

14:35～17:00 (145分)

セッション B1

テスト設計初心者向け チュートリアル

Ver.1.0.2

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）



やること



- 当日講義とワークを実施します
- ワーク資料は、当日までにチュートリアル参加される方へ事前に送付する予定です。
 - 当日チュートリアルまでに資料をご参照いただくと、ワークがスムーズになりますので事前のご参照をお願いいたします。





講義及びワークご連絡



- 諸注意
 - 講義中は音声、映像をOFFにしてください
 - ワークの一部ではONにさせていただきます
 - 撮影/録音は、ご遠慮ください
 - 音声がかえれ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的に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

右上のテストコンに変更の目印



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。



1. テスト設計の話をする前に
2. テスト設計とは
3. 今回のお話～良いテストをしていくために～
 - テストケースをキレイキレイしよう
4. テストリバースエンジニアリング
 - ヌケモレを減らす
 - ムダを減らす
5. テストリバースエンジニアリングの実践ワーク
6. さらなるレベルアップのために
7. テスト設計コンテストについて



1. テスト設計の話をする前に



- よくあるテストの問題
 - テストケースが重複する
 - テストケースがいっぱいありすぎて実行できない
 - テストケースのみが存在し、テストの意図がわからない
 - テストケースがなく、何もない
 - マニュアルから作れるがエラーケースが作れない
 - テストが終わったあとにバグが出る
 - 動かせばわかるでしょ？という箇所バグがでる
 - 何かとつなげたらバグが出る



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)



