

開発手法の開発へと繋がる 現場改善の歩み

三菱電機メカトロニクスソフトウェア株式会社

岩橋正実

Iwahashi.Masami@wak.msw.co.jp

目次

1. 現場改善の歩み
2. 現場改善のポイントと技法紹介
3. まとめ

1. 現場改善の歩み

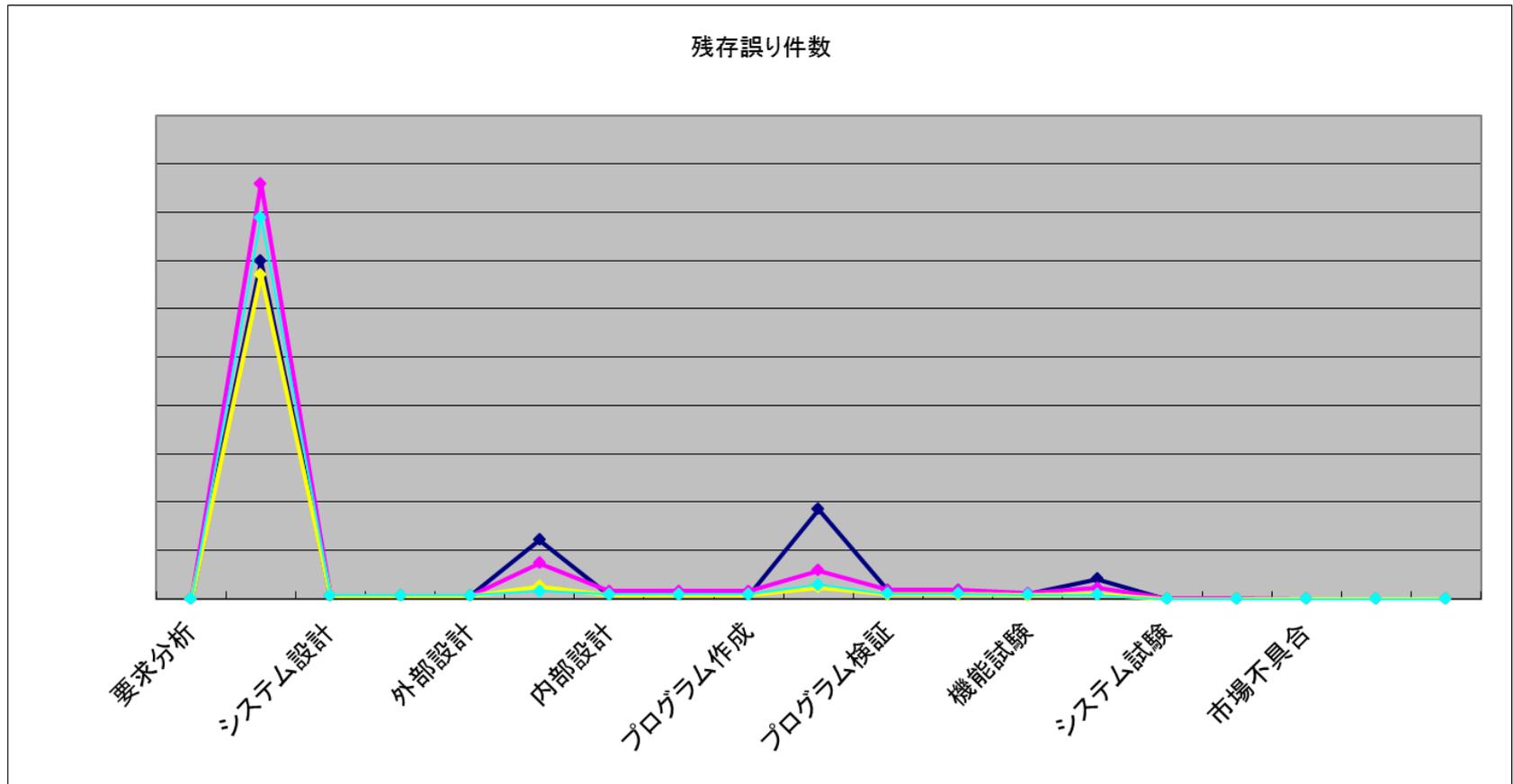
技法適用効果のご紹介

①残存不具合のトレンド

上流不具合検出率:約98.7%、プロセス実行品質向上(検出率低下)

