オフショア開発におけるテスト改善

～オフショア版テスト改善「３点セット」の紹介～

TIS株式会社　技術本部　先端技術センター　鈴木三紀夫

2010年1月29日

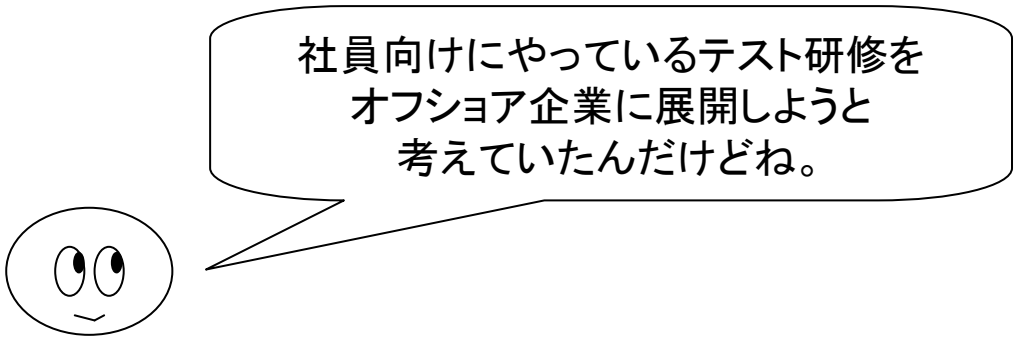
AGENDA

1. オフショア開発の品質は良い？
2. 何が問題なのか？
3. 三点セットの説明
4. あるプロジェクトの進め方

オフショア開発の品質は良い？

■ 結合テスト以降で「単体テスト不足」の不具合は一桁

- オフショアを活用したある中規模案件のプロジェクトの不具合一覧をみると、結合テスト／システムテストでないと見つからない不具合ばかりです。
- 結合テスト／システムテストで発生した「単体テスト不足」の不具合を削減しようと考えていましたが、その魂胆は早くも挫折しました。
- 新しい施策を実施するよりも、そのプロジェクトの秘密を他のプロジェクトに展開する方がよいのではないかと考えました。



