

組み系ソフトウェア開発をスピードアップ！

～ 大規模化，複雑化，短納期化，多機種化する
組み系ソフトウェア開発の改革に向けて～
(JEITA活動報告)

- 聞け!! 開発現場の声 -

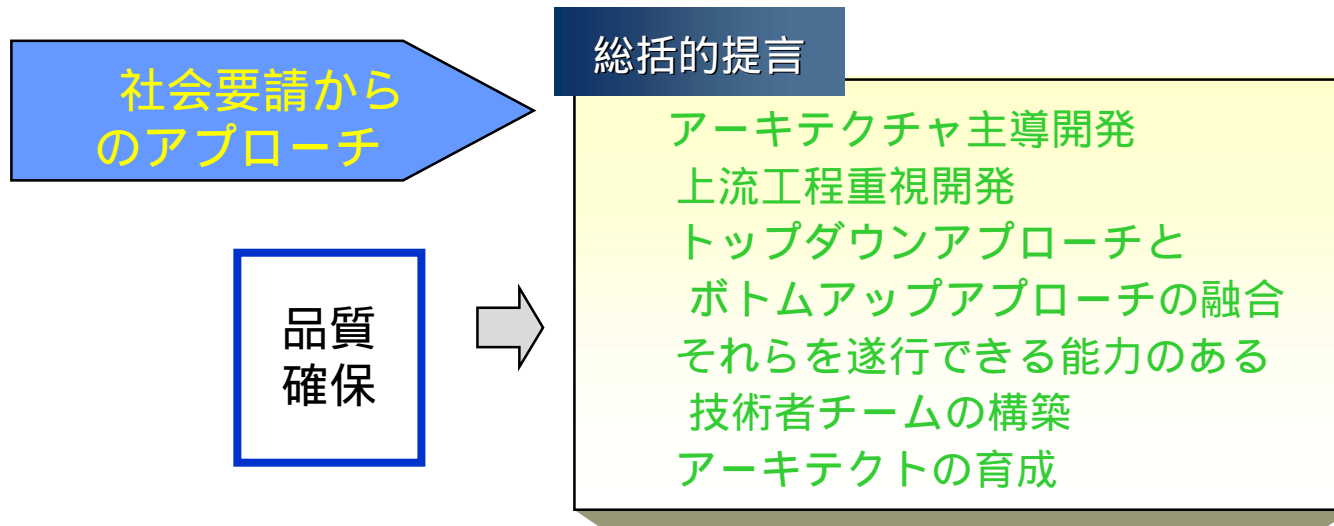
2008年10月2日

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
パナソニック株式会社
春名 修介

今年度の活動の狙い (1)

2007年度までの活動

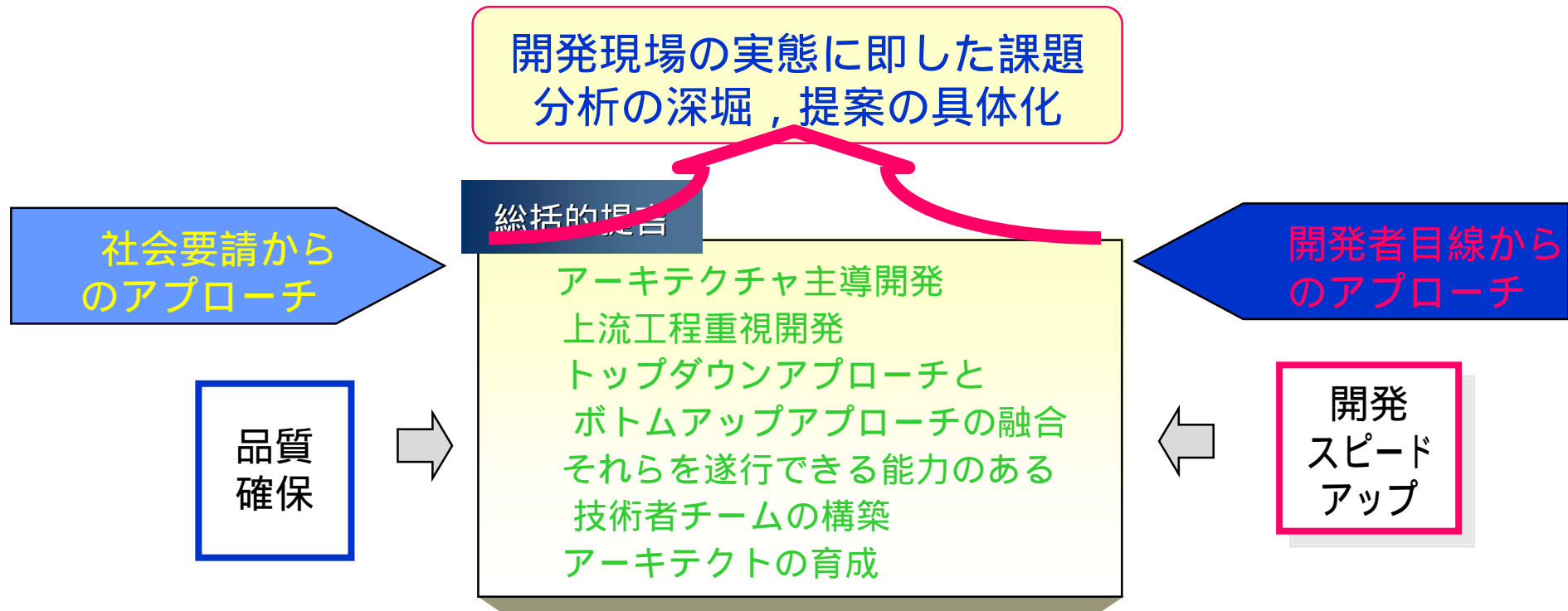
- 組込みソフトの大規模化，複雑化，短納期化，多機種化への対処施策を，社会要請ともいえる“品質確保”のアプローチから分析，総括的提言を実施．



