Empowered by Innovation NEC

使える「テスト設計テンプレート」を目指して！

－”話題沸騰ポット“での検証と改善で「使える度」UP －

NEC SWQCテスト技術コミュニティ

下前　真浩（日本電気通信システム株式会社）

根間　才治（株式会社NEC情報システムズ）

人と地球にやさしい情報社会を
イノベーションで実現する
グローバルリーディングカンパニー

NECグループビジョン2017

目次

修

1. SWQCテスト技術コミュニティの紹介
2. JaSST'09 Tokyoにて
   ー テスト設計テンプレートの説明 ー
3. テンプレートの再検証
4. テンプレートの検証結果
5. 改善施策
6. 効果
7. 今後の展開・課題

予稿集との差分の表記方法

修：修正スライド

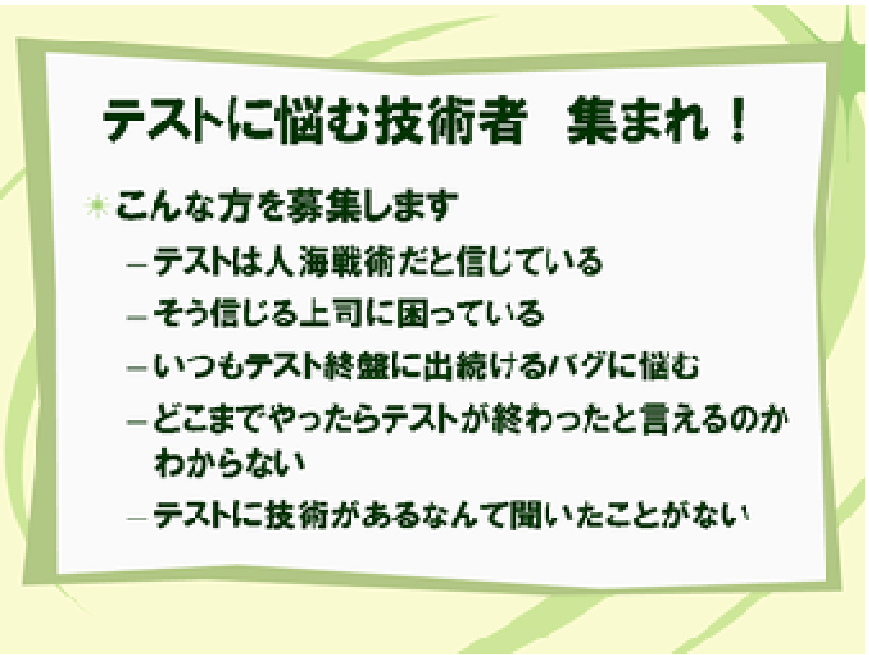
追：追加スライド

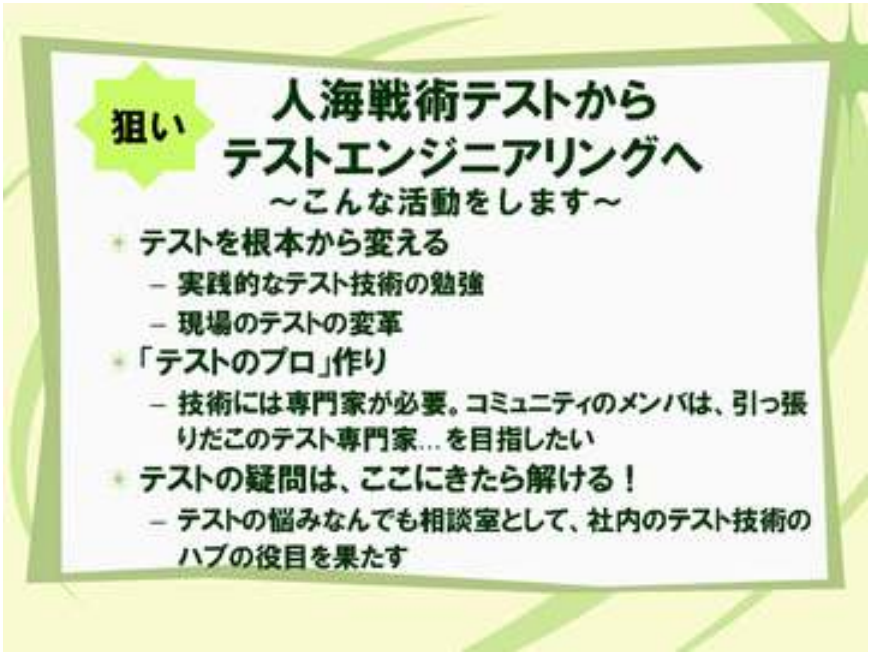
無印：変更なし

SWQCテスト技術コミュニティの紹介

NECグループのSWQC活動として、テーマ別コミュニティ活動中！

テスト技術コミュニティは、特にソフトウェアテストに興味を持ったメンバーで構成されています（総勢120名超）





SWQC活動：NECにおけるソフトウェアの総合的品質管理活動（1981年に開始）
現在は組織を超えてSW技術者が切磋琢磨する場として活動中

追

昨年のおさらい

JaSST’09 Tokyoにて（1）

テスト設計テンプレートとは

＜課題＞

◎当たり前レベルのテスト設計もれ防止
◎テスト設計からテスト項目へのリンク付け
◎設計者によるテスト設計内容のバラツキ軽減

＜解決策＞

テスト観点を網羅した「テスト設計テンプレート」を考案
- 機能・非機能テストの2つに分け、テスト観点をリストアップ
  ※機能テスト：機能仕様を分析して実施するテスト
  ※非機能テスト：機能に関係しない特性のテスト

