

2020/09/12



# JaSST'20 Kansai 基調講演

## テストの視点でシステムを分析する

### ～アジャイル開発を例に～

---

ソフトウェア品質のトータルサポート企業

**バルテス株式会社**  
江添 智之

# 会社概要



## バルテス株式会社

Value created through Testing

社名の由来は、「テストで、価値を創造」

設立： 2004年4月19日

資本金： 9,000万円

代表取締役社長： 田中 真史

- 事業内容：
1. ソフトウェアテストサービス
  2. 品質コンサルティングサービス
  3. ソフトウェア品質セミナーサービス
  4. セキュリティ・脆弱性診断サービス
  5. その他品質評価、品質向上支援サービス

拠点： 東京、大阪、名古屋、福岡

証券コード： 4442 東京証券取引所マザーズ

その他： 書籍出版

ISO/IEC 27001取得

### グループ会社



バルテス・モバイルテクノロジー株式会社

大阪・東京

アプリ開発・セキュリティ診断



VALTES Advanced Technology, Inc.

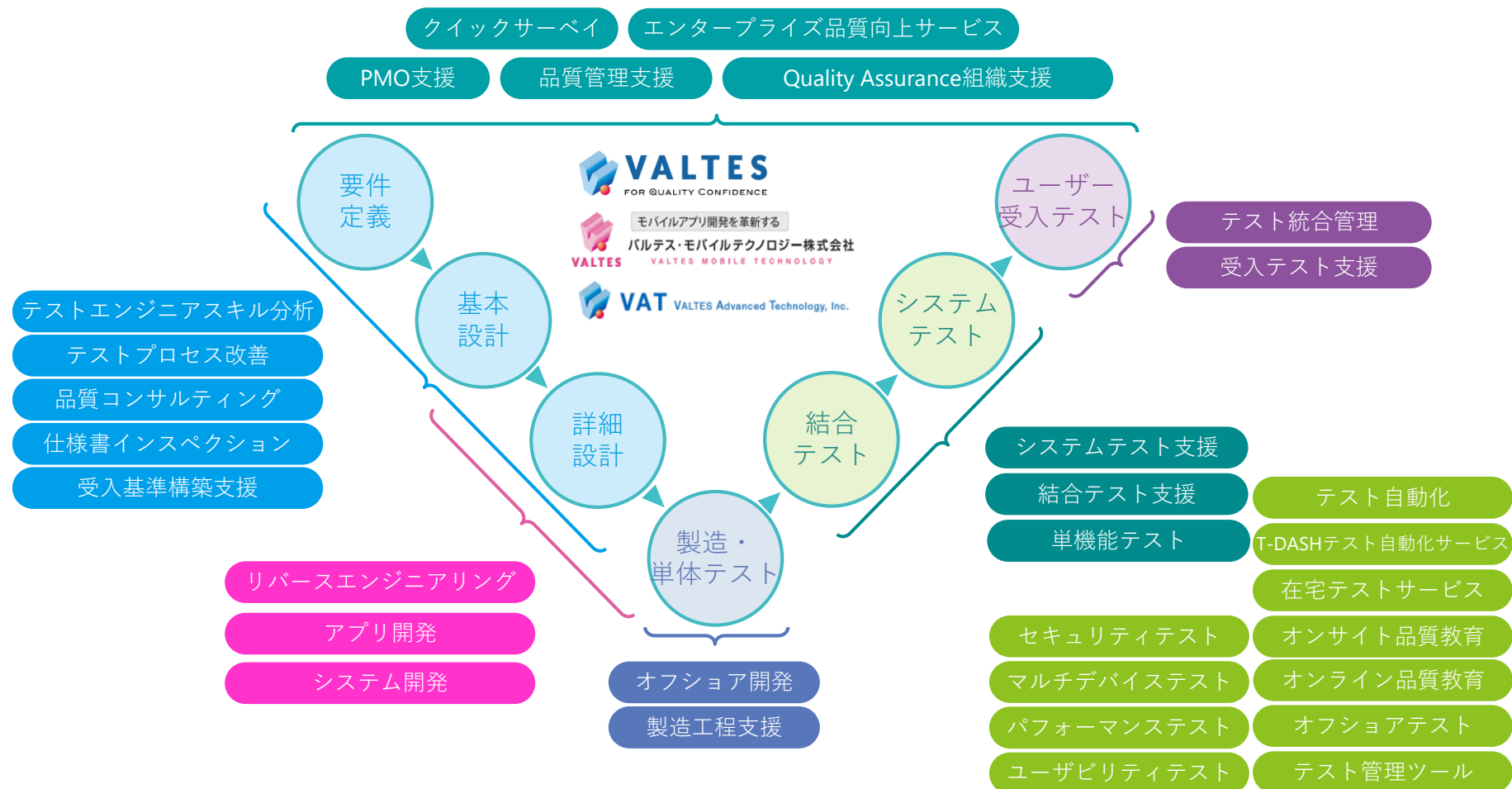
フィリピン・マカティ

オフショア（テスト・開発）



# バルテストグループのサービスマップ

ソフトウェア開発の全工程で、ソフトウェア品質の向上支援サービスを提供



# 講師紹介



## 江添 智之（えぞえ ともゆき）

バルテス株式会社 第3ソフトウェアテスト事業部  
R&C部 マネージャー

### 【主要業務】

- ・ソフトウェアテスト技術の研究開発
- ・社内・社外へのソフトウェアテストの教育業務
- ・品質コンサルティング業務
- ・テストエンジニア・プロジェクトマネージャ

### 【所有資格など】

- ・ JSTQB Advanced Level テストマネージャ
- ・ JSTQB Advanced Level テストアナリスト
- ・ Scrum Alliance 認定スクラムマスター
- ・ JDLA ディープラーニング G検定
- ・ JaSST（ソフトウェアテストシンポジウム）関西  
実行委員長



ISO/IEC/IEEE 29119  
ソフトウェアテスト規格の教科書  
(翻訳)

# 目次

**第1章** テストの現状とテスト対象の分析・再構築

**第2章** アジャイル開発における品質の考え方

**第3章** テスト観点をベースにした  
「受け入れ条件」の考え方

まとめ

# 第1章

## テストの現状と テスト対象の分解・再構築

# 1. テストの目的

開発者の視点



正しく製品を作っているか

製品が開発仕様書を  
実現できているかを確認する  
