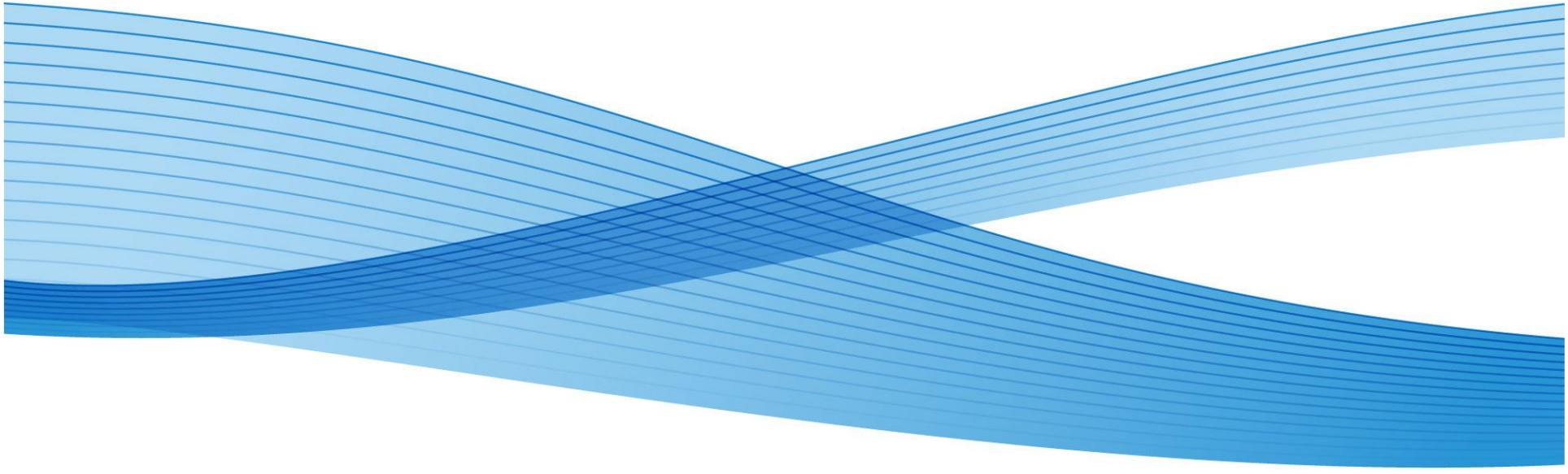
■JaSST 2013 四国

テスト分析・テスト設計入門



富士ゼロックス株式会社　ソリューション・サービス開発本部
秋山 浩一

FUJI Xerox

自己紹介

- 1985年4月　富士ゼロックス入社
  - 現在はHAYST法のコンサルティング業務に従事
- NPO ソフトウェアテスト技術振興協会（ASTER）　理事
  - JaSST東京実行委員（2003年～）
    日本最大のテストシンポジウム1600名の動員
  - JSTQBステアリング委員（2006～）
    テスト技術者資格認定を行う国際組織日本支部
- 日科技連　SQiP研究会　委員長（2011年～）
- Wモデル研究会　主査（2011年7月～）
  - 電通大 西康晴先生、NEC 吉澤智美氏、MRT 鈴木三紀夫氏
- ISO/IEC JTC 1/SC7 WG26委員（2009～）
  ソフトウェアテストの国際標準ISO29119策定中
- 共著書『ソフトウェアテストHAYST法入門』(日科技連出版社)
  (2008年度 日経品質管理文献賞受賞)
  著書「ソフトウェアテスト技法ドリル」(全国の勉強会で活用)
- 博士（工学）

FUJI Xerox

本コースの狙いとアジェンダ

- テストの作り方を理解することが狙い
  - ソフトウェアテストとは
  - テストプロセス
  - 演習
    - 6W2Hで分析
    - ユーザーストーリーで価値を理解
    - FV表で整理
    - ラルフチャートで因子を発見
  - まとめ

