# デジタル家電ソフトウェア開発における信頼性確保の取り組み

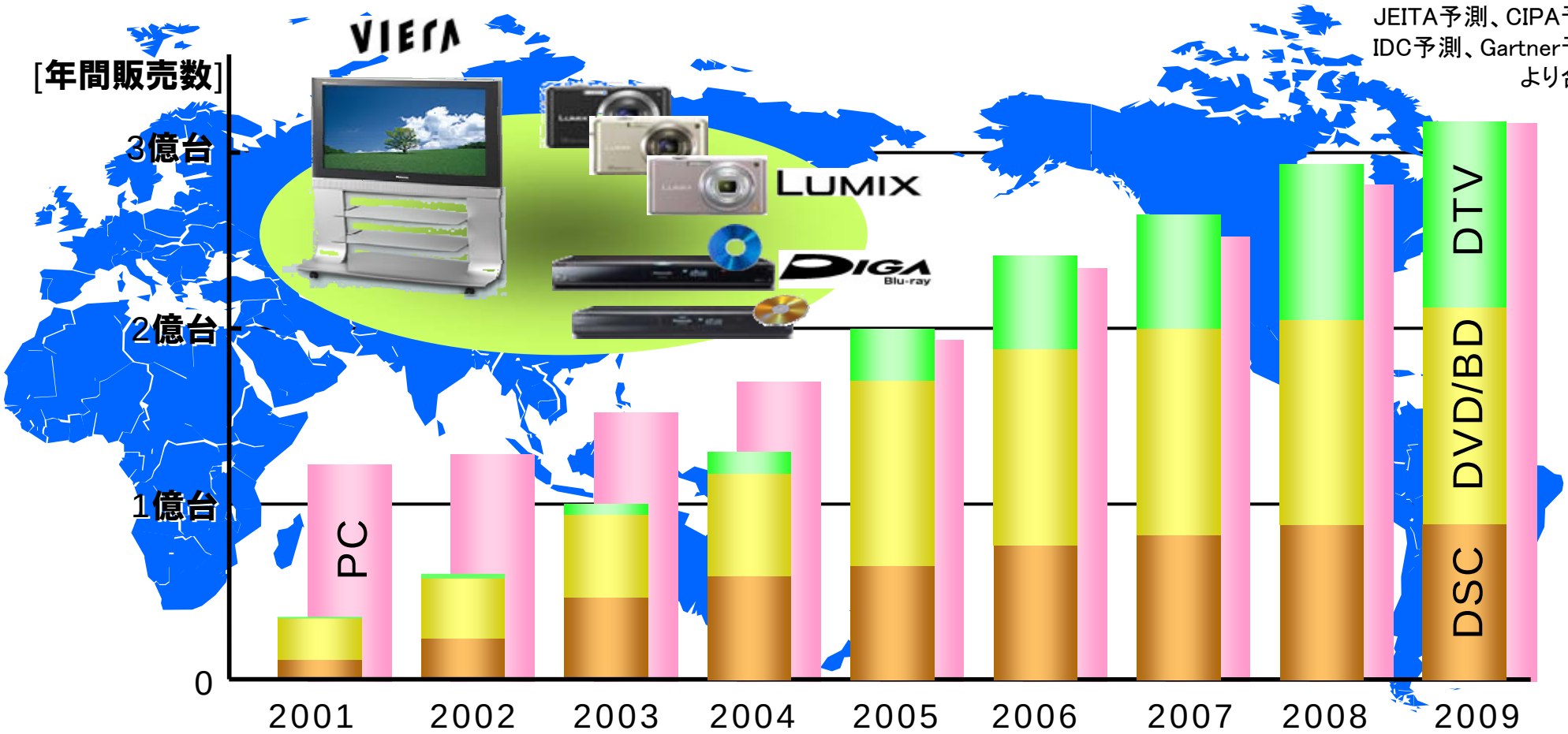
2008年3月12日

松下電器産業株式会社
システムエンジニアリングセンター
春名　修介

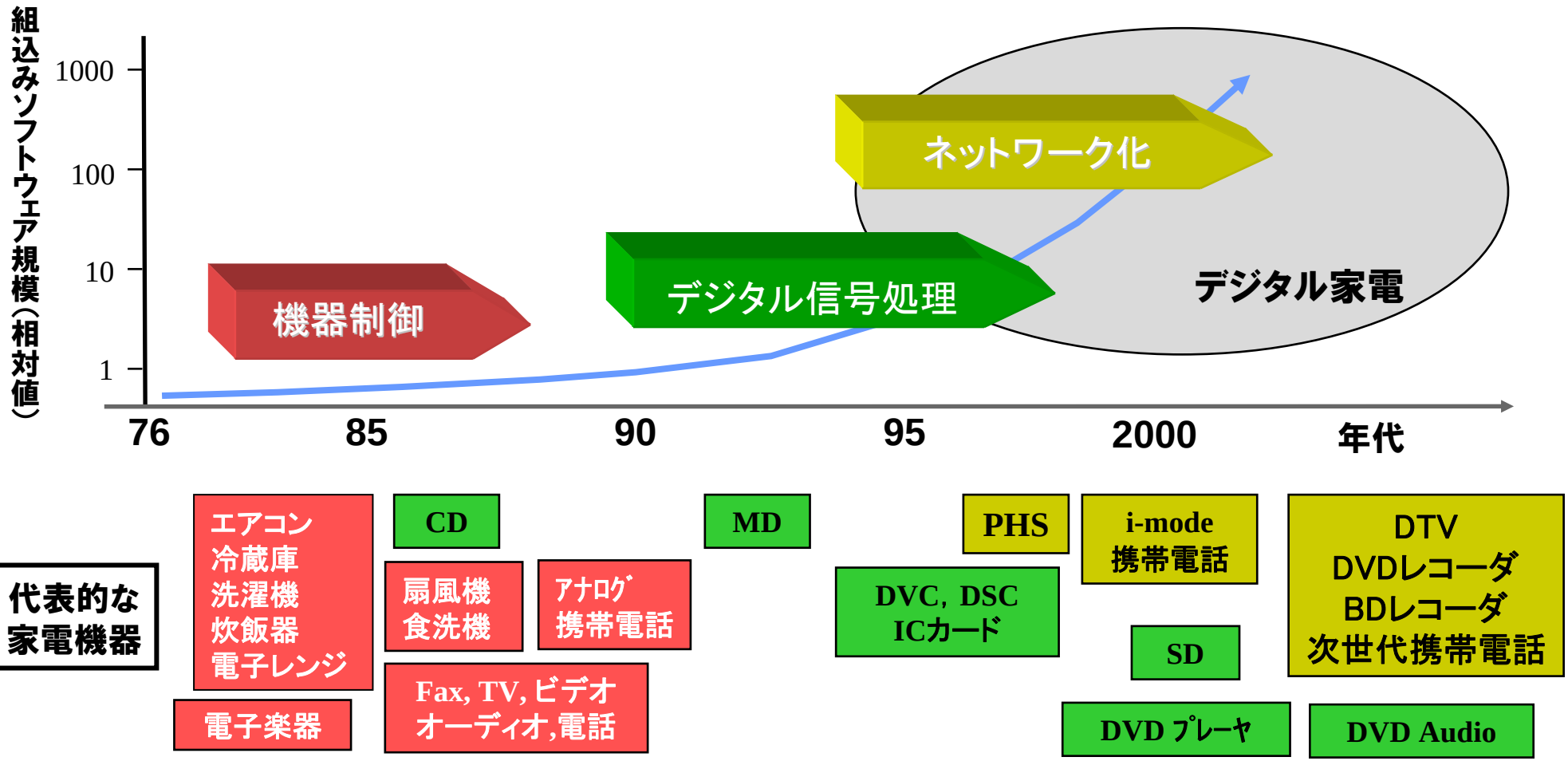
# 世界で急速に拡大するデジタル家電市場

**デジタル家電がここ数年で急速に普及**
**国内のみならず海外でも急速に伸び、世界同時で需要が拡大**



# デジタル家電ソフトウェア発展の経緯

- 家電の開発文化を引きずりながら，デジタル化による急激な規模拡大に遭遇



# 家電開発の遺伝子（1）

## 「皆さんのお宅のTVは何年製造モデルですか？」

50年代

白黒テレビ

60年代

カラー化

70年代

ﾄﾗﾝｼﾞｽﾀ/IC化
UHF対応

80年代

マイコン利用
VTR対応

90年代

SLIS化
薄型ﾌﾞﾗｳﾝ管

2000年以降

**大半は廃棄済み**

**残っていれば博物館行き** (^_^;

**まだ現役！**

**現在の商品**

HD対応・薄型ﾃｨｽﾌﾟﾚｲ
デジタル放送対応
NWアクセス対応

**家電製品の製品寿命は10-20年と非常に長い**

# 家電開発の遺伝子（2）

## 家電設計では、長い製品寿命が求められることを考慮

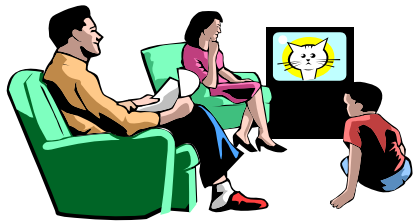
家電メーカは、長寿命時の経年変化（特にホコリ）による火災事故を特に恐れる。このためハード設計は、ホコリに弱い冷却ファンが不要となるぐらい、できるだけ熱発生しない低クロック動作を前提とする。また長寿命で故障しない品質を満たす

**家電**

BD/DVDレコーダのメイン基板例

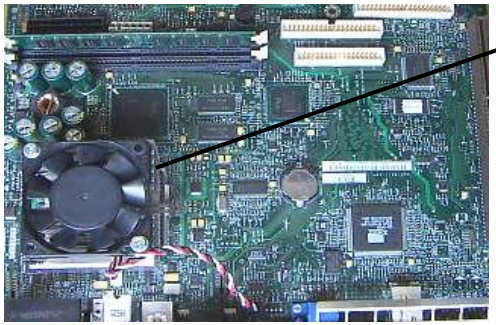
CPUクロック
= 350MHz (非常に低い)

