

テスト、うまくいっていますか？

Test Engineer's Forum 北海道 (TEF道)

安達 賢二・中嶋 信

“ソフトウェアテスト”を頭の中で イメージしてみてください



のちほどこのことを伺うタイミングがありますのでイメージを覚えておいてください。

講演概要

- みなさんのテストはうまくいっていますか？
- もし、
 - レビュー時間が長く、ダメ出しが多くて気が滅入る
 - 開発遅延のしわ寄せでテストの着手が遅れてテスト期間が短くなる
 - 開発ドキュメントをなぞるだけのテストになっている
 - リリース後のバグ発生への不安からやみくもにテストを増量してしまう
 - テストが終わったはずなのに思わぬ事象が多発する
 - 障害発生時の原因分析と再テストに必要以上の時間がかかる

などの場合はこの講演が状況打開のヒントになるかもしれません。

- 今回は、われわれの“ソフトウェアテストへのアプローチ”体験を交えつつ、**テストの難しさや楽しさ、奥深さ、ノウハウを得るための情報源、段階の登り方**などを共有します。

安達 賢二 (あだち けんじ) adachi@hba.co.jp

TEF北海道お世話係

株式会社HBA 経営企画本部 Executive Expert

<http://www.softwarequasol.com/>

株式会社Levii 共創ファシリテーター

<https://levii.co.jp/about/>



きたのしろくま
@kitanosirokuma

多感な時期を“函館”
で過ごしました

【経歴】

税金管理システム開発保守運用／金融系システム更改プロジェクトPMer／全社品質保証責任者などを経験
2012年社内イントレプレナー第一号事業者として品質向上支援事業を立ち上げ。

自律運営チーム構築・変革メソッドSaPIDをベースに、

関係者と一緒に価値あるコトを創る共創ファシリテーター／自律組織・人財育成コーチとして活動中。

【社外活動】

NPO法人 ソフトウェアテスト技術振興協会 (ASTER) 理事

JSTQB (テスト技術者資格認定) 技術委員

JaSST (ソフトウェアテストシンポジウム) 北海道

2006-2009実行委員長 2010-2018実行委員 2019～2022サポーター

JaSST-Review (ソフトウェアレビューシンポジウム) 実行委員

JaSST-nanoお世話係

ASTER研究事業「レビュー体系化」メンバー

テスト設計コンテスト本部審査委員(2015-2017)

SEA (ソフトウェア技術者協会) 幹事・北海道支部立ち上げメンバー

SS (ソフトウェア・シンポジウム) プログラム委員

第33-39期SQiP研究会レビュー分科会アドバイザー

SQuBOK_Ver3プロセス改善研究Grリーダー (with プロセス改善の黒歴史研究)

TOCFE北海道幽霊メンバー など

小2～中2：函館湯川町
中2～高3：七飯町
・湯川小学校卒
・中1～2：湯川中学校
・中2～3：七飯中学校卒
・高1～3：函館東高校卒



Test Engineer's Forum 北海道

(TEF道)

ソフトウェアのテストを中心として、ソフトウェアテストの勉強会、ワークショップの開催にとどまらず、良いソフトを作るための技術やノウハウの共有、ときには研究っぽいことをやってみて論文提出など、いろいろやっています。

#tefdo



How do you like
software test?

TEF道

Do the best it yourself
... de Show!

中嶋 信

TEF道

ソフトウェアテストシンポジウム(JaSST)北海道
某半導体製造装置メーカーのソフトウェア部門

社外活動で得たテストに関する知見を活かしながら、
開発部門と品質保証部門を行ったり来たりしている。
社内の成果を恩返しのため、
たまに事例発表なども、してたり、してなかったり。

【社外活動経歴】

- 2006~ テストコミュニティのTEF道に所属
- 2011~ ソフトウェアテストシンポジウム(JaSST)北海道 実行委員
- 2013-2016、2017 テスコンに参加