②工数分析

分析約40.6%、試験約25.7%



改善障壁と回避ポイント

①手法開発の障壁

→自主研究とOpen化

②新技術導入の壁

→導入目的の明確化と技術開発

③改善リスク

→改善目的と放置リスクの明確化

④人的要因の回避

→課題の見える化と改善目的の共有

⑤組織の壁(単一プロジェクトの改善では意味が無い)

→改善効果の見える化

→改善の先を見る

Open化による第三者評価
目的指向(本質を極める)
見える化
段階的な改善の推進
経営層向けの提案書

現場の課題解決から手法の確立へ

ツール/手法の適用のみでは、現場の課題は解決しない

①改善が進まないのは組織特性の違い

国民性の違いによる開発手法の成り立ちが異なる

国内組織間の特性差により必要とされるツール/手法は異なる

②組織の課題を特定することで課題解決が進む

課題発生源を特定して課題を発生させない方法の構築

③課題解決方法の一般化が手法確立へと進む

自部門内の課題抑制技術を他部門/他会社へ展開を進める事で手法の確立が進む

④確立手法のオープン化による技術革新

論文・書籍・セミナー等で確立手法をオープンにすることでオープン技術をベースにした更なる技術革新に繋がる

2. 現場改善のポイントと技法の紹介

手法導入によりフロントローディングを極める

人によるソフトウェア開発作業のバラツキが発生。
その結果、テストによる品質確保が困難になる。

①試験品質【試験設計要因の分析】

試験設計の品質向上による不具合除去

②レビュー品質【レビュー設計要因の分析】

分析・設計・実装・試験のレビュー品質向上による不具合除去

③作業品質【不具合発生源の分析】

分析・設計・実装・試験の作業品質向上による不具合
作り込み除去

④要求品質【要求の発生源の分析】

要求品質を向上する為の要求の作り込みと検証品質向上に
よる価値のある製品構築

課題解決型のプロセス定義と形式化

不具合の発生源を特定して作り込み要因を分析して課題を定義。課題を作り込まない技術開発を行い、アーキテクチャ定義と課題解決のタスクの形式化により組織全体の品質を改善する。

- ① 不具合の発生源の特定
- ② 作り込み本質要因を特定
- ③ 課題を定義
- ④ 課題を作り込まない作業定義（手法定義）
- ⑤ 手法を安定化させるアーキテクチャを定義
- ⑥ 課題解決を形式的に実行できるプロセス定義
- ⑦ 形式化したプロセスのシステム化

過去不具合の発生源と要因分析。同種の不具合を分類して課題を作り込まないタスク定義により開発プロセスを構築した上で継続した課題解決プロセス定義を進める事が大切です。

課題解決型のプロセス定義と形式化

課題を抑制する技法を用いる事でソフトウェア品質を向上させる

課題		課題解決方法
開発のバラツキ	要求定義のバラツキ	要求分析定義手法
	工程間の変換バラツキ	開発プロセス
重複記述		クラス分析設計手法
要求・S/Wの発散		SPL(ソフトウェアプロダクトライン)
文書精度(曖昧表現)		日本語による形式記法の推進DSL
要求精度		目的指向開発
競合及び制約	機能競合	競合分析設計
	出力競合	
	例外競合	
	時間制約	絶対時間分析設計
安定しない要求	段階的仕様詳細化	仕様安定度に基づくプロセス最適化(イテレーティブ開発)
	段階的仕様追加	変更粒度によるプロセス最適化(インクリメンタル開発)
見積り精度		見積り手法

技法の解説

自律オブジェクト指向 (AOO: Autonomic architecture base Object-Oriented development technique) が1998年に組み込みソフトウェア開発向けオブジェクト指向の開発手法として発表。その後、AOOは、プロセス、パターン、見積り、SPL、DSLへ拡張。組み込みソフトウェアの最強のオープンな開発手法として総称をA「エース」(Autonomic architecture base embedded software development technique)と呼んでいる。

Autonomicは、「自律」という意味と「自律神経」という意味も含まれる。自律は、他からの支配・制約などを受けずに、自分自身で立てた規範に従って行動するアーキテクチャを基準に開発手法を定義。

