

テストの意図、共有できていますか？
～ドキュメントレビューの視点から

森崎 修司

名古屋大学 情報科学研究科

本講演での「テストの意図」

- 「テストの意図」はテストケースの集合(テストスイート)、テストケースといった細かい粒度のテストの実施目的
 - なぜ実施するか
 - テスト対象が何ができることを確認しようとしているか
- テストの意図が共有できないとテストが妥当かどうか確認できない。
 - 網羅的なテストケースになっているか
 - ムダなテストケースはないか

本講演での「テストの意図」の例1

- 地下鉄の自動改札機の休日割引き切符のテストケース

...	分類	テストケース	...
意図なし	特殊切符	土曜 am 0:00は割引き切符不可	
		土曜 am 4:00は割引き切符可	
		...	
		月曜 am 0:00は割引き切符可	
		...	
...	...		

土日周辺で割引き切符が有効になってるか調べてるってことか..



...	意図	テストケース	...
意図あり	ドニチエコきっぷが割引きとなっているか確認する	土曜 am 0:00は不可	
		土曜 am 4:00は可	
		...	
...	...		

祝日と8日も適用日だからテストケースを増やさないと！

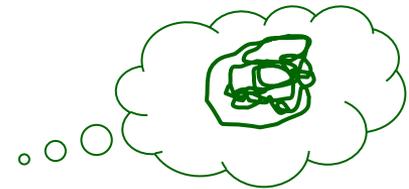


テストの意図の追加

- ただし、やみくもに意図を追加すればよいというわけではない。

...	大目的	中目的	小目的	大分類	中分類	小分類	テストケース	...
						ドニチエコ	...	
						きっぷが割	...	
						引きとなって	...	
						いるか確認		
						する		
						...		

たしかに。
情報は多い方がいい！
大目的、中目的、小目的、
大分類、中分類、小分類・・・
をつけよう！



現状に合わせた対策(1/2)

- 現状: 実施報告書以外にドキュメントがなく、テスト実施結果だけが記載されているドキュメントしかない。
- 対策: 分析、整理の経緯を加える。

テスト計画書兼実施報告書

ID	機能大分類	機能中分類	機能小分類	テスト項目	実施日	結果	デバッグ経緯	修正担当	原因・対応
1-1	起動プログラム	通常起動	ログ出力	読み込んだ設定ファイルに記法エラーがなく、読み込んだファイルがログに出力されているか？					

分析や整理に必要な情報

...	...	
...	...	
...

作成者

テスト対象、テストケースはわかったけど、この前提となる情報がほしい

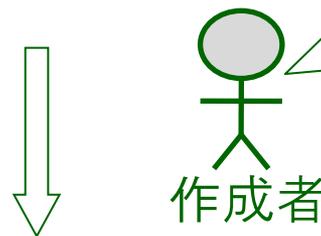
チームメンバ
(レビューア)

現状に合わせた対策(2/2)

- 現状: テストの意図がそもそも考えられていない。
- 対策: まずテストの目的や意図を整理できるようになる。

ID	機能大分類	機能中分類	機能小分類	テスト項目	詳細
1-1	起動プログラム	通常起動	ログ出力	読み込んだ設定ファイルに記法エラーがなく、読み込んだファイルがログに出力されているか？	引数 る。ロ 確認

前バージョンのテストドキュメント



作成者

前のバージョンであるってことは今回のバージョンでも使えるはず☆
どうせ減らしたら理由を聞かれるんだろうし...

ID	機能大分類	機能中分類	機能小分類	テスト項目	詳細
1-1	起動プログラム	通常起動	ログ出力	読み込んだ設定ファイルに記法エラーがなく、読み込んだファイルがログに出力されているか？	引数 る。ロ 確認