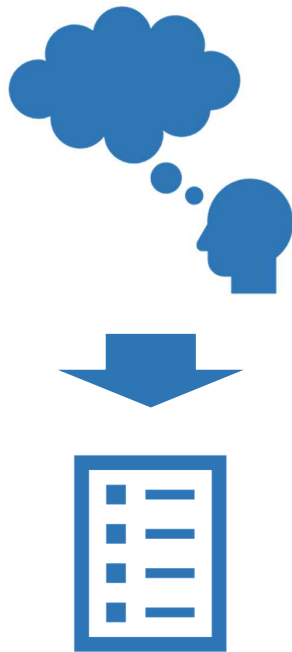
テスト設計とは



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…



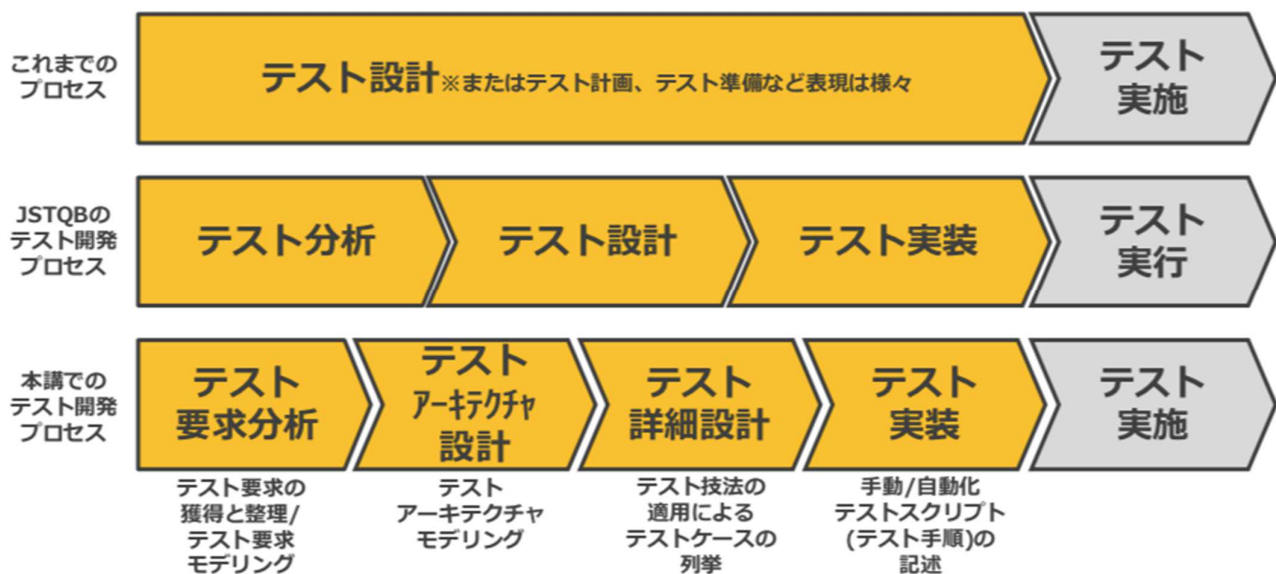
テストすべきことを考えて、 テストケースとして 具体化すること

11

ASTER Association of Software Test Engineering



テスト設計のプロセス



開発プロセスの図は以下より：

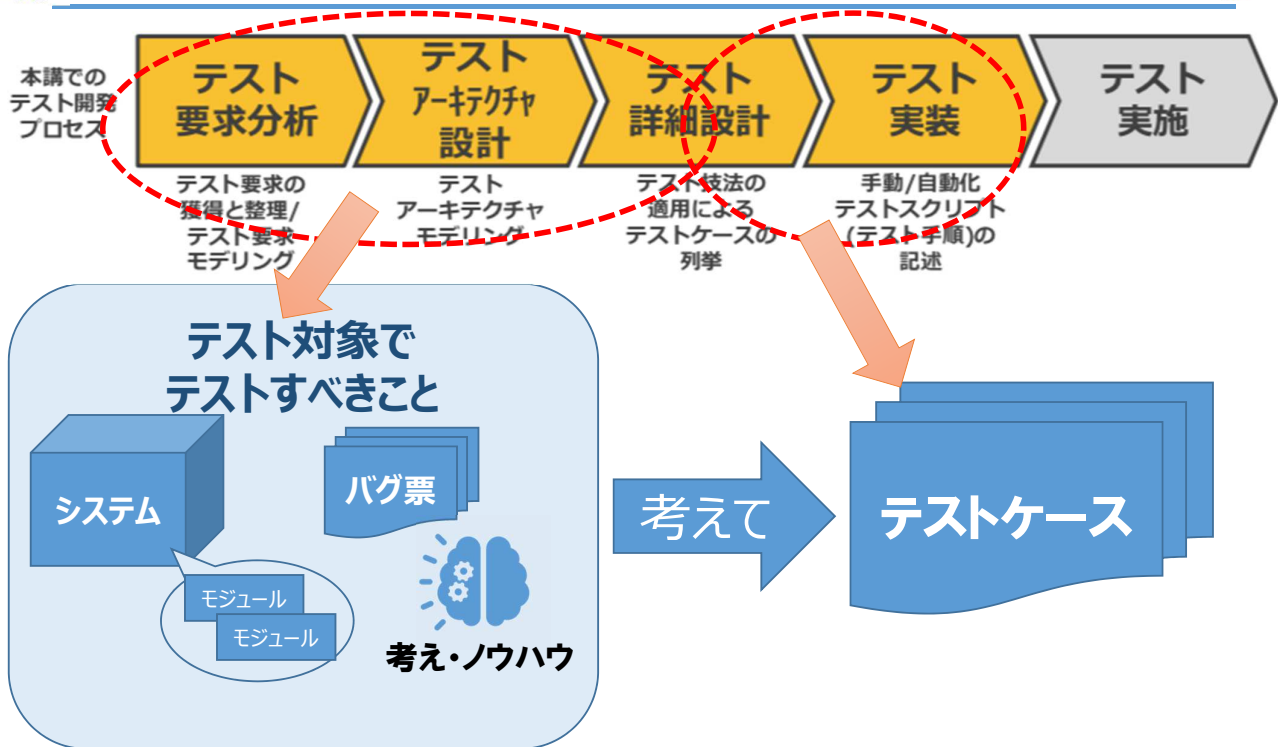
http://aster.or.jp/business/contest/doc/2019_U-30_V1.0.0%20.pdf

12

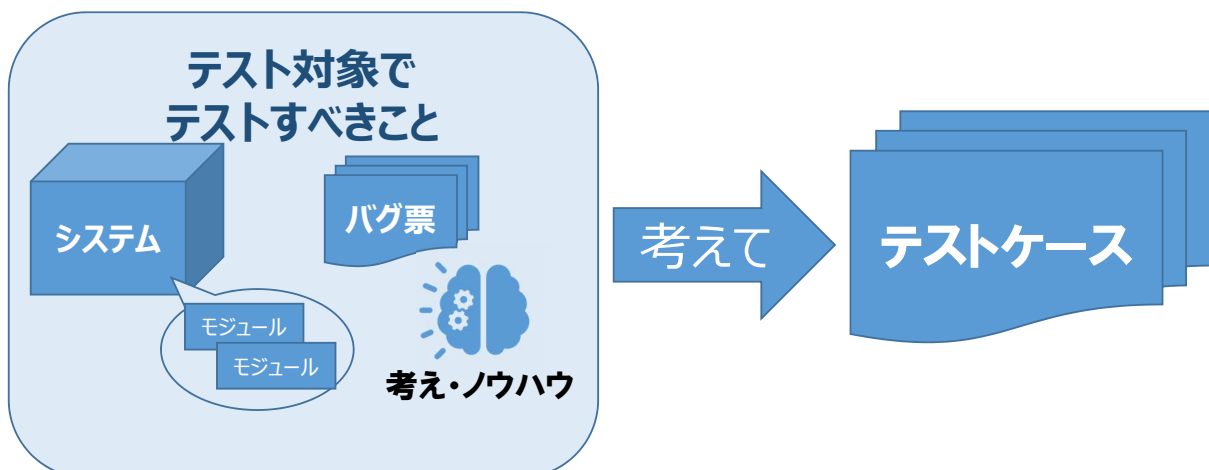
ASTER Association of Software Test Engineering



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いますよね？



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！



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
- 今あるテストを捨てるのは許されない
- どこから初めて良いかわからない





3. テストケースをキレイキレイしよう



- 今回はすでに存在しているテストケースなどからテスト設計をなおし
 - ムダを削減
 - ヌケ、モレを補強
- そのために使用する技術は

テストリバーースエンジニアリング

以後リバーース



こんな問題に効果があります。



- よくあるテストの問題
 - テストケースが重複する
 - テストケースがいっぱいありすぎて実行できない
 - テストケースのみが存在し、テストの意図がわからない
 - テストケースがなく、何もない
 - マニュアルから作れるがエラーケースが作れない
 - テストが終わったあとにバグが出る
 - 動かせばわかるでしょ？という箇所バグがでる
 - 何かとつなげたらバグが出る

今回、「解決 改善」を目指す箇所

さらに頑張ると「解決 改善」が見込める箇所



こんな問題に効果があります



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
- 今あるテストを捨てるのは許されない
- どこから初めて良いかわからない

