

# 魁!! 智美塾

テストアーキテクチャ解説  
～テストアーキテクチャ設計を  
実践するには～  
智美塾塾長＋塾生一同

# 智美塾設立経緯とその後 壺

- ・ 智美塾とは、JaSST'09 Tokyoのクロージングパネルで提示されたテストライフサイクルを進化させつつ、塾長の厳しい指導のもと、塾生たちが自らの“テスト開発方法論”を目指し活動する梁山泊である。



- ・ JaSST'10 Tokyo、JaSST'11 Tokyoではそれらを考え出す過程の議論を披露した。
- ・ JaSST'12 Tokyoではテスト要求分析 及び テストアーキテクチャ設計の章について解説を行った。

# 智美塾設立経緯とその後式

- ・ 2012年はテストアーキテクチャの成り立ちについて、いくつかのテストアーキテクチャについての具体的な解析を開発者当人を迎えて行った。
- ・ 俎上に乗ったテストアーキテクチャは下記の通り。

TETTAN(\*), あまがききてすとくらぶ(\*),  
VSTeP、ゆもつよメソッド、HAYST法、  
しんすく流テスト開発方法論、  
その他各塾生の持つテスト方法論

(\* ) JaSST' 12 Tokyoテスト設計コンテスト参照

<http://www.jasst.jp/symposium/jasst12tokyo/report.html#plan9>

# 本日の要旨と目的

## テストアーキテクチャの分析と比較

テスト要求分析やテストアーキテクチャ設計で考慮すべき点を挙げ、それぞれのテスト方法論で対応する箇所を明らかにし、それらの強み・弱みなどをパネル形式で議論する。

なお、このセッションは、HAYST法、ゆもつよメソッド、VSTePの各テスト方法論のいずれかについて基礎的な知識があることを前提として話を進める。これら3つの方法論の概略はこの後のページに掲載している。聴講希望者は概略および関連資料をあらかじめ熟読して参加することを登壇者一同希望している。以上！

# 本日の登壇者

塾長：

吉澤 智美

二号生：

鈴木 三紀夫

秋山 浩一(HAYST法)

湯本 剛 (ゆもつよメソッド)

西 康晴 (VSTeP)

# 智美塾訓

1. 塾生はテスト開発方法論を目指し、自ら技術獲得に励む事
2. 塾生は現場を尊び、地に足の着いた技術論を行う事
3. 塾生は品質向上という最終目的を常に意識した活動を行う事

# 智美塾の目的

- ・参加者それぞれが“テスト開発方法論”を構築すべく研鑽を重ねる。
- ・すなわち、“テスト開発方法論”の上位にあるメタな何かを探り理解することによって、自分なりに“テスト開発方法論”を理解し、より良く作りあげたり選択していくことができると思われる。
- ・以上について、参加者全員による議論や共通理解、同意を重ねて進めていく。決して一方通行の講釈ではない。

# はじめに

---

- 目的

- テスト方法論を考える際に戸惑うところを明らかにし、自らテスト方法論を考えられるようにするため。
- よく知られているテスト方法論を独学で適用する際に、誤解を生みやすいところを明らかにし、理解を深めるため。

- 対象者

- テスト要求分析やテストアーキテクチャ設計を現場で実践している人

# 問題提起

---

- 塾生が智美塾に参加している目的は、自分なりのテストの方法論を構築することです。
- 2012年の智美塾では、6人の方法論の発表を聴き、構築の参考にしようと思いました。
- それぞれの方法論は一つの世界を作っているため、発表内容自体は理解できます。
- しかし、各種方法論を比較検討しつつ、自分の方法論に落とし込むまでは、なかなか難しいです。
- 特にテスト要求分析と、テストアーキテクチャ設計は、技術として確立していないため、分かりづらいものでした。

# 3つの方法論のおさらい

# HAYST法



# ゆもつよメソッド

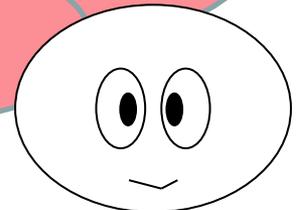


# VSTeP



# テスト要求分析

ここからみっきーさん  
にボタンタッチします



# テスト要求分析

テスト要求分析			
X.1.1	準備	テスト要求分析の準備をする	テスト要求の源泉の把握
X.1.2	獲得	テスト要求を獲得する	テスト観点の抽出、分解、体系化 リスクの識別、評価
X.1.3	分析	テスト要求を分析する	
X.1.4	作成	テスト要求分析成果物を作成する	テスト要求(テスト観点、リスク等)の記述
X.1.5	検証	テスト要求分析成果物を検証する	テスト観点の網羅性の確保

**Test.ssf**

**智美塾**

# テスト要求分析

---

- テスト要求の源泉
  - 機能仕様書の有無
  - 経験の有無
- テスト観点の抽出
  - ステークホルダの位置づけ
  - 重視するもの
  - 観点の洗い出し方
- テスト観点の網羅性の確保

# テスト要求の源泉（機能仕様書の有無）

- 機能仕様書などテストベースが無い状態で、テスト設計をしなくてはならないケースがあります。
- テスト方法論を作るとき、このようなケースを想定するかどうかを考えたいと思います。
- 三つの方法論を比較すると...

