

本スライドは、当日の講演資料の抜粋版です。

# 組込みソフトウェア開発の勘所 ～ソフトウェア開発に必要な3つの技術～

実世界のモノのイメージから技術の大切さを実感して学び取る

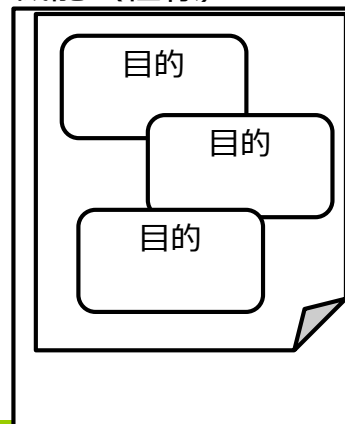
三菱電機株式会社  
設計システム技術センター  
岩橋正実

# ソフトウェアの開発が困難な理由

仕様とソフトウェアの関係の解析が困難であり曖昧な仕様記述や重複・不整合がある。

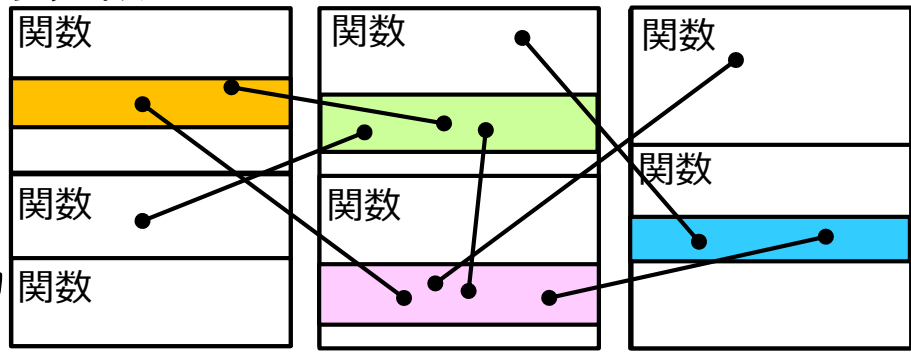
- **整理**されていない曖昧な仕様記述に重複や不整合がある。
- 機能(仕様)が目的で**整理**できていない
- 実世界の用語をソフトウェアで使用(**定義**)しない為、変更箇所の特定が困難。
- ファイル・関数・変数の意味が認識できるラベル/名称が**定義**できていない
- 1つの機能が複数のファイル内の複数の関数に実装(**変換**)されている。

機能 (仕様)