要求を1機能1目的で階層的にカテゴリで分類整理して、機能間に依存関係を持たせず自分自身の機能目的達成のための要求を定義する。

更に、共通機能、処理ブロック(操作)、名詞(属性)、バリエーション(ポイント+バリエーション)を定義して要求モデルの洗練化して日本語によるDSL化を進める。

要求から変換されるオブジェクトも目的単位で自律して振る舞い、オブジェクト間の関連は、静的結合で自分自身の1つの目的達成の為に振る舞う。

要求を人の認知方式に基づきモデル化され自律神経モデルに基づきフレームワークに変換することで要求・設計・実装・試験への双方向の紐付けを可能にして、高品質を確保した上で派生機種開発、SPL開発を可能にする。

技法の解説

要求を1機能1目的として階層的に機能ブロック(操作)、属性、時間制約、変換パターン、etcのプロパティを定義して要求定義での洗練化を進めて想定可能な競合・例外事象を定義、製品内・製品間でのバリエーションを定義することで高品質・高生産性を確保する

目的指向開発

要求機能のモデル化

機能項目/物理項目を共通目的で項目のカテゴリ化による要求定義

形式記法によるDSL化

機能ブロック・名詞の定義と条件及び操作の形式記述化と検証。共通機能ブロックの再利用

SPL分析設計

製品間の仕様のバリエーションポイント(仕様の可変ポイント)とバリエーション(可変仕様)の分析定義

絶対時間分析設計

物理・機能の時間制約の定義と解決

状態分析設計

機能目的に基づく状態定義の形式化と状態に基づく操作の形式記述化

機能競合分析設計

機能競合の解決
例外競合の解決

クラス分析設計

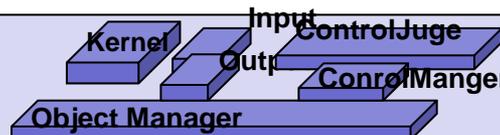
カテゴリでの機能の整理。共通機能ブロック・名詞の定義とクラスへの変換

分析設計変換パターン

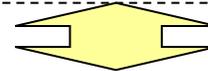
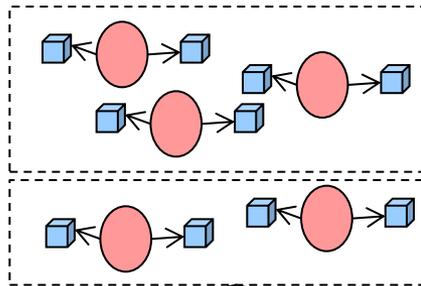
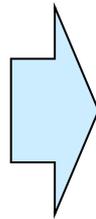
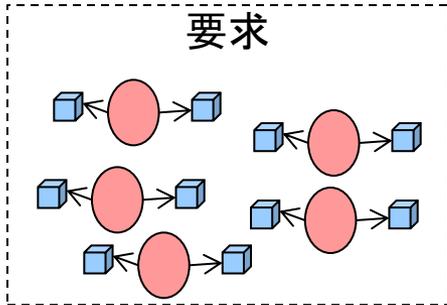
機能項目からオブジェクトへの変換パターンにより双方向のトレーサビリティを確立

フレームワーク

実装基盤



制御対象物
及び
外部環境



技法の解説

1) 課題抑制のアーキテクチャとプロセス定義(課題発生源特定)

アーキテクチャは、特定の課題を解決する為の要素の構造と要素間の関係を定義したもので、課題解決に必要なフレームを文書様式として形式化。課題解決の様式(アーキテクチャ)を使用する作業フレームを開発プロセスに定義することで高品質なソフトウェアの開発を実現。

2) 要求の目的定義による超上流改善(要求発生源特定)

文書様式で要求の目的を定義することにより要求の発生源を特定して製品仕様品質の向上を図る。要求の発生源を特定して目的を定義することにより下記の効果が期待でき、仕様の安定化と高品質化による生産性改善と市場要求の対応レスポンスの向上による市場規模拡大。