方法論	機能仕様書の有無
HAYST 	必須
VSTeP 	なくてもよい
ゆもつよ 	必須

# テスト要求の源泉（経験の有無）

- 方法論を作る／使う目的の一つに、熟練した技術者でなくても、一定以上の品質を確保したいという要望があります。
- 三つの方法論を比較すると...

方法論	経験の有無
HAYST 	必須
VSTeP 	あるとよい
ゆもつよ 	必須

# テスト観点の抽出(ステークホルダ)

- テストは「使う人」の立場で行うものという人がいます。
- 「使う人」を「ユーザ」として捉えるか、  
広く「ステークホルダ」として捉えるかを確認します。
- 三つの方法論を比較すると...

方法論	ステークホルダ
HAYST 	Who(ユーザ)、Whom(ユーザ以外)
VSTeP 	視座分析
ゆもつよ 	暗黙的

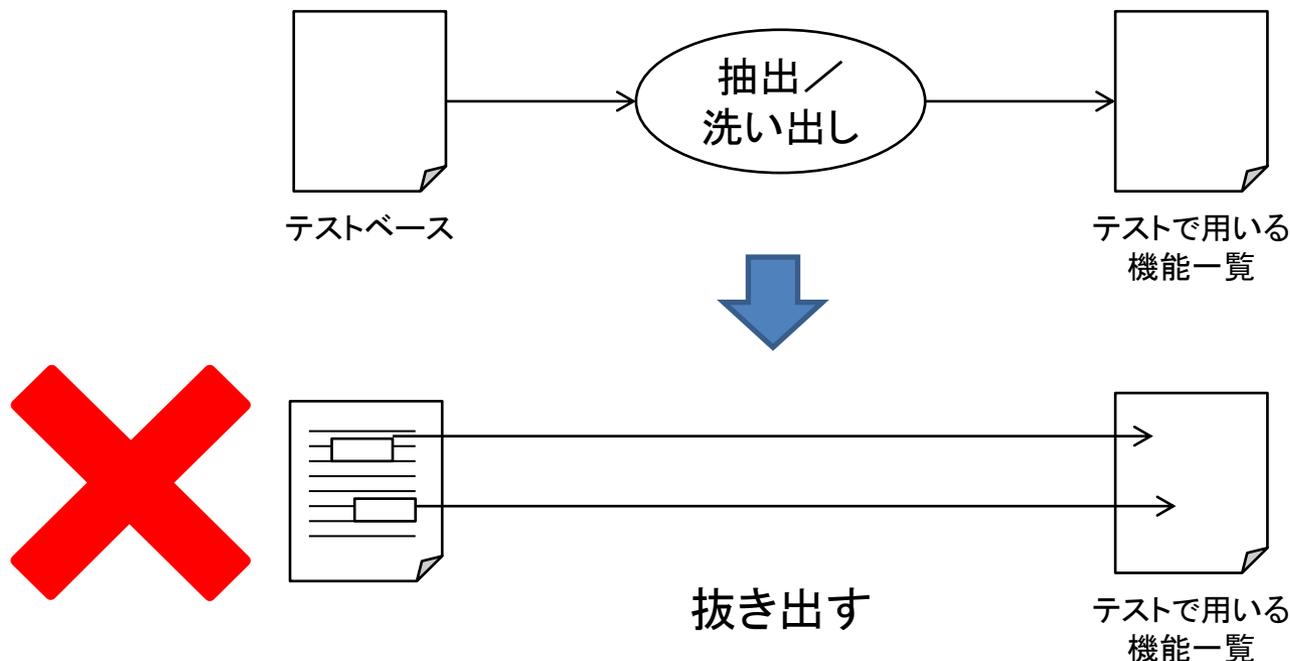
# テスト観点の抽出(重視するもの)

- テスト観点の抽出と言っても、方法論毎に重視するものが異なります。
- 三つの方法論を比較すると...

