

テスト観点に基づくテスト開発方法論 VSTePの概要(ダイジェスト版)



2013/1/30(水) JaSST2013東京 智美塾セッション

電気通信大学 大学院情報理工学研究科

総合情報学専攻 経営情報学コース

西 康晴 (Yasuharu.Nishi@uec.ac.jp)

テスト設計には実に多くの観点が必要：組込みの例

● 機能：テスト項目のトリガ

- ソフトとしての機能
 - » 音楽を再生する
- 製品全体としての機能
 - » 走る

● パラメータ

- 明示的パラメータ
 - » 入力された緯度と経度
- 暗黙的パラメータ
 - » ヘッドの位置
- メタパラメータ
 - » ファイルの大きさ
- ファイルの内容
 - » ファイルの構成、内容
- 信号の電氣的ふるまい
 - » チャタリング、なまり

● プラットフォーム・構成

- チップの種類、ファミリ
- メモリやFSの種類、速度、信頼性
- OSやミドルウェア
- メディア
 - » HDDかDVDか
- ネットワークと状態
 - » 種類
 - » 何といくつつながっているか
- 周辺機器とその状態

● 外部環境

- 比較的变化しない環境
 - » 場所、コースの素材
- 比較的变化しやすい環境
 - » 温度、湿度、光量、電源



テスト設計には実に多くの観点が必要：組み込みの例

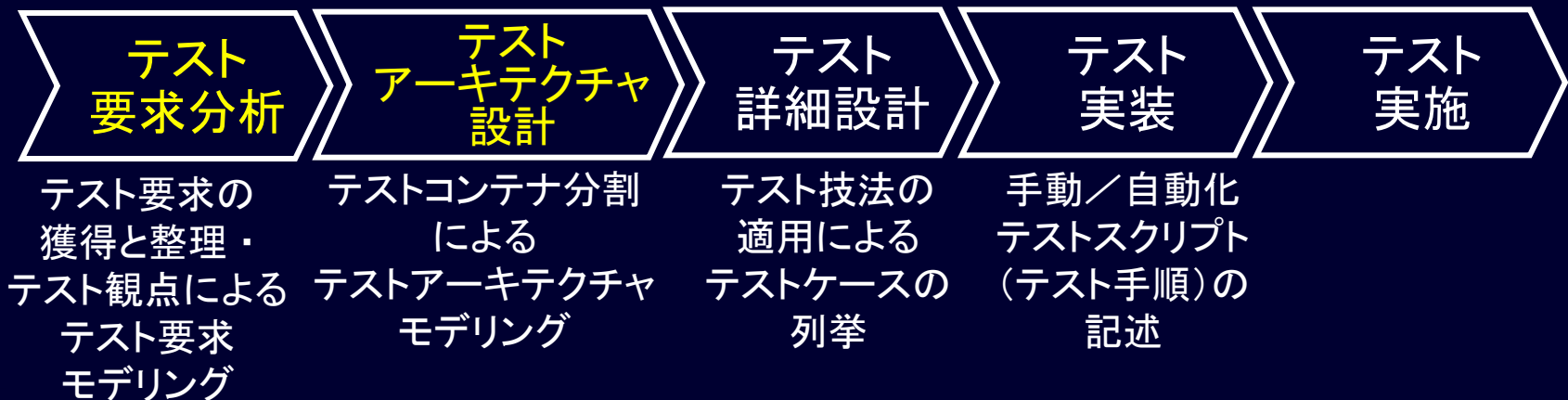
- **状態**
 - － ソフトウェアの内部状態
 - » 初期化処理中か安定動作中か
 - － ハードウェアの状態
 - » ヘッドの位置
- **タイミング**
 - － 機能同士のタイミング
 - － 機能とハードウェアのタイミング
- **組み合わせ**
 - － 同じ機能をいくつカブせるか
 - － 異なる機能を何種類組み合わせるか
- **性能**
 - － 最も遅そうな条件は何か
- **信頼性**
 - － 要求連続稼働時間
- **GUI・操作性**
 - － 操作パス、ショートカット
 - － 操作が禁止される状況は何か
 - － ユーザシナリオ、10モード
 - － 操作ミス、初心者操作、子供
- **出荷先**
 - － 電源電圧、気温、ユーザの使い方
 - － 言語、規格、法規
- **障害対応性**
 - － 対応すべき障害の種類
 - » 水没
 - － 対応動作の種類
- **セキュリティ**
 - － 扱う情報の種類や重要度
 - － 守るべきセキュリティ要件