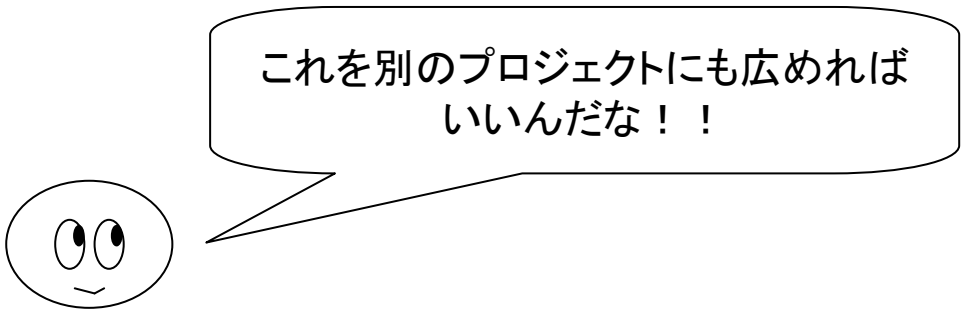
オフショア開発の品質は良い？

- あるプロジェクトの取り組みをヒアリングすると
  - プロジェクトメンバに聞いたところ、次のような取り組みをしていました。

テストケースチェックリストの整備

テスト技法勉強会の実施

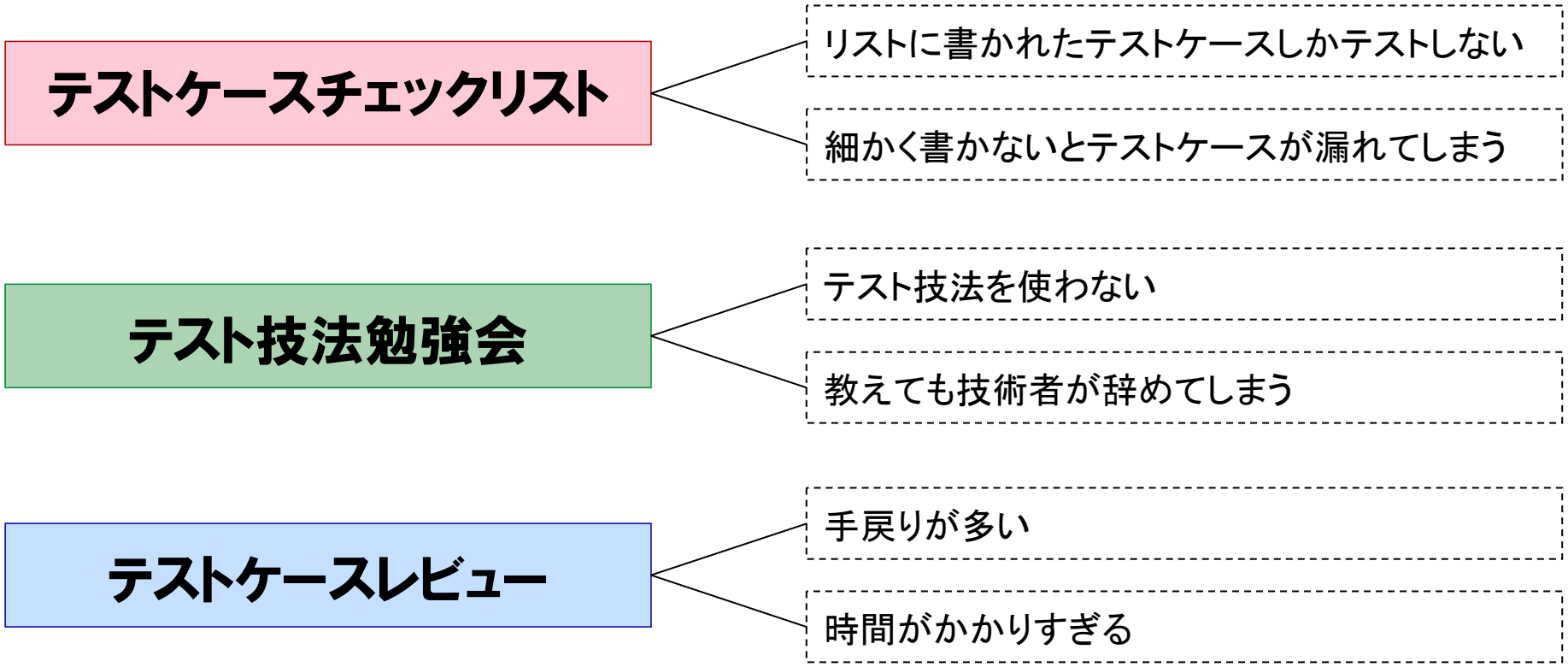
テストケースレビューの実施



何が問題なのか？

■ さらにプロジェクトのヒアリングを進めると

