

ITソリューション塾・第40期

デジタル・サービス提供の実践

昭和・平成・令和 皆さんは今の時代に生きていますか？

2022年7月6日（第8回）

株式会社戦略スタッフ・サービス

戸田孝一郎

ディスクレーム

できうる限り客観的な事実（経験）&データに基づき資料を作成しておりますが、当資料内での意見は、私 戸田孝一郎の私見です。

また一部の方々（ユーザー企業IT部門、ベンダー経営幹部、PM、SE、品質管理の皆様など）には不快な思いを抱く内容が含まれます。

デジタル・トランスフォーメーションとは？ (デジタル時代のビジネスモデル)

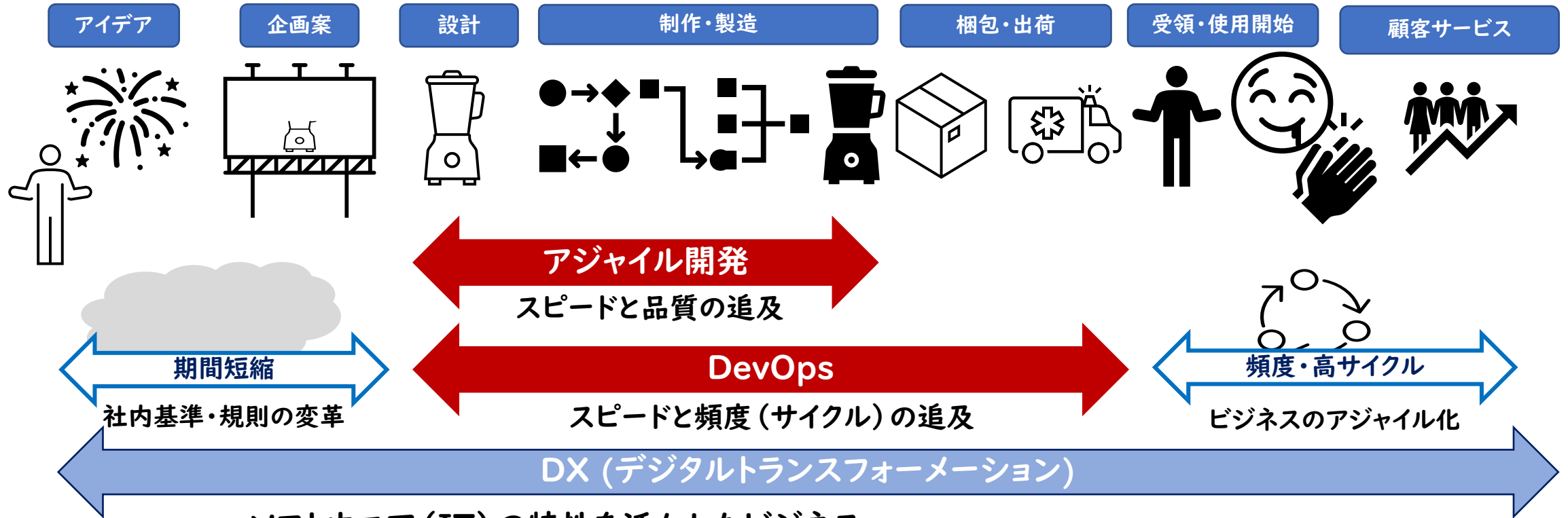
最新のデジタル技術の適用が全ての組織（営業、マーケティング、生産、経理、人事、顧客サービス等）に浸透する。デジタル技術と協働して新たなサービス・オポチュニテキを開き顧客の行動習慣を変質させることにより既存のビジネス習慣にディスラプション（破壊）を起こす。

ハイパーコンペティション

共創

- 「ITの浸透が、人々の生活をあらゆる面でより良い方向に変化させる」という概念
- 既存ビジネスをアナログからデジタルへ、デジタルからアナログへとシームレスに変換できる組織への変革
- 全ての企業はITを中心とした組織に変わる
- 全ての組織がITサービス・プロバイダーに変質するという事
- IT（情報技術）が一般化すると優位性はデータを保有する側に移る
- デジタル技術は、変化のスピードをより速くさせる
- 既存のマネジメント・モデル（静的な管理機構）が機能せずよりアジリティーを求める動的なマネジメント・モデルが必要となる

DXおかしな話



ソフトウェア (IT) の特性を活かしたビジネス
形が見えない (全体像が見えにくい)
何時でも壊せる (産業廃棄物が出ない工業製品)
製品評価は使用感がすべて (形や機能は二の次)
製品説明は概念 (空想・夢想の世界)
品質向上・顧客満足はお客様からのフィードバックが重要

19世紀の常識 vs. 21世紀のニューノーマル

権威（階層構造の組織）

物事には必ず正解が存在する。
正解は一つ。

時間の変化は無視できる。
普遍性がある。

同種の作業は、まとめて実施した方が効率が良い。

品質は管理する（専門スタッフ）

自動化は善

個を捨て、皆の幸せ

計画：精緻に作成し、死守するモノ

静的（スタティック）な管理
形式主義（ドキュメント＝死亡報告書が全て）

自律（フラット構造の組織）

正解は多数ある。（正解は無い。）
ベストでは無く、ベター（より良い）

時間によって結果は異なる。
ある一点の真実（MOT）がある。

作業は一つ一つ順番に作業した方が早くて安定している。

品質は作りこむ（現場）

自動化よりも適応力

個の幸せ = みんなの幸せ

計画：常に作り直すモノ（計画より計画作り）

動的（ダイナミック）な管理
必要な事を必要な時に

システム開発の今までの常識

要求をしっかりと定義できなければ良いシステムは作れない。

システム全体像を描かなければ、良いシステムは作れない。

プロジェクトは計画立案&管理と開発実行部隊に分けた方が効率が良く、良いシステムが作れる。

プロジェクト計画はできるだけ精緻に作成して、計画を死守できれば上手く行く。

品質は専門家によるレビューを実施すれば向上する。

役割分担など事前にしっかりと組織編制をすれば、プロジェクトは上手く行く。

DBは無闇に変更するものではないので、データ構造をしっかりと見極めて作成する必要がある。

Is it make sense ? ? ?

VUCAな世界

VUCA（ブーカ）はビジネス用語。**Volatility（変動性）**、**Uncertainty（不確実性）**、**Complexity（複雑性）**、**Ambiguity（曖昧性）**の頭文字。1990年代後半にアメリカ合衆国で軍事用語として発生したが、2010年代になってビジネスの業界でも使われるようになった。

1990年代以前の戦争は、国と国との戦いであった。参謀本部が作戦を立案し、現場の部隊が作戦を実行する。ビジネスも同様に、経営陣は経営戦略を立てて、現場が実行する。軍隊もビジネスに携わる企業も組織の形態はヒエラルキーであったと言える。

1990年代以降に発生したアルカイダがアメリカ合衆国を標的として実行された数々のテロ行為をアルカイダとアメリカの戦争と見た場合、以前のような「国と国との戦い」とは根本的に異なる状態であった。アルカイダは国ではなく、組織のようだがトップが誰かはよく分からない。また、トップが作戦を立て現場が実行しているわけでもなく、アルカイダの思想に同調した人たちが同時多発的にテロを実行している。このようなアルカイダとの戦争のスタイルを呼ぶのにVUCAという言葉が生まれ、それに応じた新しい戦い方が必要になった。

ビジネスの現場においても、テクノロジーの進歩は急速であり予測は困難、世界の市場は不確実性や不透明性を増した状況となっており、不安定なビジネスの状況を表すのにVUCAが用いられるようになってきた。

システムを作る（設計する）

従来は、漏らさず可能性の全てを網羅する事を念頭に置いた。=> 要求開発
この発想はCOBOLに最適な設計手法

1959年に事務処理用に開発されたプログラミング言語。名前は「Common Business Oriented Language」（共通事務処理用言語）

IDENTIFICATION DIVISION.....見出し部

ENVIRONMENT DIVISION.....環境部

DATA DIVISION.....データ部

PROCEDURE DIVISION.....手続き部

VUCAの時代には、

Volatility（変動性）、Uncertainty（不確実性）、
Complexity（複雑性）、Ambiguity（曖昧性）

オブジェクト指向言語（SOA）とアジリティー

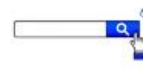
UI（ユーザーインターフェース）では無く、
UX（ユーザーエクスペリエンス）

UI/UXとは何か

UI

人とデジタルをつなぐ窓口

User Interface



- > 直ぐに分かる
- > 使いやすい
- > 迷わない など

UX

人とデジタルがつながることで得られる体験

User Experience

PASSWORD...