中山 信

中嶋 信

〔なかくき まこと〕

たまたに

中山 由信

中山 由信

〔なかやま よしのぶ〕

中山 由信

〔なかやま よしのぶ〕



じゃねえよ 

間違えないでね



前置き

テストコードとか
Unit Testとか
TDDとか
の話だと思った人

そんな話はしません

本発表は
ひたすら情報を投げます

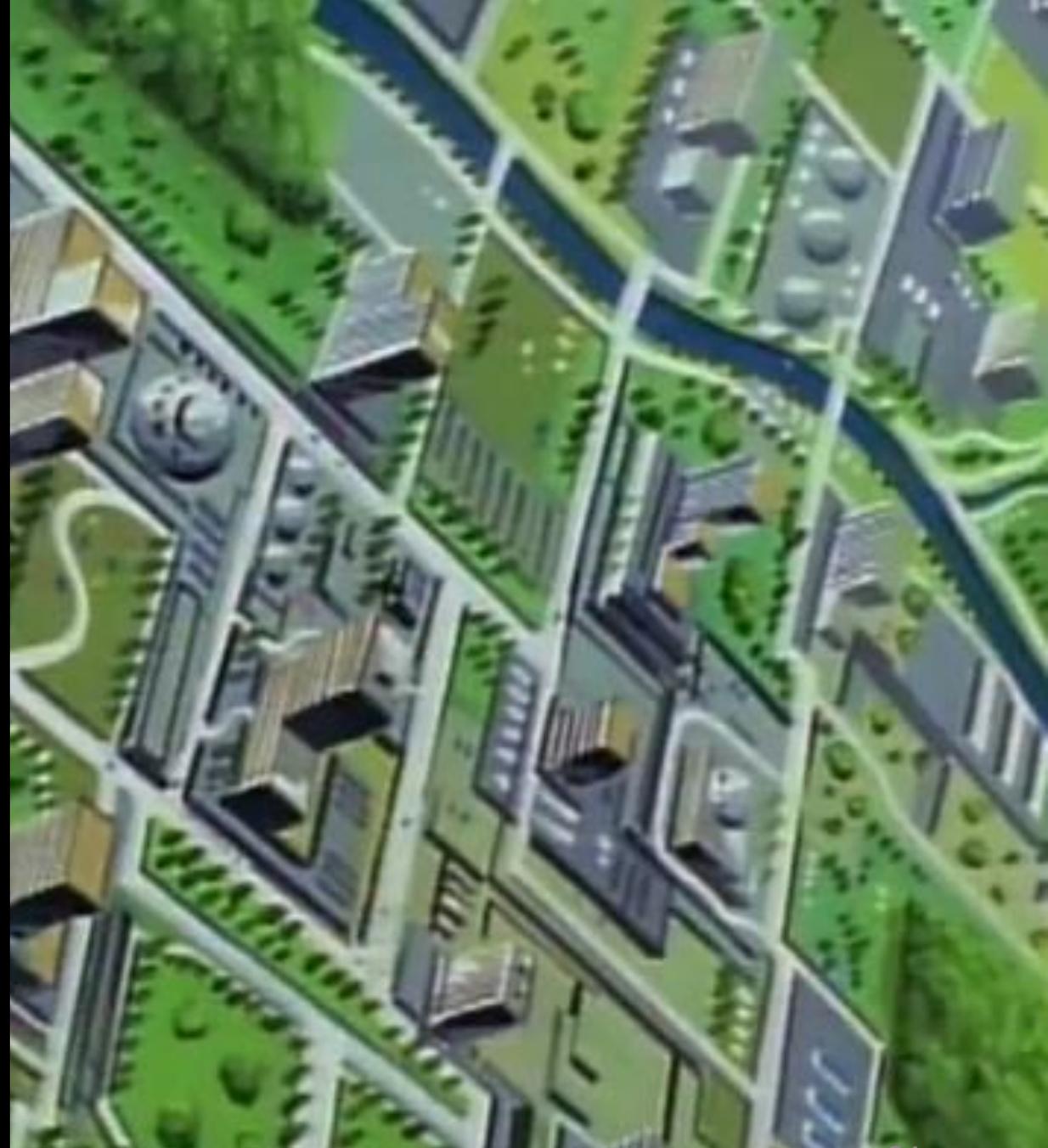


適宜拾ってください
あしからず



昨今のソフトウェア事情とテスト

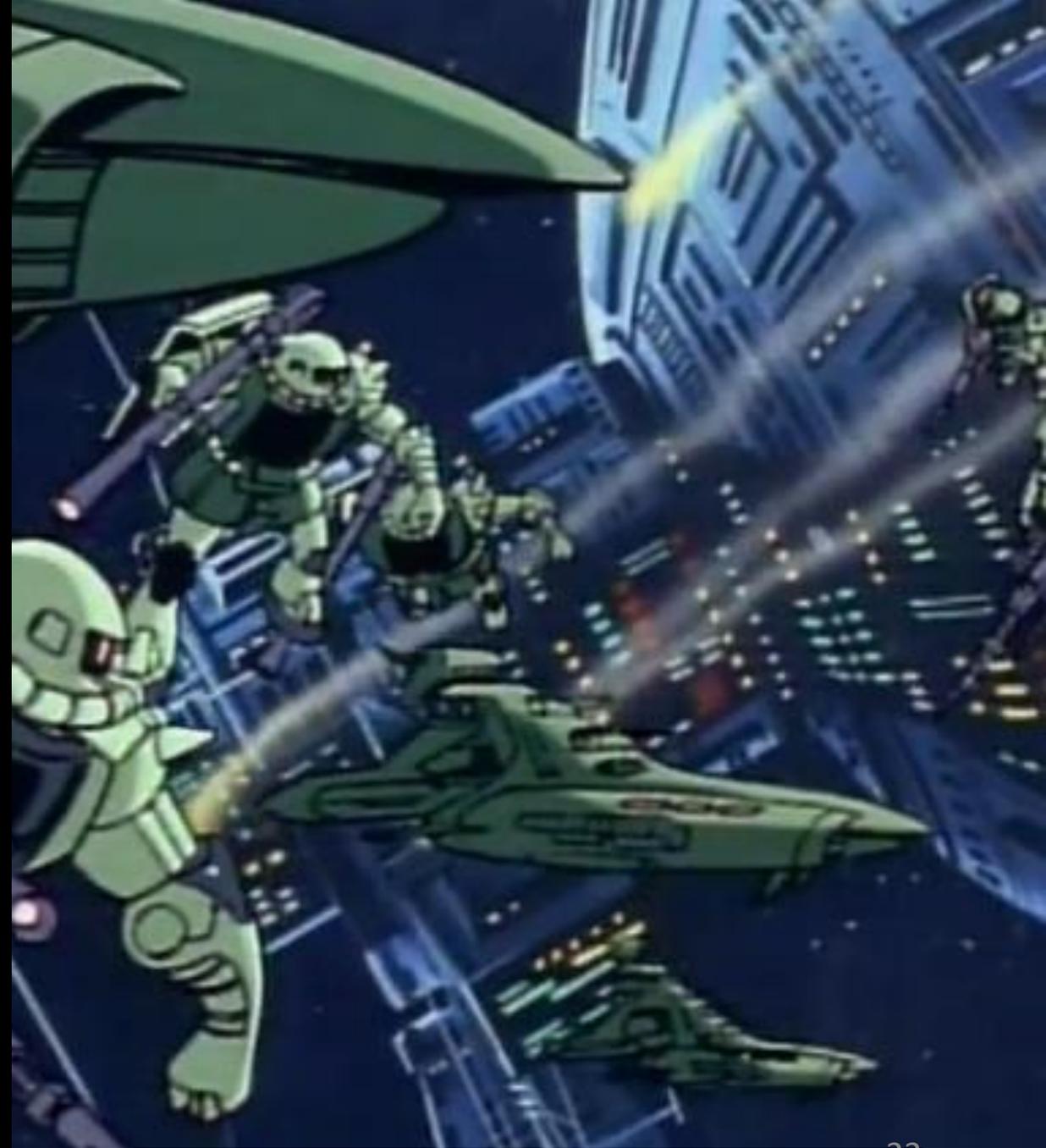
近年、
ソフトウェアは大規模になり、
工業製品の重要な要素となっている…。



近年、

ソフトウェアは大規模になり、
工業製品の重要な要素となっている…。

ソフトウェアシステムが成長するにつれて、
品質保証のためのソフトウェアテストも
また巨大で複雑になっていく…。

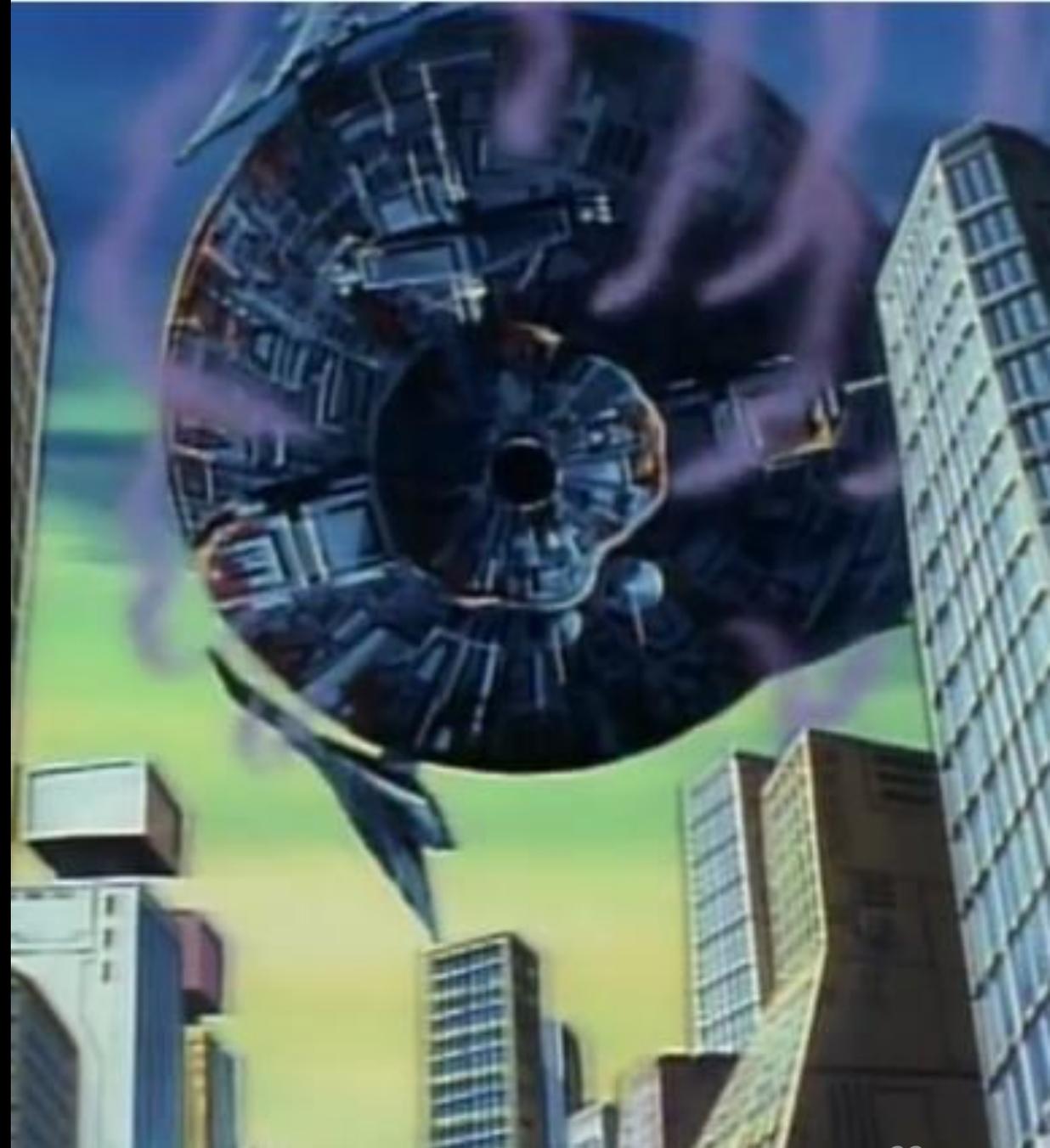


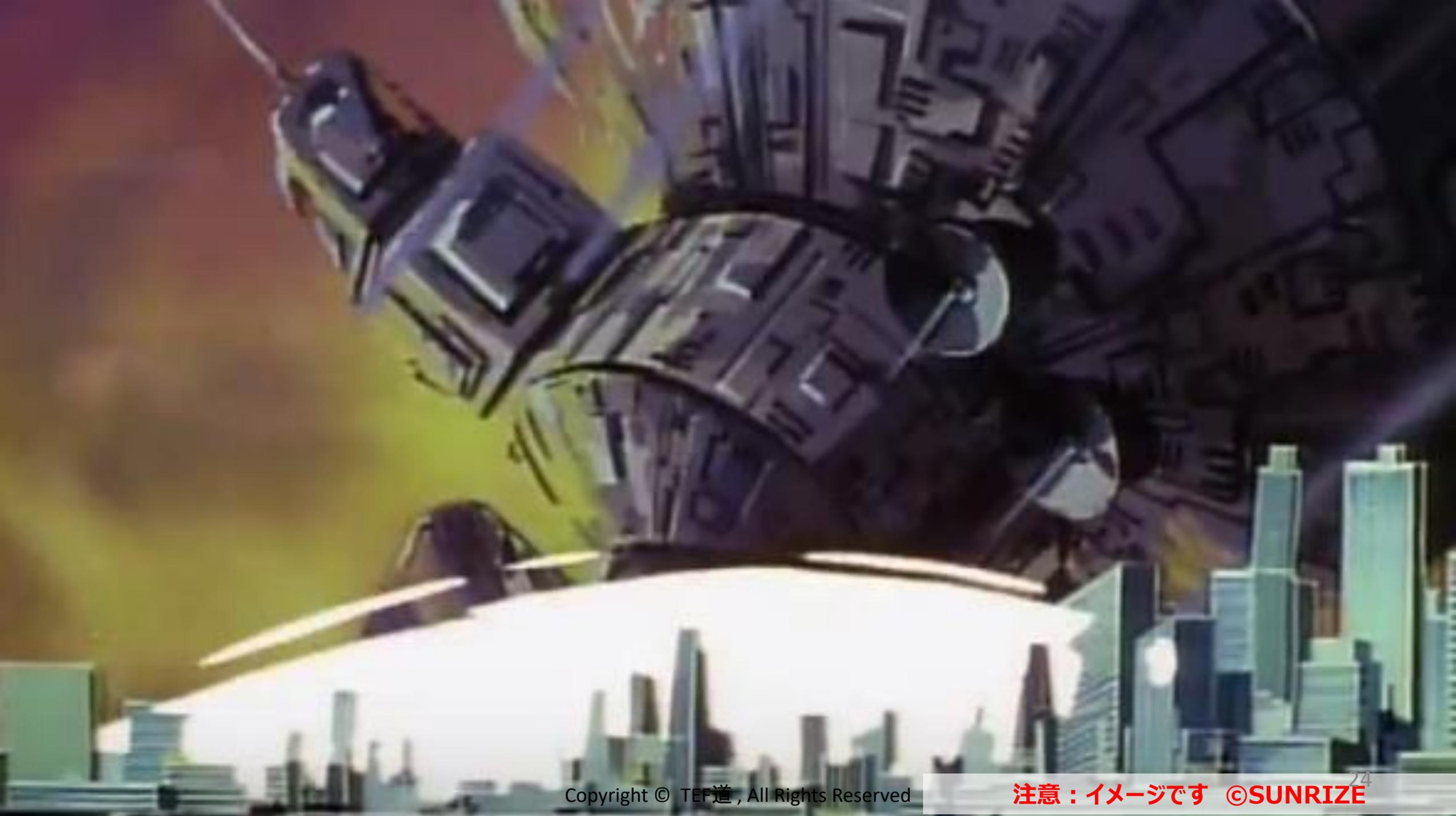
近年、

ソフトウェアは大規模になり、
工業製品の重要な要素となっている…。

ソフトウェアシステムが成長するにつれて、
品質保証のためのソフトウェアテストも
また巨大で複雑になっていく…。

1つのプロジェクトには
数百万ものテストケースが存在し、
多くのテストタイプとテストレベルを
網羅しているため
非常に複雑化している…。





たとえば

JCOM株誤発注のトラブル

みずほ証券が61万円で1株売るつもりで1円で61万株の売りを発注した単純ミス。担当者が気づいたが、取り消しができなかった。

原因

売買注文の処理が優先されるようになっていた。そのため、売買注文が入ると取り消し処理が受け付けられなくなる。



被害総額は400億円

もう一つ

NASAのトラブル

1998に開発された火星探査機マーズ・クライメットオービターの事故。火星の軌道に乗れず破壊されてしまったというもの。



原因

メートルとヤードを混用していたため、
距離の計算間違いが発生。

要因

以前はやっていた以下のことを予算削減でやめてしまった

- データの流れを最初から最後まで通した試験を行う
- 計算された結果とは別に計算を行う
- 2つの結果をチェックする

The background of the slide features several US dollar bills scattered across the frame. The bills are slightly out of focus, creating a sense of motion or a large quantity of money. The colors are primarily the green and tan of the US dollar bills.

被害総額は1億2500万ドル

こういった問題を未然に防がねば



テストって
大事ですよね...

テストって？

ユニットテストとは？

テストコードで書いたテスト？

メソッドの動作を確認するテスト？

コンポーネントの動作を確認するテスト？

機能 (Function) を確認するテスト？

ユニットテストとは？

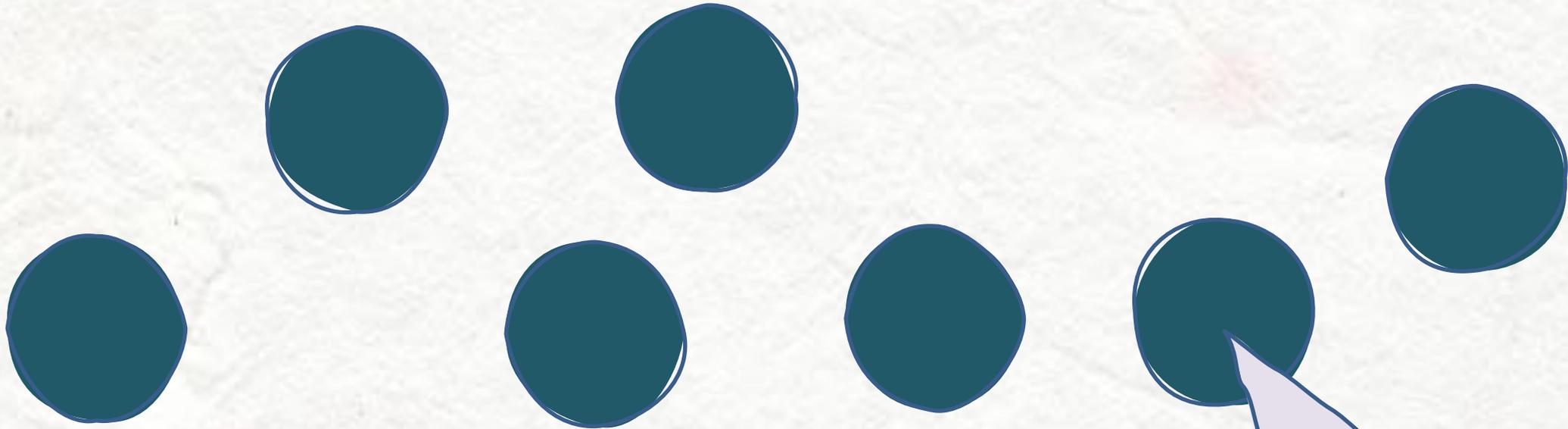
個々のハードウェアまたはソフトウェアコンポーネントに焦点を当てたテストレベル。

同義語：プログラムテスト（program testing）、コンポーネントテスト（component testing）、モジュールテスト（module testing）

ソフトウェアのイメージ

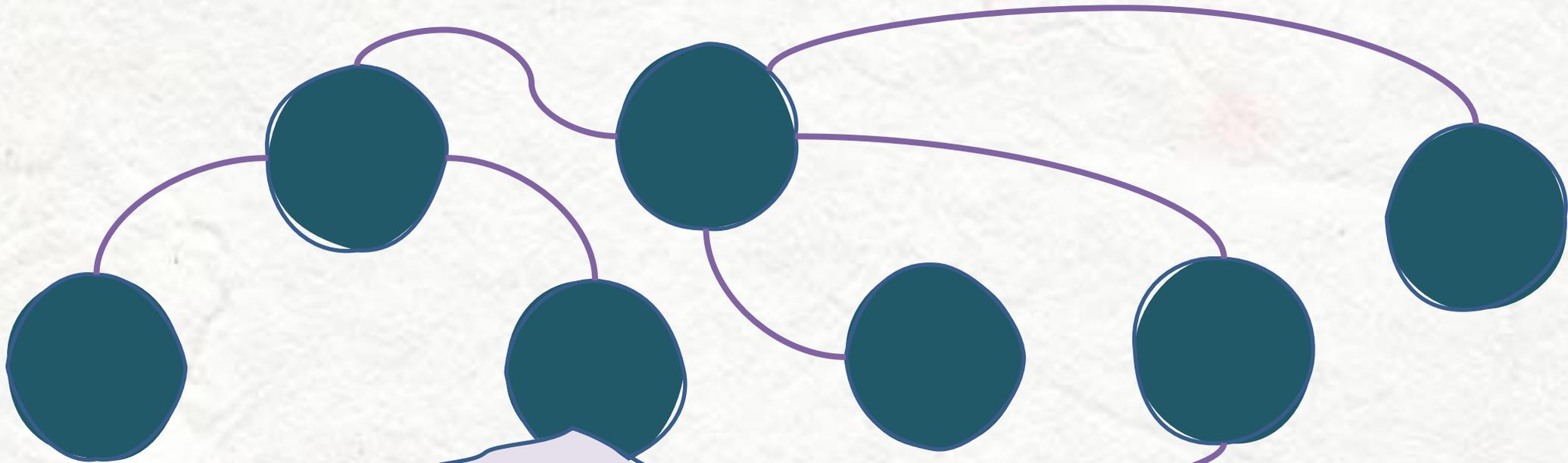


ソフトウェアのイメージ



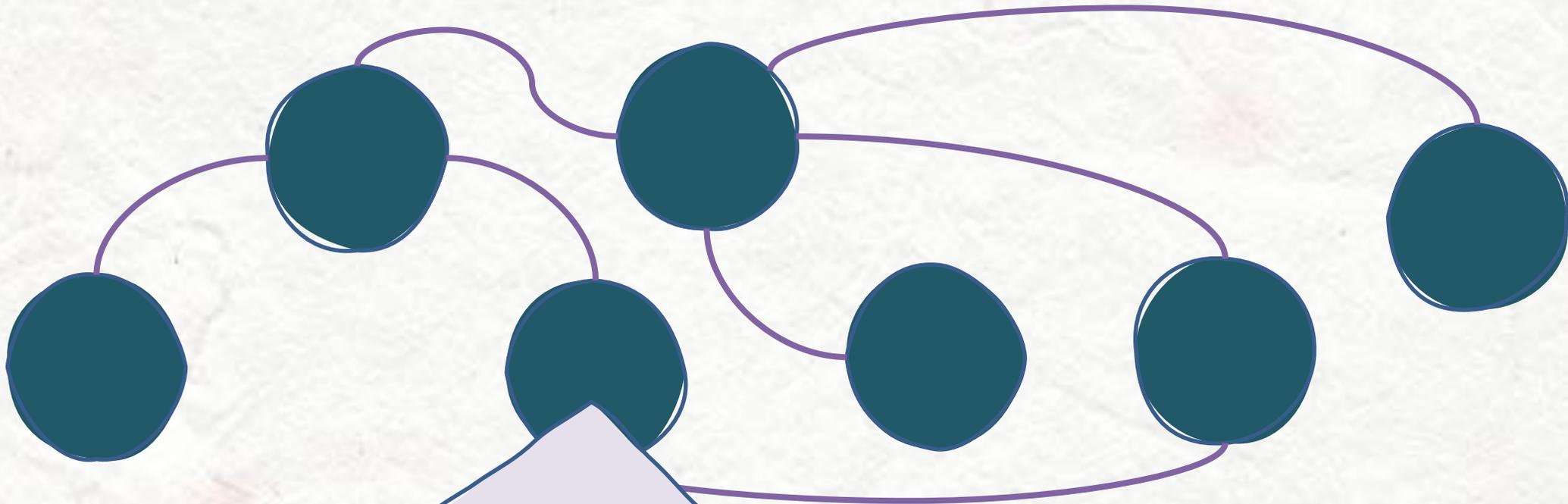
たくさんあ
る

ソフトウェアのイメージ



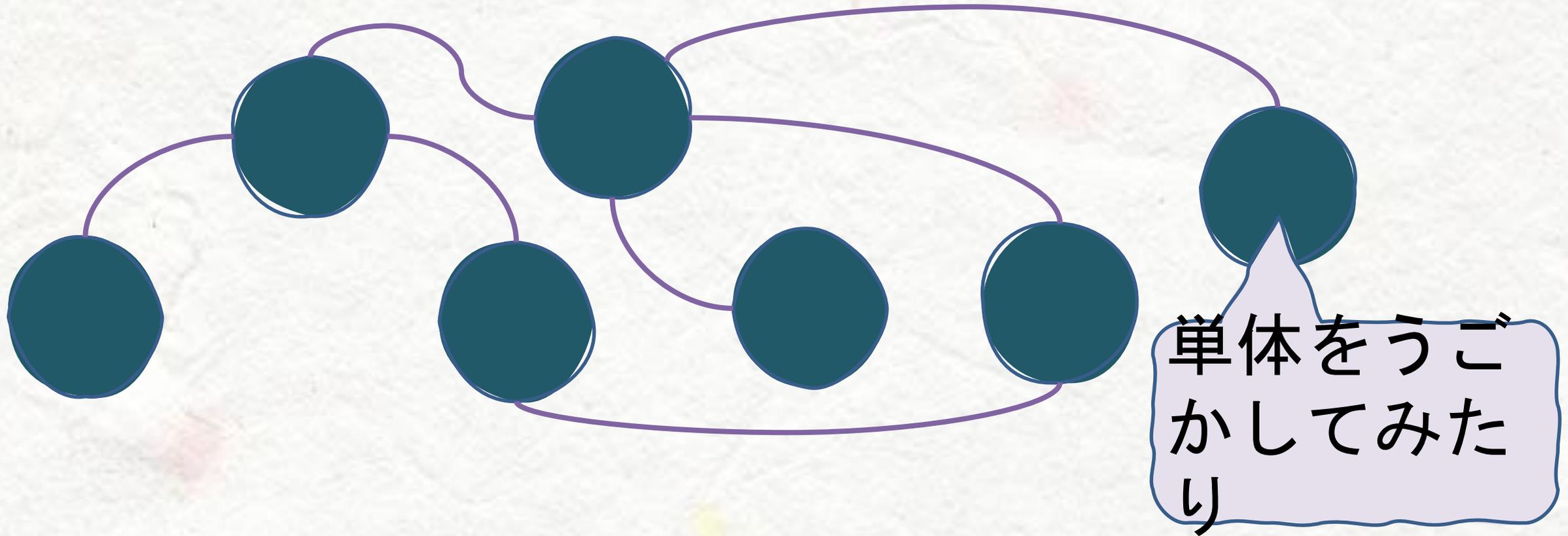
コンポーネントとか、ライブラリとか、よくわからんモノとか、集まってつながって動いていますよね？

