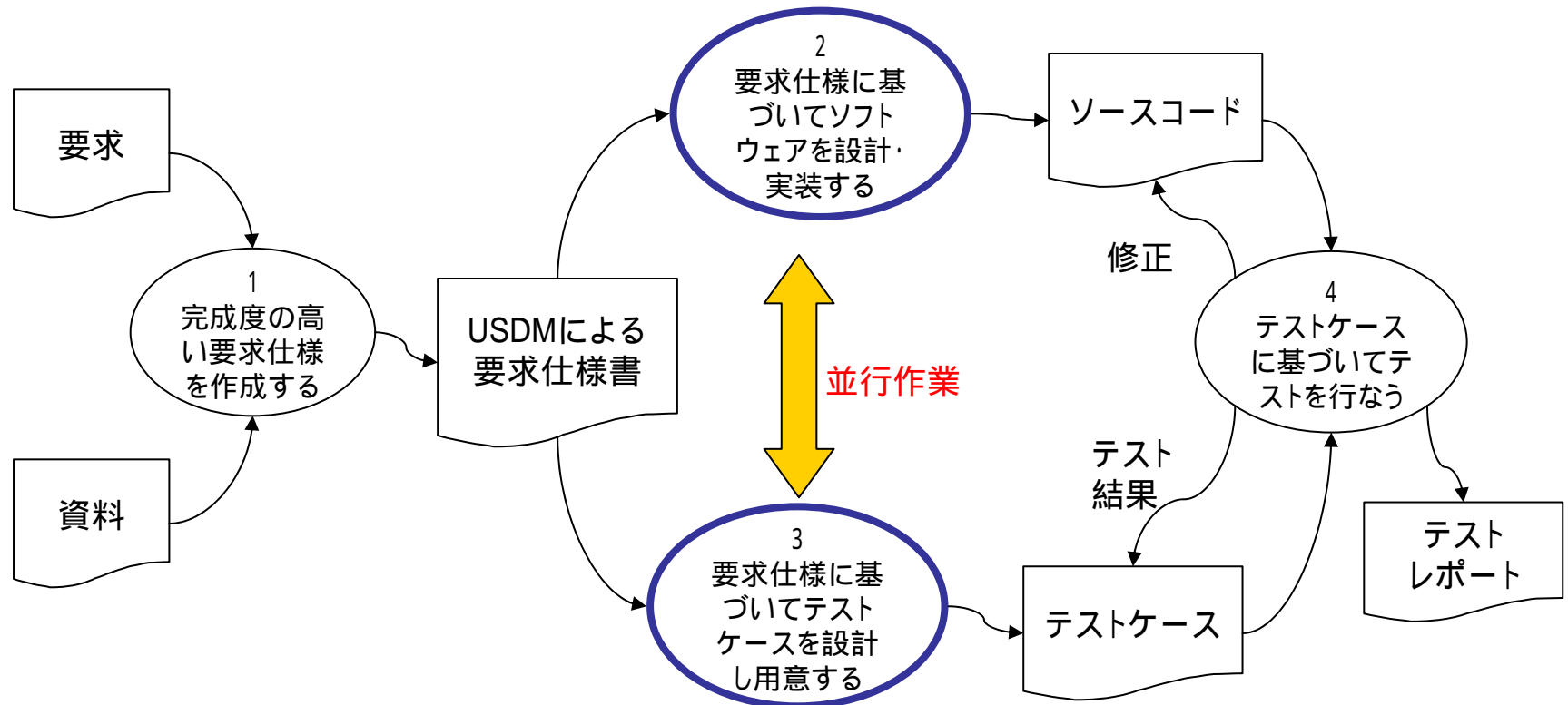
テストケースの網羅性を示す効果

要求仕様書

テストマトリクス

2.8 設計グループとの並行作業が可能となる

- 「要求仕様作成プロセス」の中で完成度の高い要求仕様書ができることで、**設計チームの作業と並行してテストケースの設計作業**に取り掛かれる



2.9 検証可能の視点からのレビューも可能

- 要求仕様書が「ベースライン設定」される前に、**テストの責任者によって“検証可能”かどうかの視点でレビューする**
 - 実現が求められている「振る舞い」がイメージできるか？
 - そこに示された仕様でテストケースが作れるか？
 - 値の制限や限界値、既定値は明らかになっているか？
 - 関連するパラメータが記述されている箇所は明らかか？