- ・仕様の変動が抑制される。
- ・仕様の最適化の提案/仕様の品質が向上。
- ・仕様の価値が判断でき対応優先度が決まる。

3) 目的指向によるクラス設計と競合解決

4) アクティビティ・タスクの目的定義による無駄取り

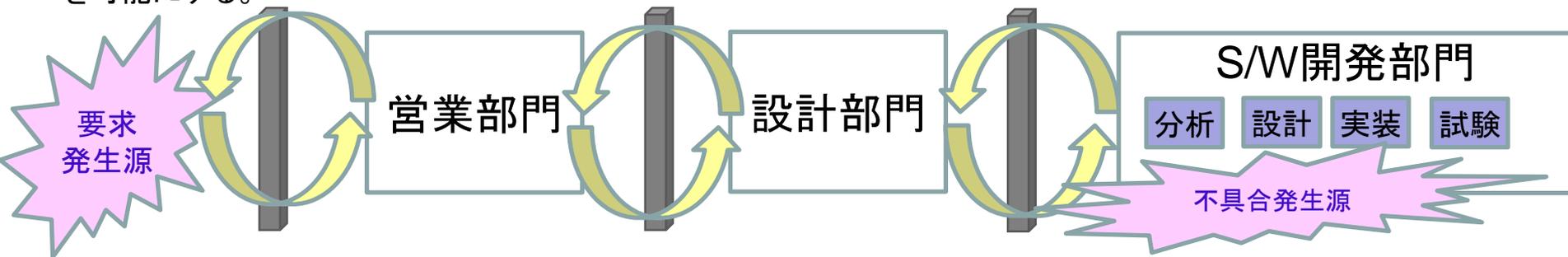
目的指向による不具合発生源及び要求発生源の特定による不具合を再発させないプロセス定義と要求の変動を抑制するプロセス定義によるロスコスト防止。工程間の重複作業の排除、タスクの実行目的を明確にしてプロジェクト特性に合致したタスクの選択による徹底した無駄取りを実現して高い生産性を実現。

5) 変更粒度・仕様安定度に応じた実行タスクの最適化

6) 派生機種の可変ポイントと可変部の定義によるSPL開発

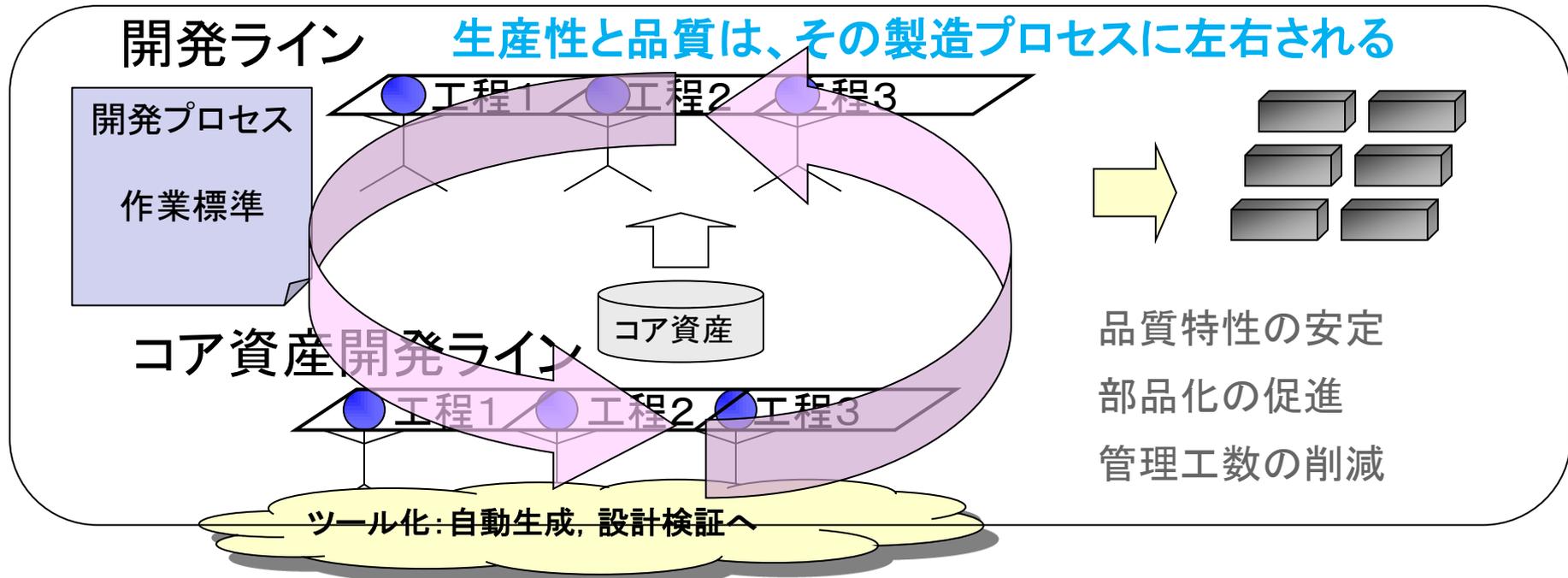
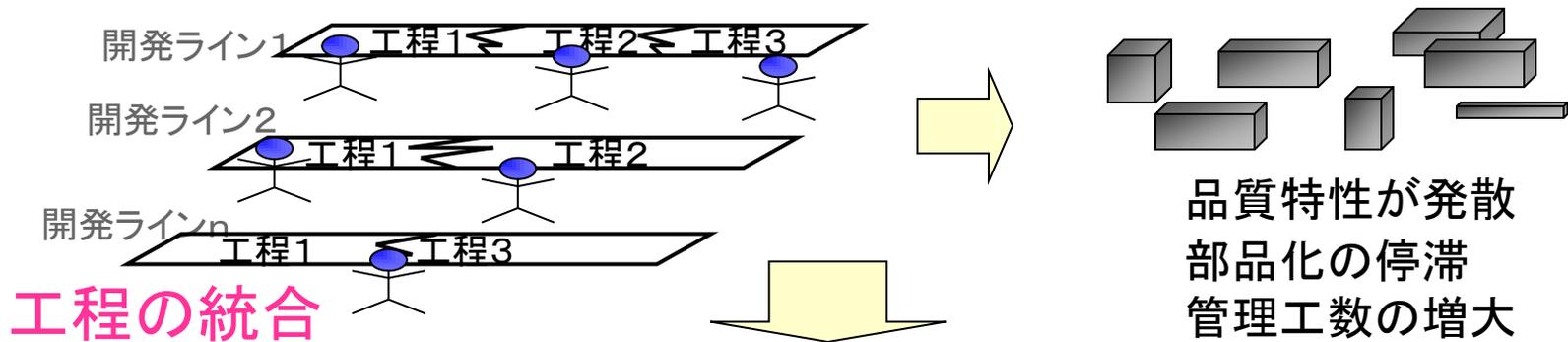
7) プロセス形式化と要求定義・検証の形式化

