ASTER Association of Software Test Engineering

JaSST'22 Hokuriku

14:30～17:10 (160分)

セッション B1

目指せ テスト設計リーダー!!

テスト設計チュートリアル

Ver.1.0.0

ASTER講師：

秋谷 勤（東海ソフト）

大段 智広（テスト設計コンテスト実行委員会）

吉澤 智美（日本電気）

講師紹介

講師プロフィール

秋谷 勤 （東海ソフト）

組込みソフトの開発に従事。主にアーキテクトを担当。
近年品質保証に携わり、開発の視点と品質保証の視点を行き来しながら良い開発を目指している。
JASAアジャイル研究会（2013年～）
テスト設計コンテスト東海審査員（2016年～）

大段 智広 （テスト設計コンテスト実行委員会）

普段の業務では、ソフトウェアテストコンサルティングや開発支援ツール導入支援に従事。過去にはテスト設計コンテスト(`16～`18)に参加。JaSST関西実行委員、テスト設計コンテストU30審査委員/実行委員。JSTQB認定Advanced Level テストアナリスト、テストマネージャ。

吉澤 智美 （日本電気）

普段の業務では、全社のソフトウェア品質向上のための各種活動に従事。テスト設計コンテストでは第一回より審査委員を務める。現在、NPO法人ASTER副理事・テスト設計コンテスト担当理事。JaSST北陸アドバイザ、JSTQB運営委員・技術委員。そのほかテストに関するカンファレンス等運営に携わっている。

もくじ

1. はじめに
2. テスト設計の話をする前に
3. テスト設計とは
4. 今回のお話～良いテストをしていくために～
5. テストアーキテクチャとは
6. テストアーキテクチャの作り方 ・・・ サンプルワーク 個人ワーク
7. テストアーキテクチャの作り方のまとめ
8. さらなるレベルアップのために
9. テスト設計コンテストについて

1.はじめに

- 本チュートリアルの対象者と目的
- 本チュートリアルの進め方

本チュートリアルの対象者と目的

- 対象者
  - テスト計画、テスト要求分析/設計活動をチームでリードしたい方
  - より効果的、効率的なテスト設計を行いたい方
  - ・テストケースを作成できるが、テスト全体を俯瞰してより良いテストを考えたい方
- 目的
  - テストアーキテクチャがなぜ必要なのかを理解する
  - テストの視点でプロダクトの要求を収集＆整理できる
  - 整理した要求を基に、なんとなくテストアーキテクチャが作れるようになる
  - テストアーキテクチャからテストケースが作れそうだなと思ってもらう

本チュートリアルの進め方

- 講義とサンプル＆個人ワークを実施します。
- ワーク資料は、チュートリアル参加される方へ事前に送付してあります。
  - 手元にない方などいましたら送付いたしますのでご連絡ください。
  - また、お送りされている予稿事前配布のURL上からダウンロードできます。
    - 「テスト設計チュートリアルワーク資料」

本チュートリアルの進め方

- 諸注意
  - 講義中は音声、映像をOFFにしてください
  - 撮影/録音は、ご遠慮ください
  - 音声が聞こえないなどトラブルがある場合は「チャット（全員宛）」を利用してお知らせください
  - 荒らし等と判断した場合は強制でご退室いただきます
- 質問方法
  - 「チャット（全員宛）」をご利用ください
    - 適宜回答をしていきます
    - 回答方法は基本的にチャットで行います
- 資料に関して
  - 予稿集から一部変更があります
    後日取得できるよういたします

2.テスト設計の話をする前に

- よくあるテストの問題
- こうならないために

よくあるテストの問題

- テストケースに同じようなものがある
- 漏れているテストケースがある
- テストケースがあっても、このテストをやらなくて良いかわからない
  - 要求仕様とテストケースとのつながりがわからない
  - どんな欠陥を見つけるためのテスト/テストケースかわからない
- テスト実行の後半になって、重大なバグがたくさん出る
- テストケースがいっぱいありすぎて実行できない

テスト設計するのです！

テスト設計コンテスト審査委員
（NPO法人　ASTER）

3.テスト設計とは

- テスト開発プロセスでのテスト設計
- テスト設計なんてできないよ！

テスト条件から、
テストケースを導出し、
明確化する活動。(ISTQBより)

?

