



# 課題認識とワークショップ2009の狙い

～ 組込み系ソフトウェア開発のキモは何か ～

2009年10月20日

社団法人 電子情報技術産業協会  
ソフトウェア事業基盤専門委員会

副委員長 春名 修介

# 組込みソフト開発の閉塞感

- 全体設計が不十分なまま、個人のスキル依存の開発
- ソフト開発の現場には閉塞感が高まっている

## 組込みソフト技術者の現場の声

上流工程の曖昧さを  
下流工程(テスト)で対応  
＜設計完成度の不足＞

企画部門やユーザ部門からの  
機能や性能のすべての要求を  
受け入れざるを得ない現状  
＜コード以外の上位視点での  
情報発信不足＞

再利用性のような生産性に関わる  
非機能要件が設計時に考慮されない  
1回目の開発は何とかできるが、  
機種展開の効率が悪く、息切れ  
＜エンジニアリングの欠如＞

事前に課題を詰めない設計文化のため、  
構造やインターフェースがコーディングの過  
程で決まり、製品毎の個別のソフトウェア  
構造になる  
＜エンジニアリングの欠如＞

＜組込み系開発スピードアップワークショップ2008＞より 主催:JEITA (社)電子情報技術産業協会

# 浮かび上がった課題

## ■ 要求分析

- 要求仕様が決まらない/決まるのが遅い
- 要求仕様が度々変更される

## ■ 設計

- アーキテクチャ設計ができていない
- 設計の質が悪い
- 実装主体の設計になってしまっている
- • • • •

## ■ 人材面

- 属人的な開発
- スキル不足
- 組織的な人材育成不足

表面上は、  
当たり前のことか  
できていないかの  
ように見える

組込みソフトウェア特有の特性/課題  
があり、既存手法  
の適応が難しいの  
ではないか？

2008年度実施の「開発スピードアップワークショップ」, 「アンケート調査」より

# 今年度の活動

- 開発者目線での開発スピードアップの阻害要因から、組み込み系ソフトウェア開発の特性「キモ」を明らかにする



やるべきことを  
やっくらん

## 組み込み系ソフトウェア開発のキモは何か

「組み合わせ」と「擦り合せ」  
「トップダウン」と「ボトムアップ」  
ソフトウェア開発 or 製品開発  
非機能要件への対応  
組み込み系での上流工程のあり方  
- 戦略／企画  
- 要求分析  
- アーキテクチャ設計  
...

