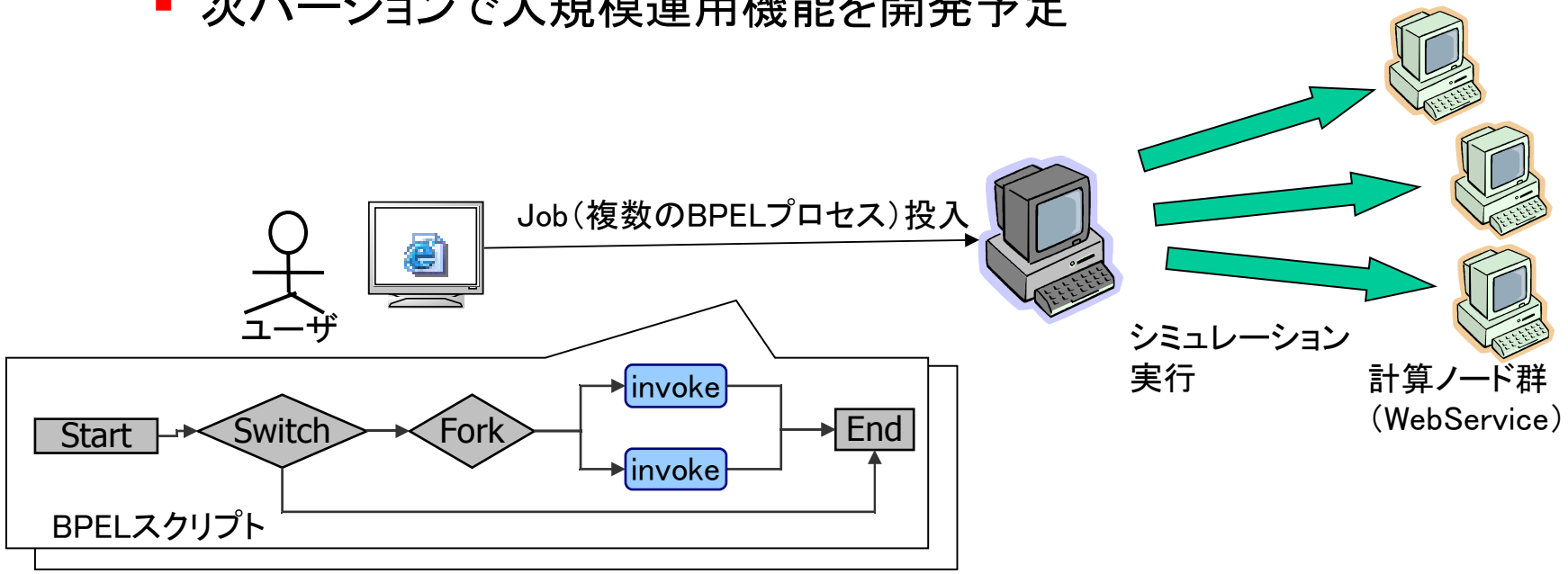
TopSE修了制作

要求仕様の精度向上の試み
～シミュレーション投入システムへの適用～

キヤノン株式会社
井上拓
inoue.taku@canon.co.jp

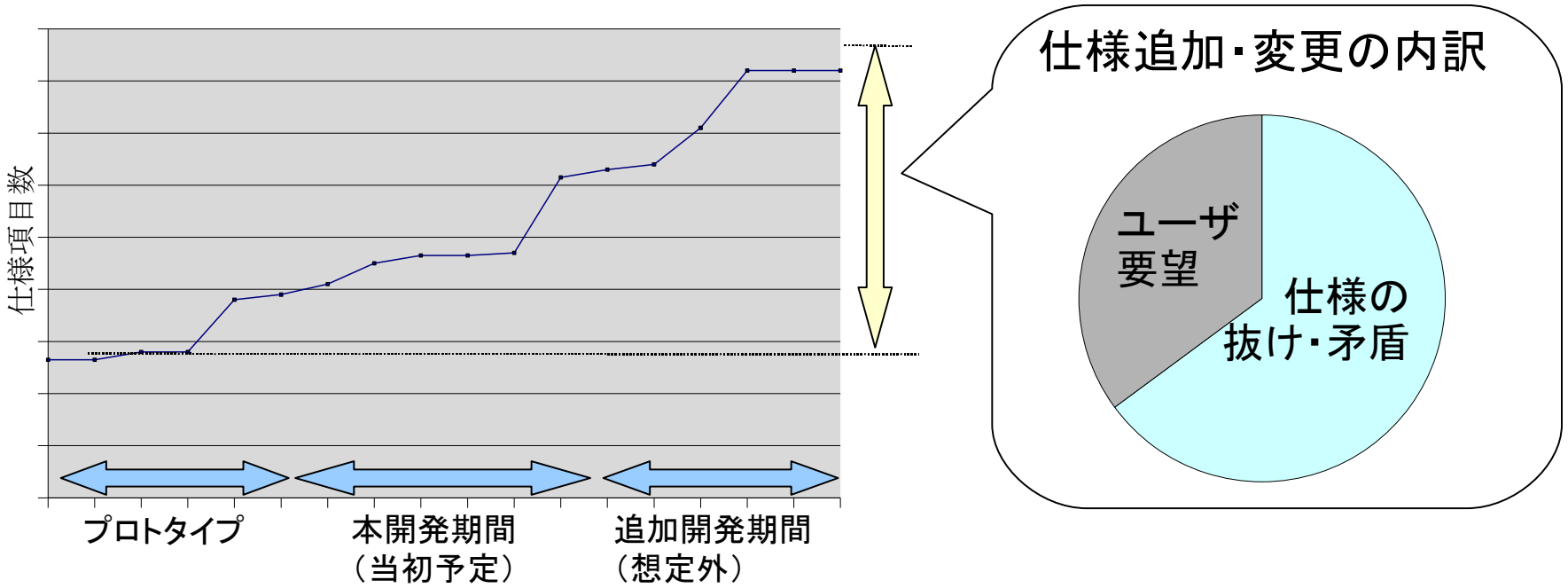
対象領域

- シミュレーション投入システム
  - ブラウザ経由でシミュレーション投入