もう少し、
やさしく言い換えると…

テストすべきこと
を考えて、
テストケースとして
具体化すること

テスト開発プロセスでのテスト設計

テスト開発プロセスの図は以下より：

テスト設計チュートリアル テスコン編 '21

http://www.aster.or.jp/business/contest/doc/2021_chibicon_V1.0.0.pdf

テスト開発プロセスでのテスト設計

「テストすべきことを考えて、テストケースとして具体化すること」

気にせずいつもやってますよね？

ということでみなさん

テスト設計するのです！

テスト設計コンテスト審査委員
（NPO法人　ASTER）

テスト設計なんてできないよ！

しかし、そもそもやろうと思っても色んな障害がある

例えば…

- テスト設計したいが、開発マネージャ/テストマネージャの理解が得られない
- テスト設計をすることで、テストの何を良くできるか/良くしたらいいかわからない
- どこから初めて良いかわからない

4.今回のお話～良いテストをしていくために～

- テストの全体像を見えるようにしよう
- テストアーキテクチャで解決・改善できる問題
- テストアーキテクチャとは
- テスト設計コンテストでのテストアーキテクチャ例

テストの全体像を見えるようにしよう

- 今回はある程度洗い出されたテスト観点からテスト設計を一から行います。
  - 下記のようなことが議論できるようにテストの全体像を見えるようにしていきます。
    - テストすることが多すぎる/ちょうど良い/少なすぎるかわかる
    - テストがやり切れるか/やり切れないかがわかる
    - 後工程や派生製品、後継のプロジェクトで何をテストすれば良いかわかる

- そのために使用する技術は

テスト観点ベースの
テストアーキテクチャ設計手法

今回は、色々なやり方の中での一例をご紹介します。

テストアーキテクチャで解決・改善できる問題

- テストケースに同じようなものがある
- 漏れているテストケースがある
- テストケースがあっても、このテストをやらなくて良いかわからない
  - 要求仕様とテストケースとのつながりがわからない
  - どんな欠陥を見つけるためのテスト/テストケースかわからない
- テスト実行の後半になって、重大なバグがたくさん出る
- テストケースがいっぱいありすぎて実行できない

今回、「解決 改善」を目指す箇所

5.テストアーキテクチャとは

- テストアーキテクチャとは
- テストアーキテクチャ設計の全体像
- アーキテクチャドライバ
- アーキテクチャドライバの種類
- テスト設計コンテストでのテストアーキテクチャ例

テストアーキテクチャとは

- 「テストの骨組みや指針を表したもの」
  - ステークホルダー(後工程や派生製品、後継プロジェクトに関連する人)にとって、テストの全体像を把握できる/見えるもの

- 見えるようになると、事前に発生しそうな問題を予見して対処することができる
- 言い換えると、全体として良いテストかどうか、作りたいテストになっているか確認できる
  - 例
    - テストすることが多すぎる/ちょうど良い/少なすぎるかわかる
    - テストがやり切れるか/やり切れないかがわかる
    - 後工程や派生製品、後継のプロジェクトで何をテストすれば良いかわかる など

テストアーキテクチャ設計の全体像

問題領域

- システムの種類
- テスト設計目的(方針)
- テスト観点
  - エンジニアリング的テスト要求
  - マネジメント的テスト要求
- 品質属性シナリオ
- 関心事

テストアーキテクチャを形作る主たる要求群(入力)

設計活動

テストアーキテクト

解決方法

テストアーキテクチャの設計物(出力)

ASTER Association of Software Test Engineering

アーキテクチャドライバ

- アーキテクチャを形作る主たる要求群
  - アーキテクチャ設計の入力となる

問題領域

- システムの種類
- テスト設計目的(方針)
- テスト観点
  - エンジニアリング的テスト要求
  - マネジメント的テスト要求
- 品質属性シナリオ
- 関心事

テストアーキテクチャを形作る主たる要求群(入力)

設計活動

テストアーキテクト

解決方法

テストアーキテクチャの設計物(出力)

ASTER Association of Software Test Engineering

アーキテクチャドライバの種類(1/3)

- テスト観点(≒テスト要求)
  - テスト対象のテストすべき側面、テストが達成すべき性質のこと
  - テスト観点は大きく二つある
    - エンジニアリング的テスト要求
      - テストケースを導くもの
      - 硬貨受付に関する仕様を満たすテストをしたい など
    - マネジメント的テスト要求
      - テストケースを導かないもの
      - 技術的もしくは組織的な限界、制限
        - 工数、人数、スキル分布、作業場所、オフショアか否か、契約形態など
        - 機材利用可否（シミュレータや試作機など）、ツール利用可否（ツールの種類、保有スキル）など
      - 顧客もしくは開発組織から得られる
      - 設計上の意思決定をする上で役に立つ

- 品質属性
  - 測定可能な特性
    - 性能、可用性、変更容易性など
  - シナリオ形式で記述
    - 例:あるテスト対象機能が変更された場合、影響を受けるテストケースが30分で特定できる

ASTER Association of Software Test Engineering

アーキテクチャドライバの種類(2/3)