- 良いことばかりではなく、いろいろ問題もあるようです。



何が問題なのか？

■ オフショア企業にヒアリングすると

- 発注部門とは異なった問題認識をしていました。

テストケースチェックリスト

プロジェクト用のリスト、案件用のリスト、自社のリストというようにチェックリストがたくさんありどれを使えばよいのか分からない。

テスト技法勉強会

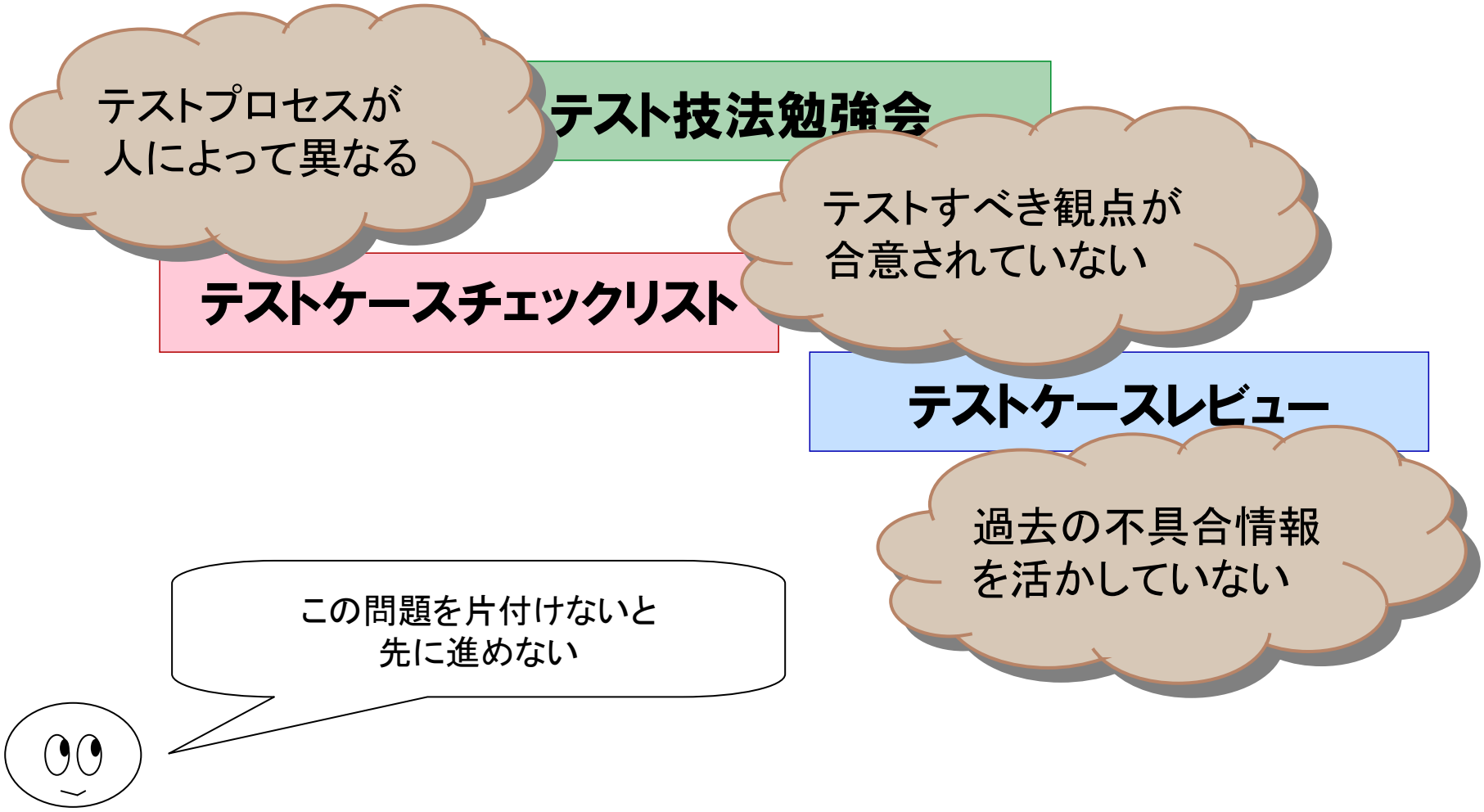
簡単なテスト技法しか知らないので品質が上がらない。不具合の検出率が高い技法を教え欲しい。