(検証: Verification)

ユーザーの視点



正しい製品を作っているか

製品がユーザーの要求を  
満たしているかを確認する  
(妥当性評価: Validation)



## 2. 現在のソフトウェア開発

正しく製品を作っているか

製品が開発仕様書を実現できているかを確認する  
(検証: Verification)

正しい製品を作っているか

製品がユーザーの要求を満たしているかを確認する  
(妥当性評価: Validation)

仕様がはっきりしない

要求が「テスト可能」な状態  
になっていない

という状況で開発・テストを進めざるをえない



## 2. 現在のソフトウェア開発

### ▶ 「サービス型」志向の強化

システムの提供よりも複数のシステムを組み合わせた  
「サービス」としての提供

IoT、ものづくりからことづくりへ

### ▶ 仮説検証型開発の必要性

システム提供形態の多様性、市場提供のスピード化



従来の（仕様ベースの）テストだけでは不足

### 3. テスト対象の分解と再構築

#### ▶ 仕様化以前の状態で品質を考える

テストに必要な要素を分解し、分析する  
「何ができればOKか」を明確にする

#### ▶ 「何を」「どのように」テストするかを構築

テストのアプローチを可能にする

## 4. バルテストの取り組み

### ▶ テスト会社としての取り組み

- 多くの案件ではテスト設計～テスト実施がメイン
- （WFにおける）上流工程（要求・要件定義）から品質について定義することも大切
- テスト・ファースト、シフト・レフト・テストティング、W字モデル

### ▶ PMO、QMO、コンサルティング案件も増加

- アジャイル開発の品質支援も実施

## 第2章

# アジャイル開発における品質の考え方

# 1. スクラムにおける要求の整理

## ▶ プロダクトバックログ

- 要求をプロダクトバックログの項目として管理
- ユーザーストーリーの形式で作成
  - 「だれが」その要求を出しているかを明確にする
- 1スプリント中に「完成」できる量をプロダクトバックログから実装

# 1. スクラムにおける要求の整理

## ▶ プロダクトバックログの例

No	プロダクトバックログアイテム	SP
1	ユーザーとして、書籍を登録したい。 なぜなら、読んだり、読みたい書籍を記録しておきたいからだ。	5
2	ユーザーとして、サービスに登録したい。 なぜなら、このサービスを使いたいからだ。	5
3	ユーザーとして、サービスにログインしたい。 なぜなら、自分の情報をセキュアな環境で扱いたいからだ。	3
	...	
19	提供者として、アカウント凍結がしたい。 なぜなら、不快なユーザーが出るからだ。	中
20	提供者として、広告が欲しい。 なぜなら、収入源だからだ。	大
	...	