要求仕様書の完成度を上げる効果大きい

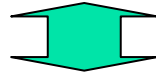
テストケースを設計するプロセスの最終調整ができる

2.10 事前にテストケースを公開できる

設計グループとの並行作業ができる



- クリティカルな機能や重要な機能に関する**テストケースを早い段階で作成し、設計グループに公開**できる
 - 設計グループとの間で事前に合意して機能を選定する



- 設計グループは、どのようなテストが行なわれるかを事前を知る事ができるので、最初から最適な設計ができる
 - 「**テスト指向設計**」が可能

2.11 変更要求仕様書がすべての変更を扱う

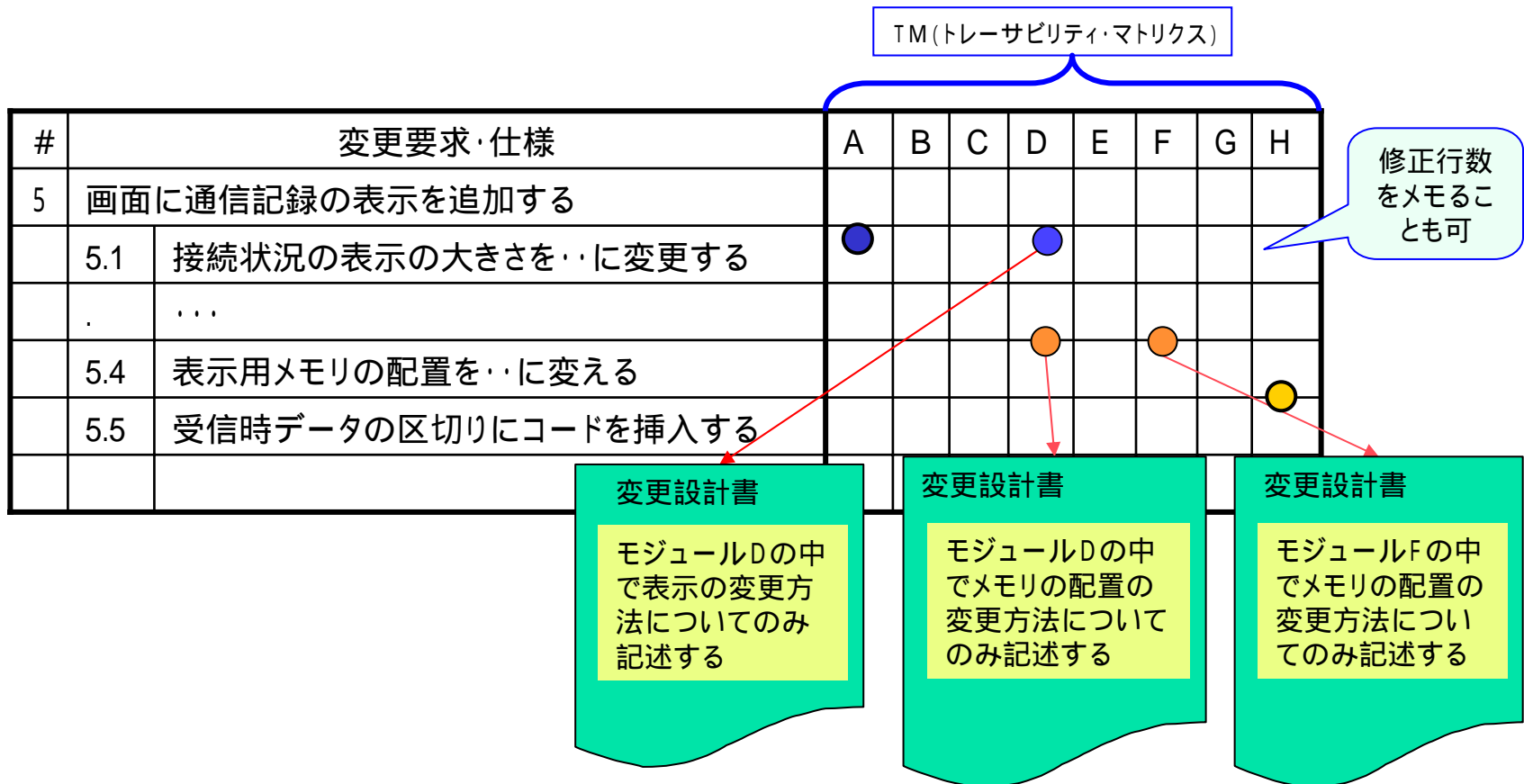
- 変更要求仕様では、**ソースコード・レベル**の仕様も「変更仕様」として扱う
 - **どの仕様が、どう変わるのかが見える**