- > とても便利
- > もっと使いたい
- > 感動した など

UI

UX

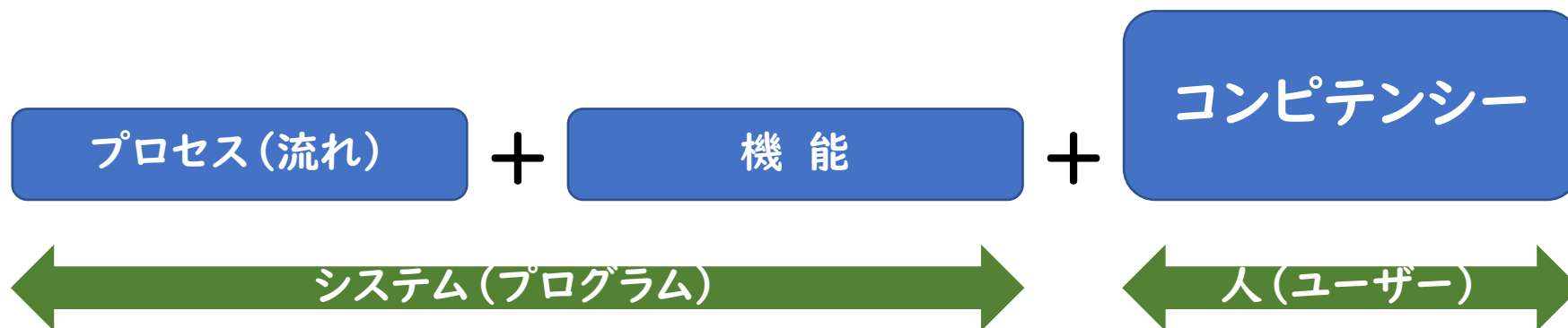
システムを作る。(昭和の話)。



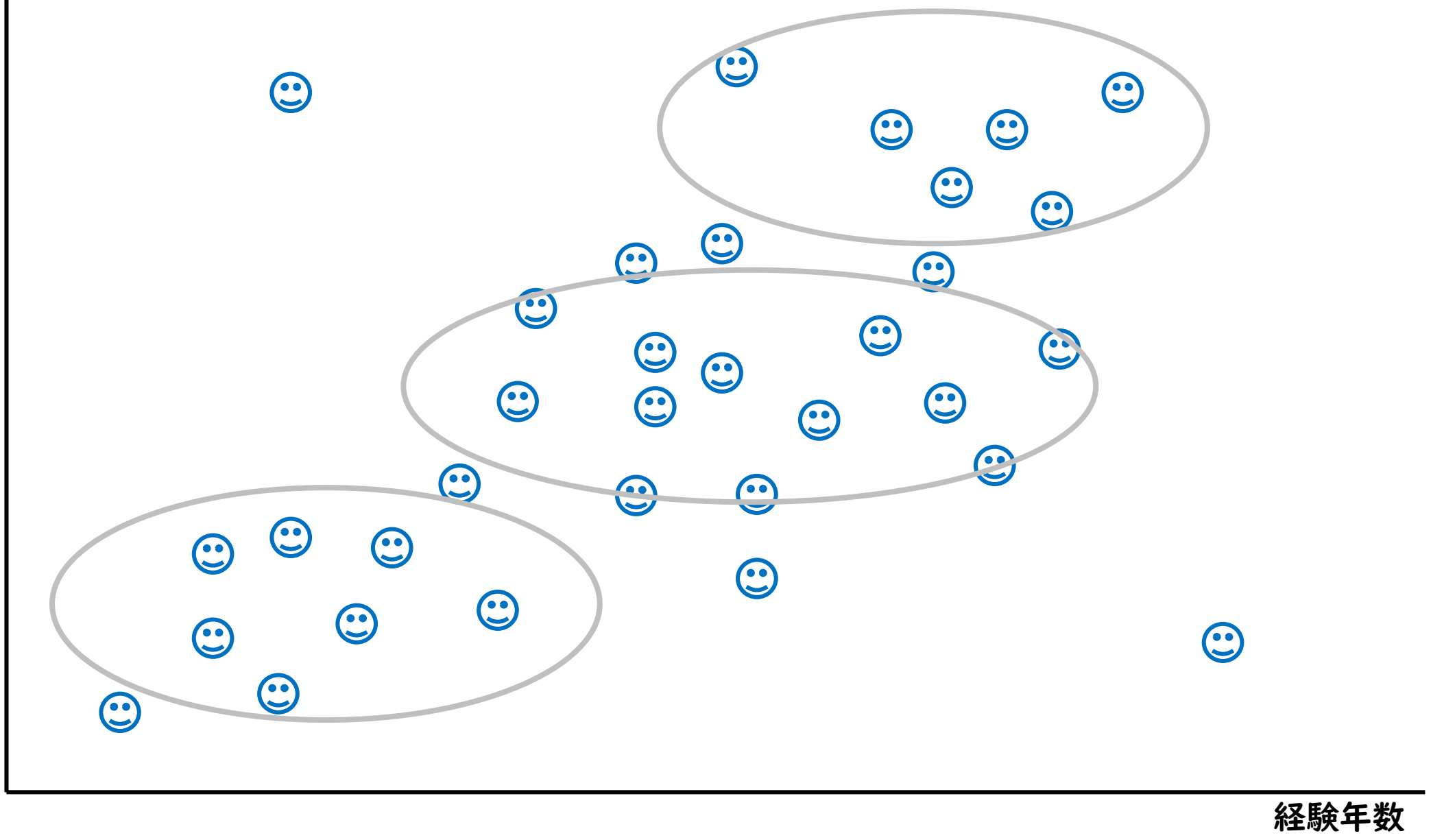
機能が単独で存在する事では無く、プロセス (流れ) の中に機能が存在する。

これが令和のIT !!