# 1. スクラムにおける要求の整理

## ▶ ユーザーストーリーの原則 (INVEST)

- Independent : 独立している
- Negotiable : 交渉可能
- Valuable : 価値がある
- Estimable : 見積り可能
- Size to fit(Small) : 適切なサイズ
- Testable : テスト可能



## 2. スクラムにおける品質の考え方

- ▶ 「完成」の定義 (Definition of DONE)
- ▶ 受け入れ条件 (Acceptance Criteria)

## 2. スクラムにおける品質の考え方

### ▶ 「完成」の定義（Definition of DONE）

- スプリント内での「完成」が何を示しているかを定義
- スプリント内で完了できず、リリースまでに行うべきことを「UNDONE」として定義
  - ✓ 非機能要件（性能、セキュリティ、ユーザビリティ）
  - ✓ 本番環境へのデプロイ
  - ✓ システムテスト、ユーザー受入れテスト
- UNDONE項目はプロダクトバックログ項目として消化する or リリース直前のスプリントで実施する
- 「完成」の定義は拡張をめざす

## 2. スクラムにおける品質の考え方

### ▶ 「完成」の定義の例

#### プロダクトバックログアイテム

- すべてのコードがコミットされている
- すべてのテストが通る
- 受入れテストがすべて設計済みで、実施済みで、通る
- ヘルプファイルが自動生成されている
- 機能テストが通る

#### スプリント

- プロダクトバックログアイテムの条件に加えて、
- プロダクトバックアップが更新されている
  - パフォーマンステストが実施済み
  - クラス図が更新されている
  - バグはすべて対応済みか延期済みである
  - ユニットテストのコードカバレッジが80%以上である

#### 内部リリース

- スプリントの条件に加えて、
- インストーラがビルドされている
  - 運用ガイドが更新されている
  - トラブルシューティングマニュアルが更新されている
  - ディザスタリカバリープランが更新されている
  - すべてのシステムテストスイートが通る

#### 本番リリース

- 内部リリースの条件に加えて、
- 負荷テストが実施済み
  - パフォーマンスチューニングが実施済み
  - ネットワーク構造図が更新されている
  - セキュリティ確認を受けている
  - 脅威モデル分析の確認を受けている
  - ディザスタリカバリープランをテスト済み

## 2. スクラムにおける品質の考え方

### ▶ 受け入れ条件（Acceptance Criteria）

- 個々のプロダクトバックログ項目の「完成」を定義
- 受入れ条件は「何ができればOKか」を指し示す
- 項目作成の段階で受入れ条件も定義する

ユーザーとして、  
私はファイルをアップロードしたい。  
なぜなら同僚とファイルを共有したいからだ。

- ・ 文書ファイル（.txt、.doc、.docx）が登録できること
- ・ 画像ファイル（.jpeg、.gif、.png）が登録できること
- ・ 1GB以下のサイズのファイルが登録できること
- ・ 実行ファイルが登録できないこと

### 3. アジャイル開発現場での実際

#### ▶ プロダクトバックログの粒度設定が難しい

- UNDONE項目など単純なタスクが入ることもある
- ユーザーストーリーはビジネス価値の単位となる
  - ⇒ 細分化が難しい
  - ⇒ タスク単位での細分化になりがち
    - ✓ 「誰のために」「何をなすべきか」が曖昧になる
    - ✓ 品質の基準があいまいになる

### 3. アジャイル開発現場での実際

1. 販売価格で検索がしたい

大きすぎ

2. 地図のプロット色の変更

タスク

3. DBにテストデータを100件インポート

タスク

### 3. アジャイル開発現場での実際

#### ▶ エピック、ユーザーストーリー、受入れ条件の粒度、定義を明確にする

- エピック⇒スプリントに入りきらないユーザーストーリー
- ビジネス価値を追求すると、えてしてエピック単位になりがち
- 受入れ条件を考えるとユーザーストーリーに分割できることが多い