    - シミュレーションはWebServiceでI/Fされている
    - BPELで記述した連携手順を実行
  - 基本機能の開発は既に完了
  - 次バージョンで大規模運用機能を開発予定



従来開発の課題

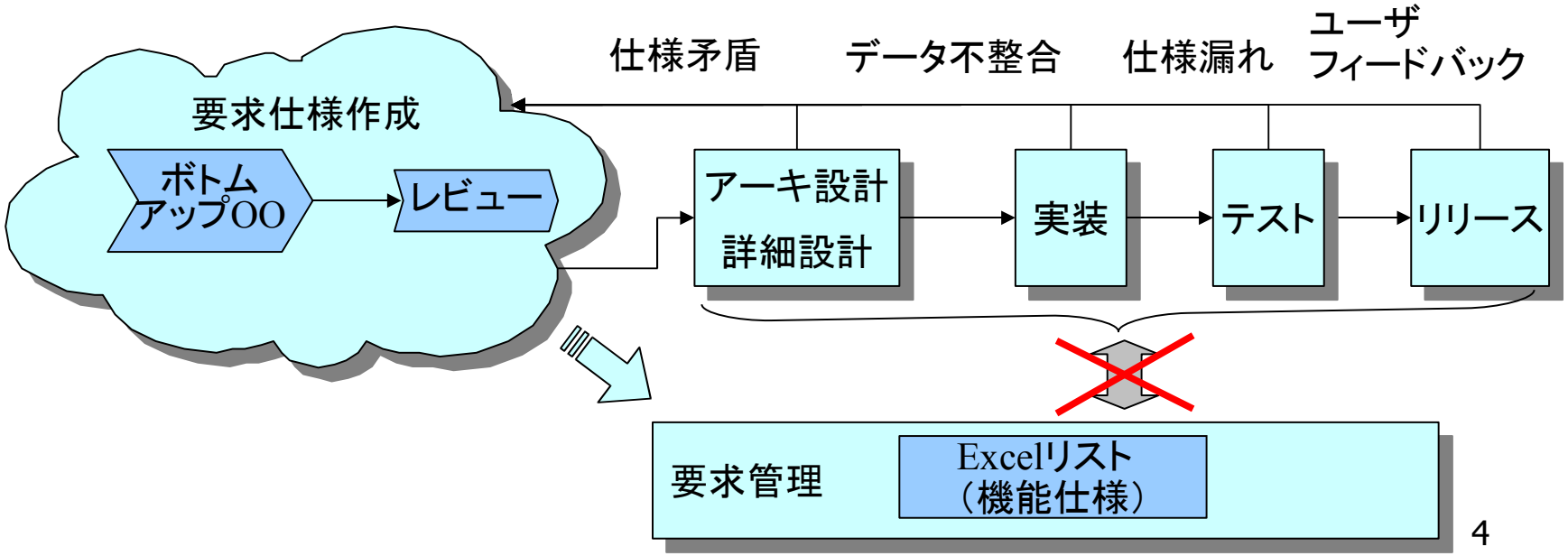
- 開発後半に要求仕様の追加・変更が頻発
  - 仕様の抜け・矛盾によるものが6割強（特にデータ不整合が顕著）
  - アーキテクチャの大幅修正／予期せぬ追加開発を招いていた