今年度の活動の狙い (2)

2008年度活動

- 4つの波への対処施策を、開発者目線ともいえる“開発スピードアップ”の新視点からアプローチし、2007年度までの総括的提言の検証と新視点から深堀する。



社会要請からのアプローチ

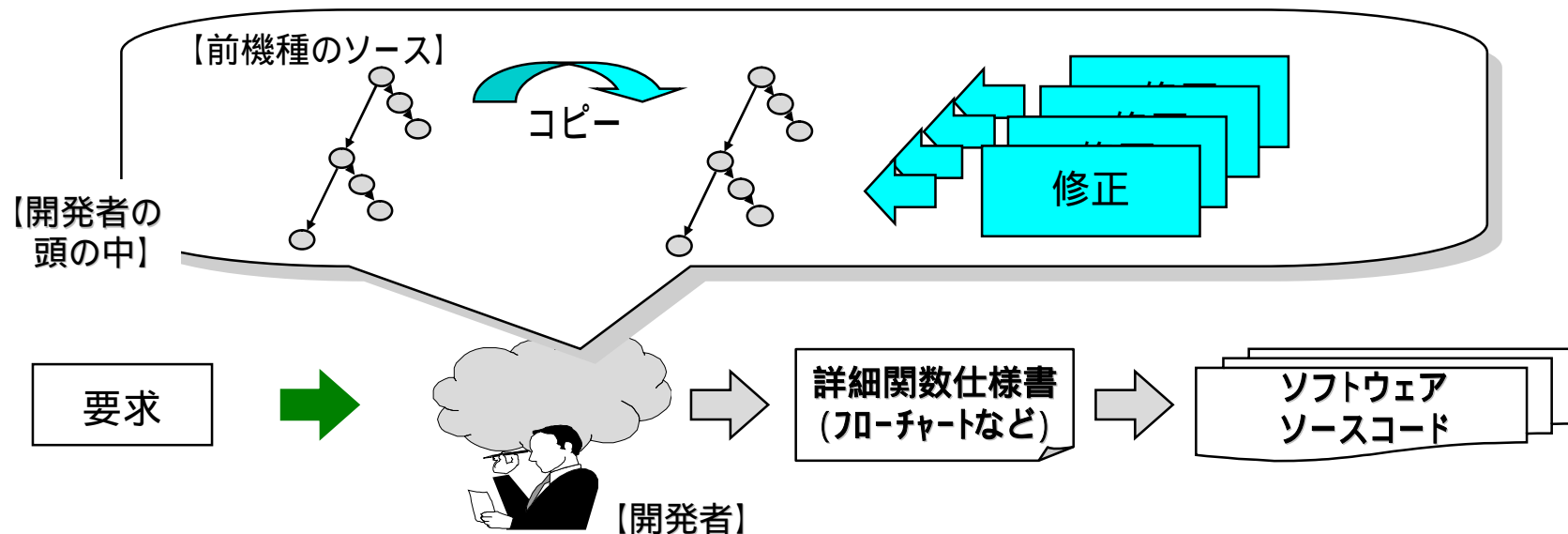
大規模化，複雑化，短納期化，多機種化下
における品質確保施策

コード中心の開発

■ コードを作りながら全体構造を決めていく開発スタイル

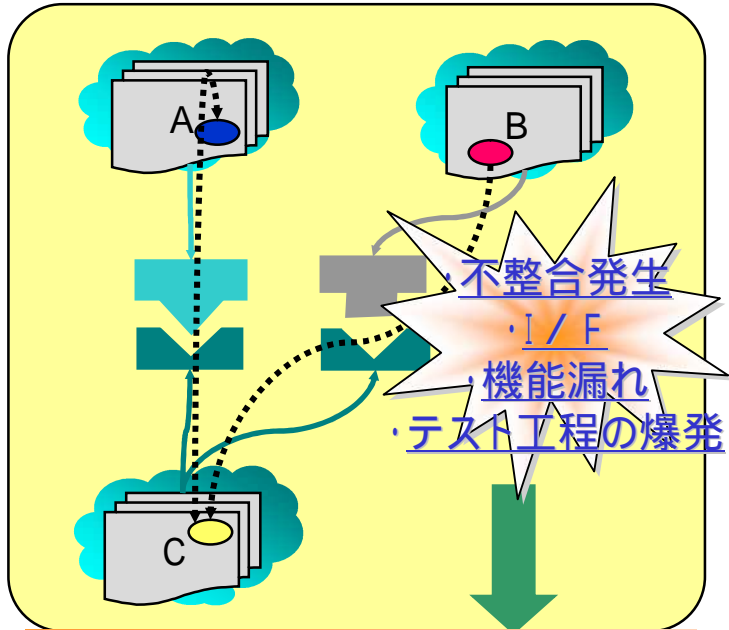
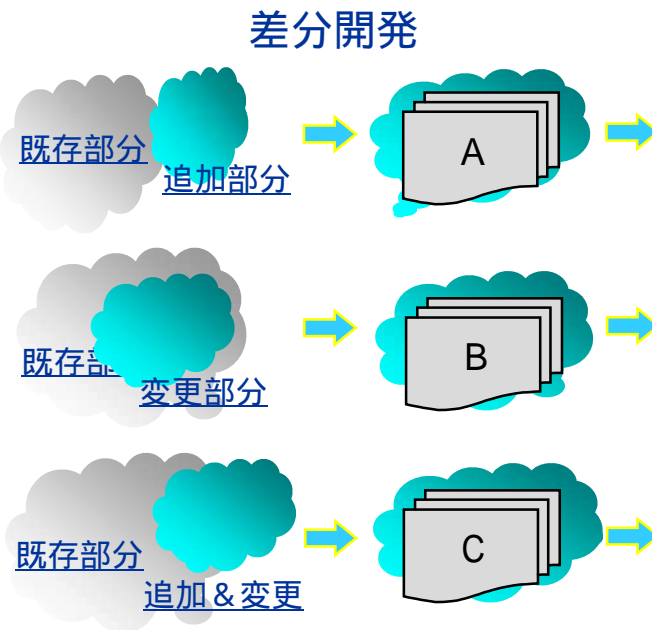