機器全体の消費電力
= 3W ～ 48W



10年後でも安心

**PC**

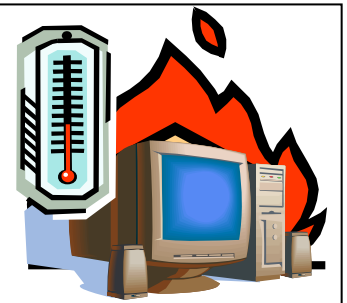


HD再生可能なPCのメイン基板例

CPUクロック
= 3GHz (熱くなる)
CPUクーラは必須

機器全体の消費電力
= 300W

２年経過後の
CPUクーラ



掃除しないでおくと
10年後は？？？
（そもそも10年使う？）

# 家電開発の遺伝子（3）

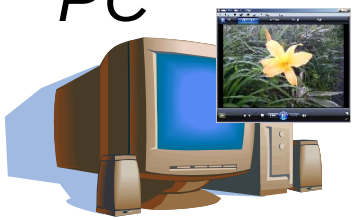
**家電は完璧な品質をメーカが保証する製品であると認知されている**

家電

BD/DVDレコーダ

- フレームスキップ等の再生不具合は、即クレームの対象
- 保証期間中は、無償交換等がメーカの責任で実行


PC



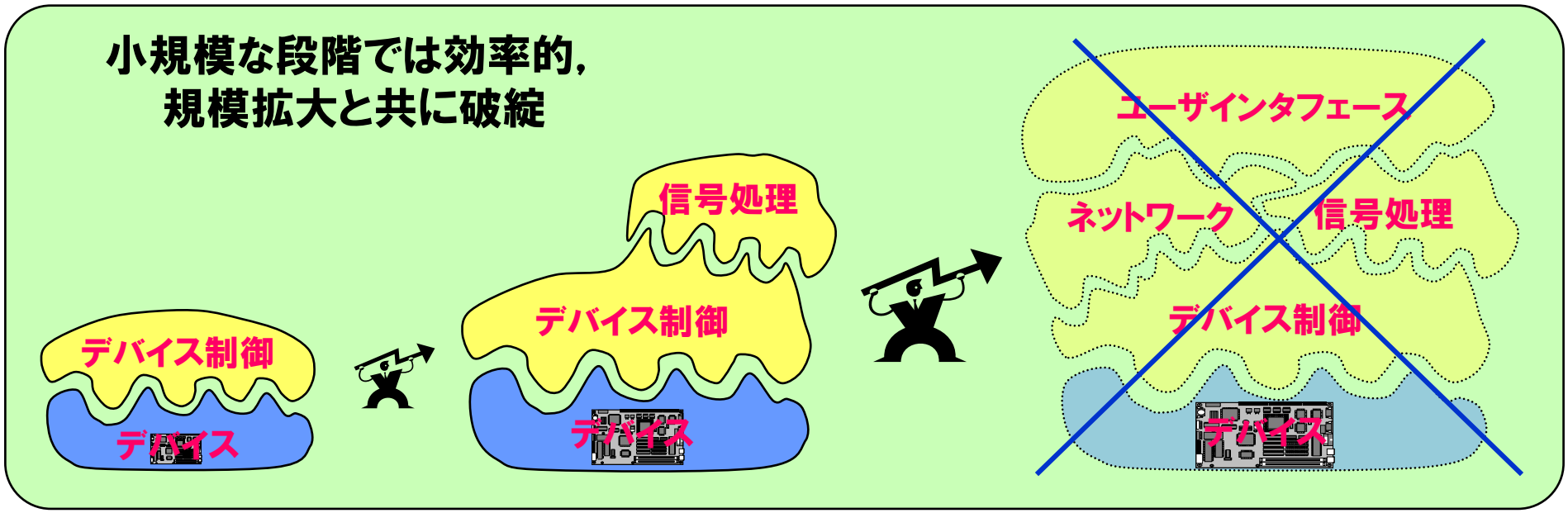
BDドライブ付きPC

- 再生性能はユーザの技量に依存
- メンテプランの契約でソフトバグフィックス等のサポート

**家電における組込みソフトは、遅いクロックや少ないメモリ空間等、ソフトにとり厳しい環境で、完璧な性能・品質の保証を求められる**

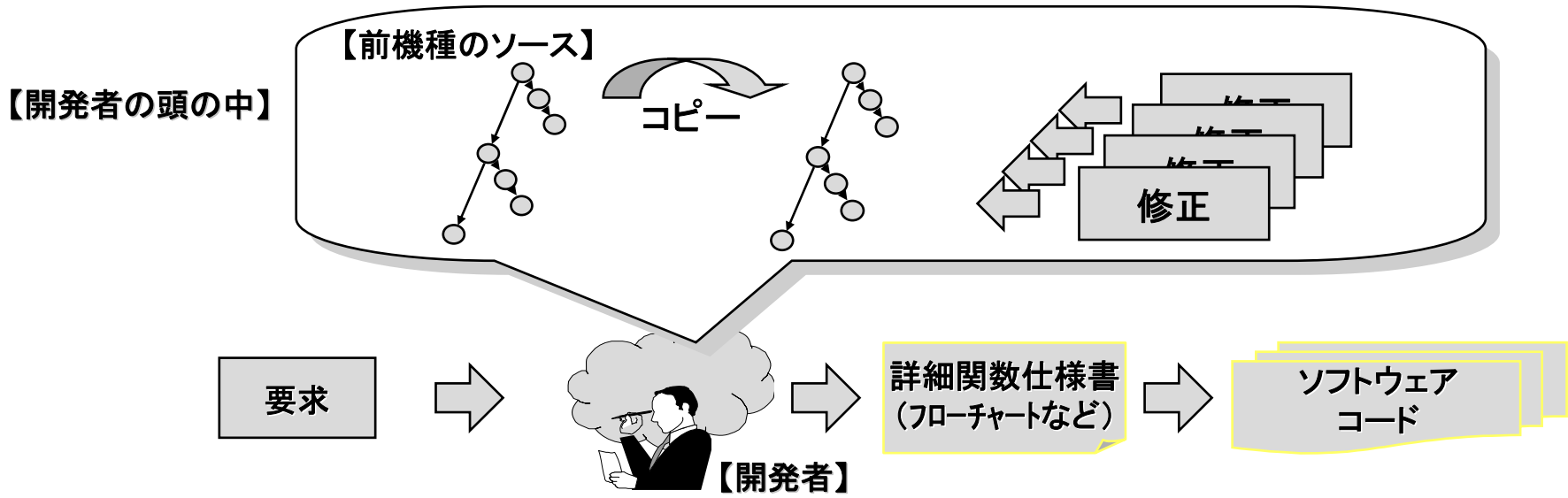
# 全体を摺り合わせで開発

- 黎明期は，ソフトウェア全体をハードウェアとの摺り合わせで開発
- ハードウェアとの同時開発の特性が全体のマネージメントスタイルを規定
  - 低消費電力ハードウェアの性能をソフトウェアがカバー
  - ソフトウェア開発時に，ハードウェアが不安定なことが多く，確実に動作するソフトウェア・ハードウェアのセットをベースに周辺を積み重ねていく



# 現場で起こっていること：事例①

- **コーディング主体の開発**
  - ＜小規模時代の開発のなごり，短納期のプレッシャ＞
  - – 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）
    - » 要求からコードへのブレークダウン過程が開発者の頭の中（属人的な開発）
    - » 場当たり的な修正によるコードの複雑化（構造劣化）
  - – 機種開発数の増加，担当者の変更で急激に開発効率・信頼性が低下