方法論		重視するもの
HAYST		6W2H、目的機能
VSTeP		テスト観点、関連、階層化／詳細化、ビュースタイル、リファインパターン
ゆもつよ		テスト目的/テストタイプ/テストレベルの割り当てをした一覧表

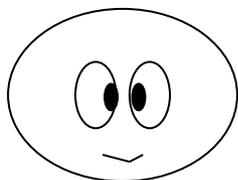
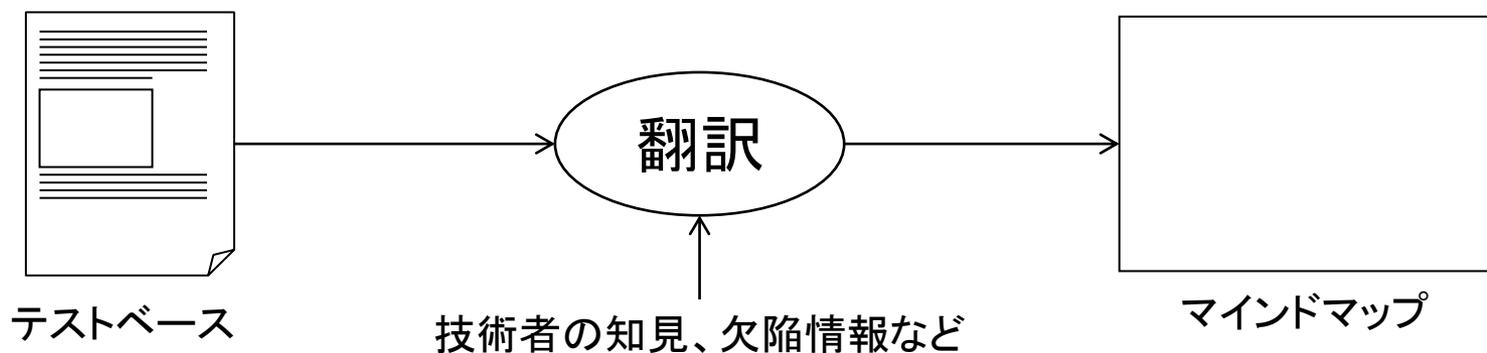
# テスト観点の抽出(観点の洗い出し方)

- テスト観点の「抽出」または「洗い出し」という言い方をします。
- この表現は誤りではありませんが、独学でテスト方法論を学ぶ人は、勘違いをしてしまいがちです。



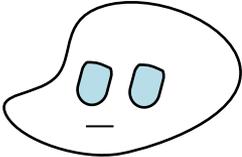
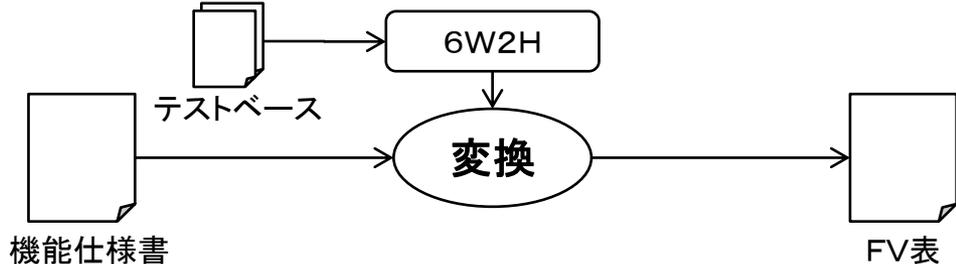
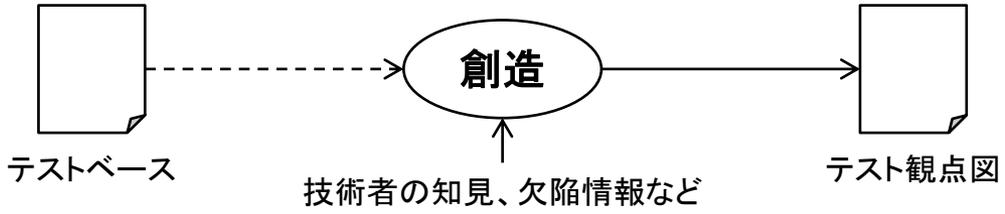
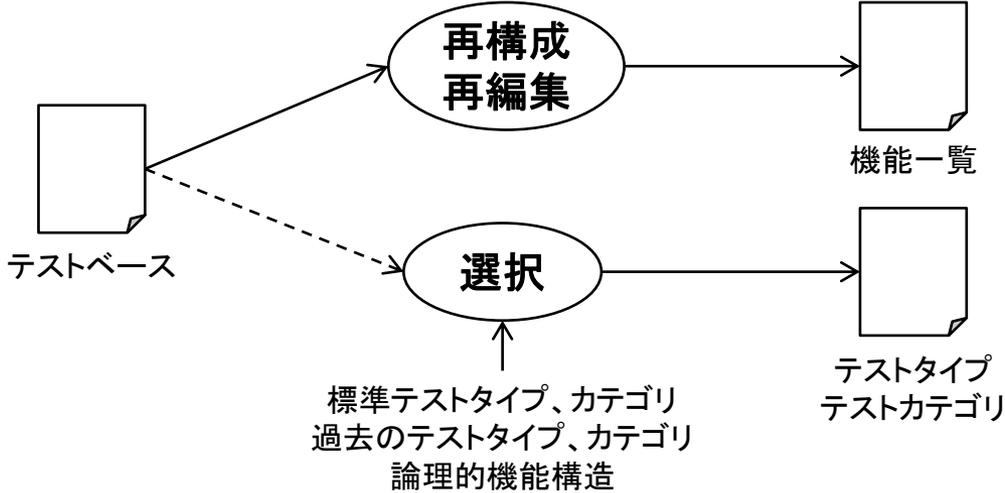
# テスト観点の抽出（観点の洗い出し方）