ITサービスを提供する (システムを上手く使う、使って貰う)。



社内システムと言えどもセグメント化が必要？



スピードが大きな競争優位性となる3つの理由

TPSを徹底研究

1. チャンスは長居してくれない

時代が激しく変化する現代において、ビジネスチャンスを掴み取れば、タイミングが重要になる。

2. 顧客ニーズの高速化

スピードが求められるのはビジネス機会のタイミングだけではない。顧客やニーズへの対応スピードも、その企業の価値を左右する大きなファクターとなってくる。ユーザーニーズに合わせ、プロダクトをどれだけの頻度でアップデートできるか、顧客の問い合わせにリアルタイムに対応できるかも顧客満足度を大きく左右する

3. 競合に対する防御策

決断と行動のスピードが速ければ、競合の動きにもうまく対応する事ができる。逆にその対応が遅れると致命的な結果にもなり得る。例えば、Netflixに駆逐されたビデオレンタルのBlockbusterや、Amazon対応が遅れたBordersなど、動きが遅すぎて、新規参入のライバルに対し完全に後手にまわってしまっただ企業も多い。

デジタルトランスフォーメーションにとって必須要件とは？

アジリティー（機敏性） = Speed + Margin

スピードを劇的に上げる為に、

- ◆ 組織をフラット化とチームプレー
- ◆ 管理の軽量化、権限委譲
- ◆ 作業品質の向上、自主管理
意志決定のスピード化を狙う。

余裕を生み出すには、

負荷の平準化とサステナブルな作業リズム

そのためには、自律した組織（チーム）・人材が求められる。

境界・限界を超え、主体的な、T型人材

デジタルトランスフォーメーションの事例

AAU's Texas Two-Step



e-Palette Concept

トヨタがアップルになる日

SoftBank TOYOTA

時価総額1位と2位が握手



CASE、百年に一度の大変革を迎えている。
自動車メーカーから脱皮
競争相手は同業者ではない。

モビリティカンパニー



リアルとデジタルの優位性

au

NTT docomo



SUZUKI



SUBARU
Confidence in Motion



DAIHATSU

ISUZU

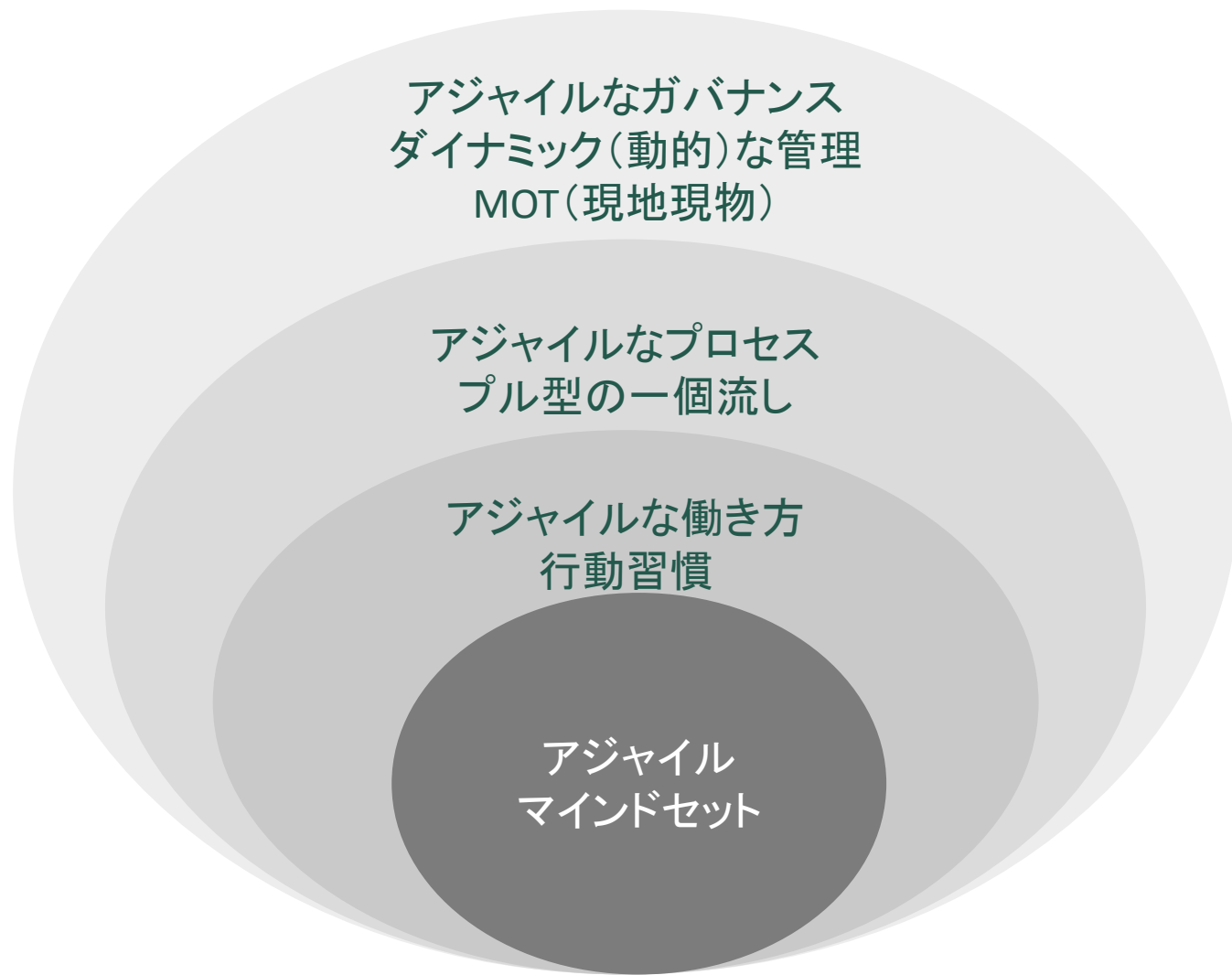
HONDA
The Power of Dreams

HINO

•MaaS クルマを所有するから移動のための道具へのパラダイムシフト

デジタルトランスフォーメーション (DX) 実践の基礎体力

筋トレを行わずして、早く走ることはできない



アジャイル・マインドセット

- タイムボックス・オペレーション (時間を意識)
- 見える化 (透明性、情報の共有)
- 品質を意識 (完了の定義)
- ダンドリ (プロセス) 重視
- 粒度と平準化 (適応力)
- 優先順位付け (本質を見抜く)
- 相互リスペクト (人間性尊重)
- 心理的な安全領域 (なんでも言える)
- チームで達成 (真の仲間)
- 先ずは行動に移す (失敗の共有)