JaSST’09 Tokyoにて（2）

追

JaSST’09 Tokyoで提案したテスト設計テンプレート

AAコンポーネント

非機能テストの１０のテスト観点

- ○操作性
- ○運用テスト
- ○性能テスト
- ○信頼性テスト
- ○セキュリティテスト
- ○障害対応テスト
- ○インストーラビリティテスト
- ○移植性テスト
- ○保守性テスト
- ○機器構成テスト

機能テストの５つのテスト観点

- ○I/F
- ○境界値
- ○エラー処理
- ○計算ミス・誤差
- ○競合・タイミング

JaSST’09 Tokyoにて（3）

修

JaSST’09 Tokyoで提案したテスト設計テンプレート

AAコンポーネント

機能テスト テスト観点	実施の有無 ○/×	詳細な観点	テストの概要 入力条件 テストの実行に必要な入力（値、テーブル名、ファイル名など）を指定	環境 テストを遂行する上で必要な環境（HW、SWなど）を指定	特別な要求 前提条件、事後条件、その他特別な手続き等	依存関係 テストケースとテストの実行順序	ID	備考
機能 仕様に示された機能に関して、過不足がないか、機能の誤りがないか、を確認するテストを行う。 機能確認すべき項目をリストアップすること。 このとき、マニュアルがあればそれも参照すること。 また、主要な機能については、繰り返してテストする必要があるかも検討すること。	○	(1)電源を供給したらタイマの表示が0になっていること (2)タイマボタンの操作で任意のタイマ値に設定できること (3)タイマが始動すること (4)タイマが計時して残り時間を表示すること (5)タイマが計時を満了して停止すること			(1)電源の供給が試験時にはされていない (4)蓋を開けても計時は継続すること			
I／F 各種インタフェースが正しいことを確認するテストを行う。 関数、クラス間の呼び出し先、戻り先が正しいこと、 受け渡しするデータの形式、数量、内容がただしいことなどをテストする。 テストすべきインタフェースをリストアップすること。 テストすべきインタフェースとして、関数間、ルーチン間、クラス間の呼び出し関係、その場合の受け渡しデータや、タスク/プロセス間の呼び出し関係、受け渡しデータなどがあり、更に、複数に共用されるデータ、ファイルなどがある場合もある。 特に、共用データ類については、その確保・初期化・廃棄・解放の関係を確認することが重要です。 また、再入性、再帰性の確認や、エラー時のインタフェース、割込み時のインタフェースについても確認が必要なことがあります。	○	(1)タイマボタンを（タイマ値更新に必要な時間）押下したら タイマカウンタが更新されて表示される。 (2)蓋の開閉とタイマ動作について (3)タイマボタンが押下されてブザーが鳴ること(50ms) (4)タイマをリセットをしたらタイマ停止のブザーが鳴ること(100ms) (5)タイマ満了時にブザーが鳴ること(100ms×3回) ※インタフェースする相手を明確にしてから書くほうが良い 蓋・タイマ・ブザー・ディスプレイ			ブザー音の結果は、トレーニングした試験者の経験によって判断する			
境界値 値などに有効な範囲がある場合には、その上限・下限と、これを externally 外れた		(1)タイマ設定値が60分を超えたら1分に戻ること (2)一回タイマボタンを押下したらタイマ残り時間が1分						(3)タイマの計時は、即時には不明、一分後にタイマ値のカウントダウンで判断する
エラー処理								
計算								

ブザー音の結果は、トレーニングした試験者の経験によって判断する

I／F
各種インタフェースが正しいことを確認するテストを行う。
関数、クラス間の呼び出し先、戻り先が正しいこと、
受け渡しするデータの形式、数量、内容がただしいことなどをテストする。

（1）タイマボタンを（タイマ値更新に必要な時間）押下したら
タイマカウンタが更新されて表示される。
（2）蓋の開閉とタイマ動作について

◎JaSST’09 Tokyoテスト設計テンプレートの課題

- **簡単な試行のみで、検証が不十分**
- **特に機能テストテンプレートの使い勝手が悪い**

追

今年の活動

テンプレートの再検証

修

＜目的＞

(1)テンプレートの使い勝手やテスト観点の問題点発見
(2)現場導入を想定し、導入時の問題点発見

＜検証方法＞

・SESSAMEの“話題沸騰ポット“を題材に使用
・実際の現場の使い方を想定して3チームで検証
　Aチーム：テンプレートのみ使用
　Bチーム：マインドマップとテンプレートの併用
　Cチーム：ユースケースとテンプレートの併用