- 僕の方法は「抽出」ではなく、「翻訳」ですが、三つの方法論はどのようなとらえ方をしているでしょうか？



テストの言葉に翻訳  
しているんだ

# テスト観点の抽出(観点の洗い出し方)

方法論	観点の洗い出し方
<p>HAYST</p> 	
<p>VSTeP</p> 	
<p>ゆもつよ</p> 	

# テスト観点の網羅性の確保

- 洗い出したテスト観点が網羅したかどうかをどのようにチェックしているかを確認してみます。
- 三つの方法論を比較すると...

方法論	網羅のチェック方法
HAYST 	例示・間・対称・類似・外側・破壊・縦横
VSTeP 	ズームイン／アウト、 ステレオタイプ分析、確定分析
ゆもつよ 	特にやっていない (通称、ゆもつよマトリクスで代替?)

# テストアーキテクチャ 設計

塾長に戻ってきました



# テストアーキテクチャ設計

番号	テストアーキテクチャ設計		
X.2.1	準備	テスト要求分析成果物の準備	
X.2.2		テストベースの準備	
X.2.3	獲得	アーキテクチャスタイルに関する要求の獲得	
X.2.4	分析	アーキテクチャスタイルに関する要求の分析	
X.2.5		アーキテクチャスタイルの選択	
X.2.6		テスト全体構造の設計	★
X.2.7		テスト全体バランスの調整	★
X.2.8		テスト環境の構築方針・方法の検討	
X.2.9		テスト詳細設計の指針・原則の検討	
X.2.10		作成	テストアーキテクチャ設計成果物の作成
X.2.11	検証	テストアーキテクチャ設計成果物の検証	

6と7を  
取り上げます

Test.ssf



# テストアーキテクチャ設計のお題

---

- 今回は、アーキテクチャを構成するもののうち、構造をターゲットとして議論します。
- テストレベルをどう考えていますか？
- テストアーキテクチャを表現する際に、どんな図・表を用いますか？
- テストアーキテクチャの構成要素と構成要素間の関係(構造)を教えてください。

# テストレベルに対する考え方

- テストレベルはJSTQBによれば、4レベルあります。それよりも多い組織も少ない組織もちろんあります。
- 三つの方法論を比較すると...

方法論	テストレベルの考え方
HAYST 	<u>組織レベル</u> で標準を持っている。 (テストアーキテクチャを作成する前におおよそ決まっている)
VSTeP 	テストレベルが必要であれば、 <u>テストアーキテクチャ設計時に最適なテストレベルを作る。</u>
ゆもつよ 	<u>テスト計画時にテストレベルを決める。</u> (テストアーキテクチャを作成する前におおよそ決まっている)

# テストアーキテクチャの表現方法

- テストアーキテクチャの表現方法は何ですか？
- 三つの方法論を比較すると...

