

XDDPから「IoT」に挑む

- 困難な状況をチャンスに変える -

派生開発カンファレンス2016

基調講演

派生開発推進協議会(AFFORDD)

代表 清水 吉男

<http://affordd.jp/>

株式会社 システムクリエイツ

代表取締役 清水 吉男

URL=http://homepage3.nifty.com/koha_hp (更新停止中)

<http://kohablog.cocolog-nifty.com/> (ブログ)

shimz@nifty.com

自己紹介：清水 吉男



USDM



XDDP



- 株式会社 システムクリエイツ 代表取締役
- 派生開発推進協議会 代表

プロフィール

- **1968年**からソフトウェアの世界に入り、企業の業務システムや**オンラインシステム**の開発を手掛ける。
- 1977年に**組み込みシステム**の世界に転じ、POSシステムやインクジェットプリンターなどの多くの製品の開発に携わる。
- CMMとの出会いを機に、自ら考案した要求の仕様化技法(**USDM**)や派生開発向けの開発プロセス(**XDDP**)やプロセスの設計ツールとしての**PFD**を提案し、1995年から**プロセス改善のコンサルティング**を開始。
- 2010年に派生開発推進協議会を設立し、これらの普及活動へ。
- 『**硬派のホームページ**』 主催者

1973~1995の22年間
(40数PJ) 納期遅れなし

- 「SEの仕事を楽しくしよう」 (SRC)
- 「わがSE人生に一片の悔いなし」 (技術評論社：新書)

本日のお話の内容



lot

IoT への時代のうねり

日本の開発現場が犯した過ち



lot

XDDPによる派生開発から新規開発へ準備



lot

XDDP : eXtreme Derivative Development Process の略。派生開発に対応するように開発したプロセスです。(参考文献②)

USDM : Universal Specification Describing Manner の略。要求仕様がモレない表現方法として開発したものです。(参考文献①)

PFD : Process Flow Diagram の略。合理的なプロセスを設計するためのツールでDFDをアレンジしたものです。

CMM[®]、Capability Maturity Model、およびCapability Maturity Modeling は、米国特許商標局に登録されています。

CMMISM はカーネギーメロン大学のサービスマークです。

IoTって？

売れるものがなく、日本経済は停滞したまま

官製円安や政府主導の成長戦略も息切れ状態

そんな時に・・・

IoT



IoT = Internet of Things

一般に「モノのインターネット」
と訳されているが・・・

IoTって
何をすれば
いいの？



機能の競争から価値の競争へ

部品のデジタル化



新規参入が容易 / 価格競争の激化

新しい機能もすぐに追いつかれる



機能の競争から価値の競争へ

背後にあるのは...



