

JEITA組込みソフトウェア・ワークショップ2010

FUJITSU

shaping tomorrow with you

富士通の組込みソフト開発技術者から見た、

擦り合わせ開発の強み

富士通九州ネットワークテクノロジーズ株式会社

撰 暁久(えらみ あきひさ)

erami@jp.fujitsu.com

2010年10月29日

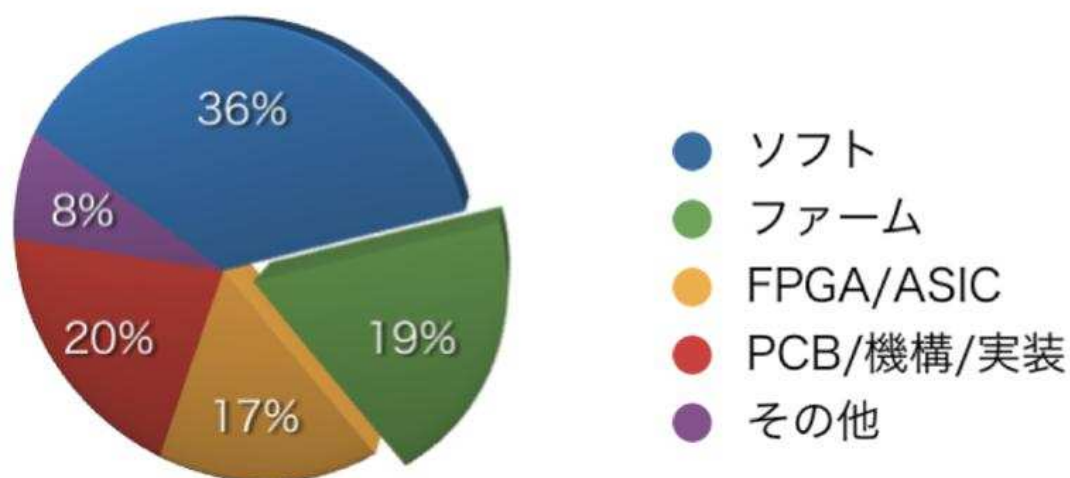
発表内容

- 会社紹介
- 組込みソフト開発上の課題と取り組み紹介
- テーマ選定の背景
- 組込みソフト開発の特殊性と擦り合わせ要素
- 英・米と日本での開発事例と、組織文化比較
- 組織文化の特徴と擦り合わせ易さ考察
- まとめ

■ 社名: 富士通九州ネットワークテクノロジーズ株式会社

- 本社: 福岡市早良区百道浜2-2-1 富士通九州R&Dセンター
- 設立: 1980年代設立の九州のR&D子会社を再編し、2005年7月に設立。
富士通株100%出資のエンジニア集団!
- 事業内容: ネットワークソリューション、通信機器開発
- 従業員: 824名(2010年4月1日現在)

■ 組み込みソフト(ファーム)開発比率は約2割(人員ベース)





システム・ソリューション

サービス・マネージメント・ソリューション ネットワーク・マネージメント・ソリューション



ネットワーク・テクノロジー

IP系システム開発

- ・IPネットワーク技術
- ・IPTランスポート技術

フォトニック系システム開発

- ・有線系アクセス技術
- ・WDM技術

モバイル系システム開発

- ・WCDMA技術
- ・FMC技術
- ・モバイル系アクセス技術

ネットワーク制御技術
移動無線ネットワーク技術

キー・テクノロジー

画像処理技術
ASIC/FPGA設計技術
アナログ設計技術

ネットワークアプリケーション技術
リアルタイムシステム技術
通信プロトコル技術



デジタル信号処理技術
IP電話制御技術
実装 / 構造設計技術
etc



応用技術

セキュリティ技術
生体認証技術

応用分野

モバイル端末
デジタル家電

- 開発費に占めるソフト開発コスト増大
 - ハード共通化・標準化 ソフト定義
 - 携帯電話では約80%がソフト開発費
- CPUのマルチコア化に伴い、性能のソフトアーキ依存度増大
 - 組み込みでは約27%()がマルチコア
- 3rd partyミドルウェアや、オフショア開発利用増大での、品質確保