方法論	テストアーキテクチャの表現方法
HAYST 	ラルフチャート: 図 FV表: テーブル
VSTeP 	テスト観点モデル テストコンテナモデル
ゆもつよ 	テスト設計一覧

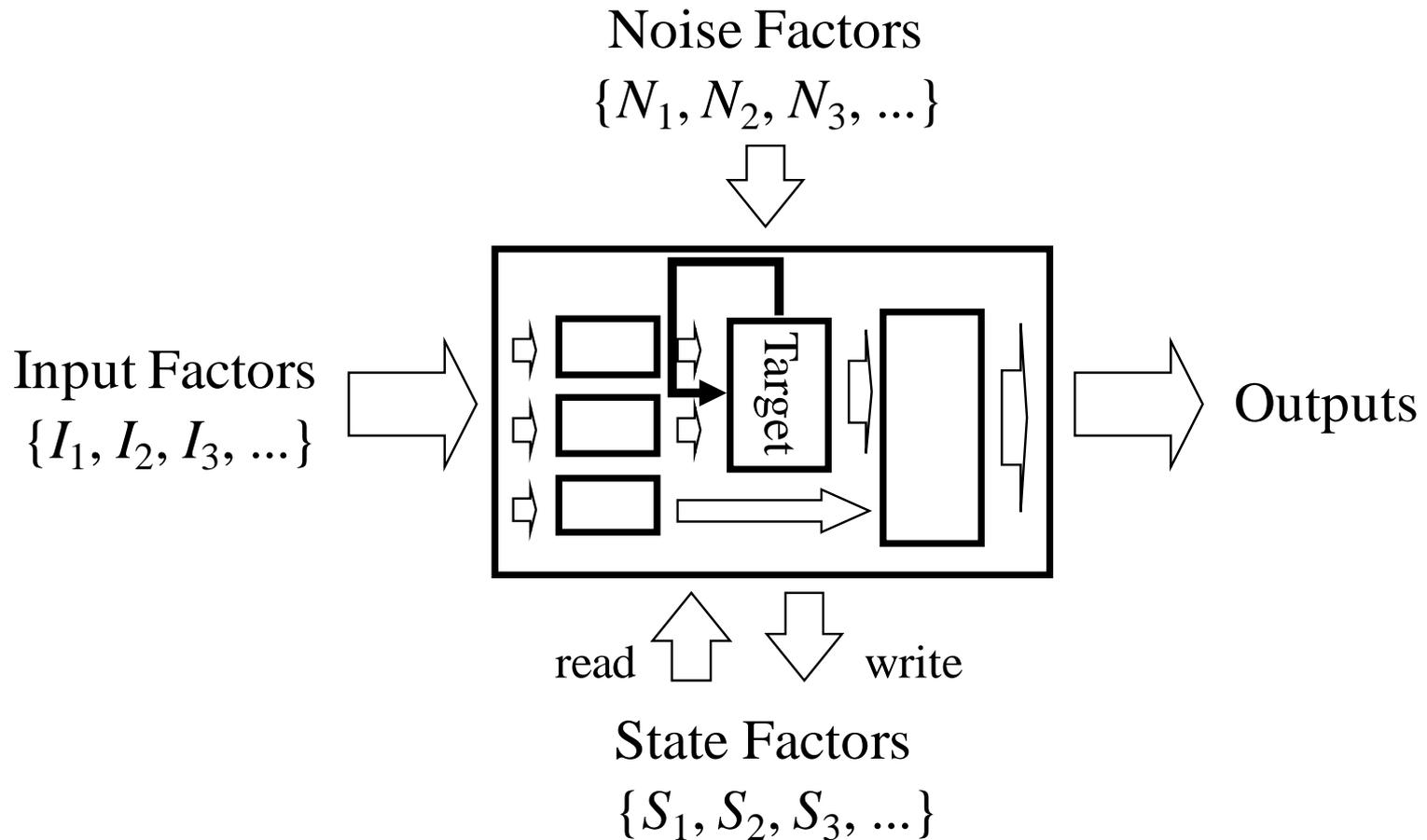
# テストアーキテクチャの構成要素

- テストアーキテクチャを構成する要素は何ですか？
- 三つの方法論を比較すると...

方法論	テストアーキテクチャの構成要素															
HAYST 	ラルフチャート(参照:p.30)															
VSTeP 	テスト観点、テストコンテナ															
ゆもつよ 	機能名、機能小分類、テストタイプ、テストカテゴリ、 テスト条件、仕様項目、期待結果、テスト設計方針、 主なテストケース条件 <table border="1" data-bbox="625 1172 1831 1258"> <thead> <tr> <th>機能名/ 機能小分類</th> <th>テスト タイプ</th> <th>テスト カテゴリ</th> <th>テスト条件と すべき仕様項目</th> <th>期待する結果/ 確認内容</th> <th>テスト設計 方針</th> <th>主なテストケース 条件</th> </tr> </thead> <tbody> <tr> <td> </td> </tr> </tbody> </table>		機能名/ 機能小分類	テスト タイプ	テスト カテゴリ	テスト条件と すべき仕様項目	期待する結果/ 確認内容	テスト設計 方針	主なテストケース 条件							
機能名/ 機能小分類	テスト タイプ	テスト カテゴリ	テスト条件と すべき仕様項目	期待する結果/ 確認内容	テスト設計 方針	主なテストケース 条件										