ソフトウェアテストとは

- ソフトウェアテストの目的
- 従来のテスト方法（初期テスト）

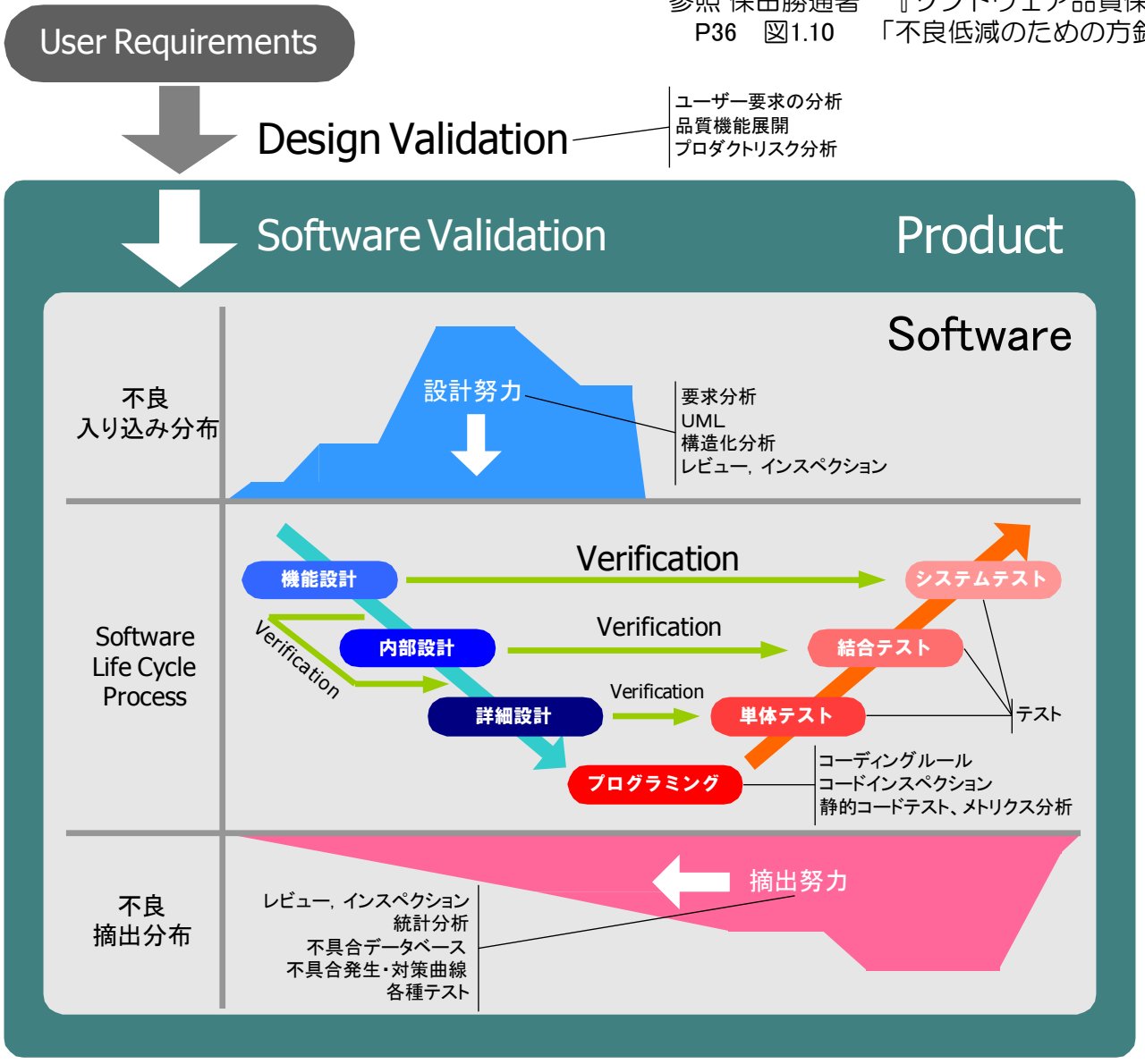
V字モデルによる信頼性マップ

参照 保田勝通著　『ソフトウェア品質保証の考え方と実際』
P36　図1.10　「不良低減のための方針」（1995年11月）

① 不具合を
作り込まない努力

② プロセスで抑
える努力

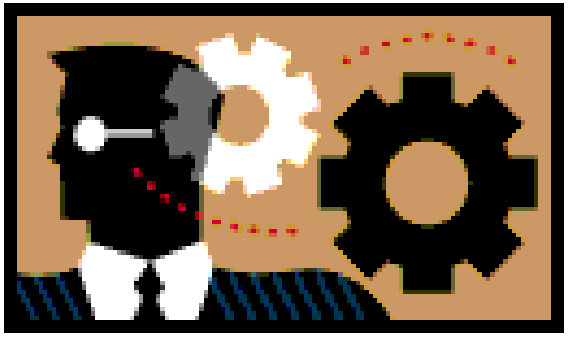
③ 不具合を
摘出する努力



FUJI Xerox

ソフトウェアテストの目的

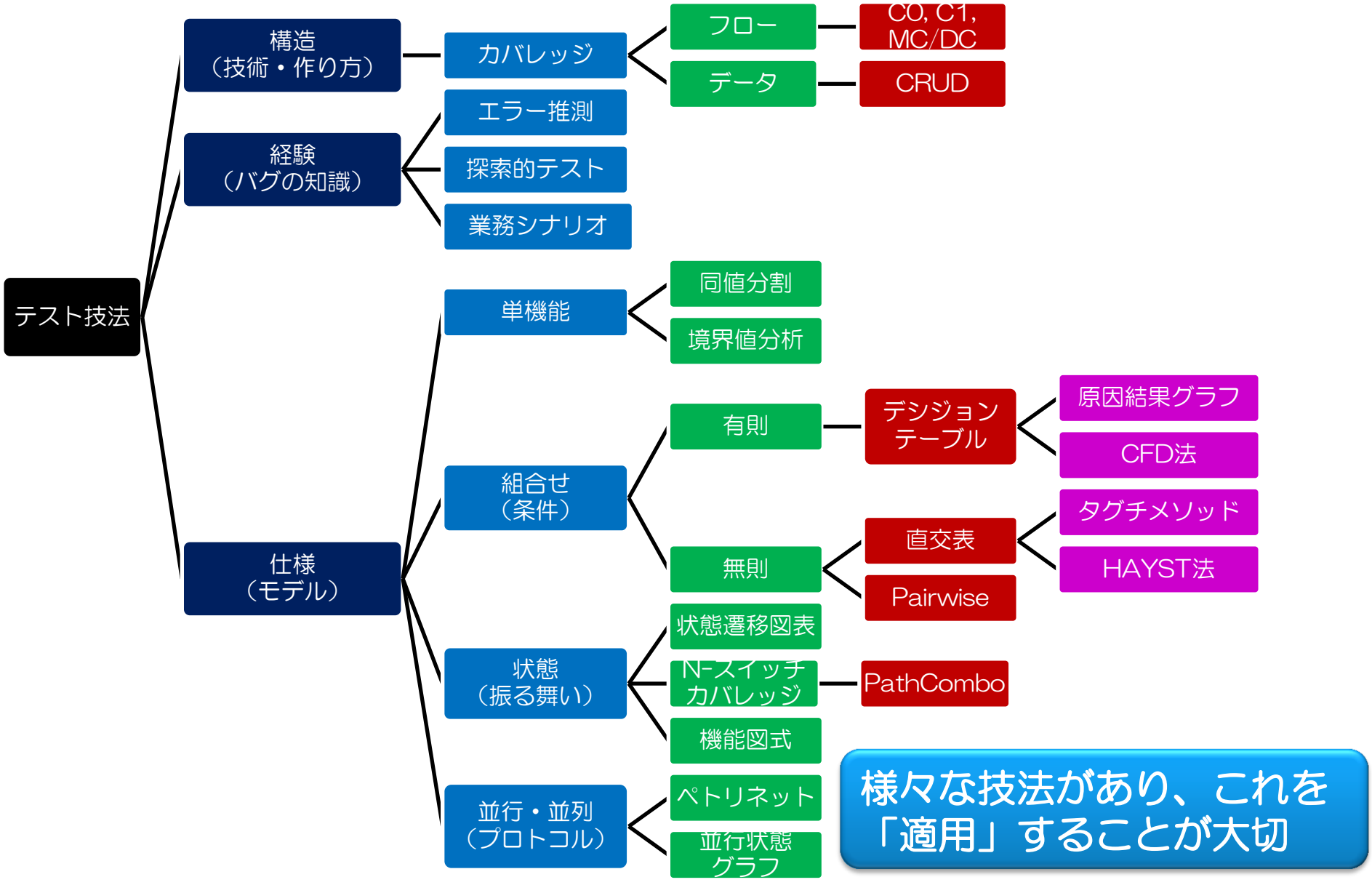
- 欠陥を摘出する
  - 定義（要求・仕様・設計）された通りに動くことを確認する
  - 未定義部分で何が起こるか分からないので網羅的に確認する（★）
- 対象ソフトウェアの品質レベルが十分であることを確認する
  - 品質を保証するためのエビデンス（証拠）を残す（★）
  - 実際のデータを用いて擬似的に動作確認を行いメトリクスを確認する
- 意思決定のための情報を示す
  - テスト計画時に、リスクがどれだけ減るか示す
  - テスト結果報告時に、残リスクを示す（★）