ソフトウェアのイメージ

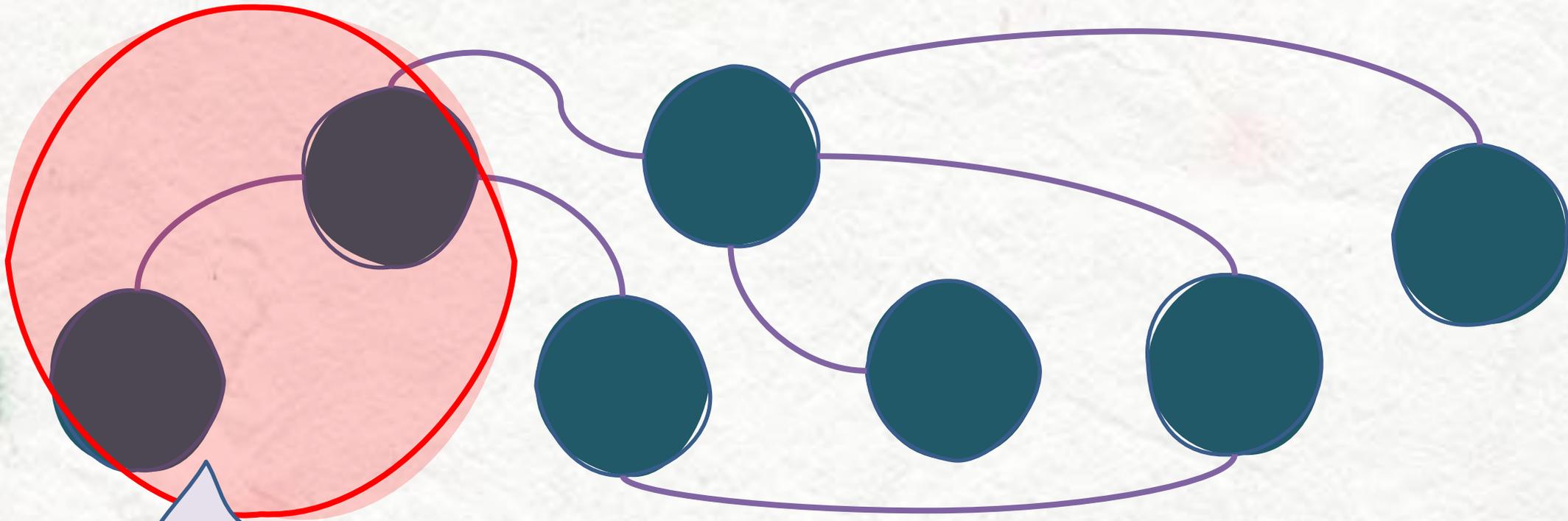


ユニットテストでテストしているのは、だいたいこのサイズ

ソフトウェアのテストって

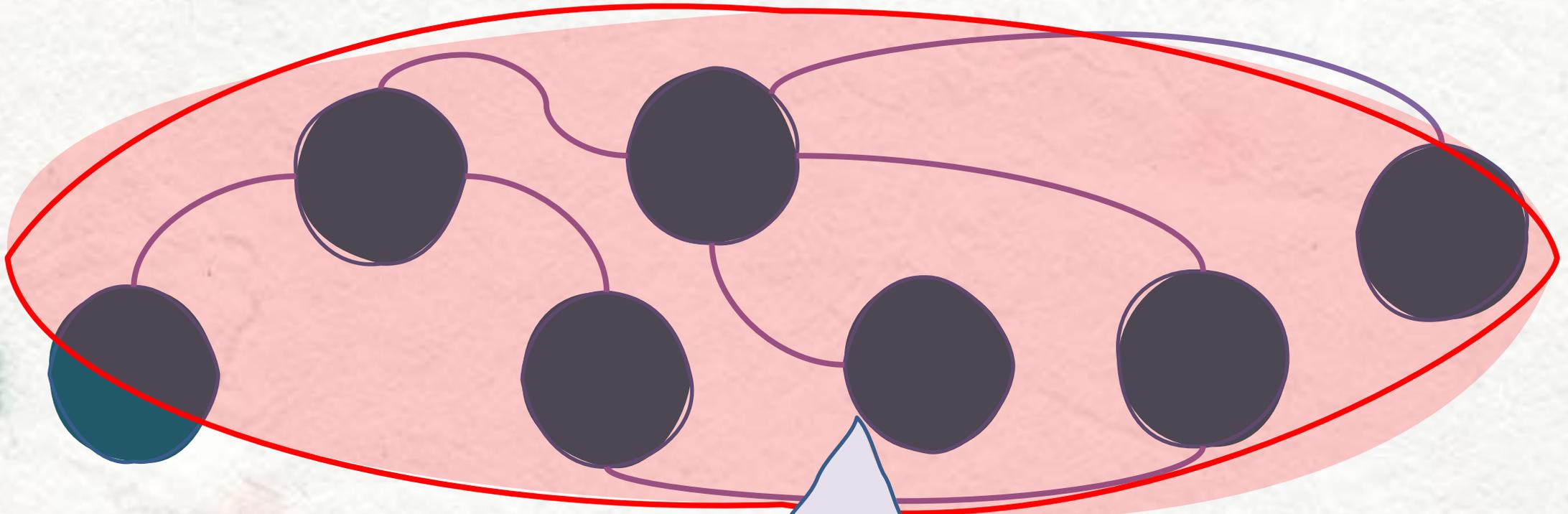


ソフトウェアのテストって



部分的にくっつけて動かしてみたり

ソフトウェアのテストって



全部まとめて
動かしてみたり

ソフトウェアのテストって

テストする範囲は
思っている以上に広い

部分的にくっつ
けて動かしてみ
たり

全部まとめて
動かしてみた
り

単体をうご
かしてみた
り

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。

ソフトウェア開発のライフサイクル

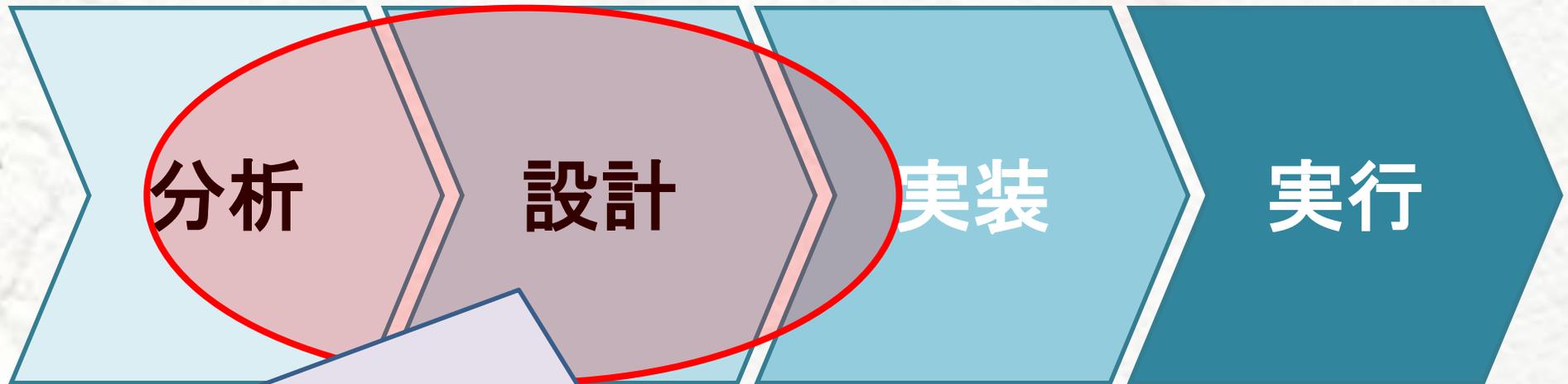


「テスト」とか言いますが、準備から実施までひとくくりになっています。ちょっとズームインしてみますね。

テストのライフサイクル



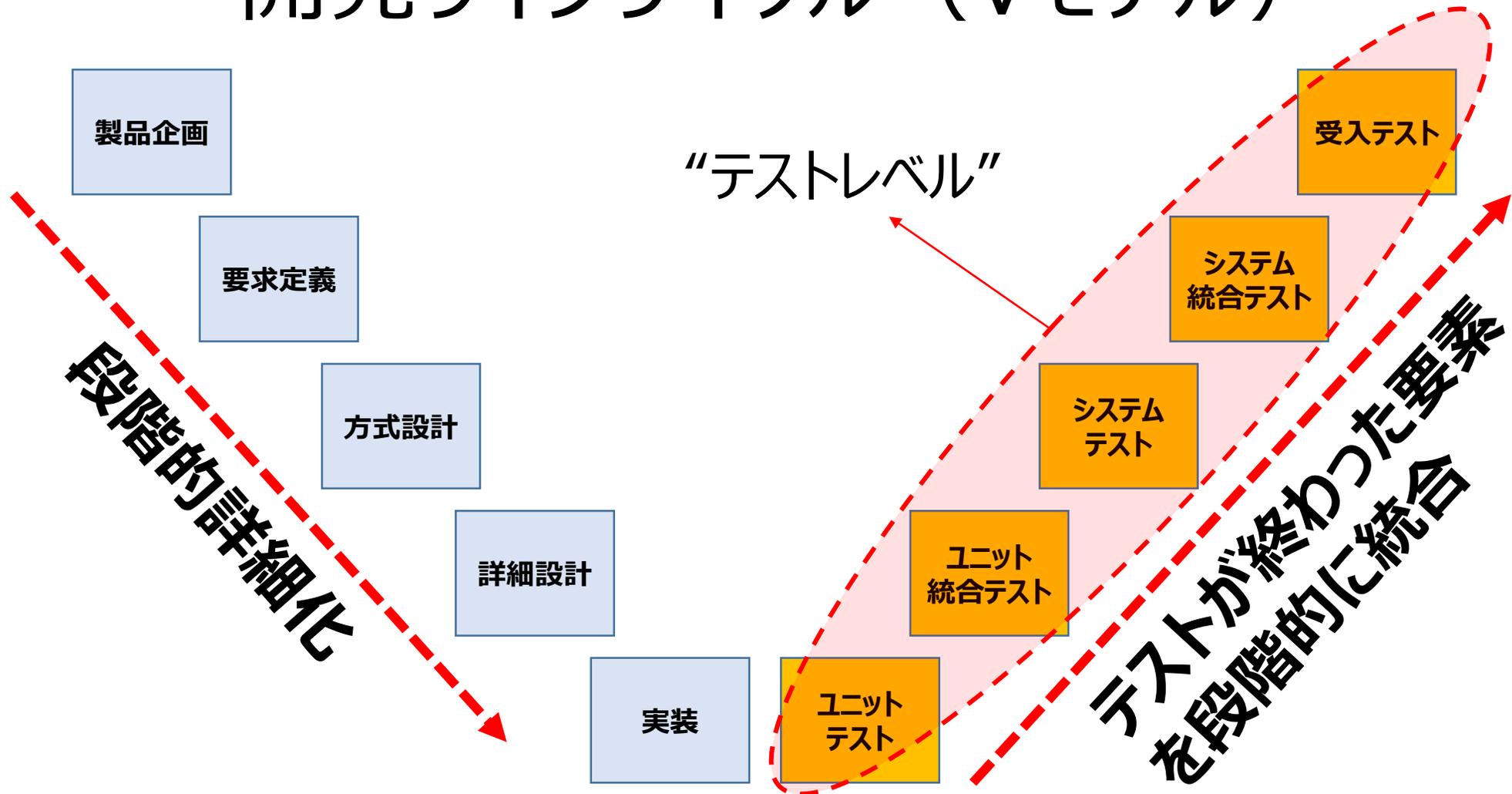
テストのライフサイクル



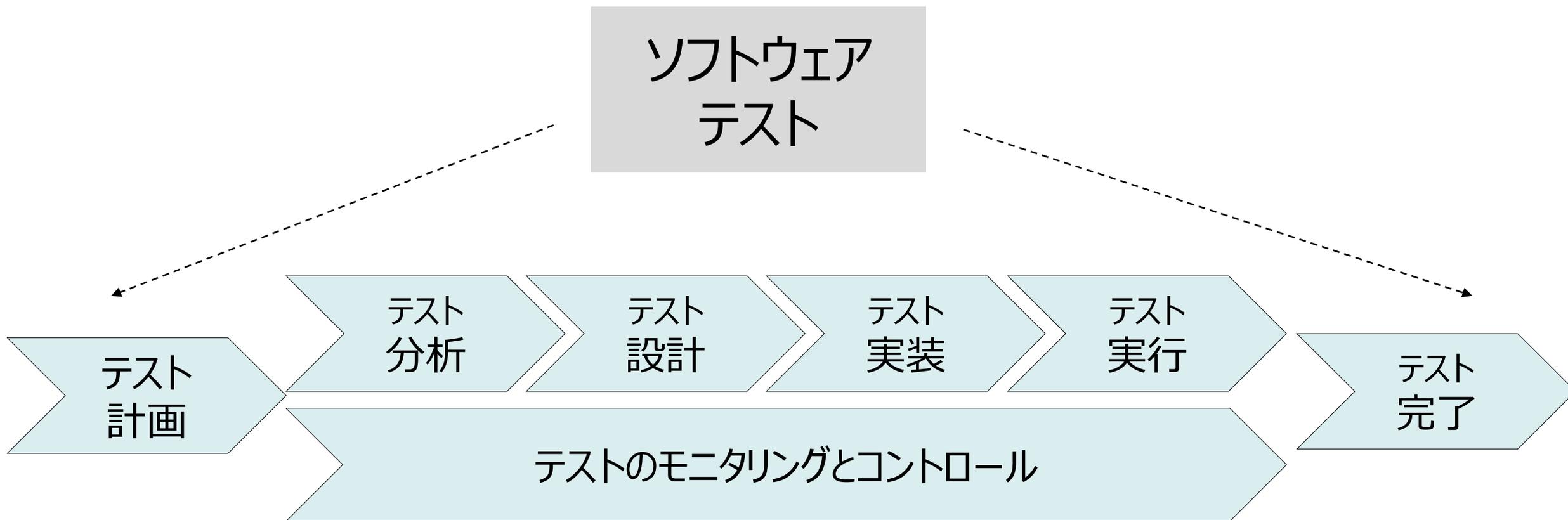
テストの設計とその前後では、「テスト設計技法」と言われる技術が（無意識化もしれないけど）使われます。