# ラルフチャート

- FV表で認識した目的機能



# テストアーキテクチャの構成要素間の関係

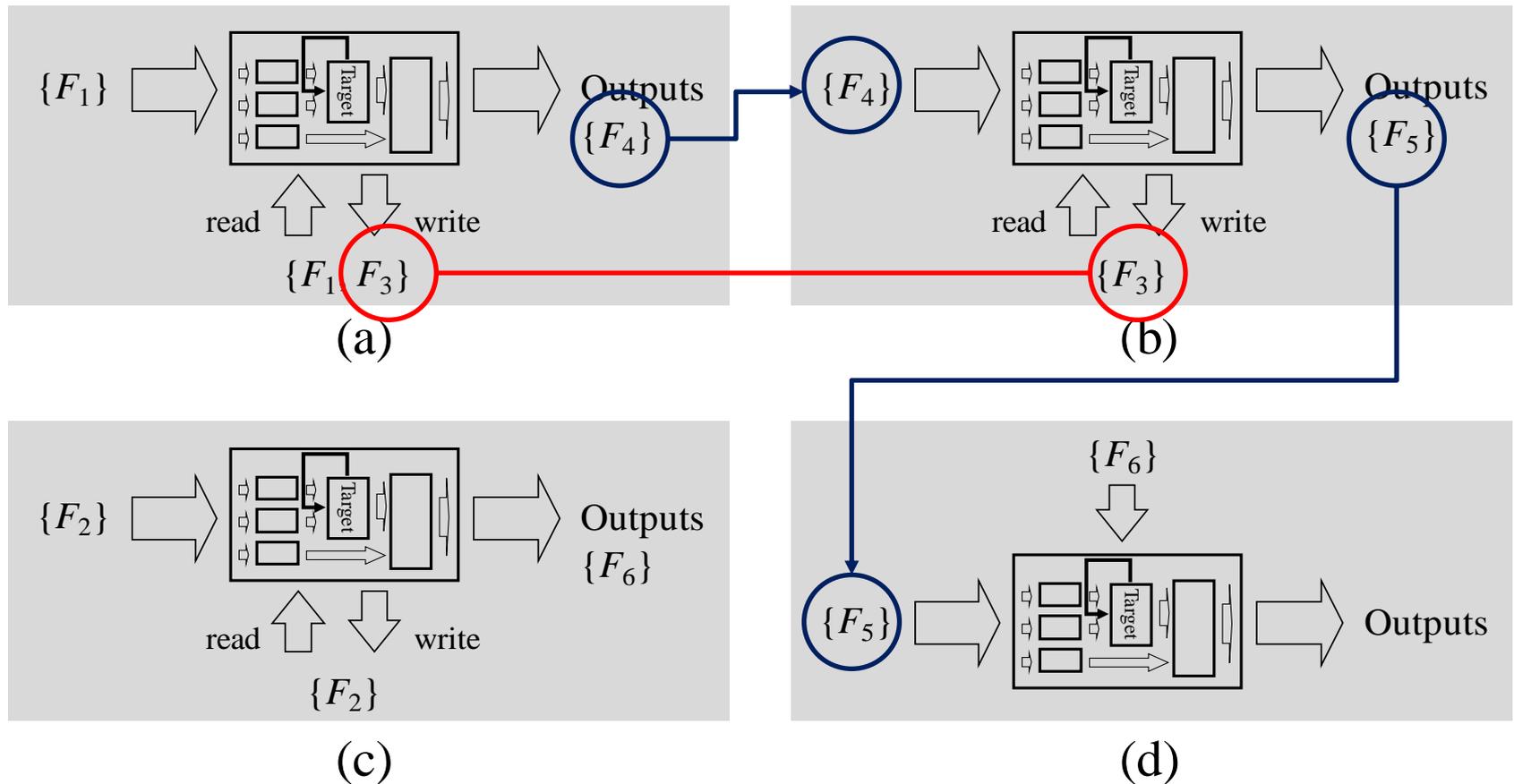
- テストアーキテクチャの構成要素間の関係はどのように定義されていますか？
- 三つの方法論を比較すると...