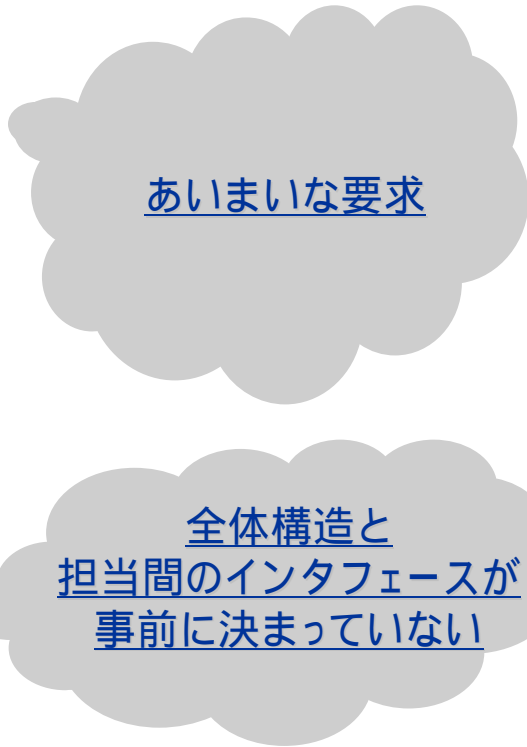
以前の機種のソフトウェアをコピーし、
必要な部分のみを修正・追加（差分開発，コピー＆ペースト開発）
小規模時代の開発のなごり，短納期開発のプレッシャ

属人的な開発：要求からコードへのブレイクダウン過程が開発者の頭の中
場当たりの修正によるコードの複雑化
開発する機種数の増加，担当者の変更により，**急激に開発効率が低下**



全体構造が把握されないまま進行する開発

- 分担だけは決まっているが、全体把握ができていない



分担間の仕様調整に
 時間がかかる (n 対 m)
 曖昧な仕様を基に、分担開発
 が進行 (見切り発車)

システムテスト工程で
不整合多発!!