要求	PRC05	撮影範囲とスピードアップで1台の防犯カメラでカバーする範囲を広げる	
	理由	モニターも含めて設置コストを下げる	
		PRC.05.1	防犯カメラの首振り角度を30度から45度に変更する
		PRC.05.2	DCモーターの電流値を から に変更して、首振り動作の速度を50%スピードアップする
		PRC.05.3	画像のブレを押えるためにスキャン速度を から に変更する

要求	COM30	通信の速度を から2倍にする	
	理由	解像度の向上でデータ量が80%増えたことに対応するため	
		COM30-01	セッション確立時の通信速度のパラメータを から に変更する
		COM30-02	タスク起動時に獲得する受信バッファを1本につき4Kから8Kに変更する
		COM30-05	DMAコントローラーに対する開始アドレスと転送サイズを以下のように変更する（具体的なアドレスとサイズの変更内容を示す）

2.12 派生開発では「3点セット」を活用する

- 「3点セット」が変更に関する**必要な情報を提供する**
 - 「変更要求仕様書」「トレーサビリティ・マトリクス」「変更設計書」



2.13 「3点セット」の役割

- **変更要求仕様書**
 - 具体的な変更箇所(変更仕様)が明記されている
 - この変更情報に基づいてテストケースを更新する
 - 関連して影響する箇所もまとめられているので一緒に更新する
 - 変更はソースコードとは限らない
- **トレーサビリティ・マトリクス(TM)**
 - 変更したモジュールに関係する機能をたどることができる
 - 変更の影響を受ける可能性からテスト範囲を調整できる
- **変更設計書**
 - 具体的な変更方法が明らかにされることで、「**耐性**」を狙い撃ちした**テストケース**を用意する
 - “今回限りのテストケース”も必要

付) テストケースの有効性も考慮すること

- **テストケースの有効性**

$$TCE = N_{tc} / N_{tot} * 100\%$$

参照: 「Validating and Improving Test-Case Effectiveness」
IEEE Software Jan/Feb 2001

N_{tc} : テストケースによって発見されたバグの件数
 N_{tot} : テストサイクルの中で発見されたバグの総件数

- 「TCE = 100%」という事は、すべてのバグがテストケースによって発見されたことを意味する
- 元来、テストケースは事前に想定できた範囲でしか用意されないため、100%という事は、そのテスト結果は信用できない可能性がある
- テスト結果などから、効果的なテストを追加して「TCE」を下げる

事前に適切なテストケースが用意されているからこそ、
 テスト作業の中でテストケースに依存しないテストが可能となる

(参考文献)

- 文献(単行本)
 - 「要求を仕様化する技術・表現する技術」(技術評論社)
- 雑誌
 - 「要求の仕様化入門」(Software People Vol.4 : 技術評論社)