非常に多くの観点を網羅的に設計する必要がある

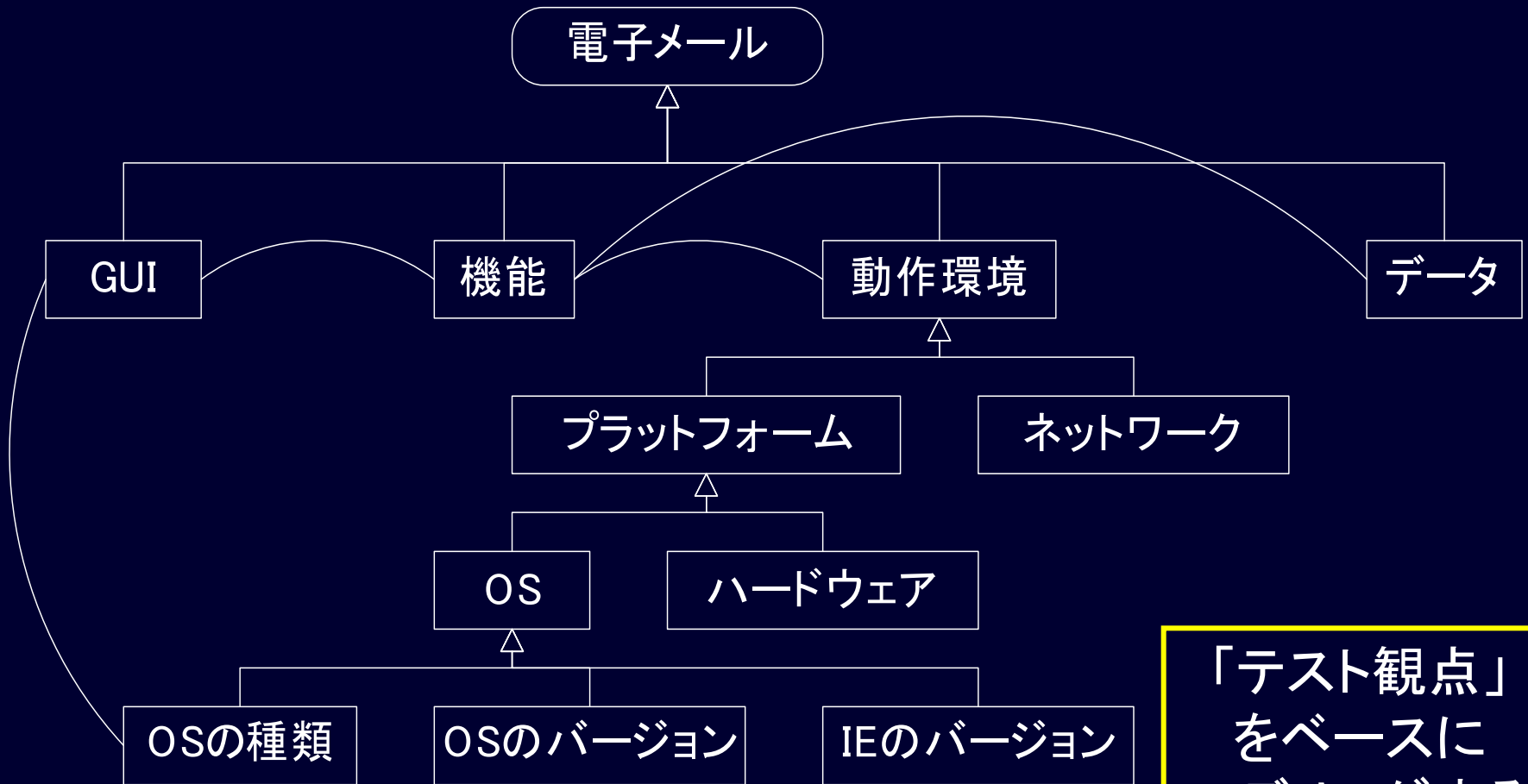
テスト観点に基づくテスト開発方法論：VSTePの概要

- VSTeP(Viewpoint-based Software Test Engineering Process)とは、テスト観点のモデリングを核としたテスト開発方法論である
 - － テスト観点を核にして、テストの要求からテストケース、テスト手順までをシームレスに開発していく方法論である
 - ≫ テスト要求分析やテストアーキテクチャ設計など、軽視しがちなテストの上流工程に力を入れることができ、質の高いテスト設計やレビュー、経験の蓄積が可能になる
 - ≫ ソフトウェア開発と同様にモデリングスキルを求められるが、パターンやSPLE、モデル駆動開発などソフトウェアエンジニアリング的な技術をテストに応用しやすい
 - ≫ 既存のテストをリバースエンジニアリングできるのでテストの再利用や改善もしやすい