今回の話は
ヒントになるかも

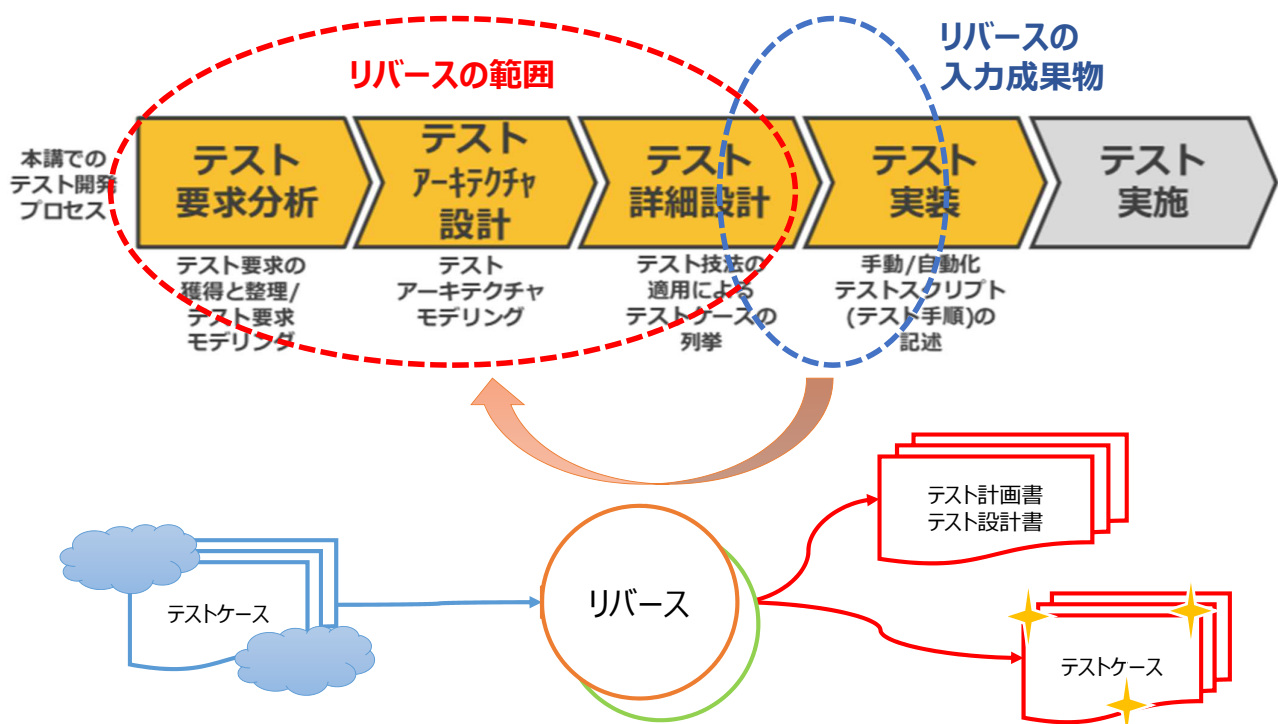


19

ASTER Association of Software Test Engineering



4. テストリバーズエンジニアリング(リバーズ)



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

20

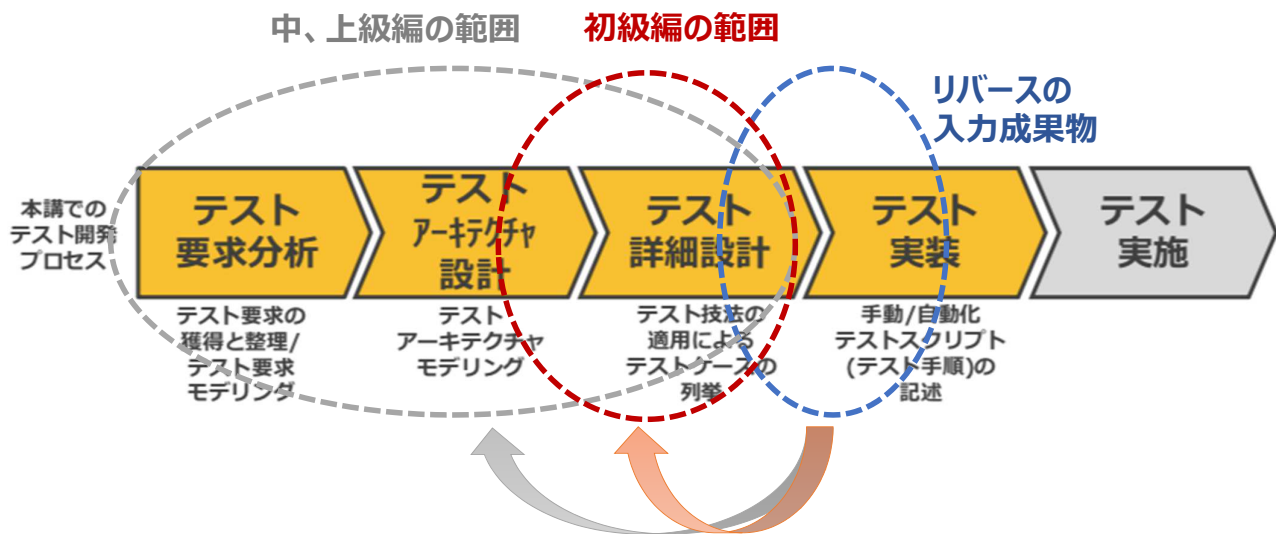
ASTER Association of Software Test Engineering



今回のスコープ



- 今回のリバー스는初級編
 - 中、上級については最後に少しご紹介します。



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

21

ASTER Association of Software Test Engineering



今回のスコープ



- 初級リバー스의目的
 - 目の前にあるテストケースの整理
 - 考える範囲を狭める
- 目標
 - テスト設計の一步目が踏み出せるテクニックを知る
 - やってみる気になる！



