

テスト設計コンテスト チュートリアル ちびこん編 2026



2026/05/08

テスト設計コンテストU30クラス審査委員
神田 繁良



自己紹介

◆本日の講演者

神田 繁良 (テスト設計コンテストU-30クラス 審査委員)

◆テスト設計コンテスト(テスコン)との関わり

2012 : テスコン初参戦(Q魂) : 選手<一般>

2017 : OPENクラス準優勝(わんだーず♪) : 代表<企業>

2024 : U30クラス優勝(きつつき1.0) : オブザーバー<企業>

2025,2026 : テスコンNOW企画推進 : JaSST東京実行委員

テスコンファンの一人です♪



テスト設計コンテストについて

◆テスト設計コンテスト U-30クラス

指定のテストベースに対するテスト設計を行ない、その優劣を競います
U-30クラスは30歳以下の方のみ参戦可能です

◆テスト対象 U-30クラス

割り勘支援アプリ Warikan

割り勘を支援するためのスマートフォンアプリケーションである。
割り勘の割合、人数、支払合計金額を入力することで集金金額を計算できる。
また外部のサービスとの連携で送金処理もサポートする。



このチュートリアルについて

◆ゴール

テスコンにおけるテスト開発プロセスの具体的なイメージを持っていただく
テスト要求分析からテスト実装までの一通りの流れをお伝えします

◆想定聴講者

- ・テスト設計の勉強をしたいと思っている方
- ・業務においてテスト設計に携わっている方
- ・テスコンに興味を持っていただいている方

◆使用する用語

用語は説明している場合を除き、基本的に本資料作成時点の
JSTQB/ISTQB の用語定義に従います



もくじ

第1章 テスコンにおけるテスト開発プロセス

1. テスト開発プロセス
2. テスト要求分析
3. テストアーキテクチャ設計
4. テスト詳細設計
5. テスト実装

40分

20分

第2章 昨年度の審査コメント分析

1. U-30クラスで評価されるテスト設計

5分



第1章

テスコンにおけるテスト開発プロセス



テスト開発プロセス



テスト要求の
獲得と整理/
テスト要求
モデリング

テスト
アーキテクチャ
モデリング

テスト技法の
適用による
テストケースの
列挙

手動/自動化
テストスクリプト
(テスト手順)の
記述



テスト開発プロセス

昔の
テスト開発
プロセス



JSTQBの
テスト開発
プロセス



本講での
テスト開発
プロセス



テスト要求の
獲得と整理/
テスト要求
モデリング

テスト
アーキテクチャ
モデリング

テスト技法の
適用による
テストケースの
列挙

手動/自動化
テストスクリプト
(テスト手順)の
記述

引用元：テスト設計チュートリアル ちびこん編'21 資料 P-67

テスト要求分析



テスト要求の
獲得と整理/
テスト要求
モデリング

テスト
アーキテクチャ
モデリング

テスト技法の
適用による
テストケースの
列挙

手動/自動化
テストスクリプト
(テスト手順)の
記述



テスト要求とは

◆テスト要求

ステークホルダーがテストに求めていること≡テスト活動に求められていること
要求は曖昧なことが多いため、「どのような目的で」「どこまで」「何を」テストするか？を
明確にし、ステークホルダーと合意形成を図りましょう

テスト要求は「エンジニアリング的テスト要求」と「マネジメント的テスト要求」の2種類があります

エンジニアリング的テスト要求

- ・システム要求
- ・ソフトウェア要求
- ・機能要求
- ・非機能要求
- ・利用シーン
- ・新しく開発するところ
- ・変更したところ
- ・ミスしそうなところ
- ・バグが多そうなところ
- ...

テスト観点、テストアーキテクチャ設計、
テストケースに反映

マネジメント的テスト要求

- ・体制
- ・スキル
- ・作業場所
- ・作業工数
- ・納期
- ・組織戦略
- ・プロジェクト指標
- ・機材利用可否
- ・ツール利用可否
- ...

テストアーキテクチャ設計に反映

テスト要求分析

テスト要求分析	テスコンU-30クラス	チュートリアル
定義	テスト要求の獲得と整理 テスト要求モデリング	テスト要求を仕様書や補足書などから獲得しよう テスト要求を構造化や図式化しながらテスト観点を導き出そう
INPUT テストベース	割り勘支援アプリ Warikan仕様書 テストプロジェクト要求補足書	仕様書、補足書以外にも・・・ テスト要求の源泉を準備し、テスト要求分析しましょう
OUTPUT 想定している成果物	テスト観点の一覧 テスト観点=テストすべきこと 例:テスト条件・テストパラメータやそれを抽象化したもの	テスト観点以外にも・・・ テスト目的、テストスコープ、テスト対象なども整理し、次工程に繋げよう

テスト要求の源泉準備

入手できる限り、テスト要求の源泉を準備しましょう

プロダクト情報

プロダクト要求仕様
機能、非機能、…
プロダクト企画
プロダクト目的、ベースモデル、…
…

ドメイン情報(知識)

業界知見
法規、規格・標準、…
類似製品
UI、動作、不具合情報、口コミ、…
…

プロジェクト情報

プロジェクト計画
プロジェクト目的、指標、QCD制約
体制、スケジュール、予算、…
マスターテスト計画
テストレベル、テストタイプ、…

テスト情報(知識)

ISTQB/JSTQB
テスト用語、テストプロセス、…
ISO25010
品質特性、…
…

※注意：テスコンにおいて源泉準備はテスト要求分析の審査基準ではありません！
源泉を活用し、テスト観点抽出に繋げてください
源泉準備に時間をかけすぎるとテスト実装までたどり着きません

テスト目的を理解しよう

◆実際のテスト活動

プロダクト目的、プロジェクトの方針、テスト組織のテストポリシーなどから、対象プロジェクトのテスト目的を導出していきます

◆テストコンU-30クラス

テストプロジェクト要求補足書に「テスト目的」が指定されています！！

- ・アプリケーションが用途を満たしていることを確認する。
- ・アプリケーションがリリースできる品質レベルであることを確認する。
- ・テストエンジニアの観点を活かし、テストの活動を通して、テストベースへ改善のフィードバック(曖昧さ、記述不足、矛盾に対する指摘や改善提案)を行なう。

◆審査基準:テスト妥当性

指定されたテストの目的に対して成果物は妥当か？

妥当性を示すためにも、テスト目的の具体化を行ないましょう！

テスト目的を理解しよう

◆テストコンU-30クラス

テストプロジェクト要求補足書に「テストへの要求」が指定されています！！

- ・高頻度の仕様変更、リリースに対応する。仕様変更への対応しやすさや、テストの繰り返し再実施のやりやすさを確保できるように、テスト成果物を作成すること。
- ・将来的なチーム内でのテスト担当のローテーションや引継ぎのため、テスト実行者が変わってもテストの再現性をある程度確保できるようにテスト実装を行うこと。
- ・アジリティ重視の開発のため、テスト活動の工数はなるべく小さく実施できるようにする。テストの責務分析に基づく最適化(例:コンポーネントテスト、統合テストとの重複部分の削減)や探索的テストのアプローチやテストプロセス全体へのAI活用導入といった、アジリティ確保の施策の活用を推奨する。