精度の良い安定した要求仕様モデルを確立したい！

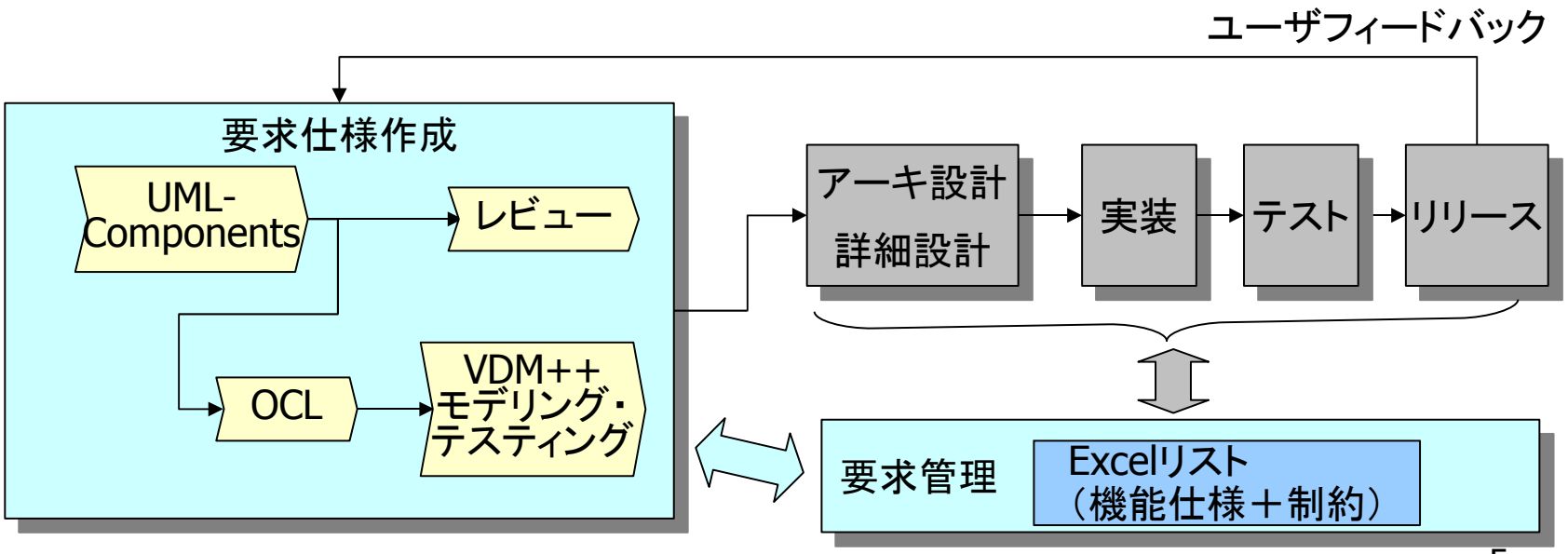
従来の開発プロセス

- 要求仕様の精度不足に起因する手戻りが大
  - ボトムアップ的なOOアプローチ、データ制約の発見が個人のスキルに依存
  - クラス図／シーケンス図／状態遷移図では表現しきれないケースあり
  - レビューで発見できなかった仕様バグが後工程／次Iterationで発覚
- 要求管理の粒度が荒い
  - 後工程の進捗管理や実装の優先度付けに不十分