- 欠陥の作り込みを防ぐ
  - ドキュメントやソースコードのレビューを実施する
  - 本質的な「バグの知識」をきちんと考え、蓄え、開発に活用する（★）

★は特にテストで重要なこととして電気通信大学の西康晴先生が主張されました。

テスト技法の位置づけ（テスト技法の種類）



FUJI Xerox

テスト技法の時間的適用

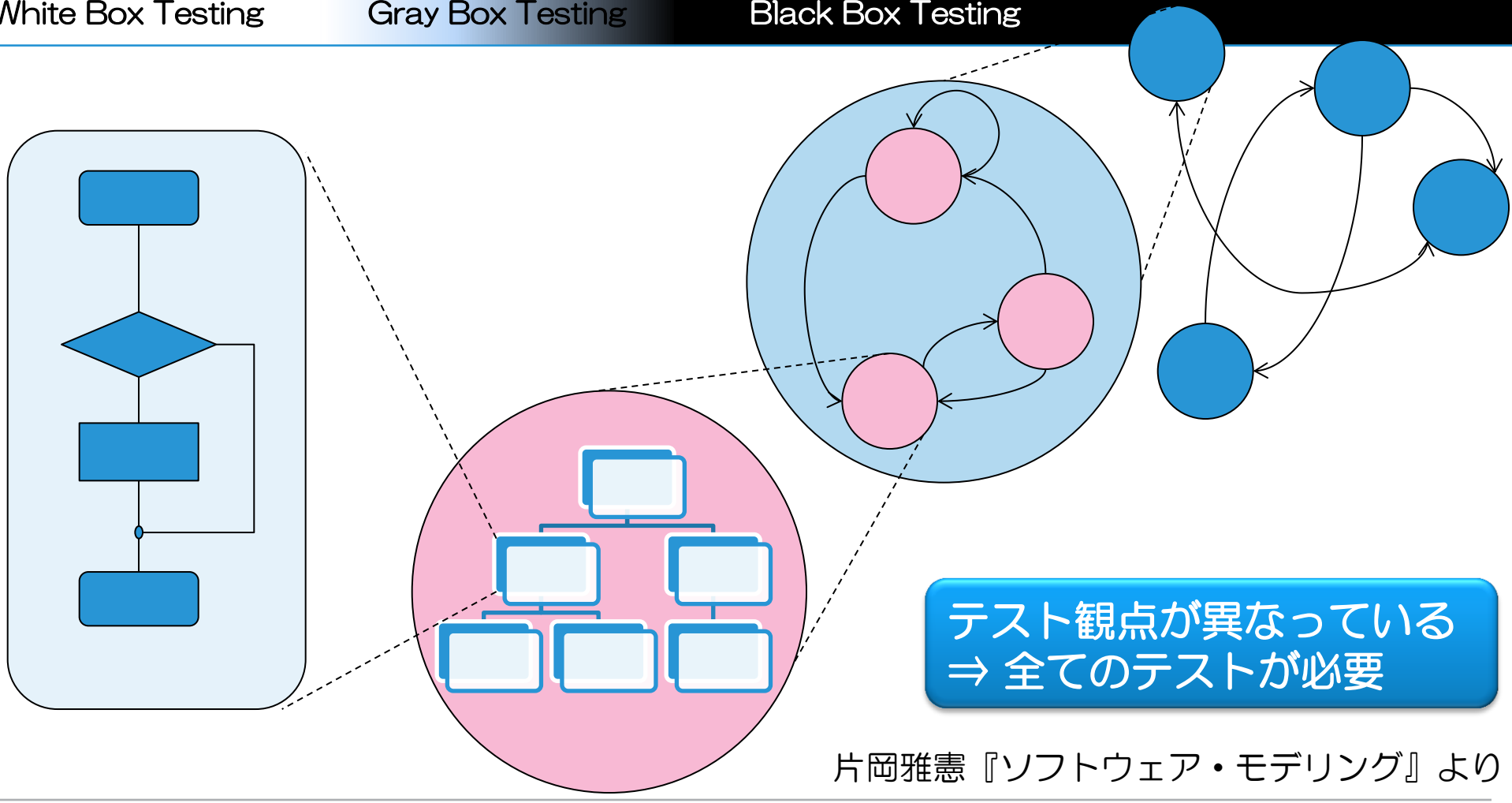
◆初期テスト（JSTQB「テストの7原則の3」）

- 早く欠陥を見つけるために、テストはソフトウェア開発もしくはシステム開発のライフサイクルのなるべく早い時期に開始し、あらかじめ定義した目的に集中すべきである。

✓関数ができたら構造のテストを行う
✓少しでも動くようになったら単機能テストを行う
✓いくつか動き始めたら組合せテストを徐々に広げていく
✓システムとして動くようになってきたら状態遷移テストをする
✓全部ができたら経験ベースのテストを行う
✓品質保証のため、各種計測を行う（非機能要件を含む）

出来上がるにつれて、構造→機能→振る舞い→並行順にテスト

構造（1：1）	機能（多：1）	振る舞い（1：多）	並行並列（多：多）
C0, C1, MC/DC, CRUD	単機能、DT, CEG/CFD, HAYST	状態遷移図表, N-switch, ユースケース, カバリングアレイ	ペトリネット, シナリオ, Model Checking
White Box Testing	Gray Box Testing	Black Box Testing	



テスト観点が異なっている
⇒ 全てのテストが必要

片岡雅憲『ソフトウェア・モデリング』より

FUJI Xerox

テストプロセス

- なぜ、従来のテストではだめなのか
- 智美塾やTest.SSFが提案するテストプロセス
- HAYST法のテストプロセス

従来のテストの問題点

◆三遊間が漏れる

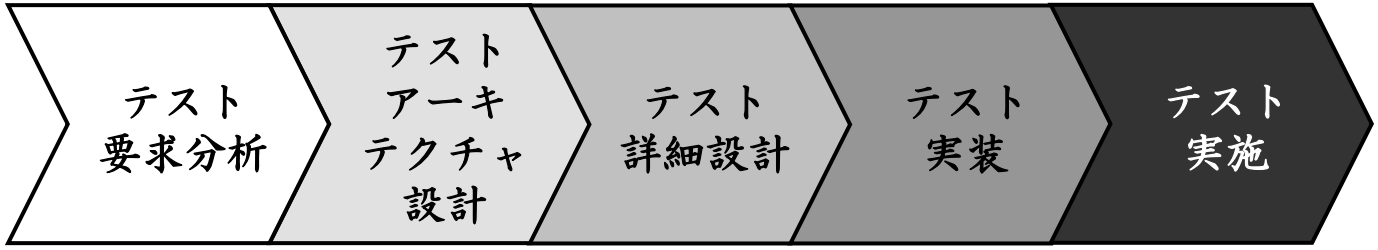
- 機能拡張では上手くいくが、新規機能追加では上手くいかない
  - ✓新技術が入った場合に何をテストしたらよいか分からない

- テストの全体が開発に依存するので分かり難い
  - ✓テストの重複が発生する
  - ✓テストの抜けが発生する
  - ✓要求に対しての確認が不十分で「使えない機能」が生まれる
  - ✓非機能のテストが不足しやすい

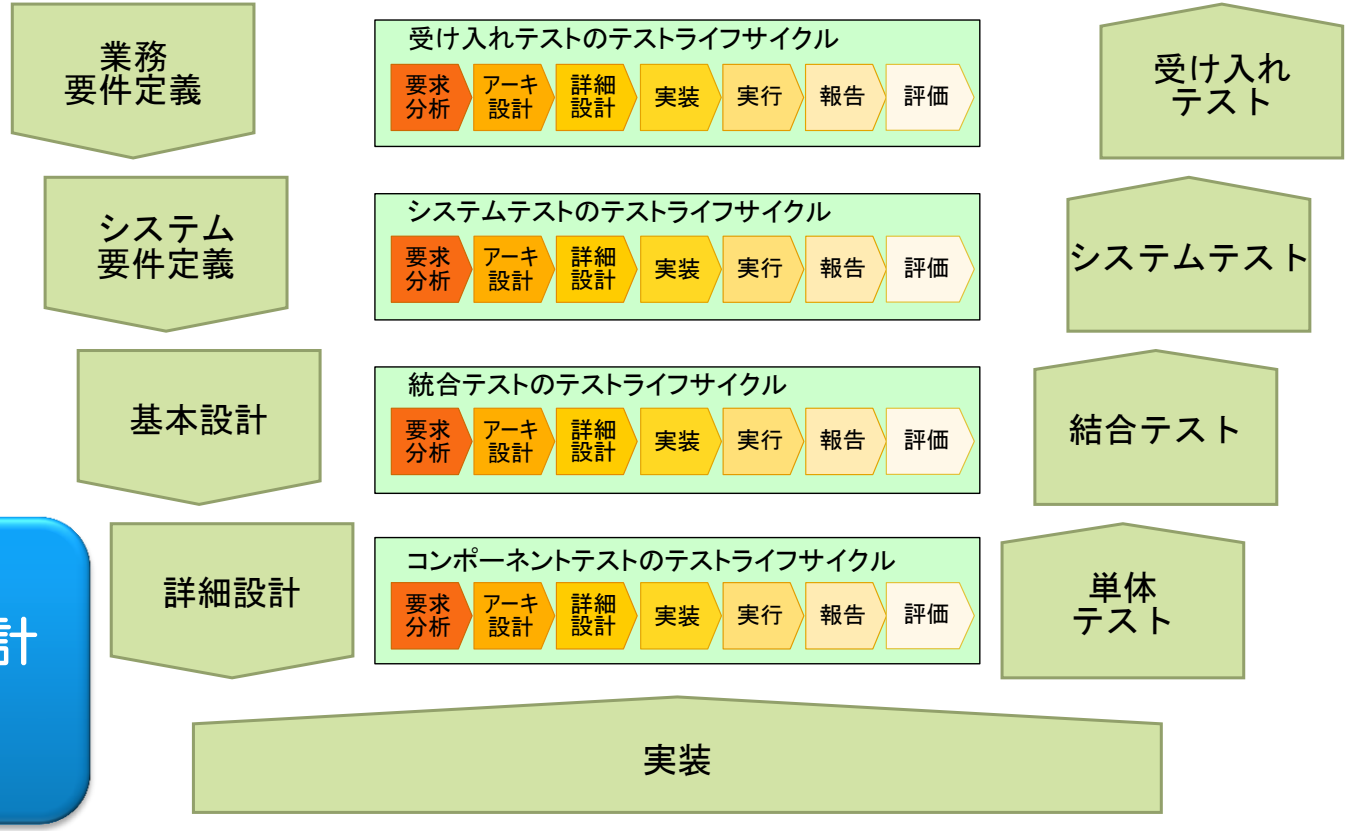
- ユーザーの観点が少ない
  - ✓市場導入後、基本的な問題が出る（やりたいことができない）
  - ✓市場の変化に対応できず、すぐに陳腐化してしまう