有効な施策と現実のギャップ

- 開発の大規模化，多機種開発への対処として，本来ならば，**再利用が有効なはずだが**，**下流工程での擦り合わせ開発が横行**

既存ソフトを流用，**上手く行く筈・・・でも動かない！**
原因を特定しようとドキュメントを調べるけど肝心なことが書いていない，判らない！

では，かつての開発者に聞いてみよう！
残念ながら，その開発者はもう居ない！

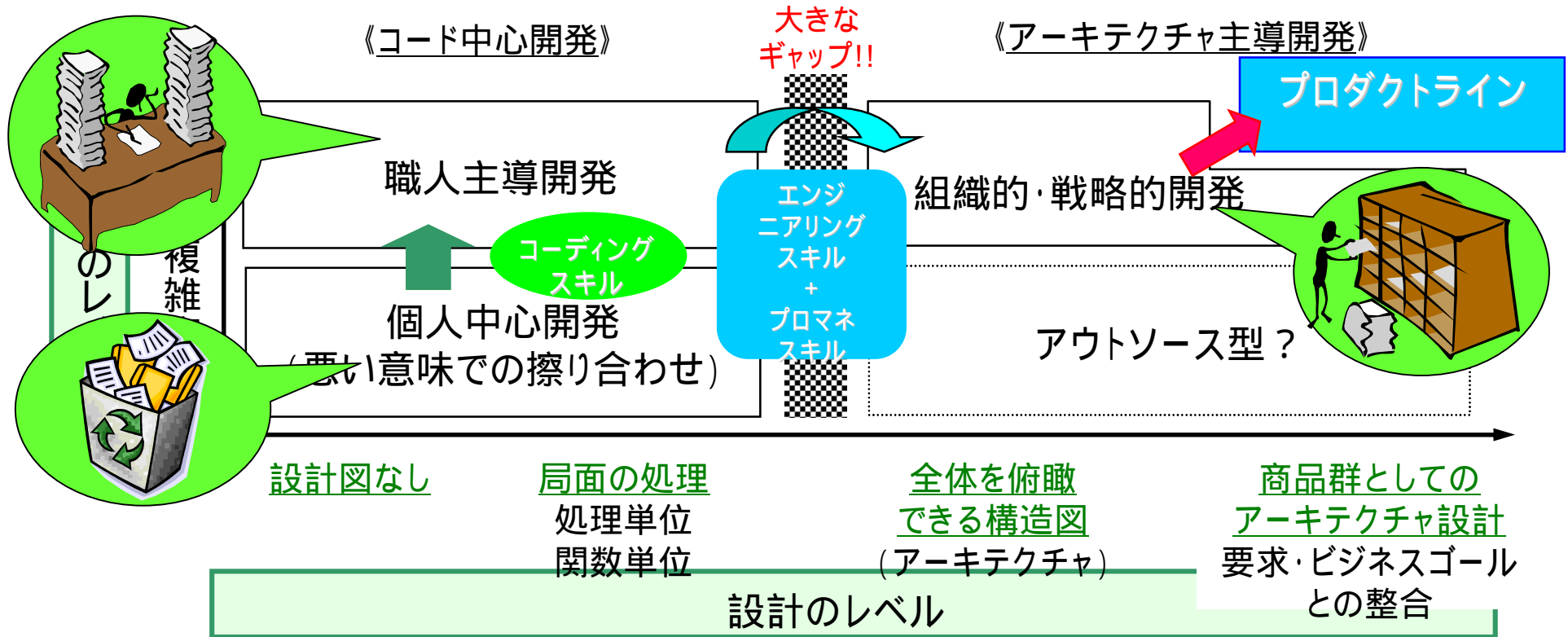
再利用は昔から叫ばれているが，現場に定着した例は？
キーマンが変れば，元の木阿弥・・・