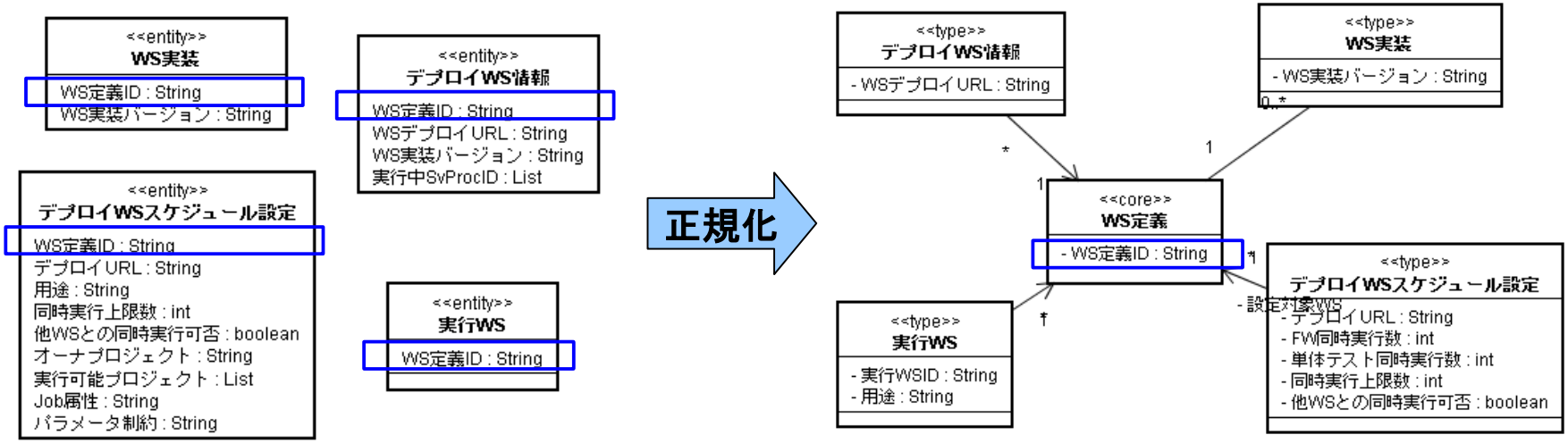
試行した開発プロセス

- 複数の仕様モデリング方法の併用
  - UML-Components： 安定したアーキモデルを早期に構築
  - レビュー： 従来開発と同様にUMLモデルに対して実施
  - OCL： データの整合性を分析
  - VDM++モデリング／テスティング： 形式記述により仕様の妥当性を確認
- 要求仕様モデル間のトレーサビリティの確保
  - 要求管理リストの活用



UML-Components

- 軽量なコンポーネントベース開発方法論
  - データ指向のアーキテクチャ構築に有効
  - 要求⇒仕様（⇒提供⇒組み立て⇒テスト⇒配備）
- 独自の工夫点
  - Coreタイプ抽出時にモデルを正規化し、データ間の依存関係を明確化
  - ライフサイクルを考慮したコンポーネント分割（分散協調モデル）