22

ASTER Association of Software Test Engineering

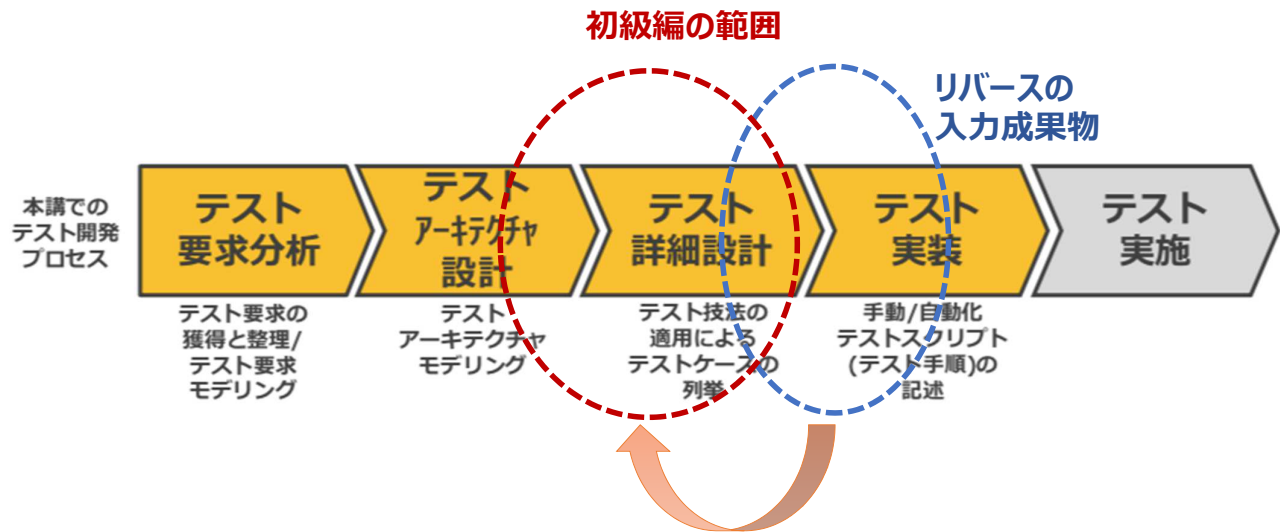


今回のスコープ



・前提

- ・ そもそもテストを行っている
- ・ こんなテストをしているというドキュメントがある



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

23

ASTER Association of Software Test Engineering



テスト設計の抽象的なプロセス



1. 方向づけ

- ・ 目的や基準、方針決め

2. 発散

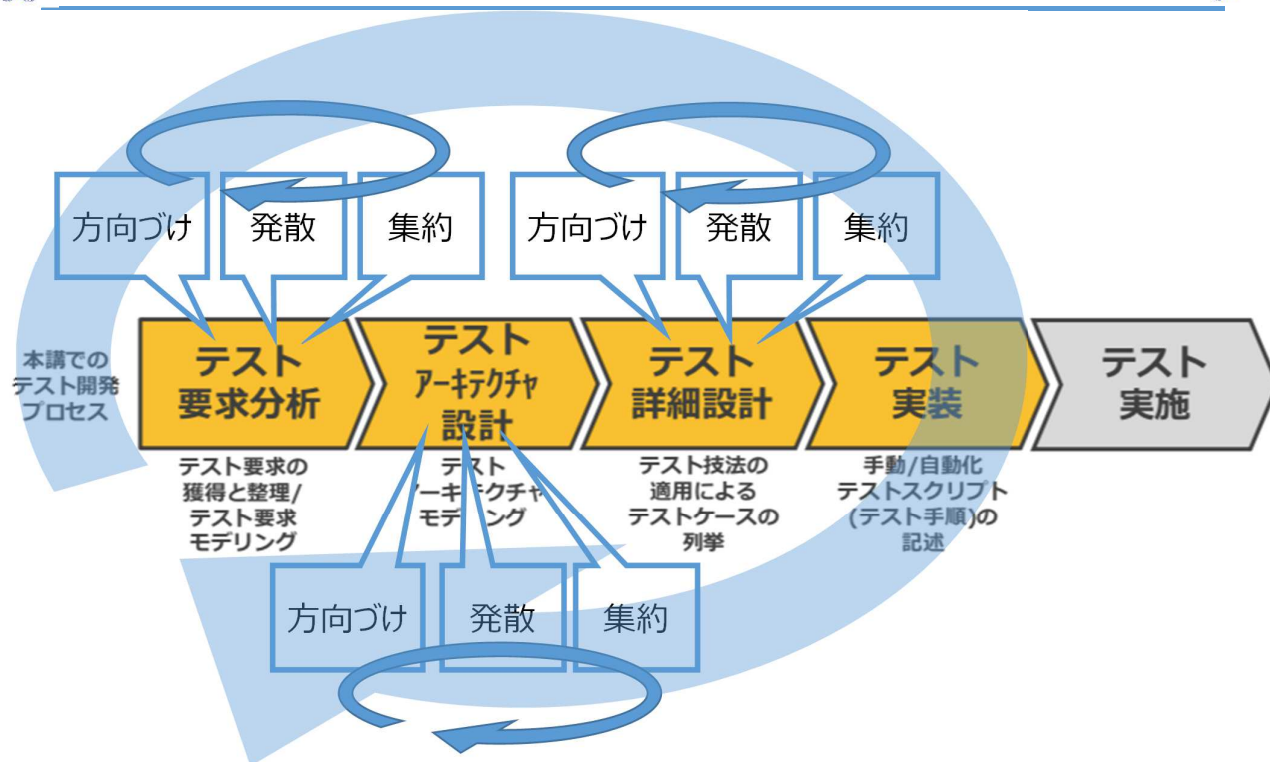
- ・ 決めた方向に沿って考えやワードを発散させる

3. 集約(再構築)