テストケースレビュー

あらかじめ言ってくれればよいのだが、レビューのときに、初めて聞くことが多すぎる。

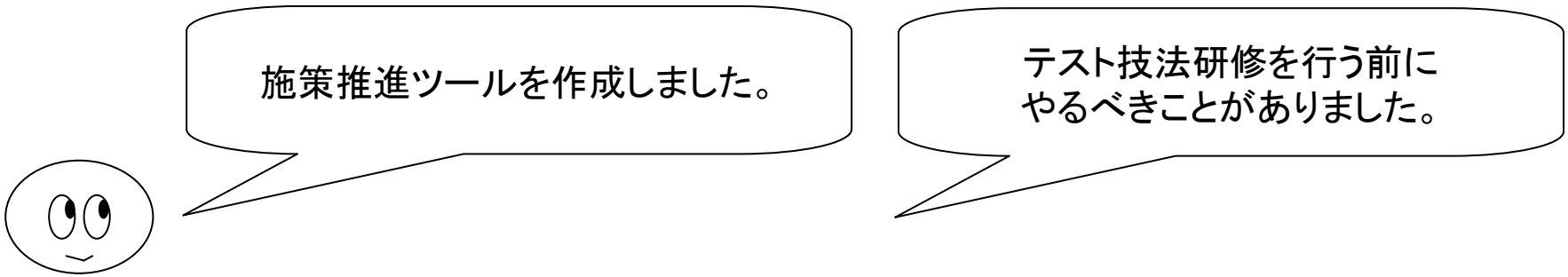
何が問題なのか？

- さらにオフショア企業のヒアリングを進めると
  - いろいろなことが分かり始めました。

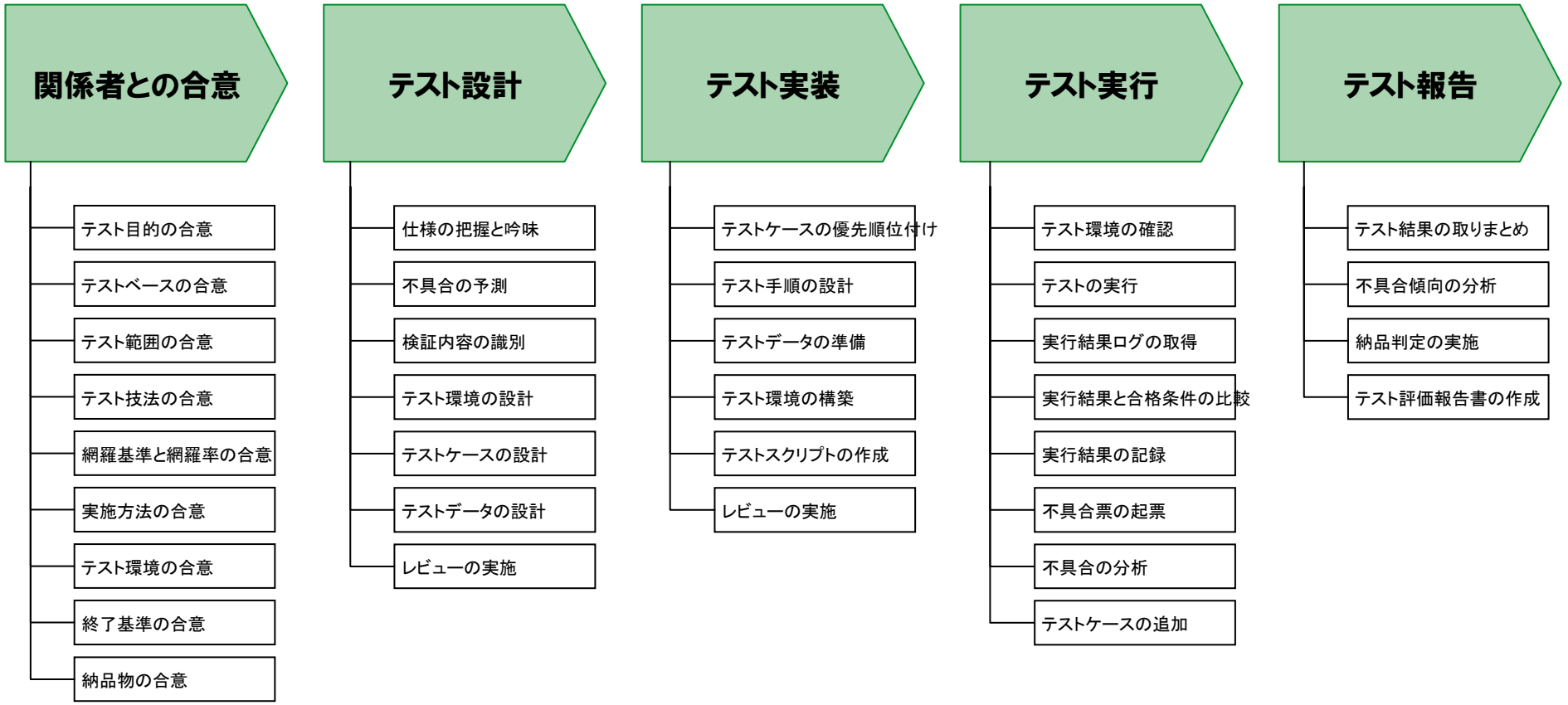


何が問題なのか？

隠れていた問題	施策
テストプロセスが貧弱	**テストプロセス（WBS）**
テスト観点が漏れる	**テスト観点ツリー＆リスト**