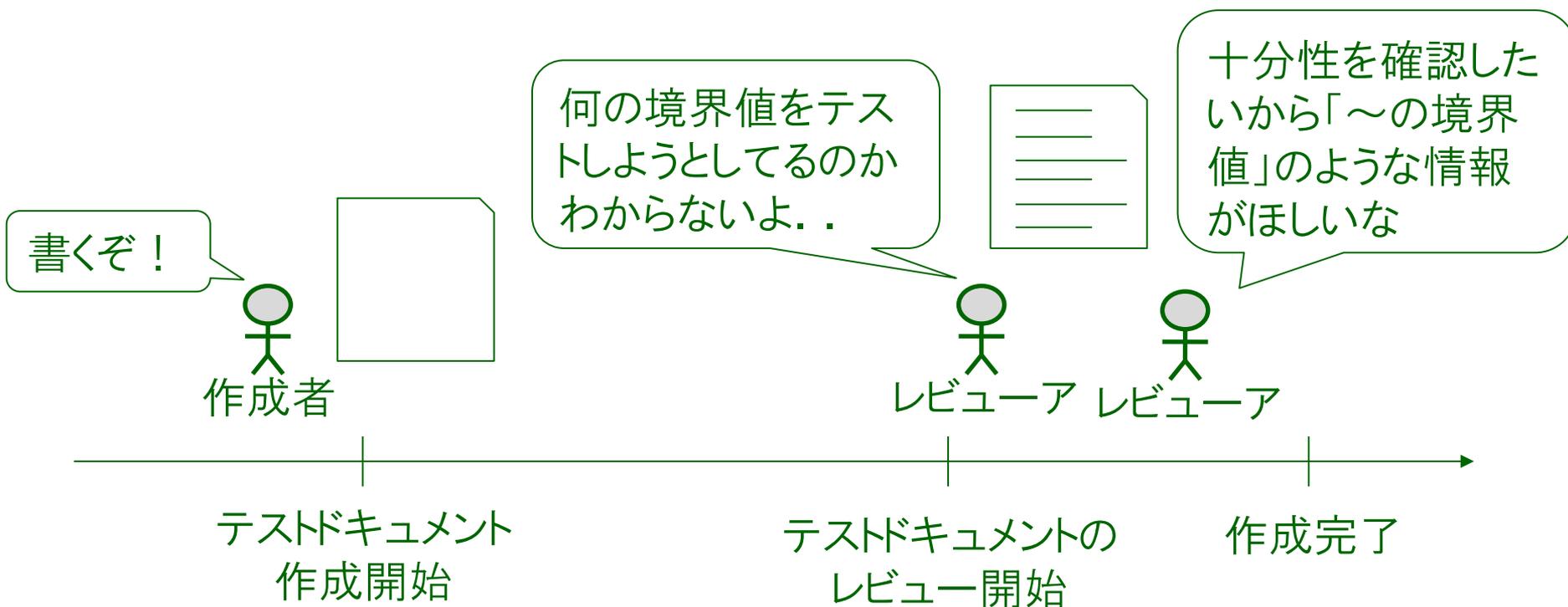
今回のテストドキュメント

本講演でのアプローチ

- 仕様、設計のドキュメントレビューの課題と解決策を示す。
 - 課題
 - ただなんとなくレビューしている。
 - 自由度が高すぎる。
 - 解決策
 - 目的を設定し、注力する。
 - 検出すべき問題を設定し、検出するためのシナリオを準備する。
- テストに付加する情報を会場一体で考える。
 - 課題
 - テストの意図を共有するための情報が揃っていない場合がある。
 - 解決策
 - 適切なテストかどうかを考える上で必要(最小限)な情報を追加する。

テストドキュメントに付加する情報

- レビューで何を確認したいかを明らかにすることで、テストドキュメントに何を記録しておくべきかを考える。



ドキュメントレビューの定義

- プログラムを動作させずに中間成果物に問題がないかを目視で確認する静的解析技法
- 目的
 - 不具合の修正コストの低減
 - 状況把握と承認
 - 保守性向上
- インプット
各種ドキュメント(ソースコードを含む)
- アウトプット
インプットのドキュメントに含まれる問題点のリスト
- 課題
 - 自由度が高く、効果や効率が参加者や場の設定に左右されやすい。

レビューでの指摘例

レビューでは、箇所の指定と間違っている理由を指摘する。

工程	対象ドキュメントと指摘例
計画	文書ID PS-20081121-1 の顧客向け提案書8 ページに記されている冗長化を実現するためには、13 ページのシステム構成ではサーバが足りない。
要求	文書ID CS-1002-4 の要件定義書15 ページ24 行目「リクエストから5 秒以内に結果もしくは、タイムアウトしたことを示す結果をユーザに提示すること」と22ページ12 行目「国際ネットワークを通じた処理の場合、待機時間7 秒以上とすること」を同時に満たせない場合がある。
設計	文書ID D-1002-12 の詳細設計書42 ページ5 行目、及び、表3-2 の「次の状態遷移まで現行の値を保持する」という記述があるが、47, 52, 53 ページ、及び、表4-5, 図4-1には状態を保持するための仕組みが記述されていない。
コーディング	receive request.c 727 行目、関数send status への引数に変数status となっているが、現在の状況はstatus ではなく、current status に入っている。
テスト (テスト計画)	文書ID TP-1002-122 のテスト計画書3 ページ項目B-10 のテスト項目「ステータス0xBBA8 に対して例外処理が実行されることを確認」とあるが、該当する例外処理が複数あるため、例外処理コード等の具体的な記述が必要である。
運用	文書ID OP-1002-202 復旧手順書133 ページハードウェア障害復帰時のデータ回復のために必要な情報がウィークリーバックアップとされているが、このバックアップは差分のみを記録しているため、元のデータを復元できない。

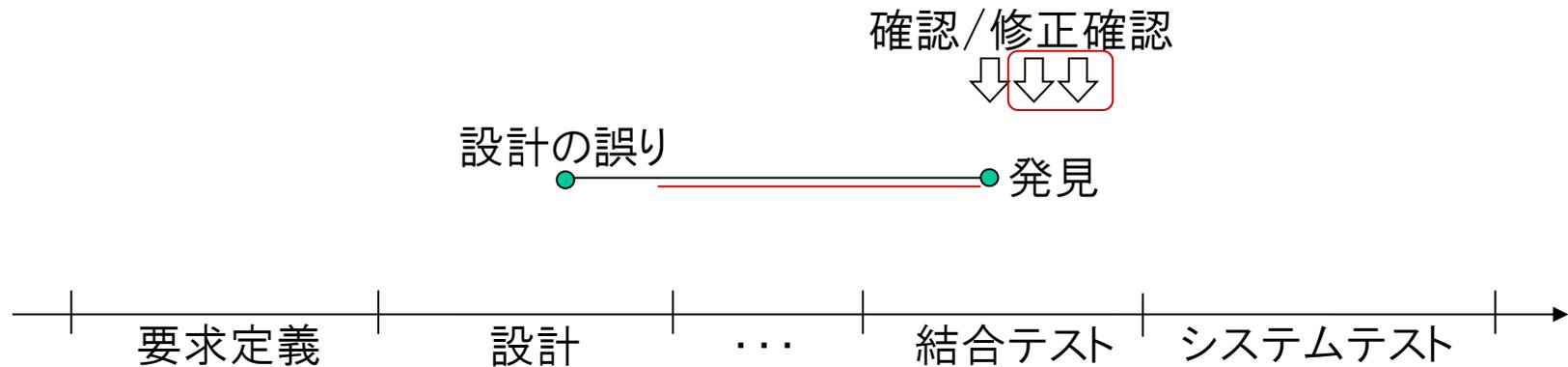
レビューの効果が得られる前提

- どのような問題を検出すべきかが明らかになっていて共有されている。
 - どのような問題を検出すればよいか明確になり共有されている。
- 人間関係が持ち込まれない。
 - 「生意気な作成者を指摘でねじ伏せる」といった本来の目的とは異なる理由で問題が指摘されない。
- 適切に時間配分されている。
 - 「前半で息切れしてしまい後半はおろそか. . .」という状態になっていない。
- 詳細は「間違いだらけの設計レビュー」をご覧ください。