- ・ 発散させた物が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



テスト設計の抽象的なプロセス



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

25

ASTER Association of Software Test Engineering



初級リバースのプロセス



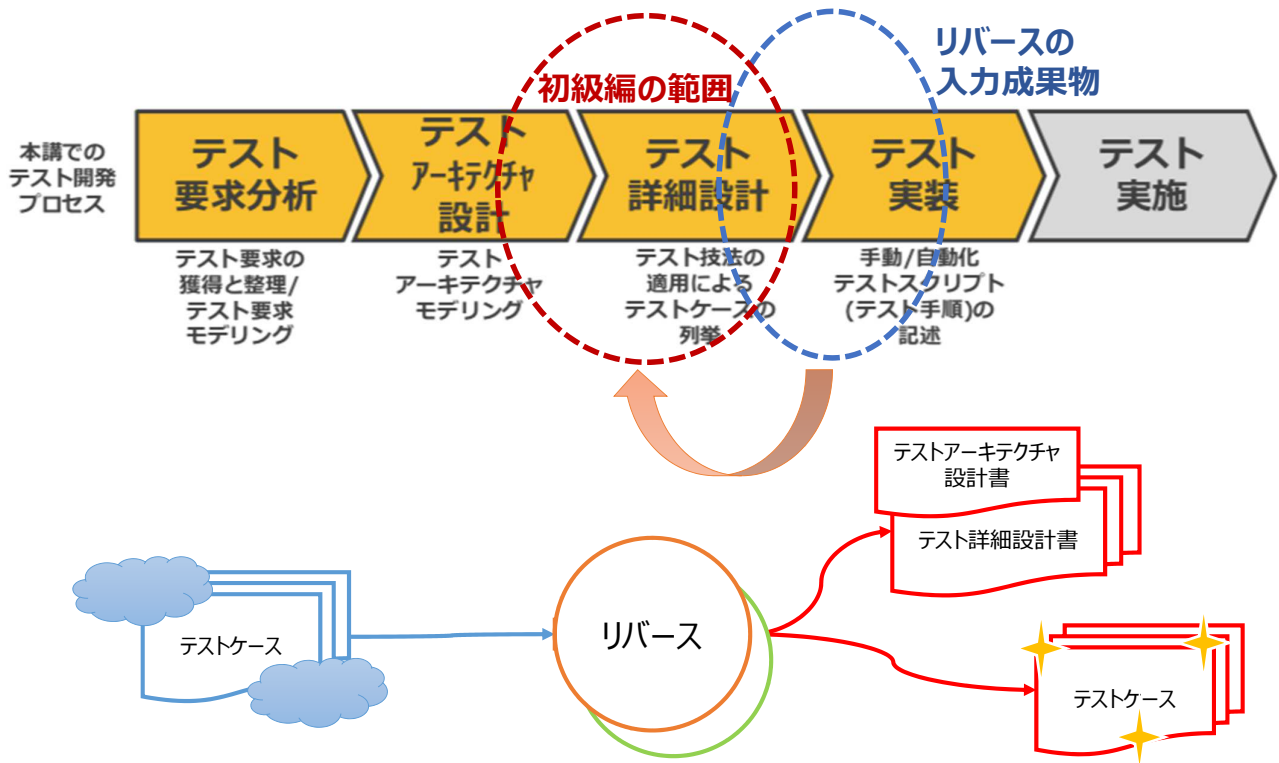
1. 方向づけ
 - 目的や基準、方針決め
2. 発散
 - 決めた方向に沿って考えやワークを分散させる
3. 集約(再構築)
 - 分散させた物があるルールに基づいてグルーピング
 - 粒度合わせ
 - 抽象表現の具体化
 - 具体表現の抽象化