【VSTePの流れ】



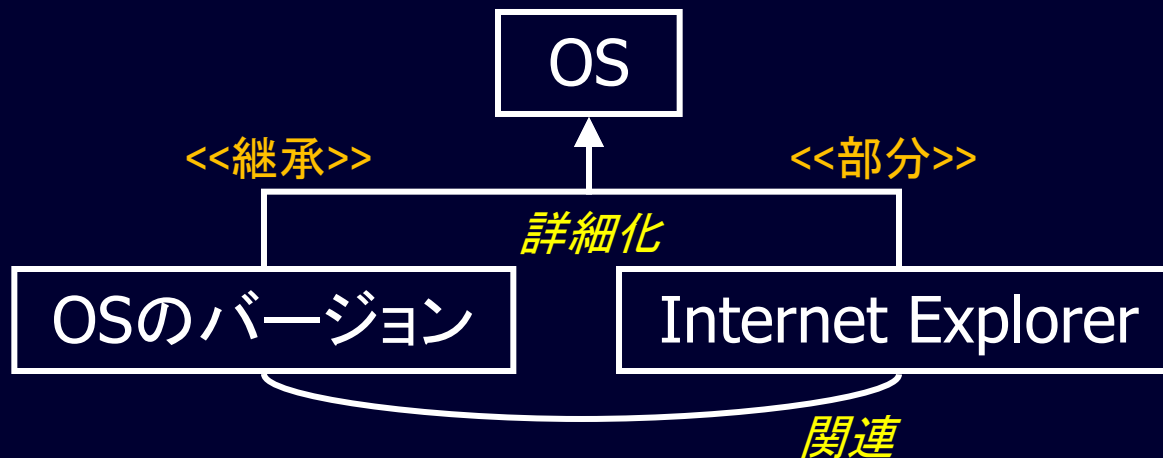
NGT (Notation for Generic Testing) のテスト観点図の例



「テスト観点」
をベースに
モデリングする

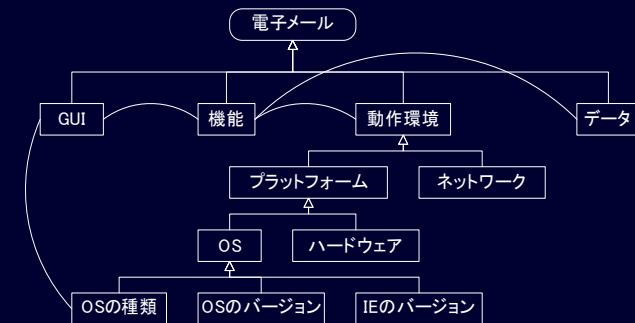
テスト観点と関連、ステレオタイプ

- テスト観点は2つの基本的な関係を持つ
 - 詳細化(Hierarchy relationships)
 - » テスト観点を階層的に詳細化してテスト条件を導き出し、直線の矢印で表す
 - » 継承、合成集約(部分)、属性化、原因-結果など
詳細化すべき理由をステレオタイプで表すこともできる
 - 関連(Interaction relationships)
 - » 組み合わせることでテストすることが必要なテスト観点を示し、曲線で表す
 - » 組み合わせるべき理由をステレオタイプで表すこともできる
- 関係の種類を“<<ステレオタイプ>>”を用いて表すと分かりやすい