過去の不具合を活かしていない	**不具合推測リスト**



【三点セット】オフショア版ソフトウェアテストプロセスWBS



【三点セット】テストプロセスWBSの特徴

- **「関係者との合意」を重視しています。**
  - オフショア開発では事前の合意が大切です。
  - 関係者との合意を重視しています。

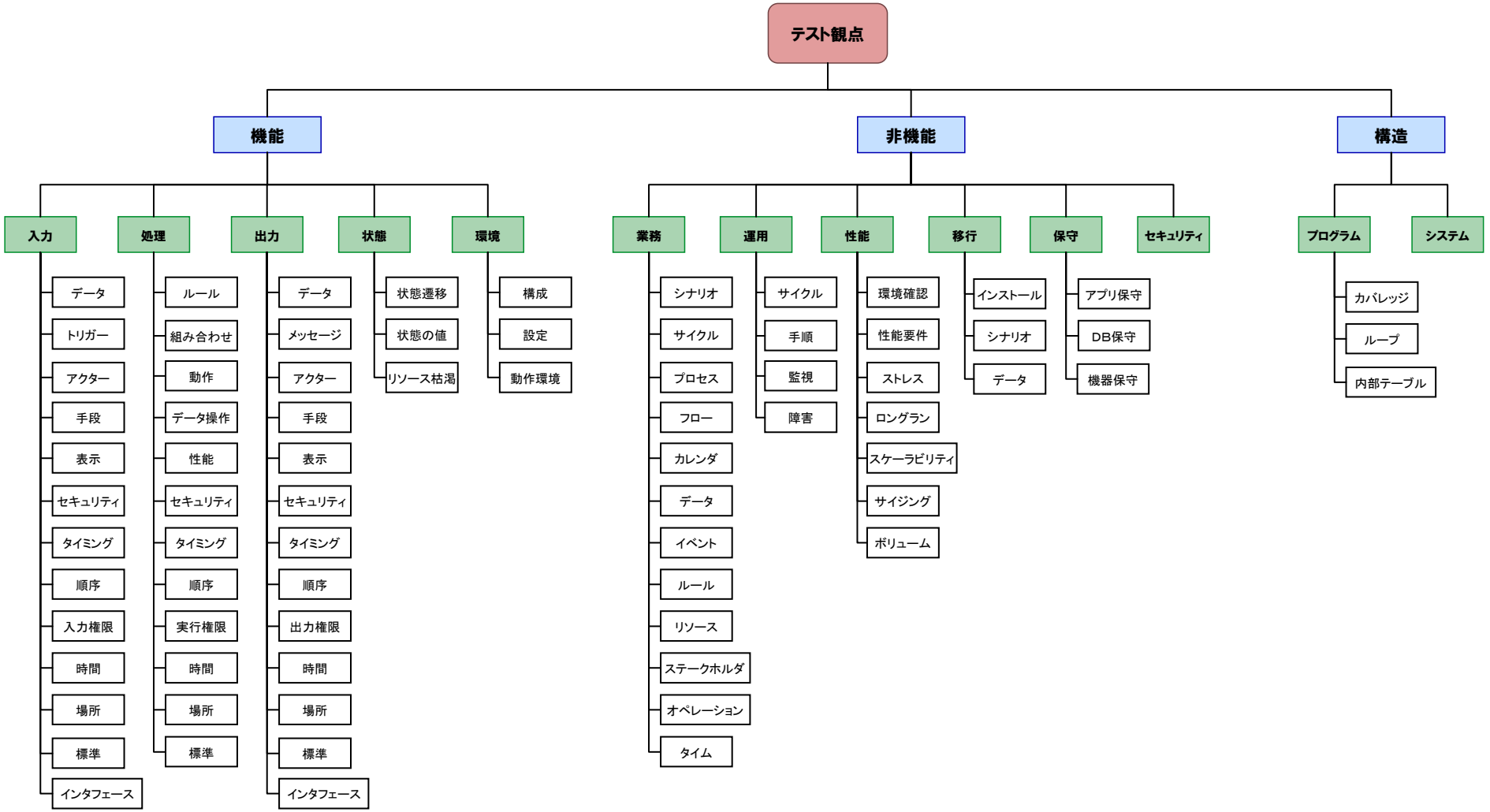
- **「テスト設計」を明示的に入れています。**
  - よいテストケースを作成するためには、テストにも「設計」が必要です。
  - 設計に必要なタスクを漏れなく記述しています。