作ってからの再利用は効果が薄い
再利用を考えた**戦略的な開発への発想転換が必要**

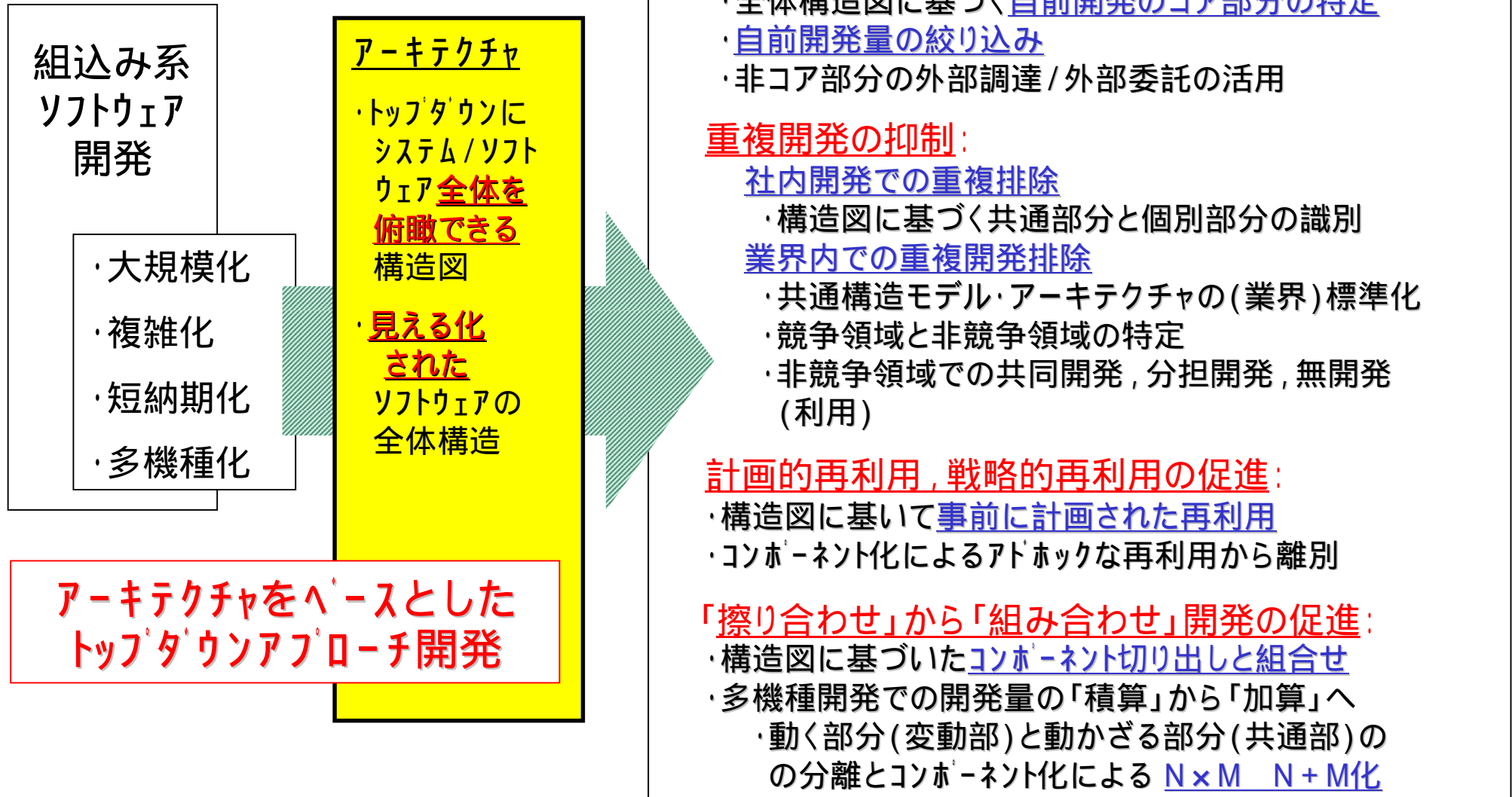
現状からの脱出：現状認識の重要性

- 開発レベルの認識とレベルに合った処方箋が必要

個人中心開発の状況から、いきなり戦略的な再利用（プロダクトライン）には行けない
 まだまだ、コード中心開発の現場が多いのではないか？



アーキテクチャ主導開発（トップダウンアプローチ）

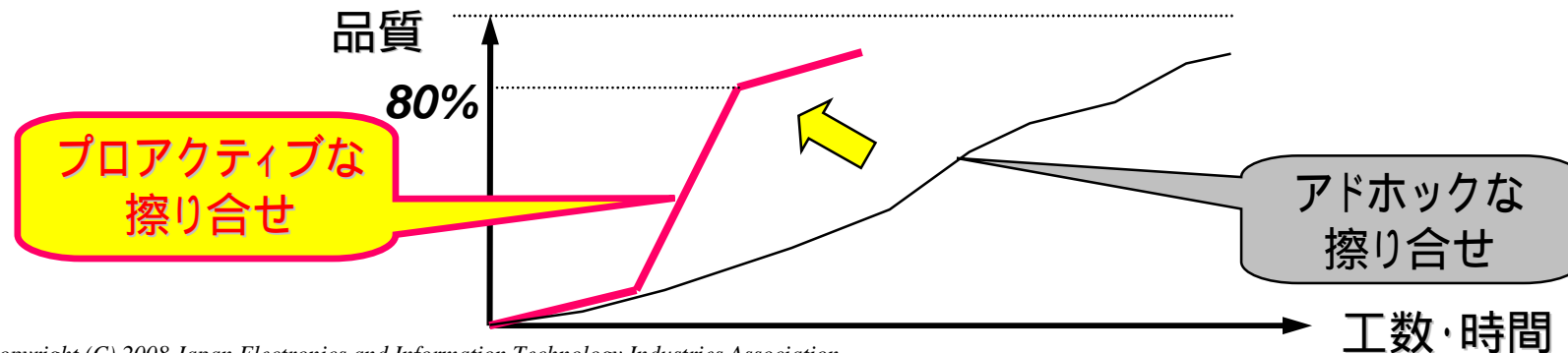


トップダウンアプローチとボトムアップアプローチの融合

- **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

アドホックな擦り合わせから、プロアクティブな擦り合わせへ

- ・ 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
- 産官学一体となったアーキテクト育成への取組み
- 構造を作るアーキテクトと取って替えて構造を崩せるスーパープログラマ、その両者が共存・補完する技術者チームの構築



開発者目線からのアプローチ

組込み系ソフトウェアの国際競争力のアップ

大規模化, 複雑化, 短納期化, 多機種化下
における開発スピードアップ

開発スピードアップとは？

本専門委員会での議論では・・・

- ・ 日本製品は機能を作りすぎ(機能のテンコ盛り)？ (なぜそうなるのか)
- ・ 何を作るかが明確ではないのでは？ (超上流の問題)
- ・ 各社で作らなくても良いのでは？ (非競争領域と競争領域の整理が必要では)
- ・ スピードアップのためには”何をやめればよいか”では？ (発想・視点の転換)
- ・ スピードアップと品質とのバランス/トレードオフをどう考える？ (製品の性格など)

生産性向上の枠を超えて、変化する市場や顧客の要求に対して、
如何に早く変化し対応できるか。いかに開発するか(how to develop)だけでなく、