＜Aチームと比較した時の結果の特徴＞

Bチーム：最も多くの視点の「詳細な観点」が挙げられた
　→視点の分解をマインドマップが促してくれた
Cチーム：操作，動作時の条件を変えた「詳細な観点」が多かった
　→ユースケースを元にしているためと思われる

テンプレートの検証結果

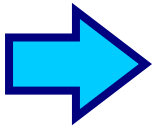
修

＜テンプレートの効果＞

- 各項目を埋めようと一所懸命考えるため、モレが減った
- テスト対象外にしたことが、結果として残った
- 既存ツールとの併用が可能であることが分かった

＜テンプレートの問題点＞

- 「テスト観点」の説明が曖昧なため、表を書くときに迷いがあった
  →計算精度にも境界値にも入るような「詳細な観点」があった
  　例：ボタンを100ms押下したらブザーがなる
  →I/Fの「詳細な観点」に、”人間とのI/F” と”システム間のI/F”の2つの捉え方があった
- 「テスト観点」の説明が不足していたため、表に埋められない「詳細な観点」があった
  →起動時の環境条件の「詳細な観点」を書ける「テスト観点」がない？



まだ、モレとバラツキがあった

改善施策

修

＜テンプレートの問題点＞

- 「テスト観点」の説明が曖昧なため、表への記載時に迷いが発生
- 「テスト観点」の説明が不足していたため、表に埋められない「詳細な観点」があった

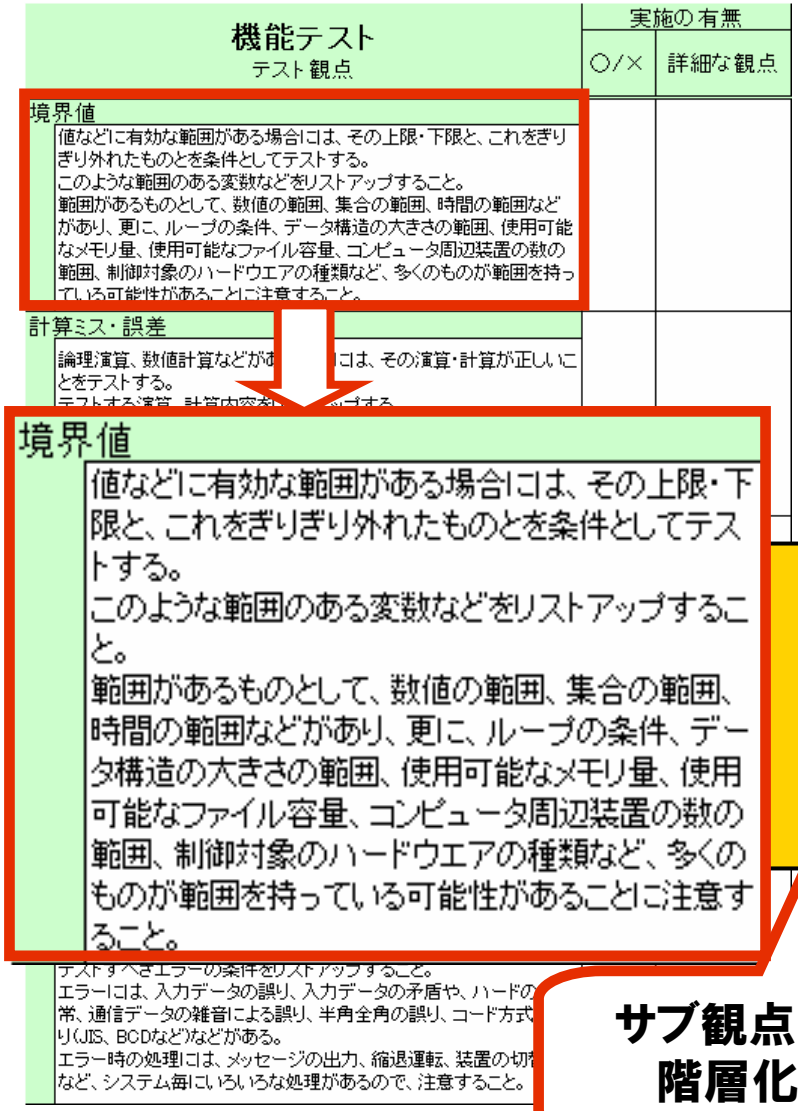
＜改善施策＞

1. 「テスト設計テンプレート」の見直し
   - テスト観点の階層化，再整理
   - ヒントワードの追加
2. 「テスト設計テンプレート ガイド」の作成

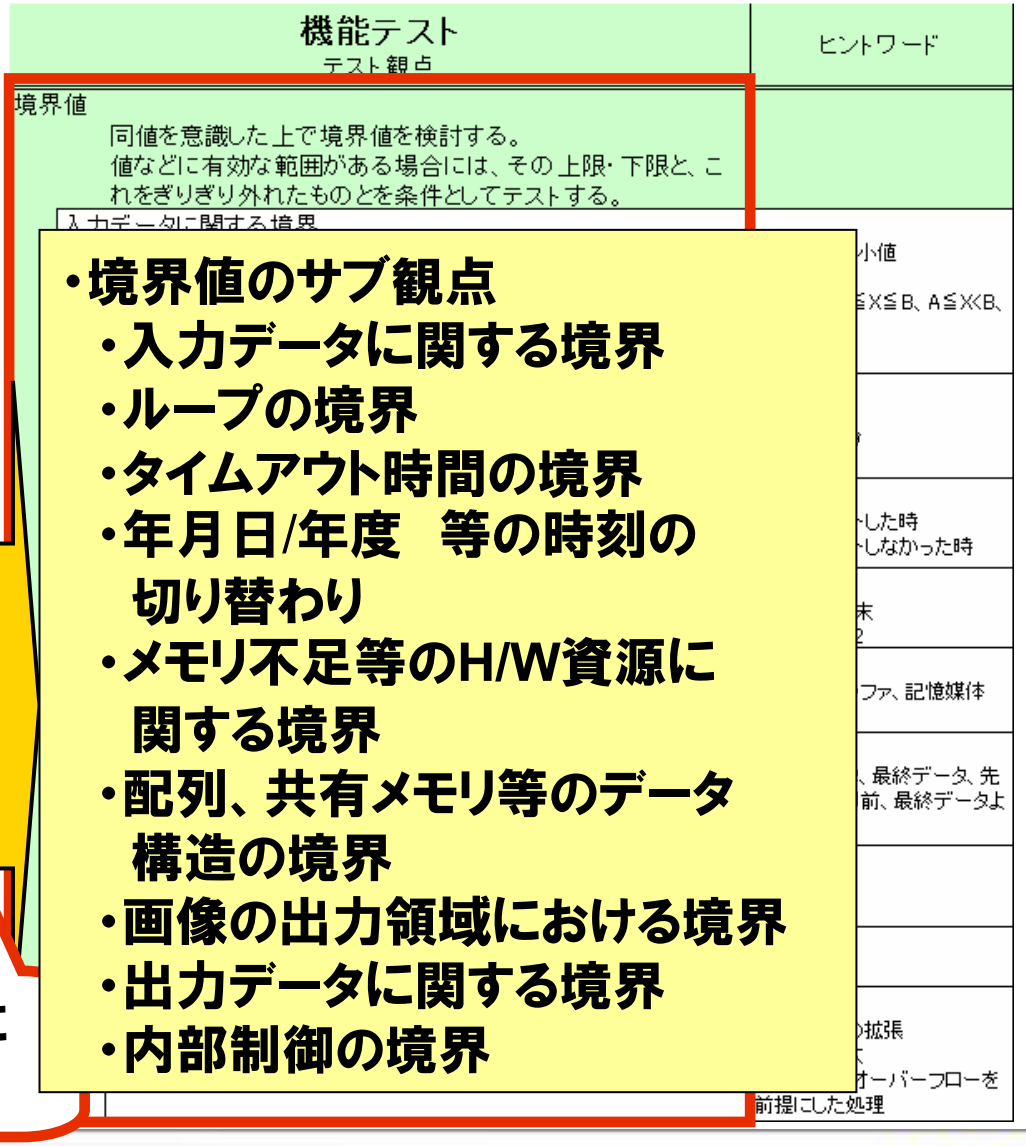
施策1：テスト設計テンプレートの改善

修

旧テスト設計テンプレート



新テスト設計テンプレート



- 境界値のサブ観点
  - 入力データに関する境界
  - ループの境界