方法論	テストアーキテクチャの構成要素の関係
HAYST 	ラルフチャートの組合せ(参照:p.32)
VSTeP 	縦(is-a、has-a、属性化、原因-結果)、横(関連)、包含関係、依存関係、合成関係(参照:p.33)
ゆもつよ 	p.34参照

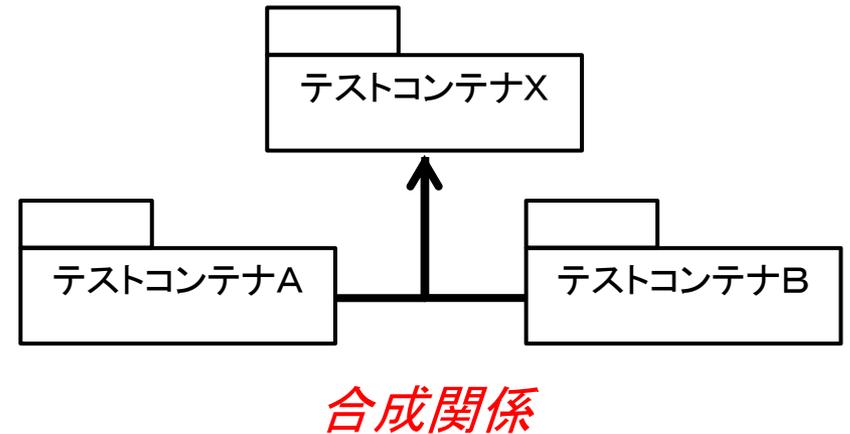
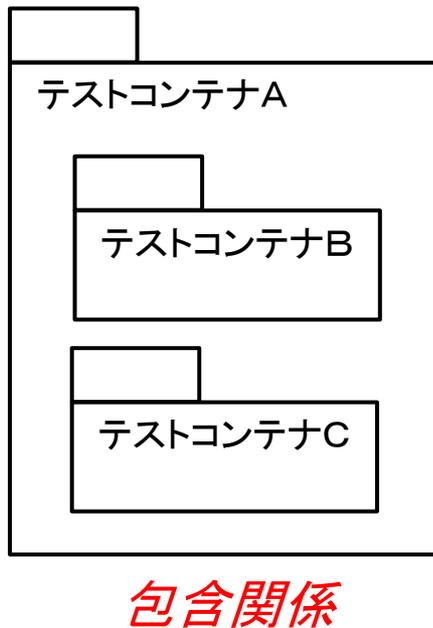
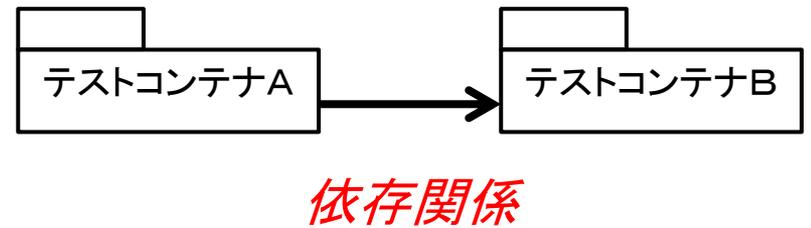
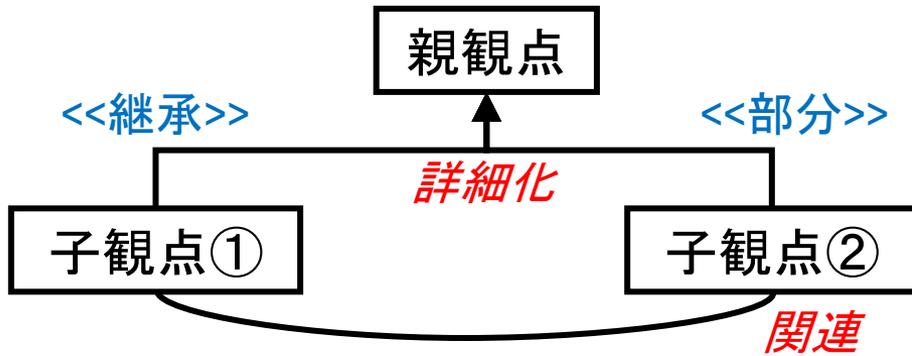
# ラルフチャートの組合せ