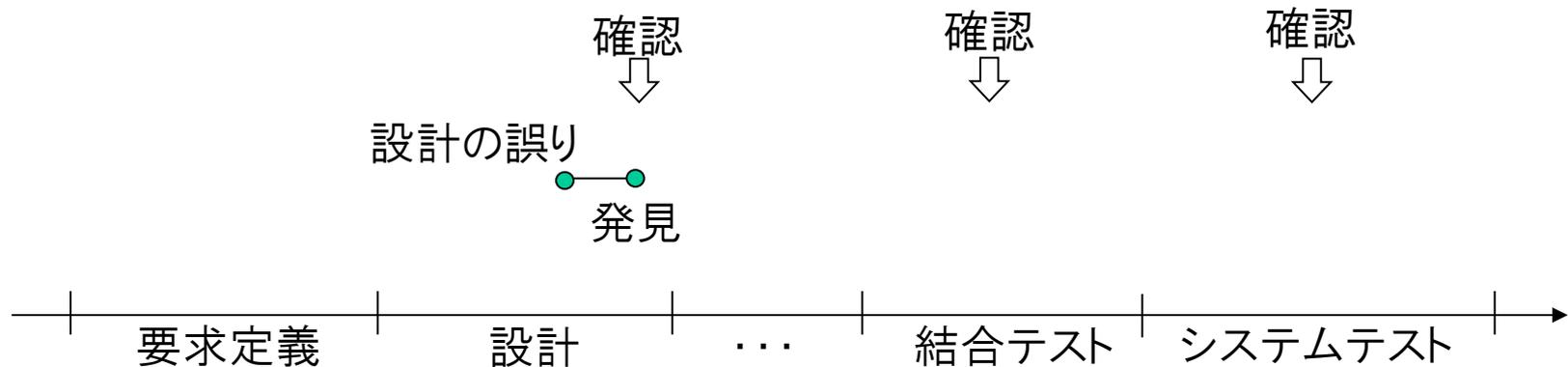
レビューの効果

- テストでの修正確認コストの低減
- 誤りに基づいた設計や実装コストの低減

テストだけで確認する場合

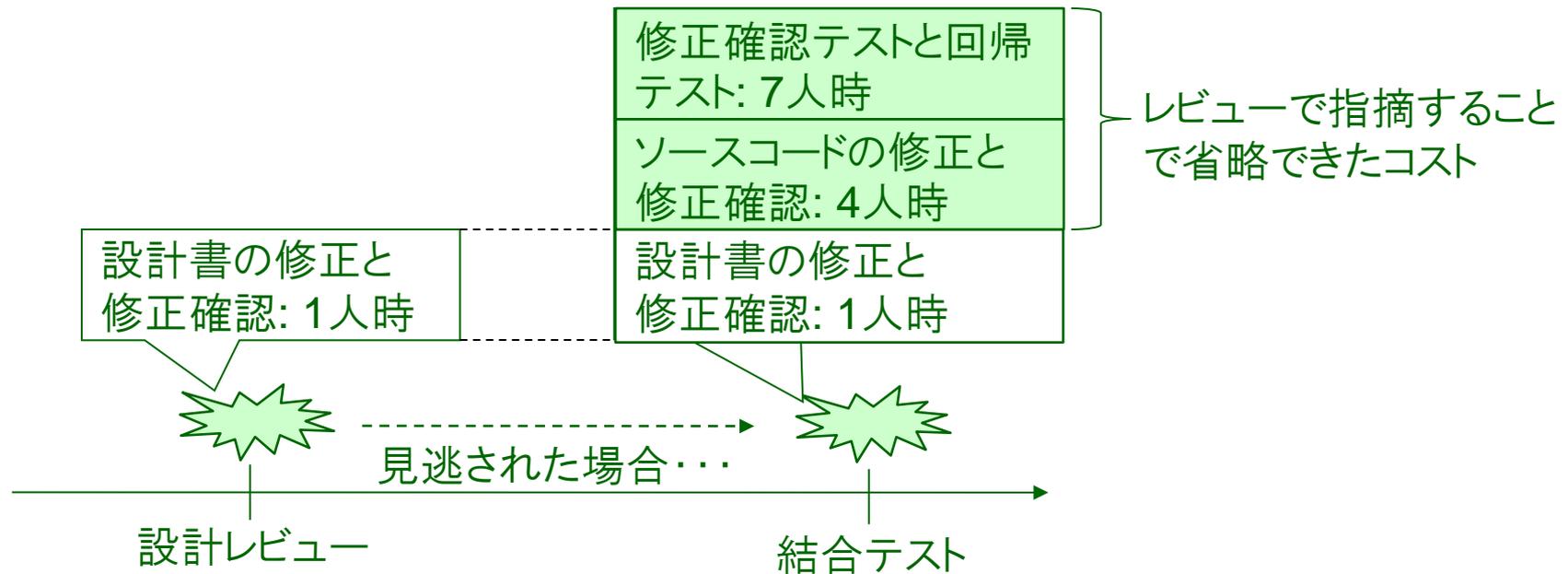


レビューも併用する場合



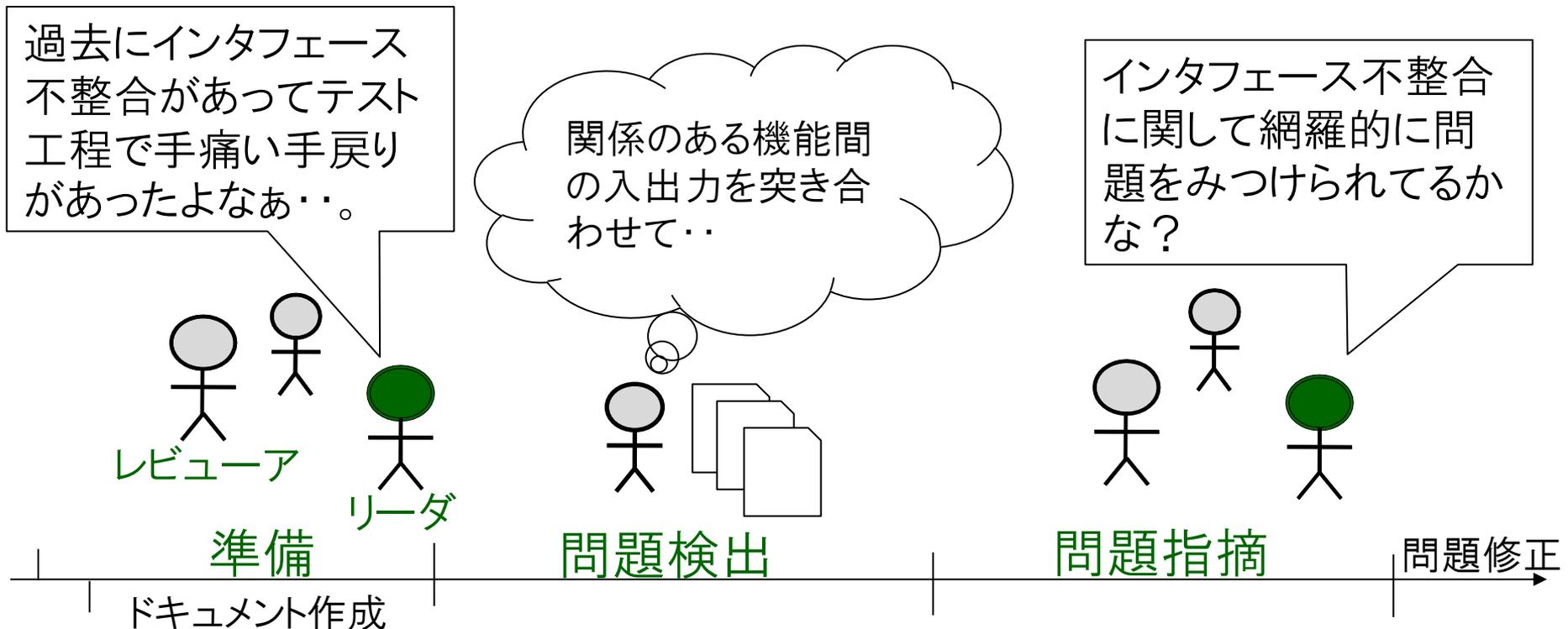
レビューの効果が見逃される例

- レビューで見逃してテストで発見された場合に、修正コストが大きくなる欠陥を発見できると、効果が得られる。



ドキュメントレビューのプロセス

- 準備: どのような問題を検出すべきか検討する。
- 検出: 問題を検出する。
- 指摘: 検出した問題を会議で指摘する。



問題検出シナリオを使ったレビューの手順

- 準備
 - 「絶対に見逃したくない」問題種別を決める(10件以下)。
 - その問題種別の検出方法をシナリオとする。
 - シナリオをレビューアに割り当てる。
- 問題検出
 - レビューアがシナリオにそって個別に問題検出する。
(決めた問題種別以外の問題も検出してよい)
- レビュー会議
 - 個別に検出した問題を指摘し共有する。
 - 準備で決めた問題種別で検出漏れがないか確認する。

問題検出シナリオ

- 問題検出シナリオを使ったレビュー手法は多数ある。
- Perspective-based reading
 - 開発に関わる立場からシナリオを作成する。
 - 「顧客」の立場「プログラマ」「テストエンジニア」・・・
V. Basili, S. Green, O. Laitenberger, F. Lanubile, F. Shull, S. Sorumgard and V. M. Zelkowitz, The Empirical Investigation of Perspective-Based Reading, International Journal on Empirical Software Engineering, Vol. 1, No. 2, pp. 133-164. (1996)