ソフトウェア開発とテスト

開発ライフサイクル（Vモデル）



ソフトウェアテストにもライフサイクルがある



テストにも要求分析がある

何に対してどのようなテストを行う必要があるのか？を明らかにする

☑関係者は誰か？それぞれの関心事は何か？誰がどのような状況で使うのか？
利用者像、解決すべき課題、利用シナリオ、利用者の背景、利用時リスク、他の関係者の関心事など

☑さまざまな要求（機能・非機能）は？
システム要求、ソフトウェア要求、機能要求、非機能要求、理想的な使い方、差別化要因、目的機能など

☑どのような流れや構造なのか？
状態（画面）遷移、機能連携、外部システム連携、システム基盤との連携方法など

☑分析の結果をまとめる

- 何をテストする？
- テストの範囲は？
- テストの規模は？
- テストレベルは何？
- テストタイプやテスト観点は？
- 顧客や関係者が特に重視しているのは？
- テストの期間やコストはどれくらい？

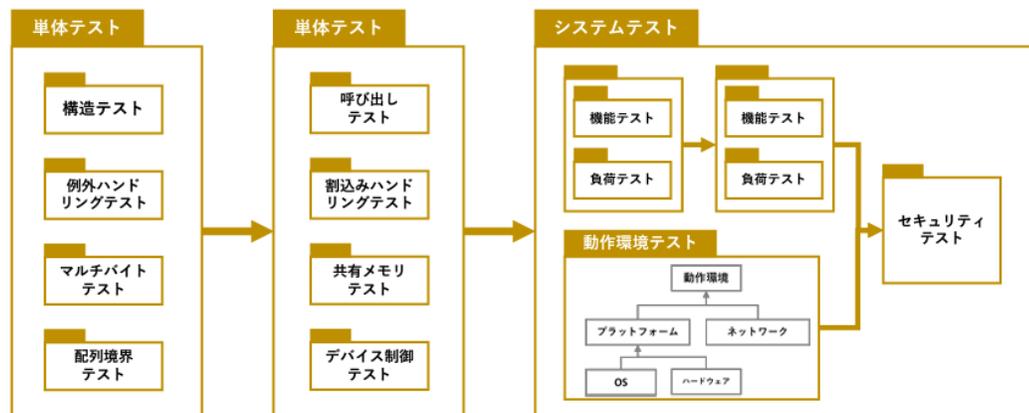
☑バグのありそうなところは？

- ユーザがミスしそうなところ
- 構造上問題が起きそうなところ
- 前工程までの検証作業（レビューやテスト）が足りなかったり滞ったりしたところ
- 類似製品や母体系製品の過去バグ、顧客クレームから分析した知識
- スキルの足りないエンジニアが担当したところ
- 設計中に不安が感じられたところ
- 進捗が滞ったり、エンジニアが大きく入れ替わったりしたところ

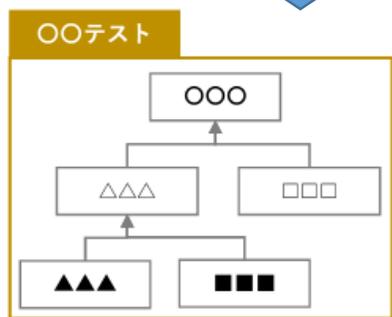


参考：[テスト設計チュートリアル 2021 ちびこん編資料](#)
[テスト設計チュートリアル テスコン編資料\(講義編\)](#)

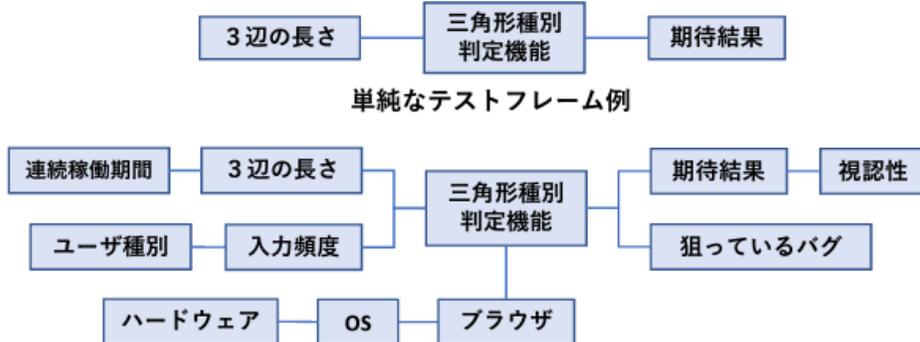
そして、テスト設計→実装→実行へ テストにもエンジニアリングが必要



テストアーキテクチャ(設計)/コンテナモデリング



テストケースの構造化/テストフレーム



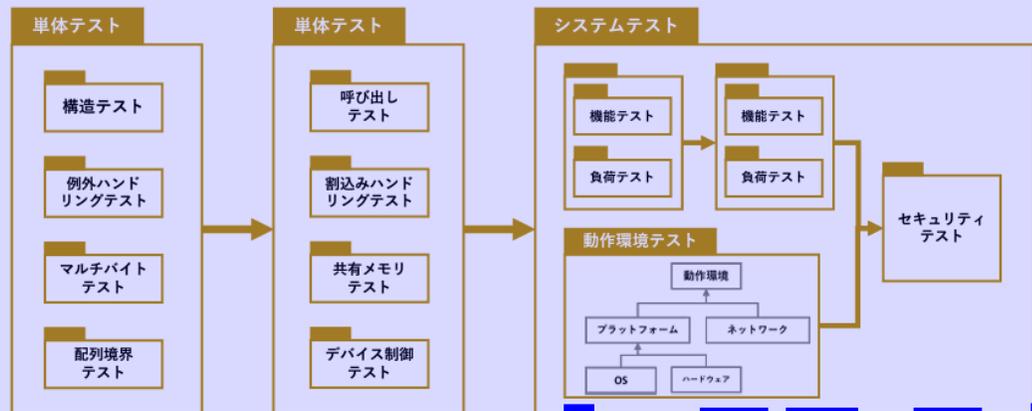
ID	3辺の長さ	ブラウザ	期待結果
1	(3, 4, 5)	Safari	不等辺三角形
2	(3, 3, 4)	Safari	二等辺三角形
3	(3, 3, 3)	Safari	正三角形
4	(3, 3, 6)	Safari	潰れて三角形にならない

テストケース表



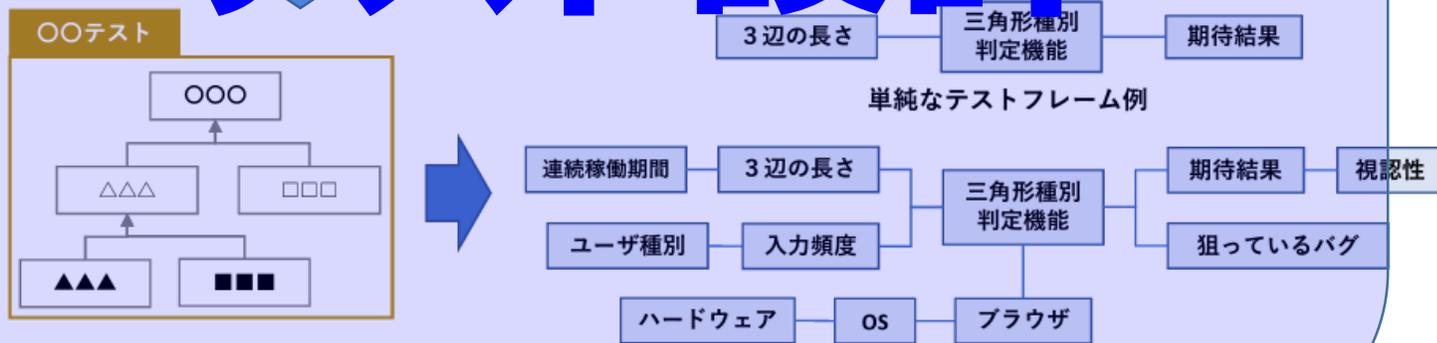
テスト設計チュートリアル
テスコン編資料(講義編) より

そしてテスト設計→実装→実行へ テストにもエンジニアリングが必要



テストアーキテクチャ設計 / コンテンツ設計

テスト設計



テストケースの構造化 / テストフレーム

ID	3辺の長さ	ブラウザ	期待結果
1	(3, 4, 5)	Safari	不等辺三角形
2	(3, 3, 4)	Safari	二等辺三角形
3	(3, 3, 3)	Safari	正三角形
4	(3, 3, 6)	Safari	決して三角形にならない

テスト
実装

テストケース表



テスト実行

テスト設計チュートリアル
テスコン編資料(講義編) より

テスト設計技法の全体像

テスト技術者資格制度 Foundation Levelシラバス Version 2023V4.0.J01
テスト技術者資格制度 Advanced Level シラバス日本語版 テストアナリスト Version 3.1.1.J03

テスト技法は、テスト担当者がテスト分析やテスト設計の際に、テスト条件を定義し、カバレッジアイテムを識別し、テストデータを識別する。

テスト分析（何をテストするか）とテスト設計（どのようにテストするか）において、テスト担当者をサポートする。
また、比較的少ないながらも十分なテストケースのセットを体系的に開発するのに役立つ。

ブラックボックステスト技法 （仕様ベース技法）

テスト対象の内部構造を参照せずに、仕様上の振る舞いの分析に基づくテスト

- 同値分割法
- 境界値分析
- デシジョンテーブルテスト
- 状態遷移テスト
- クラシフィケーションツリー技法
- ペアワイズテスト
- ユースケーステスト

ホワイトボックステスト技法 （構造ベース技法）

テスト対象の内部構造や処理の分析に基づくテスト

- ステートメントテスト
- ブランチテスト
（制御フローテスト）

経験ベースのテスト技法

テストケースの設計・実装にテスト担当者の知識と経験を効果的に利用する

- エラー推測
- チェックリストベースドテスト
- 探索的テスト
- 欠陥ベースのテスト技法

「テスト設計技法実践ワークテキスト」より

テスト設計技法例

「テスト設計技法実践ワークテキスト」より

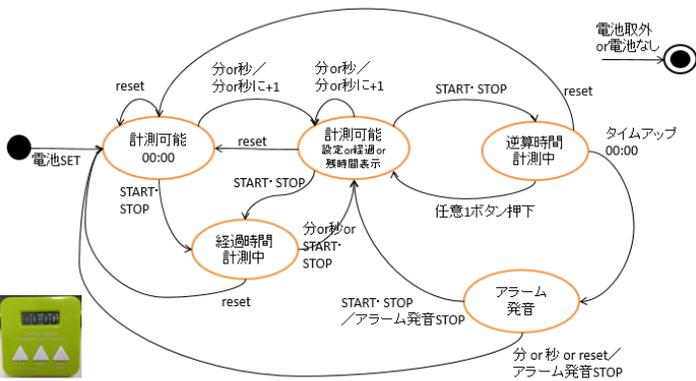


図:100円均一ショップキッチンタイマー状態遷移図

状態遷移図

状態	(0)使用不可	(1)計測可能 00:00表示	(2)経過時間 計測中	(3)計測可能 時間表示	(4)逆算時間 計測中	(5)アラーム 発音
電池SET	→(1)	N/A	N/A	N/A	N/A	N/A
Reset	N/A	→(1)	→(1)	→(1)	→(1)	→(1) 音STOP
START/STOP	N/A	→(2)	→(3)	→(2)or(4)	→(3)	→(3) 音STOP
分or秒ボタン 押下	N/A	→(3) 分or秒に+1	→(4)	→(3) 分or秒に+1	→(3)	→(1) 音STOP
タイムアップ	N/A	N/A	N/A	N/A	→(5)	N/A
電池取外or 電池なし	N/A	→(0)	→(0)	→(0)	→(0)	→(0)

状態遷移表

関係行列表

遷移前	(0)	(1)	(2)	(3)	(4)
(0)	Con入(T)				
(1)	Con抜(N)	標準分押(K)	強設定押(K)		
(2)	Con抜(N)	切分押(C)	強設定押(K)		
(3)	Con抜(N)	切分押(C)	標準分押(K)	強設定押(K)	
(4)	Con抜(N)	切分押(C)	標準分押(K)	強設定押(K)	

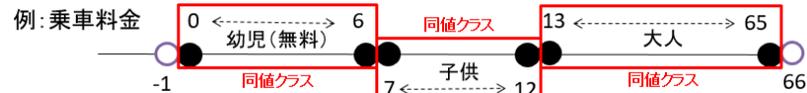
2乗 ×

遷移前	(0)	(1)	(2)	(3)	(4)
(0)	Con入(T)				
(1)	Con抜(N)	標準分押(K)	強設定押(K)		
(2)	Con抜(N)	切分押(C)	標準分押(K)	強設定押(K)	
(3)	Con抜(N)	切分押(C)	標準分押(K)	強設定押(K)	
(4)	Con抜(N)	切分押(C)	標準分押(K)	強設定押(K)	

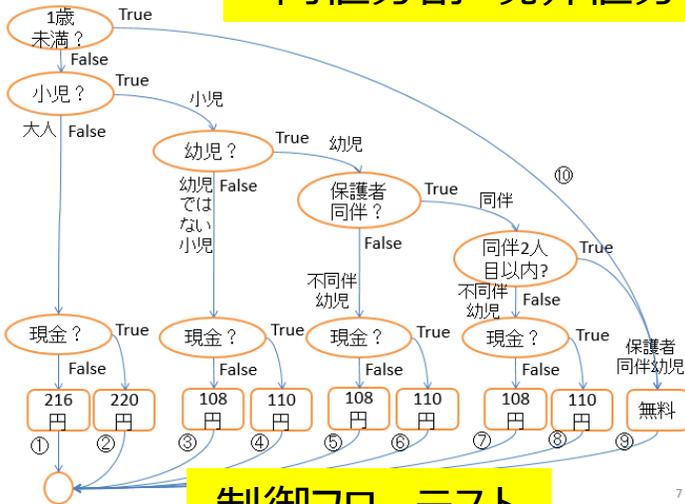
1スイッチカバレッジ表

遷移前	(0)	(1)	(2)	(3)	(4)
(0)	TN		TH		TK
(1)	HN+KN	NT+HO+KO	KH		HK
(2)	ON+KN	NT+KO	OH+KH		OK
(3)	ON+HN+KN	NT+HO+KO	OH+KH		OK+HK+KK
(4)	ON+HN+KN	NT+HO+KO	OH+KH		OK+HK
(4)	ON+HN+KN	NT+HO+KO	OH+KH		KK