◆審査基準:テスト妥当性

指定されたテストの制約(工数、リソース、テストレベル、テストタイプなど)を満たしているか？

テスト要求から、テストの制約(QCDや責務範囲)を明確にしましょう

テストスコープを理解しよう

◆実際のテスト活動

マスターテスト計画書を確認し自組織・自プロジェクトでのテストレベルやテストタイプの定義を確認しましょう。なければ、マスターテスト計画書を作成し、定義しましょう

◆テストコンU-30クラス

テストプロジェクト要求補足要求書にテストレベルが指定されています(一部要約、省略)

・コンポーネントテスト

Class、Widgetごとに開発者判断でC0カバレッジ100%網羅のテストを作成実行。
CIでシステムテスト開始時は全合格状態を確保する

・統合テスト

結合したアプリケーションに対し、エミュレータで起動できること、一通りの画面遷移ができることを確認。CIでシステムテスト開始時は全合格状態を確保する

・システムテスト

リリース版アプリケーションに対し、実機を用いて手動で実施するテスト

このテスト設計コンテストでは上記のうちシステムテストを担当する。

**担当であるシステムテストの
責務を具体化しましょう**

テストタイプの定義は必要！

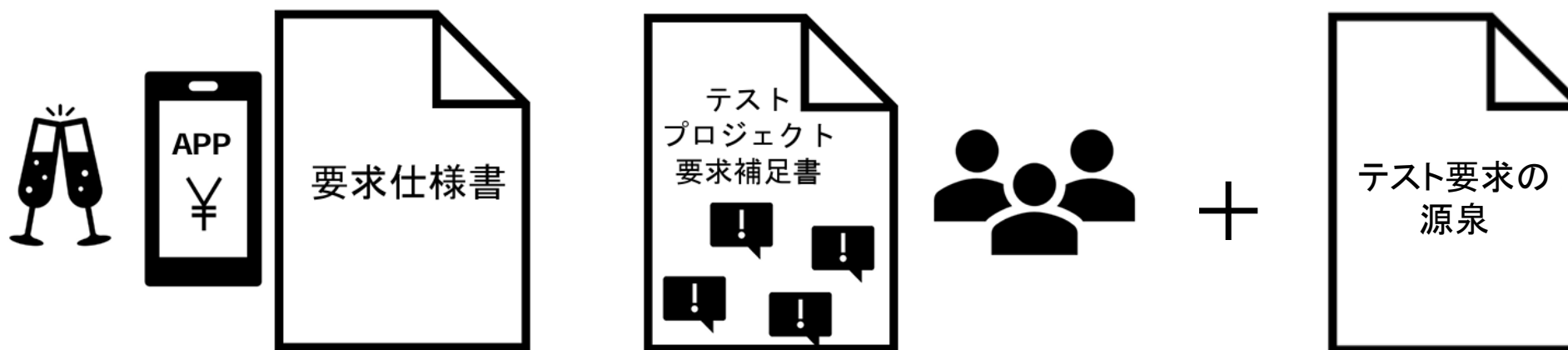
テスト対象を理解しよう

◆実際のテスト活動

要求仕様書や開発文書などのドキュメントや、実際のテスト対象の動作、ステークホルダーから得た情報などから、テスト対象のシステム構成、機能、使われ方、弱点などを分析します

◆テスコンU-30クラス

Warikan 要求仕様書とテストプロジェクト要求補足要求書(+テスト要求の源泉)から、テスト対象を分析・理解していきましょう



Warikanはどんなシステム構成？ どんな機能を提供する？
誰がいつどのように使う？ 類似製品はどんな動きか？

テスト観点を抽出してみよう

◆実際のテスト活動

テスト目的、テストスコープを踏まえ、テストすべきこと(テスト観点)を考え抽出します

◆テストコンU-30クラス

テスト対象の機能、品質、ユースケースなどをモデル化(構造化、図式化)し、テスト観点を可視化します

過去のテスト設計コンテストチュートリアルでモデル化の具体例が多数掲載されています
是非参考にし実践してみてください

◆審査基準:テスト要求分析

テスト要求分析でテストすべきこと(テスト観点)が考慮されているか

テスト観点がすべて盛り込まれているか

テスト観点が適切な粒度で表現されているか など

要求仕様からテスト観点を考える

次の文章を読んで、必要なテストケースをあげてみましょう

- このプログラムは入力ダイアログから3つの整数を読む
- この3つの値は、それぞれ三角形の三辺の長さを表すものとする
- プログラムは、三角形が不等辺三角形、二等辺三角形、正三角形のうちのどれであることを示すメッセージを表示する



引用元：Glenford J. Myers著「ソフトウェア・テストの技法第2版」P-2

要求仕様からテスト観点を考える

1. 有効な不等辺三角形
2. 有効な正三角形
3. 有効な二等辺三角形
4. 有効な二等辺三角形の際の
三種類の辺の組み合わせ
5. 一つの辺がゼロ
6. 一つの辺が負の値
7. 二辺の和がもう一边と等しい
8. 二辺の和がもう一边と等しい際の
三種類の辺の組み合わせ
9. 二辺の和がもう一边より小さい
10. 二辺の和がもう一边より小さい際の
三種類の辺の組み合わせ
11. 三辺がゼロ
12. 整数でない辺
13. 辺の数が三つ以外
14. 各ケースに、入力値に対する出力値を
示している