  - タイムアウト時間の境界
  - 年月日/年度　等の時刻の切り替わり
  - メモリ不足等のH/W資源に関する境界
  - 配列、共有メモリ等のデータ構造の境界
  - 画像の出力領域における境界
  - 出力データに関する境界
  - 内部制御の境界

サブ観点に階層化

施策１：テスト設計テンプレートの改善

修

旧テスト設計テンプレート

機能テスト テスト観点	実施の有無 ○/×	 詳細な観点
境界値 値などに有効な範囲がある場合には、その上限・下限と、これをぎりぎり外れたものとを条件としてテストする。 このような範囲のある変数などをリストアップすること。 範囲があるものとして、数値の範囲、集合の範囲、時間の範囲などがあり、更に、ループの条件、データ構造の大きさの範囲、使用可能なメモリ量、使用可能なファイル容量、コンピュータ周辺装置の数の範囲、制御対象のハードウエアの種類など、多くのものが範囲を持っている可能性があることに注意すること。		
計算ミス・誤差 論理演算、数値計算などがある場合には、その演算・計算が正しいことをテストする。 テストする演算、計算内容をリストアップする。 論理演算のミス、数値計算のミスに注目すると共に、数値計算の場合には計算誤差、オーバフロー、アンダフローにも注意すること。 また、直接的な計算ではないが、スタック領域がオーバフローしないかについても必要に応じてテストすること。		
I／F 各種インタフェースが正しいことを確認するテストを行う。		
エラー処理 エラーの検出、エラー時の処理が正しいことをテストする。また、入力誤りを検出して、以降のエラー発生を予防することができているかもテストする。 テストすべきエラーの条件をリストアップすること。 エラーには、入力データの誤り、入力データの矛盾や、ハードの異常、通信データの雑音による誤り、半角全角の誤り、コード方式の誤り(JIS、BCDなど)などがある。 エラー時の処理には、メッセージの出力、縮退運転、装置の切替えなど、システム毎にいろいろな処理があるので、注意すること。		