- **テスト観点の使用や不具合推測を取り入れています。**
  - テスト観点を用いたテスト設計のやり方や、不具合推測を使ったテスト設計をタスクとして入れています。
  - 他のツールと整合性がとれています。

【三点セット】テストプロセスWBSの例

ID	内容	説明	備考
1000	関係者との合意		
1100	テスト目的の合意	テストの目的を合意する。	
1200	テストベースの合意	テスト設計／実装で使用するテストベースの一覧を作成し、合意する。	テストベースとは仕様書や設計書のこと。
1300	テスト範囲の合意	テストを実施する範囲を合意する。	
1310	対象機能の列挙と合意	テスト対象となる機能を列挙し、合意する。	
1320	立ち位置の合意	テスト観点を挙げる立ち位置を合意する。	
1330	テスト観点の列挙と合意	テストベースを基にテスト観点を洗い出し、合意する。	テスト観点の広さについて合意する。
1340	テスト観点の粒度の合意	テスト観点の粒度をどこまで細かくするかを合意する。	テスト観点の深さについて合意する。
1350	想定不具合の吟味	過去の不具合傾向やシステム特性などから想定不具合を吟味する。	
1360	テスト観点の追加	想定欠陥から導き出されたテスト観点を追加する。	
1370	機能毎のテスト観点の合意	機能毎のテスト観点を合意する。	
1400	テスト技法の合意	使用するテスト技法について合意する。	
1500	網羅基準と網羅率の合意	網羅基準と網羅率について合意する。	
1600	実施方法の合意	テストの実施方法について合意する。	
1700	テスト環境の合意	テストを実施する環境について合意する。	
1800	終了基準の合意	テスト終了の基準を合意する。	
1900	納品物の合意	テスト関連の納品物について合意する。	
1910	テストケース記述方法の合意	テストケースの記述方法について合意する。	
1920	不具合票記述方法の合意	不具合票の記述方法について合意する。	

【三点セット】テスト観点ツリー



【三点セット】テスト観点ツリーの特徴

- **過去のテストケースを分析し、必要なテスト観点を網羅しています。**
  - すべてのテスト観点ではありませんが、必要な観点を挙げています。
  - 大きなテスト観点漏れが無くなります。

- **テストケースレビューの時間が短縮できます。**
  - テスト観点ベースで互いにレビューを実施することで、
    テストの意図が分かり、レビュー時間が短縮できます。

- **他プロジェクトの「オフショアテストガイド／チェックリスト」の評価ができます。**
  - 他プロジェクトで活用されているチェックリストをそのまま適用してもよいのか、カスタマイズすべきなのかの判断ができるようになります。