※ これで網羅できているのか？
※ 他にないのか？



引用元：Glenford J. Myers著「ソフトウェア・テストの技法第2版」P-3を基に要約

構造化

Before

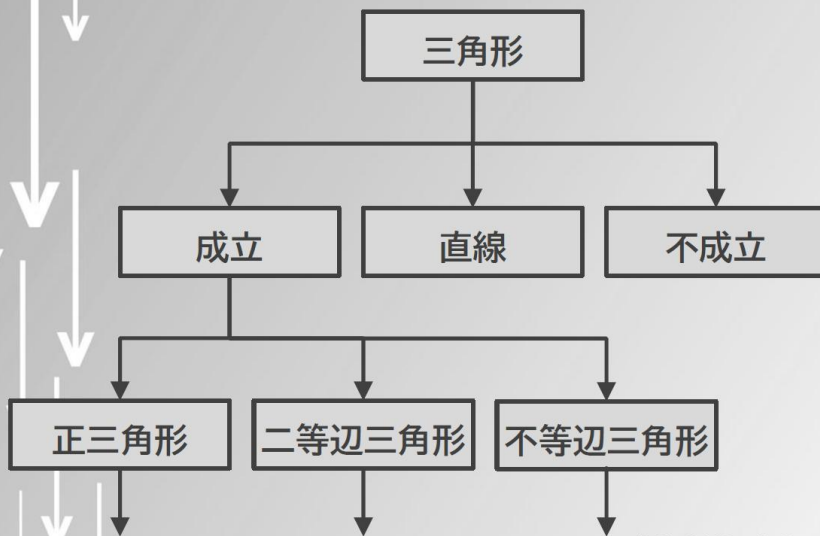
1. 有効な不等辺三角形
2. 有効な正三角形
3. 有効な二等辺三角形
4. 有効な二等辺三角形の際の三種類の辺の組み合わせ
5. 一つの辺がゼロ
6. 一つの辺が負の値
7. 二辺の和がもう一辺と等しい
8. 二辺の和がもう一辺と等しい際の三種類の辺の組み合わせ
9. 二辺の和がもう一辺より小さい
10. 二辺の和がもう一辺より小さい際の三種類の辺の組み合わせ
11. 三辺がゼロ
12. 整数でない辺
13. 辺の数が三つ以外
14. 各ケースに期待結果を示している

After

1. 三角形の形に関するテスト
 - 三角形が成立する場合のテスト
 - 有効な不等辺三角形
 - 有効な二等辺三角形
 - 有効な正三角形
 - 三角形が成立しない場合のテスト
 - 直線になってしまう場合
 - 二辺の和がもう一辺と同じ
 - 面を構成できない場合
 - 二辺の和がもう一辺より短い
 - 長さが0の辺がある
2. 辺の組み合わせに関するテスト
 - 三種類の辺の組み合わせ
3. 入力の仕様に関するテスト
 - 整数でない辺
 - 辺の数が三つ以外
4. 期待結果を示してあるかどうか

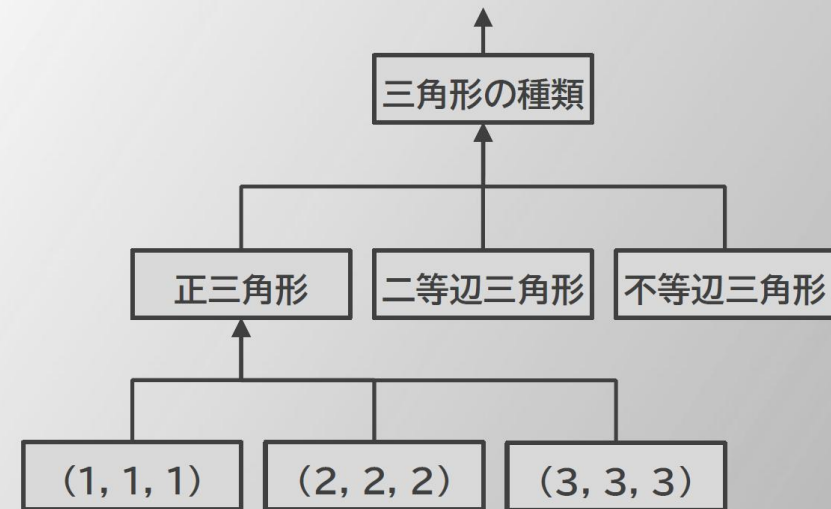
構造化

トップダウン的に観点を整理する方法



© 2016-2021 NISHI Yasuharu / YAMASAKI Takashi 14

ボトムアップ的に観点を整理していく方法



© 2016-2021 NISHI Yasuharu / YAMASAKI Takashi 13

実際にはトップダウンとボトムアップを循環させながらテスト観点を網羅していく

引用元：テスト設計チュートリアル ちびこん編'21 資料 P-13,14,15

要求仕様からテスト観点を考える

画面遷移について明らかにするために、分解・分類、構造化を試みる。

「三角形判定アプリ」にはTOP画面、入力画面、判定結果画面、設定画面がある。TOP画面で開始ボタンを押すと入力画面に遷移する。入力画面では、辺の長さとして値を入力できる。また、判定ボタンを押すと、判定結果画面に遷移する。判定結果画面では、判定結果を表示する。3辺の長さが入力済みであり、すべて有効な値の場合は、その値に応じて三角形の種類を表示する。判定結果画面には、もう一度ボタンがあり、押すと入力画面に遷移する。それぞれの値は有効な値だが三角形が成立しない場合、判定結果画面には、三角形不成立エラーのメッセージを表示する。入力された値のうち1つ以上が有効な値でない場合は無効値エラーのメッセージを、未入力がある場合は未入力エラーのメッセージを表示する。TOP画面以外の画面には戻るボタンがあり、押すと一つ前の画面に戻る。ただし、判定結果画面の戻るボタンを2秒以上押した場合は、一つ前の画面ではなく、TOP画面に戻る。