- システムの種類
  - 新しい領域の新規開発のシステム
    - 例：AIシステム
    - 良くわかっているドメインではなく、よりイノベーティブ
  - 成熟した領域の新規開発のシステム
    - 標準的なPCアプリ、モバイルアプリ
    - イノベーティブではないが良くわかっているドメイン
  - 既存開発のシステム
    - 既存システム(テストアーキテクチャ含)の変更

ASTER Association of Software Test Engineering

アーキテクチャドライバの種類(3/3)

- テスト設計目的(方針)
  - プロトタイプ開発
    - 初期ソリューションのため、迅速に設計する必要がある
  - 顧客用システム
    - 時間・コストが確立している。リリース後にはあまり進化しない
  - 継続的に進化するシステム など

- 関心事
  - 要求や達成したい事として明示的に記述されているかに関わらず必要となる設計上の決定
    - テスト成果物の作成、コミュニケーション、テスト成果物管理、バックアップなど

ASTER Association of Software Test Engineering

テストアーキテクチャ設計の全体像

問題領域

- システムの種類
- テスト設計目的(方針)
- テスト観点
  - エンジニアリング的テスト要求
  - マネジメント的テスト要求
- 品質属性シナリオ
- 関心事

テストアーキテクチャを形作る主たる要求群(入力)

設計活動

テストアーキテクト

解決方法

テストアーキテクチャの設計物(出力)

ASTER Association of Software Test Engineering

テストアーキテクチャ設計の全体像

問題領域

- システムの種類
- テスト設計目的(方針)
- テスト観点
  - エンジニアリング的テスト要求
  - マネジメント的テスト要求
- 品質属性シナリオ
- 関心事

テストアーキテクチャを形作る主たる要求群(入力)

ここを中心に取り上げて、設計します。

設計活動

設計上の決定

テストアーキテクト

解決方法

テストアーキテクチャの設計物(出力)

引用:テスト設計チュートリアル テスコン編(講義編)

ASTER Association of Software Test Engineering

テスト設計コンテストでのテストアーキテクチャ例

例1:チュートリアルサンプルのテストアーキテクチャ

例2:電気ポッドのテストアーキテクチャ

引用:テスト設計チュートリアル テスコン編(講義編)

ASTER Association of Software Test Engineering

6.テストアーキテクチャの作り方

サンプルワーク　個人ワーク

ワークの進め方
個人ワーク テスト対象の説明
テストアーキテクチャ作成手順

ワークの進め方

- テストアーキテクチャの作り方を紹介し、サンプルワーク→個人ワークを繰り返しながら、段階を踏んで進めていきます。
  - ワークの内容
    - サンプルワーク　：簡単なお題
    - 個人ワーク　：事前に提示した仕様書に基づくワーク

- ワーク資料は、チュートリアル参加される方へ事前に送付してあります。
  - 手元にない方などいましたら送付いたしますのでご連絡ください。

- 個人ワークの結果のフィードバックを希望の方は、下記のGoogleドライブへ格納していただければ、あとでコメントいたします。
  - Googleドライブ:個人ワーク共有フォルダ
    - ＜予稿集、当日のみ案内＞

個人ワーク

テスト対象の説明

- 仕様書やワークの資料は読んできましたか？

- だんだん動物園券売機
  - ちょっと古い券売機
    - コインしか受け付けない
    - タッチパネルでない
    - いくつかのシステムにより構成されている

だんだん動物園券売機
要求仕様書

Ver.1.0.6

ASTER-テスト設計コンテスト実行委員会

© 2020 ASTER-テスト設計コンテスト実行委員会

今回のチュートリアルでのテストアーキテクチャ作成手順

手順

① テスト要求分析で洗い出されたテストすべきテスト観点を整理しよう

② アーキテクチャスタイルを選ぼう

③ アーキテクチャスタイルにグループをマッピングしよう

④ 関係者と検討して、テストについて合意しよう

①テスト要求分析で洗い出されたテストすべきテスト観点を整理しよう(1)

- まずは、テスト要求分析で洗い出されたテストすべきテスト観点を確認します。
- テスト観点(≒テスト要求)
  - テスト対象のテストすべき側面、テストが達成すべき性質のこと
  - テスト観点は大きく二つに分類されます。
    - エンジニアリング的テスト要求
      - テストケースを導くもの
      - 硬貨受付に関する仕様を満たすテストをしたい など
    - マネジメント的テスト要求
      - テストケースを導かないもの(主にテスト計画で明確にするもの)
      - 技術の制限、もしくは組織的な限界、制限
        - 工数、人数、スキル分布、作業場所、オフショアか否か、契約形態など
        - 機材利用可否（シミュレータや試作機など）、ツール利用可否（ツールの種類、保有スキル）など
      - 顧客もしくは開発組織から得られる
      - 設計上の意思決定をする上で役に立つ