智美塾やTest.SSFのテストプロセス

智美塾のテストプロセス



Test.SSFのテストプロセス



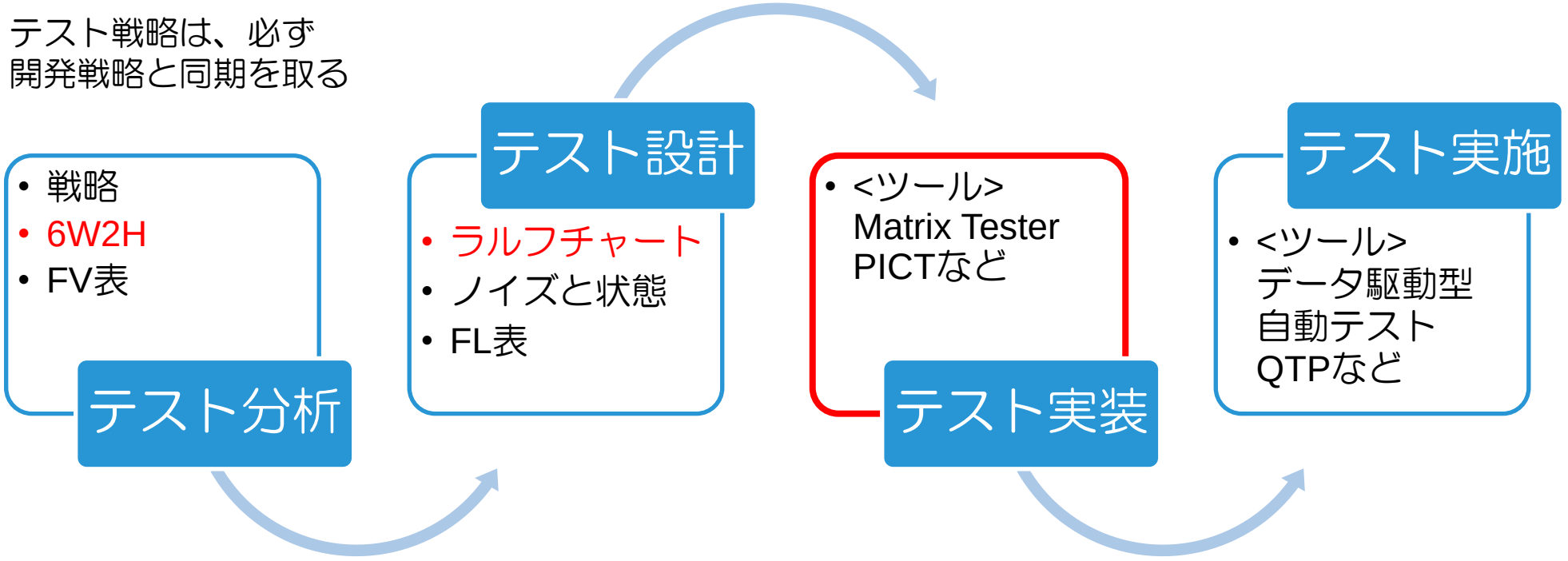
- テスト要求分析
- テストアーキテクチャ設計
- テスト詳細設計

が必要（具体的には？）

FUJI Xerox

HAYST法のテストプロセス（智美塾やTest.SSFの具体例の一つ）

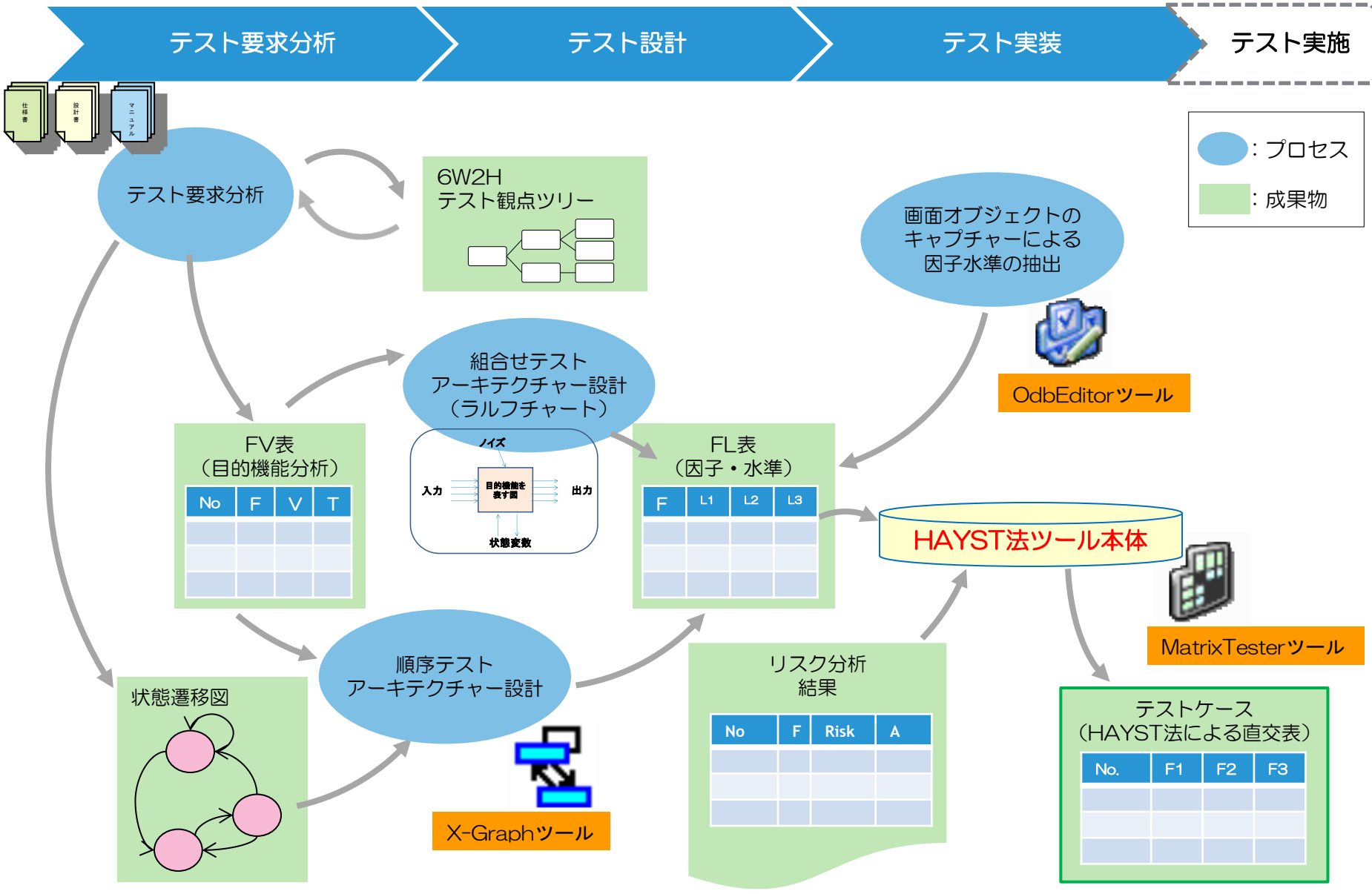
テスト戦略は、必ず
開発戦略と同期を取る



- HAYST法は初めは、直交表を用いたテスト実装（赤枠）部分を意味していた
- それでは三遊間問題が起こるのでテスト全体のプロセスを考えた
- 今では全体をHAYST法と呼んでいる
- HAYST法はシステムテストが対象。コンポーネントテスト、統合テストはABC（当たり前のことを、ぼんやりせずに、ちゃんとやる@菅野文友先生）で良い。つまりTDD/C1カバレッジ/境界値分析/デシジョンテーブル/自動化

FUJI Xerox

HAYST法のテストプロセス



FUJI Xerox

HAYST法のテストプロセス前半の詳細

テスト要求分析（TRA： Test Requirements Analysis）

＜ノイズ＞ • 要求の変化/追加要求 • 技術の変化への対応の要求		＜アクティブノイズ＞ • 機密漏洩
＜入力＞ • 要求の源泉 • 文書/ビデオ/対話	＜アクティビティ＞ • テスト対象/テスト目的/制約の分解と理解 • 関係を体系化：is-a, has-a, property • リスクの識別	＜出力＞ • テスト要求 • 6W2Hツリー • リスク分析結果
＜参照/ノウハウ＞ • 過去のテストタイプや成果物/テスト標準 • スキル/当該テスト対象の経験 • リソース（コスト/人数/スケジュール）		＜使用技法/ツール＞ • 6W2H