新テスト設

代表的な
キーワードを記述

	ヒントワード
境界値 同値を意識した上で境界値を検討する。 値などに有効な範囲がある場合には、その上限・下限と、これをぎりぎり外れたものとを条件としてテストする。	
入力データに関する境界 入力されたデータが想定される範囲内の場合と範囲外の場合の処理が正しく行われていることを確認する。 ・数値の取りうる範囲の内外 ・集合の内外 ・データ長の範囲の内外	・最大値、最小値 ・範囲内外 ・A<X<B、A≦X≦B、A≦X<B、A<X≦B
ループの境界 ・回数のループ ・条件のループ	・0回ループ ・無限ループ ・再帰
タイムアウト時間の境界 一定時間以上、次のイベントが起こらない場合の処理をタイムアウトという。	・タイムアウトした時 ・タイムアウトしなかった時
画像入力処理等において 所が画面の範囲外になるような条件になった場合の処理を確認する。	
出力データに関する境界 出力結果が変わる条件をテストする。	
内部制御の境界 外から見えない境界(プログラム内部のメモリやバッファの境界等)により処理が変わる場合を確認する。	・内部領域の拡張 ・桁数の拡大 ・カウンタのオーバーフローを前提にした処理

内部制御の境界
外から見えない境界(プログラム内部のメモリやバッファの境界等)により処理が変わる場合を確認する。

内部制御のヒントワード
- **内部領域の拡張**
- **桁数の拡大**
- **カウンタのオーバーフローを前提にした処理**

新テスト設計テンプレートの使用例と効果

修

話題沸騰ポット ： タイマー機能

機能テスト テスト観点	ヒントワード	実施の有無（○・×）	
境界値 同値を意識した上で境界値を検討する。 値などに有効な範囲がある場合には、その上限・下限と、これをぎりぎり外れたものとを条件としてテストする。			
入力データに関する境界 入力されたデータが想定される範囲内の場合と範囲外の場合の処理が正しく行われていることを確認する。 ・数値の取りうる範囲の内外 ・集合の内外 ・データ長の範囲の内外	・最大値、最小値 ・範囲内外 ・A<X<B、A≦X≦B、A≦X<B、A<X≦B	○	ボタン押下回数によるタイマセット時刻（上限・下限）
ループの境界 ・回数のループ ・条件のループ	・0回ループ ・無限ループ ・再帰	○	タイマ値が60分 タイマカウンタが
タイムアウト時間の境界 一定時間以上、次のイベントが起こらない場合の処理をタイムアウトという。	・タイムアウトした時 ・タイムアウトしなかった時		
年月日、日時、分、秒、期、年度　等の時刻の切り替わり 分替り、時替り、日替り、月替り、年替り、年度替りといった時間の切り替わりの処理が正しいことを確認する。	・閏年 ・期末 ・月末 ・4/2		
メモリ不足等のハードウェア資源に関する境界 メモリ、バッファ等のハードウェア資源が不足した場合の処理を確認する。	・メモリ、バッファ、記憶媒体 ・縮退処理		
配列、共有メモリ等のデータ構造の境界 配列、共有メモリ等のデータ構造体に対して、正しく処理できることをテストする。	・先頭データ、最終データ、先頭データより前、最終データより後ろ		