ビジネス概念モデル　　ビジネスタイプモデル

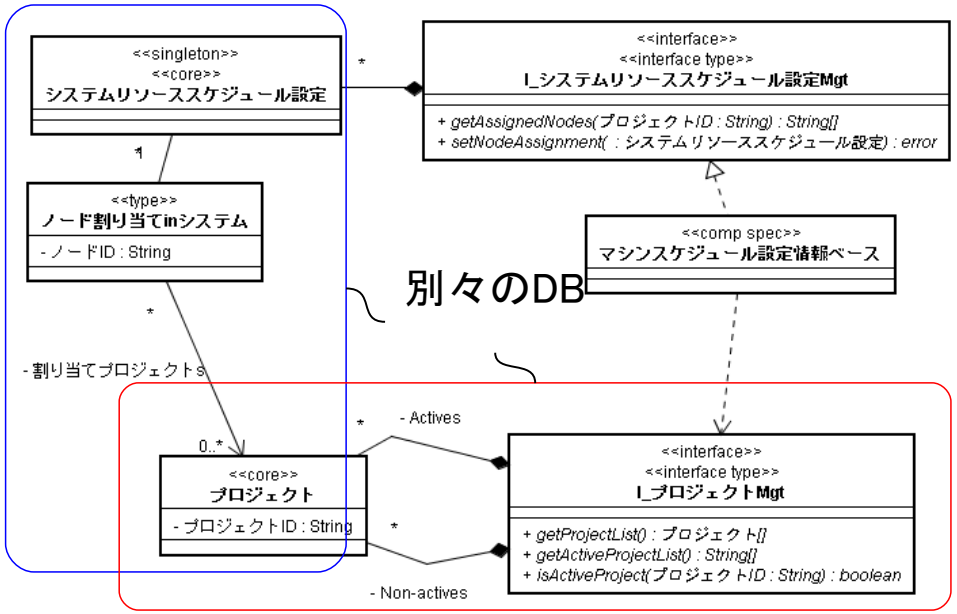
レビュー（UML-Components）

- 出席者： 社内の開発チームメンバ＋外部委託先
  - ドメイン知識は豊富
  - 初回レビュー時にUML-Componentsについて説明、抵抗なく受け入れられた
- レビューで発見されたモデルの抜け/制約
  - ユースケースレベルの抜け
  - 複数IFに跨るシナリオでの抜け
  - 実装上の制約（Webコンテナの仕様など）

  } 広範囲を効率的にチェック可能
- 要求管理リスト（Excel）の有用性
  - UMLモデルだけではレビューア毎に異なる解釈をする場合がある
  - 自然言語による文書化＋口頭説明は必須
  - モデルの根拠、対案は記載必須

OCL

- オブジェクト制約言語
  - UMLモデルに制約事項を追加することが出来る
- 抽出した制約
  - 分散DBに保持されているデータ間の整合性
    - オブジェクトの同一性ではなく属性の同一性で制約を表現
  - 例：システムリソーススケジュール設定中の全てのプロジェクトがActiveである



UML記述（クラス図）

```
context マシンスケジュール設定情報ベース::
   changeSysResSchedule(システムリソーススケジュール設定)
pre: システムリソーススケジュール設定.
      ノード割り当てinシステム.割り当てプロジェクトs->
   forall(elem| I_プロジェクトMgt.Actives->
      exists1( prj | prj.プロジェクトID=elem.プロジェクトID))
```

OCL記述

VDM++

- 状態型の形式仕様記述言語
  - Explicitなモデルを実行することにより仕様のテスティングが可能
- UML-Componentsモデル→VDM++の変換ルールを定義

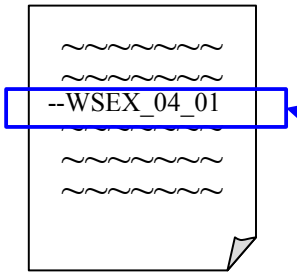
	UML-Components上の表現	VDM上の表現
①	システムインターフェース， ビジネスインターフェース	抽象Class
②	Comp spec	具象Class
③	Coreタイプ,typeタイプ	Class（属性は全てPublic）
④	String, Int等、	Token型
⑤	多重度が多の関連	Map

- 抽出した仕様バグ
  - 組み合わせパターン（設定値、状態）の漏れ
  - 複数IFに跨るシナリオ／時系列的な制約の抜け
- VDM++モデリング・テストの効能
  - 仕様カバレッジを計測できる
  - 実際に動かして検証できる

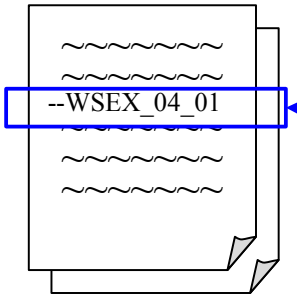
} ロジックの抜け・思い込みの防止に有効

要求管理

- 要求管理リストを用いてモデル間のトレーサビリティを確保
  - モデリングによって詳細化された要求仕様を、リストに反映
  - 各モデルに要求管理IDを保持
- 設計・実装以降の工程での進捗管理にも利用
  - 仕様の詳細化により、管理精度が向上



OCL記述



VDM++記述・テストコード

OCL・VDMにより詳細化された仕様項目

各種モデルの有無

優先度付け・進捗管理に利用

仕様ID	カテゴリ	内容	備考	関連ID	UML	OCL	VDM	Test	優先度	ステータス
WSEX	サービス実行									
WSEX_01		…							…	…
…		…							…	…
WSEX_04		計算ノード側で、リクエスト内容／ノード状態／スケジュールポリシーが整合していることを確認する			X					
WSEX_04_01		リクエスト元のプロジェクトが、該デプロイWSのオーナプロジェクトか実行可能プロジェクトのいずれかと一致する必要がある	【理由】…	MSCH_01		X	X	X	高	実装済み
WSEX_04_02		単体テスト用のWS実行数が、ノードスケジュール設定中の上限値未満かチェックする。	【代替案】…	MSCH_01		X	X	X	中	設計済み
WSEX_05		…							…	…