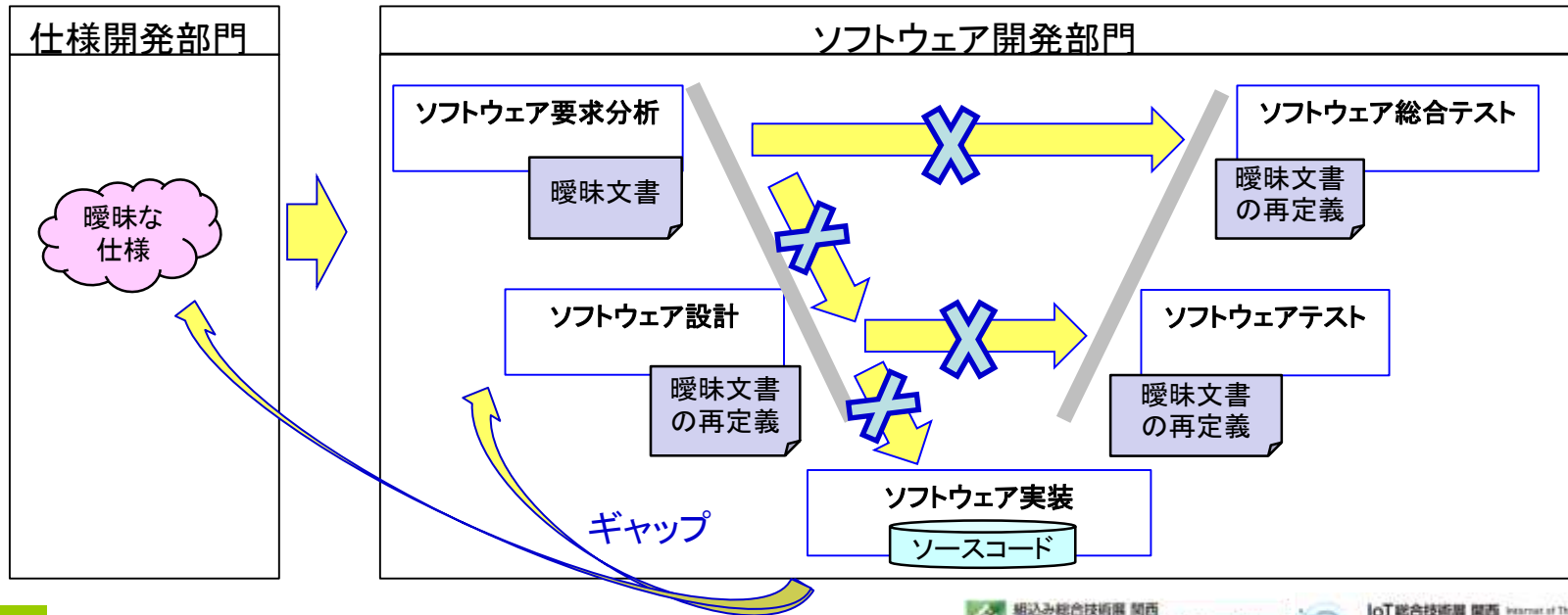
ソフトウェア

ファイル



# 整理/定義/変換技術が解決する課題

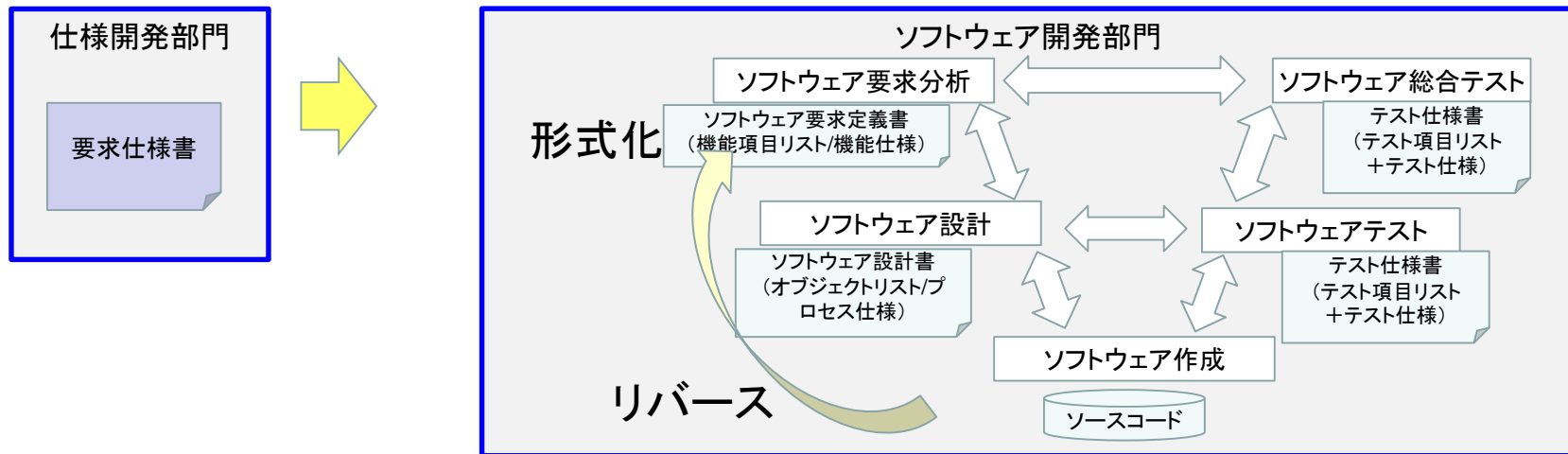
- ① 仕様/設計書記述の重複と不整合 **【整理する技術】**
- ② 仕様/設計書の曖昧記述 **【定義する技術】**
- ③ 仕様/分析/設計/実装/テストのギャップ **【変換する技術】**