アジャイルなプロセス

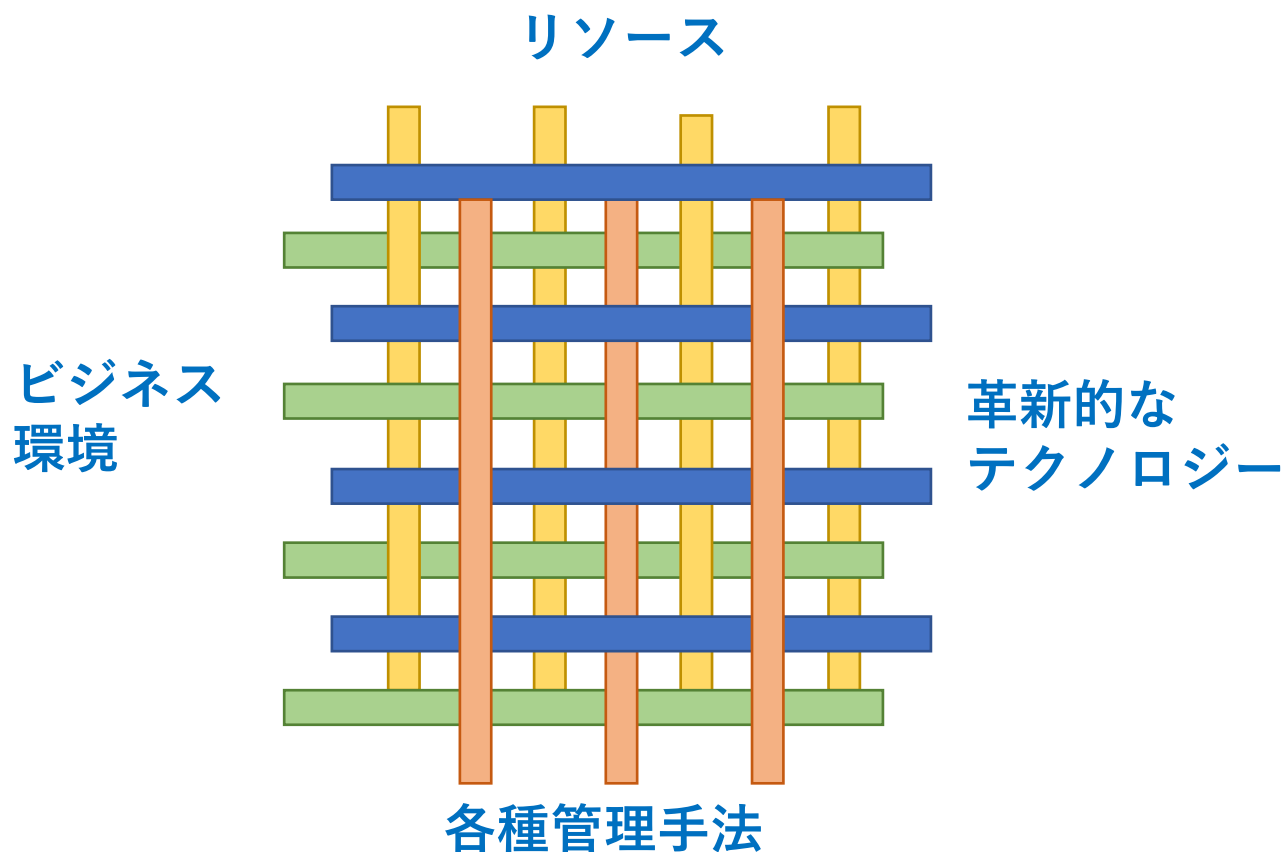
- プル型の一個流しのプロセスを設計する。
 - **品質不良、作業不良が即座にわかる仕組み**
- 全ての作業、プロセスはお客様に価値をジャストインタイム (JIT) で届ける為に設計される。
- 計画は毎日見直されて、その都度修正される。(毎日見直されなければ計画ではない。)
- 全てのプロセスに一切の滞留が無くなる様にカイゼンを続ける。(流水化のプロセスが究極のゴール)
- 当たり前品質を維持して、リードタイムを短縮することが、最大の関心事
- 自動化できるプロセスはすべて自動化する。(Automationでは無く、ニンベンのついた自動化)
- 作業の標準はすべて現場(作業員)が設定する。
- 管理者はこの流水化したプロセスの設計と維持が基本の責務(常に流れを見る)

アジャイルなガバナンス

- ・ダイナミック（動的）な管理へ
変革する。

現在スタティック（静的）な管理で使用されている各管理項目の元になるデータの発生源とプロセスを紐付け、ビジネスの目的、目標に基づいて見直し、必要最低限の管理項目に絞る。3ヶ月～半年に一回は管理項目の妥当性を見直し修正できる。（絶対に変えてはいけない管理項目（事業継続性）と環境変化に応じて変えなければならない（管理項目を明確化）

マネジメントメッシュの導入



ドキュメント (Shiryō)

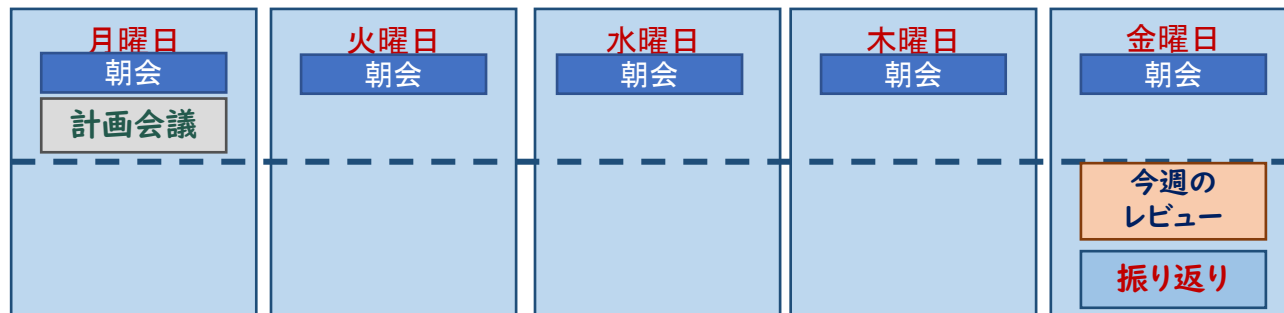
資料 (Source Material) = データ (必要な)
価値で測る。今日、価値がある。

紙料 (Paper Material) = レポート (診断書・カルテ)
定常的に作成されるドキュメント
重さで測る

死料 (Dead Material) = 報告書 (死亡診断書)
生きている人には意味がない。
歴史。

アジャイルな働き方（行動習慣）

- ・ タイムボックスを適用した一週間の基本的な仕事のパターン：月曜日から金曜日までのタイムボックスが設計されている。【基本的に残業、休日出勤は無い。すべての仕事は通常勤務時間内で完結する】



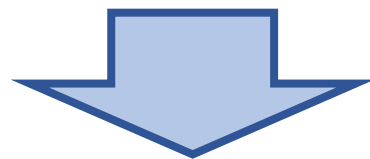
- ・ タイムボックスは、仕事の進捗も確認できる様に設定：1時間、1日（5時間）、1週間（4日間；20時間）
- ・ ダンドリ重視：あらゆる遣るべき仕事を週初めに全て60分程度の細かなタスクに分解して見える化している。
- ・ 作業の完了基準を予め設定し、その検査のためのゲージ（簡易測定ツール）を自作している。
- ・ 品質的な不良が確認されれば即座に全ての作業を中断してまず品質を確保するための修復作業に全員が携わる
- ・ あらゆる作業にはすべて優先順位（順番）が付けられている。またいつでもこの優先順位は見直されて修正される。
- ・ 作業対象となるモノの本質を捉えて、シンプル（単純化）に考え、行動する。
- ・ 同時に二つの事（マルチタスク）は行わない。一つずつ仕事は優先順位に基づいて実行される。
- ・ チームで行動、チームの連帯&連携、本音の会話（コミュニケーション）
- ・ 失敗を共有し、直に振り返る

サービスと言う考え方

ソフトウェア工学において、サービス指向アーキテクチャ(Service-Oriented Architecture、SOA、1990年発表)とは、大規模なコンピュータ・システムを構築する際の概念あるいは手法の一つ。業務上の一処理に相当するソフトウェアの機能をサービスと見立て、そのサービスをネットワーク上で連携させてシステムの全体を構築していくことを指す言葉である。業務処理の変化をシステムの変更に素早く反映させたいという需要に応えうるものとして、2004年頃からIT業界において注目を集めている。2009年頃からクラウドコンピューティングの台頭とともに、その必要性が再認識されるようになってきている。

決められたインターフェースに従ってソフトウェアの一部をカプセル化、部品化し、それを組み合わせて全体を構成するという考え方
分散オブジェクト、メッセージング、EAI (Enterprise Application Integration) などの技術を使用し、ネットワークを介してソフトウェアを連携(疎結合)させる