- Defect-based reading
 - 要求文書を対象とし検出したい問題種別を定義し、各々のシナリオを作成する。
 - 漏れ、曖昧、誤りを問題種別としたシナリオを定義している。
A. Porter, L. Votta and V. Basili, “Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment,” IEEE Transactions on Software Engineering, Vol. 21, No. 6, pp. 563-575 (1995)
- 日本ではシナリオを明示的に使ったレビューは浸透していないが暗黙の役割分担がシナリオを使ったレビューに該当する

当研究グループでのシナリオ作成手法

- 保守開発を前提にし、過去のバージョンの不具合で修正工数が大きかったものをシナリオ作成のインプットにする。
森崎 修司, 久保 匡, 荻野 利彦, 阪本 太志, 山田 淳 “過去の不具合の修正工数を考慮したソフトウェアレビュー手法,” 電子情報通信学会論文誌 D Vol.J95-D, No.8, pp.1623-1632(2012)
- Fault-proneモジュール予測の結果、モジュールごとの不具合を含む可能性をシナリオ作成のインプットにする。
笠井 則充, 森崎 修司, 松本 健一 “目視評価と判別モデルを組み合わせたfault-proneモジュールのランク付け手法,” 情報処理学会論文誌 Vol.53, No. 9, pp. 2279-2290(2012)
- テストで発見、修正したときの回帰テスト件数をシナリオ作成のインプットとする。
田村 晃一, 亀井 靖高, 上野 秀剛, 森崎 修司, 松本 健一, “修正確認テスト規模の低減を目的としたコードレビュー手法,” 情報処理学会論文誌, Vol.50, No.12, pp.3074-3083 (2009)