開発実態に踏  
み込んだ議論

できていない理由  
があるはず



# エンタプライズ系と組込み系の違い例：設計視点①

## ■ エンタプライズ系： <機能の網羅性（静的）>

– 機能漏れの排除に注力

» 例：データベース設計 など



機能： 多  
時間概念： 少

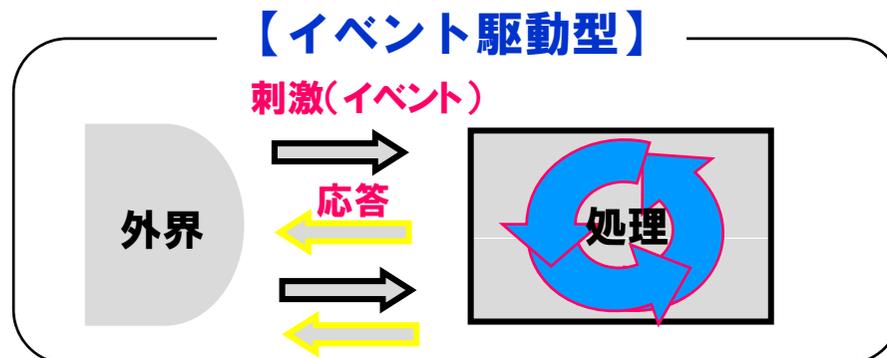
有限

モデル化  
し易い

## ■ 組込み系： <動作の網羅性（動的）>

– 破綻無く動作し続けることに注力

» 例：並行動作，応答性，例外時の対応，資源制約 など



機能： 少  
時間概念： 多

無限

モデル化  
し難い

## エンタプライズ系と組込み系の違い例：設計視点②

- 組込み系ソフトウェアは、本質的に「ボトムアップ」・「擦り合わせ」の方向に流れ易い

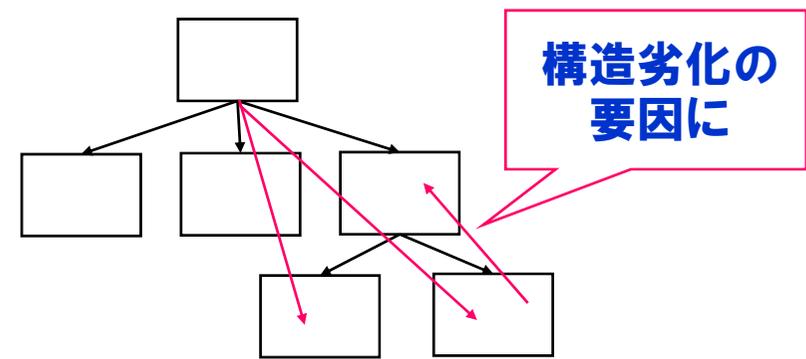
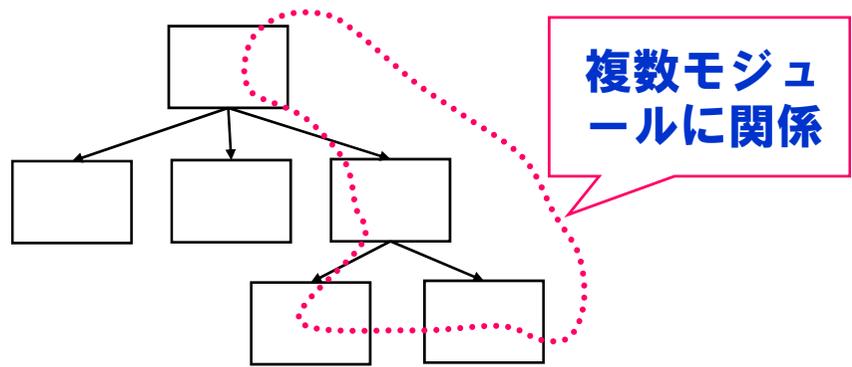
### ① ハードウェアとの同時開発

### ② 特有の**非機能要件**

- 応答性, リアルタイム制約
- 例外処理 (割り込み, 中断など)
- 低消費電力
- 高信頼性・安定性

組込み技術者はここで苦勞している

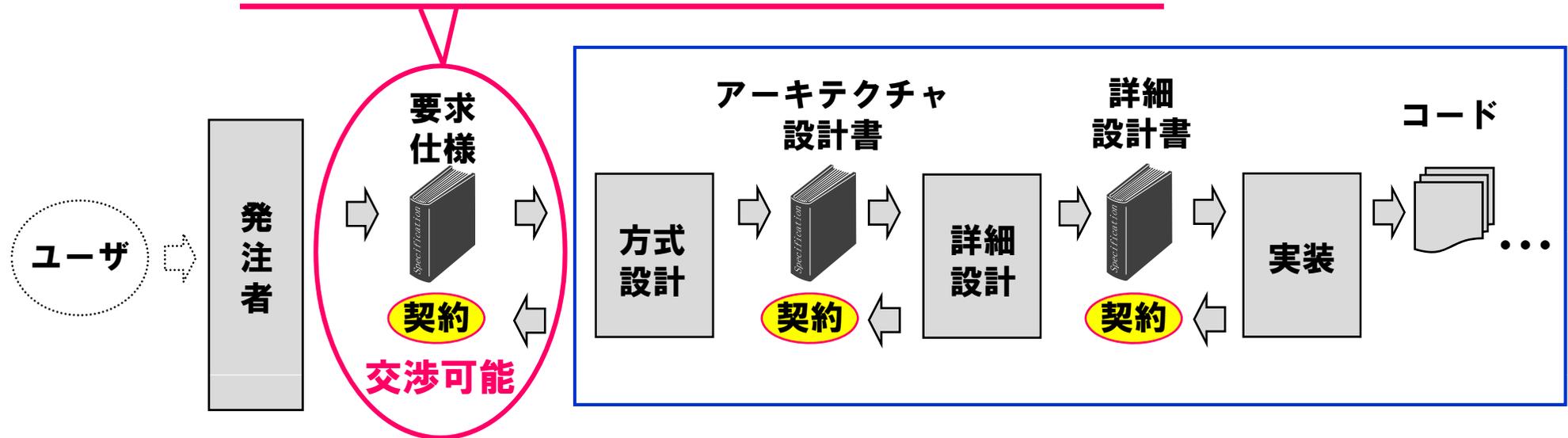
一つのモジュールに閉じない  
全体的な設計項目  
(横断的関心事)