オブジェクト指向、コンポーネント指向
SOAPプロトコル (RESTインターフェース)
疎結合



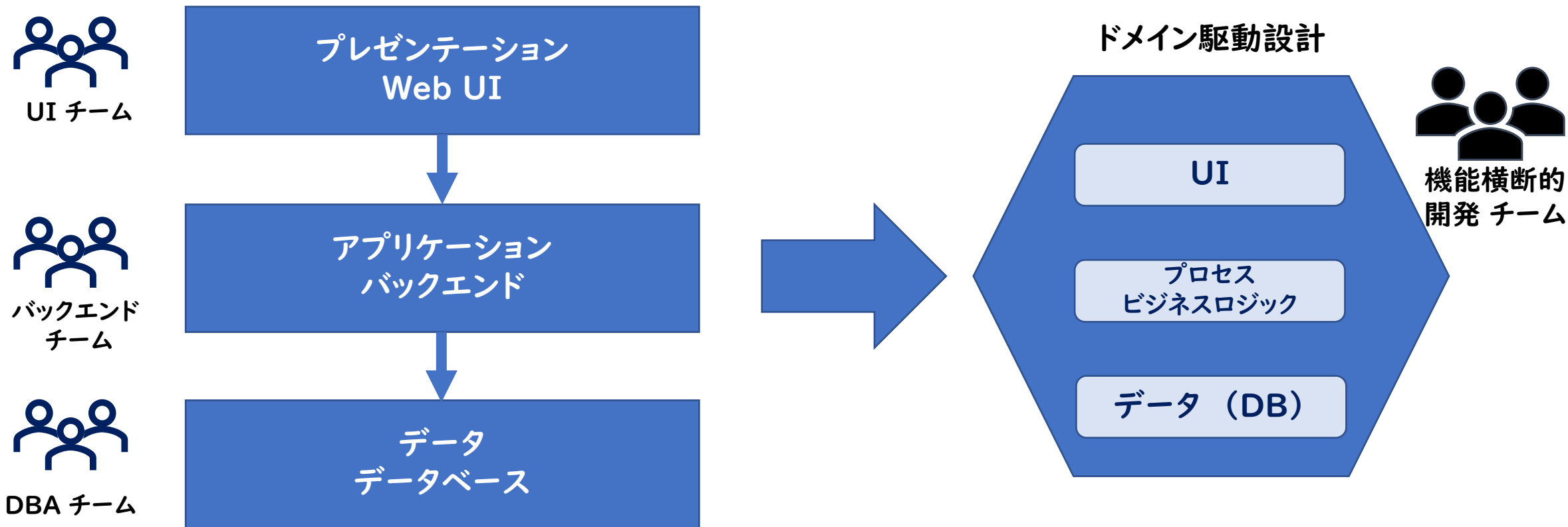
アジャイル開発

マイクロサービスアーキテクチャとは、

マイクロサービスとは、ThoughtWorks社のマーチン・ファウラーとジェームス・ルイスが提唱したソフトウェアアーキテクチャで、モノリシック（一枚岩）なアーキテクチャとは異なり、ビジネス機能に沿った複数の小さい「マイクロサービス」という部品を疎結合してシステムを構成する。

- ❑ マイクロサービスはSOA(サービス指向アーキテクチャ)のひとつの実現形態であり、DDD(ドメイン駆動設計)、CI・CD(継続的インテグレーション・デリバリ)、インフラ仮想化、自動化、アジャイル開発プロセス、といったさまざまな分野の技術や方法論を組み合わせることで成り立っており、これらが総合的に実践できなければその価値を享受できない。
- ❑ DDDの「境界付けられたコンテキスト」が、ひとつのマイクロサービスの単位になる。
- ❑ 参考「DDDの構成要素とマイクロサービスの単位をどう合わせるべきか」
- ❑ マイクロサービスは、外部から利用されるためのAPIを公開する。その公開APIの呼び出し元(=コンシューマ)に影響を与えることなく単独で変更・本番リリース可能である。
- ❑ あるマイクロサービスが停止しても、他のマイクロサービスは停止せずにサービスを提供し続け、システム全体としては、一部機能を切り離して稼働し続けることができる。
- ❑ マイクロサービスは、必要なサービスだけを、必要なときにスケールさせることができる。
- ❑ マイクロサービスは、自律的なチームによって開発され、所有・維持される。採用する技術や方法論は所有チームが自由に決定する。
- ❑ マイクロサービスは、簡単に捨てたり、作り直したりできる。(2週間で作り直せる程度の大きさ)

マイクロサービスによる設計のトランスフォーメーション



システム的设计 (アーキテクチャ) が開発組織の構造に影響を受ける。

自分のデータを内包する。
データベースの共有を避ける。

プログラムの特性

一から十まで、手順を詳細に記述する。プログラムの設計は経験的にパターン化できる。
プログラムの品質は、使用頻度×期間の大きさで向上（洗練）する。

デザインパターン：

GoF (Gang of Four; 四人組) と呼ばれる4人の共著者は、書籍『オブジェクト指向における再利用のためのデザインパターン』の中でデザインパターンという用語を初めてソフトウェア開発に導入した。GoFは、**エーリヒ・ガンマ**、リチャード・ヘルム、ラルフ・ジョンソン、ジョン・ブリシディースの4人である。彼らは、その書籍の中で23種類のパターンを取り上げた。

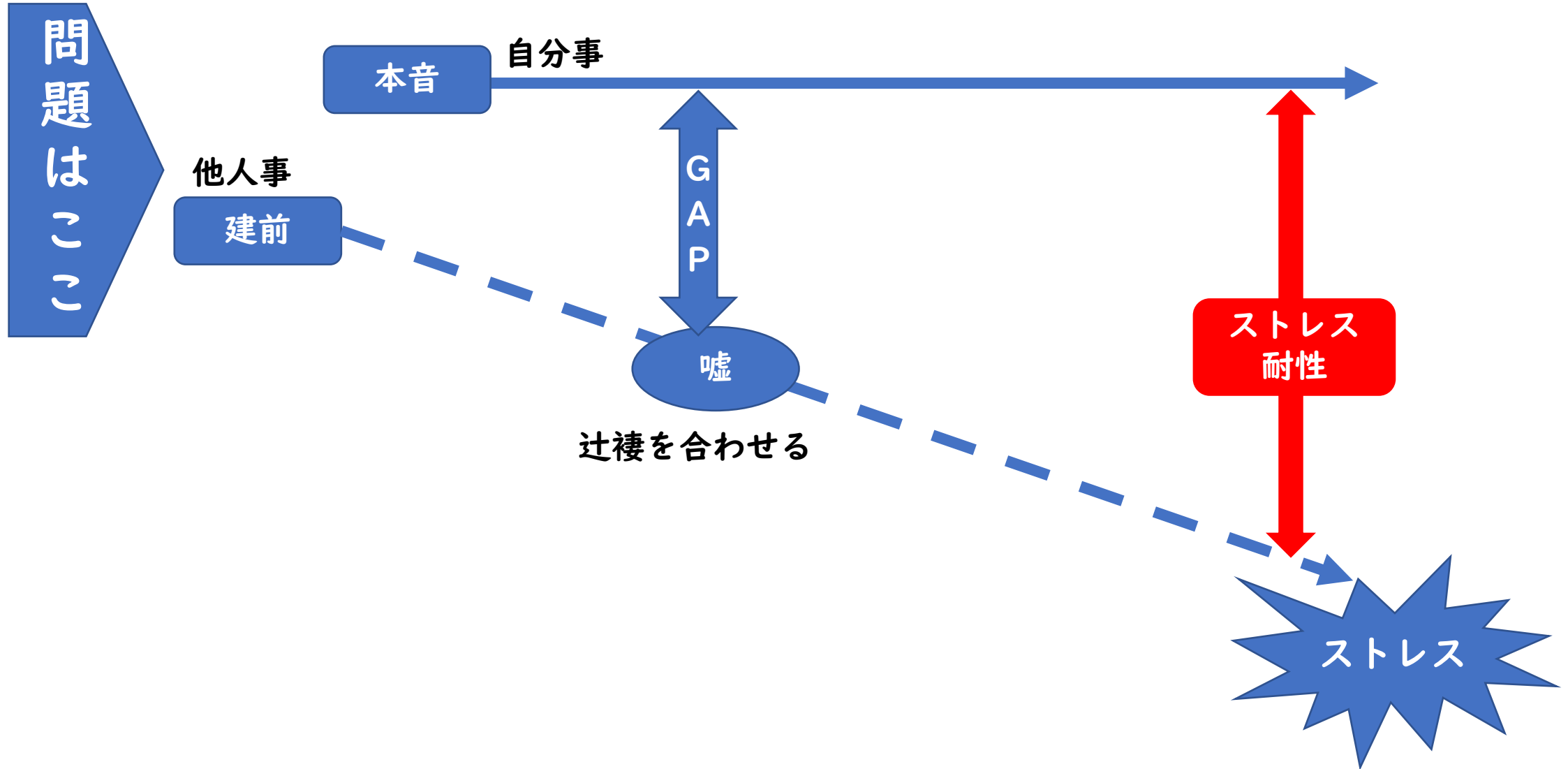
- 生成に関するパターン (5パターン)
- 構造に関するパターン (7パターン)
- 振る舞いに関するパターン (11パターン)
- マルチスレッドプログラミングに関するパターン (14パターン)
- アーキテクチャに関するパターン (12パターン)
- その他のパターン (7パターン)