- 目的機能の構造

- (a)(b)組合せ連結、(a)→(b)→(d)順序構造



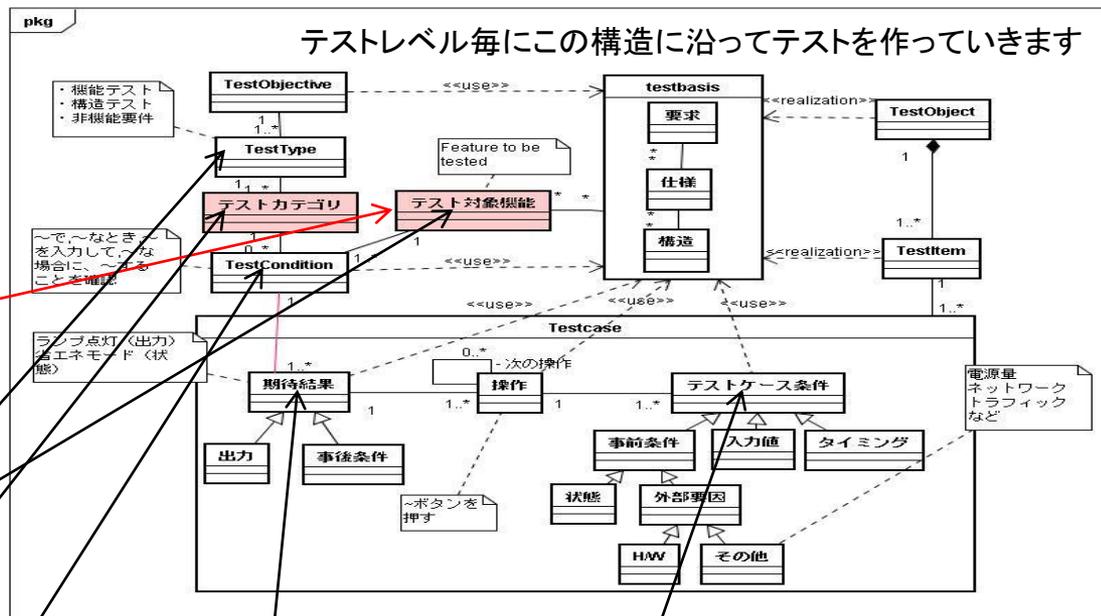
# テスト観点間・テストコンテナ間の関係



# ゆもつよのテストアーキテクチャ要素間の関係

計画時に作る一覧(ここでテストレベルは分離される)

テストの目的	テスト対象アイテム	テストレベル	テストタイプ	役割
MT1 ・自身で書いたコードが想定どおり動くことを確認	モジュール	単体テスト	機能テスト 構造テスト	プログラマ
MT2 ・既存モジュール含め、コンポーネント単位でビルドし機能が正しく動くことを確認	コンポーネント	コンポーネントテスト	機能テスト	開発チーム
MT3 ・コンポーネント間シーケンスの確認 ・S/Wシステムテストへのリリース判定	複数ブロック (評価用試作機)	コンポーネント統合テスト	機能テスト 構造テスト	システム 検証チーム
MT4 ・ソフトウェア全体の機能要件・非機能要件の確認 ・システムテストへのリリース判定	S/Wシステム (量産試作機)	S/W		
MT5 ・H/W含めた製品としての機能要件・非機能要件の確認 ・出荷判定	システム	シス		



テスト対象機能の一覧がゆもつよで作る機能一覧です

下記一覧とクラス図の関係

機能名/ 機能小分類	テスト タイプ	テスト カテゴリ	テスト条件と すべき仕様項目	期待する結果/ 確認内容	テスト設計 方針	主なテストケース 条件
---------------	------------	-------------	-------------------	-----------------	-------------	----------------

## おわりに

---

テストアーキテクチャを行う際、  
どんな表や図を書くか、  
と言うことは見ればわかりますが、  
意図とか全体をどうとらえるかといったところは  
なかなか読み取れません。

今回は方法論を作った人でないと  
答えられないような考え方を比較しました。

## おわりに2

- 自分なりの方法論を作るために
  - ちゃんと目的、意図を理解して方法論を作りましょう。
  - 既存の方法論は、もともとの考え方に差があるので、  
スコープとかやり方を自分たちに合ったものを作る際に  
選ぶようにしましょう。
  - 個々の方法論は進化をしています。  
ソフトウェアの方法論は100点満点なものはありません。  
皆さんの方法論もまず作ってみて、100点でなくても、  
使ってみてブラッシュアップさせましょう。
  - 各方法論にも強弱があります。自分との相性もあります。
- そういったところを見極めたところで皆さんの方法論を考えるきっかけになればと思います。

# 来たれ！智美塾へ

- ・ 月一回程度、夜間に開催する(東京地区)。
  - ・ 参加案内はTEF(テスト技術者交流会ML)およびASTER Webへ提示
- ・ 参加希望者は参加案内に記載される参加要件(智美塾への参加動機、あるいはその回のお題について四百字以上で記述すること、など)をよく読み、応募すること。
  - ・ 他に参加資格としての要件はない。  
腕に覚えのある者なら誰でも応募可能である。
- ・ 参加メールを塾長などで審議し、資格ありと認められた猛者のみ参加を許される。