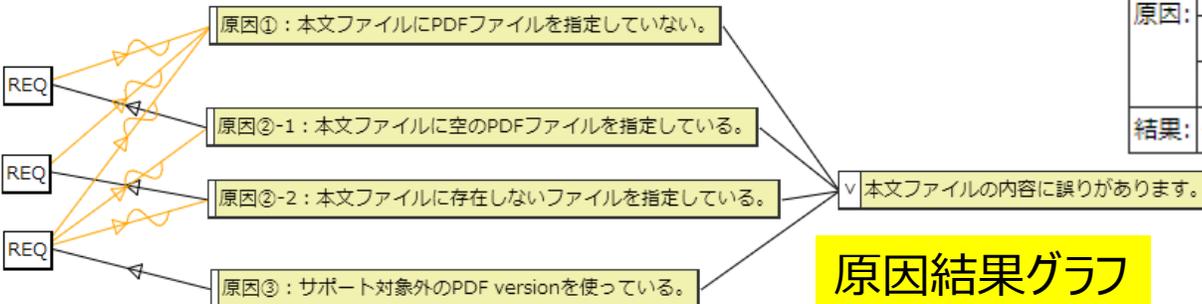
Nスイッチテスト



同値分割・境界値分析



制御フローテスト



原因結果グラフ

世界大学の合格基準は、以下の通りです。
 $950 \leq A \times 100 + B$
 A: 高校3年時主要5教科最終成績(平均)
 $A \geq 5.0$
 B: 入学試験の合計点数(10点単位)
 $B \leq 500$

	4.4	4.5	4.6	4.7	4.8	4.9	5.0	
500	⑤	⑤			③			
490								
480			⑨			①		
470								②
460					⑩			
450								⑦
440								⑧

	Type	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩
A 成績 平均	on	5.0	-	-	-	-	-	-	-	-	-
	off	-	5.1	-	-	-	-	-	-	-	-
B 入試 点数	In	-	-	4.8	4.9	4.5	4.4	5.0	5.0	4.7	4.8
	in	480	470	-	510	-	-	-	-	-	-
期待 結果	-	合格								合格	不合格

ドメイン分析テスト

因子	水準1	水準2
フォント	明朝	ゴシック
文字の大きさ	標準	大
太字	標準	太字
下線	なし	あり



組合せテスト

	フォント	大きさ	太字	下線
1	明朝	標準	標準	なし
2	明朝	標準	標準	あり
3	明朝	大	太字	なし
4	明朝	大	太字	あり
5	ゴシック	標準	太字	なし
6	ゴシック	標準	太字	あり
7	ゴシック	大	標準	なし
8	ゴシック	大	標準	あり

デジジョンテーブル

ノード名		#1	#2	#3	#4	#5
原因:	原因①	T	F	F	F	F
	原因②-1	F	T	F	F	F
	原因②-2	F	F	T	F	F
結果: {obs}	MSG出力	T	T	T	T	F

デジジョンテーブル

CPM (copy & paste & modify) 法

※電気通信大学にしさんが発見し命名したもの

仕様書

〇〇と□□が入力され、処理ボタンを押下したら計算を開始し、結果を出力する。

①コピー
②貼り付け

テスト仕様書

〇〇と□□が入力され、処理ボタンを押下したら計算を開始し、結果を出力することを確認する。

③テストケースっぽい表現に修正

<CPM法の利点>

- ・ほとんど頭を使わずにテスト (っぽいこと) をやった気になれる。
- ・テスト (っぽいこと) を早く終わらせられる。やり終えた気持ちも早く味わえる。
- ・たまにうまくいく。
- ・仕様書記述を網羅することでテスト側の責任を（表向きは）回避できる。

<CPM法の弱点・問題点>

- ・テストケース量と内容の良し悪しが仕様書記述に依存してしまう。
- ・こんなにがんばりました、時間が来ました、全て実施しました&検出したバグは対応済みです、が終了判定基準になってしまう。
- ・一方で、テスト中～終了後に大量のバグに翻弄されやすい。

<背景に存在する認識>

「テストは仕様の裏返し＝頭を使わないから簡単！」「テストは設計・実装のおまけ」

固定3レイヤ法

電気通信大学にしさんが命名したもの
テストケースを大項目・中項目・小項目の様式で設計していく方法

大項目	中項目	小項目	テストケース
沸騰機能	沸騰開始	沸騰ボタン押下	沸騰ボタン押下
機能	沸騰	沸騰行為 + カルキ抜き	沸騰行為 + カルキ抜き
沸騰機能	気圧低い	0.7気圧 (富士山頂)	0.7気圧で沸騰
機能組合せ	沸騰機能	給湯機能	沸騰×給湯
.	.	.	.
.	.	.	.

<固定3レイヤ法の利点>

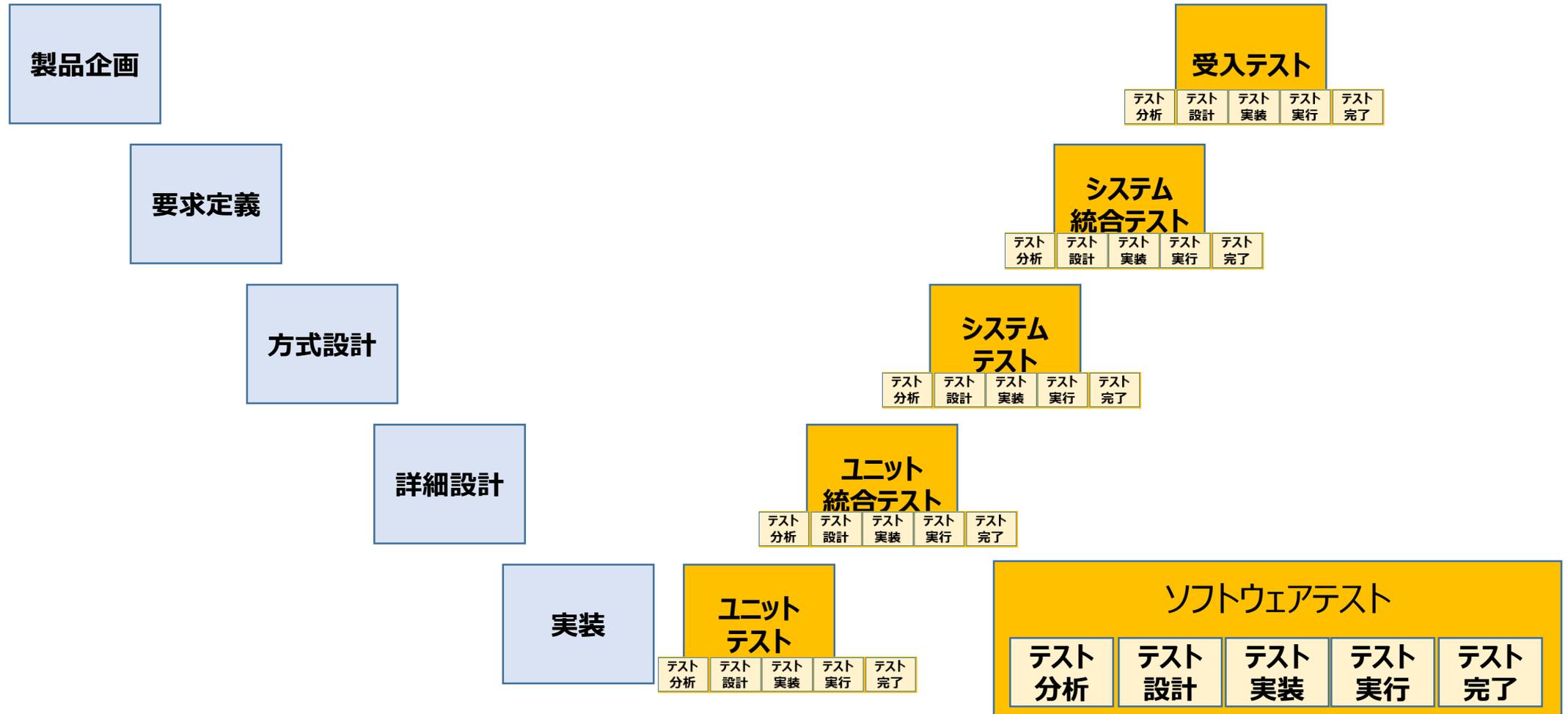
- ・テスト設計（段階的詳細化）をやった気になれるし、それっぽくも見える。
- ・腕の良い人がやるとうまくいく場合もある。

<固定3レイヤ法の弱点・問題点>

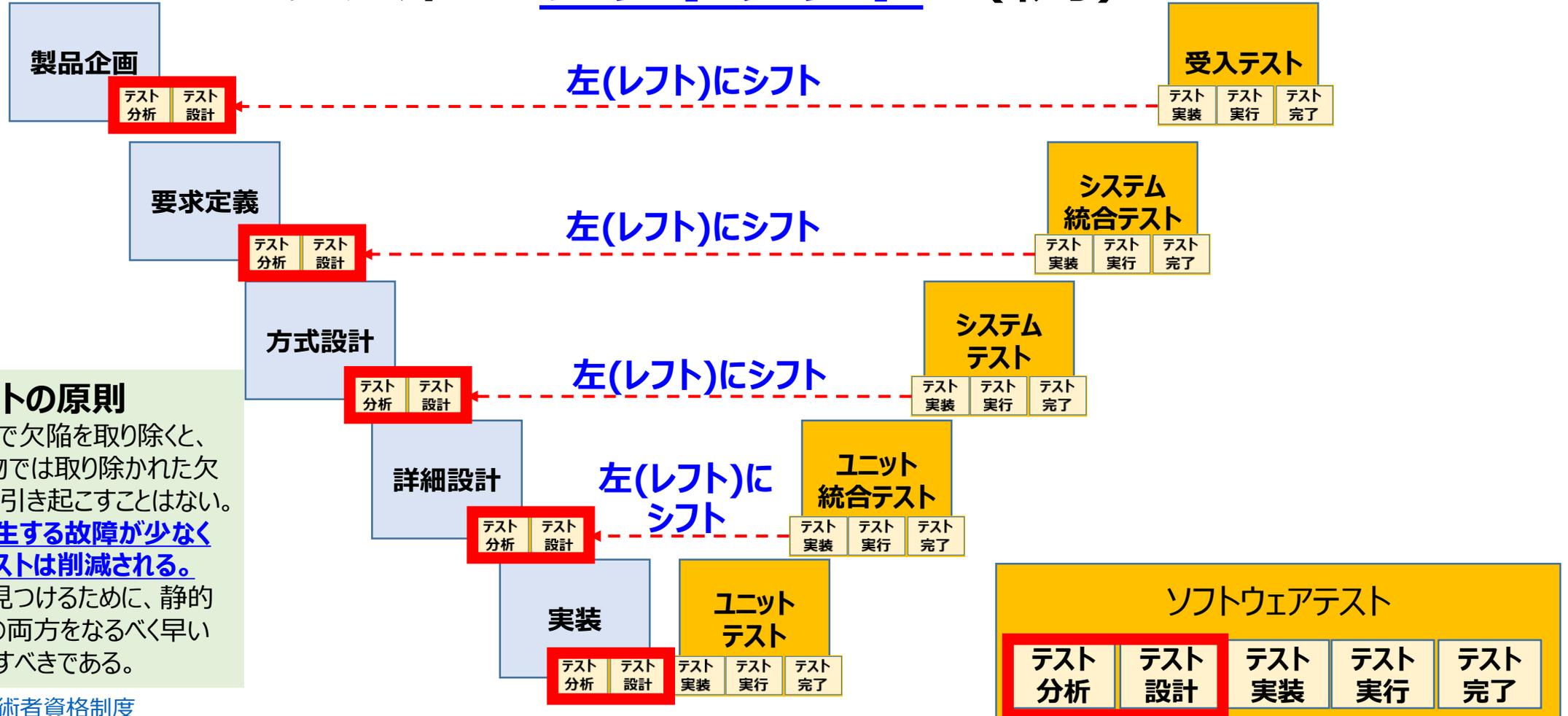
- ・思いつきを含め沢山埋めたら満足しがち。
- ・詳細化のレベルが合わない、ばらつく。
- ・異なるテスト観点が混在してしまう。
- ・網羅できない。抜け・漏れが多くなる。
- ・詳細化ではないものなども混在する。

開発とテストの融合～シフトレフト

ソフトウェアテストにもライフサイクルがある



ソフトウェアテストの前段部分を先出しする ～テストのシフトレフト（例）



早期テストの原則

プロセスの早い段階で欠陥を取り除くと、その後の作業成果物では取り除かれた欠陥に起因する欠陥を引き起こすことはない。

SDLCの後半に発生する故障が少なくなるため、品質コストは削減される。

早い段階で欠陥を見つけるために、静的テストと動的テストの両方なるべく早い時期に開始すべきである。

引用：[ISTQBテスト技術者資格制度](#)

[Foundation Level シラバス 日本語版 Version](#)

[2023V4.0.J01](#)

JaSSST Review2023「レビュー体系化の経過報告：レビュー体系とレビューアーキテクチャー」より

シフトレフトの原理を活用した テスト主導ソフトウェア開発

×テスト手法
○プログラミング手法

BDD

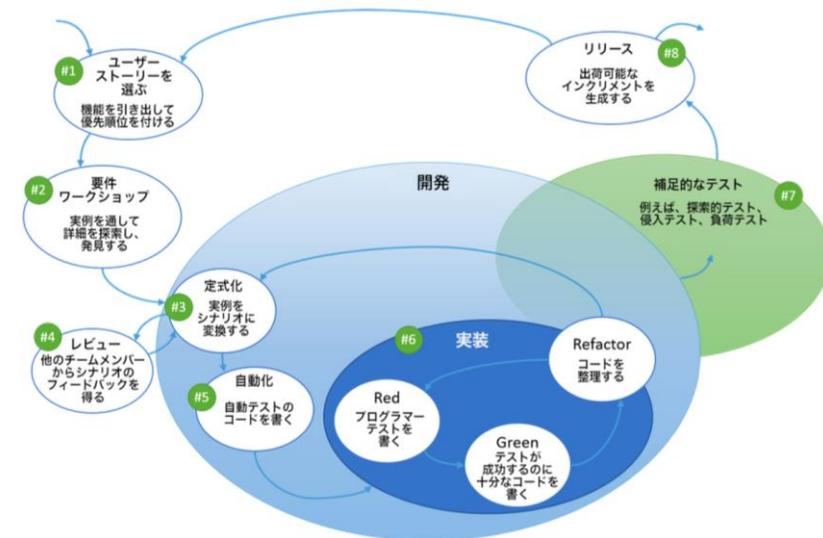
Behavior Driven Development
振る舞い駆動開発

ATDD

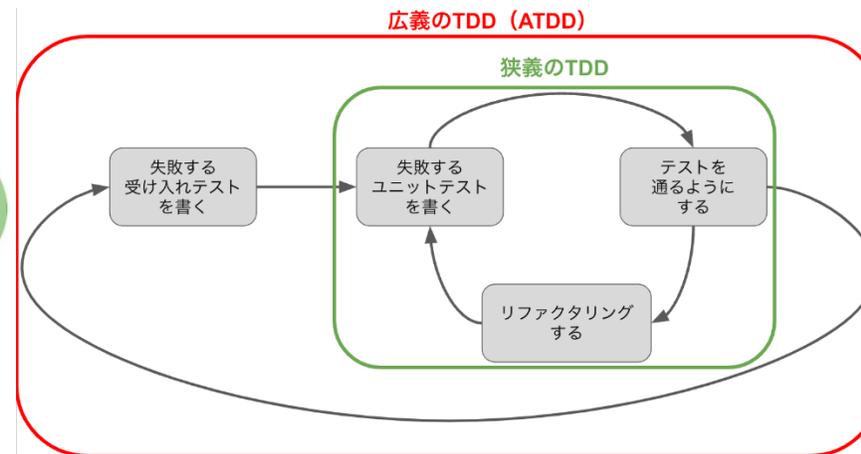
Acceptance test-driven
development
受け入れテスト駆動開発