# 品質生産性向上に即効性がある開発方法

要求を機能項目リストで整理してベースラインを確立する。仕様が安定した項目からソースコードを解析して記法と表形式による形式化で機能仕様をバースする。リバースした機能仕様に基づき変更開発を実施。仕様記述を設計およびテスト設計記述に使用する。

- **分析精度向上**: ソースコードから機能仕様をリバースすることでソースコードの分析精度向上とモデルベース開発への移行
- **仕様取違い抑制**: リバースした機能仕様を仕様開発部門と読み合わせることで仕様の取違いを抑制
- **手戻り抑制**: 仕様不安定時はソースコード修正を優先して分析/設計/テストを簡略化
- **仕様部品化**: 機能仕様を分類整理することで重複部分の整理と仕様の部品化を実現
- **重複記述の抑制**: 機能仕様を設計/テストに使用することで重複記述の抑制と不具合の混入リスクを低減



# AOO\_DSL モデル記述構成

**STEP3:**仕様から抽出したドメインで共通な仕様の目的を示す名称を付加して表形式モデルとする。共通仕様を隠ぺいして可変部のモデル化により仕様開発スピードおよび可読性の向上を図る。

## 表形式モデル

<仕様の可変点のモデル化>



**STEP2:**仕様の部分記述を機能ブロック及び処理ブロックとして目的を表す文言で抽象化と厳密化を実現させて可読性を向上させる。  
更に機能ブロック/処理ブロックのライブラリ化を実現して仕様の再利用を促進させる。

## ライブラリ化

<機能/処理ブロックの仕様記述のライブラリ化>

<正常運転判定>

- ① and ②
- ②運転/停止指令 = 運転
- ③異常状態 = 正常

**Step1:**形式記述の記載に意味がある仕様定義を表形式モデルとすることで可読性を向上させる。

## 仕様定義フレーム

JG		OP	
JG		OP	

JD		SD	
JV		SV	
JV		SV	

**Step1:**日本語による形式化と仕様と設計の要素を関連付けることによりモデルとコードの双方向のトレーサビリティを確立してモデルベース開発を実現する。

## 形式記法

- ① and ② and ③
- ①<異常切替> = TRUE
- ②運転/停止指令 = 運転
- ③異常状態 = 正常
- ・異常状態 ← 異常停止

- <異常切替>
- ①and②and③
- ①SW1=ON
- ②システムタイプ=Y
- ③外気温度<0°C

# AOO技術のご紹介

AOO (Autonomic architecture base Object-Oriented development technique) は、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したオープンな技術である。その後、AOOは、プロセス(AOO\_PRS)、プロダクトライン(AOO\_SPL)、見積り(AOO\_EST)、形式手法(AOO\_DSL)と手法を拡張して書籍の再版準備中。

AOOは、「整理」「定義」「変換」の3つの指針で構成される。

要求の発生源、目的、目的達成で得られる価値を定義する。要求は1機能1目的でカテゴリに階層的に整理して機能仕様を定義する。カテゴリで整理することでカテゴリ共通部とカテゴリ間インタフェースの定義により対象ドメインのフレームワーク構築を可能にする。

機能仕様の記述は、記法と表形式により形式的に記述することで機能ブロックのライブラリ化、可変性の表形式モデルにより増大する要求に対応する。

分析で定義したモデルと設計・実装・テストの双方向変換ルールを用いトレーサビリティも確立を可能にすることでモデルベース開発を実現する。

AOOは、開発上の課題に基づく解決技術であり開発上で発生する課題の真因を分析して未然防止するためのフレーム(ドキュメントフレーム/ソースコードフレーム)とフレームに実装するための開発プロセスを定義することで組織的な課題抑制を可能にする。