クリストファー・アレグザンダー
グラディ・ブーチ
ケント・ベック
ウォード・カニンガム
マーティン・ファウラー
ロバート・セシル・マーティン
ジム・コプリエン
ダグラス・C. シュミット
リンダ・ライジング

アジャイルの基本

- 小さなチーム（仲間）
- 共同所有（全員が全て知っている）
- タイムボックス=イテレーション（時間を区切って、時間内で完了できる工夫をする）
- シンプルな設計（たればの排除）
- 素早く作る（可能な限り自動化）
- 小さく作る（一日の作業範囲で）
- 持続可能なペース（息切れしないで限りなく早く）
- 品質重視（譲らない）
- 現地現物（今起きていることが全て）
- 本質の追求（なぜ?なぜ? 表出する結果に惑わされない）
- 失敗を恐れない（論より証拠 失敗は成功の一ステップ 失敗を隠さない）
- ネバーエンディング（ゴールが見えない）

プラス 心理的安全性



心理的安全性とは

心理的安全性とは、英語の「psychological safety」を和訳した心理学用語で、主にビジネスの場で使われる言葉です。直接的な意味は、心理的に安心や安全を覚える状況や環境を指します。具体的にビジネスの場で使われる際の意味としては、周囲の反応に怯えたり、羞恥心を感じることなく、自分自身が自然な状態でいられる環境のことを指します。人が自分らしさをさらけ出す環境作りをするためには、安心感や共感のあるオープンな場であることが大切です。Google社は2012年から約4年間、効率的なチームを構成するための「プロジェクトアリストテレス」という大規模な労働改革を実施しました。そこで発表されたのが、チームの生産性を高めるために心理的安全性が重要な要素であるということです。

組織の中で自分の考えや気持ちを誰に対しても安心して発言できる状態のことです。

組織行動学を研究するエドモンドソンが1999年に提唱した心理学用語で、「チームの他のメンバーが自分の発言を拒絶したり、罰したりしないと確信できる状態」と定義しています。メンバー同士の関係性で「このチーム内では、メンバーの発言や指摘によって人間関係の悪化を招くことがないという安心感が共有されている」

効果は？

チームメンバーのパフォーマンス向上（のめり込んで仕事をしている状態ではドーパミンが分泌され、仕事に対するストレスを感じづらくなる）

イノベーションや改善が生まれる

離職率が低くなる（個々のモチベーションがアップ）

目指すビジョンが明確になる（共通の目標）

チームや部署の心理的安全性が高いかどうかを測定するために効果的な質問

1. チームの中でミスをする、非難される
2. チームのメンバー間で、難しい問題や浮上している課題について指摘し合える
3. チームのメンバー間で、「自分とはどこか違う」ことを理由に他者を拒絶したり非難することがある
4. チームに対してリスクな対応をとっても非難されることはない
5. チームのメンバーに助けを求められない
6. チームのメンバーの中に、自分の仕事をおとしめる行動を取る人はいない
7. チームで仕事をする際に、自分のスキルや知識を尊重してもらえ、活躍していると感じる