QCDの同時達成技術の普及

agile 開発も

さらに、価値の継続的提供の競争へ

ソフトウェアで提供が可能に！

IoT

高性能な
センサー技術

高速通信の
環境

安価な
クラウド環境

ディープラーニングなどの
ビッグデータ解析技術

ソフトウェア開発における
QCDの同時達成技術

新しい時代の背景
にあるもの

「Software Defined XXX」での競争へ

◆ Software Defined XXX



Car
Engine
Machine
Device
など

早く買って後悔しない



- **継続的**な価値の提供
- デファクトスタンダード化
- 価格競争に巻き込まれない
- 更新されたソフトを無償提供

防災システムのような
ライフの長い製品には
不可欠な対応

どうして、日本の家電産業が停滞しているのかお気づきですか？

「家電」は無くならないのに、
「家電」に対する要求に対応できない？

自動車にも「価値」を提供できるか

自動運転

1つの「機能」と扱うか？



- 機能としてのゴールを定義するのが難しい
- 統一性をどうやって確保する？

ユーザーにとって「価値」であるには？



- 買った後の進化
- ユーザーによる運転スタイルの選択
- さらに「ステータス」も



早く買って後悔しない

テスラ 「モデル3」

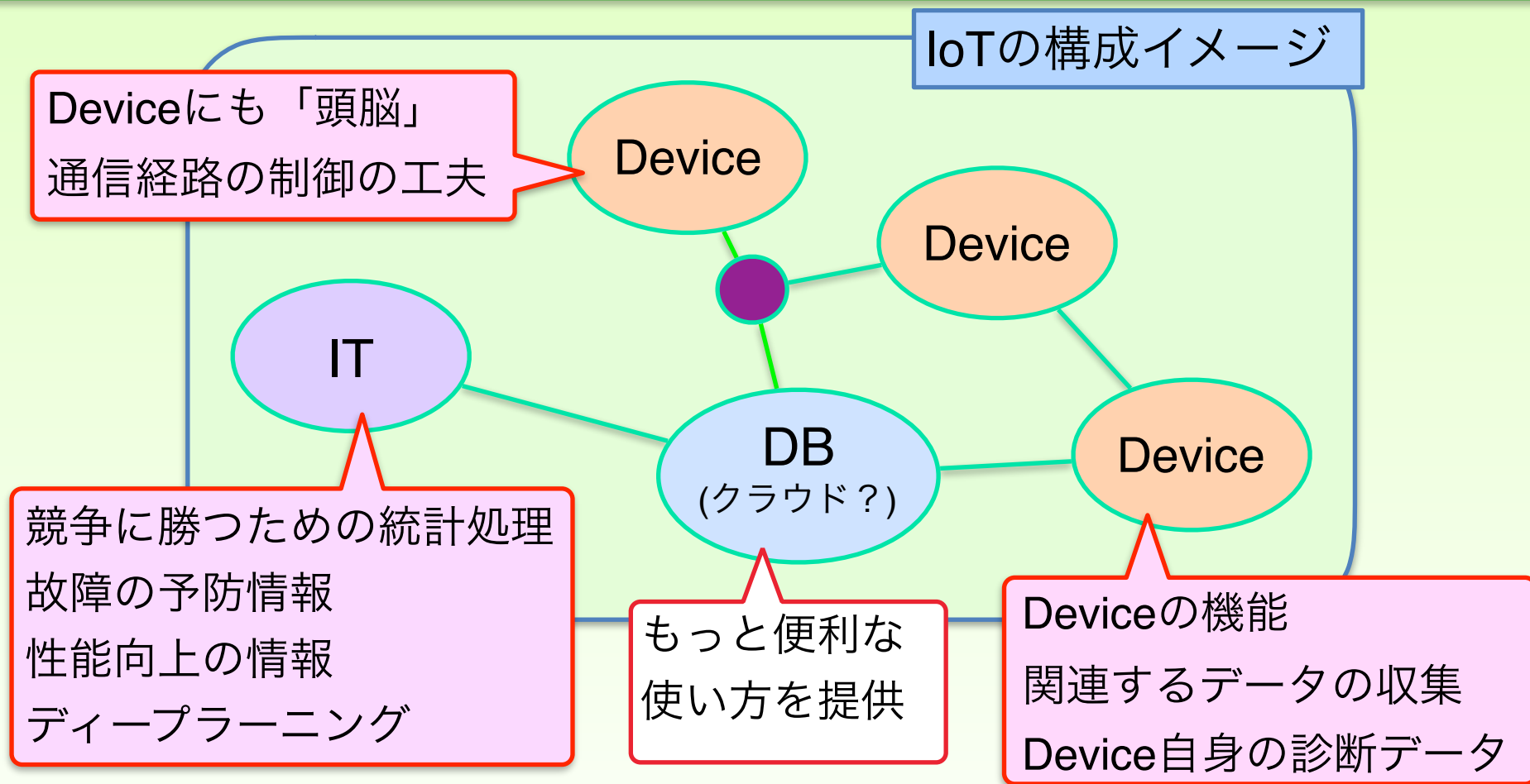
受付3週間の予約台数 「40万台」

この数字の意味するものは？

「段階的に自動運転機能を追加する」

(テスラ社HP)