画面

イベント

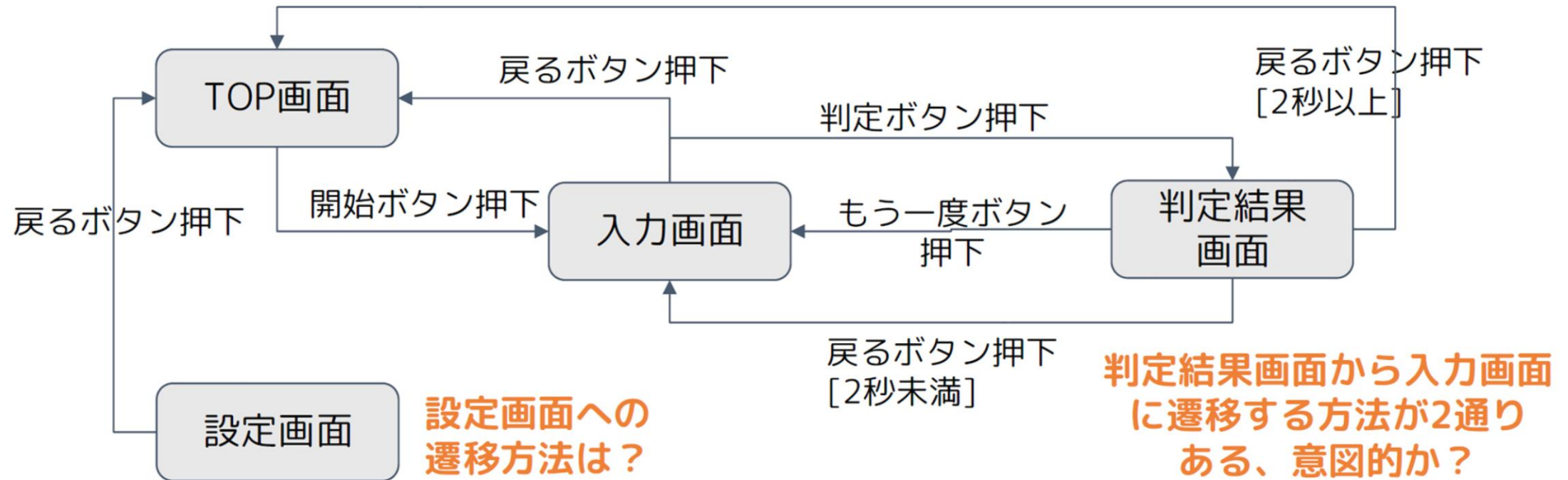
条件

遷移元や遷移先の画面・遷移を引き起こすイベント・条件に分類する。

「三角形判定アプリ」にはTOP画面、入力画面、判定結果画面、設定画面がある。TOP画面で開始ボタンを押すと入力画面に遷移する。入力画面では、辺の長さとして値を入力できる。また、判定ボタンを押すと、判定結果画面に遷移する。判定結果画面では、判定結果を表示する。3辺の長さが入力済みであり、すべて有効な値の場合は、その値に応じて三角形の種類を表示する。判定結果画面には、もう一度ボタンがあり、押すと入力画面に遷移する。それぞれの値は有効な値だが三角形が成立しない場合、判定結果画面には、三角形不成立エラーのメッセージを表示する。入力された値のうち1つ以上が有効な値でない場合は無効値エラーのメッセージを、未入力がある場合は未入力エラーのメッセージを表示する。TOP画面以外の画面には戻るボタンがあり、押すと一つ前の画面に戻る。ただし、判定結果画面の戻るボタンを2秒以上押した場合は、一つ前の画面ではなく、TOP画面に戻る。

図式化

結果を図や表で表現すると、曖昧な点や疑問が見えやすくなる。
必要に応じて更に情報収集し、テストへの要求を理解する。



なお、この例では、更に状態遷移表のように表で表現しても効果的である

テストアーキテクチャ設計



テスト要求の
獲得と整理/
テスト要求
モデリング

テスト
アーキテクチャ
モデリング

テスト技法の
適用による
テストケースの
列挙

手動/自動化
テストスクリプト
(テスト手順)の
記述



テストアーキテクチャ設計

テスト要求分析	テストコンU-30クラス	チュートリアル
定義	テストアーキテクチャモデリング	テスト観点をモデリングしテストの全体像を表現する 複数のテスト観点をまとめ、テストケースの構造を定義する
INPUT	テスト要求分析の成果物	テスト要求分析の成果物 テスト目的、テストスコープ テスト対象、テスト観点 テストアーキテクチャスタイル
OUTPUT 想定している成果物	テスト観点の全体像をなにがしかの関心事に沿ってモデル化したもの	テスト全体像 テストケースの構造