### 3. アジャイル開発現場での実際

#### 1. 販売価格で検索がしたい

- UI開発
- DB連携

エピックを分割  
(受入れ条件で判別)

#### 2. 地図検索がしやすい

- UI改善
  - 地図のプロット色の変更

タスクをユーザー  
ストーリーへ

#### 3. テスト準備

- DBにテストデータを100件インポート

品質保証活動に必要な項目

## 第3章

# テスト観点をベースにした 「受け入れ条件」の考え方

# 1. 受け入れ条件をどう考えるか

## ▶ ウォーターフォールでも大事な視点

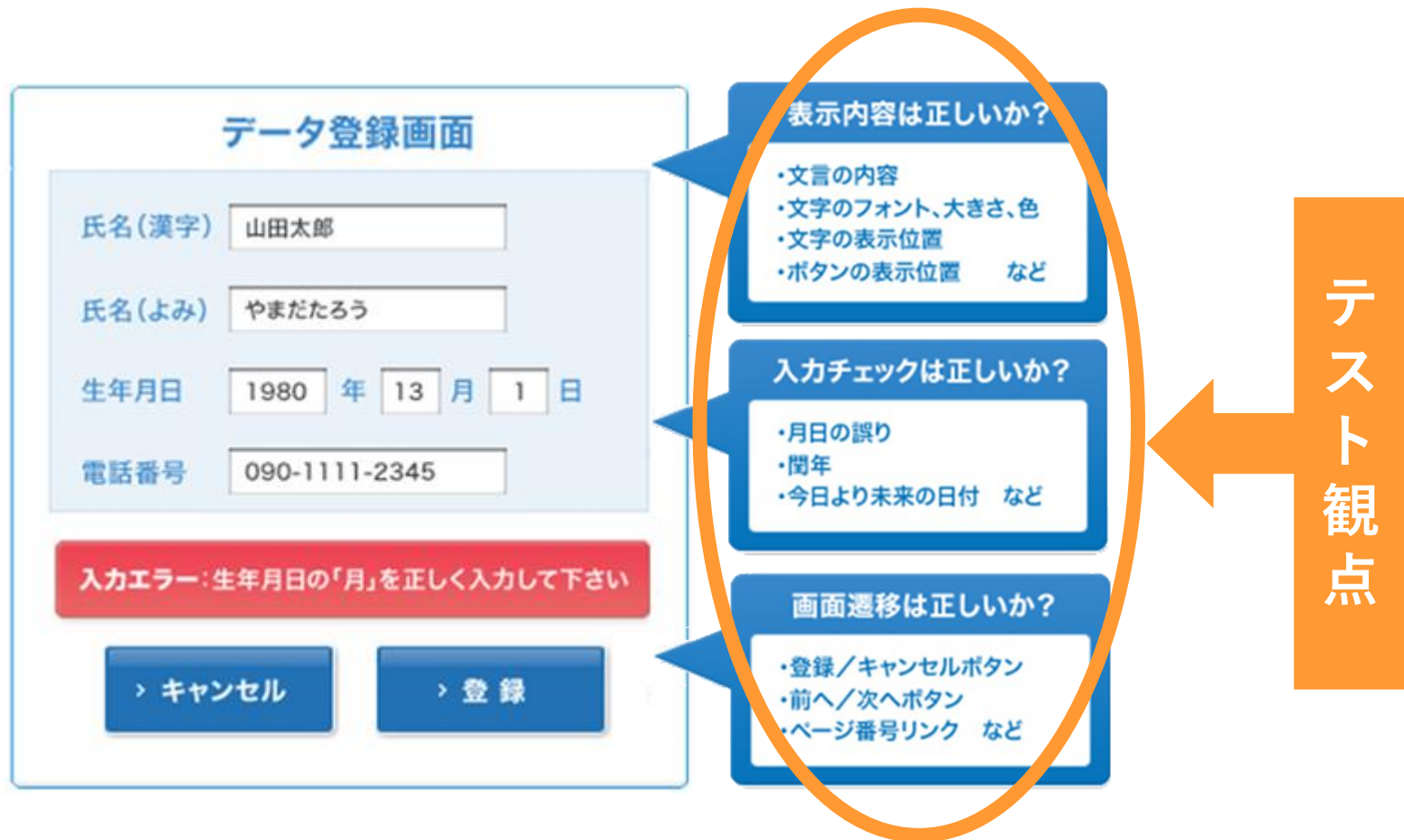
- 「何ができればOKか」を上流工程から考える
- アジャイル開発では要求（プロダクトバックログ）の段階で細分化・詳細化