テストアーキテクチャ設計（TAD： Test Architecture Design）

＜ノイズ＞ • 寝不足/割り込み（電話他） • 情報を整理する紙が小さい		＜アクティブノイズ＞ • 機密漏洩 • 具体的成果物の報告プレッシャー
＜入力＞ • テスト要求	＜アクティビティ＞ • 機能を目的機能へ書換える： FV表作成 • 複雑な目的機能の分析： ラルフチャートの作成 • ラルフチャート間の関係性の整理/体系化	＜出力＞ • FV表 • ラルフチャート • RC関連図
＜参照/ノウハウ＞ • 過去のテストタイプや成果物/テスト標準 • スキル/当該テスト対象の経験 • リソース（コスト/人数/スケジュール）		＜使用技法/ツール＞ • FV表 • ラルフチャート • ラルフチャート関連図

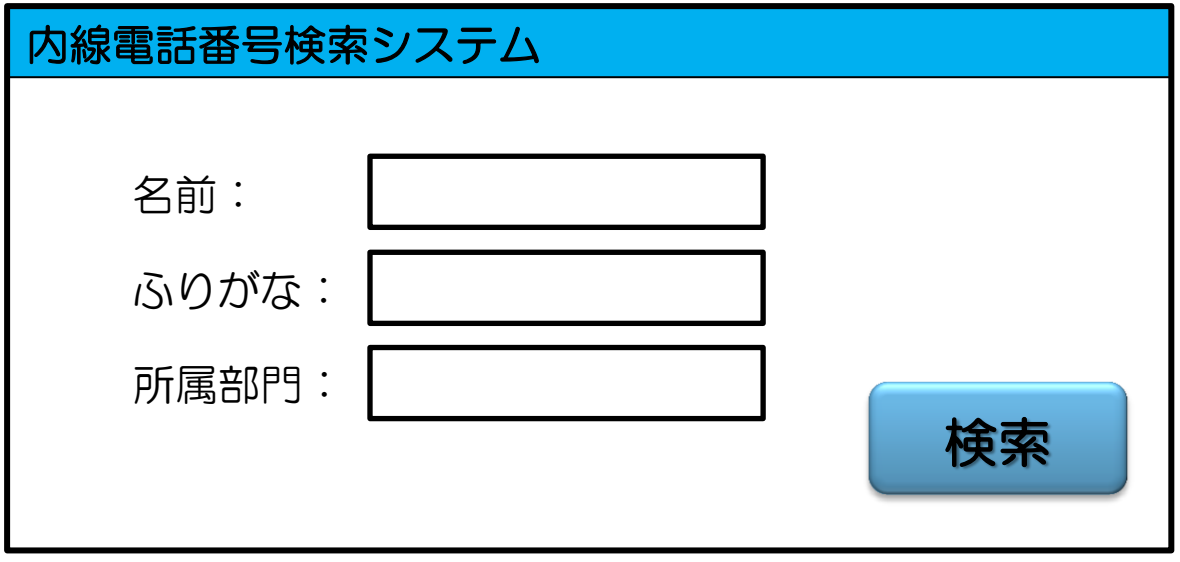
FUJI Xerox

演習

- 6W2Hで分析
- ユーザーストーリーで価値を理解
- FV表で整理
- ラルフチャートで因子を発見

従来のテスト方法で解いてみましょう（10分間）

- 下記の「内線電話番号検索システム」に対してこれまでの経験から、テストケース（自分ならどんなテストをするか？）をあげなさい。
- 確認したいこと（仕様等）があった場合は、メモしておくこと。



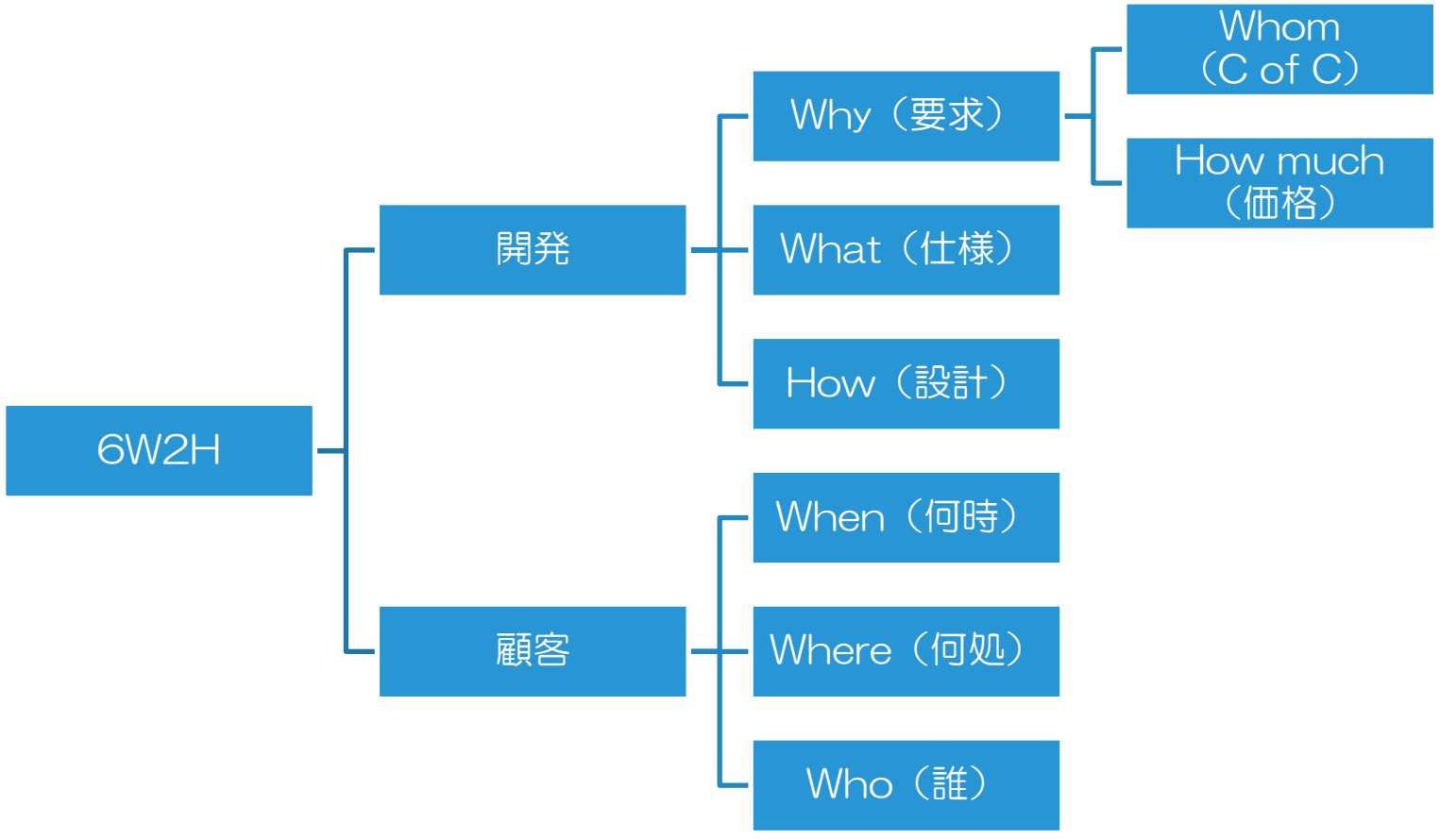
テスト要求分析： 6W2H

企画書

要求書

仕様書

設計書



FUJI xerox

6W2Hを書くコツ

- 書く順番
  - 仕様（What）の枝から書き始める。残りは自由に
  - 上から書く
- 書く言葉
  - 書くときに「その他」や「非A」という名前はNG
    - ✓これらは分類を放棄しているから
  - 書く言葉は基本的に単語
    - ✓仕様書のノードを書くときには、仕様の各文を単語に分解しながら「～」でつなぎ、単語がノードになる
    - ✓例） 名前 ～ 0文字以上 ～ 20文字以下 ～ 漢字 ～ 受け付ける
    - ✓最初はMECEは考えない（2つ上以上のノードは見ない）
  - 要素を増やす
    - ✓それぞれの単語に対して例示・間・対称・類推・外側・破壊を増やす