テストアーキテクチャ設計とは

◆テストアーキテクチャ設計

テスト要求分析では、テスト観点「テストすべきこと」を導出しました

また「どのような目的で」「どこまで」「何を」テストするか？も導出しました

テストアーキテクチャ設計では「どのように」テストするか？を明確にしていきます

テストの構成要素とその関係性(前後関係、並列関係、依存関係、包含関係など)で表現します

◆テストの全体像

同じまとまりとしてテストすべき「テスト観点」をまとめ、テストの全体像を整理しましょう

テストの制約を守りながらテスト目的を達成するために最適なテスト全体像を描きましょう

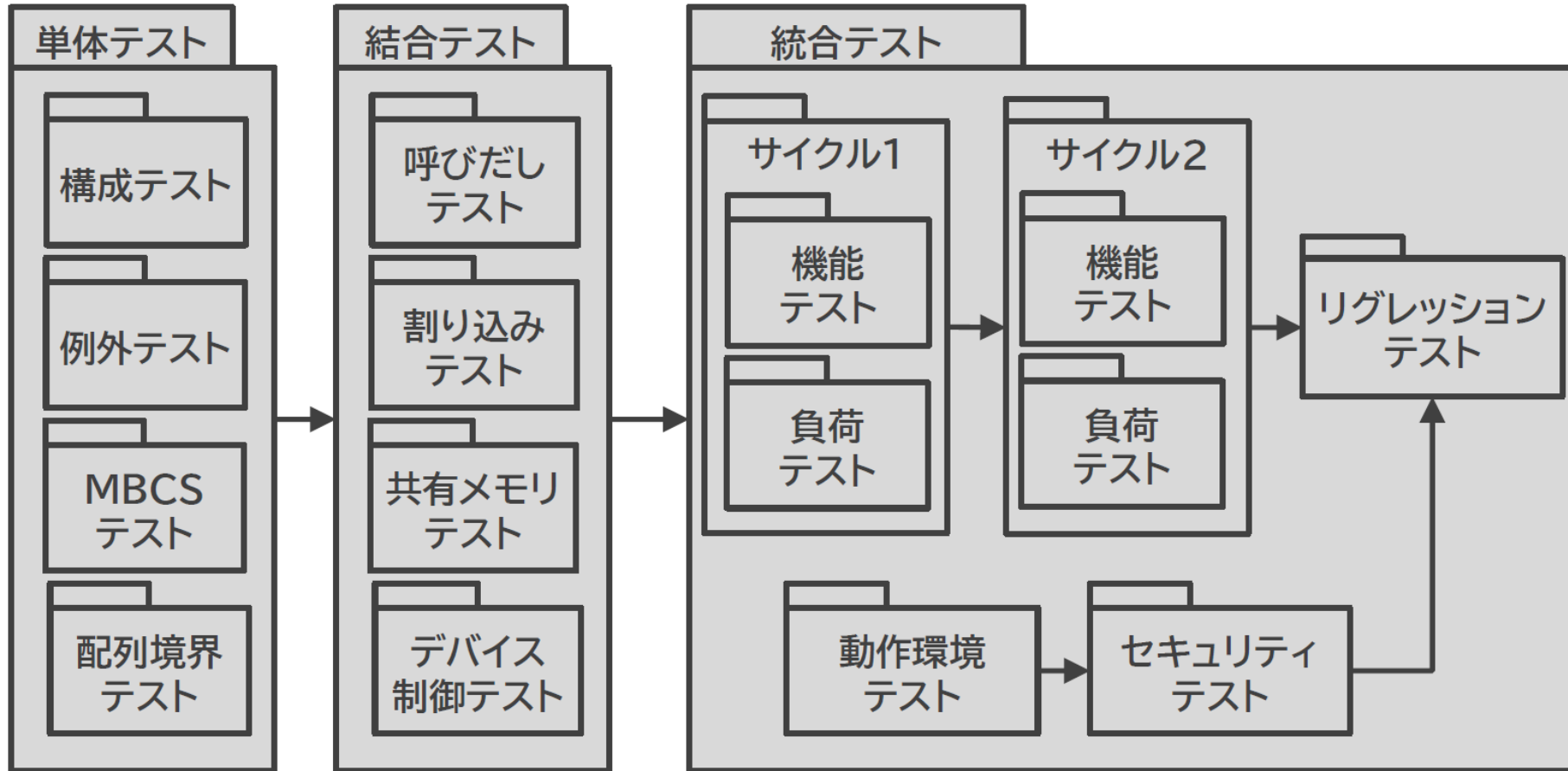
◆テストケースの構造

次工程であるテスト詳細設計をしやすいくするために、複数の「テスト観点」をまとめた

テストケースの構造を定義しましょう

テストの全体像を示してみよう

テストレベルごとにテストタイプや実行回数を表示したテストの全体像 例

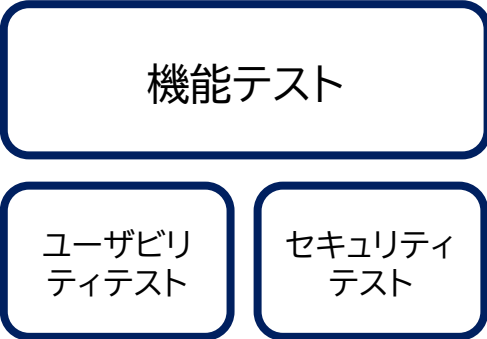
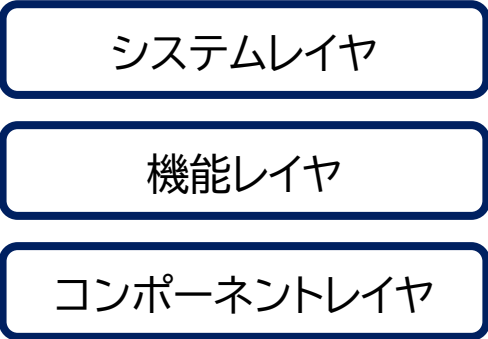



テストアーキテクチャスタイル

ゼロからテストアーキテクチャを作るのは大変です

まずは、テスト全体を俯瞰するためによく使われるテストアーキテクチャスタイル(≡考え方、指針)を基に考えてみましょう。

◆代表的なテストアーキテクチャスタイル

テストタイプ型	レイヤー型	フィルター型	...
			複合型 ビュー型 ...