参考：問題検出シナリオとチェックリスト

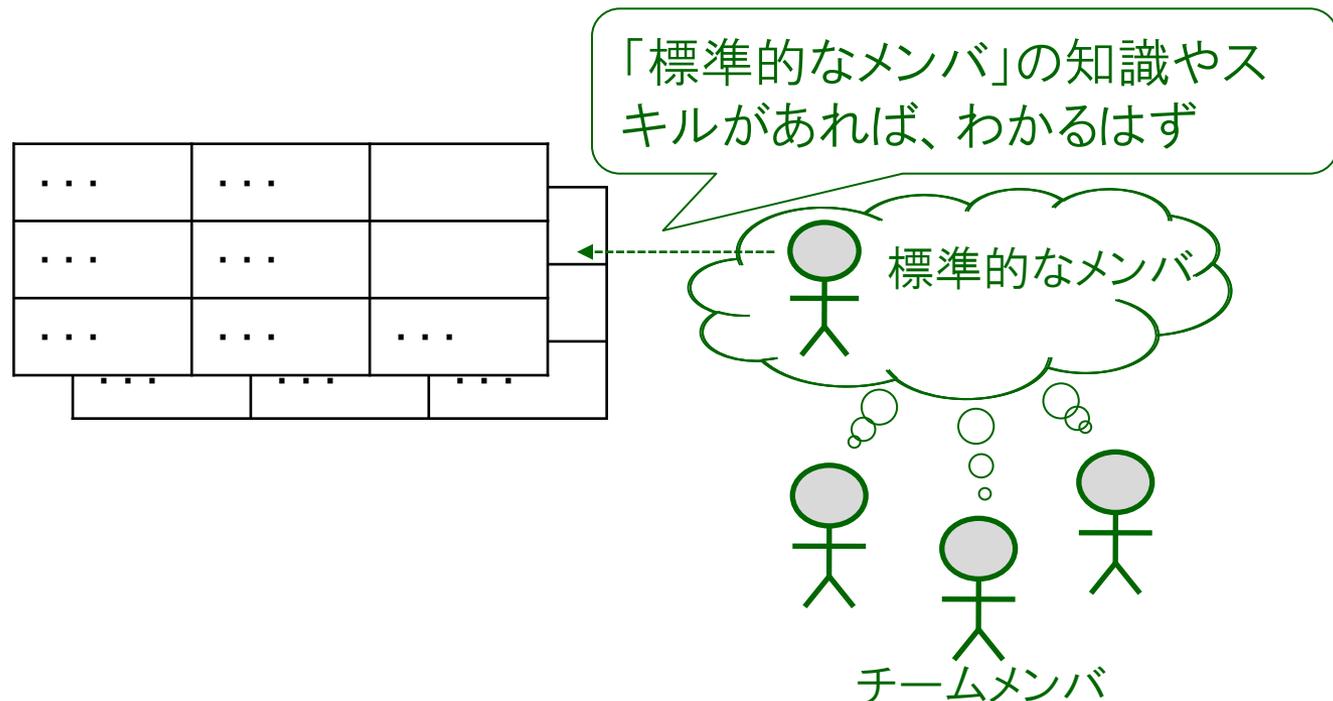
- レビューで確認すべき質問をリスト化したもの
- 時間経過とともにチェックリストの内容を見直す必要がある
- うまく使えば読み進める方針となる。
 - 「どこをどのように確認すれば質問に回答できるか？」を事前に検討する。
 - 検討結果にもとづいてレビュー対象を見る。
- チェックリストには確認すべき場所を指定してあるものもある。
- 質問数が多いと網羅的なレビューが難しくなるので工夫が必要である。
 - 優先順位をつける。
 - 類似の質問項目は同時に調べる。

ドキュメントへの追加情報の選択手順

1. ドキュメントとして必須情報は書く。
 - エビデンス、納品条件、法令、プロジェクトルール、社内標準、プロジェクトの判断…
2. 「標準的なメンバ」を想定し、「自明」、「常識」に該当する部分は書かない。
3. テストドキュメントのレビューで効果が得られるか確認する。
4. テストドキュメントのレビューで必要となる情報を書く。