IoT...新しい競争の時代へ



- あらゆるものが「ソフトウェア」の上で成り立つ時代へ

“ソフト主体”の意味

- 従来の考え方・・・

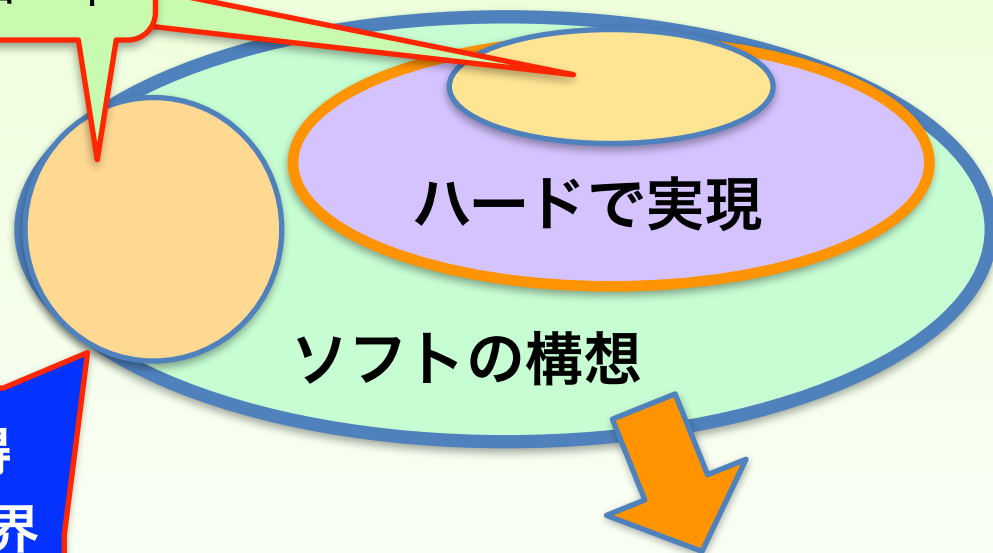
ソフト（プログラムコード）はハードが提供する機能を動かすための手段



- これからは・・・

ハードはソフトウェアを実現するための手段の一部

プログラムコード



IoTを機に、新しい市場を獲得するために、「ソフト」の世界からも一斉に参入してくる

社会のルールを変える

日本の開発組織は
このような時代の要求に
対応できるのか？

市場の要求に日本の組織はどう対応したか？

- 1980年代の品質要求への対応ミスで、90年代以降の納期やコストの要求に対してプロセスで対応できない事態に陥った

時代	市場の要求	支援ツール・考え方等	日本の対応
80年代	品質への要求	F・クロスビー「Quality is Free」 ISO、CMM、6Σ	テスト工程で品質に対応 この後の全ての対応に悪影響
90年代 前半	+ 納期の短縮要求	EA, SPL, スケジューラー 自動生成ツール、CASEツール パッケージソフト BPR	ツールに合わせてプロセスを変化させる ことができなかった 外注化と一人の分担を狭くして人海戦術 で対応
90年代 後半	+ コスト削減要求	開発拠点の分散 プロセスの多様性 → agile開発へ	プロセスの必要性を見極めることができ なかった 一層の外注化と「オフショア」へ ソースコードの劣化と開発技術の空洞化
2000年 以降	QCDの同時要求 機能から価値へ	クラウド、IoTなど	??? (何をすればいいのかわからない?)

80年代 プロセスで品質要求に対応すべきだった

「品質」に対して強い要求

世界はプロセスで対応

支援

- 「Quality is Free」 (フィリップ・クロスビー)

- ◆ 「最初から正しく仕事をすれば高品質は確保できる」
- ◆ 品質コスト (予防コスト + 評価コスト + 失敗コスト) は、ものごとを正しく行わないことの代償



当時の日本の品質は「テスト」によって獲得したもの

多くの組織では、プロセスを**変化させる技術**も
変化させたプロセスを**安定させる技術**も習得せず

90年代 ツールや「標準」が提案されたのに

- 納期の短縮要求が出る中で、2つの象徴的な出来事がある

① CASEツール（構造化分析のツール）

「このツールは我々の開発現場のやり方に合わない」で却下

② パッケージソフト（業務系）

自分たちの業務に合うように大幅なカスタマイズを要求

納期の短縮のために、自分たちのプロセスを変化させて
「標準」を取り入れるという発想に至らなかった

➡ 「標準」も決まったあとは「固定」が定着

結局

プロセスで、変化するQCDの
要求に応えるスキルを手に入れ
る必要性に気づけなかった

90年代～ 過度な外注（オフショア）化の後遺症

- 納期短縮要求 = 人海戦術で細分化
- コスト削減要求 = 外部（オフショアを含む）へ委託

外注管理もできない

要求仕様も書けない

見積もりもできない

ソースコードの劣化

RTOSも搭載していない



派生開発も外注で回しているだけで
新規開発もやったことがない

開発技術の空洞化

IoTもオフショアでやるつもりですか？

企業の収入源

新機開発で
競争力を組み入れる

- 「IoT」で勝つための仕掛けが必要
- 劣化したソースコードへの対応
- RTOSだって搭載しなくちゃ
- 保守性にも対応しないと・・・

IoT

IT

DB

Device

派生開発の技術でも他社に負けない技術が必要

派生開発で
機敏で効率的に
価値を提供し続ける

デジタル化の取り組みをアウトソースしてはならない

ジェフ・イメルト (GE会長)