テストアーキテクチャスタイル:テストタイプ型

◆考え方、指針

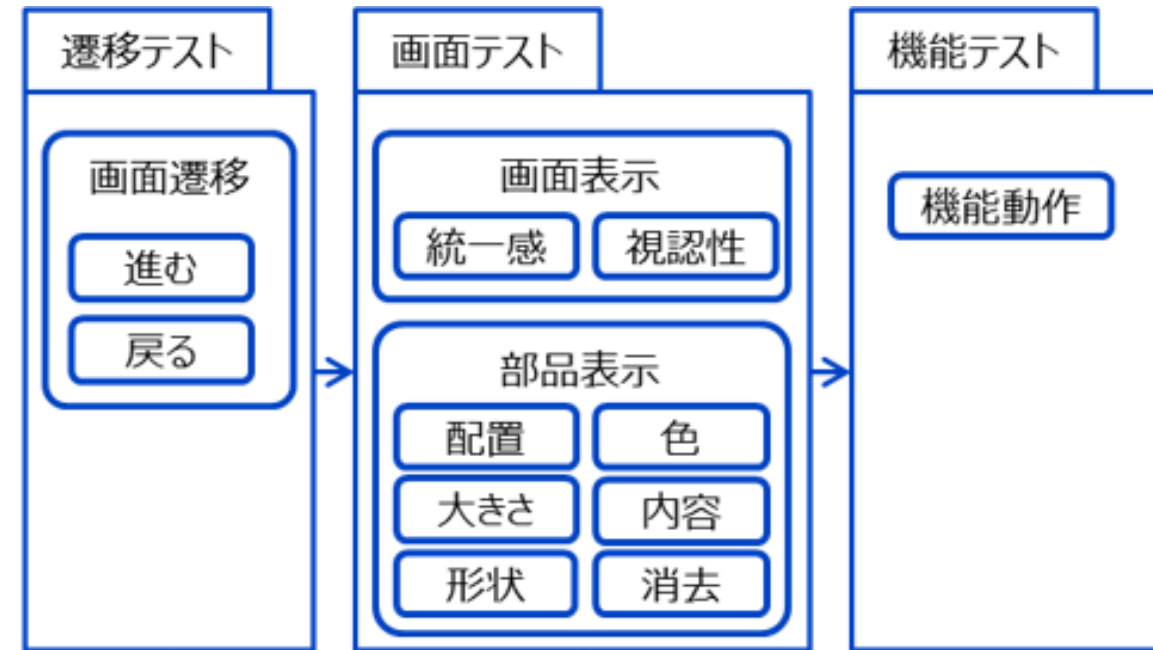
どんなテストを実施するか?から考える

◆適用方法

組織のテストタイプにテスト観点を割り当てる

or

類似のテスト観点を集め、テストタイプにまとめる



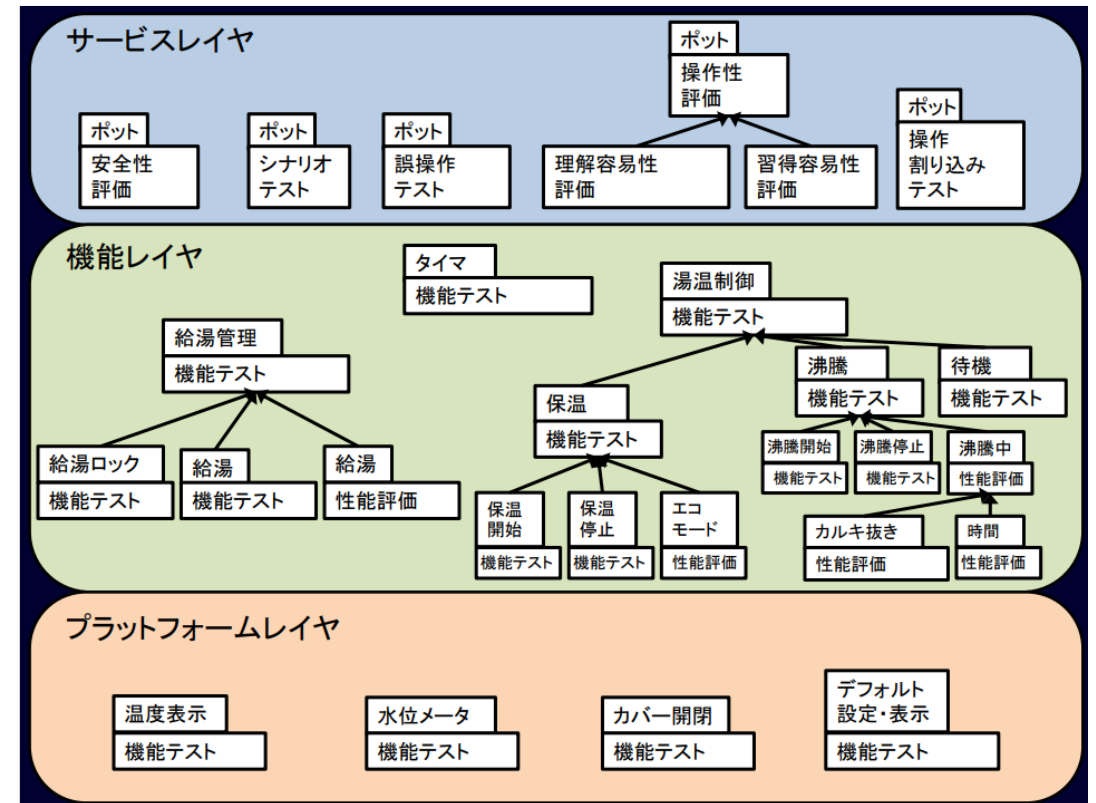
テストアーキテクチャスタイル:レイヤー型

◆考え方、指針

どんな構造に対してテストを実施するか?から考える

◆適用方法

レイヤーを定義し、テスト観点を割り当てる



引用元: テスト設計チュートリアル テスコン編'22 P-46

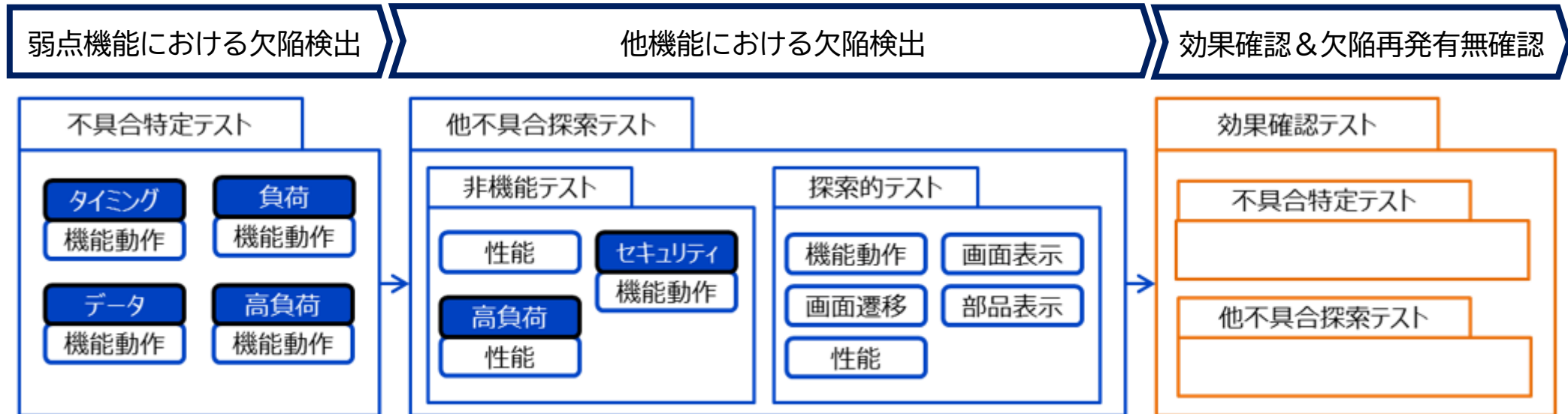
テストアーキテクチャスタイル:フィルター型

◆考え方、指針

どんな欠陥を見つけるためにテストを実施するか？から考える

◆適用方法

検出したい欠陥を定義し、テスト観点を割り当てる



引用元：テスト設計コンテスト'17 OPENクラス わんだーず テストアーキテクチャ抜粋

テストの全体像

◆テストアーキテクチャ(テストの全体像)設計で意識すべきこと

- ・テスト要求分析結果を根拠にしたテスト全体像にしましょう

テスト目的を達成するためのテストになっているか？

エンジニアリング的テスト要求だけでなく、マネジメント的テスト要求も満たしているか？

- ・テストごとのテストアプローチや網羅基準も検討しましょう

マネジメント的テスト要求にはQCDに関わる要求も多々あります

「無限の中から有効な有限に絞り込んだ」テスト全体像を描きましょう

例:コスト要件達成のためにテストアプローチを変える:テストケース→探索的テスト/チャータ

- ・全体像が一つである必要はありません

違う視点で全体像の見せ方を変えてもよい

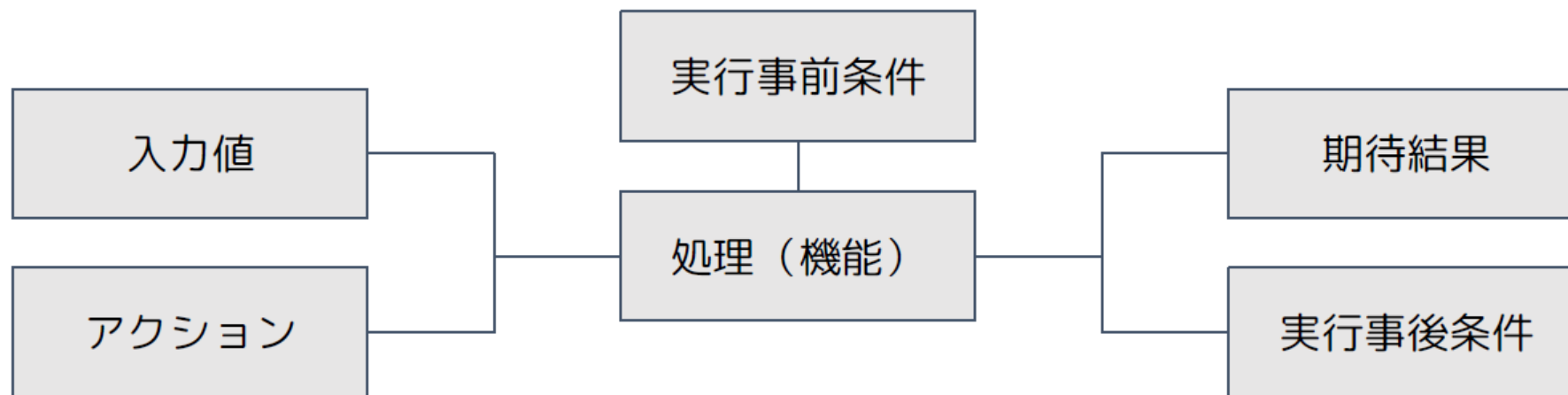
新たなテストアーキテクチャスタイルを作ってもよい

テストケースの構造を示してみよう

◆テストケース（ISTQB用語集より引用）

実行事前条件、入力値、アクション(適用可能な場合)、期待結果、および実行事後条件のセットであり
テスト条件に基づいて開発されたもの

テストケースの構成を図示してみる



テストケースの構造を示してみよう

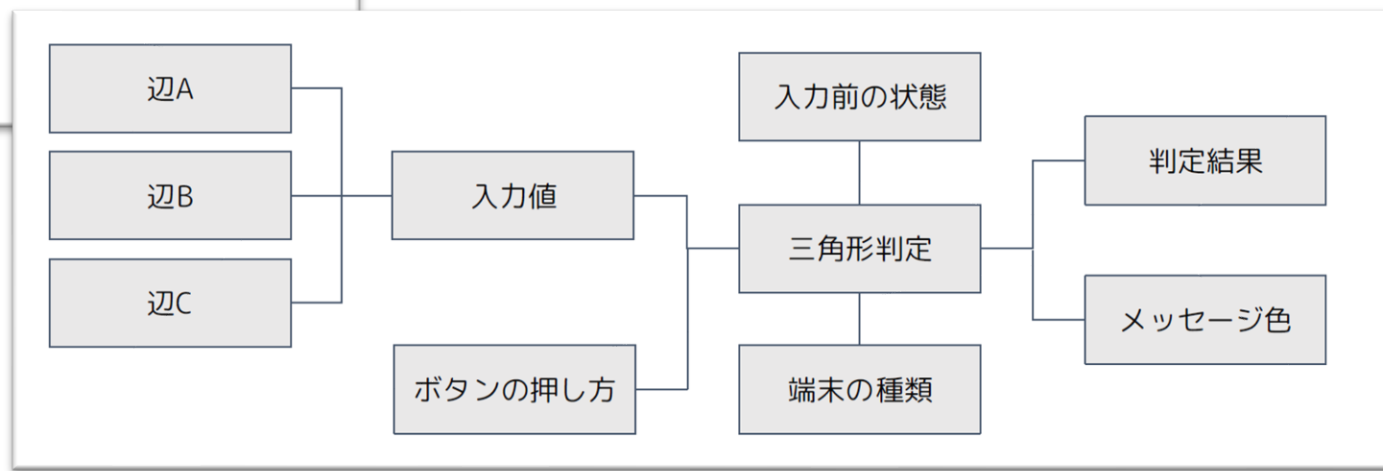
◆テストケースの構造

「三角形判定」というテスト観点に対し、必要な構成要素を出す



シンプルに考える人もいれば……