<<テスト観点>>
<<エンジニアリング的テスト要求>>

日本硬貨以外を受け付けないこと

<<テスト観点>>
<<マネジメント的テスト要求>>

通しのテストはテスト工数は60人日で抑えたい

<<テスト観点>>
<<マネジメント的テスト要求>>

xxx規格に合ったテストを実施してほしい

テスト要求洗い出す方法については、下記をご参照ください
テスト要求分析チュートリアル(2:00:15)
https://www.youtube.com/watch?v=KrMwXNtotbU

ASTER Association of Software Test Engineering

①テスト要求分析で洗い出されたテストすべきテスト観点を整理しよう(2)

- テストを作りやすく、検討しやすくするために、下記のような単位でグルーピングしていきます。
  - 下記のような方針でグルーピングしたものは一緒にテストしやすくなります
    - グルーピングの方針
      - テスト対象の機能
      - テスト対象の利用者
      - テスト対象の利用シーン など
  - 慣れていたり、やりやすいと思う方針でやってみましょう。
- グルーピングして方針に沿って整理することで、抜け/漏れている、不明確なテスト観点に気付くこともあります。

ASTER Association of Software Test Engineering

グルーピングの方針：テスト対象の機能

- 仕様書から機能っぽい言葉をさがす
  - ずばり「機能」と書いてあるもの
- 物理的なものを探す
  - 組込みならハード機器
  - アプリ
    - 画面
    - ユーザー操作
      - スマフォ
        - タップ
        - スワイプ
    - PC
      - クリック
      - ドラッグアンドドロップ

機能を示す言葉

だんだん動物園券売機なら以下の表など

No.	名称	説明
1	電源ランプ	電源が供給されていることを通知する。
2	購入ボタン	購入形態に応じたチケットの購入を行う。
3	返金ボタン	投入されたコイン(硬貨)の返金処理を行う。
4	購入ボタンランプ	購入ボタンに応じた購入可否を通知する。
5	釣銭切れランプ	釣銭が切れている旨を通知する。
6	7セグLED	投入金額の表示やエラー発生時のエラーコードを表示する
7	コイン投入口	コイン(硬貨)を投入するためのの口。
8	コイン返却口	コインメックからコイン(硬貨)を返却される口。
9	コインメック	投入および購入されたコイン(硬貨)の金銭処理を行う。
10	プリンタ	購入したチケットを印字、カットする。
11	発券口	プリンタで印字、カットされたチケットを出す。
12	券取り出し口	出されたチケットを受け取るためのの口
13	マザーボード	プリンタ、コインメックなど接続する機器への通信および処理を行う。(※内部処理が中心のため、記載は割愛)
14	開閉鍵口	券売機の前面を開閉するための鍵口。

購入

・・・

発券

・・・

テスト対象の機能

購入(グループ)

テスト観点　テスト観点

発券(グループ))

テスト観点　テスト観点　テスト観点

グルーピングの方針：ステークホルダー(利害関係者)

- 製品の利用者はその関係性によって目的が変わります
  その機能は誰に向けての機能であるかによってグルーピングする

テスト対象の利用者

ASTER Association of Software Test Engineering

グルーピングの方針：テスト対象の利用シーン

- 製品の利用者は目的をもってその製品を利用します
- 「ステークホルダー(利害関係者)」とつなげるとイメージしやすい
- 使う人の目的・やりたいことをイメージしよう

サンプルワーク

テスト観点をグルーピングしよう

- 都道府県の【地方】を判定するプログラムのテスト観点をグルーピングしてください。(例：神奈川→「関東地方」)
- グルーピングして、気になることを見つけましょう。

所要時間:3分

15:27まで

<<テスト観点>>
福井

<<テスト観点>>
愛知

<<テスト観点>>
石川

<<テスト観点>>
宮城

<<テスト観点>>
新潟

<<テスト観点>>
三重

<<テスト観点>>
京都

<<テスト観点>>
東京

<<テスト観点>>
神奈川

<<テスト観点>>
富山

<<テスト観点>>
大坂

ASTER Association of Software Test Engineering

サンプルワーク

テスト観点をグルーピングしよう：解答例

- 都道府県の【地方】判定するプログラムのテスト観点をグルーピングしてください。

所要時間:3分

北陸地方
- <<テスト観点>> 石川
- <<テスト観点>> 富山
- <<テスト観点>> 福井
- <<テスト観点>> 新潟

東北地方
- <<テスト観点>> 宮城
- <<テスト観点>> 新潟

東海地方
- <<テスト観点>> 愛知

関西(近畿)地方
- <<テスト観点>> 大坂
- <<テスト観点>> 京都
- <<テスト観点>> 三重
- <<テスト観点>> 新潟

関東地方
- <<テスト観点>> 東京
- <<テスト観点>> 神奈川

特にテスト観点【新潟】がどのような地方になるのかは、グルーピングの方針(**国土交通省、気象庁の区分など**)によっても変わります。
(他にも、中部地方、甲信越地方、関東甲信越地方にグルーピングすることもできます。)