# 事例②

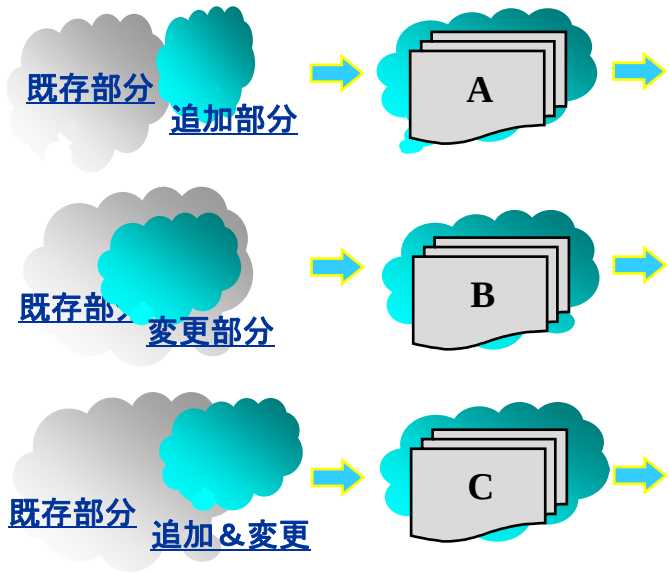
■ 分担のみ決まっており，全体把握ができていない

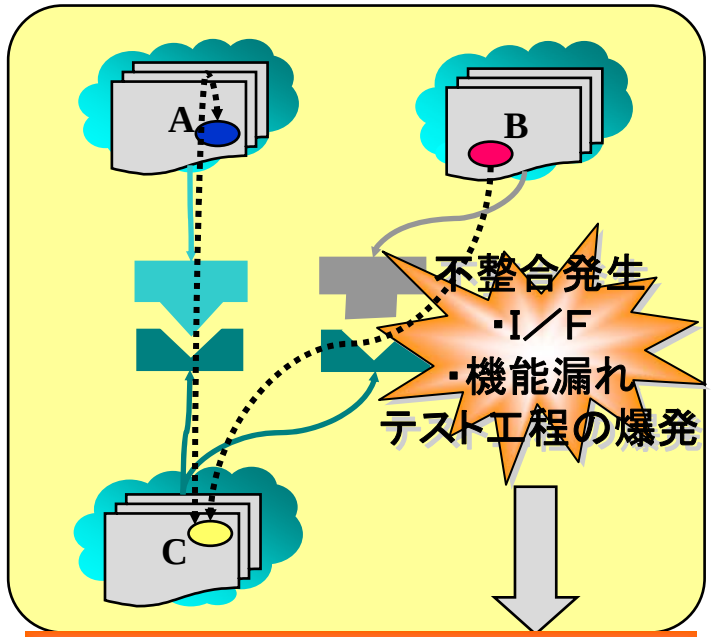
要求仕様 → 設計 → 実装 → 結合・システムテスト

あいまいな要求

全体構造・担当間のインタフェースが事前に確定していない

差分開発



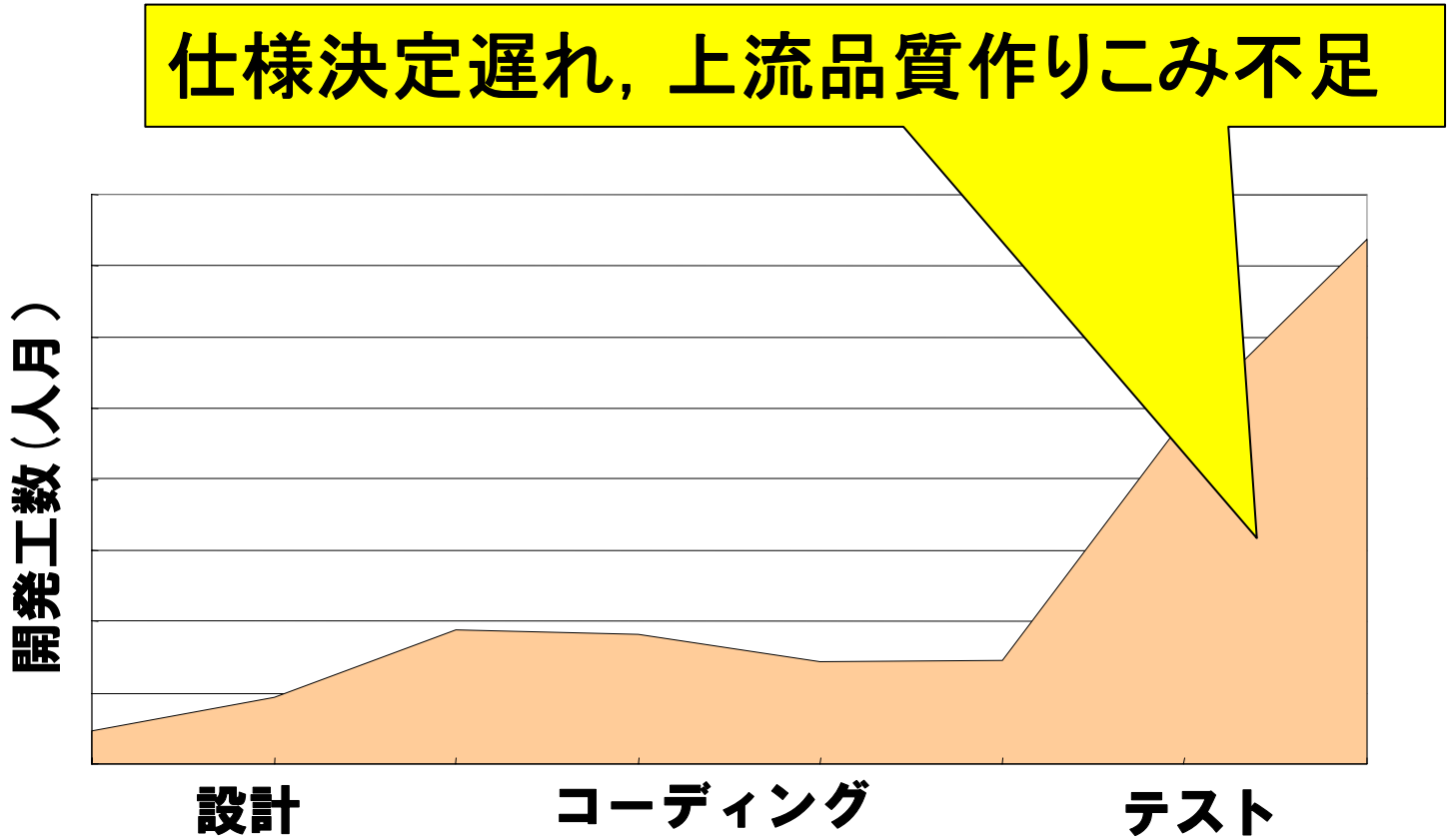

分担間の仕様調整に時間がかかる