TDD

Test-Driven Development
テスト駆動開発



TDDとBDD/ATDD(4)
ツールとしてのBDDとプロセスに
組み込まれたBDD より



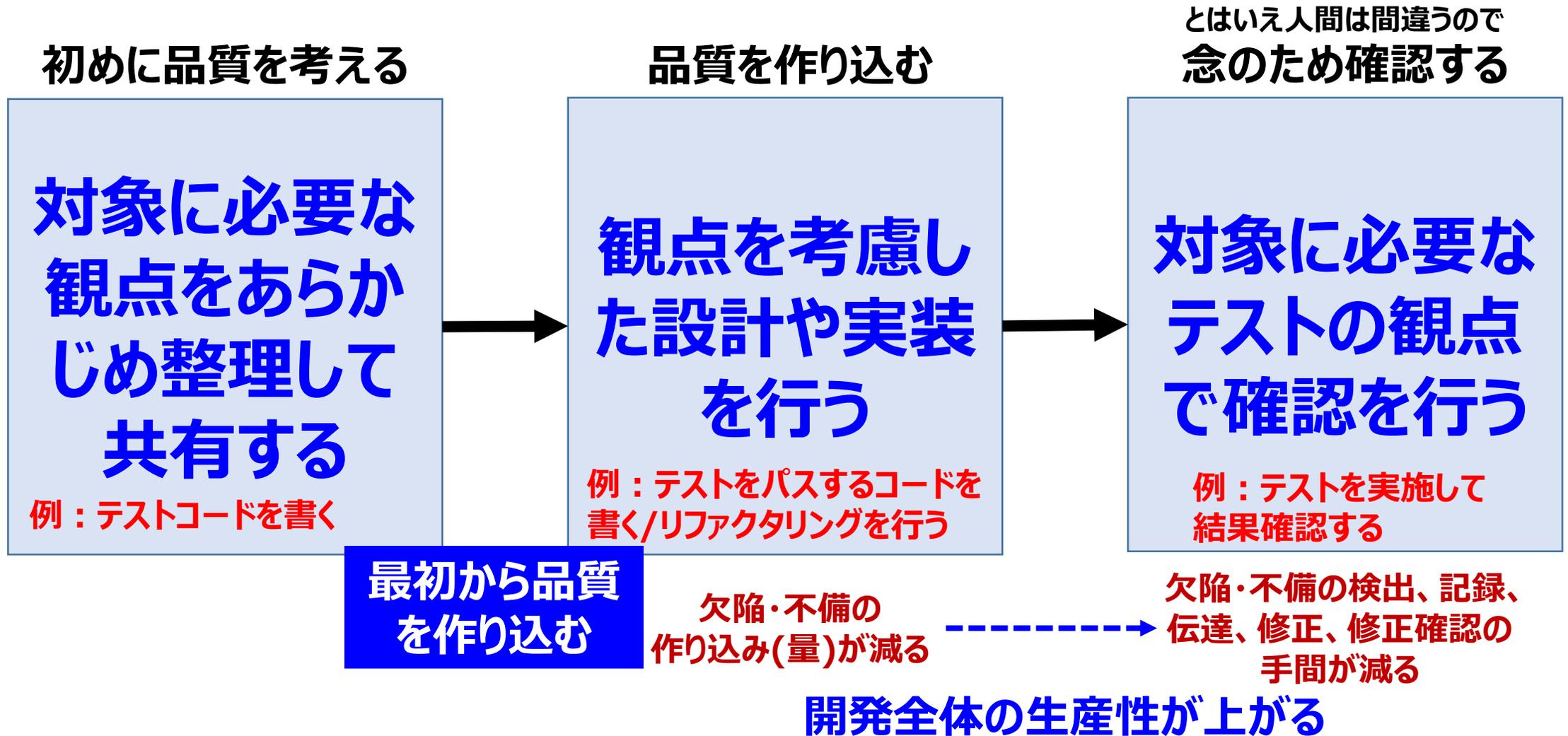
TDDとBDD/ATDD(3)
BDDとATDDとSbE より

TDDのサイクル

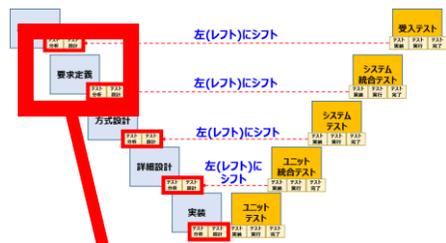
1. 次の目標を考える
2. その目標を示すテストを書く
3. そのテストを実行して失敗させる (Red)
4. 目的のコードを書く
5. 2で書いたテストを成功させる (Green)
6. テストが通るままでリファクタリングを行う (Refactor)
7. 1~6を繰り返す

50分でわかるテスト駆動開発 /
TDD Live in 50 minutes より

シフトレフトでやっていること(原理)

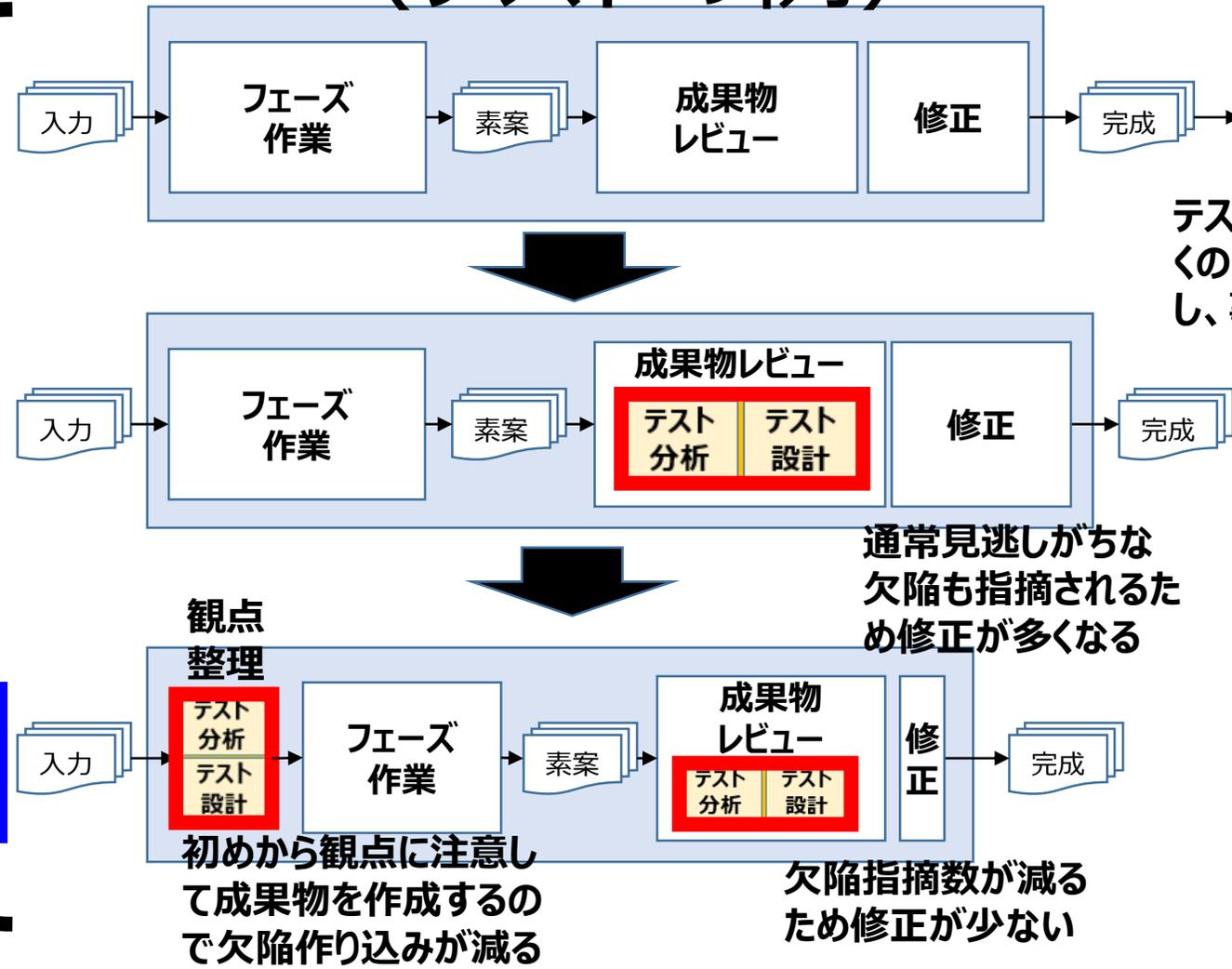


同一フェーズ内でのシフトレフト **ライクなこと** (テストの例)



例えば「要求定義フェーズ」

**最初から品質
を作り込む**



テスト分析 | **テスト設計**

テスト分析、設計過程で多くの質問や欠陥指摘が発生し、事後修正が多くなる

通常見逃しがちな欠陥も指摘されるため修正が多くなる

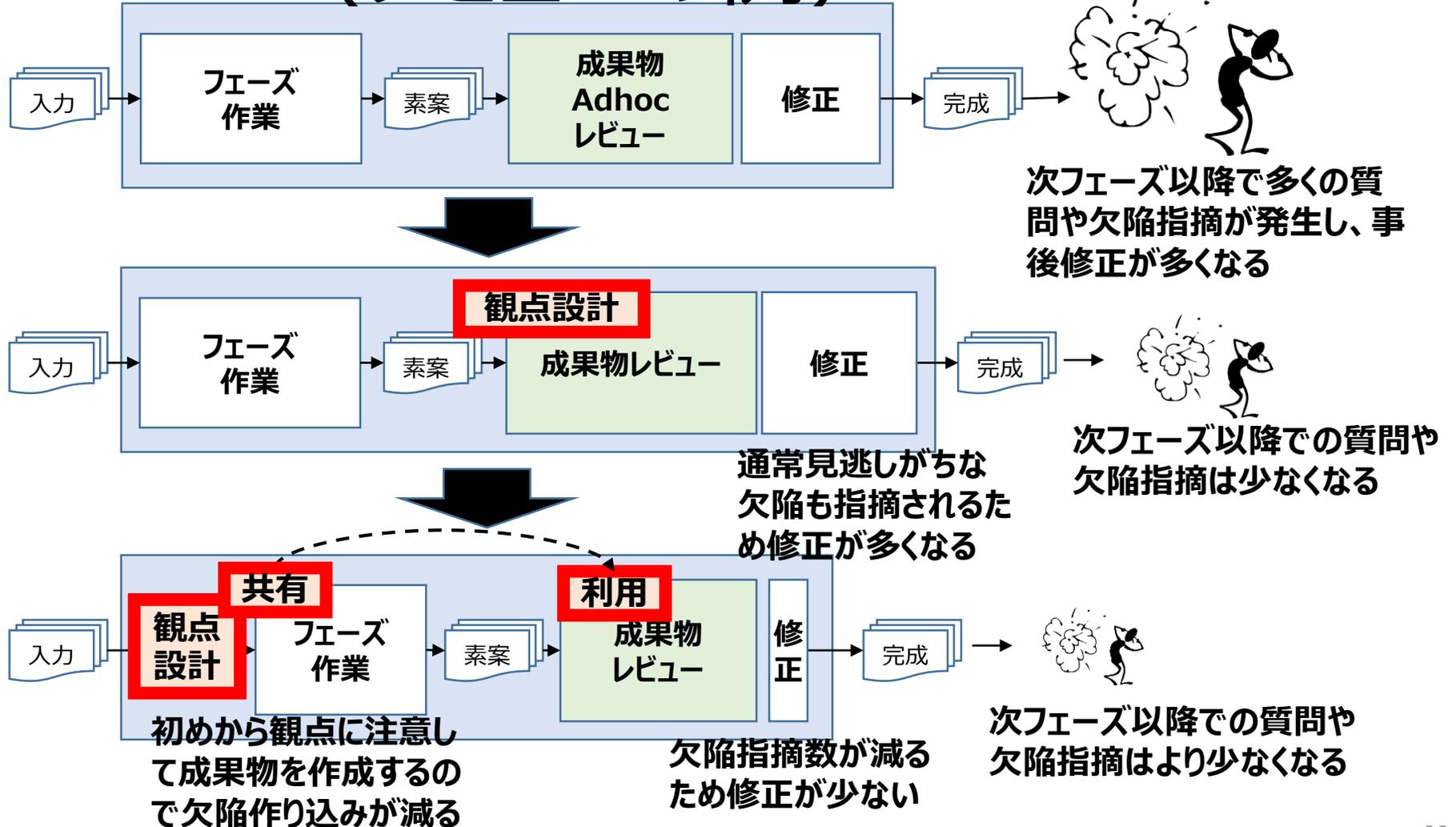
初めから観点に注意して成果物を作成するので欠陥作り込みが減る

欠陥指摘数が減るため修正が少ない

JaSST Review2023
「レビュー体系化の経過報告：レビュー体系とレビューアーキテクチャー」より

レビューにおけるシフトレフト **ライク**なこと (レビューの例)