【三点セット】テスト観点リストの例

機能 ― 入力 ― データ ― 入力フィールド

必須項目／入力禁止項目	
	必須入力項目に対して、未入力時の動作を確認しているか。
	必須入力項目だけで、登録が可能であることを確認しているか。
	入力禁止項目に対して、入力できないようになっているか
	NULL禁止項目にNULLが入力された場合、NULLチェックが行われているか。
	通常項目に対して、未入力時の動作を確認しているか。
日付フィールド	
	うるう年を考慮しているか。
	うるう日を考慮しているか。
	存在しない日に対してエラー処理を考慮しているか。
	存在しない日に対してエラー処理を考慮しているか。
	業務上の特殊日に対して考慮しているか。
	春分／秋分の日やハッピーマンデーなど移動する祝祭日に対して考慮しているか。
	元号の切り替わる前後を考慮しているか。
	日付の入力スタイルを考慮しているか。
	和暦を考慮しているか。
	月末日の入力・更新が行えることを確認しているか。
	特殊日の入力・更新が行えることを確認しているか。

【三点セット】テスト観点リストの特徴

- **テスト観点の具体例である観点リストを用意しています。**
  - テスト観点は抽象化しているため、そのままテストケースを作成できない場合もあります。
  - そこで、具体例であるテスト観点リストを用意することで、テストケースをイメージしやすくしています。

- **８９０個のテスト観点リストを使うことで、合意形成が容易になります。**
  - どこまでテストをして欲しいのかを、テストケース記述の前にオフショア企業に伝えるのは難しい作業です。
  - 具体例を使うことで、テストケースの粒度を伝えることができます。

【三点セット】不具合推測リストの例

金額	
	値の精度
		値の精度が画面によって異なる。
	マイナス表示
		マイナス金額の表示方法が画面によって異なる。
	桁区切り
		金額の桁区切りが画面によって異なる。
	桁数
		カンマ無し9桁を想定していたが、カンマ有り9桁で実装されていた。
		最大桁数で入力した後、フォーカスアウトでカンマ付き表示したら桁落ちが発生した。
	画面間の金額
		画面間で表示される金額があっていない。
	データ型
		金額を文字列型として持っていたにも関わらず、数値として扱ってしまった。
	最大金額
		9桁（9億）以下であることを想定していたが、実際には10桁（10億）を超えるデータが入金された。

【三点セット】不具合推測リストの特徴

- **ピンポイント型のテストケースを作成するために、不具合推測リストを用意しています。**
  - テストケースの種類に網羅型とピンポイント型（検出型）があります。
  - ピンポイント型は経験豊富な人が作成したり、不具合データベースに基づいて作成したりします。
  - この不具合推測リストを用いることにより、ある種の不具合を予測することができます。

- **７４５個のリストを読むことで、不具合を実装しなくなります。**
  - このリストは「エラー推測」というテストを行うために用いるものです。
  - しかし、あらかじめ埋め込まれやすい不具合を知っておければ、多くの場合、プログラム実装段階で避けるようになります。
  - そのため、テストは単なる確認行為になり、デバッグ工数が削減できるようになります。

あるプロジェクトの進め方

■ 現状をヒアリングします。

- １～２時間程度を想定しています。
- リーダまたはブリッジＳＥの方を想定しています。

■ 導入研修を実施します。

- 最短で、講義２時間、アンケート記入１０分程度です。
- メンバー全員に直接教えるのか、ブリッジＳＥの方に教えてそれから展開するのかは、お任せします。
- 実際のプロジェクト文書を用いて演習を実施する場合、２時間以上掛かります。

■ プロジェクトで仕様するテスト観点と不具合推測を選択します。

- ４～８時間程度を想定しています。プロジェクトの規模が大きい場合、それ以上の時間がかかることもあります。
- 仕様を知っている人に入ってもらいます。
- 担当者の教育を考慮する場合、担当者からテスト観点と潜在不具合を挙げてもらい、それをチェックするという過程を踏むので、想定以上の工数がかかることもあります。

■ 導入後の定例ミーティングを実施します。

- １時間程度です。１～２週間に１回程度です。