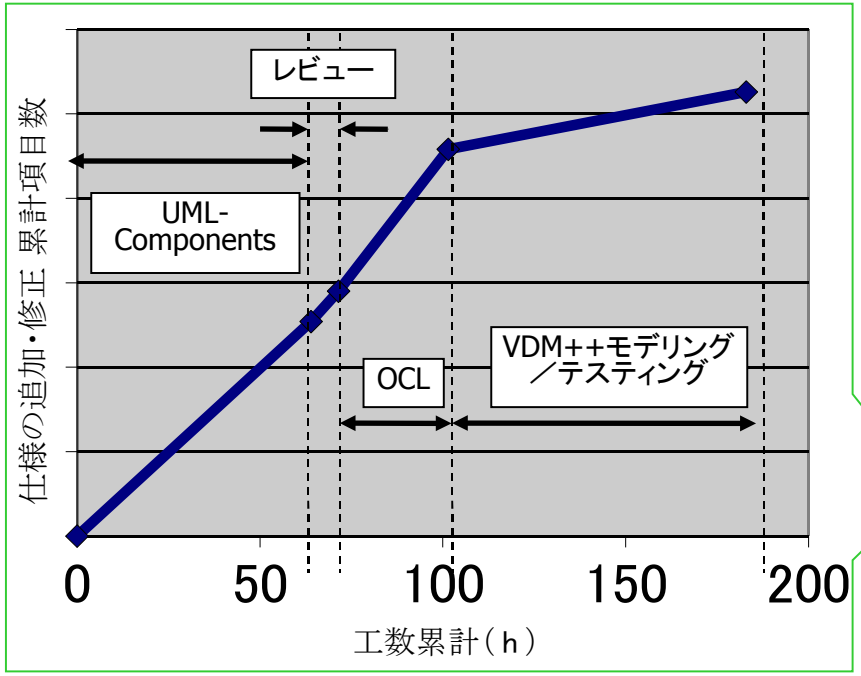
要求管理リスト

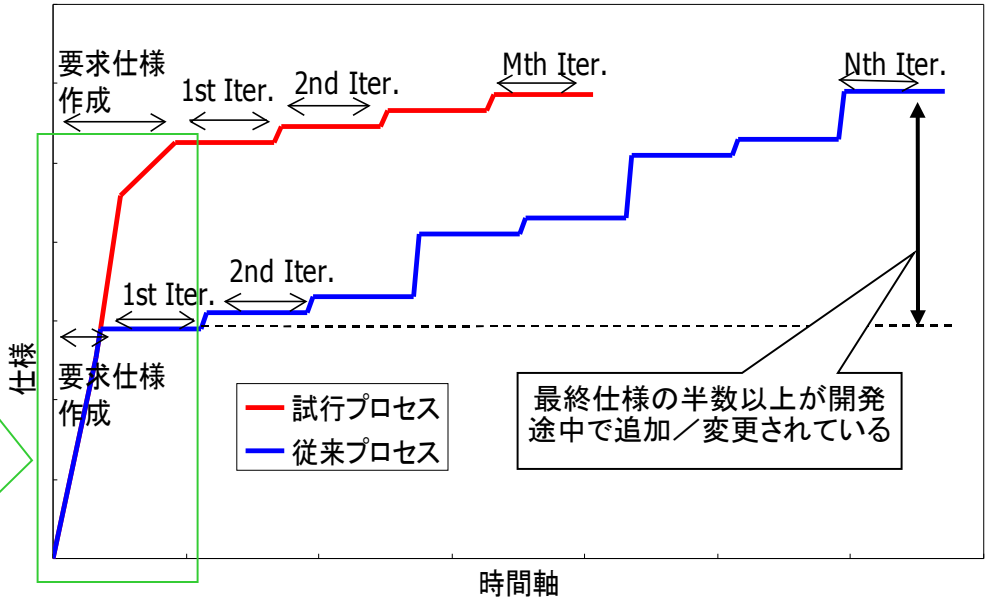
仕様の精度とモデル作成工数

- 仕様項目数(≒精度) ⇒ 従来の約1.8倍
- モデル作成工数 ⇒ 従来の約2.5倍

UML-Components+レビューを従来プロセス相当とした場合

- 仕様の精度向上を実現、ただし工数との間にトレードオフ
- 期待効果： 設計以降の手戻りレスにより全体の開発期間が短縮





まとめ

- モデリング手法の併用により要求仕様の精度向上が可能
  - ドメインの性質やプロジェクトの日程制約を考慮したテーラリングが必要
- 今後の展開： 組み込み領域への適用検討（並行性・HW要件を考慮）