補足： 例示、間、対称、類推、外側、破壊について

- （階乗計算を例に）例示を行う
  - 階乗計算の例示（数値ｎを3, 5, 10, …と具体的な値に置き換える）
    - 3! ＝ 3×2×1 ＝ 6
    - 10! ＝ 10×9×8×7×6× 5×4×3×2×1 ＝ 3,628,800
- ピンポイントの5個の視点
  - 間： 間について考える
    - 3.5は入力できないだろうか？？
  - 対称： 引っ繰り返してみる
    - 3を引っ繰り返して「−3」や「1/3」が見つかる
  - 類推： 似たものを探してみる
    - １０から0を類推する
  - 外側： 集合の外側を考える
    - 数値の外側は？　→　文字を入力してみる
  - 破壊： 入出力の関係を乱したり負荷をかけるなどの意地悪条件を科す
    - 1000などの大きな数を入力してみる、表示桁あふれの確認

そ（外側）
れ（例示）
は（破壊）
あ（間）
た（対称）
る（類推）
と覚える。

6W2Hを書くコツ（続き）

- チェックの仕方
  - 全体をチェックする（MECE：漏れなくダブリなくを意識する）
    - ✓ぶら下がっているノードに漏れ・ダブリがないこと
    - ✓上下関係に矛盾がないこと
      - ツリーを下から追って、子ノードをまとめて親ノードになっているといえるか
      - ツリーを上から辿って、子ノードに違和感のあるものはないか
    - ✓離れた場所に同じ概念のノードは無いか（同じ名前で違う概念のものはよくある。これはOK）
  - HAYST法ではチェックは重視しない
    - ✓チェックよりもノードを出し尽くすことに注力する
    - ✓6W2Hはあくまでもテスト要求を理解するためのツール

内線電話番号検索システムの6W2H（10分間）

- 仕様を読みながら6W2Hでまとめてください。
- 分からないところは、これまでの経験で想像して書きましょう。

テストアーキテクチャ設計： FV表

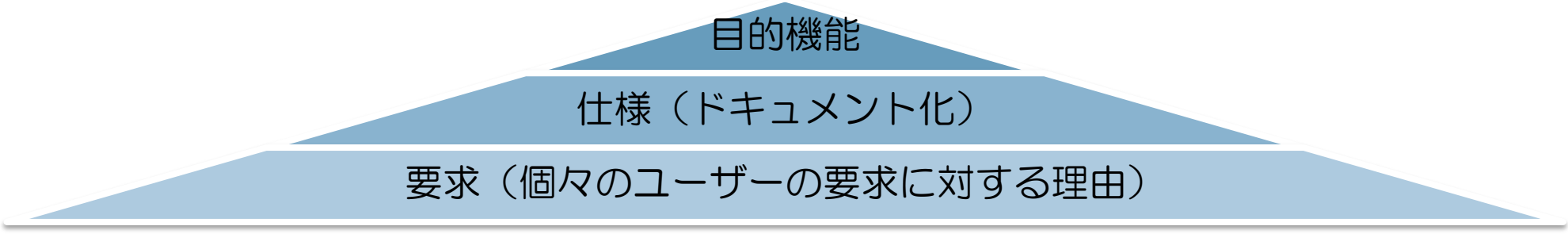
- テストの十分性を確保するためテストベースを整理する表
  - テスト対象物のリストアップ
    - テストできなくても良い、見落とさない
  - FV表の作成（目的機能でテストの十分性を確保する）
    - 目的機能（F）： お客様は何をしたいのか？（Why？）
    - 検証（V）： 何を確認したら機能したといえるのか？（What?）
    - テスト技法（T）： どのテスト技法でどこまで検証するのか？（How？）
  - FV表はGQM（ビクター・バシリ先生提唱）に対応する
    - 目的機能（F）： Goal層（概念レベル）：目的の定義
    - 検証（V）： Question層（運用レベル）：評価すべき質問
    - テスト技法（T）： Metric層（定量レベル）：測定可能なメトリクス

目的機能で整理する理由

- 仕様書の「機能」で整理することの問題点
  - 「正しく動作」しているかの確認しか出来ない
    - 仕様書には「機能」の動きしか書いていない
      - コンテキスト（使用の文脈）の理解が重要
      - 仕様書には暗黙の仕様（前バージョンの仕様、開発者の常識）は書かれない
    - 本来は「正しい動作」をしているかの確認がテストの目的
  - 要素還元的な見方しか出来ない
    - 分解した小さな機能が全て動けば全体が問題なく動くと思ってしまう
    - 組合せの視点が欠ける
- 「機能」から導き出した「目的機能」で整理する意義
  - 必ずユーザの使用目的や市場条件を考えることになる（動的）
  - 非機能要件のテストが楽になる
    - 目的機能単位に非機能要件のメトリクスを計測する
    - 性能、信頼性、セキュリティ要件は目的ごとに異なる（カプセル化）

目的機能がなぜ必要か

用語	定義	対応技術	対応英語
テスト	合格/不合格を決める行動	モデルベースドテスト： ・モデルを明確にする ・網羅基準	Testing
検証	仕様通りか？	CFD法： ・機能が動作すること ・エラー処理がされること	Verification
妥当性確認	ユーザー要求通りか？	USDM： ・要求と仕様の対応 ・理由の記述	Validation
評価	将来にわたって多くのユーザーの期待に応えられるか？	FV表： ・目的機能の記述 ・予測する	Estimation (not evaluation)



目的機能　≒　ユーザーストーリー

ユーザーストーリーとは

- フィーチャーや機能の価値をユーザーに伝え、会話を引き出すためにKent Beckによって、2000年にその概念が作られた
- ユーザーストーリーのテンプレート（Mike Cohn：2004年）
  - ＜ユーザーの役割＞として、＜ゴール＞を達成したい。[それは、＜理由＞のためだ。]
    - ユーザーの役割： ゴールの達成により恩恵を受けるユーザーの役割
    - ゴール： ユーザーストーリーのゴールであるフィーチャーや機能
    - 理由： ゴールを達成した時にユーザーが得られる利益や利点
- ユーザーストーリーの評価軸（INVEST）
  - 独立していること（Independent）、交渉可能であること（Negotiable）、価値があること（Valuable）、見積もり可能であること（Estimatable）、適切なサイズであること（Sized appropriately）、テスト可能であること（Testable）
- ユーザーに直接接点を持つ目的機能はユーザーストーリーで表現できる

内線電話番号検索システムのユーザーストーリー（10分間）

◆内線電話番号検索システムのユーザーストーリーを書きましょう。

内線電話番号検索システムのFV表（10分間）

◆内線電話番号検索システムのFV表を2行書きましょう。

No.	Fr（目的機能、ユーザストーリー）	V（検証、どんな評価をするか）	T（テスト技術、どの技法を使うか）
1			
2			

ラルフチャートとは

ラルフチャートとは

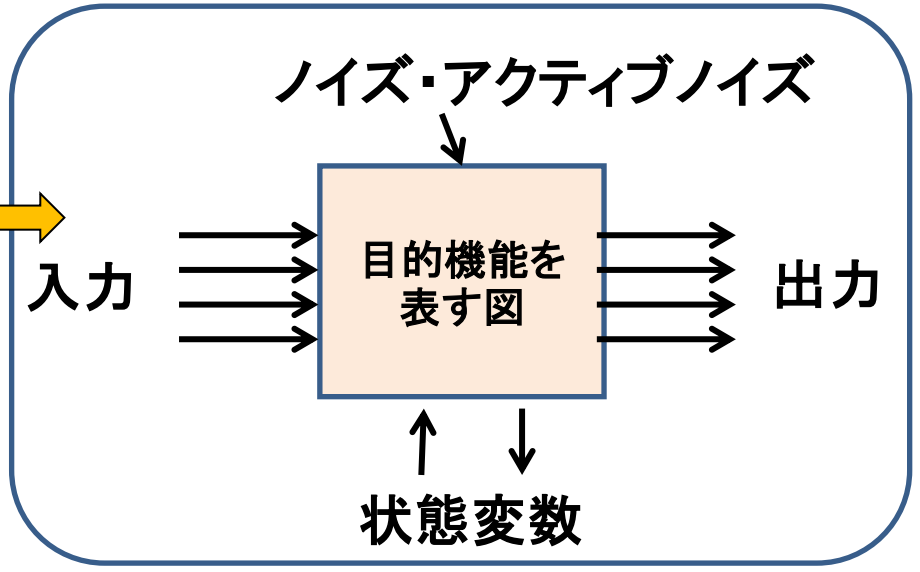
- FV表で抽出された目的機能の一行に対して、
- それが複雑ですぐにテスト詳細設計に移れない場合に、
- 目的機能を図示して因子・水準を抽出し、
- さらに、テスト技法を発見するチャート