出典：「2010年版組み込みソフトウェア産業実態調査」2010/06 経済産業省

- 開発費に占めるソフト開発コスト増大
 - ハード共通化・標準化 ソフト定義
 - 携帯電話では約80%がソフト開発費
 - CPUのマルチコア化に伴い、性能のソフトウェア依存度増大
 - 組込みでは約27%()がマルチコア
 - 3rd partyミドルウェアや、オフショア開発利用増大での、品質確保
- プロダクトライン開発により、製品当りの開発コスト低減
 - ✓ ➤ 性能を犠牲にせずに、マルチコア構成の違いを抽象化し、アプリ開発を省力化するフレームワーク開発
 - CMMIを通じて、開発プロセスの継続的改善

出典：「2010年版組込みソフトウェア産業実態調査」2010/06 経済産業省

マルチコア・フレームワーク開発の取組み

- プロダクトライン開発のコア資産として、タスク間通信手段を隠蔽するフレームワークを開発。

マルチコア・マルチプロセッサ環境で、コア割付を意識せずに高性能な通信を可能とするコードを自動生成

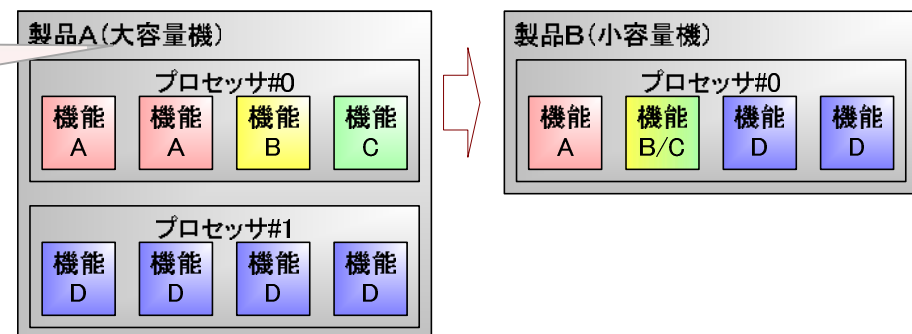
- 定義を変え、コード生成し直すことで、異なるプロセッサ数へ対応。

様々な処理容量のハードへの派生開発が容易

派生製品のファーム開発を
少ない工数で実現！



タスクのコア割付変更が容易に



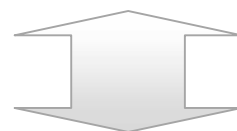
擦り合わせ型の製品開発としての強みの源泉が
組み込みソフト開発の何処にあるか？

日本と英・米の組織文化の対比を通して
マルチコア・フレームワーク開発事例を考察したい。

比較対象の開発事例

< 英・米での開発事例 >

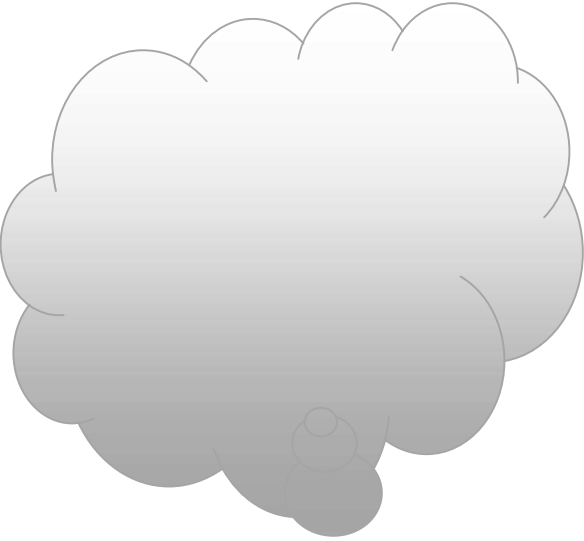
- 内容：通信機器のプロトコル試験用対向治具開発(英)
通信機器のハード診断機能の強化(米)
- 役割：開発者として英・米の開発拠点の組織へ入り、
支援業務として、結合試験や上記機能開発を実施