# エンタプライズ系と組込み系の違い例：ビジネスモデル①

## ■ エンタプライズ系：契約モデル

- 開発の起点は，発注者と合意する要求仕様に集約

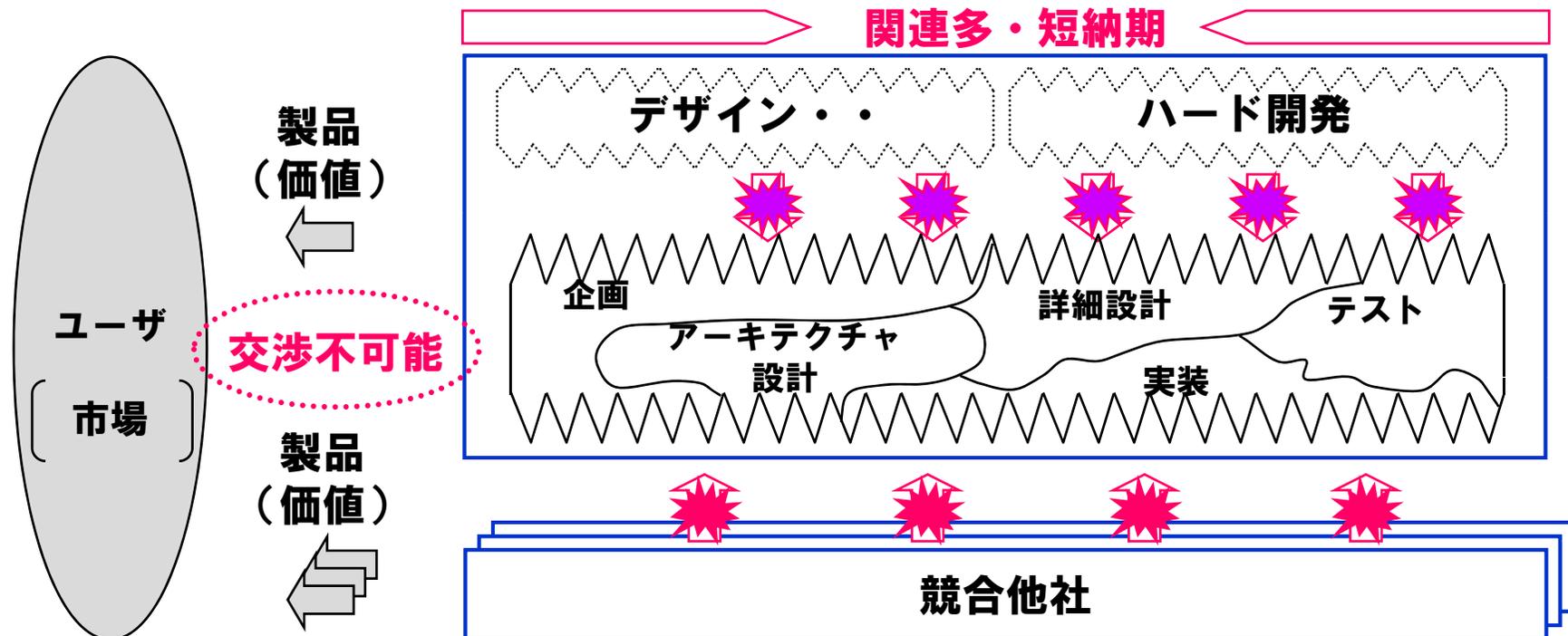


- 発注者と受注者の間で，契約を明確にする交渉が可能
- 同様の契約モデルを開発に持ち込むことが可能

## エンタプライズ系と組込み系の違い例：ビジネスモデル②

### ■ 組込み系：価値提供モデル（競争モデル）

- 製品開発の一部であり，本質的に外乱が不可避な側面がある
  - » 開発の起点は，市場予測，他社との競争・市場の反応により変化
  - » 開発納期も市場が決める（交渉不可能）
  - » 自社の開発工程の都合を度外視しても対応せざるを得ない