＜FV表＞

No.	目的機能	検証	テスト技法

・仕様書の項番で【テスト対象】の網羅性を確保する
・目的機能とは機能に果たすべき目的を書き加えたもの

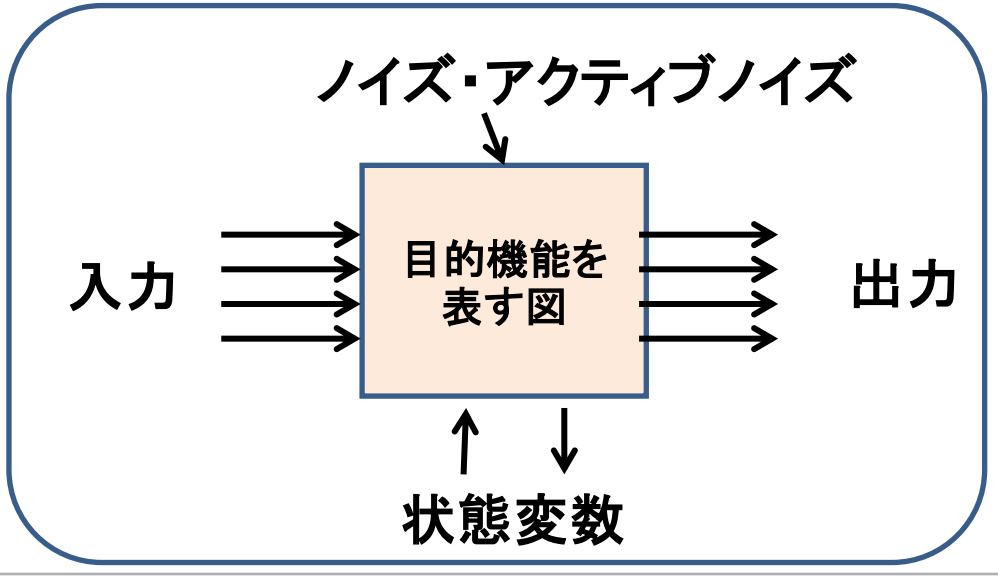
＜ラルフチャート＞



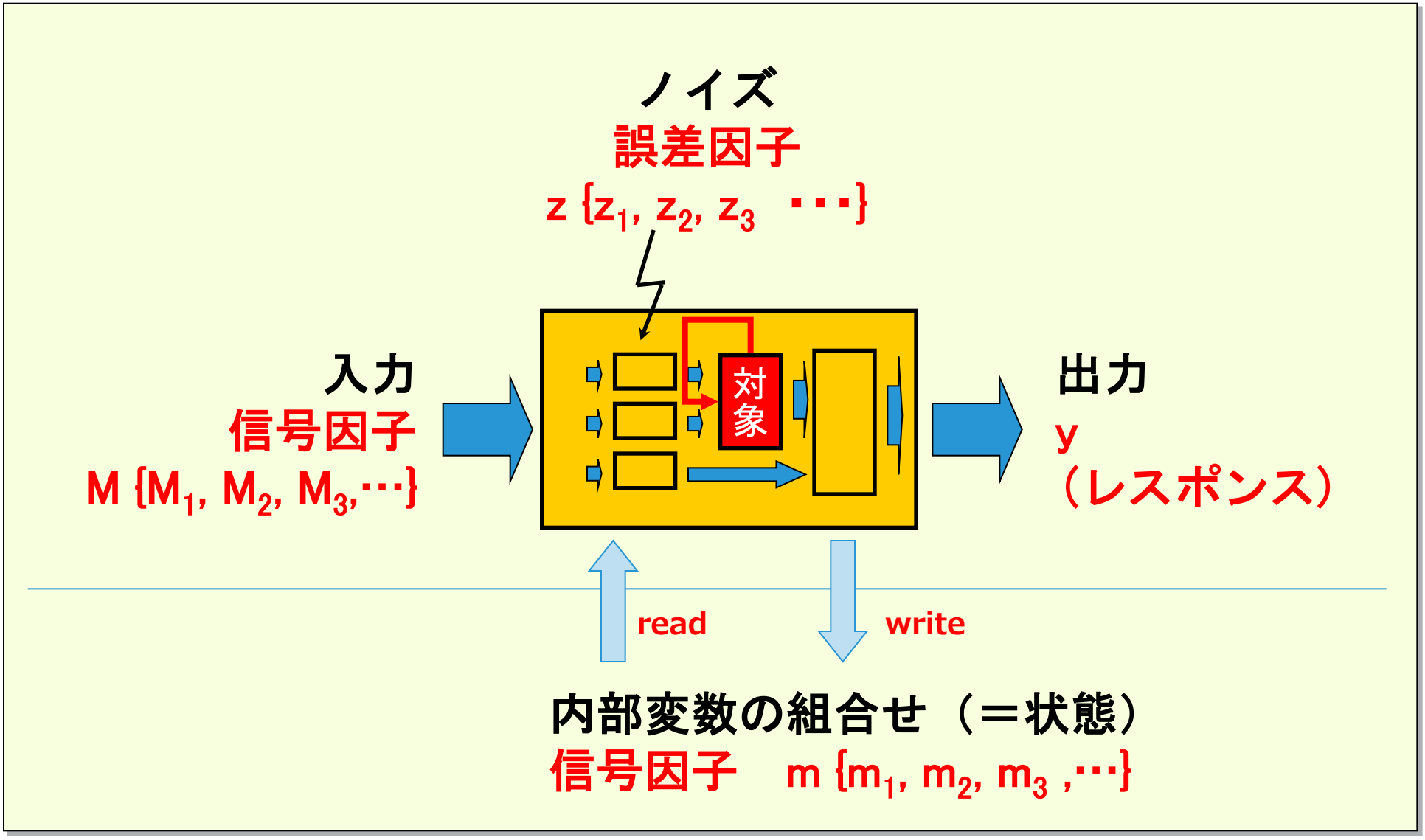
ラルフチャートのフォーマット

ラルフチャートのフォーマットと抽出する因子

- 入力（使用者の意思）
- 出力（目的＝得たい結果）
- 状態変数（システムの状態）
  - プログラムが動作中に参照または書き換える変数
- ノイズ（市場環境の状態、システムの外部であり制御不可能な要因）
  - ノイズ（入出力の関係を妨げる要因）
  - アクティブノイズ（人がわざと行う要因。たとえば犯罪やいたずら要因）