ASTER Association of Software Test Engineering

個人ワーク

テスト観点をグルーピングしよう

- テスト観点のエンジニアリング的テスト要求をグルーピングしてください。
- 「これも必要だ」と思うテスト観点を追加しても構いません。

所要時間:15分

15:50まで

事前配布したワーク資料をご参照ください。

APPENDIX：テスト観点に含まれるテスト設計者にとってわかりやすい情報(1)

- 本来はテスト要求分析でテスト観点をテスト設計者がわかりやすい表現にまとめ、分類しておくと確認しやすい。
  - 最初は、テスト観点名だけでも良いが、確認する必要がありそうならまとめておくとより良いです。

例

簡潔にテストで何をする？

なんでテストするの？

誰から聞いたんだっけ？

ID	カテゴリ	テスト観点名	詳細説明	理由	テスト対象	導出元	事前に確認するテスト観点
TestVP-001	エンジニアリング的テスト要求	日本硬貨以外を受け付けないこと	海外硬貨でも受け付けてしまう	不正にお釣りを取られる	コイン投入口	00_だんだん動物園券売機_要求仕様書_v106	TestVP-XXX
TestVP-002	エンジニアリング的テスト要求	チケットが仕様通り印刷されること	印刷内容を実機で確認する	実機で印字がずれたり、かすれたりすることがあるため	プリンタ	00_だんだん動物園券売機_要求仕様書_v106	-
TestVP-003	マネジメント的テスト要求	通しのテストはテスト工数は60人日で抑えたい	-	人が調整しにくいため	-	開発マネージャ	-
・・・	・・・	・・・	・・・	・・・	・・・	・・・	・・・

テストケース関連？
テスト計画関連？

詳しく言うと？

どこをテストするの？

事前にやっておくテストはどれ？

ASTER Association of Software Test Engineering

APPENDIX：テスト観点に含まれるテスト設計者にとってわかりやすい情報(2)

- 対応すべきテスト観点を抽出する
- テストアーキテクチャを導くもの(アーキテクチャドライバ)になりそうな要求を確認しておく。
  - テストをするために工夫をする必要がある下記のテスト観点が該当します。
    - テストするのが特に難しそうなテストエンジニアリング要求
    - テストに制約をもたらすテストマネジメント要求
  - 作ったテストアーキテクチャが妥当なものか評価するための項目になります。

例

ID	カテゴリ	テスト観点名	詳細説明	…	導出元	事前に確認するテスト観点	対応採否	採否理由	アーキドライバ
TestVP-001	エンジニアリング的テスト要求	日本硬貨以外を受け付けないこと	海外硬貨でも受け付けてしまう	…	00_だんだん動物園券売機_要求仕様書_v106	TestVP-XXX	採	(略)	
TestVP-002	エンジニアリング的テスト要求	チケットが仕様通り印刷されること	印刷内容を実機で確認する	…	00_だんだん動物園券売機_要求仕様書_v106		採	(略)	
TestVP-003	マネジメント的テスト要求	通しのテストはテスト工数は60人日で抑えたい	-	…	開発マネージャ		採	(略)	○
・・・	・・・	・・・	・・・		・・・	・・・	・・・	・・・	・・・

ASTER Association of Software Test Engineering

②アーキテクチャスタイルを選ぼう

- ゼロからテストアーキテクチャを作るのは大変です。
- まず第一歩としては、テスト全体を俯瞰するために、良く使われるテストアーキテクチャスタイル(≒考え方、指針)を基にして作ってみると良いです。

- 代表的なアーキテクチャスタイル(型)
  - テストタイプ型
  - レイヤー型
  - フィルター型
  - 複合型 etc,..,

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
- 適用目的
  - 組織の典型的なテストタイプにグループをマッピングする
  - ISO25010等の品質特性毎にグループをマッピングする
- 主な関心事
  - 既存のテストタイプを用いてどんなテストをするか
  - 各テストタイプでテストすべき/しないことを明確にする

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

ASTER Association of Software Test Engineering

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するか」から考える
- 適用方法
  - レイヤーを定義する（システム、アプリケーション、テストレベル）
  - 定義したレイヤーにグループをマッピングする
- 主な関心事
  - 各レイヤーでテストすべきことを明確にする
    - 特にテスト対象のシステムの場合
      - テスト対象の何に対してどんなテストをするか
  - 他レイヤーでテストしないことを明確にする

引用:テスト設計チュートリアル テスコン編(講義編)