## ▶ バルテスではテスト観点をベースに発展させる

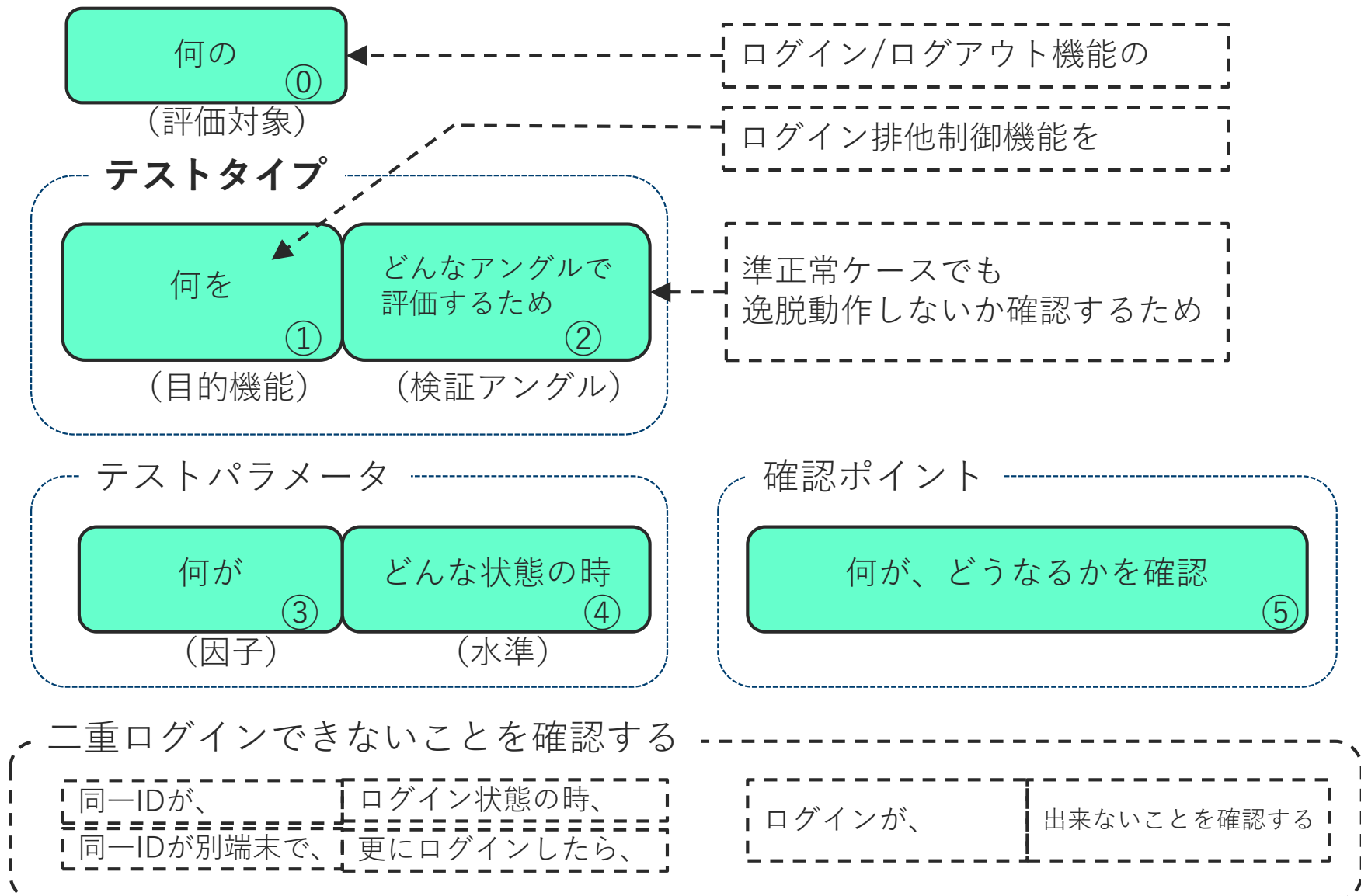
- V&Vやテスト条件の要素を押さえながら段階に応じて導き出していく

## 2. テスト観点とは

### ▶ テスト観点のイメージ



### 3. テスト観点の考え方（一例）



# 3. テスト観点の考え方（一例）

検証アングル	機能適合性	性能効率性	互換性	使用性	信頼性	セキュリティ	ポイント・例
通常バリエーションテスト	○						●テストパラメータを通常バリエーションで振り、評価ポイントが期待結果と一致していないところを探索 ・無効操作: 無効なキー操作などが本当に無効か(余分な操作をしないか)も確認
正常限界値テスト	○						・テストパラメータの上限、上限-1、下限、下限+1
準正常系テスト	○				○		●通常でも起こりえるイリーガルな状態でも逸脱動作しないか確認する (例)・途中中止: 操作途中で処理中断する、処理中に対抗デバイスを電源OFFするなど ・連続操作: 急いで操作するなど ・繰り返し操作: 他デバイスと「接続→切断→接続」など
異常系テスト	○				○		●異常値などの想定外のテストパラメータを設定しても、機能不良に陥らないことを確認する (例)・仕様スペックの上限+1、下限-1 ・対抗機から数値データでなく文字列データが送信されたなど、想定外のデータ
機能複合・競合テスト	○						●複数機能を組み合わせた機能試験 (例)・印刷用紙設定やページ割付などの組み合わせ ・「「言語変更」後に戻る」、「動画撮影中に静止画撮影」、「ハンズフリー中に音声認識」 ・「音楽再生中に音声着信」、「音楽再生中にメッセージ通知、更に音声着信」
構成テスト			○				(例)・「接続対抗機の機種別」、「接続対抗機のOSバージョン別」、「SDカードのメーカー別」
ローカリゼーションテスト	○		○				・仕向先別のテスト(例、日本向け、アメリカ向け、欧州向け、中国向け等々)
ストレステスト		○			○		●評価対象にストレスをかけても、機能不良に陥らないことを確認する (例)・「SPEC上限以上の電話帳をimport」、「キー連打」