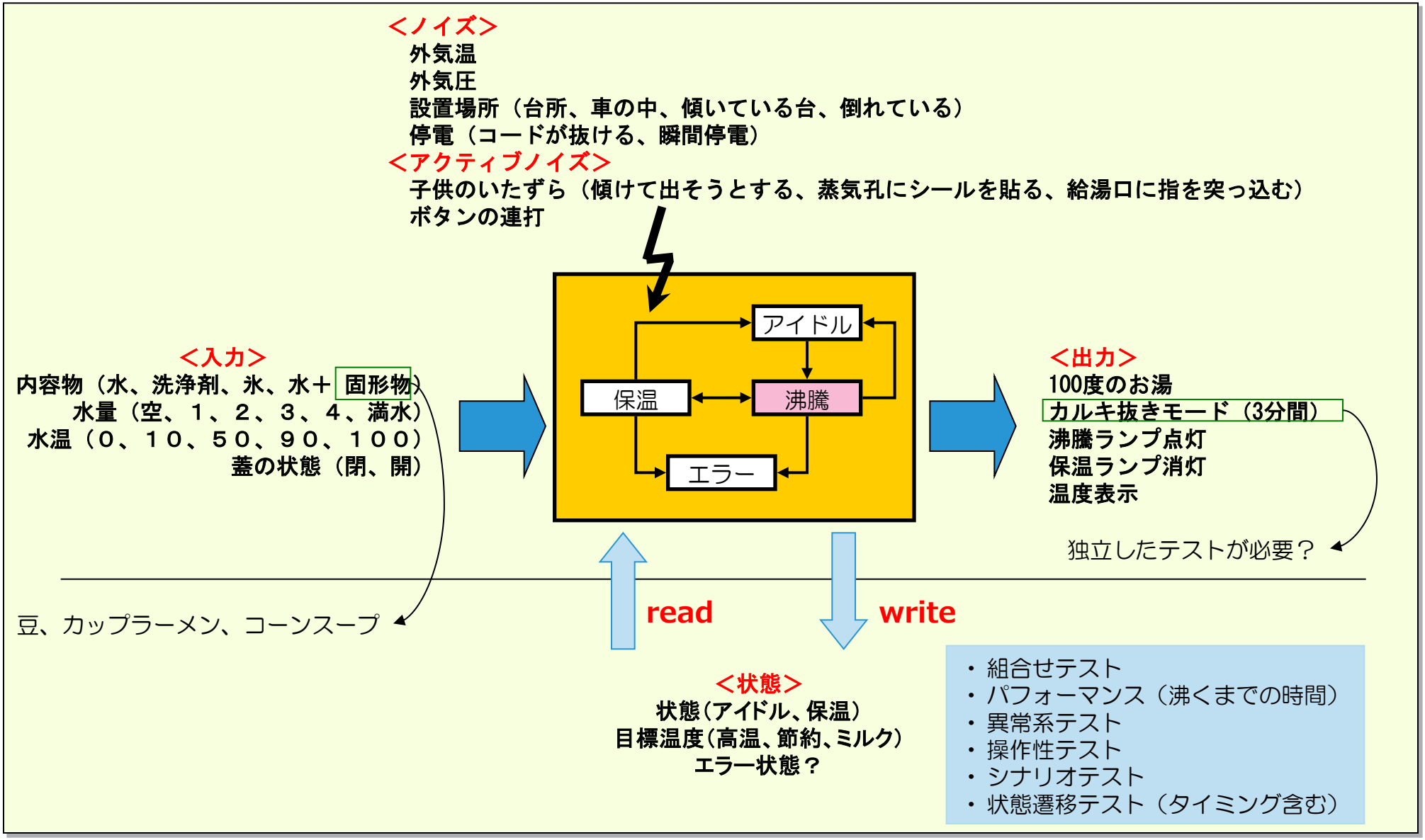


ラルフチャート



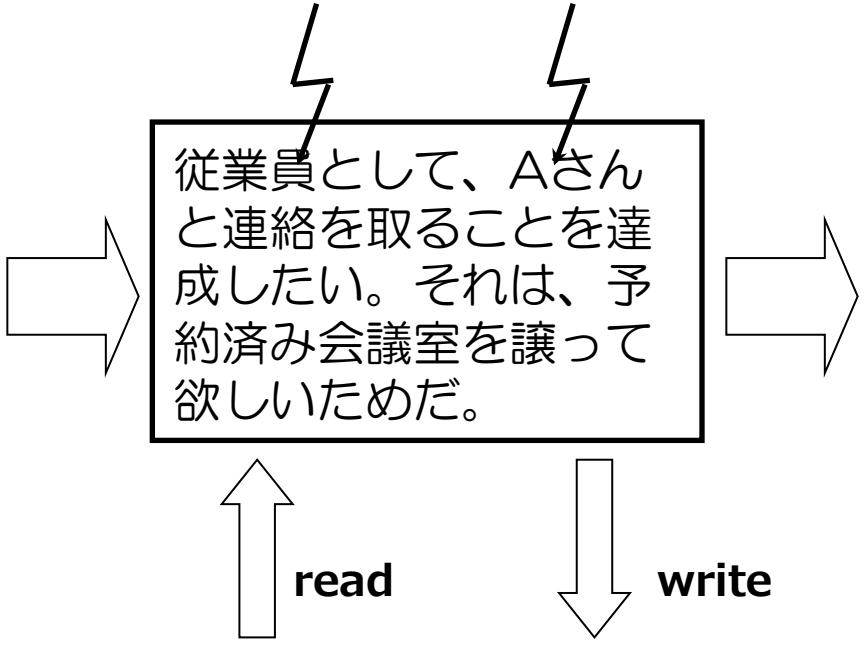
FUJI xerox

ラルフチャートの具体例： 電気ポットのラルフチャート



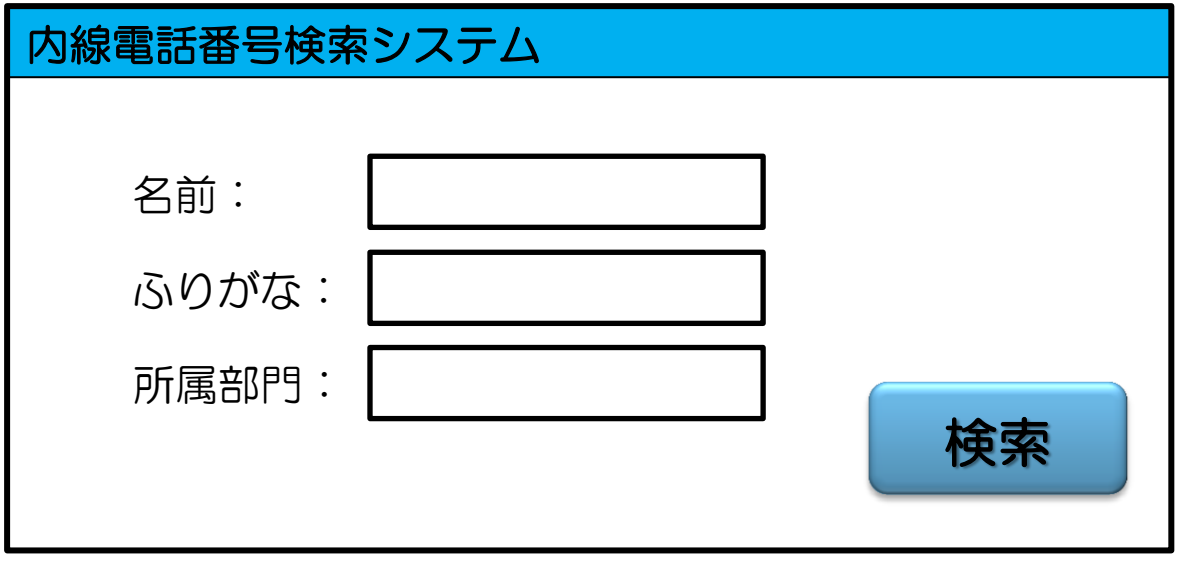
FUJI Xerox

内線電話番号検索システムのラルフチャート（10分間）



従来のテスト方法と比較

- スライド17で、最初にテストケースを考えました。その時に書いたテストケースと、テスト要求分析・テストアーキテクチャ設計を行った後にテストしようとしている内容を比較しなさい。



まとめ

- 本講座のまとめ

まとめ

- テスト分析とは
  - テスト対象を理解する
    - 製品（仕様）をしっかり理解する
  - テスト目的（制約含む）を理解する
    - ユーザーをしっかり理解する
- テストアーキテクチャ設計とは
  - テストを構築する
  - 目的機能（ユーザーストーリー）で整理する
    - テストで確認すべき価値が明らかになる
    - テストがきれいに分割できる
  - ラルフチャートで複雑な目的機能の因子を発見する
    - 結果に影響を与える要因のなかでテストすべきものを見つける


FUJI xerox