ASTER Association of Software Test Engineering

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てるためにテストするか」から考える
- 適用方法
  - 検出したい欠陥のフィルターを定義する
    - 基本は、小さな粒度の欠陥からフィルターの層を定義する
  - フィルターにグループをマッピングする
- 主な関心事
  - どの欠陥フィルタで何の欠陥を取るか
  - どのフィルタでどんなテストをするか

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

サンプルワーク

アーキテクチャスタイルを選ぼう

- 自分たちの考え方に近いアーキテクチャスタイルを選んでください。
- また、どこが近いのかの理由を記載してください。
- 普段の自分たちがどんな考え方、指針でテストで検討してるか振り返ってみましょう。

所要時間:3分

16:17:30

アーキテクチャスタイル

①テストタイプ型　②レイヤー型　③フィルター型　④それ以外

解答例

記入欄

アーキテクチャスタイル:　①テストタイプ型

理由：

通常は、組織標準のテストレベルやテストタイプが存在しており、それに基づいてテストを作成しているため。
用意されてるテストレベルやテストタイプで検討することで、関係者に説明しやすいため。

ASTER Association of Software Test Engineering

③アーキテクチャスタイルにグループをマッピングしよう（1）

- まとめたテストグループをアーキテクチャスタイルにマッピングして検討する
  - 検討観点例
    - アーキテクチャスタイルの考え方、指針に沿って考えられているか
    - 一緒にテストした方が良いものはあるか
    - どの順番でテストをすれば、効率よく欠陥を見つけられるか
    - 特定のタイミングじゃないとテストできないものはないか
    - マネジメント的テスト要求を満たしているか　など
      - 例:通しのテストはテスト工数は60人日で抑えたい

ASTER Association of Software Test Engineering

③アーキテクチャスタイルにグループをマッピングしよう（2）

- 悩むようだったら、複数のテストアーキテクチャ案を作るのもありです。
  - そのあとに関係者と検討するときに役に立ちます。
    - どちらがより良くテストできるかが考えやすくなります。

案1 テストアーキテクチャA

案2 テストアーキテクチャB

テストアーキテクト

ASTER Association of Software Test Engineering

④関係者と検討して、テストについて合意しよう(1)

- テストで検討したいことを議論する
  - 関係者のやりたいことすべてを叶えるのことは難しい。トレードオフ関係となっている、対立したやりたいことがあることをお互い認識して話し合う。
    - テスト実行範囲:列挙したすべてをテストしたい ⇔ 実行速度:早くテストを終えたい
  - アーキテクチャドライバを並べて、対応して実現できているか確認する
- 合意できなければ、前の手順(①～③)に戻り、再検討して、テストアーキテクチャを洗練させていきます。

ASTER Association of Software Test Engineering

個人ワーク

アーキテクチャスタイルにグループをマッピングしよう

- アーキテクチャスタイルを選び、個人ワーク1のシートで作成したグループをマッピングしてください。
- 個人ワーク1のシートからグループをコピーして利用してください。

所要時間:15分

16:41まで

事前配布したワーク資料をご参照ください。

共有先 https://drive.google.com/drive/folders/1VUl5TOcJHXx_u_qzNI0r19dsdhj7lQVj?usp=sharing

⑤関係者と検討して、テストについて合意しよう（2）

- 合意後は、テストアーキテクチャのテスト観点一つまたは複数に注目して、テストケースを作成します

テスト詳細設計では、テスト観点に着目し、必要があればテスト技法を用いて、テストケースを作成する

入力データA	テスト手順	期待結果	テスト結果
…	…	…	…
…	…	…	…
…	…	…	…
…	…	…	…

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

7.テストアーキテクチャの作り方のまとめ

- テストアーキテクチャでテストの全体像を見えるようにしよう
- テストアーキテクチャで解決・改善できる問題

テストアーキテクチャでテストの全体像を見えるようにしよう

- 「テストの骨組みや指針を表したもの」
  - ステークホルダー(後工程や派生製品、後継プロジェクトに関連する人)にとって、テストの全体像を把握できる/見えるもの

- 見えるようになると、事前に発生しそうな問題を予見して対処することができる
- 言い換えると、全体として良いテストかどうか、作りたいテストになっているか確認できる
  - 例
    - テストすることが多すぎる/ちょうど良い/少なすぎるかがわかる
    - テストがやり切れるか/やり切れるかがわかる
    - 後工程や派生製品、後継のプロジェクトで何をテストすれば良いかがわかる など