（n 対 n）
曖昧な仕様を基に，分担開発が
進行（見切り発車）

システムテスト工程で
不整合多発

# テスト工程の工数増大

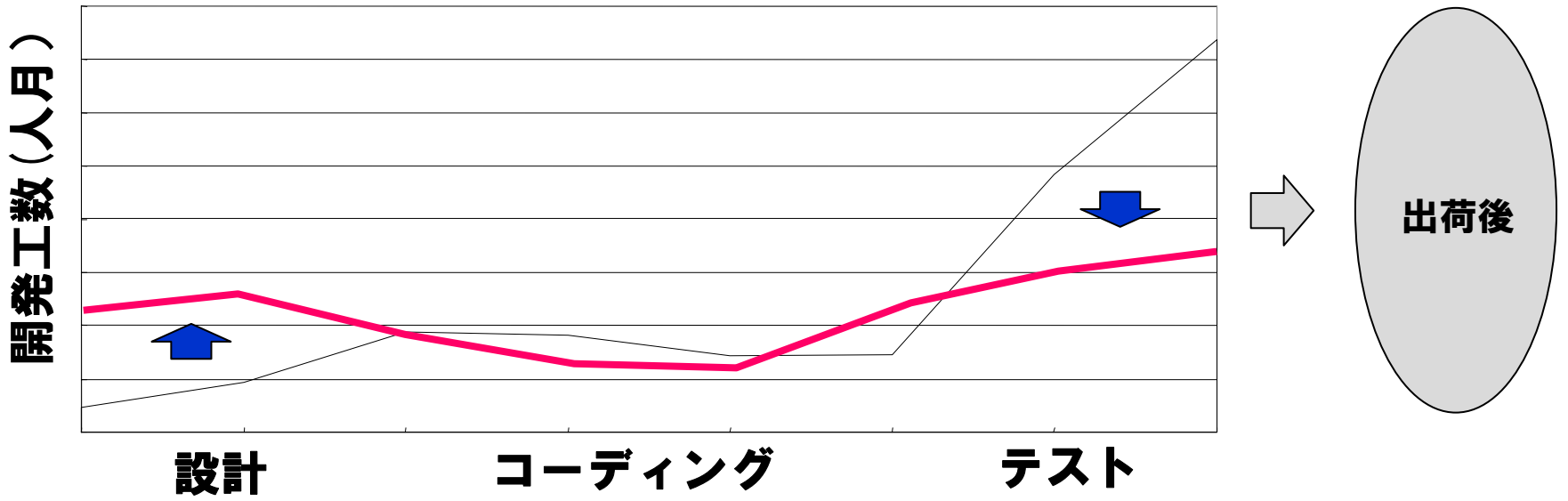
■ 後工程（テスト工程）重視の信頼性確保

ある開発事例より



# 対策

- テスト工程以前での完成度を向上させ、テスト工程の負担を減らす




戦略的な再利用

プロダクトライン

モデル駆動開発

コード解析・不具合検出（メモリリークなど）