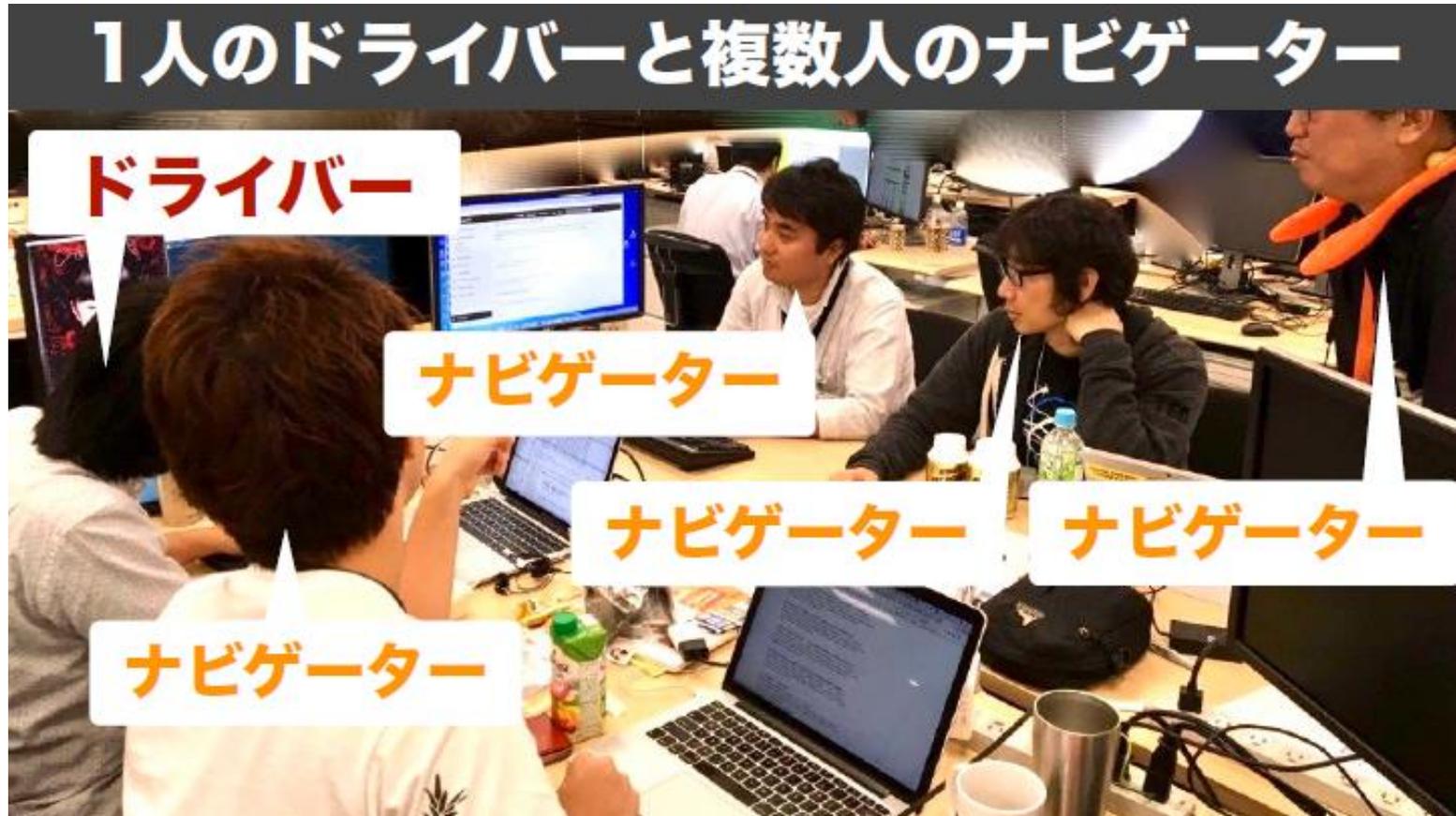
JaSST Review2023
「レビュー体系化の経過報告：レビュー体系とレビューアーキテクチャー」より



**最初から品質
を作り込む**

モブプログラミング

JaSST Review2018 @TAKAKING22さん「レビュー再定義」講演資料より



エビデンスやドキュメントを詳細に書き表す
ことが品質活動（そのもの）だと勘違いし
ている人も多いので注意

テスト(ソフトウェア品質)のノウハウは 開発をより効果的に、かつ効率的に 進めるために役立つ

別モノ扱いにするとこの効能は失われ
開発が混乱していく

テストスキルの磨き方

テスト技術は

- 開発者も必要な技術
- 知っていても損はしない



では、
どうやって技術を身に付けるか？



独学・独習

もしくは

書籍を読む

これだけだと
つまらん



コミュニティ系

最新技術や情報を得るためには

Slack！！

TestingCommunityJP

testingcommunityjp.slack.com



Test Engineer's Forum 北海道

(TEF道)

ソフトウェアのテストを中心として、ソフトウェアテストの勉強会、ワークショップの開催にとどまらず、良いソフトを作るための技術やノウハウの共有、ときには研究っぽいことをやってみて論文提出など、いろいろやっています。

#tefdo



How do you like
software test?

TEF道

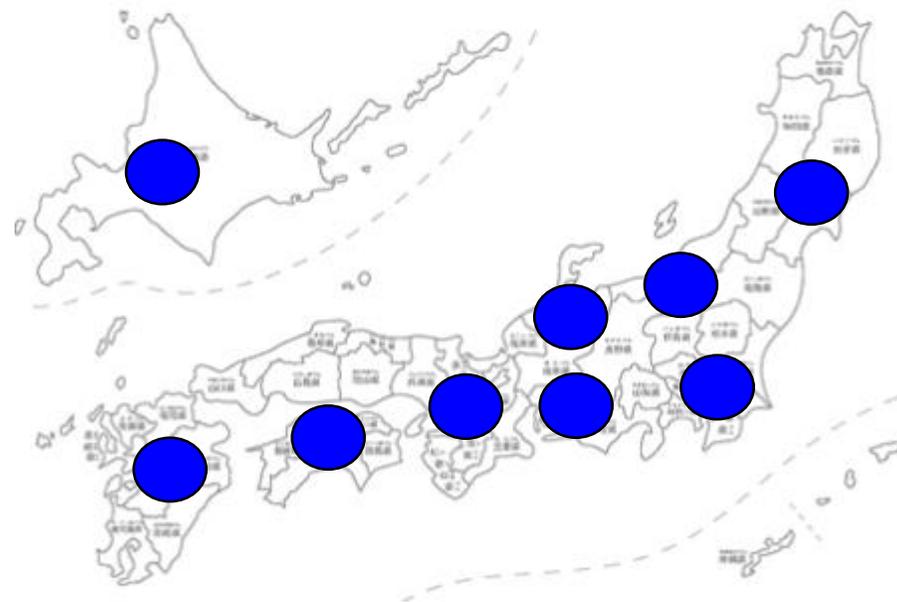
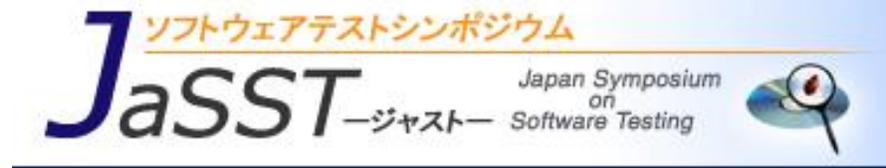
Do the best it yourself
... de Show!

イベント系



ソフトウェアテストシンポジウム(JaSST)

- 2003年に東京で開始
 - 参加費は実費 + α / 実行委員は手弁当
 - 地域JaSST : [北海道](#)、東北、東京、新潟、北陸、東海、関西、四国、九州の9箇所
 - **JaSST Review** : 東京 / オンラインで, **JaSST Online**を開催
 - **JaSST nano** : 2021年6月より毎月第3火曜日に開催
- JaSST北海道は、7月頃に開催
 - 毎回のべ100名前後が参加
 - 著名な講演者 / 異分野講演者を招聘
 - 事例発表・企画セッションなど
 - 2023年 : **UIデザインワークショップ**や**経験・事例発表**など



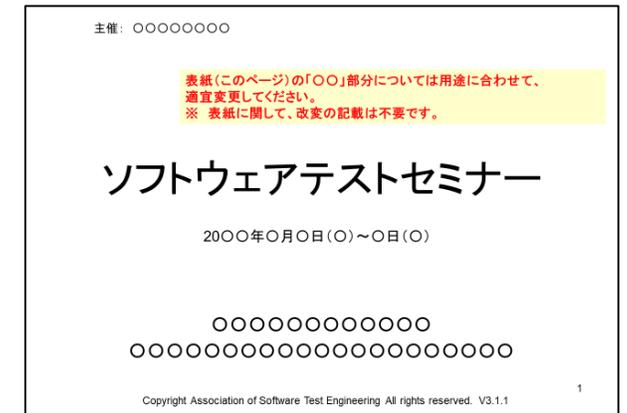
ASTER標準テストセミナー／テスト設計コンテスト

• 各地でソフトウェアテストの教育や活動支援を実施

- 札幌, 宮城, 新潟などで自治体系団体と共同で教育
- 九州や東北などで勉強会の支援により地場の産業振興の定着を図る
- 「ASTERセミナー標準テキスト」を2018年9月公開

(Ver3.1.2 : 2023年3月改訂)

- 基本的な考え方をきちんと押さえているので、すべてのエンジニアに一読してほしい
- 商用/非商用、公開/非公開を問わず利用可能という柔軟なライセンスなので、個人でも組織でも教育機関でも営利セミナーでもどんどん使っていただきたい



ASTERセミナー標準テキスト

• テスト設計コンテスト

- (特に上流の)テスト技術の向上と促進の機会を提供
- JaSST'11 Tokyoより開催、2012年から全国展開
- 同一のテスト対象に対してテスト設計の良さを競う
 - OPENクラスと若手対象のU-30クラスの2つのクラス

• **毎年参加チーム受付中!**



テスコン

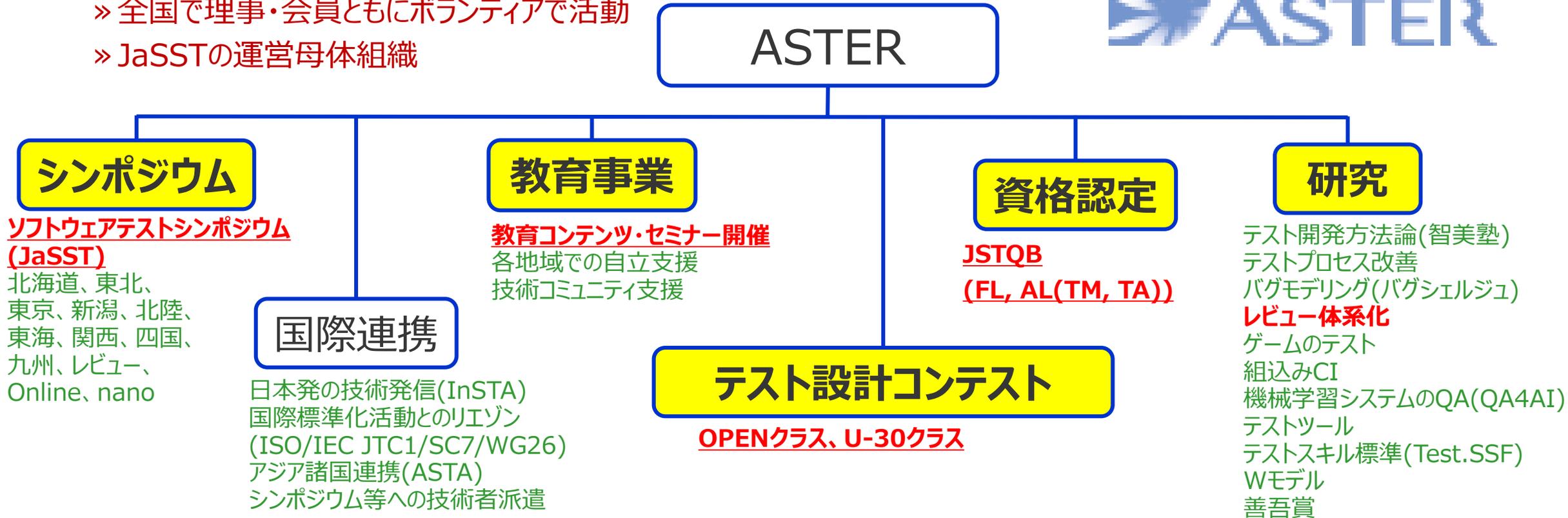
ちびこん

ASTER: ソフトウェアテスト技術振興協会

Association of Software Test EngineerRing

ソフトウェアテストを軸にして、ソフトウェア品質向上に関する研究開発、普及振興、教育、国際連携、資格認定などの事業を行う特定非営利活動法人(NPO法人)

- » 2006年4月に設立
- » 全国で理事・会員ともにボランティアで活動
- » JaSSTの運営母体組織



ソフトウェアレビュー勉強会

<https://softwarereview-studygroup.connpass.com/>

- [ソフトウェア品質管理\(SQIP\)研究会 研究コース2：ソフトウェアレビュー](#)の指導講師3名で実施した、JaSST2022東京レビューワークをきっかけにしたソフトウェアレビューの非定期開催勉強会。

ソフトウェアテストシンポジウム2022東京 レビューワークショップ

ワークショップ

セッション F6 ▶ 概要

「そゆことね！よくわかるレビューテクニック
～明日から使える技術をSQIPレビュー研究会からあなたに～」

安達 賢二 (HBA)
上田 裕之 (DTS-インサイト)
中谷 一樹 (TIS)

[Download](#) 講演資料 (PDF : 4,379KB)



ソフトウェアレビュー勉強会



これまでの開催テーマ

- (11)最新のレビュー観点導出手法体験
- (10)レビューの問題点と解決策
- (9)ChatGPT+ソフトウェアレビュー
- (8)レビュー会議における発言のしやすさ
- (7)製品リスクベースのレビュー観点導出
- (6)認知バイアスに着目した欠陥予測
- (5)欠陥知識の活用
- (4)レビュー結果の評価とふりかえり
- (3)レビューファシリテーション
- (2)HDR法を使った仮説検証アプローチ
- (1)レビュー目的の構造化
- (0)JaSST2022東京ワークふりかえり

他にもこんなイベントがある

国内

- SQiPシンポジウム(ソフトウェア品質シンポジウム)
- SS(ソフトウェアシンポジウム)
- ソフトウェアプロセス改善カンファレンス

海外

- ICST
- STAR WEST/EAST
- Agile Testing Days



〔これは一部情報、まだまだあるよ〕

資格系

ソフトウェアテスト技術者資格試験(JSTQB)

- 2005年にISTQBに加盟
 - 加盟国は現在**130ヶ国以上**
 - 全世界で**のべ120万人以上が受験、91.4万人以上認定** (2023年6月時点)
- Foundation Level
 - 36,334名の受験者、**20,579名の合格者** (2006年～2022年2月時点)
- Advanced Level (テストマネージャ)
 - 4,047名の受験者、**717名の合格者** (2010年～2022年2月時点)
- Advanced Level (テストアナリスト)
 - 2,540名の受験者、**548名の合格者** (2016年～2022年2月時点)
- ISTQBパートナープログラムを2015年より開始
 - 現在、64社がISTQB認定パートナー (2023年9月時点)



Foundation Level
シラバス



認定試験にピアソンVUE によるコンピューター試験(CBT)を導入

日本語化されたテストシラバス群(無償でPDF参照可能)