整理されていないテストケースは
発散している状態とし、
ここからスタートする





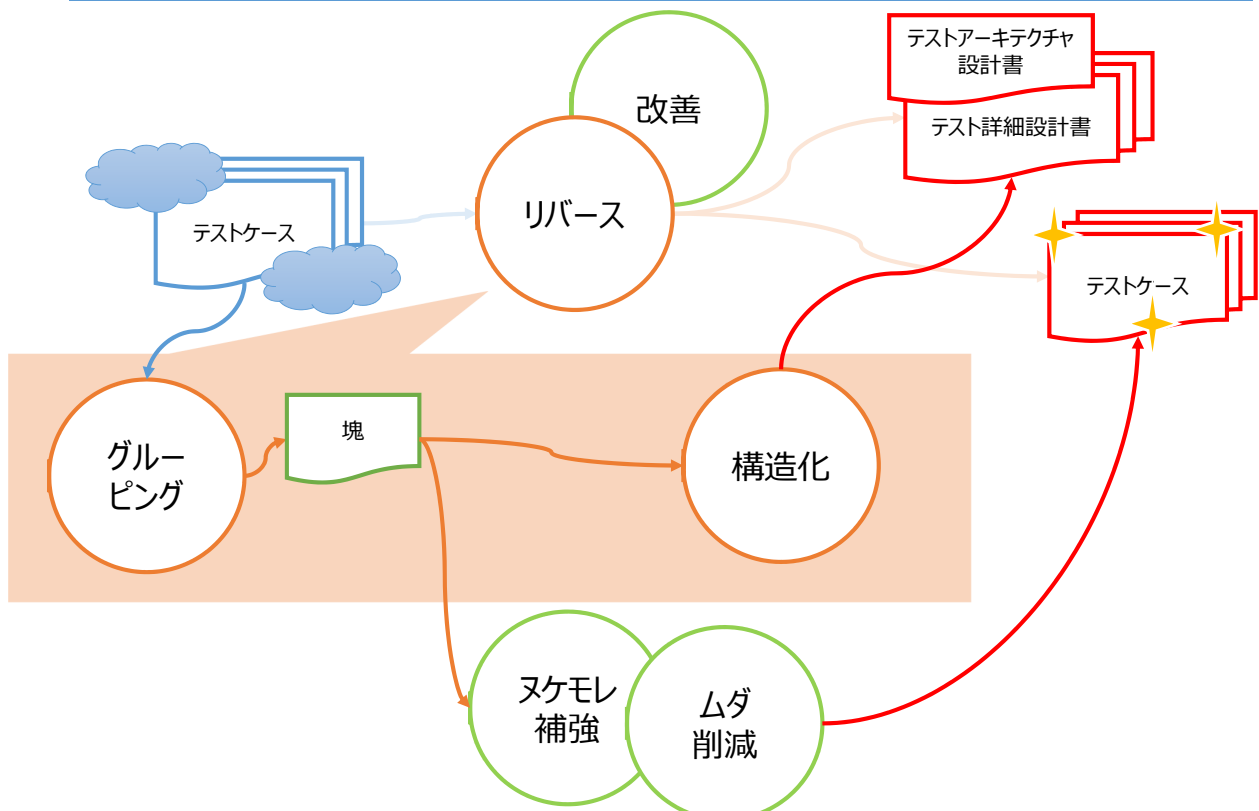
初級リバースのプロセス



27



初級リバースのプロセス



28



初級リバーズ



発散と集約を繰り返し、ヌケモレ・ムダを削減しやすくする

・ヌケモレ・ムダが削減するメカニズム

・グルーピングを行う

・ヌケモレ発見

- ・ スコープを小さくすることで考えが行き届く
- ・ 粒度を合わせることで広がりを見つけられる

ミニワーク1

ミニワーク2

・ムダ発見

- ・ テストパラメータと値の分解ができるようになる



ヌケモレ発見

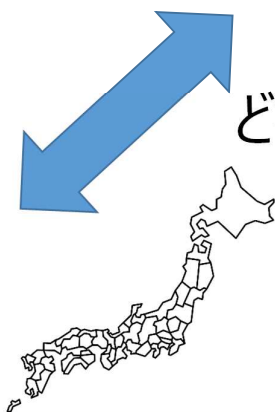


スコープを小さくすることで考えが行き届く

例：ヌケている都道府県はどこでしょうか？

都道府県名	
愛知県	島根県
青森県	千葉県
秋田県	東京都
石川県	栃木県
茨城県	鳥取県
岩手県	富山県
愛媛県	長崎県
大分県	長野県
大阪府	奈良県
岡山県	新潟県
沖縄県	兵庫県
香川県	広島県
鹿児島県	福井県
神奈川県	福岡県
岐阜県	福島県
京都府	北海道
熊本県	三重県
群馬県	宮城県
高知県	宮崎県
埼玉県	山形県
佐賀県	山口県
滋賀県	山梨県
静岡県	和歌山県

地方	都道府県名
北海道地方	北海道
東北地方	青森県 岩手県 宮城県 秋田県 山形県 福島県
関東地方	茨城県 栃木県 群馬県 埼玉県 千葉県 東京都 神奈川県
中部地方	新潟県 富山県 石川県 福井県 山梨県 長野県 岐阜県 静岡県 愛知県
近畿地方	三重県 滋賀県 京都府 大阪府 兵庫県 奈良県 和歌山県
中国地方	鳥取県 島根県 岡山県 広島県 山口県
四国地方	香川県 愛媛県 高知県
九州・沖縄地方	福岡県 佐賀県 長崎県 熊本県 大分県 宮崎県 鹿児島県 沖縄県



どちらが見つけやすいですか？



ヌケモレ発見



粒度を合わせることで広がりを見つけられる

例：状態

- 2 状態
 - 表、裏
 - ON、OFF
- 3 状態
 - 前、ちょうど、後

上記のようにセットを用意しておいてガイドワードとして使用することができる

テストでの例

- 10 秒以上経過後〇〇を押す があれば
- 10 秒未満のとき〇〇を押す というテストが考えられる

グルーピング化することで見つけやすくなる



ムダ発見



テストパラメータと値の分解ができるようになる

- テストパラメータ：条件
- 値：条件の取りうる値

例：検知温度 3 7. 5 度以上でお知らせしてくれる体温計

- 表示温度は 3 5. 0 度～4 2. 0 度
- お知らせを確認するテスト
 - テストパラメータ
 - 検知温度
 - 取りうる値
 - 35.0度、35.1度、35.2度、35.3度、35.4度
 - ...
 - 41.8度、41.9度、42.0度

いくつかのデータでテストしますか？





グループの見つけ方



- 機能を見つける
- 使う”人”を見つける
 - システムの状態を表しているものを見つける
- 使う”時”を見つける



5. テストリバーズエンジニアリングの実践ワーク



サンプル製品(券売機)を使用し、ワークを行っていきます

ワークの流れ

1. 実際のグルーピング例を説明します
 - 機能を見つける
 - 使う”人”を見つける
 - 使う”時”を見つける
2. 各自リバーズを行ってもらいます
3. リバーズした結果をグループで紹介します



製品説明！



・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
- ・ ワークでは販売周りを使用します
 - ・ 販売できるかテスト
 - ・ 販売ランプのテスト