< 国内での開発事例 >

- 内容：通信機器を構成するマルチプロセッサ構成のボード上
の各ファームウェアを国内4拠点で分散開発
- 役割：プロダクトライン開発のコア資産部分として、
全プロセッサ共通で使うマルチコア・フレームワーク開発

- 性能達成するまで、飽くなきコード改良
 - 膨大なタスク相互の通信インタフェースは常に改善対象
 - インテグレーション時に性能不足が発覚、「擦り合わせ」の上、コード改良することが多い
- ハード・ソフトの最適なバランスへチューニング
 - 設計余裕を超えたリソースはムダ、量産前に削られる
 - ハード問題や外乱をファームで調整
- ハード依存性、ミドルウェア依存性の有無次第で可試験性、開発生産性に関き
 - 動作しない場合のベンダーとの対話は加速困難
 - 安易に同じ品質指標や開発プロセスを適用できない



擦り合わせ
要素
インタフェース、
性能

組織文化の比較

日本

集団主義、大部屋制

曖昧な責任分担、
総合力主義

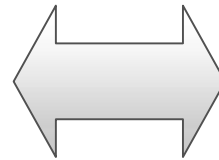
終身雇用・利害の共有
系列・高い対話密度

米(英)

個人主義、キュービクル

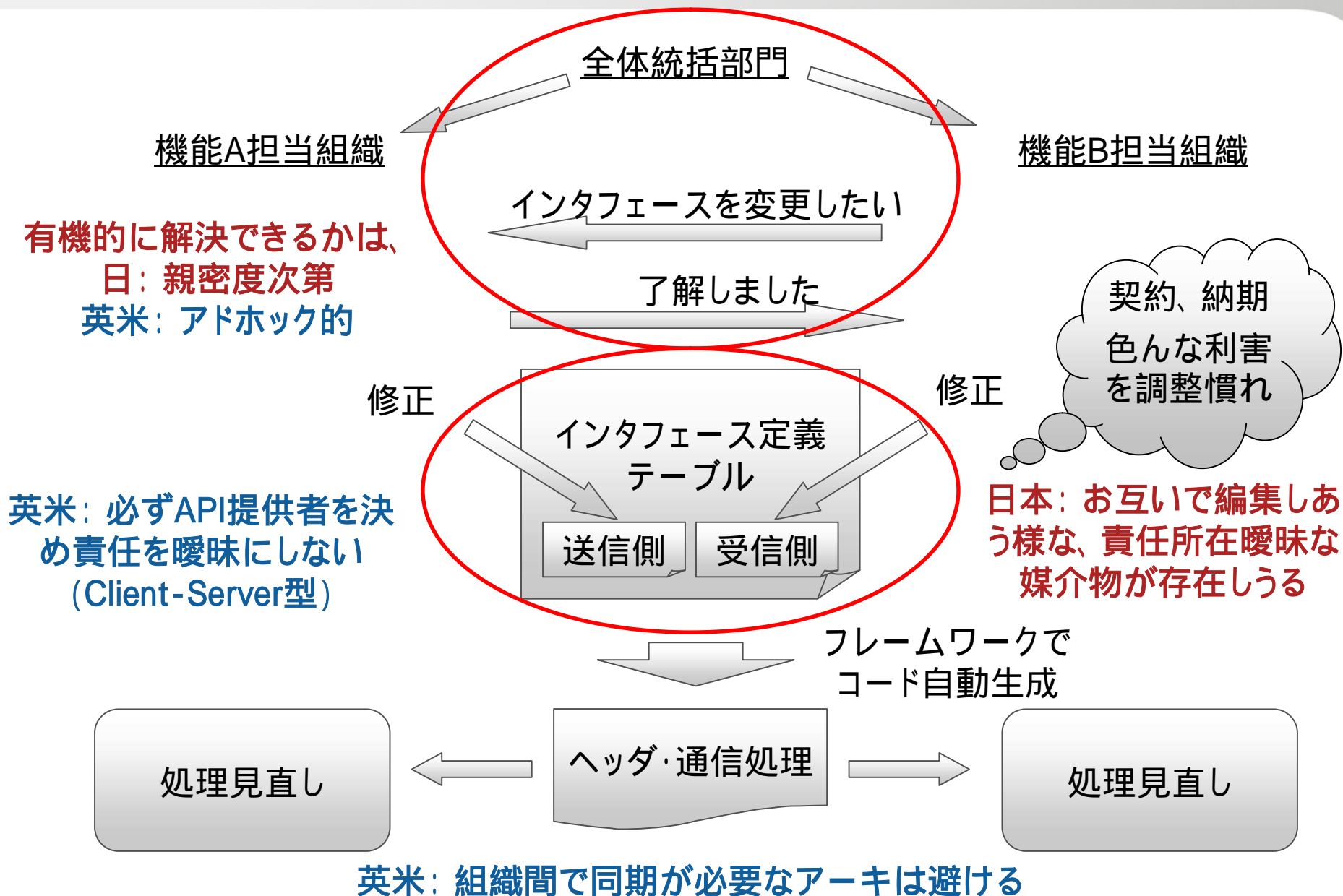
明確な責任分担、
専門力主義

流動的雇用・契約社会
オープンでアドホックな対話



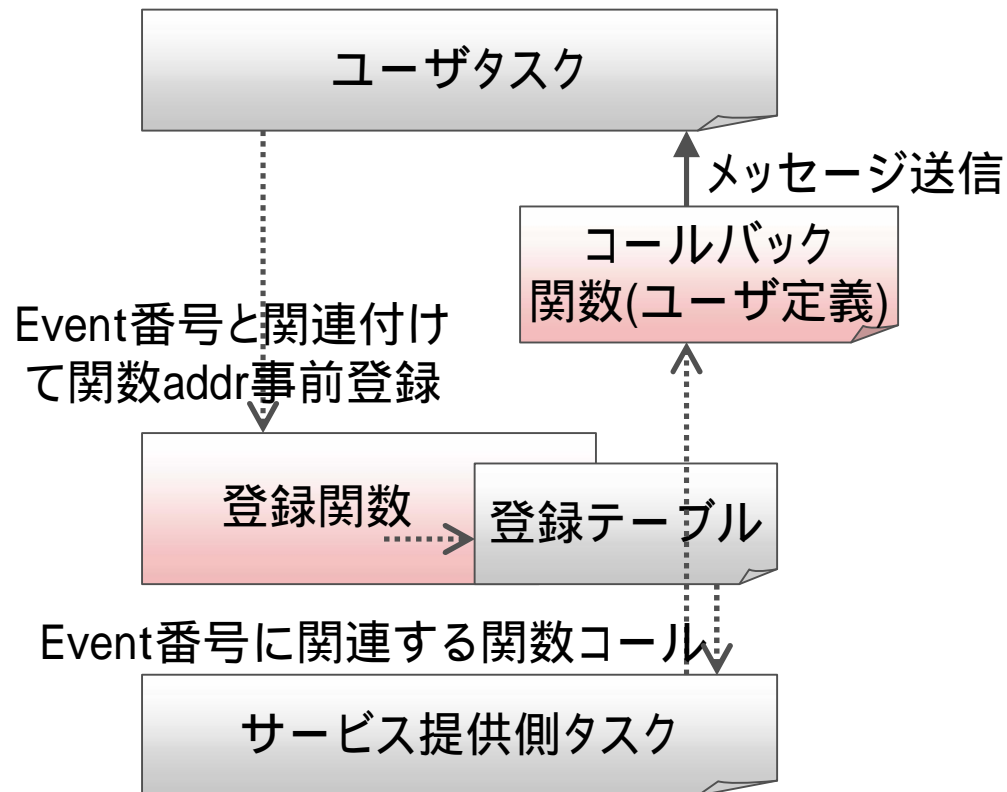
キュービクル：個人ごとに壁で仕切られたオフィス、アドホック：「特定の目的のための」というラテン語
出典：「情報技術と組織変化－情報共有モードの日米比較」2001/2/28 日本評論社 岡部曜子

1. インタフェース改善を擦り合わせる速さ



2. 疎なインタフェースを適切に (無用な擦り合わせ削減)