観点の抽出
モレがなくなった

設計者による
バラツキが
なくなった

境界値

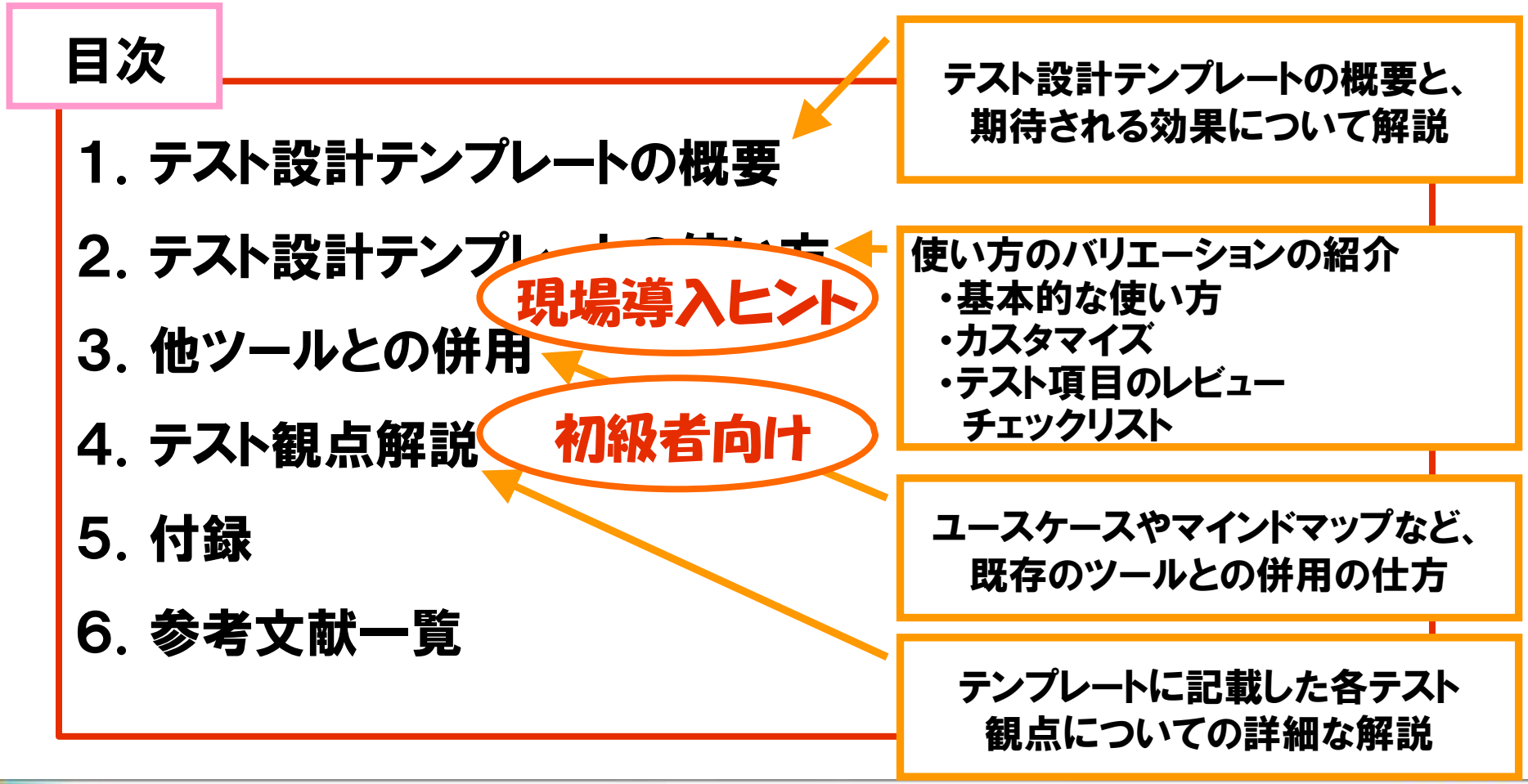
値などに有効な範囲がある場合には、その上限・下限と、これをぎりぎり外れたものとを条件としてテストする。
このような範囲のある変数などをリストアップすること。
範囲があるものとして、数値の範囲、集合の範囲、時間の範囲などがあり、更に、ループの条件、データ構造の大きさの範囲、使用可能なメモリ量、使用可能なファイル容量、コンピュータ周辺装置の数の範囲、制御対象のハードウエアの種類など、多くのものが範囲を持っている可能性があることに注意すること。