機能をみつける



・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

- ・ 組み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

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。



1. 要求仕様書の作りかけの テストケースを分類してみよう

テストリバスエンジニアリング
ミニワーク1



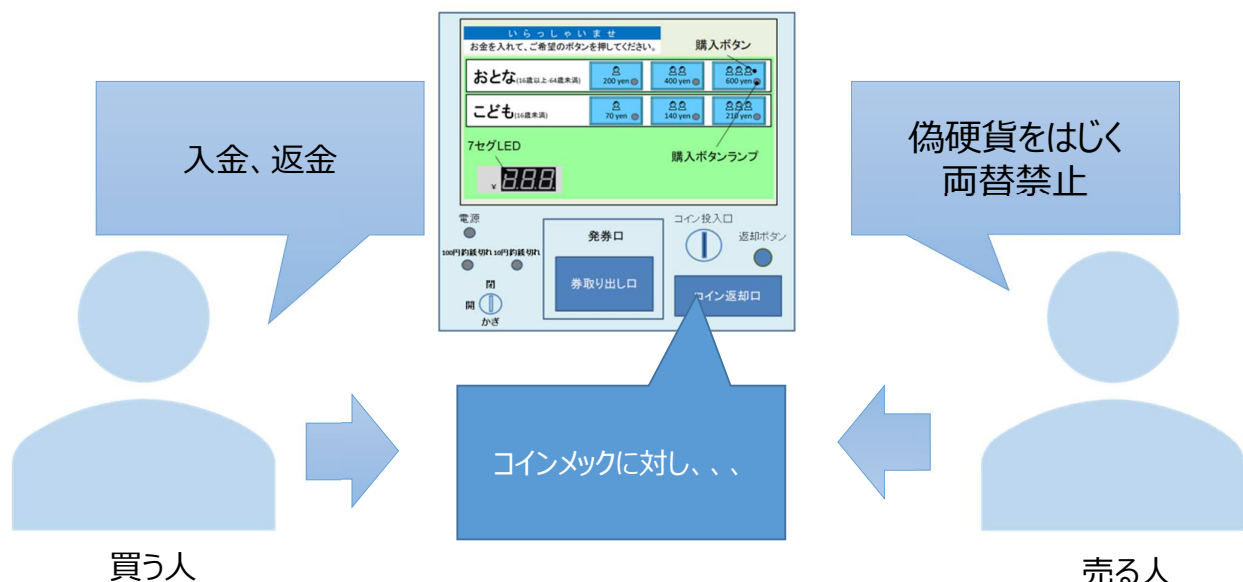
37



使う“人”を見つける



- 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分解する

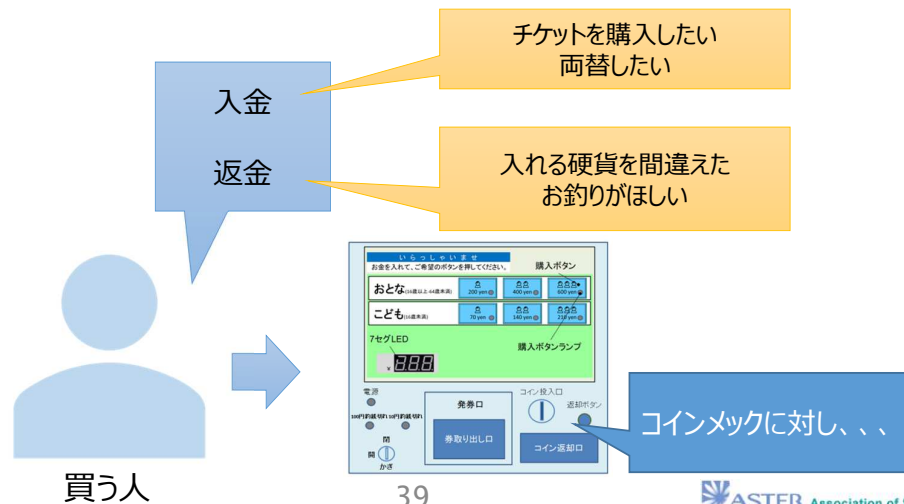




使う“時”を見つける



- 使う“シーン”ともいいます
- 製品の利用者は目的をもってその製品を利用します
- 先程の使う人と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



39

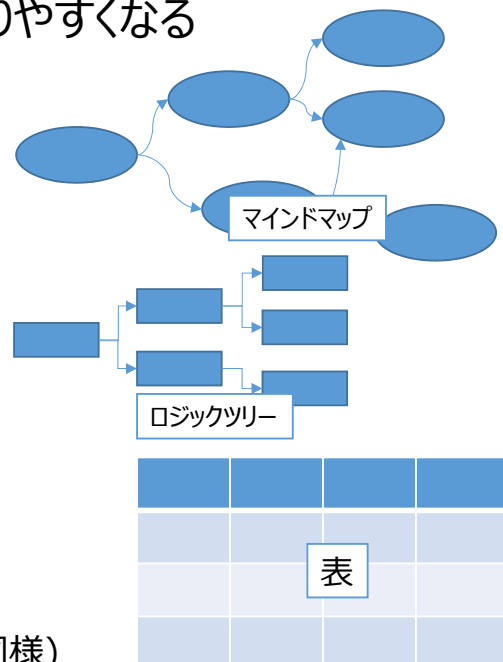
ASTER Association of Software Test Engineering



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る
 - 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(ミニワーク1と同様)
 - 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る
 - ミニワーク2で考えます