何を開発すべきか(what to develop)の重要性が増し、両者の統合にまで及ぶ問題

開発スピード = (開発能力・工程能力) / (開発規模・難易度)

例えば、機能要求がてんこ盛りの要求仕様を思い切ってグローバル要求仕様として
大きく共通化して「難易度」を低減するとか

端的に言うと、以下の4つ：

- 戦略的なソフトウェア開発
- 既存資産の再利用(作らないで済みます)
- 自動化
- 無駄の排除

組込み系開発スピードアップ・ワークショップ2008

組込み系ソフトウェア開発をスピードアップ！

～大規模化、複雑化、短納期化、多機種化する
組込み系ソフトウェア開発の改革に向けて～

2008年8月27日(水) 13:30～17:55
@ベルサール神保町

全員参加型100人ワークショップ：

開発スピードアップを阻害する要因
開発スピードアップを促進する要因



<http://home.jeita.or.jp/is/committee/software/080827/index.html> 《10月10日公開予定》

100人ワークショップ: 事前アンケートから

スピードアップすることが、関係者(特に、設計、実装、テストの担当者)の利益になる体制と仕組みの確立が重要。そうでないと、これまでのように、担当者の過重労働の犠牲の上に、スピードアップが実現するという構図になる。そもそもスピードアップ以前にやることもっと沢山あると思うのだが。

スピードアップのために品質が犠牲になることは本末転倒であり、品質を確保した上でいかにスピードアップできるかが鍵であるが、これが本課題を難しくしている。

スピードアップ = 無駄の排除

- ・低コスト、高品質を維持してスピードアップは可能か。スピードに限界はないのか。
スピードが要求される今の状況そのものに問題はないのか。
- ・手の抜きどころ、見切りポイント、過剰ではない中庸な落としどころはあるのか。
それを決めるためにはビジネス戦略が重要か。
- ・個人のスキルアップは解決策のひとつと思うが、スキル格差は個人の問題か。
- ・管理工数の増大はマネージャにも開発者にも負担になり、モチベーションの低下を招いている。