テスト要求分析

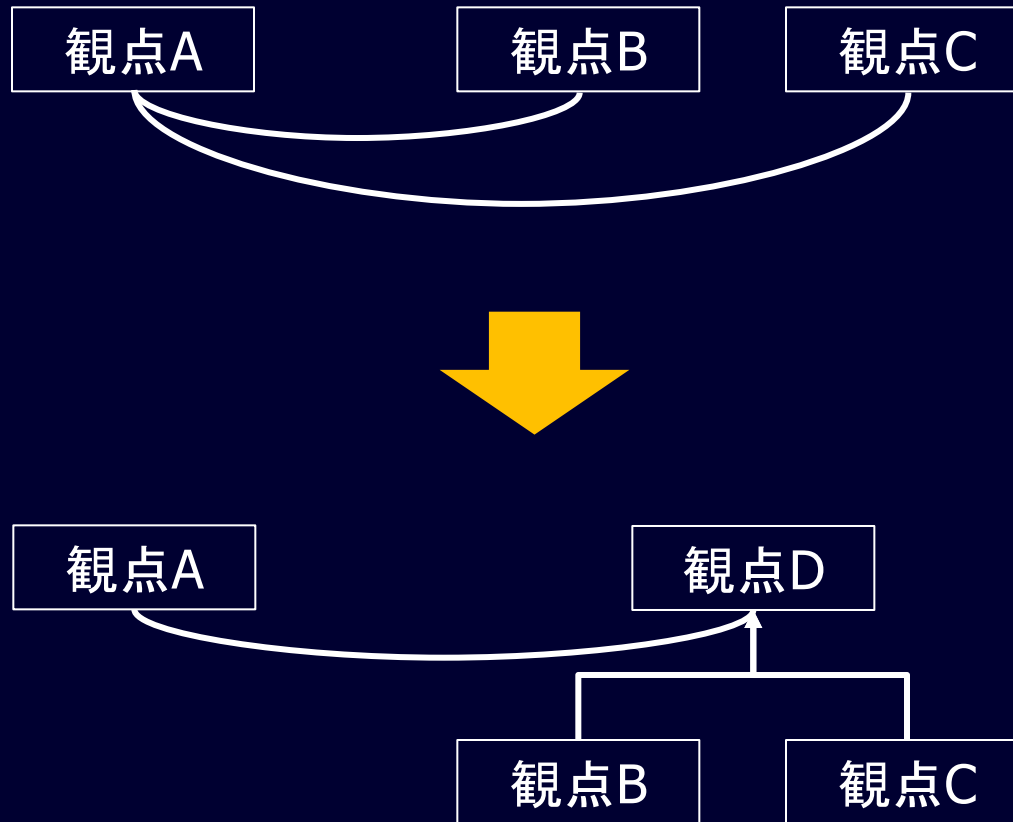
- **何をテストしなくてはいけないのか、を「テスト観点」と呼ぶ**
 - テスト観点とは、テスト対象のテストすべき側面や達成すべき性質を示す
 - ≫ 品質特性やユーザシナリオ、ハザード、負荷、構成、状態、処理経路、ボトルネック、バグのパターン、など様々なテスト観点がある
 - ≫ テスト観点を具体化するとテストケース(の一部)になる
 - ≫ 顧客がイメージするシステムの完成像や利活運用、仕様書や開発成果物、現在のプロダクトの出来、過去のテストケースやバグなどからテスト観点を抽出する
- **テスト観点の階層モデルを作成し、テスト観点図として表現する**
 - テストで考慮すべき観点を一覧でき、俯瞰的かつビジュアルに整理できる
 - ≫ 最下層のテスト観点(ボトムビューポイント)はテスト詳細設計でテスト条件や同値クラスとなる
 - ≫ 離れたテスト観点を組み合わせてテストしたい場合は「関連」として横線でつなぐ
 - ≫ NGT(Notation for Generic Testing)という記法で定義されている
 - ≫ マインドマップの各種ツールを用いると便利である
 - テスト観点モデルをリファインすることが重要である
 - ≫ テスト観点モデルは組織やエンジニアによって様々であり、正解はない
 - ≫ ツリー型の記法なので、Excelで表を埋めるよりも“エンジニアリング”っぽい



テストアーキテクチャ設計

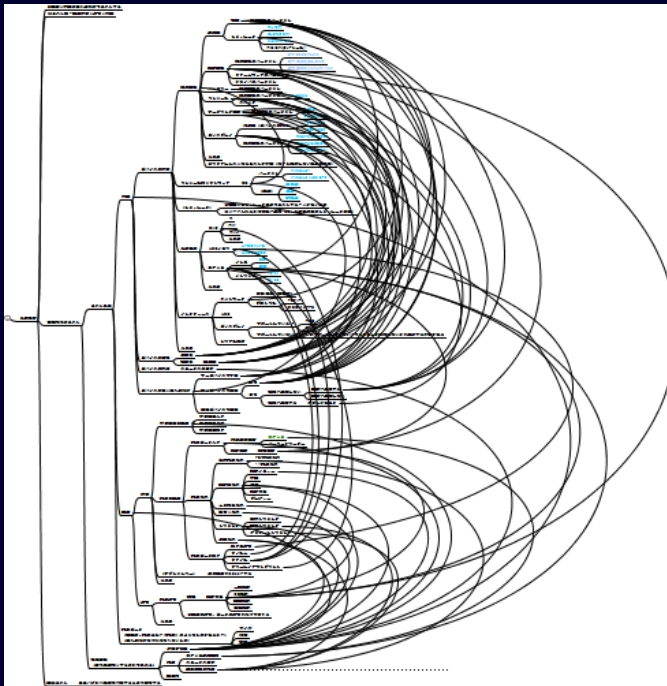
- **テスト要求分析で得られたテスト要求モデルを
テスト詳細設計・実装・実行しやすいように分割・整理する**
 - VSTePにおけるテストアーキテクチャとは、テストの全体像を
テストタイプやテストレベル、テストサイクル(の抽象度)で表したものである
 - » VSTePにおけるテストアーキテクチャ設計では、
テスト観点モデルのリファイン、テストコンテナ分割、テストフレーム構築を行う
 - » テストにも保守性など特有の「テスト品質特性」があるが、
どのようなテスト品質特性を重視するかによって異なるテストアーキテクチャとなる
 - **テストコンテナ分割**: テスト観点群をまとめて、テストケース群全体を俯瞰する
 - » 複数のテスト観点を一つのテストコンテナにまとめる
 - » テストコンテナは、テストレベルやテストタイプ、テストサイクルを意味する
 - » テストケース群全体がコンテナの集合として大まかに示されるので俯瞰性が高まる
 - » コンテナ間の結合度や凝集度を考慮してテストアーキテクチャを設計する必要がある
 - » 各テストコンテナに含めるテスト観点が明らかになるため、現場で経験的に用いている
テストレベルやテストタイプ、テストサイクルの妥当性を評価できるようになる
 - **テストフレーム構築**: テストコンテナをテストケースの構造に合わせる
 - » テストコンテナに含まれたテスト観点(テスト目的由来やテスト対象由来の観点など)
を対応づけてテストフレーム(テストケース構造)を明らかにし、
テスト詳細設計を行う際のテンプレート(テストケース構造)を作成する