テクノロジーとアジャイル

アジャイル開発をより強力にするテクノロジー&メソドロジー

デザインパターン

☆マイクロサービスアーキテクチャー

データベース・リファクタリング

クラウド&コンテナ

☆サーバーレス

何故、少人数のチームが有効か？



3 people, 3 lines



4 people, 6 lines



5 people, 10 lines



6 people, 15 lines



7 people, 21 lines



8 people, 28 lines



9 people, 36 lines



10 people, 45 lines



11 people, 55 lines



12 people, 66 lines



13 people, 78 lines



14 people, 91 lines

United States Navy SEALs

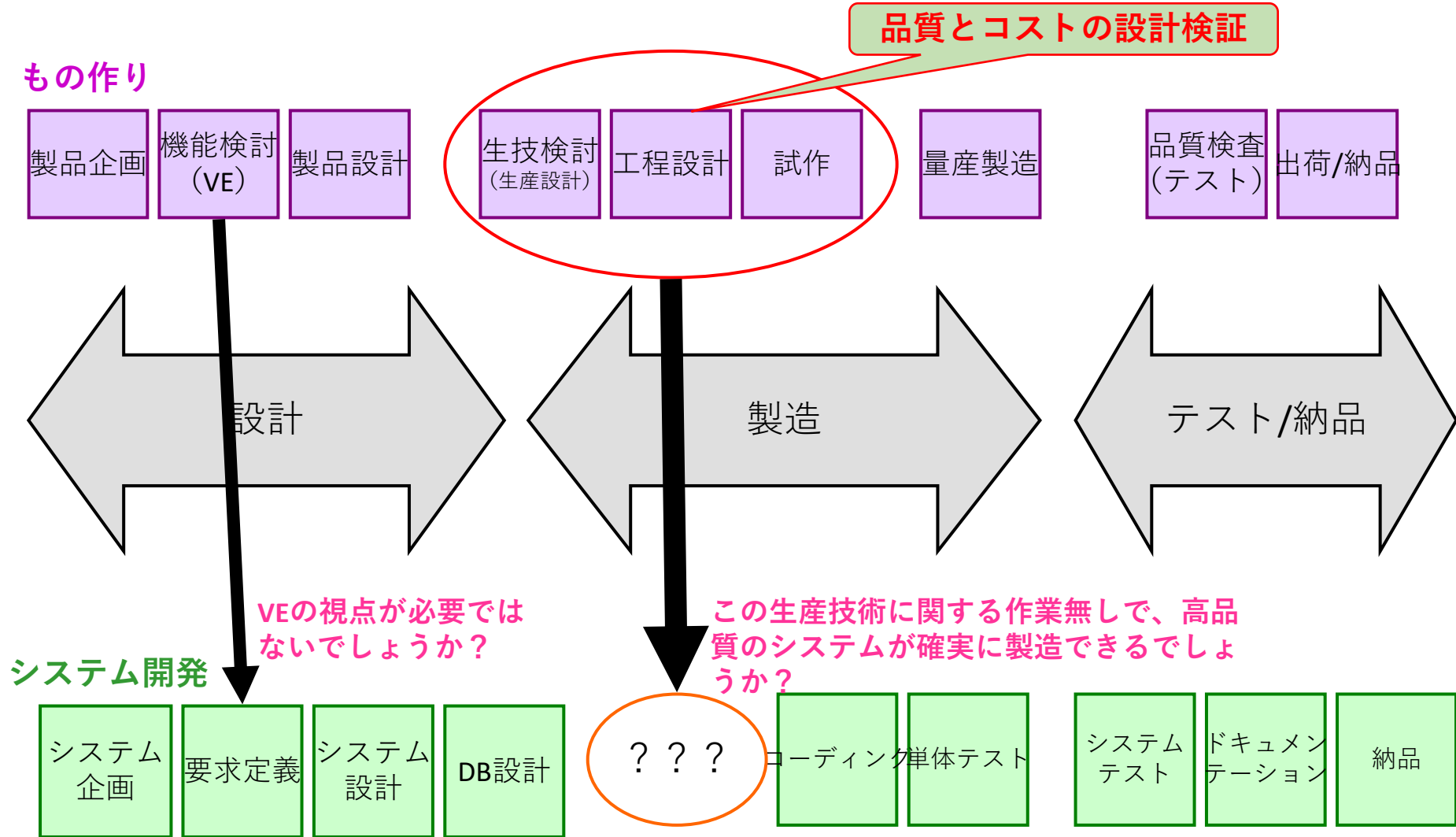


任務は通常2名から4名で行動する

(場合によっては2の倍数で増員される) 小隊は14人～16人。士官2人、兵曹長・先任下士官各1人、下士官・水兵10人～12人の編成とされる。

隊員はT型人才

『もの作り』と『システム作り』の相違

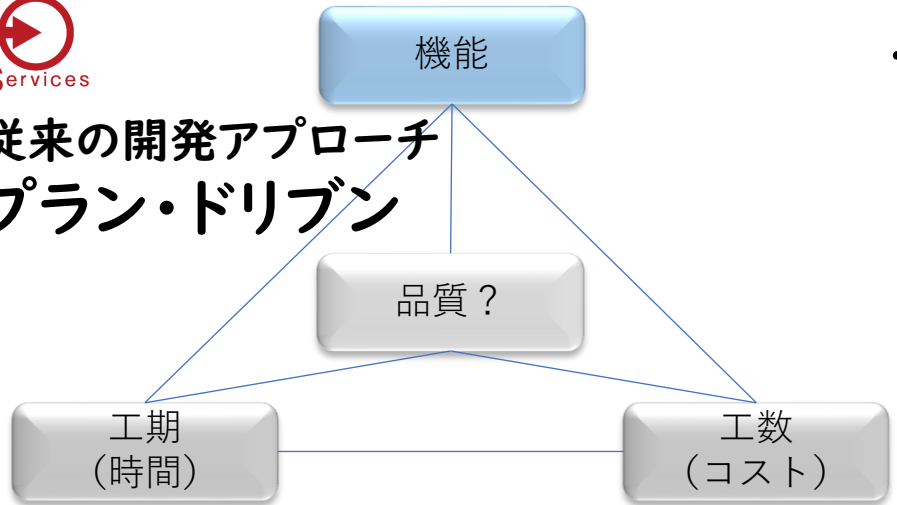


設計仕様を図面上のみの検討で、十分でしょうか？

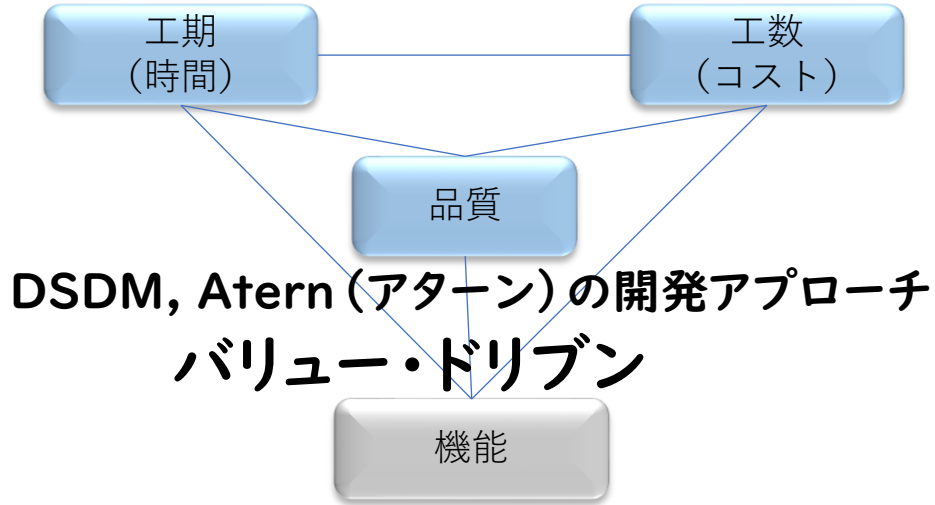
擦り合わせ、微調整が必要ではありませんか？（誰が、何時、どの様に作業できますか？）