エージングテスト					○		(例)・「単純操作の繰り返し」、「ある画面のまま放置」
性能テスト		○					●性能テストの項目を明らかにすること(例: 電話帳importの処理時間、メモリ使用量など) ●性能テストの目標値を明らかにすること。機能仕様書などで定義されていないものは、内部で目標値を定めること。特に指定されていない限り、既存機能は過去機種と同等とすること。
セキュリティテスト						○	●セキュリティ要件のある機能は、その検証を行う
ユーザビリティテスト				○			●使い勝手の良さ/悪さを評価
シナリオテスト	○	○	○	○	○	○	●評価対象全体を巨大なブラックボックスと見なしてシナリオを設定して総合的に評価

### 3. テスト観点の考え方（一例）

成果物	評価対象	目的機能	検証アングル	テストパラメータ	確認ポイント
	①何の	①何を	②どんなアングルで	③何が、 ④どんな状態の時	⑤何が、どうなるか
テスト設計仕様書	ログイン/ ログアウト機能	—	準正常ケースを検証する	—	—
	踏襲 ↓	↓機能分割	踏襲 ↓	↓案出	↓案出
	ログイン/ ログアウト機能	ログイン 排他制御機能の	準正常ケースを検証する	同一ID：ログイン中 同一ID：更にログイン	ログインできない
テスト明細	ログイン/ ログアウト機能の	ログイン 排他制御機能の	準正常ケースを検証するために	同一ID：ログイン中 同一ID：更にログイン	ログインできない
	踏襲 ↓	踏襲 ↓	踏襲 ↓	↓手順化・具体化	↓具体化
テストケース	ログイン/ ログアウト機能の	ログイン 排他制御機能の	準正常ケースを検証するために	同一IDがログイン中の状態で、 更に、同一IDが別端末でログイン操作すると、	ログインエラー画面を表示することを確認する

開発プロセスの段階に応じて  
詳細化・具体化していく



# まとめ

- ▶ テストを意識した要求の分析
- ▶ 「何ができればOKか」を上流工程から考える
- ▶ アジャイル開発では要求（プロダクトバックログ）の「完成」の定義と受け入れ条件で整理
- ▶ バルテスではテスト観点をベースに発展させる
  - アジャイル開発では受け入れ条件に適用
  - ウォーターフォール開発ではテスト設計の段階的詳細化に適用

ご清聴ありがとうございました