関連のリファインのパターンの例：親観点統合

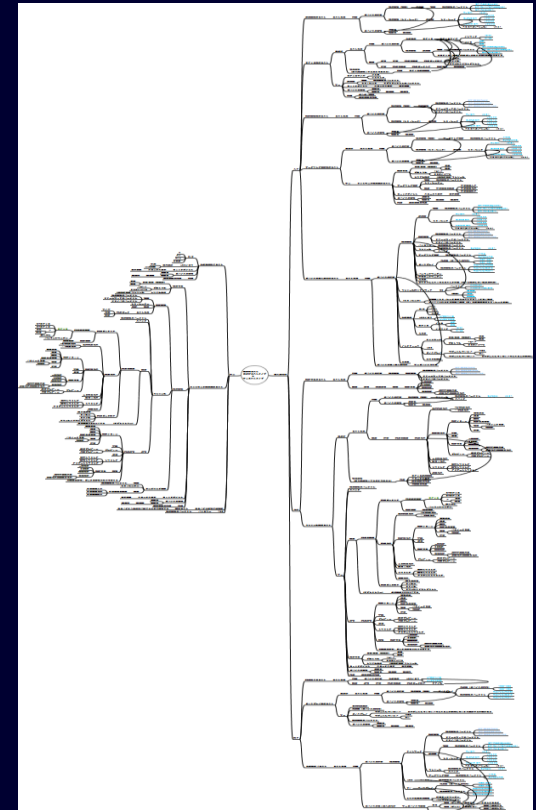


リファインや関連の整理、コンテナ分割を行った例

- 見るからにテスト詳細設計しやすくなったと感じられる
 - 右図では2~3層目のサブツリーがテストコンテナを示している



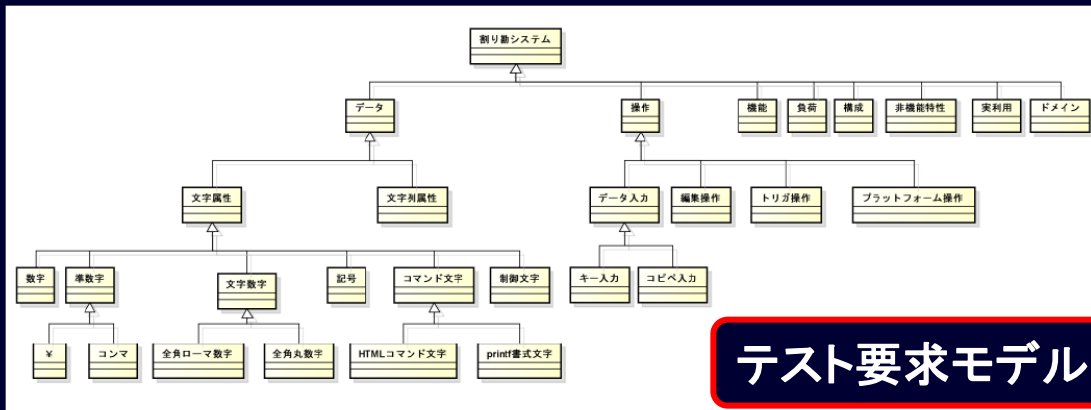
リファイン



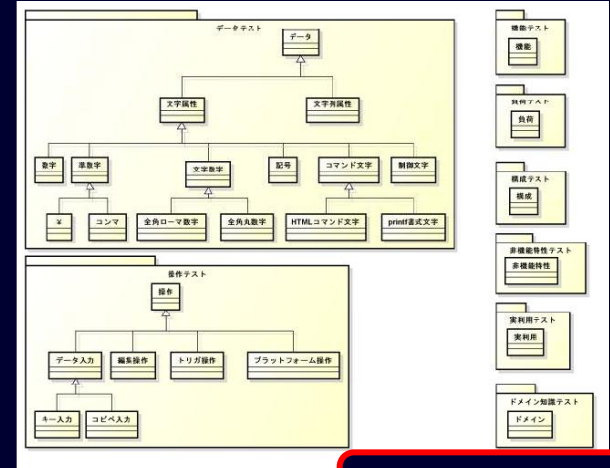
テストスイートの品質特性

- **テストスイートにも品質特性が考えられる**
 - アーキテクチャを検討するほど大規模で複雑なテストスイートには、テストスイート自身の品質特性を考慮しなければならない
 - しかしテストスイートの品質特性についてはほとんど議論されていない
 - ソフトウェアの品質特性は十分議論されている
 - » ISO/IEC9126sの品質特性: 機能性、信頼性、使用性、効率性、保守性、移植性など
 - » ソフトウェアの品質特性はテストスイートの品質特性と同じではないが、参考程度にはできるかもしれない
- **テストスイートの品質特性の違いによって、異なるテストアーキテクチャを設計する必要が生じる**
 - 保守性、派生容易性、自動化容易性、網羅保証性、ピンポイント性などがあるかもしれない