形式記法を用いて要求分析定義書の記述精度(文書共通性と品質)を向上。高品質な要求分析定義書を基に双方向のトレーサビリティを確保する事で定義された文書は、複数の工程で異なる視点で分析され高品質なソフトウェア開発を可能にする。

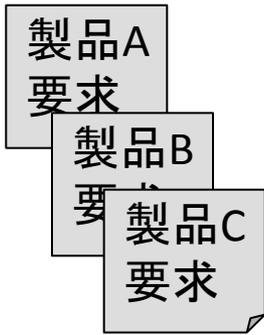


技法の解説(SPL)

- ①開発手順のバラツキにより品質が安定せず管理負荷が増大
- ②開発上の課題発生への傾向が不安定で品質が安定しない
- ③要求及びソフトウェアが不安定で共通化が進まない



技法の解説(SPL)

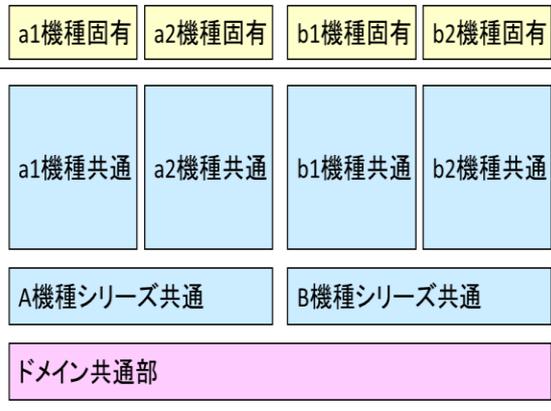


1機能項目1目的で
カテゴリで分類整理



機能項目		機能仕様				バリエーション ポイント	バリエーション ID	機種A	機種B	機種C	機種D
		目的	操作	目的	属性						
1.xxx	1.1.xxx					標準	0				
						開始条件 北米	1	0	0	1	2
						欧州	2				
物理項目		機能仕様				バリエーション ポイント	バリエーション ID	機種A	機種B	機種C	機種D
		目的	操作	目的	属性						
1.xxx	1.1.xxx										
2.xxx	2.1.xxx	2.1.1.xxx									

オブジェクトに変換(可変性の継承)



オブジェクト		機能仕様				バリエーション ポイント	バリエーション ID	機種A	機種B	機種C	機種D
		目的	操作	目的	属性						
1.xxx	1.1.xxx					標準	0				
						開始条件 北米	1	0	0	1	2
						欧州	2				
2.xxx	2.1.xxx	2.1.1.xxx									
1.xxx	1.1.xxx										
2.xxx	2.1.xxx	2.1.1.xxx									