ひとつのテストケースに
複数のテスト観点をまとめる人もいる



テスト詳細設計



テスト要求の
獲得と整理/
テスト要求
モデリング

テスト
アーキテクチャ
モデリング

テスト技法の
適用による
テストケースの
列挙

手動/自動化
テストスクリプト
(テスト手順)の
記述



引用元：テスト設計チュートリアル ちびこん編'21 資料 P-67

テスト詳細設計

テスト要求分析	テストコンU-30クラス	チュートリアル
定義	テスト技法適用によるテストケースの列挙	テスト観点の網羅基準を定め、適切なテスト技法を選定し、テストケースに落とし込みましょう
INPUT テストベース	テストアーキテクチャ設計の成果物	テストアーキテクチャ設計の成果物 テスト全体像 テストケースの構造
OUTPUT 想定している成果物	テスト設計技法などによって作成されたテストケース群	テストケース仕様 網羅基準も明確にしましょう

テスト詳細設計とは

◆テスト詳細設計

テストアーキテクチャ設計では、テストの全体像を表しました
また、テストケースの構造も示しました

テスト詳細設計では、テスト観点(テストすべきこと)の網羅基準(どのくらいやるか)を設定し、
様々なテスト技法を駆使しながらテストケースに落とし込みます

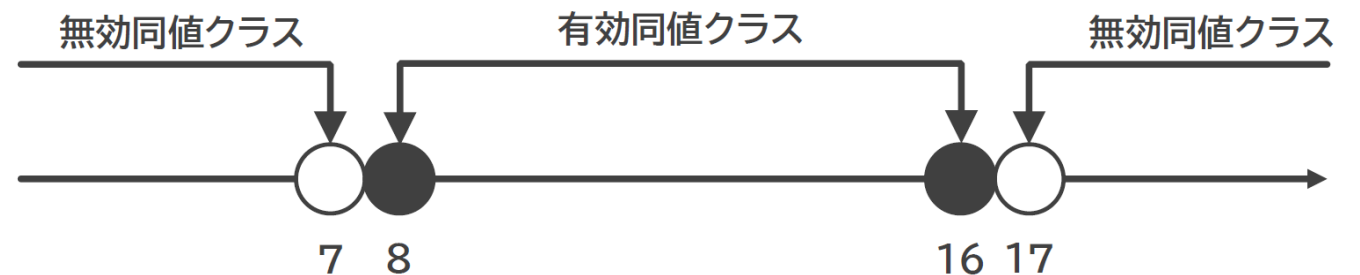
◆網羅基準、テスト技法

仕様:パスワードの文字列長は8文字以上、かつ、16文字以下が有効である

テスト観点 : 文字列長

網羅基準 : 境界値網羅

テスト技法 : 同値分割法・境界値分析



テスト技法

◆ソフトウェアテスト技法に関する参考図書

テスト技法の正しい使い方を学び、テスト設計コンテストで実践しましょう



引用元：Amazonページ



引用元：Amazonページ

テスト実装



テスト要求の
獲得と整理/
テスト要求
モデリング

テスト
アーキテクチャ
モデリング

テスト技法の
適用による
テストケースの
列挙

手動/自動化
テストスクリプト
(テスト手順)の
記述



引用元：テスト設計チュートリアル ちびこん編'21 資料 P-67

テスト実装

テスト要求分析	テスコンU-30クラス	チュートリアル
定義	手動/自動化テストスクリプト (テスト手順)の記述	テスト実行効率や保守性を意識して テスト手順を明確にしましょう
INPUT テストベース	テスト詳細設計の成果物	テスト詳細設計の成果物 テストケース
OUTPUT 想定している成果物	テストケースを実行可能に落とし込んだ (手動/自動)スクリプト	テスト手順仕様