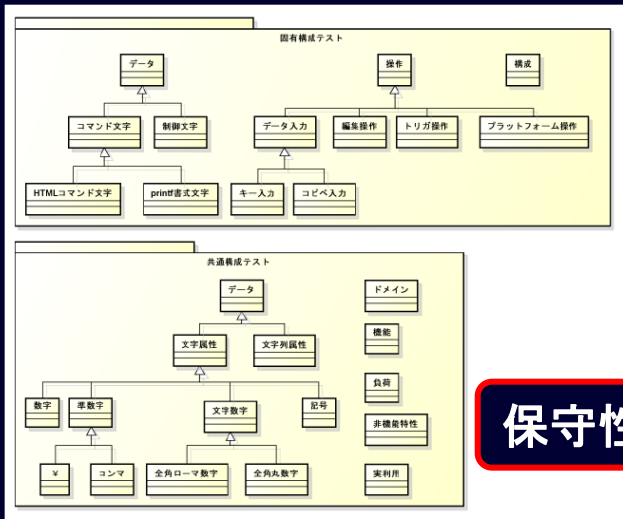
テストスイートに求められる品質特性によって異なる テストアーキテクチャ(コンテナ分割)の例



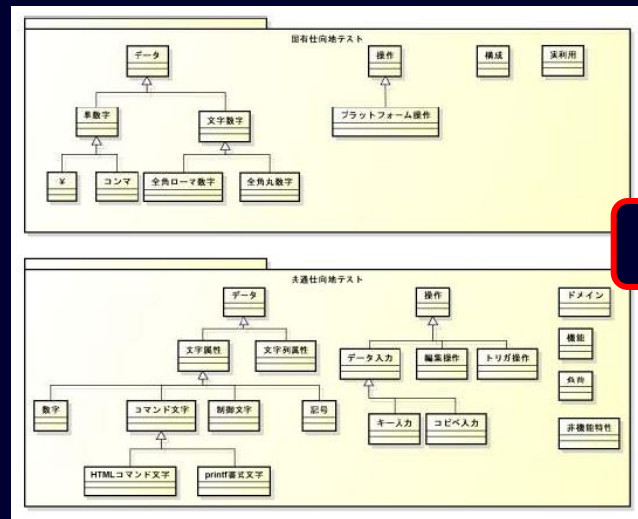
テスト要求モデル



単純に分割

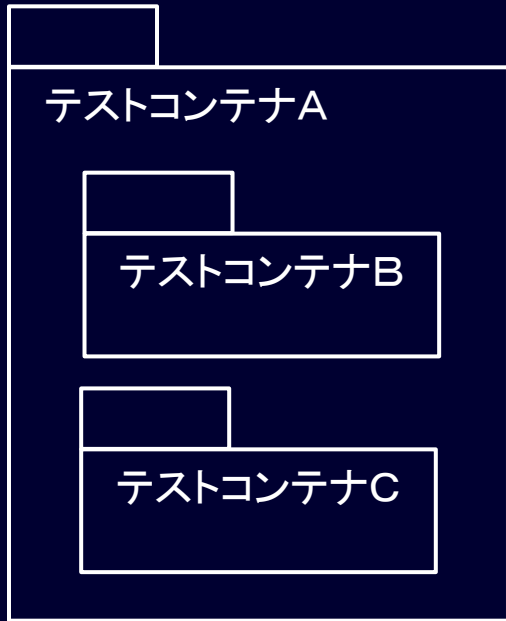


保守性重視

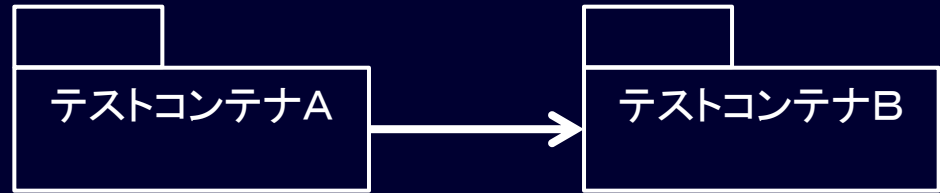


国際化重視

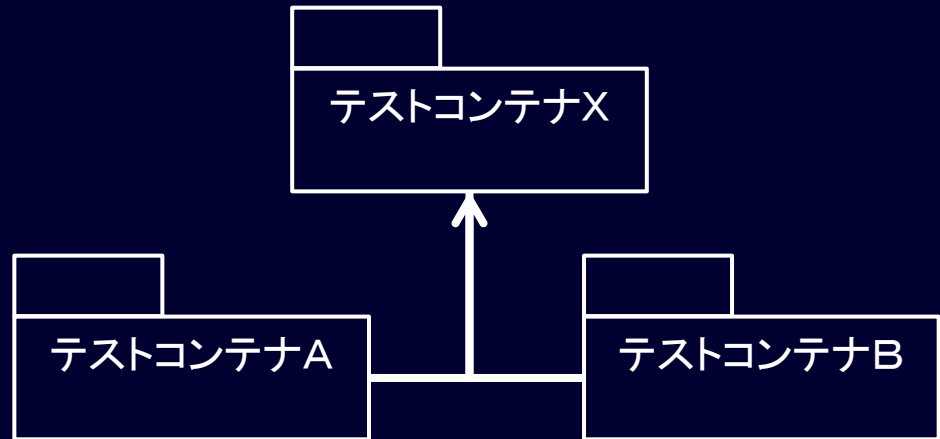
テストコンテナ間の関係



包含関係

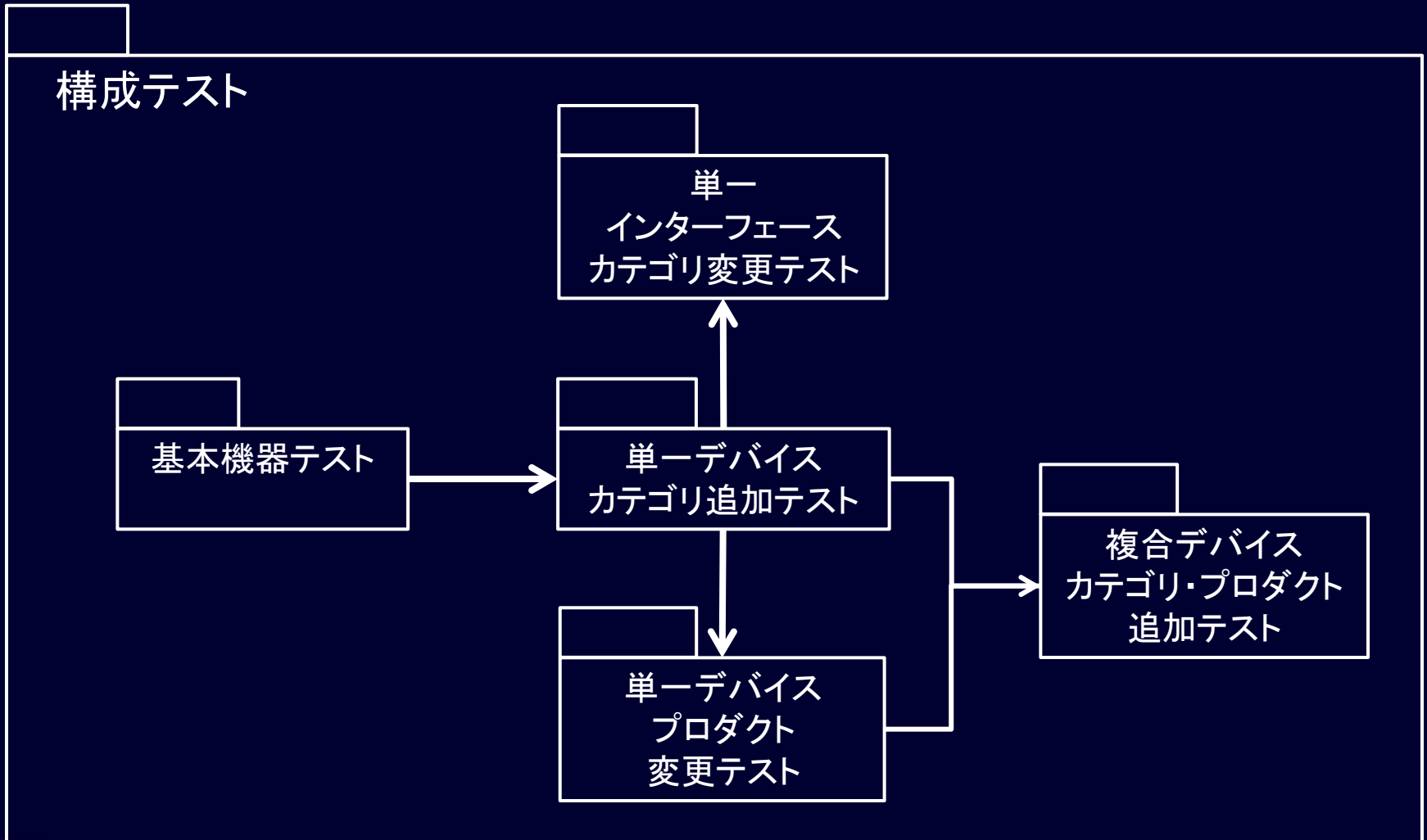


依存関係



合成関係

テストコンテナ間関係の例：構成テストタイプの例