ASTER Association of Software Test Engineering

今回のチュートリアルでのテストアーキテクチャ作成手順

手順

① テスト要求分析で洗い出されたテストすべきテスト観点を整理しよう

② アーキテクチャスタイルを選ぼう

③ アーキテクチャスタイルにグループをマッピングしよう

④ 関係者と検討して、テストについて合意しよう

テストアーキテクチャで解決・改善できる問題

- テストケースに同じようなものがある
- 漏れているテストケースがある
- テストケースがあっても、このテストをやらなくて良いかわからない
  - 要求仕様とテストケースとのつながりがわからない
  - どんな欠陥を見つけるためのテスト/テストケースかわからない
- テスト実行の後半になって、重大なバグがたくさん出る
- テストケースがいっぱいありすぎて実行できない

今回、「解決 改善」を目指した箇所

さらに頑張るとより「解決 改善」が見込める箇所

ASTER Association of Software Test Engineering

8.さらなるレベルアップのために

- 何をレベルアップさせていくべきか
- テスト要求分析⇔テストアーキテクチャ設計 繰り返し＆洗練
- テストフレームモデリングによるテスト詳細設計
- テストリバースエンジニアリング
- テスト設計内容を伝える技術
- やる気になった方々へ

何をレベルアップさせていくべきか

①テスト要求分析⇔テスト
アーキテクチャ設計 繰り返し
＆洗練

②テスト技法を使ってテストケースを
網羅的に作ろう

③テストリバースエンジニアリング

④テスト設計内容を伝える技術

①テスト要求分析⇔テストアーキテクチャ設計 繰り返し＆洗練

- 一回じゃできないので、何回も繰り返し考えて、洗練させていきましょう。

1. 方向づけ
   - 目的や基準、方針決め
2. 発散
   - 決めた方向に沿って考えやワードを発散させる
3. 集約(再構築)
   - 発散させた物をあるルールに基づいてグルーピング
   - 粒度合わせ
     - 抽象表現の具体化
     - 具体表現の抽象化

テスト開発プロセスの図は以下より：

http://aster.or.jp/business/contest/doc/2019_U-30_V1.0.0%20.pdf

②テスト技法を使ってテストケースを網羅的に作ろう

- 出来上がったテストアーキテクチャを基にテストケースを作りましょう。
  - テスト観点を詳細化して、テストパラメータを導出し、テスト技法と技法に対応するカバレッジを適用することで、網羅的にテストケースを提示できるようになります。

- 例

<<テスト観点>>
<<エンジニアリング的テスト要求>>

日本硬貨以外を受け付けないこと

→ テスト設計技法
同値分割法

カバレッジ：識別されたパーティション数 ⇒ 2

有効パーティション (硬貨を受け付ける)	無効パーティション (硬貨を受け付けない)
・10円硬貨 ・50円硬貨 ・100円硬貨 ・500円硬貨	・1円硬貨 ・5円硬貨 ・旧500円硬貨 ・海外硬貨A

テスト観点の内容を詳細化して
洗い出したテストパラメータ

③テストリバースエンジニアリング

- 既存のテストから作ってみよう
- 新たな発見があるかも
  - 新規のテストした方が良いところ(≒新たなテスト観点)
  - 冗長なテストの削除、テストの効率化

ASTER Association of Software Test Engineering

④テスト設計内容を伝える技術

- テスト設計には様々な技術が必要です。
- 今回様々な人が作成した資料を見たと思います。

- 伝える技術もレベルアップしていきましょう！
  - 目標基準
    1. 今の自分が読んで伝わる
    2. 未来の自分が読んで伝わる
    3. 他の人が読んで伝わる

- テスト設計って他にどんな伝え方があるか・・・

ASTER Association of Software Test Engineering

やる気になった方々へ

- 自分の作ったテスト設計が良いものか知りたくないですか？
- 他の人がどんなふうにまとめているのか知りたくないですか？

- そんな方のためにASTERでは、

テスト設計コンテストを開催しています！！

9.テスト設計コンテストについて

- テスト設計コンテストの目的
- テスト設計コンテストの形式
- テスト設計コンテスト参加のメリット！
- テスト設計コンテストのこれまでの歩み
- テスト設計コンテスト OPENクラス審査基準
- テスト設計コンテスト OPENクラス成果物

テスト設計コンテストの目的

- ソフトウェアテストを分析設計から行うことを周知し、ソフトウェアテストエンジニアに対する教育の機会を提供する
- コンテストという形式をとることにより、ソフトウェアテストが創造的な作業であり楽しいということを経験してもらい、若年層及び初級テストエンジニアからベテランテストエンジニアまでテストへの興味を高める
- ソフトウェアテスト業界における技術開発を競技を通じ、促進する

テスト設計コンテストの形式

- 参加者を問わないOPENクラスと、
  全チームメンバが30歳以下限定のU-30クラスがあります。
- テスト設計成果物の良さを競うコンテスト。
  バグ出しコンテストではありません。
- 各チームは共通のテストベースに対するテスト設計をおこなって、
  成果物を提出し、予選会や決勝戦でプレゼンテーションする。