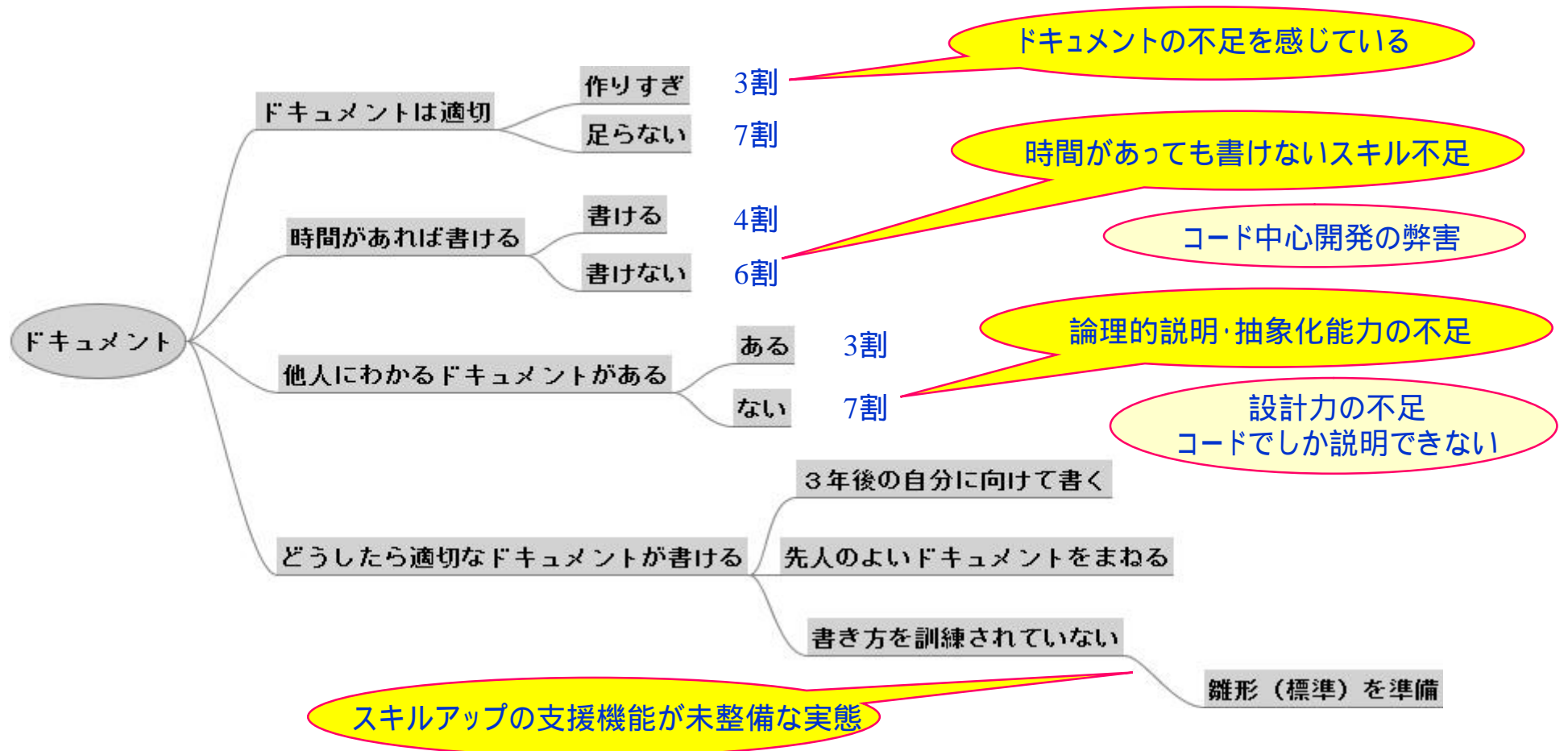
スピードアップして開発期間の短期化が実現したとして、その余裕が、即、次のプロジェクトへのリソース投入となってしまうと、エンジニアのモチベーションも下がり、スキルアップに繋がらないケースが多々見られる。その余裕を、スキルアップや新しい技術の研究に使えるような環境構築が必要だが、マネジメント層の理解がなかなか得にくいのが課題