並行性・厳密性

仕様網羅度の計測

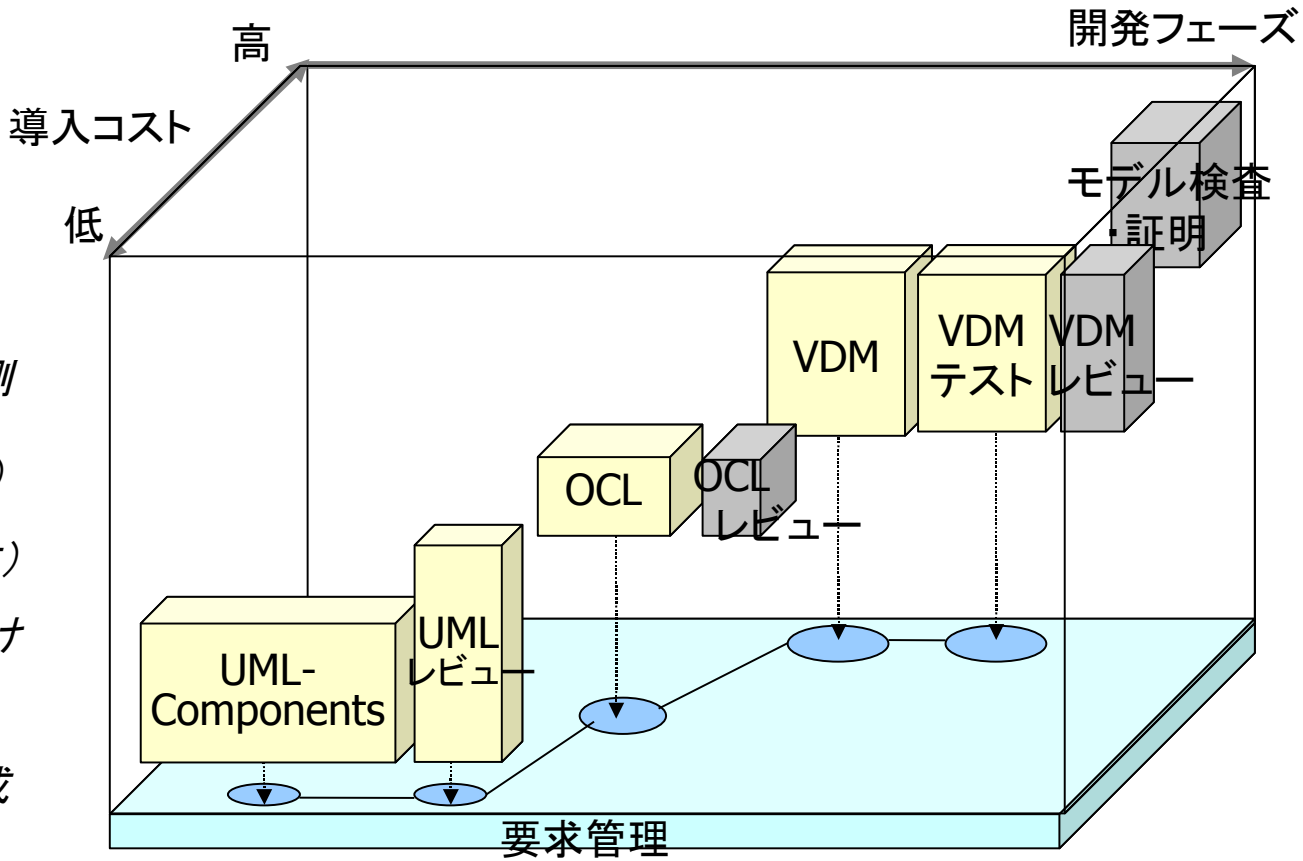
データ制約（複合）

データ制約（IF単体）

ユースケースの抜け

理解容易性

概念モデルの作成



End