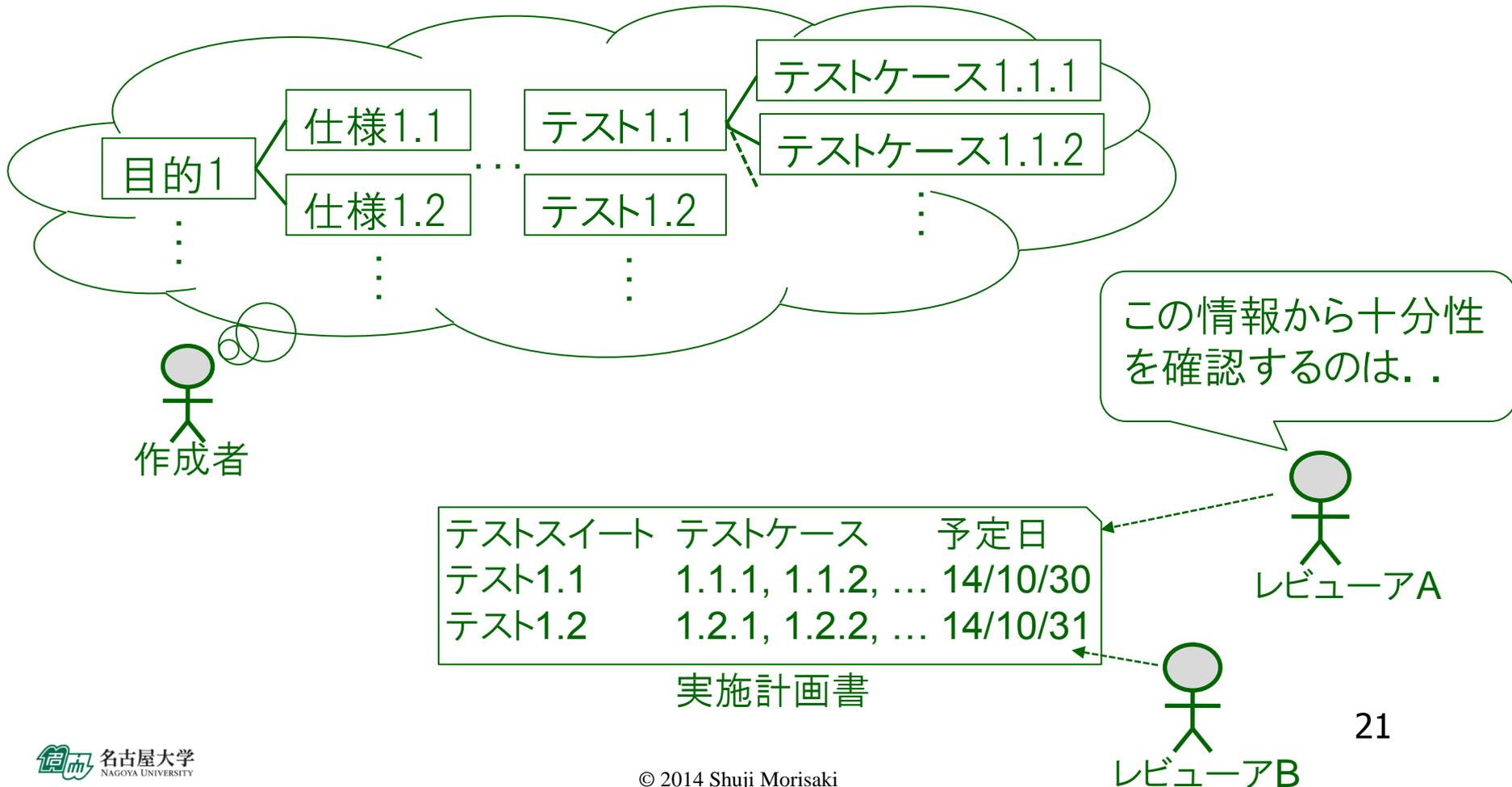
自明な情報と常識

- 「標準的なメンバ」を想定して合意する。
 - 想定したスキルや知識を前提として「自明」、「常識」に該当する部分は書かない。
 - 異動が多い、委託先のメンバのスキル、知識がばらつく等も考慮する。



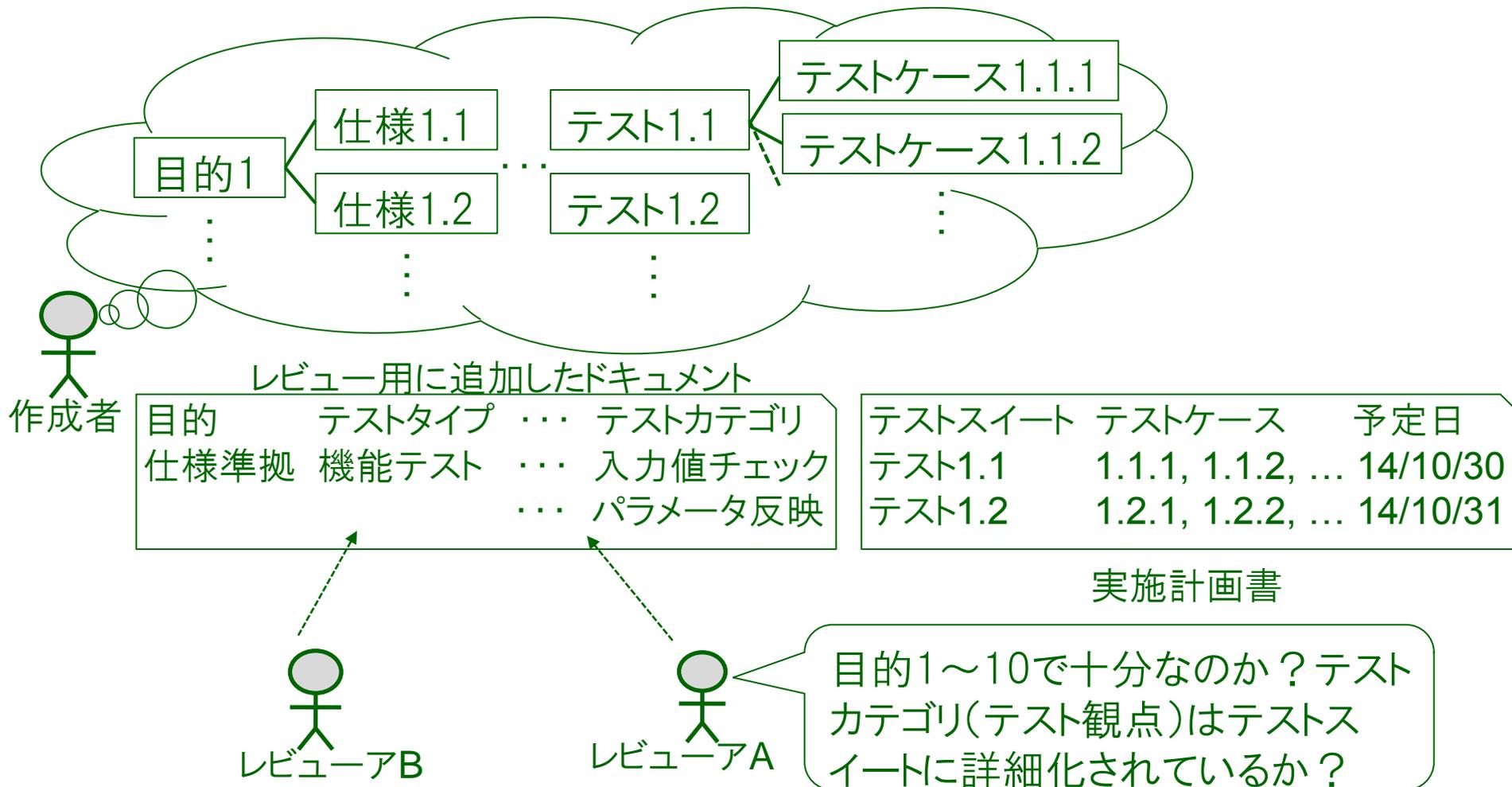
ドキュメントレビューの前提(1/3)