現場の声：開発スピードアップの阻害要因

- 機能中心の設計となっていて、再利用性のような非機能要件が設計時に考慮されないため、1回目の開発は何とかできるが、機
 悪く、息切れしてしまっている。
 アーキテクチャ設計
- エンジニアのプライドに掛けて、企画部門やユーザ部門が
 や性能要求のすべてを受け入れる設計や実装を行っている。
 ビジネス戦略との整合
- そもそも事前に課題を詰めるという設計の文化がない。構造やインタフ
 ェースがコーディングの過程で決まってくるため、製品
 トウェア構造となっている。
 アーキテクチャ設計
- ドキュメントを作成する文化がない、ガイドラインがなく法
 ーマットや手順がないなどのために、意思疎通が悪く、機
 、不良作り込み、作業の手戻りが発生している。
 ドキュメント
 属人性の排除
- 会議スコープが曖昧なため、どのくらいの頻度でどの
 告するのか曖昧となり、過剰報告・過剰管理につながっている。
 プロマネ、
 支援プロセス
- プロジェクトの初期段階でリスクが特定できず、リスク
 対応することになり、手戻りが発生している。
 戦略・組織
 マネージメント
- 上流工程の曖昧さを下流工程（テスト）で対応せざるを得ない。
 上流工程重視

具体的な現象面から：ドキュメント

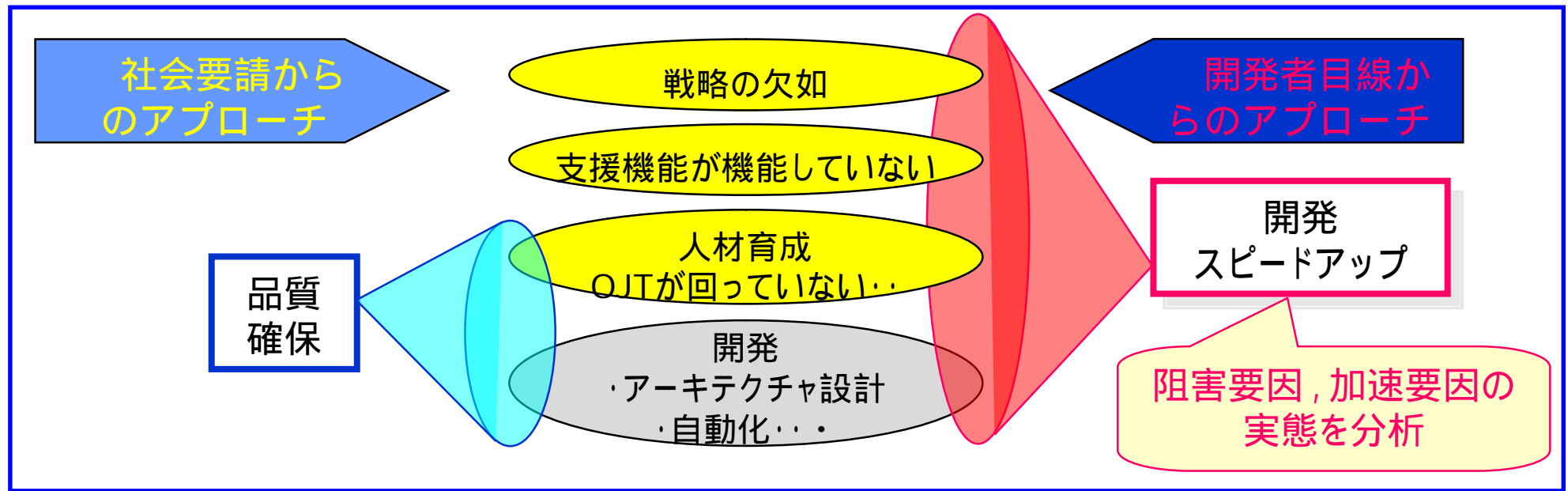
- ドキュメントの不備(現象面)からの意見集約結果
 - ドキュメント一つを取っても根が深い様々な課題が現場には集積



これからの活動

■ 開発スピードアップ100人ワークショップでの議論

- 開発(特に,設計)に関する課題は,両視点から見ても同じ傾向
- 開発以外の戦略・支援機能・人材育成・技術者の考え方などの課題も顕在化



➡ 実態アンケート, 海外調査などにより, 更に課題を掘下げ

開発スピードアップの障害要因と促進要因の実態を分析,
それを通して, 現場感覚を反映した施策提言とその具体化

ご清聴ありがとうございました