<https://jstqb.jp/syllabus.html>

Core Advanced

Advanced Level シラバス
テストマネージャ
Version2012.J04

Advanced Level シラバス
テストアナリスト
Version 3.1.1.J03

Advanced Level シラバス
テクニカルテストアナリスト
Version2012.J02

Agile

Advanced Level Syllabus
アジャイルテクニカル
テスト担当者 Version
1.1.J01

Foundation Level
Extension シラバス
アジャイルテスト担当者
Version 2014.J02

Specialist

Advanced Level Specialist シラバス
テスト自動化エンジニア
Version 2016.J01

Foundation Level Specialist シラバス
モバイルアプリケーションテスト
担当者 Version 2019.J01

Foundation Level Specialist シラバス
AIテストイング Verison1.0.J01

Foundation Level Specialist シラバス
性能テスト担当者 Version
2018.J01

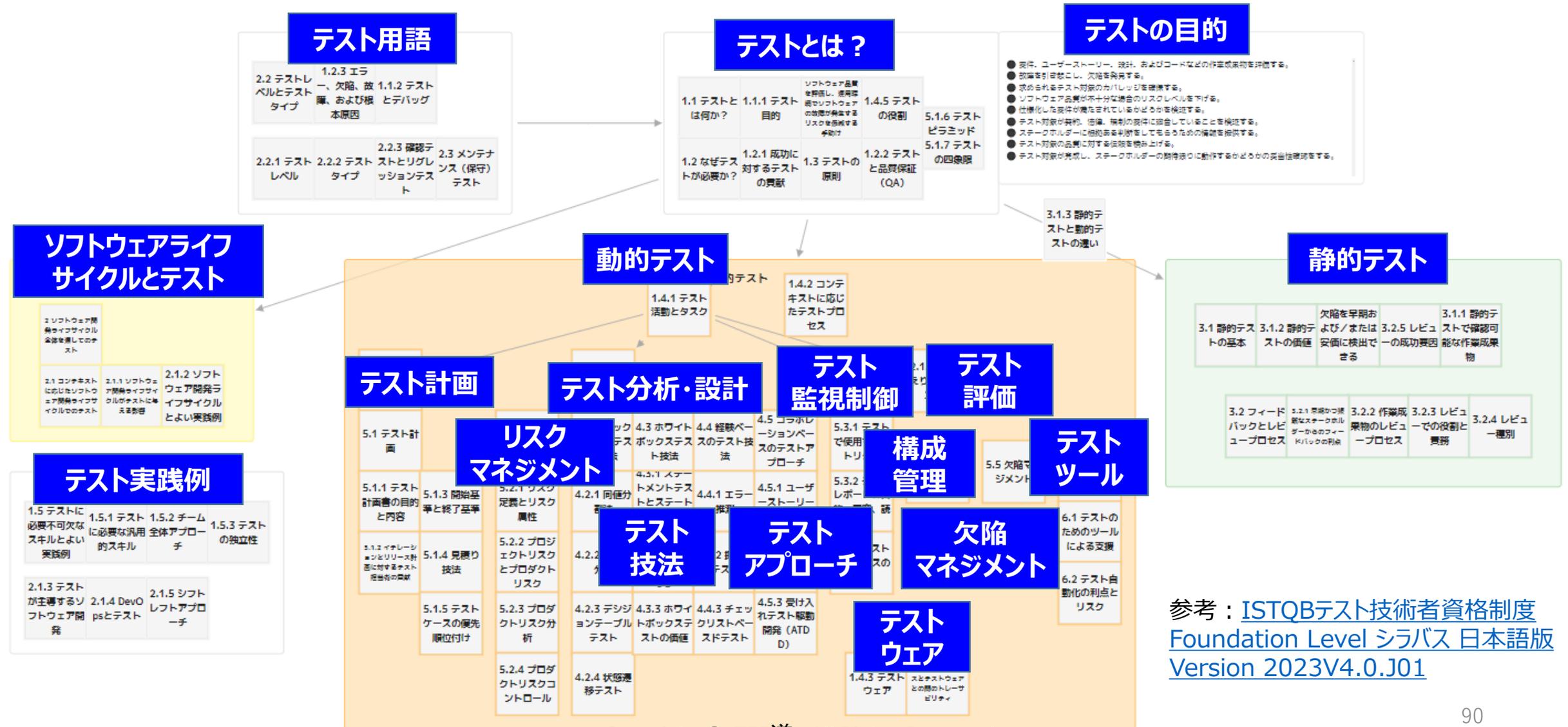
Foundation Level Specialist シラバス
自動車ソフトウェアテスト担当者
Version 2018.J03

Core Foundation

Foundation
Level シラバス Version
2023V4.0.J01



JSTQB Foundation Level シラバスの全体像



参考 : [ISTQBテスト技術者資格制度 Foundation Level シラバス 日本語版 Version 2023V4.0.J01](#)

他にもこんな資格がある

ソフトウェアのテストに関する資格

- IVEC:IT検証技術者認定試験

ソフトウェアの品質に関する資格

- JCSQE:ソフトウェア品質技術者資格認定



書籍

ソフトウェアテスト関連書籍（例）

テスト実務の基本を学ぶ



知識ゼロから学ぶ
ソフトウェアテスト

テスト設計技法を演習問題で身につける



はじめて学ぶ
ソフトウェアテスト技法



ソフトウェアテスト技法
練習帳

テスト設計を具体的事例で学ぶ



マインドマップから始める
ソフトウェアテスト



ソフトウェアテスト
技法ドリル

スキル向上への情報源 (例)

スキル



テスト設計を具体的事例で学ぶ

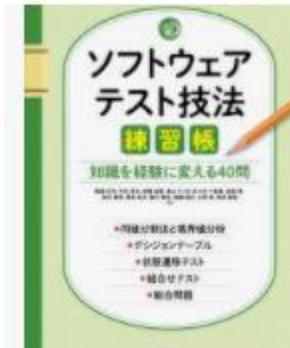


- 論文投稿
- 事例投稿
- テスト設計コンテストへの挑戦
- ソフトウェアテストコミュニティ活動への参加

テストの基本を学ぶ



テスト設計技法を演習問題で身につける



ソフトウェアテストを体系的に理解する



<http://jstqb.jp/syllabus.html>

JSTQB認定テスト技術者資格制度



知識

テストを学びたい開発者のためのソフトウェアテスト読書マップ

<https://speakerdeck.com/kzsuzuki/software-testing-reading-map-for-developers>

ソフトウェアテスト読書マップ

Another View

Feb. 9th, 2023 @kz_suzuki

- このマップは、鈴木が好きな本を中心に並べたものに対し、マップ作成チームにいただいたコメントを取り込んだものです。
- ここにはない本に対する否定的な意図はなく、おそらく単にわたしが読めずにいるだけです。不勉強でスママセン。
- 開発者テストやテスト分析への言及も薄いです、これも不勉強です…。

マネージャーなどがソフトウェアテストの全体をより深く押さえる

全体概要エリア



もっと詳しく具体的な内容を知りたい場合



ソフトウェアテスト全般について比較的平易に書かれた入門書

テスト計画・管理エリア



テストマネージャとしての第一歩目を学ぶための良書

テスト設計エリア



名著・古典。ちゃんと読んでない…

ホワイトボックステストもカバーしたい場合

演習 + 勉強会で強くなる!



テスト設計技法とツール、テスト分析について平易に説明

スルメエリア



テストを何周か経験し、悩んだ後に読むと得るものが大きい実践書

モダンなテスト文化エリア



アジャイルテスト全般の解説と事例紹介

テスト自動化エリア



ツールやドメインにあまり依存しない一般論を学ぶ

Webシステムに対する自動テストについてStep by Stepで説明

アジャイルにおけるテストの位置づけや、先進企業のテストを学ぶ

テストの作り方はわかったので、自動化するぜ!

Case Study

私のソフトウェアテストへのアプローチ (当初)

2000～2006年頃



欲しい知識・スキルは
首都圏開催の研修で
しか手に入らない

会社に相談



なぜそのスキルが現業務
に必要なの？
そんなお金はないよっ！



有休をとり、自腹で
首都圏研修受講×n



お財布ががががが



実業務ががががが

地元で関連イ
ベントを開催すれ
ば首都圏に行か
なくても解決する
かも！



地元で品質関連コ
ミュニティ立ち上げ
(2005年)

TEF道



地元でソフトウェア
テストシンポジウム
開催(2006年～



地元で試験会場を呼び
寄せテスト技術者資格
JSTQBに挑戦

その後の活動

● 公開型継続的勉強会を運営



● コミュニティ(TEF道)メンバーとテストを深掘り研究



ソフトウェアテスト技術振興協会ASTER加入
JaSST運営統括・教育事業などを担当



国内外のソフトウェア品質関連イベントに投稿・発表



● 関連イベントやIT関連企業でワークショップ実施



2007～現在までASTER標準テストセミナー講師を担当(札幌・新潟などのべ20回超)



毎月第2火曜日開催
JaSST nano運営



● テスト設計コンテストに仲間と出場
2年目に全国準優勝



● テスト設計コンテストの地元運営+コンテスト審査員として活動



● JaSST北海道運営企画セッション提供



● JSTQB FL合格



● テスト技術者資格認定
JSTQBに入り技術委員として活動



ソフトウェアレビュー研究会
アドバイザー / 公開型継続的レビュー勉強会運営

まとめ

いかがでしたでしょうか？

今回は、

- ☑昨今のソフトウェア情勢とソフトウェアテストの重要性
- ☑開発ライフサイクルにおけるテスト
- ☑テストのライフサイクルと実際（エンジニアリング実践）
- ☑シフトレフト～開発との連携による効果
- ☑ソフトウェアテストスキルの磨き方

についてお伝えしました。

“テスト”に対する見方・捉え方は変わりましたか？

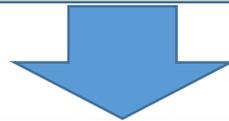


テスト、うまくいっていますか？ / テストと聞いて何をイメージしますか？
に対して

- 「テスト実施やバグ票」のことだけをイメージした。
- テスト・・・開発には関係ないねと思った。
- 網羅だのドキュメント化だの面倒な話なんだろうなーと思った。



方が多かったのではないのでしょうか？



**テストには分析や設計があるんだー！
たしかに開発と連携させるといいことがありそう！
自分も何かやってみようかなー**



って思ってもらえたならうれしいです。



もしよかったら一緒に**テスト沼**に入りませんか？
お疲れさまでした！

参考文献

- ソフトウェアテスト技術振興協会（ASTER）：<https://www.aster.or.jp/>
- ソフトウェアテストシンポジウム（JaSST）：<https://www.jasst.jp/>
- JaSST nano：<https://jasst-nano.connpass.com/>
- ソフトウェアレビュー勉強会：<https://softwarereview-studygroup.connpass.com/>
- ソフトウェア品質管理(SQIP)研究会 研究コース2：ソフトウェアレビュー：
<https://www.juse.or.jp/sqip/workshop/outline/index.html#kenkyu2>
- TestingCommunityJP：testingcommunityjp.slack.com
- Test Engineer's Forum 北海道（TEF道）：<https://ameblo.jp/tef-do/>
- JaSST Review2018 「レビュー再定義」：<https://www.jasst.jp/symposium/jasstreview18/pdf/S2.pdf>
- JaSST Review2023 「レビュー体系化の経過報告：レビュー体系とレビューアーキテクチャー」：
<https://www.jasst.jp/symposium/jasstreview23/pdf/S1.pdf>
- 「テスト設計技法実践ワークテキスト」
- JSTQB用語集：https://glossary.istqb.org/ja_JP/
- 50分でわかるテスト駆動開発 / TDD Live in 50 minutes
<https://speakerdeck.com/twada/tdd-live-in-50-minutes?slide=9>

参考文献の続き

- TDDとBDD/ATDD(3) BDDとATDDとSbE : <https://sqrpts.com/2023/08/07/61460/>
- TDDとBDD/ATDD(4) ツールとしてのBDDとプロセスに組み込まれたBDD : <https://sqrpts.com/2023/08/28/61494/>
- テスト設計チュートリアル2021 ちびこん編資料 : https://aster.or.jp/business/contest/doc/2021_chibicon_V1.0.0.pdf
- テスト設計チュートリアル テスコン編資料(講義編) : https://aster.or.jp/testcontest/doc/2023_tescon_V1.0.0.pdf
- IT検証技術者認定試験 (IVEC) : <https://www.ivia.or.jp/item43>
- ソフトウェア品質技術者資格認定 (JCSQE) : <https://www.juse.jp/jcsqe/>
- ASTER標準テストセミナー : <https://www.aster.or.jp/activities/education.html>
- テスト設計コンテスト : <https://www.aster.or.jp/testcontest/>
- 知識ゼロから学ぶソフトウェアテスト (ISBN-10 : 4798182435)
- はじめて学ぶソフトウェアテスト技法 (ISBN-10 : 4822282511)
- ソフトウェアテスト技法練習帳 (ISBN-10 : 429711061X)
- マインドマップから始めるソフトウェアテスト (ISBN-10 : 4774131318)
- ソフトウェアテスト技法ドリル (ISBN-10 : 4817197668)
- テストを学びたい開発者のためのソフトウェアテスト読書マップ
<https://speakerdeck.com/kzsuzuki/software-testing-reading-map-for-developers>

参考文献の続き

- ISTQBテスト技術者資格制度 Foundation Level シラバス 日本語版 Version 2023V4.0.J01
https://jstqb.jp/dl/JSTQB-SyllabusFoundation_VersionV40.J01.pdf
- ISTQBテスト技術者資格制度 Advanced Level シラバス 日本語版 テストマネージャ Version2012.J04
https://jstqb.jp/dl/JSTQB-Syllabus.Advanced_TM_Version2012.J04.pdf
- ISTQBテスト技術者資格制度 Advanced Level シラバス 日本語版 テストアナリスト Version 3.1.1.J03
https://jstqb.jp/dl/jstqb.jpdlJSTQB-SyllabusALTA_V311.J03.pdf
- ISTQBテスト技術者資格制度 Advanced Level シラバス 日本語版 テクニカルテストアナリスト Version2012.J02
https://jstqb.jp/dl/JSTQB-Syllabus.Advanced_TTA_Version2012.J02.pdf
- ISTQBテスト技術者資格制度 Foundation Level Specialist シラバス AIテストイング 日本語版 Verison1.0.J01
https://jstqb.jp/dl/JSTQB-Syllabus.Foundation_CT-AI_v1.0.J01.pdf
- ISTQBテスト技術者資格制度 Foundation Level Specialist シラバス 自動車ソフトウェアテスト担当者 日本語版 Version 2018.J03
https://jstqb.jp/dl/JSTQB-SyllabusFoundation-AutoSpecialist_Version2018.J03.pdf
- ISTQBテスト技術者資格制度 Foundation Level Specialist シラバス モバイルアプリケーションテスト担当者 日本語版 Version 2019.J01
https://jstqb.jp/dl/JSTQB-SyllabusFoundation-MobileSpecialist_Version2019.J01.pdf
- ISTQBテスト技術者資格制度 Foundation Level Specialist シラバス 性能テスト担当者 日本語版 Version 2018.J01
https://jstqb.jp/dl/JSTQB-SyllabusFoundation-PTSpecialist_Version2018.J01.pdf
- ISTQBテスト技術者資格制度 Advanced Level Specialist シラバス テスト自動化エンジニア 日本語版 Version 2016.J01
https://jstqb.jp/dl/JSTQB-Syllabus.Advanced_TAE_Version2016.J01.pdf
- ISTQBテスト技術者資格制度 Advanced Level Syllabus アジャイルテクニカルテスト担当者 日本語版 Version 1.1.J01
https://jstqb.jp/dl/JSTQB-Syllabus.Advanced_ATT_Version1.1.J01.pdf
- ISTQBテスト技術者資格制度 Foundation Level Extension シラバス アジャイルテスト担当者 日本語版 Version 2014.J02
https://jstqb.jp/dl/JSTQB-SyllabusFoundation-AgileExt_Version2014.J02.pdf