旧テスト観点

施策2：テスト設計テンプレートガイドの作成

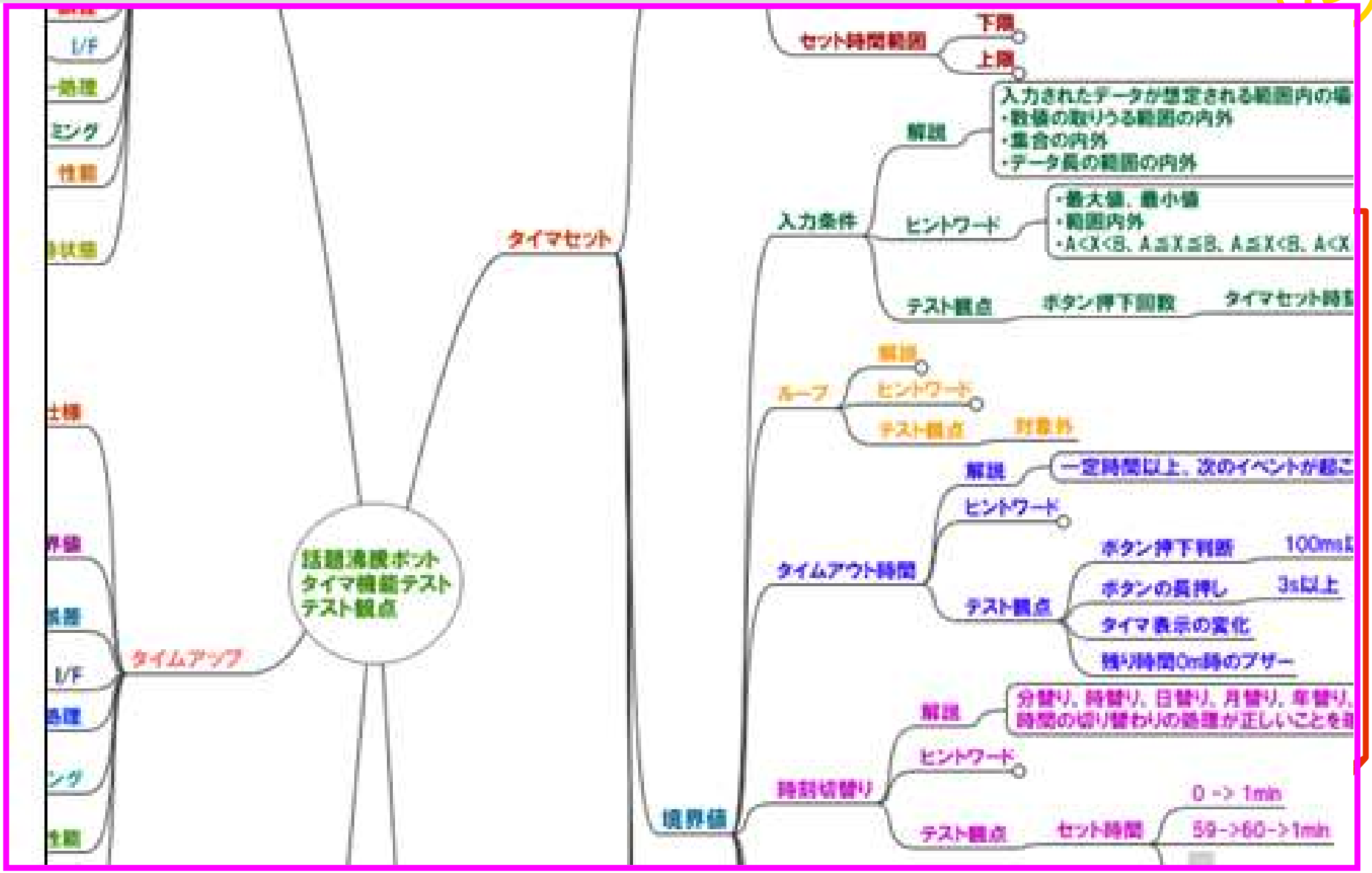
修

テンプレートの具体的な使用方法をはじめ、
テスト設計に関する基礎知識についても記載



施策2：テスト設計テンプレートガイド記述例①

修



施策2：テスト設計テンプレートガイド記述例②

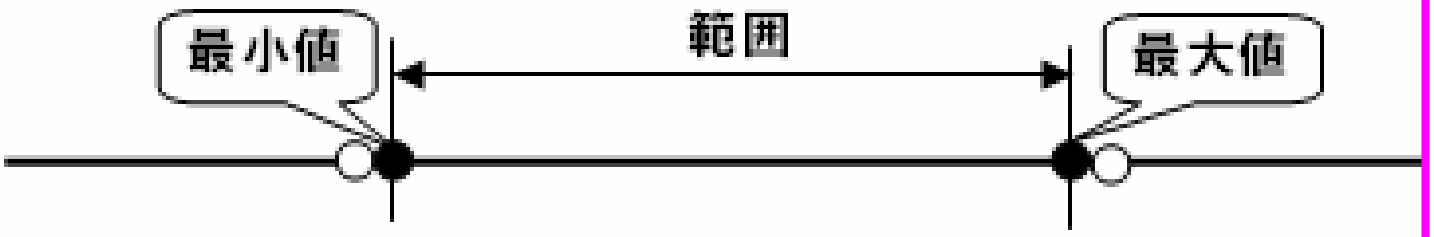
修

テスト観点解説

テストデータは少なくとも、範囲の最小値、最大値、中間値、最小値より少し大きい値を用意する。

①数値の取りうる範囲の内外

入力されるデータが数値の場合、採りうる範囲内、範囲外の数値での動



効果

修

「テスト観点」を洗い出す
スピードが約4倍に！
（現場の感想）

- 使いやすくなった
  - →利用者が“迷わずに”検討できるようになった
- モレ・バラツキが更に減った
- 現場に導入しやすくなった
  - →チェックリストとしての適用が可能
  - →マインドマップやユースケースなど従来手法と併用が可能