自前で開発した方がよいと
わかっているても・・・

「IoT」に対応できるエンジニアがいない！

「日本のソフトウェア業界はブルーカラーを作ってきた

これは、ソフトウェアイノベーションを阻害するもの」

ロバート・コール（CA大学バークレー校名誉教授／同志社大学大学院）

人海戦術で
狭い守備範囲

長いこと設計も
していない



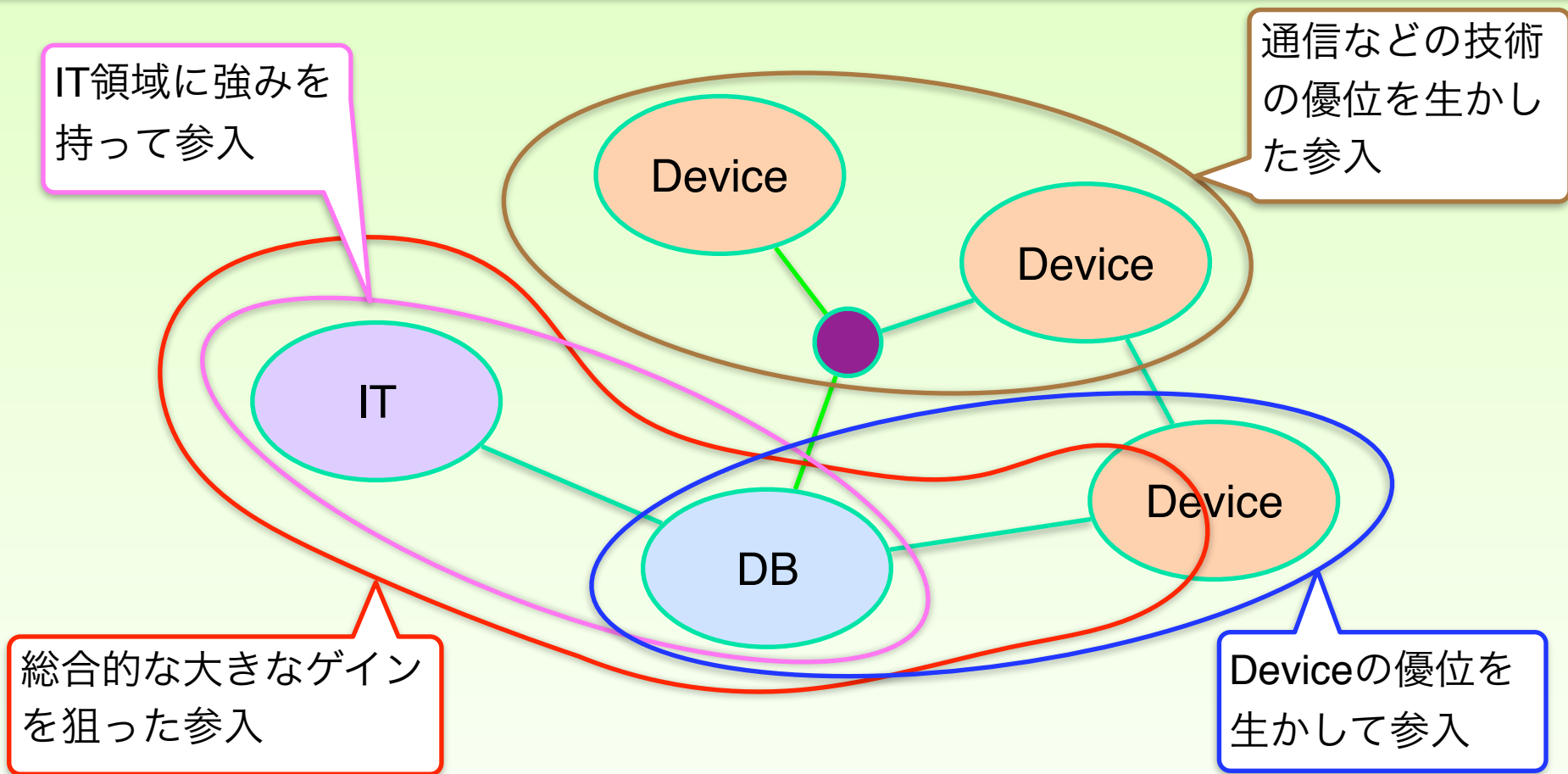
外注任せで要求仕様
も書けない

開発技術も空洞化



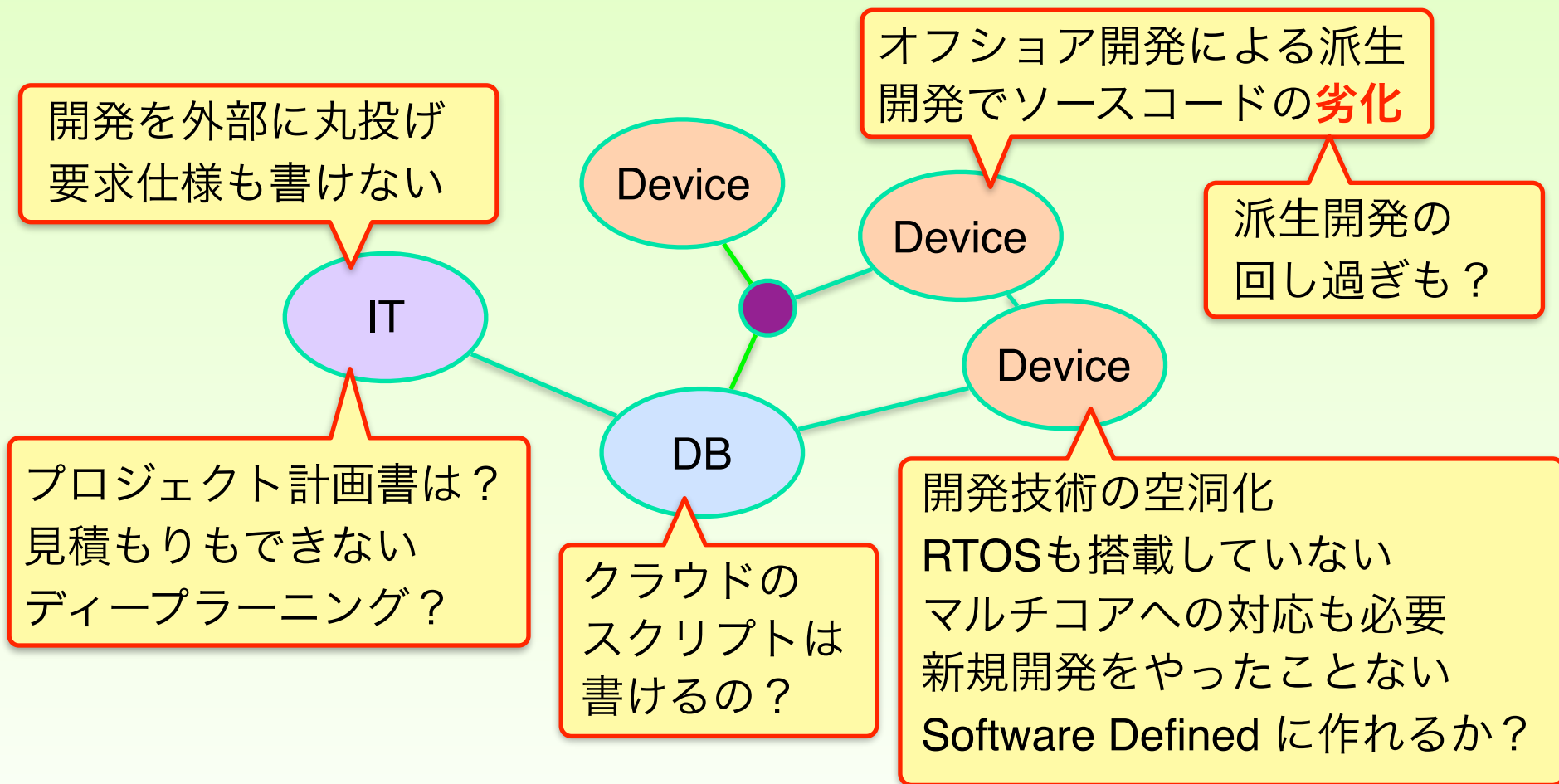
この状態で、誰が「IoT」の構想を考え、対応できるのか？

過激になる参入競争



高度なソフトウェア技術を絡ませないと、価格競争に巻き込まれる

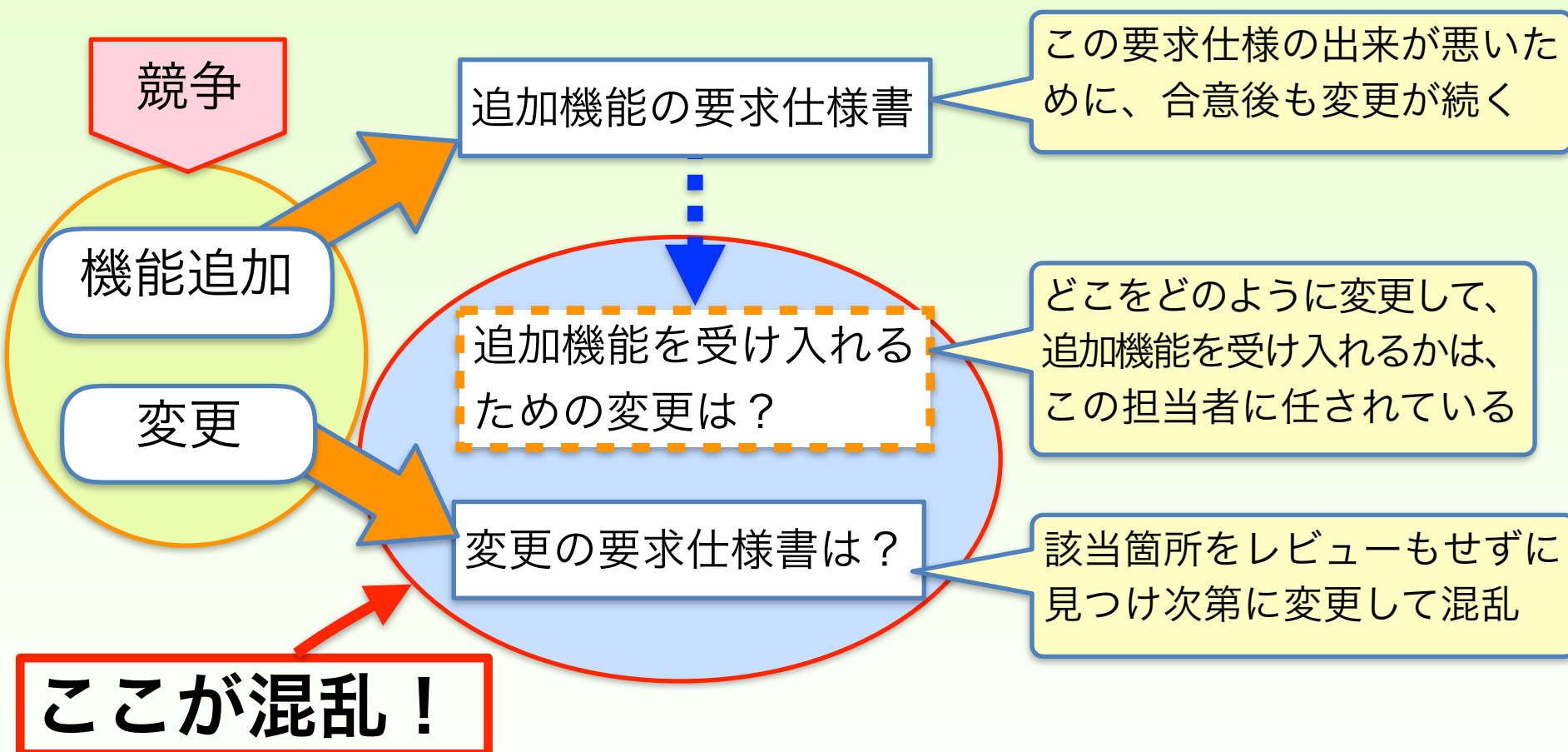
「IoT」に対応するにも、問題山積！



多くの「Device」で新規開発は不可欠になる

混乱する派生開発

- 一般に派生開発では「**秩序**」が失われている
- 見つけ次第に変更することで「**部分理解の罫**」にはまる