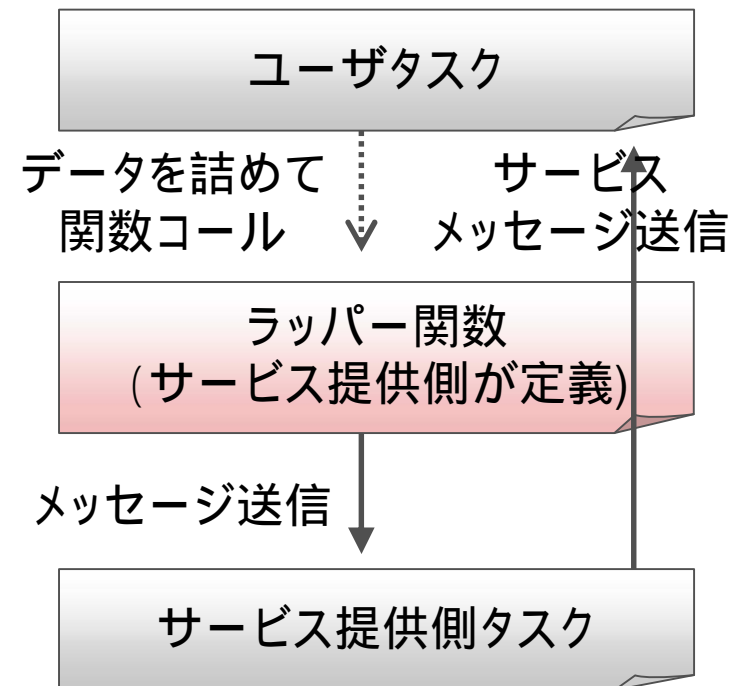
コールバック関数



コード独立性が高まるが可読性低

日本: 共通部ライブラリにのみ存在

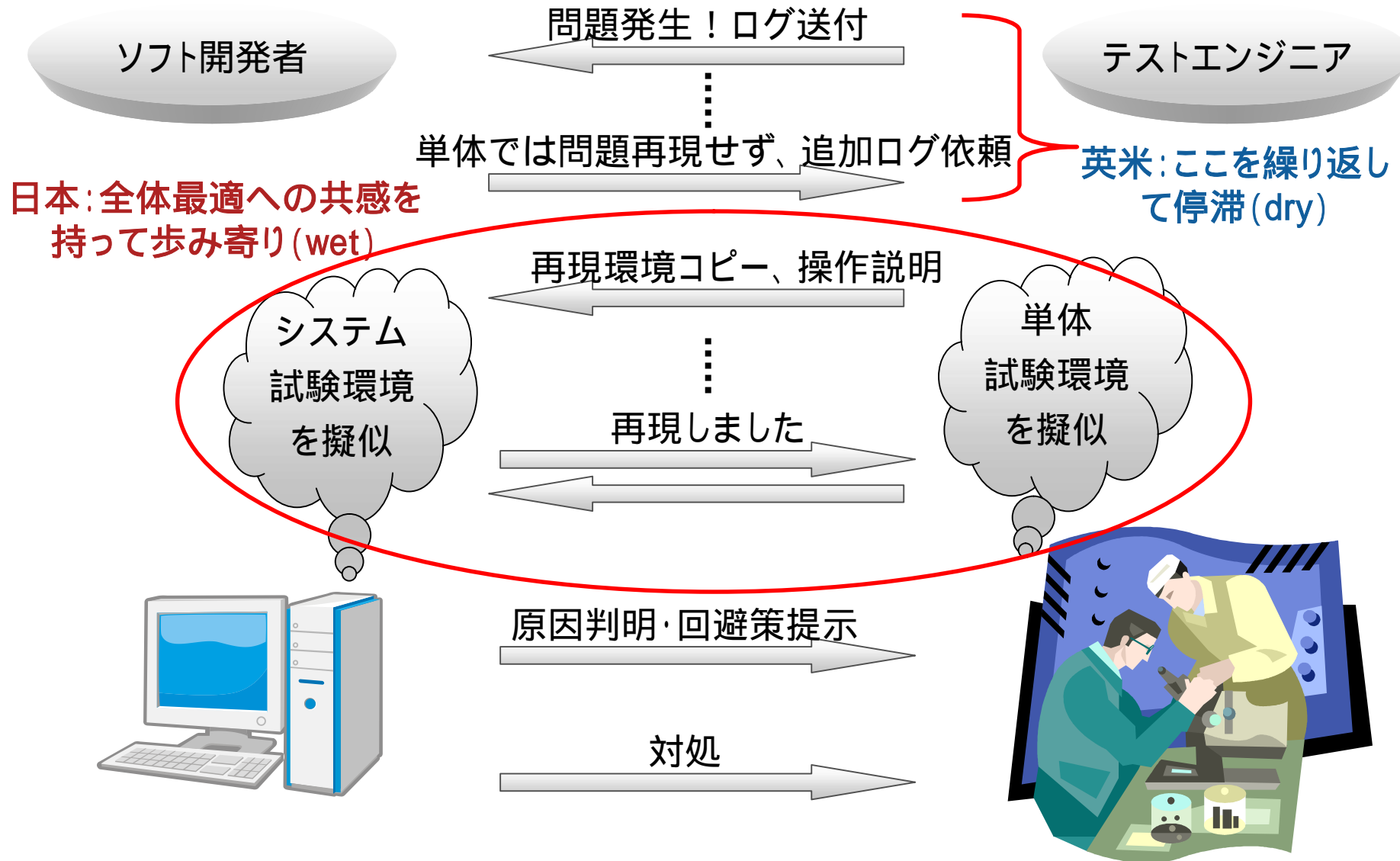
ラッパー関数



疎結合になるが、処理冗長

英・米: タスク担当境界毎に存在

3. 組織の垣根を越えて歩み寄り(組織同士擦り合わせ)



■ 強み

- 個々の責任分担が際立った疎結合なモジュールが作られ易い

■ 弱み

- インテグレーションで責任グレーな問題がいつまでもくすぶる
- 疎結合にしたい余り、部分最適な技巧に走り、全体のバランスを取りにくい

■ 擦り合わせ易さ考察

- アドホック的対話からの擦り合わせの速さは日本的組織より優れる
製品全体の使い勝手等、俯瞰して技術バランスを
指示できるアーキテクトが居れば全体最適に向かう



■ 強み