- レビューで確認したい内容が記載されていなければレビューの効果は期待できない。



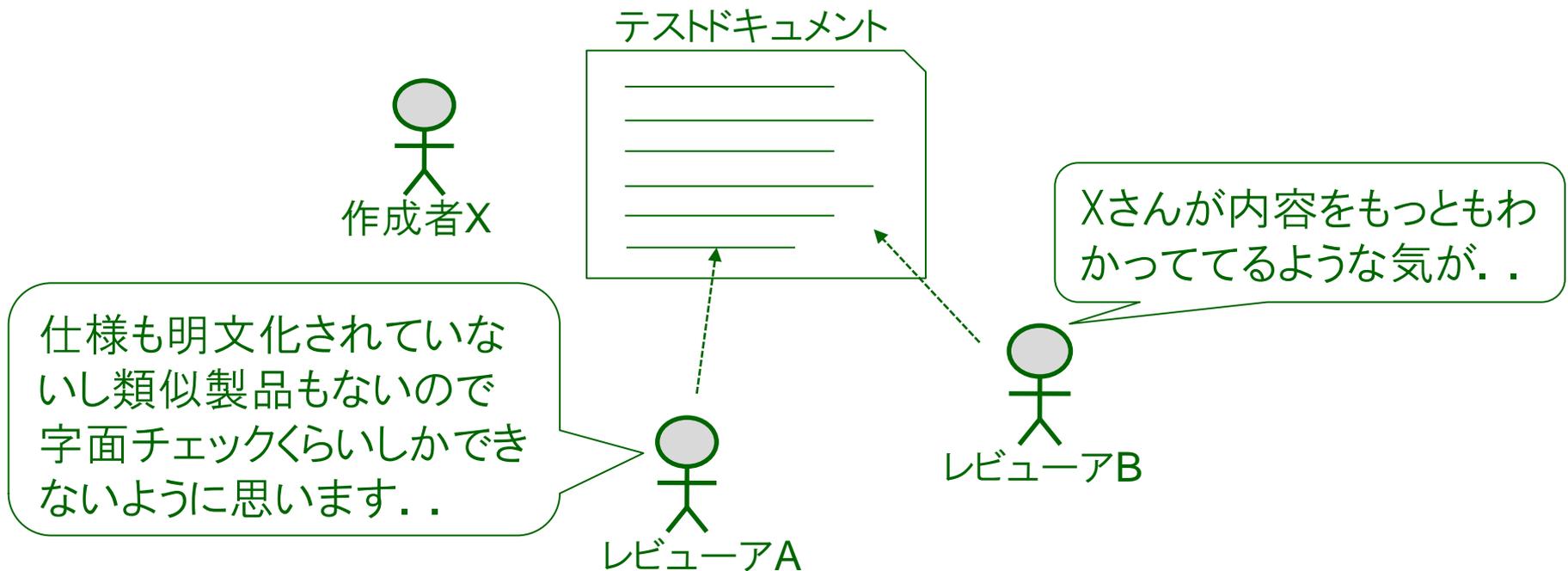
ドキュメントレビューの前提(1/3)

- レビューで確認したい内容を記載したドキュメントを準備する必要がある。



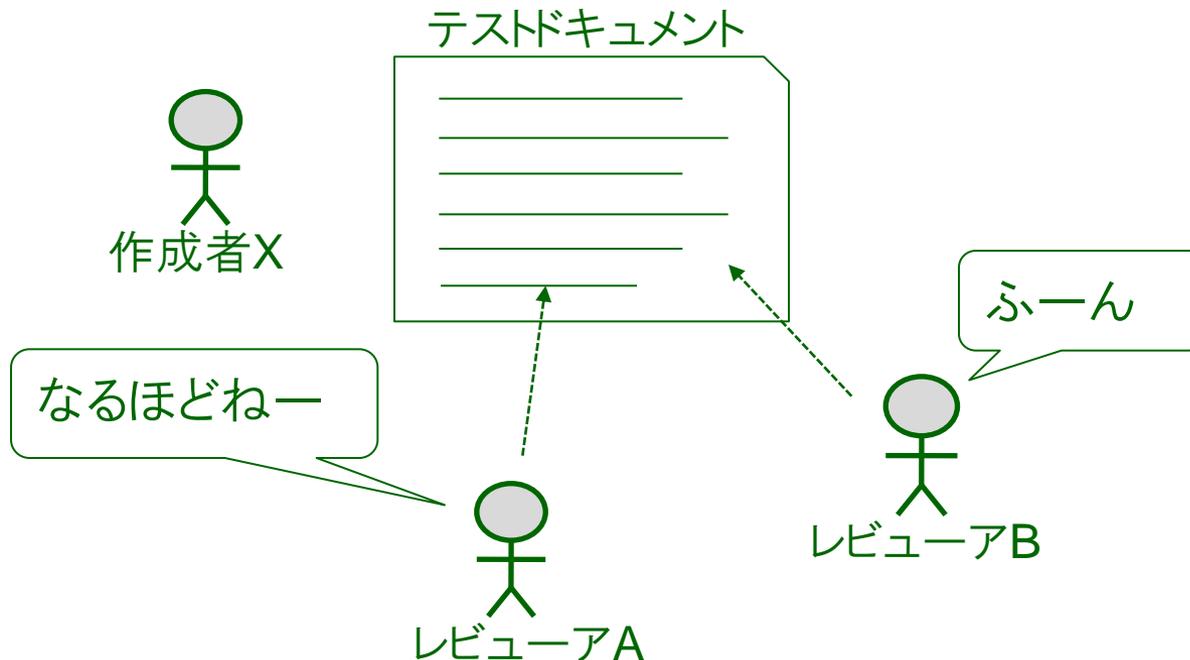
ドキュメントレビューの前提(2/3)

- 他のレビューアが確認すれば本質的な問題を検出できる。
→ 形式的チェックに終わるようであれば、効果は得られにくい。



ドキュメントレビューの前提(2/3)