2. テストケースの粒度を 合わせてみよう

テストリバーエンジニアリング
ミニワーク2



41



とここで休憩です



42



さらなるレベルアップのために



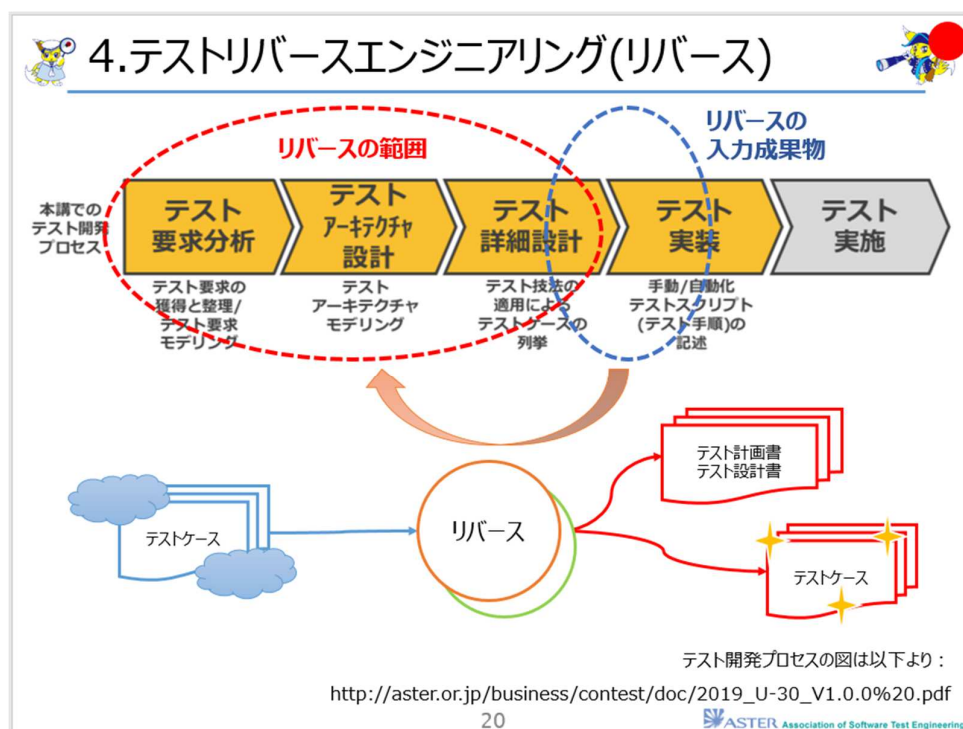
43



6.さらなるレベルアップのために(中、上級リバース)



テストケースを見直しテスト要求を見つける





テストで何をしたいかをリバーズしよう



・テスト要求分析チュートリアルより

テスト要求分析の役割

- ・テスト要求分析結果は、他のプロセスの根拠や背景になる
 - ・テスト要求分析の成果物は他の成果物と整合するべき
 - ・他のプロセスと行き来しフィードバックを取り込みながら進める

意図的に説明できるようにしておかないと、なかなかできない

なぜそう考えたか？説明できステークホルダーと合意できる内容であること

ASTER Association of Software Test Engineering

・今あるテストケースで厚い箇所、薄い箇所を見つけて理由を明確にする

- ・ 厚いには厚い理由があったりします
 - ・ 重要なステークホルダーに影響がある
 - ・ 製品としてメインの機能である
 - ・ ソフトの仕組み上不具合が起きやすい
 - ・ 過去不具合があった

テスト要求分析チュートリアル資料は以下より：

http://aster.or.jp/business/contest/doc/2020_TRA_1-3_V1.0.0.pdf

45

ASTER Association of Software Test Engineering



テスト要求分析が気になる方へ



・テスト設計コンテストではテスト要求分析チュートリアルを行っています

・Youtubeにもありますので気になる方は御覧ください

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



46

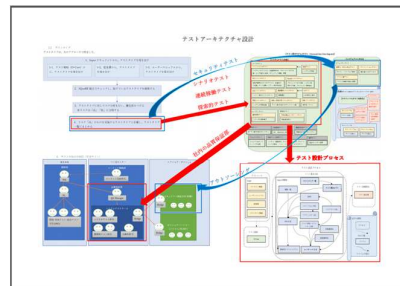
ASTER Association of Software Test Engineering



テストアーキテクチャ



- テストを構造化しよう！
 - テスト要求からどんなテストを行うのか見えるようにしよう
 - 意識する点
 - スコープ
 - 俯瞰
 - 目論見
 - テスト要求分析とのトレーサビリティ
 - どの塊でどこをどこまでテストするのか
 - どの順でテストするのか
- はじめはなんとなくで良い
 - アーキテクチャの例
 - どんな機能からテストするのかリバーシした塊を並べる
 - 並べた塊に重要度を紐付け塊の大きさを変える



テスト設計コンテスト2020 決勝
出席番号となり同士 抜粋
http://aster.or.jp/business/contest/contest2020/pdf_finals/exhibition_Tonaridoshi.pdf

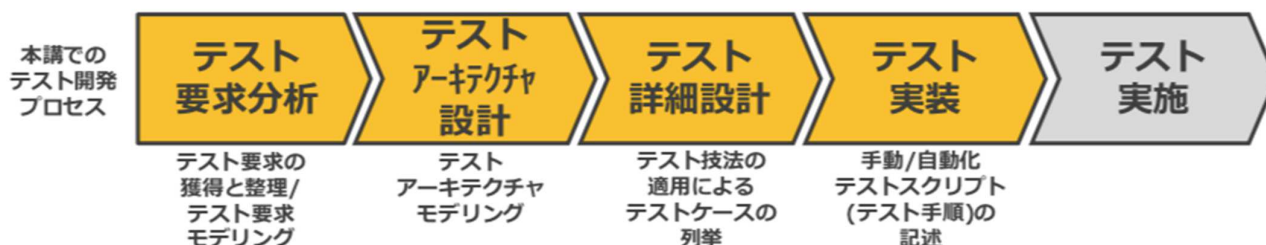
- 今日の知識でできますね！！！！



リバーシで気をつける点



- 何が目的なのか見失わないようにする
 - 現状そのままをリバーシしたいか
 - 本来やるべきテストをみつけない



- 本来やるべきテストをみつけない場合
 - 詳細設計→要求分析→アーキテクチャ→詳細設計
 - これらをぐるぐる繰り返すことが重要
 - 順番にリバーシすると現状の動きを優先してしまう
 - リバーシをしたものがすべてとなってしまう



- 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・・・



- 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と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！！





テスト設計コンテストについて



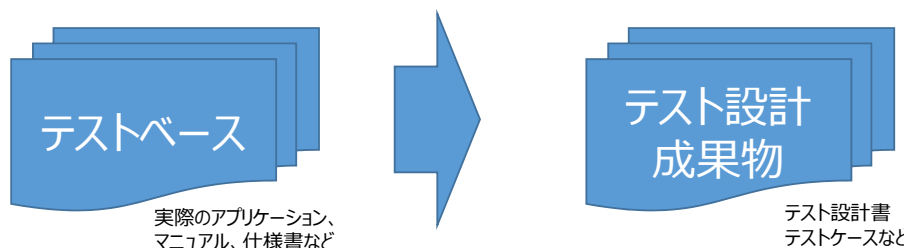
51



7. 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 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】T F C K A・R I・Y A（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】わんだーず（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



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

- 参照

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



- 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



ラスト Q&A





ご清聴ありがとうございました。