- 親密度が増した組織(間)では、全体最適を意識してエンジニアが有機的に対話して品質を高めていける(多くのベータテスターの目がある状態)
- 組織(間)で責任範囲を超えて歩み寄る事で早期に問題解決できる

■ 弱み

- 組織(間)が親密で無い時期、インタフェース決めや問題解析依頼を逐一リーダーにエスカレーションしがちで進捗が遅い

■ 擦り合わせ易さ考察

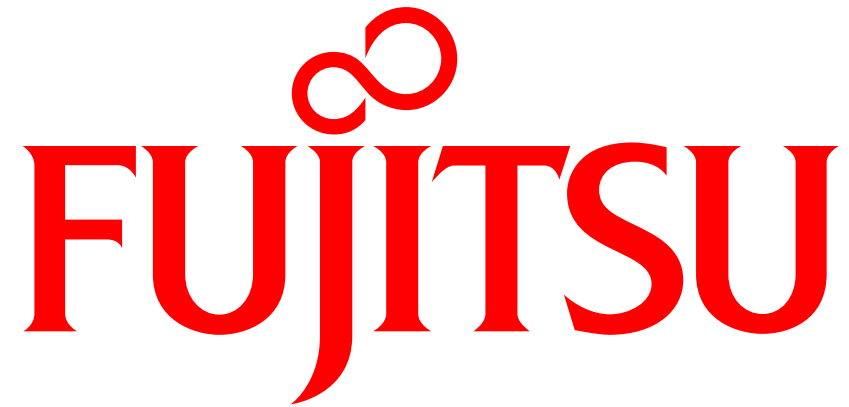
- 後工程になる程、全体仕様理解の深さと対話密度向上により、擦り合わせ易くなる！？
組織間の対話密度を早期に上げる工夫があると強くなる



擦り合わせ型の製品開発としての強みの源泉が
組み込みソフト開発の何処にあるか？



ハード/ソフト間、機能間の
インタフェース策定時・性能確認時にある。
全体最適を考え機動的に動ける組織(連携)が鍵。



shaping tomorrow with you