テスト設計コンテスト参加のメリット！

- テスト設計経験豊富な審査委員からのフィードバック
  - U-30クラスは提出したテスト設計成果物に対してフィードバック
  - OPENクラスはさらにプレゼンテーションについてもフィードバック
  - 予選会後に、審査を担当した全審査委員からのフィードバックコメントシートが届きます。

- 他チームのテスト設計成果物から学ぶ
  - どのチームも同じテストベースを扱う
  - 自分たちが一度テスト設計をしたテストベースなので、他チームのテスト設計成果物を見たときの学びが深い

テスト設計コンテストのこれまでの歩み

開催年	結果	
2011	【大賞】めいしゅ館（東海） 【湯本賞】奥村 健二（東海）、【にし賞】堀米 賢（東京）	
2012	【最優秀賞】TETTAN（東京） 【審査員特別賞】あまがさきてすとくらぶ（関西）	
2013	【優勝】TETTAN（東京） 2連覇 【準優勝】Yuki Da RMA（北海道）	
2014	【優勝】ＴＦＣ　ＫＡ･ＲＩ･ＹＡ （東海） 【準優勝】MKE98（審査委員推薦枠：東海）	
2015	【優勝】しなてす（東京） 【準優勝】TEVASAKIplus （東京）	
2016	【優勝】SASADAN Go （書類選考） 【準優勝】しなてす （東京）	
2017	OPENクラス 【優勝】STUDIO IBURI（北海道） 【準優勝】わんだーズ （東京）	U-30クラス 【優勝】でこパン462 【準優勝】SHINNOSUKE
2018	OPENクラス 【優勝】てすにゃんV3（審査委員特別枠） 【準優勝】フワパン（東京）	U-30クラス 【優勝】TBD 【準優勝】新米
2020	OPENクラス 【優勝】セクシーゴリラ 【準優勝】出席番号となり同士	U-30クラス 【優勝】一等米 【準優勝】王バーフロー
2021	OPENクラス 【優勝】テス豆 【準優勝】シン･田町補充計画	U-30クラス 【優勝】はじめての共同作業 【準優勝】まちがいさがし。

テスト設計コンテスト OPENクラス審査基準

- 成果物審査点　合計120点満点
  - テスト分析・テスト設計点(40点)
  - テスト詳細設計・実装点(30点)
  - 工程間一貫性点(10点)
  - 文書点(20点)
  - 総合点(20点)

- プレゼンテーション技術点　15点

審査基準はテスト設計コンテスト 公式ページにて公開

http://aster.or.jp/business/contest/rulebook.html#agreement

テスト設計コンテスト OPENクラス成果物

- 成果物１　：　テスト設計の最終成果物：40ページまで
  - メインの審査対象
- 成果物２　：　テスト設計に関わる成果物一式
  - テスト設計を行う過程
- 成果物３　：　アピールシート
  - 各チーム紹介、狙いなどを表現

成果物4：プレゼンテーション用資料

- プレゼンテーション：20分(発表15分、質疑応答5分)で自チームのテスト設計を説明

APPENDIX:参考文献

- 資料
  - [Kazman] Humberto Cervantes, Rick Kazman, ADD 3.0: Rethinking Drivers and Decisions in the Design Process, SATURN2015.
    - https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=436536
  - JaSST'16 Tohoku
    - http://www.jasst.jp/symposium/jasst16tohoku/report.html
  - ソフトウェアテストシンポジウム 2021 東海【初心者向け】テストを導くためのテストアーキテクチャの組み立て方
    - https://speakerdeck.com/goyoki/cetam

- チュートリアル資料
  - テスト設計チュートリアル ちびこん編
    - http://aster.or.jp/business/contest/tutorial_chibicon.html
  - テスト設計チュートリアル テスコン編
    - http://aster.or.jp/business/contest/tutorial_tescon.html
  - ソフトウェアテストシンポジウム 2020/2021 北陸 テスト設計初心者向けチュートリアル
    - http://www.jasst.jp/symposium/jasst21hokuriku/pdf/B1.pdf

- Youtube
  - テスト要求分析チュートリアル(2:00:15)
    - https://www.youtube.com/watch?v=KrMwXNtotbU

APPENDIX:図や表の活用

- 図や表を活用すると考えがまとまりやすくなります。
  - 俯瞰
  - ルール
    - 配置
    - 線の意味
- よくあるツール
  - 図
    - マインドマップ
    - ロジックツリー
  - 表
    - デシジョンテーブル
- 使いこなすためのポイント
  - 条件が複雑なときは分解する
    - 考える範囲を減らす
  - バラバラな粒度をあわせる
    - 同じレベル、隣の行、列で浮いていないかがチェックできます。

ASTER Association of Software Test Engineering