テスト実装

◆テスト実装

テスト詳細設計では、テストケースを作成しました

テスト実装では、テストケースを実行可能な状態にすべくテスト手順を明確にします

その際、**テスト実行効率や保守性を意識しましょう**

◆テスト実行効率

準備やテスト実行を同時にできないか考えてみましょう

環境準備に時間がかかる複数のテストケースをまとめて行うテスト手順にしましょう

複数のテストケースを同時に確認できるテスト手順にしましょう

◆保守性

同じテスト手順を繰り返し書くと可読性が下がり、仕様変更などによる修正量も増えます

複数のテスト手順に共通する内容は別紙にまとめて書くなど保守性を意識しましょう

テスト開発プロセス



テスト要求の
獲得と整理/
テスト要求
モデリング

テスト
アーキテクチャ
モデリング

テスト技法の
適用による
テストケースの
列挙

手動/自動化
テストスクリプト
(テスト手順)の
記述



引用元：テスト設計チュートリアル ちびこん編'21 資料 P-67

第2章

昨年度の審査コメント分析



U-30クラスで評価されるテスト設計

◆説明力

審査コメント	テスト観点、テスト技法の導出理由が読み取れない
アドバイス	<p>「何をしたか」だけではなく「なぜそうしたか」も重要</p> <p>全ての成果物で以下の3点を明示するようにしましょう</p> <ul style="list-style-type: none">Why なぜその観点が必要なのか？ なぜその技法が必要なのか？How どうやって導出したのか？ どの情報から導出したのか？What 何をどこまでテストするのか？ <p>表や図を置くだけではなく、1～2行の意図説明を添えるだけでも説得力は大きく変わります</p>

U-30クラスで評価されるテスト設計

◆PFDと成果物の一貫性

審査コメント	PFD(Process Flow Diagram)と成果物の対応がとれておらず、一貫性が失われている
アドバイス	<p>PFDは「提出用の目次」ではありません。「テスト開発プロセスの設計図」です 最低限、以下を一致させましょう INPUT名やOUTPUT名 ↔ 成果物名</p> <div data-bbox="555 882 1447 1035"><pre>graph LR; INPUT[INPUT] --> PROCESS((プロセス)); PROCESS --> OUTPUT[OUTPUT]</pre></div> <p>▶成果物0: テスト開発プロセス及び成果物の全体像を示したもの</p> <p>提出いただく成果物がどのようなプロセスで作成されたかが分かる、全体の流れが俯瞰できる資料を作成してください。</p> <p>想定ページ数: 最大2ページ以内に納めてください。想定しているもの: PFD (Process Flow Diagram) などプロセスと成果物の関係がわかるもの PFDに限定しません。そのほかの記法でもかまいません。</p> <p>なお、成果物0はPFDに限定はしていません プロセスと成果物の関係や定義が伝わるように記載しましょう</p>

U-30クラスで評価されるテスト設計

◆ やらない判断

審査コメント	制約(工数・期間)を踏まえた優先度判断の説明が不足 何を優先し何を割り切ったのか不明確
アドバイス	<p>「すべてやった」は評価されません(制約が考慮されていない)</p> <p>以下を言語化、明記しましょう</p> <ul style="list-style-type: none">・テスト目的・テストの制約(期間・人数など)・優先順位・あえて実施しなかったものと、その理由 <p>※実務に最も近い評価ポイントです</p>

U-30クラスで評価されるテスト設計

◆新技術・新手法挑戦

審査コメント	新技術・新手法の採用理由や効果が不明確
アドバイス	<p>流行っているから使ってみた！は評価に繋がりません</p> <p>新しいことに挑戦する場合は以下を明記しましょう</p> <ul style="list-style-type: none">・採用理由・従来の手法との差分・得られた効果(想定でも可)

U-30クラスで評価されるテスト設計

◆テスト詳細設計、実装

審査コメント	テストケースが抽象的で、テスト実行時、テスト実行者が迷う
アドバイス	<p>テスト詳細設計、実装は「再現できる」レベルまで落としましょう 誰が実行しても同じ結果になるか？で自己レビューしましょう</p> <p>最低限明示すべきもの</p> <ul style="list-style-type: none">・前提条件・操作手順・合否判定可能な期待結果 <p>探索的テストを行なう場合でも、狙いやテスト観点は明示しましょう</p>

おまけ

◆テスト設計チュートリアル テスコン編

2026/05/13(水) 19:00-21:00 OPENクラスのチュートリアルも開催されます

<https://aster.connpass.com/event/390502/>

◆過去のチュートリアル動画

チュートリアルは発表者ごとに伝え方に差異があります

過去の動画も確認することで、また違った発見、新たな気づきがあるかもしれません

私自身も本資料作成で何度も見返しましたが沢山の気づきがありました♪

非常におすすめです☆

テスト設計コンテストチャンネル

<https://www.youtube.com/@aster-test-design-competition>

もしテストコンに興味を持っていただける方がいらっしやいましたら・・・

テストコン参戦 心待ちにしております♪

<https://www.aster.or.jp/testcontest/>



参考文献/引用文献

いずれも本書作成時点の情報

テスト設計コンテスト	https://www.aster.or.jp/testcontest/
テスト設計コンテスト U-30クラス	https://www.aster.or.jp/testcontest/u30.html
テスト設計コンテスト説明会資料	https://www.aster.or.jp/testcontest/doc/2026_tdc_guidance.pdf
割り勘支援アプリ Warikan仕様書	http://aster.or.jp/testcontest/doc/Warikan_specification_2026.pdf
ASTER U30 テスト設計コンテスト テストプロジェクト要求	https://aster.or.jp/testcontest/doc/aster_U30_tdc_supplement_2026.pdf
ISTQB テスト技術者資格制度用語集	https://glossary.istqb.org/ja_JP/search
ASTERセミナー標準テキスト	http://aster.or.jp/business/seminar_text.html

参考文献/引用文献

いずれも本書作成時点の情報

テスト設計チュートリアル ちびこん編 '25 資料	https://www.aster.or.jp/testcontest/doc/2025_chibicon_v1.0.0.pdf
テスト設計チュートリアル ちびこん編 '24 資料	https://www.aster.or.jp/testcontest/doc/2024_chibicon_v1.0.0.pdf
テスト設計チュートリアル ちびこん編 '23 資料	https://speakerdeck.com/goyoki/test-design-tutorial
テスト設計チュートリアル ちびこん編 '21 資料	http://www.aster.or.jp/business/contest/doc/2021_chibicon_V1.0.0.pdf
テスト設計チュートリアル テスコン編 '25 資料	https://www.aster.or.jp/testcontest/doc/2025_tescon_v1.0.0.pdf
テスト設計チュートリアル テスコン編 '22 資料	https://www.aster.or.jp/testcontest/doc/2022/2022_tescon_V1.0.0.pdf
JaSST' 22 Hokuriku 目指せテスト設計リーダー!! テスト設計チュートリアル 資料	https://www.jasst.jp/symposium/jasst22hokuriku/pdf/B1.pdf

YouTubeチャンネル

ソフトウェアテストに関する情報やコンテストに関する情報を配信していますので、ぜひご活用ください。

ASTERソフトウェアテスト チャンネル



ソフトウェアテストに関する情報はこちら！
(オンラインセミナー動画など)

テスト設計コンテスト チャンネル



テスコンに関する情報はこちら！
(過去のチュートリアル動画など)