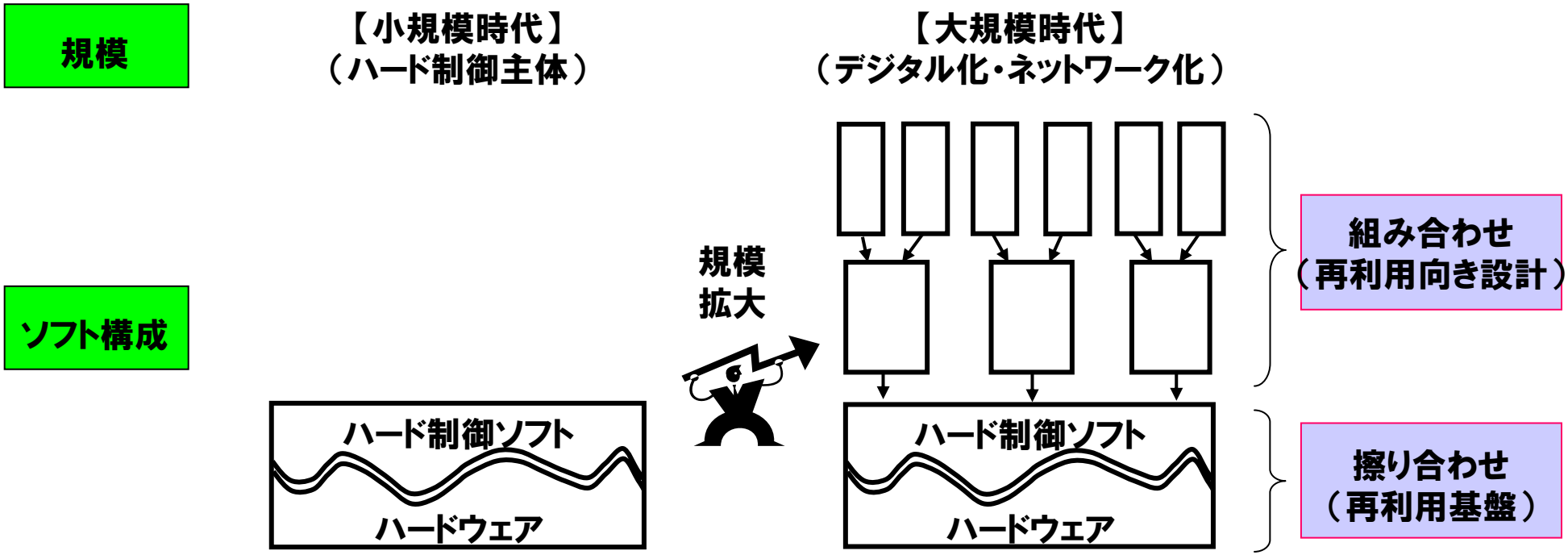
シミュレーション

モデルベーステスト

バックスラッシュ側の強化

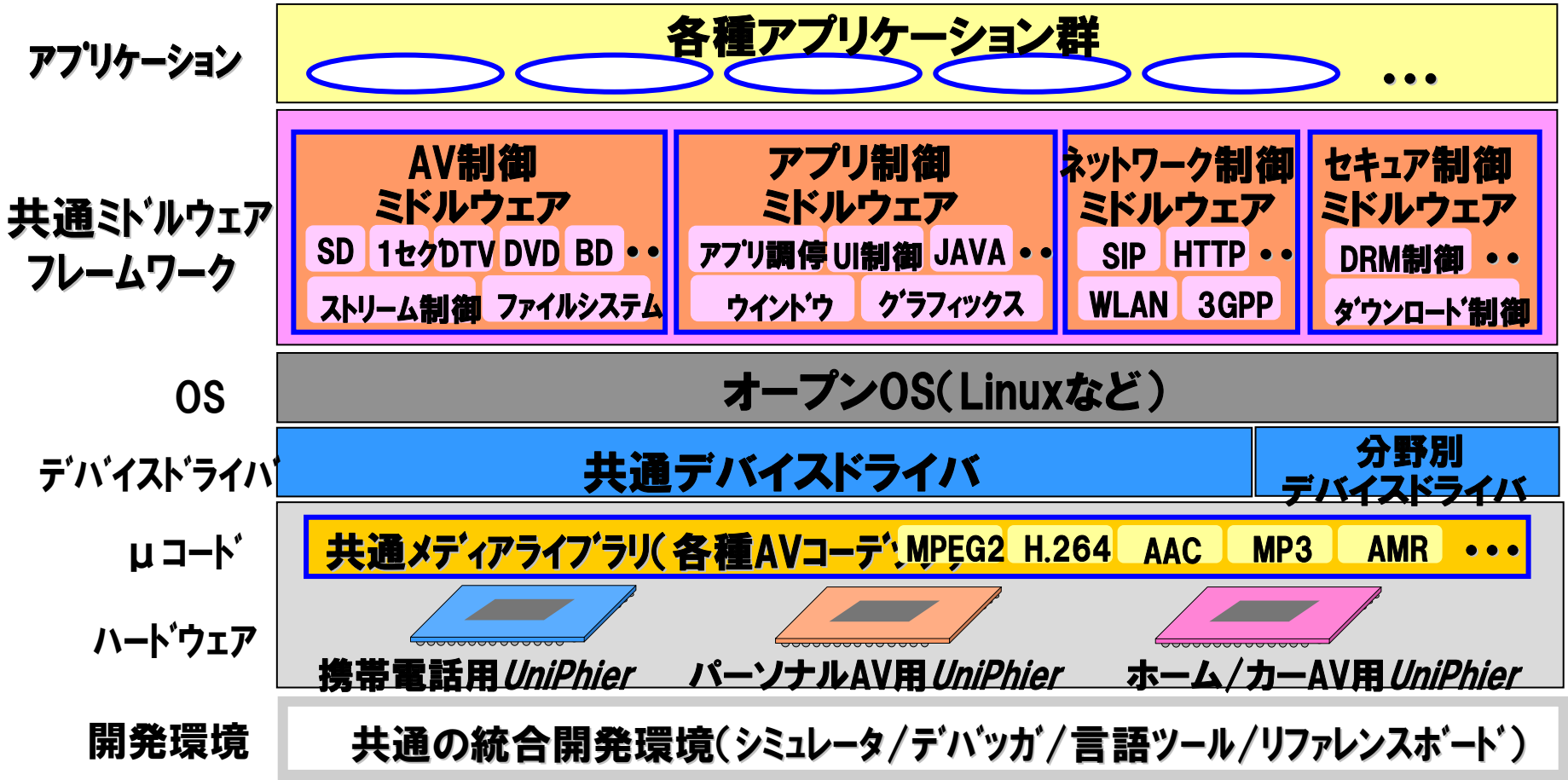
# 戦略的な再利用

- **擦り合わせ部分　⇒　①プラットフォーム化**
  - 組込みソフトウェア開発のノーハウを隠蔽。製品を超えて再利用できるソフトウェア資産の構築
- **組み合わせ部分　⇒　②設計力の向上**
  - 資産価値の高い（再利用が容易な）ソフトウェア部品の構築