製品間の可変性を分析して可変ポイント(バリエーションポイント)と可変部分(バリエーション)を定義する

- ①要求の物理と目的(機能)で分解整理してソフトウェアと双方向に紐付することでSPL開発を可能にする。
- ②フレームワークをドメイン依存として構築すると中長期開発で破綻するリスクがある。
状態遷移、状態に基づく操作、競合解決、バリエーション解決は、多くの組込み制御システムに共通であり、共通フレームワーク上で実現することでSPL開発を成功させる。

技法の解説(DSL)

1) 要求のカテゴリ化と機能ブロック/属性の形式化

2) 形式記法を用いた機能仕様の日本語による形式化

3) 機能ブロック/属性の再利用によるDSL化の推進

機能項目リスト

機能項目		目的	操作	目的	属性
1.xxx	1.1.xxx				
	1.2.xxx				
2.xxx	2.1.xxx				
	2.2.xxx	2.2.1.xxx			
		2.2.2.xxx			
制御マネージャ					

物理項目リスト

物理項目		目的	操作	目的	属性
1.xxx	1.1.xxx				
	1.2.xxx				
	1.3.xxx				
2.xxx	2.1.xxx	2.1.1.xxx			
		2.1.2.xxx			
デバイス					

1機能項目1目的として
階層的にカテゴリ化を進める

機能
ブロック

機能
ブロック

機能仕様で扱う名詞の定義して形式記法で仕様定義
更に1機能項目内を目的で分解して機能ブロックを定義
機能ブロックの再利用による要求分析定義の生産性向上
※文書コード生成検証の実現

DSL

ドメイン
形式記述

形式記述言語

①and②

①外気温度 > 10°C

②運転モード = 冷房

・XXX制御状態 ← 制御中

【XXX制御状態 = 制御中】

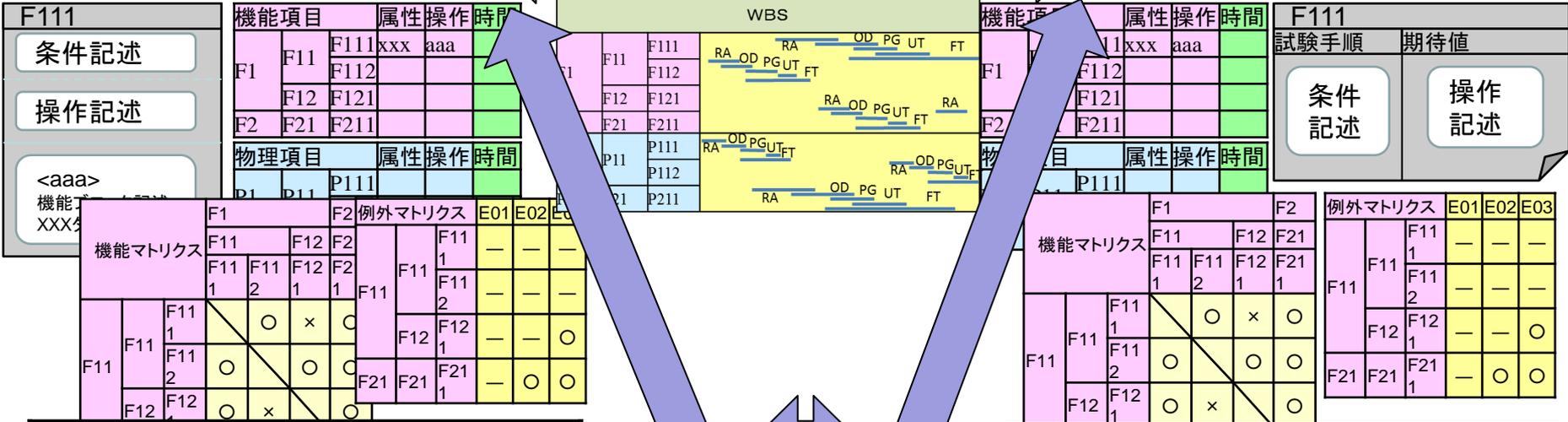
SelectSetDataTB:

JD		SD	
JV		SV	
JV		SV	
JV		SV	

開発プロセス

ソフトウェア要求分析

ソフトウェア総合テスト



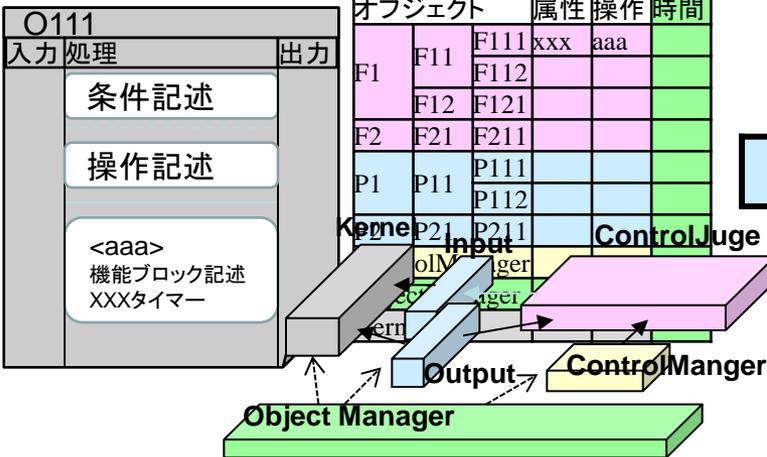
ソフトウェアアーキテクチャ設計

ソフトウェア結合および結合テスト

ソフトウェア詳細設計

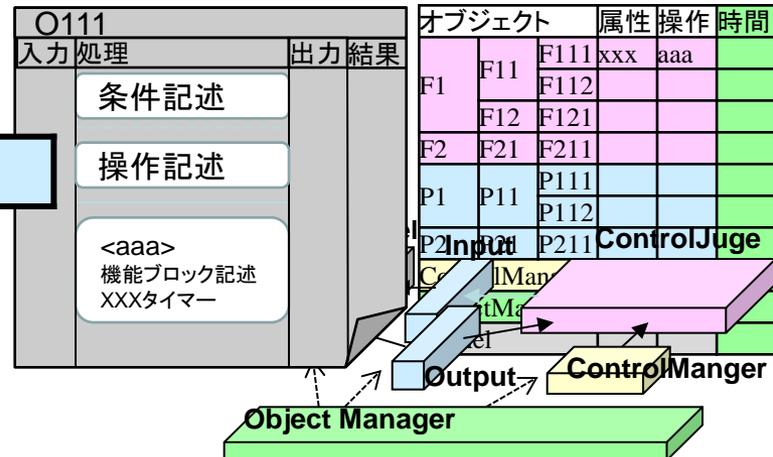
単体テスト

実装



```

O111(ソースコード)
if(XXXX)
    aaa=st-xxxx*2
;
aaa();
XXXXXXXXXX;
XXXXXXXXXXXX;
Xxx_ti=START;
    
```



プロセスによる組織文化の構築

優れた企業は、創設者が社員に伝えた仕事のやり方（プロセス）を脈々と継承して組織文化を構築している

品質が安定しない未成熟な組織でも、組織の課題を特定して、組織全体の仕事のやり方を開発プロセスとして展開する事で組織の成熟度を上げ組織文化を構築することが可能です。

組織文化構築のポイント

- 1) 課題解決型プロセス改善
課題の発生源特定して課題を作り込まない継続的なプロセス改善
- 2) 客先思考開発
要求発生源特定による超フロントローディングの推進
- 3) 目的指向開発
目的明確にして本質を見極め最適化と徹底したムダ取り

4. まとめ

改善と開発プロセスのまとめ

目的を持って継続して改善し続ける

目的を定義して本質を見極める

目的達成の価値を判断する

同種の目的で整理して共通部と本質定義を実施する

作業目的でタスクを階層的に分類整理してプロセスを定義する

各開発工程の作業目的の定義により最適化を実施する

ドメイン非依存のプロセス定義とドメインの分類整理による
カテゴリ単位のプロセス定義を実施する

製品シリーズA
開発プロセス

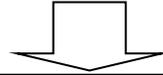
製品シリーズB
開発プロセス

ドメイン非依存
開発プロセス

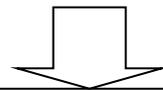
まとめ

品質・生産性改善手法の確立:

課題と要求の発生源の本質分析を実施して課題・要求の解決策の設定と実行



- ・課題を再発させないためのタスク定義
- ・共通アーキテクチャとフレームワークの徹底
- ・分析/設計/実装/試験の双方向トレーサビリティ確保
- ・要求分析の形式記述
- ・仕様のバリエーション定義
- ・仕様の変更粒度と安定度で最適プロセスの実行
- ・開発プロセスの安定化(形式化)
- ・開発プロセスの継続的な改善管理