- 他のレビューアが確認すれば本質的な問題を検出できる。
→ 情報共有程度に終わるようであれば、意図を書くのではなく、説明をしてQ&Aすることも選択肢になる。



テストドキュメントの検出シナリオ例(1/2)

- テストの十分性を確認するシナリオ
 - テスト漏れがないかを確認する。
 - テスト対象機能、テストスイート、テストケース、実施方法をはじめ様々な粒度で漏れがないか調べる。
- テスト実施の効率性を確認するシナリオ
 - 必要のないテストがないかを確認する。
 - 期待結果を同時に確認できるテストケース、省略できるテストケースはないか調べる。
- 要求/仕様との対応を確認するシナリオ
 - 要求/仕様の変更時に影響を受けるテストケースを確認する。
 - どのテストケースがどの要求/仕様を満足しているかが明らかになっているか調べる。

テストドキュメントの検出シナリオ例(2/2)

- 対応するモジュール、サブシステムを確認するシナリオ
 - 回帰テストの選択基準を明らかにする。
 - テスト対象となるモジュールやサブシステムが明確か調べる。
- 実施環境を確認するシナリオ
 - 実施環境が限定されるテストを明らかにする。
 - 実機や稼働環境によるテストが必要なテストスイートを選別する。
- 保守性を確認するシナリオ
 - 次バージョン以降で再利用しやすく作られているか調べる。
 - 再利用の可能性の高い箇所を特定し、その箇所と他の箇所の間の相互依存が極小化できているか確認する。
- . . .

テストドキュメントの検出シナリオ例 追加

- 母体か派生(特定仕向け)かによりテストの濃淡を変えられているかを確認するシナリオ
- 多重障害を想定したテストになっていないかを確認するシナリオ

シナリオの例と追加情報の例

- テストケースの十分性を確認するシナリオ
 - テストスイートとテストベースをもとに漏れがないか確認する。
 - 入出力、実行条件、状態、イベント
 - 確認する情報
 - 入出力、実行条件、状態、イベントを確認の様子を記録した情報(一覧表等)
- 追加情報
 - 入出力、実行条件、状態、イベントを確認した根拠となる情報(一覧表)

テストスイート: 複数のテストケースを分類したもの

テストベース: 仕様書をはじめとしたテストの根拠になるもの

十分性を確認するシナリオの例

- テストスイートの十分性を確認するシナリオ

...	意図	テストスイート	...
	ドチエコきっぷが割引きとなっているか確認する	土日	
		祝日	
		8日	
	...		

年末年始と盆休み期間中の土日ダイヤが抜けてない？
業務仕様の「特別ダイヤ」のところに注記されてるよ



記録を残す情報の取捨選択

- 対象

- すべてのテストケース、テストスイートに同種の情報を残さなくてもよい場合がある。
 - 例) ユーザにとって重要な機能、フィーチャに限定して情報を追加する。
- 暗黙的な依存関係もあるので、注意しながら情報を減らす。

- ドキュメント化のコスト

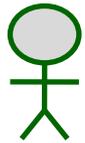
- 見栄えがきれいで丁寧なドキュメントの作成には時間がかかることが多い。
- 得られる効果に応じたドキュメントを選ぶ。

テストツールへの入力／テストコード

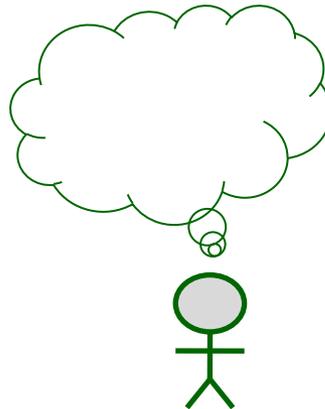
- テストケース以上に詳細であり、意図や経緯を書かなければわかりにくいものもある。
- 自明なものを除き、ファイルに説明を記入するかメモを添付する。
 - テストツールへの入力: ファイル名、アノテーション
 - テストコード: テスト用クラスに単純に“Test”とつけるだけでなく、意図も含まれるような命名規則を合意しておく。
 - コメントに記入すべき情報を合意しておく。
 - ドキュメントの更新日時のみが古いものは、最新の情報が反映されていないとして、参考程度にする。

「考える」と「記録に残す」

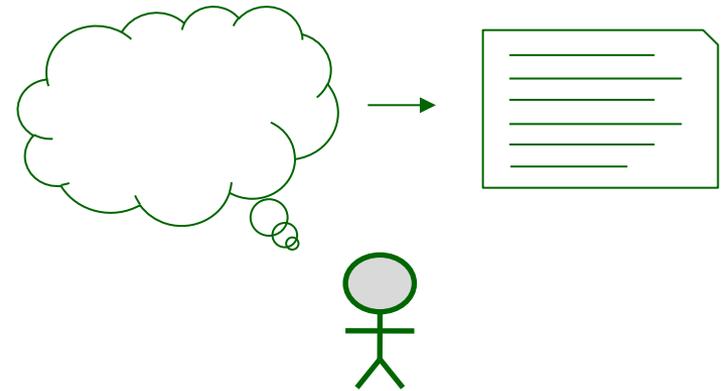
- 記録に残すコストが大きすぎる場合でも途中まで考えることはできる。



考えない



考えるけど
記録に残さない



考えたことを
記録に残す

まとめ

- レビューから遡ってテストドキュメントにどういう情報を残すべきかを考える概略を述べた。
- 仕様、設計書を対象としたドキュメントレビューを紹介した。
 - レビューの効果が得られる例
 - 問題検出シナリオによるレビュー
- テストドキュメントの問題検出シナリオを整理し、何を記録しておくべきかを考えた。
 - 問題検出シナリオとして十分性や効率性を挙げた。
 - 十分性や効率性の確認に必要な情報が何かを挙げた。
- 本資料の洗練への協力、この場を作っていただいたソフトウェアテストシンポジウム東海実行委員の皆様に感謝する。