# ①プラットフォーム化：UniPhier

- 商品群に対応したソフトウェアのベースを構築
  - システムLSIの共通化をベースにしたソフトウェアの再利用基盤



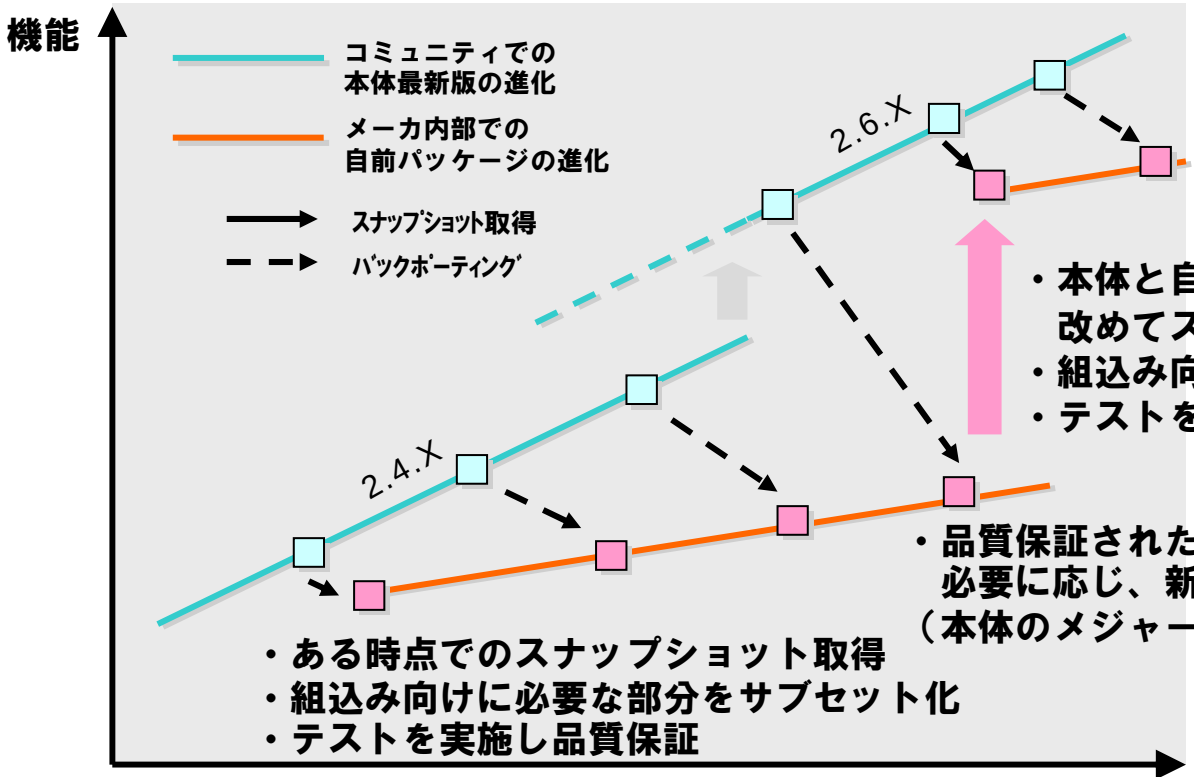
# プラットフォームの品質確保例

- OSSは多くの機能があり、利用により全体の開発効率は上がる

  しかし以下の点は留意すべし

- OSSは製品品質の保証をするものではないことを認識して活用

## 1. 自前の品質保証パッケージの確保（徹底的なテスト実施）




- 本体と自前パッケージの乖離が大きくなると、改めてスナップショット取得
- 組込み向けに必要な部分をサブセット化
- テストを実施し品質保証