こんな状態で新規開発できますか？

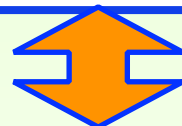
新規に作り直す必要
があるのはわかる



でも現行製品の
バージョンアップ止める
わけにはいかない

派生開発も
自前でやって
いない

自社で派生開発に対応へ



新規開発への準備も必要

すぐには
新規開発
できない

当面は、派生開発で回すしかないが



XDDPを活用して、
目の前の派生開発に対応
しながら新規開発に備え
る方法を説明します！

限られる選択肢

派生開発を続けながら新規開発の準備をする

XDDPを推奨！



機能追加
と変更を
別プロセス
で対応

- 機能追加

要求仕様書をUSDMで作成することで
完成度を上げる

- 変更

変更情報を「変更3点セット」で書き出す
ことで「**部分理解の罠**」を回避する

生産性と品質
の両立へ

派生開発の中での新規開発への備え

機能追加

• 追加分の要求仕様書

仕様化作業では「60~100仕様/H」へ

ベースライン設定後の仕様変更率 < 3%

ストップウォッチを使う

生産性 → 要求引き出し技術の向上

新規開発時の「量」
に耐える準備

• 追加分の設計

機能単位で良い設計を積み上げる

設計技術の蓄積

- モジュールの尺度を満たす設計
- 保守性を満たす設計
- 「設計（の意図）」を表現する

Switch文を使わない状態遷移など
→ リファクタリングで活用する

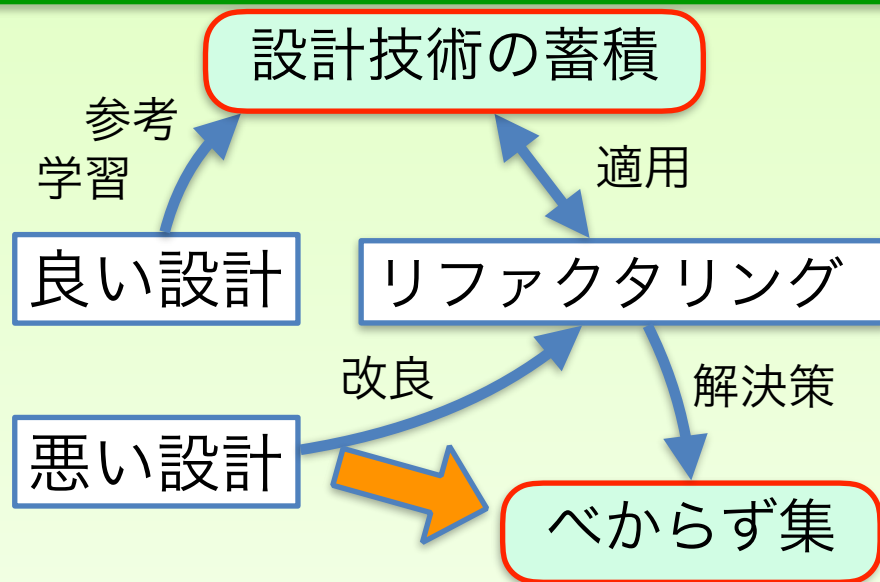
ソースコードを読む力へ

派生開発の中での新規開発への備え

変更

- ソースコードの読解
 - 事前調査／スペックアウトで良い設計などを学習できる

- 「変更3点セット」によるXDDP



適切にレビューを実施

実装生産性 > 80行/Hを目指す

⇒ 「部分理解」の状態を緩和

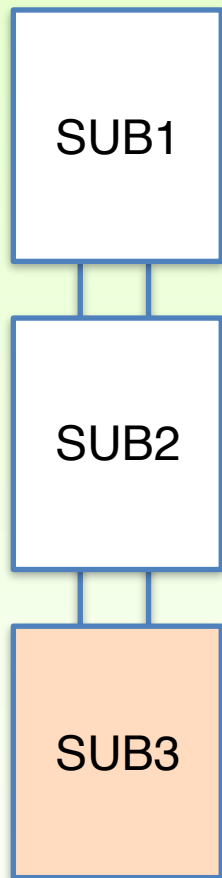
変更3点セットの出来栄え

⇒ バグの未然防止

⇒ 開発工数を余らせる

XDDPに初めての外注SEの成果が示すもの

連携する3つのサブシステムの派生開発をA社が受注



「SUB3」で3名増員の申し出

提案

「SUB3」を契約から切り離し

別の外注先から1名確保

条件=設計ができる人

5年目の人、新規開発のみの経験

準備・・・3週間（半日×7日程度）

製品、ドメイン、ソースコードの構成、**「XDDP」のトレーニング**

着手 > 期間2.5ヶ月で完成

バグ=0件

「変更3点セット」のレビューには発注側の経験者（1名）が参加

	変更（全）	内削除
変更行数	7000行	1500行
実装工数	50H（削除作業を含む）	
実装生産性	140行/H（全変更行を対象）	
	110行/H（削除行を対象外）	

この時、A社は未完了

A社の工数の1/3以下で完了したことになる

せっかく捻出した工数を、
しっかり新規開発の準備
に活かすこと

決して「不当応援」に
回さないこと

捻出した工数で新規開発の準備をする

新規開発の方法の検討

- アーキテクチャ・スタイルなどの検討
- 全体構成の検討
- 新規開発のしやすさを検討

データ中心アーキテクチャ
データフロー・アーキテクチャ
コール&リターン・アーキテクチャ
オブジェクト指向アーキテクチャ
レイヤー・アーキテクチャ
(実践ソフトウェアエンジニアリング)