# 小規模時代の擦り合わせ型開発

- 確実に動作するソフト・ハードのセットをベースに周辺を積み重ね
  - ハードウェアとの同時開発の特性が組み込みソフト開発のマネジメントスタイルを規定
  - 非力なハードでの性能確保など組み込み特有の非機能要件 など
- 高品質開発が可能な反面、悪い方向に進むと,
  - 全体が見えない開発・属人的な開発に陥る
  - やって見なければ分からない、動いている所は、絶対触りたくないという開発風潮

➤ 小規模時は効率的、規模拡大と共に破綻



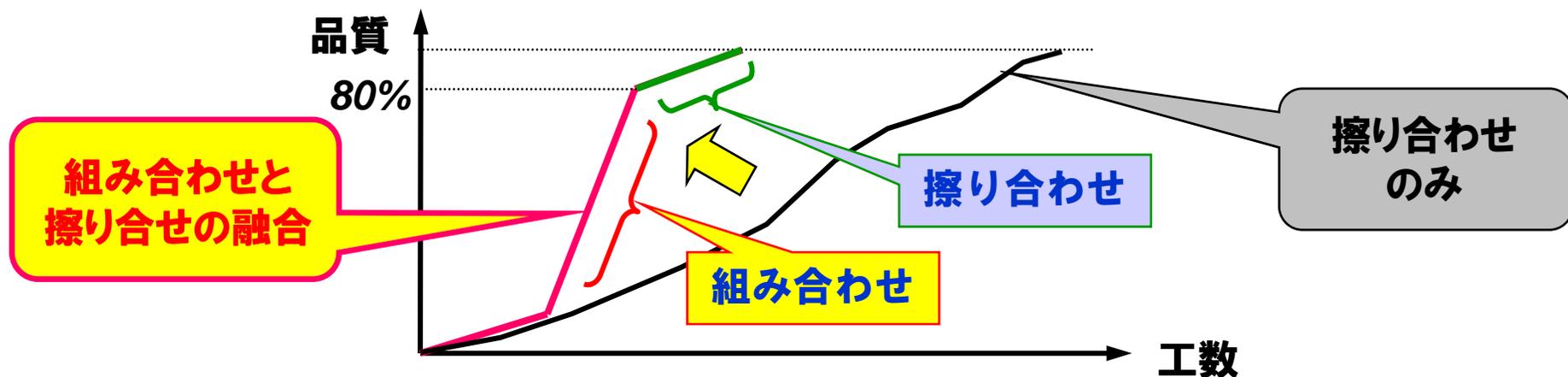
# 組込み系ソフトウェア開発の目指す姿は？

- **擦り合わせによる高品質開発**が日本の競争力の源泉
- しかし、全体が見えない時点からの「**アドホックな擦り合わせ**」では、**大規模化・短納期化**などに対応できない



## 《 組み合わせと擦り合わせの融合 》

- ・ 8割までは、**組み合わせ**(設計・アーキテクチャ)の補完により、すぐに
- ・ 残りの2割を、**擦り合わせ**で → 組込み特有の非機能要件対応など



## ワークショップの狙い

---

- **組込みソフトウェア開発者は、今やっていることに自信を持つべき！**
- **しかし、組込みソフトウェア開発の「キモ」が形式化され、認識されていないと、アドホックな開発に陥る**  
**例：**
  - **何が組み合わせ「モジュラー」で、何が擦り合わせ「インテグラル」か**
  - **エンタープライズ系との違いは**
  - **組込み系特有の非機能要件に対応する設計手法 などなど**
- **組込み系ソフトウェア開発の目指す姿(何を守り、何を変えるか など)を皆様と議論させて頂きたい**

**ご清聴ありがとうございました**