- 品質保証された自前パッケージに必要に応じ、新機能をバックポート（本体のメジャーバージョンが変わっていても）

- ある時点でのスナップショット取得
- 組込み向けに必要な部分をサブセット化
- テストを実施し品質保証

take側

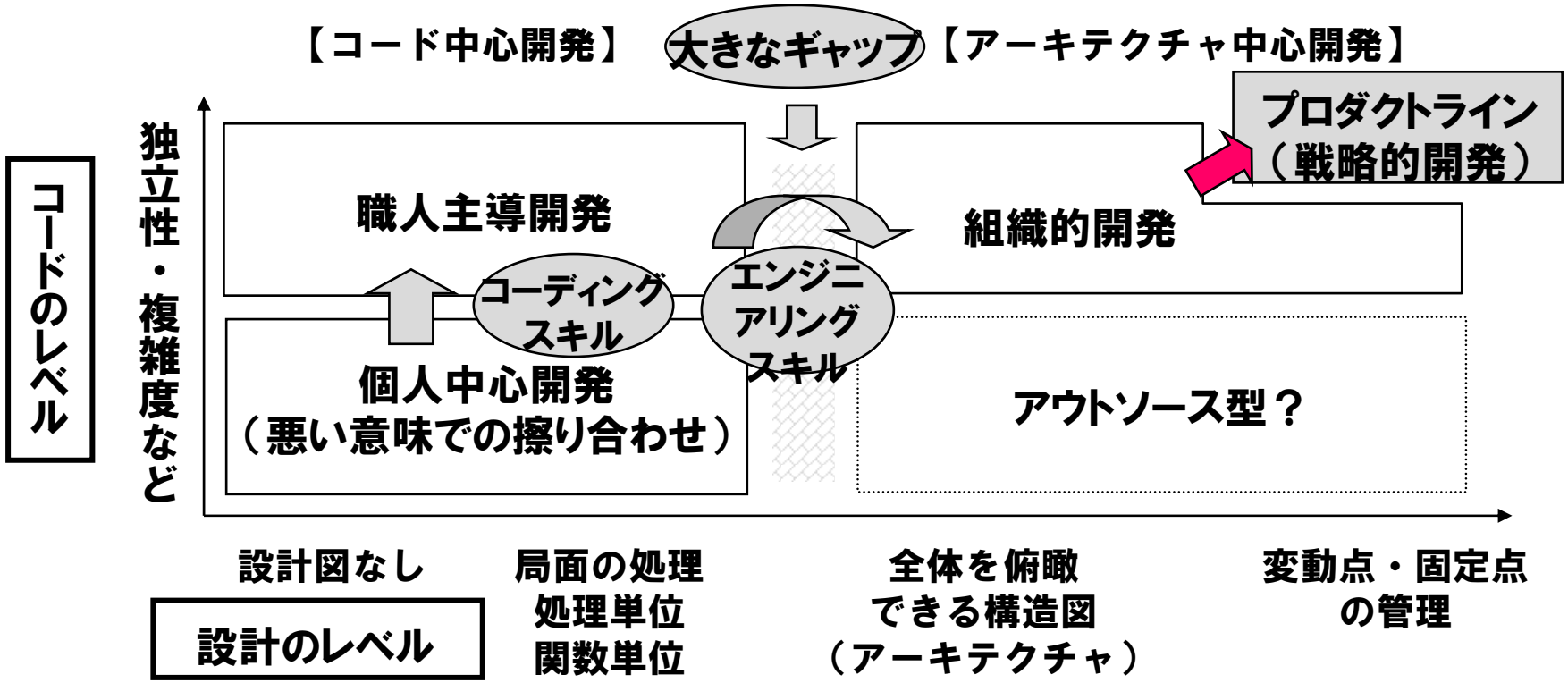
- *個別最適であっても品質保証成果は都度開示する*
- *個別最適の中で汎用化できる部分はコミュニティに返す (CELF等を通じた活動)*

give側

# ②設計力の向上:ソフトウェアの資産価値

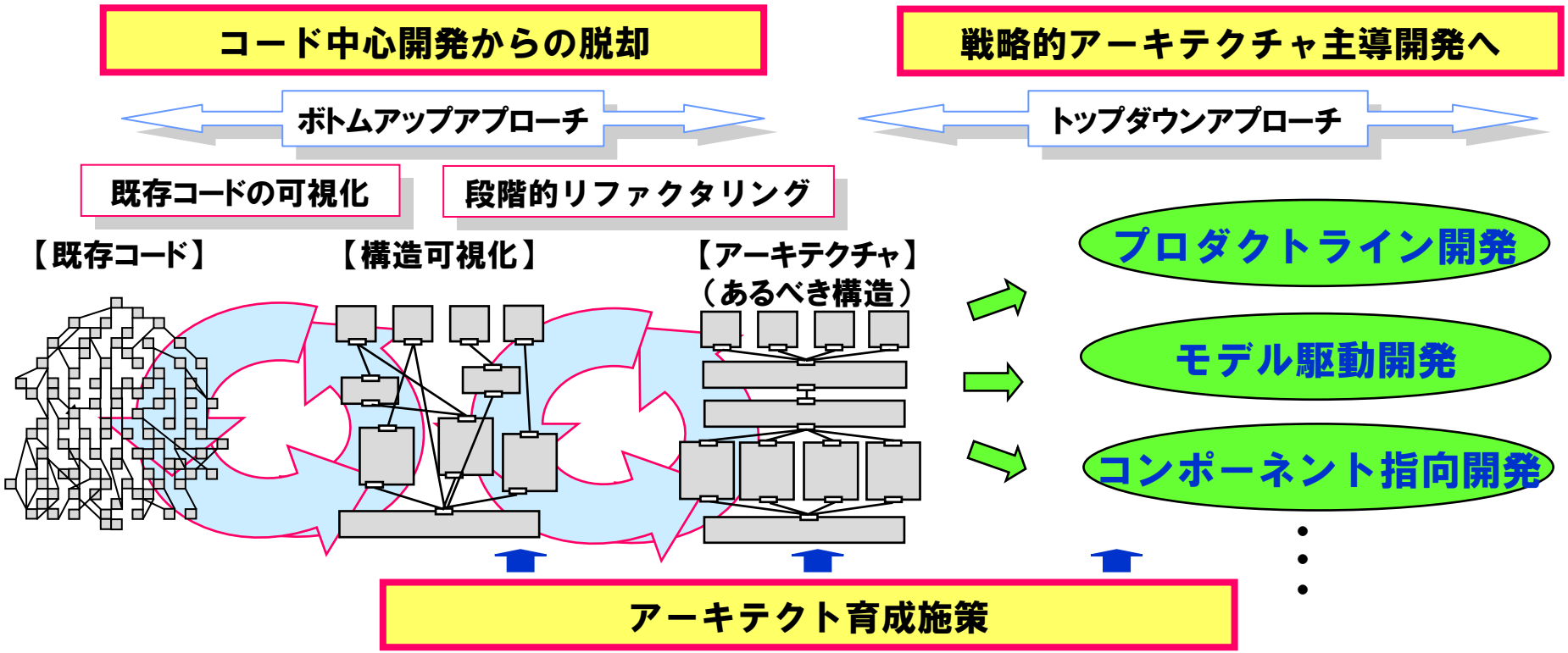
- ソフトウェアの資産価値を指標化
  - 現状の開発レベルを認識し，戦略的な価値向上マネージメントを推進
    - » 価値向上のためのリファクタリング，コンポーネント開発