新規開発の机上演習

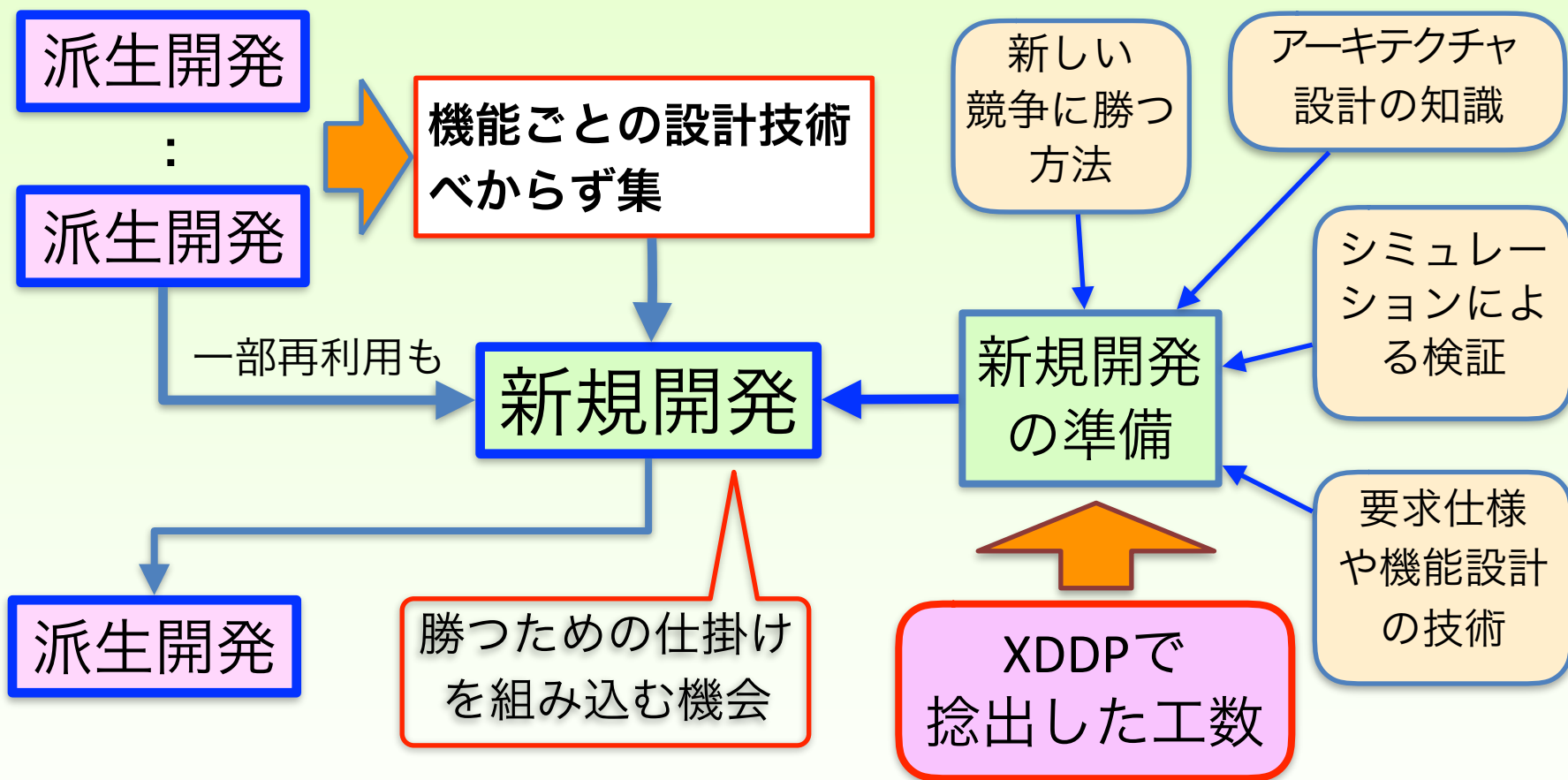
- 本格的な新規開発の演習
- 短期間での「**アーキテクチャ再構築法**」

新規開発へのその他の準備

- 保守性などの品質特性への対応方法
- 新規開発時の設計ルール

準備を整えて新規開発に着手

- 現実問題として派生開発を中断できない
- 並行して準備を進めた上で新規開発に着手する



新規開発と派生開発の両方の技術を

- IoTの時代では、比較的短い間隔で新規開発が必要になる可能性がある

「遷宮」の考え方を活用

ソフト開発の
世界にも応用

派生開発
(XDDP)

新規開発

勝ち続ける組織へ

両方の開発力なくして
事業は続かない

一定期間で新規開発するメリット

方針 = 5 ~ 10年で作り変える

組織の文化へ

- 新規開発の要員を確保できる
- 派生開発と両方に対応できるように育成
- 新規開発を意識することで時代の変化に敏感になる
- 勝つためのアーキテクチャの研究を継続できる
- 社内の技術ロードマップに沿って教育できる

「派生開発」のあり方も変わる

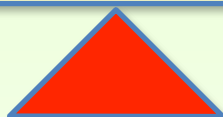
勝つための選択肢も増える

ソフトウェア抜きに日本経済は回復しない

製品の競争力



ソフトウェアの競争力



優れたソフトウェア技術者



パパ！
僕らの仕事を
作ってくれたんだ
よね

国を挙げてここに**投資**
すること！

ソフトウェア技術者のスキル向上
が日本の製品の競争力の源泉

短い時間でしたが・・・

IoT 時代において**派生開発** (XDDP)

重要かおわかりいただけただけでしょうか。