機能テンプレートの
「テスト観点（6個）」を
サブ観点35個に分割し、
具体化

今後の展開・課題

- NECグループ内への
  「テスト設計テンプレート」の普及
- テスト項目の気づきを与えるためのヒント集、
  例示集の作成

ご静聴
ありがとうございました

参考文献

修

1. 基本から学ぶソフトウェアテスト　テストのプロ」を目指す人のために
Cem Kaner, Jack Falk, Hung Quoc Nguyen著　テスト技術者交流会訳
日経BP社 ISBN978-4-8222-8113-7
2. ソフトウェアテスト技法　自動化、品質保証、そしてバグの未然防止のために
ボーリス・バイザー著　小野間彰，山浦恒央訳
日経BP出版センター ISBN978-4-8227-1001-9
3. はじめて学ぶソフトウェアのテスト技法
リー・コープランド著　宗雅彦訳
日経BP社 ISBN978-4-8222-8251-6
4. 現場の仕事がバリバリ進む ソフトウェアテスト手法
高橋寿一、湯本剛著
技術評論社 ISBN978-4774127118
5. 「インターネットアプリケーションのためのソフトウェアテスト」
Hung Q. Nguyen,Michael Hackett,Bob Johnson著 松村淳ら訳
ソフトバンククリエイティブ　ISBN978-4797322064
6. ソフトウェアテスト標準用語集（日本語版）Version 1.1 (2006 年12 月22 日)
国際ソフトウェアテスト資格認定委員会用語集作業班
（http://www.jstqb.jp/dl/JSTQB-glossary.V1.1.pdf）
7. ベリサーブの「システムカテゴリ」http://www.veriserve.co.jp/r_and_d/
8. ソフトウェア品質知識体系ガイドーSQuBOK Guide
オーム社　ISBN978-4274501623
9. IT用語辞典(e-Words) http://e-words.jp/
10. ソフトウェアエンジニアリング基礎知識体系ーSWEBOK　松本吉弘訳
オーム社　ISBN978-4274946769
11. 要求工学 第23回：非機能要求　山本修一郎著
ビジネスコミュニケーション社 月刊ビジネスコミュニケーション 2006年9月号
12. 特定非営利活動法人組込みソフトウェア管理者・技術者育成研究会(SESSAME)
話題沸騰ポット(GOMA-1015)要求仕様書　第7版
13. マインドマップから始めるソフトウェアテスト
池田　暁、鈴木　三紀夫 著

Empowered by Innovation

これ以降の資料は、質疑応答用。

テスト設計テンプレートとガイドの利用例

追

新テスト設計テンプレート：非機能テスト

追

非機能テスト テスト観点	ヒントワード
操作性 ソフトウェア製品が、指定された条件下で理解され、学びやすく、使用しやすく、ユーザにとって魅力的である能力を判定するためのテスト。	ISO 13407(インタラクティブシステムのための人間中心設計プロセス) ISO/TR 16982:2002（人間とシステムのインタラクション－人間中心設計のためのユーザビリティ評価手法）
操作のしやすさ	
ユーザ視点で、使い勝手に問題がないか、覚えやすいかどうか(習得性)を検証するテスト。	・画面のボタンの位置 ・操作の順番が「上から下」「左から右」
操作の一貫性	
ソフトウェアの操作(メニュー、キーボード入力、他)に一貫性があるかを検証するテスト。国際標準や業界標準に準拠しているか、バージョンアップ時に操作性が変わらないか、などについても考慮が必要。	・カーソルの移動手段「リターン」「タブ」「マウス」「タッチパネル」などの可否の混在
アクセシビリティテスト	
ソフトウェアを使用することが想定される身体的な制約を持つ人を含む全てのユーザ(子供や高齢者、障がい者、など)が、どの程度容易にコンポーネントやシステムを利用できるか判定するテスト。	・ダブルクリックの代替 ・連続押下扱い ・同時複数キー押下の扱い ・字の大きさの自由設定
運用テスト 操作環境の中で、コンポーネントやシステムを評価するテスト。	・定常／非定常時の業務の流れ ・日業務／月業務／半期／年
シナリオテスト	
業務運用にそったシナリオを作成して実行するテスト。	オペレータ、日常業務
非通常運用テスト	
計画停電によるシステム停止や年末年始などの特別運用、臨時スケジュール変更の対応を確認するテスト。	停電、連休、年末年始、夏期休暇
運用サイクルテスト	
システムタイムチャート図の運用サイクル(日次／月次など)が時間通りに運用できること、実時間での業務フロー運用に問題がないこと、業務の排他関係を適切に実装していることを確認するテスト。	日次、週次、月次処理 業務繁忙期
性能テスト システム能力を確認するテスト。確認する能力としてレスポンス等の性能面を運用状態・負荷状態の条件での性能をテストする。	・長時間運用 ・同時刻に処理が集中 ・H/Wの選択
メモリ	
運用状態や負荷状態でのメモリ使用量・使用効率を計測し要求性能が満たされているかテストする。	ガベージ、上限、チューニング
CPU	
運用状態や負荷状態でのCPU使用率量・使用効率を計測し要求性能が満たされているかテストする。	マルチコア、並列処理、マルチスレッド
レスポンス	
運用状態や負荷状態でのコマンド要求等に対する応答やデータベース等の応答性能が要求性能が満たされているかテストする。	遅延、運用状態、業務ピーク(日、時間、期間)
スループット	
運用状態や負荷状態での通信回線の単位時間での転送効率が要求機能が満たされているかテストする(補：単位時間あたりのCPUの処理能力を示すこともある。CPU参照)。	業務効率、通信環境、システム環境、業務ピーク