**＜個人レベルのコード中心開発から，いきなり戦略的な開発には行けない＞**



# 今後の進むべき方向

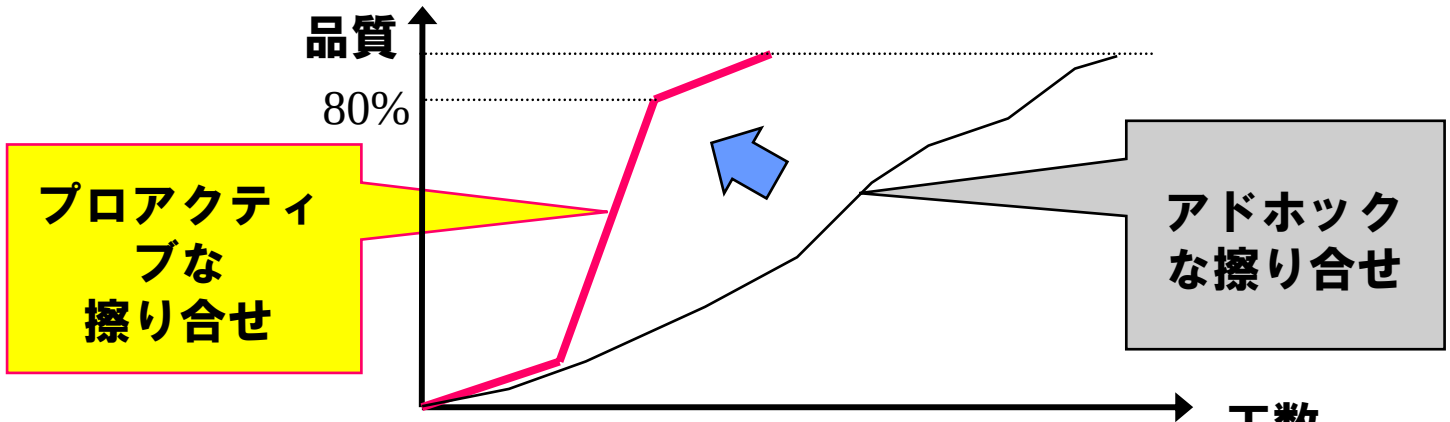
- ステップ１：コード中心開発からの脱却
  - 既存ソフトウェアの資産価値向上，アーキテクト育成
- ステップ２：戦略的アーキテクチャ主導開発へ（PLE，MDDなど）



# 日本の強み・弱み

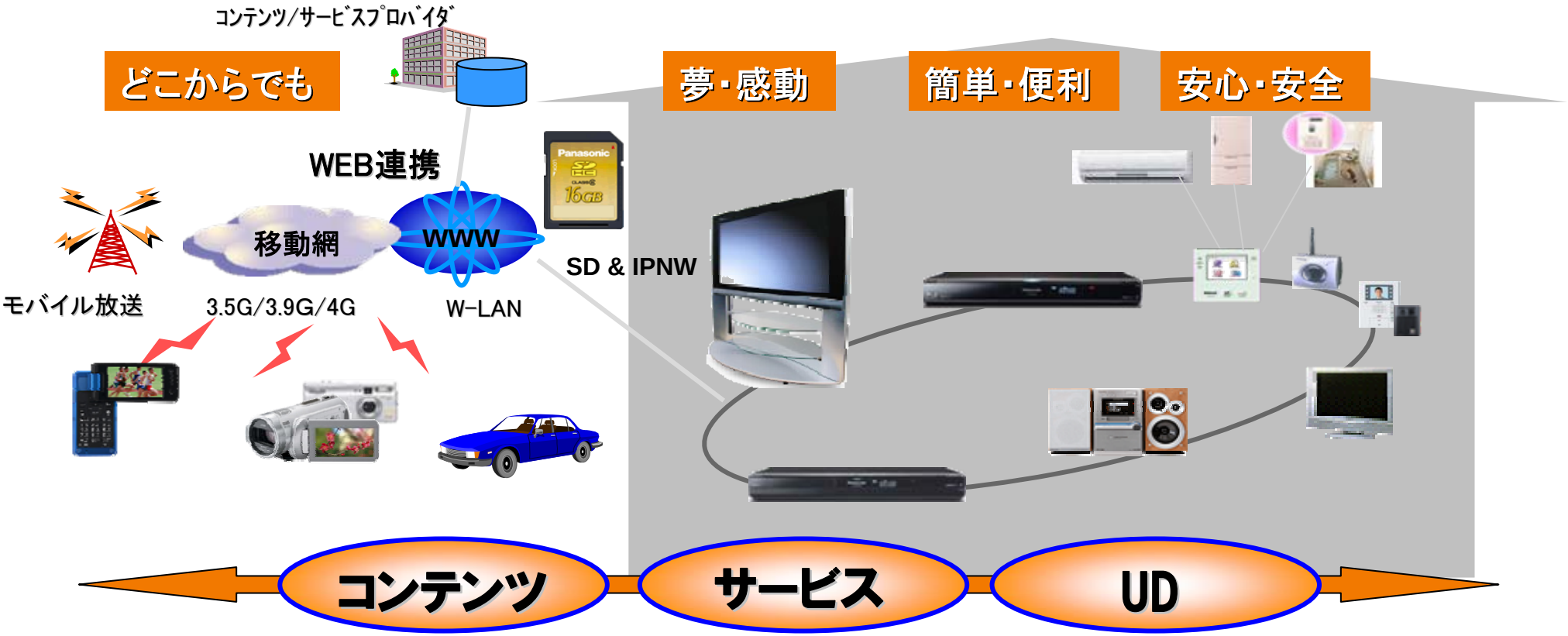
- 擦り合せによる高品質開発が日本の競争力の源泉
- しかし，全体が見えない時点からの「アドホックな擦り合わせ」では，大規模化・短納期化に対応できない

- 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
  - 組み合わせ（設計・アーキテクチャ力）の補完により，8割まですぐにできるようにする
  - 残りの2割をすり合わせる



# 家電機器が連携するユビキタス社会の招来

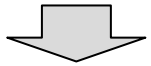
**ネットワークを活用し、利用環境を超えたデジタル家電のバリューチェーンを創出。お客様へ新しい価値をご提供**




モバイル～ホーム シームレス

# ディペンダブルシステム構築に向けて

- ユビキタスネットワーク環境においては，機能分散・製品出荷後の動的機能進化が起こる



- 機能分散ネットワークワーク系全体での信頼性確保
- ダウンロードモジュールの正当性検証
- 未知の機器や新規のサービスに繋がっても動き続ける頑強性
- 形式手法による数学的な検証？

# ご静聴ありがとうございました