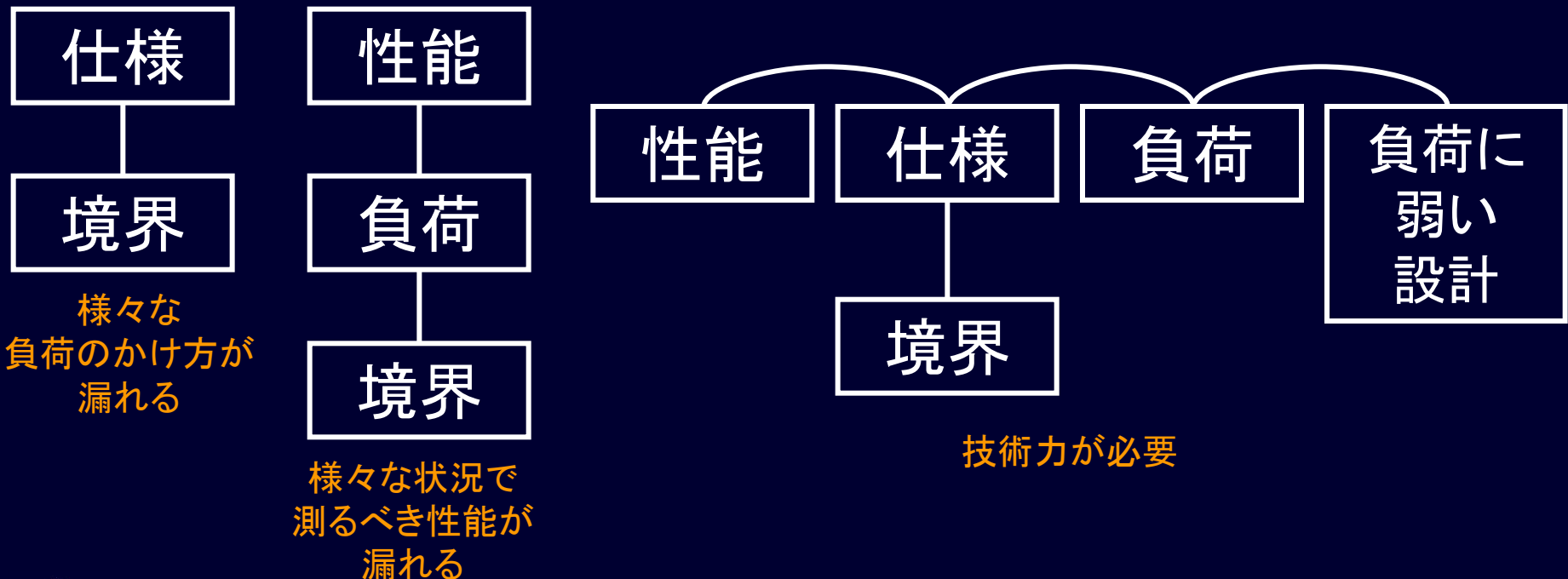
テスト詳細設計・テスト実装

- **テスト観点がボトムビューポイントまで詳細化され、テストフレームが構築されると、テスト詳細設計を行う**
 - テストフレームやボトムビューポイントに対応するテスト詳細設計モデルを定め、網羅アルゴリズムと網羅基準にしたがってテストケースを生成する
 - 例) 状態モデルに対して、C0の遷移網羅基準で、深さ優先探索法を用いて生成する
 - テストフレームやボトムビューポイント、網羅アルゴリズム、網羅基準が明確であれば、モデルベースドテストとしてテストケース生成を自動化できる可能性が高い
 - テストケースに対応するテスト観点を常に明確にすることができるので、目的のよく分からない漫然と設計／再利用されるテストケースを撲滅できる
- **テストケースが生成できたら、テスト実装を行う**
 - テスト対象のシステムやソフトウェアの仕様、テストツールなどに合わせてテストケースをテストスクリプトとして具体化する
 - 手動テストスクリプトの場合もあるし、自動化テストスクリプトの場合もある
 - 複数のテストケースを1つのテストスクリプトに「集約」して効率化を行う
 - 同じ事前条件やテスト条件、テストトリガのテストケースを一つに集約する
 - 現場では経験的かつ暗黙的に行っているがVSTePでは明示的に集約を行う



Reverse VSTeP: 既存の資産を活かしながら移行する

- いきなり新たな方法論を一から導入するのは難しいので、既存の資産を少しずつリバーシしながら導入すると無理がない
 - テスト観点をリバーシするとテスト設計者の意図が透けて見えてしまうので、既存の方法によるテスト設計の改善や技術向上にもつながらる



まずは使ってみて下さいな



電気通信大学 西 康晴
Yasuharu.Nishi@uec.ac.jp