プロセス管理負荷軽減と精度向上
品質・生産性の向上と安定
戦略的製品開発へ

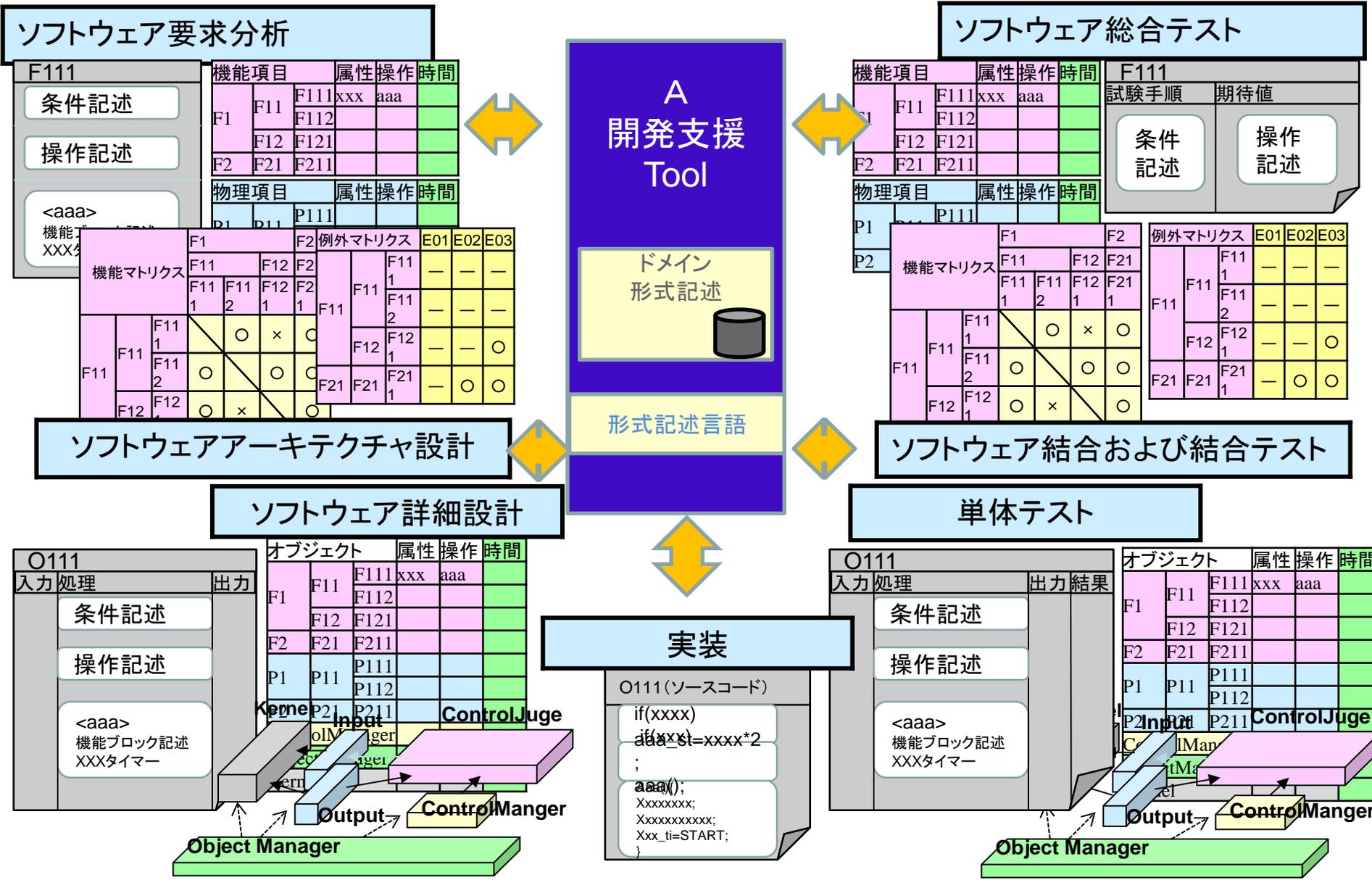


コア資産



プロセスの定型作業化
自動生成・検証の実現
モデル駆動開発へ

A開発支援ツール



ご清聴ありがとうございました。

三菱電機メカトロニクスソフトウェア株式会社

岩橋正実

073-436-0776

Iwahashi.Masami@wak.msw.co.jp