アジャイル開発の基本的な価値観

従来の開発アプローチ
プラン・ドリブン

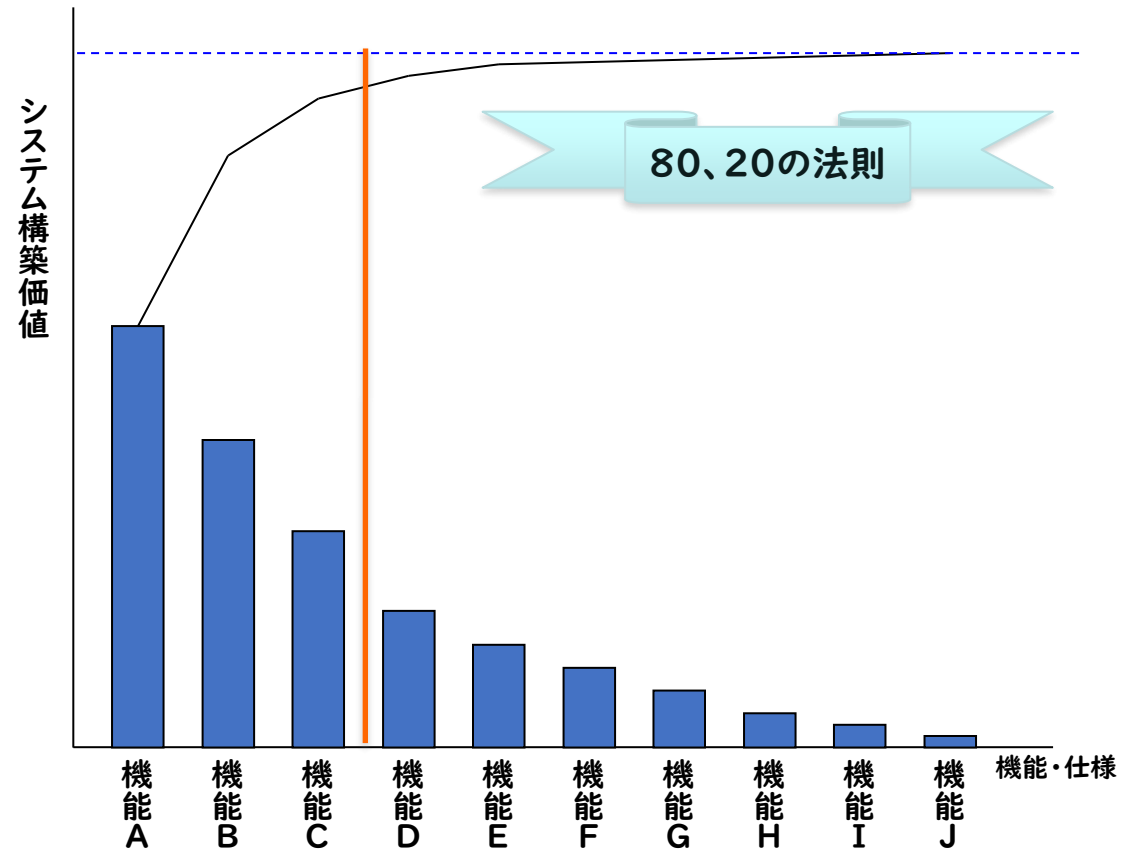


変える



DSDM, Atern (アターン) の開発アプローチ
バリュー・ドリブン

J. Juranのパレート図を用いたプロダクト・バックログ管理

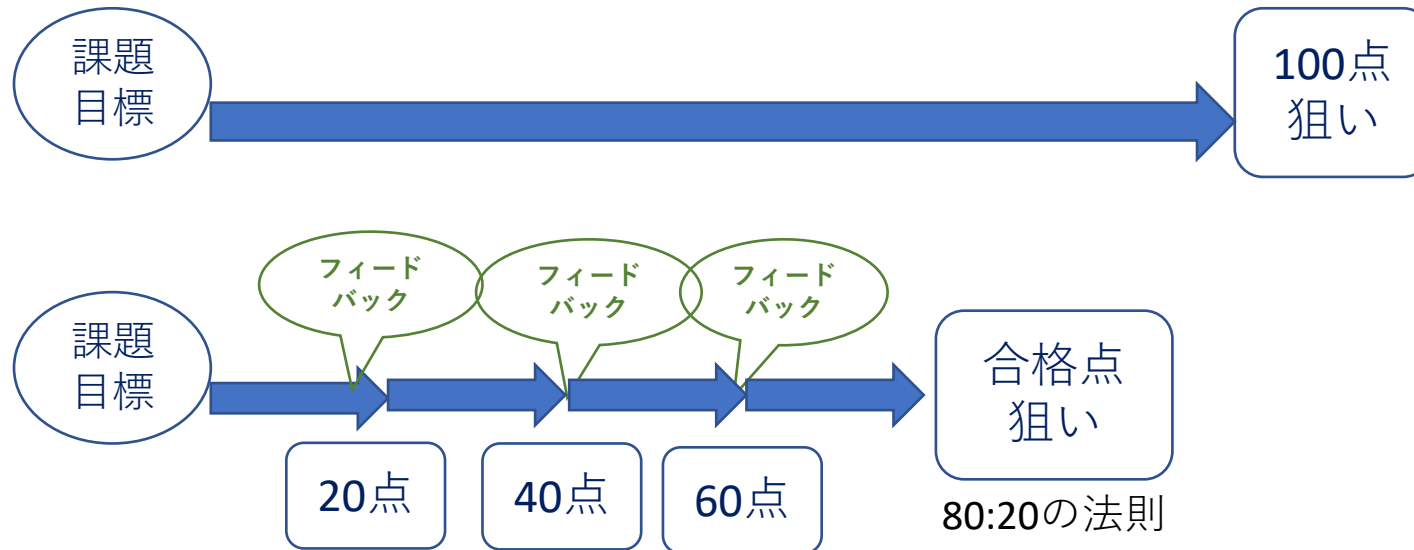


アジャイル開発の仕事の基本構造

イテレーション(スプリント)

- 何故、反復的に開発を行った方が良いのか？
- 何故、そうしなければならないのか？

- ✓ 時間の使い方 (タイムボックス)
- ✓ 透明性
- ✓ 品質



現代手法と旧手法の違い

デジタルトランスフォーメーション(DX)
DevOps
アジャイル開発

目的(対象):ITサービス(ビジネス)
大事な価値観:スピード、機敏性、適応力
ファクト・コントロール

要求は定義できない。ころころ変わる。
計画は立てられない。適応力が全て。
完全な自律型運営（現場に主導権）
管理者の役割：ファシリテーション、支援役
人事評価；チーム、グループでの業績管理

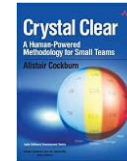
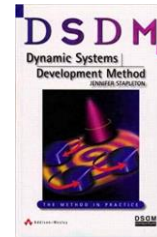
ウォーターフォール開発

目的(対象):プロジェクト
大事な価値観:計画の確実な遂行
企画、計画

要求の定義
周到な計画
強力な管理機構（スタッフ機能の増大）
管理者の役割：管理監督、監視
人事評価：個人別目標管理

アジャイル開発手法

- ダイナミックシステムズ開発技法 (Dynamic Systems Development Method)
 - ✓ デイン・フォルナー (Dane Faulkner) ほか
- アダプティブソフトウェア開発 (Adaptive Software Development)
 - ✓ ジム・ハイスミス (Jim Highsmith)
- クリスタルメソッド (Crystal Methods)
 - ✓ アリスター・コックバーン (Alistair Cockburn)
- スクラム(Scrum)
 - ✓ ケン・シュエイバー (Ken Schwaber)
 - ✓ ジェフ・サザーランド (Jeff Sutherland)
- XP (エクストリームプログラミング)
 - ✓ ケント・ベック (Kent Beck)
 - ✓ エリック・ガンマ (Eric Gamma) ほか
- リーンソフトウェア開発 (Lean Software Development)
 - ✓ トム/メアリー・ポップエンディーク (Tom and Mary Poppendieck)
- フィーチャ駆動開発 (Feature-Driven Development)
 - ✓ ピーター・コード (Peter Code)
 - ✓ ジェフ・デルーカ (Jeff DeLuca)
- アジャイル統一プロセス (Agile Unified Process)
 - ✓ スコット・アンブラー (Scott Ambler)



スクラム (Scrum) とは

スクラム (Scrum) は、軽量なアジャイルプロジェクト管理メソッドであり、小規模で自由裁量の自己組織的なチーム、完全な可視性、および敏速な適応力が特徴である

- 小さく、機能横断的なチームが緊密に協力し、オープンな環境でインクリメンタルな製品リリースを反復的に行う
- チームはそれぞれの反復 (スプリント) の目的を満たすことに対して、自ら方向を定め、自己裁量権を持つ
- チームの活動はスクラムマスターによってファシリテーションされる
- 活動はプロダクトバックログによって組織化される

スクラムの価値基準

- **透明性 (Transparency)** ↔
見える化、共通認識、共通理解 (共有)

- **検査 (Inspect)** ↔
望ましくない変化を検知、自身の作業結果
(良質な作業)、作業標準は現場で作成

- **調整・適応 (Adapt)** ↔
変化への対応、ビジネスを支援 (お客様第一)

スクラムの基本 行動原則

TPSの概念

見える化、大部屋

自工程完結

平準化、プル型一個流し



1. Boundary (境界=時間Time Box)

- スクラムにおいてタイムボックス化は非常に重要である
- 反復 (スプリント) やミーティング (スタンドアップミーティング、スプリント計画ミーティング、振り返り、スプリントレビュー) など、スクラムのイベントはすべてあらかじめ規定された期間 (タイムボックス) 内に完了する
- タイムボックスは開発チームのリズムを作る

2. Goal (共有された目標)

- チームメンバーは目標を共有し、自ら方向を定め、自らの日々の仕事を管理、改善する

3. Fixed Team (固定されたチームメンバー)

- チームメンバーは専任かつ固定化されたほうが良い
- 10人以内のチームメンバー (一人多役)

(参考) アジャイル開発におけるタイムボックスの価値 = やる気と集中力

『仕事の量は、完成の為に与えられた時間を全て使い切るまで膨張する』

イギリスの歴史学者・経済学者であるパーキンソンの言葉

時間には弾力性がある。

時間は、何となく使ったのではいくら有っても足りない。

同じ仕事量でも、意識の違いでかかる時間は全く異なる。

スピードを上げるほど、脳は活性化する。

集中していればミスは少ない。

時間を計ればムダに気づく事ができる。

集中力は長く続かない。休む事で充電される。

時間が読めるからリラックスできる。

生産性はやる気と集中力で高まる。

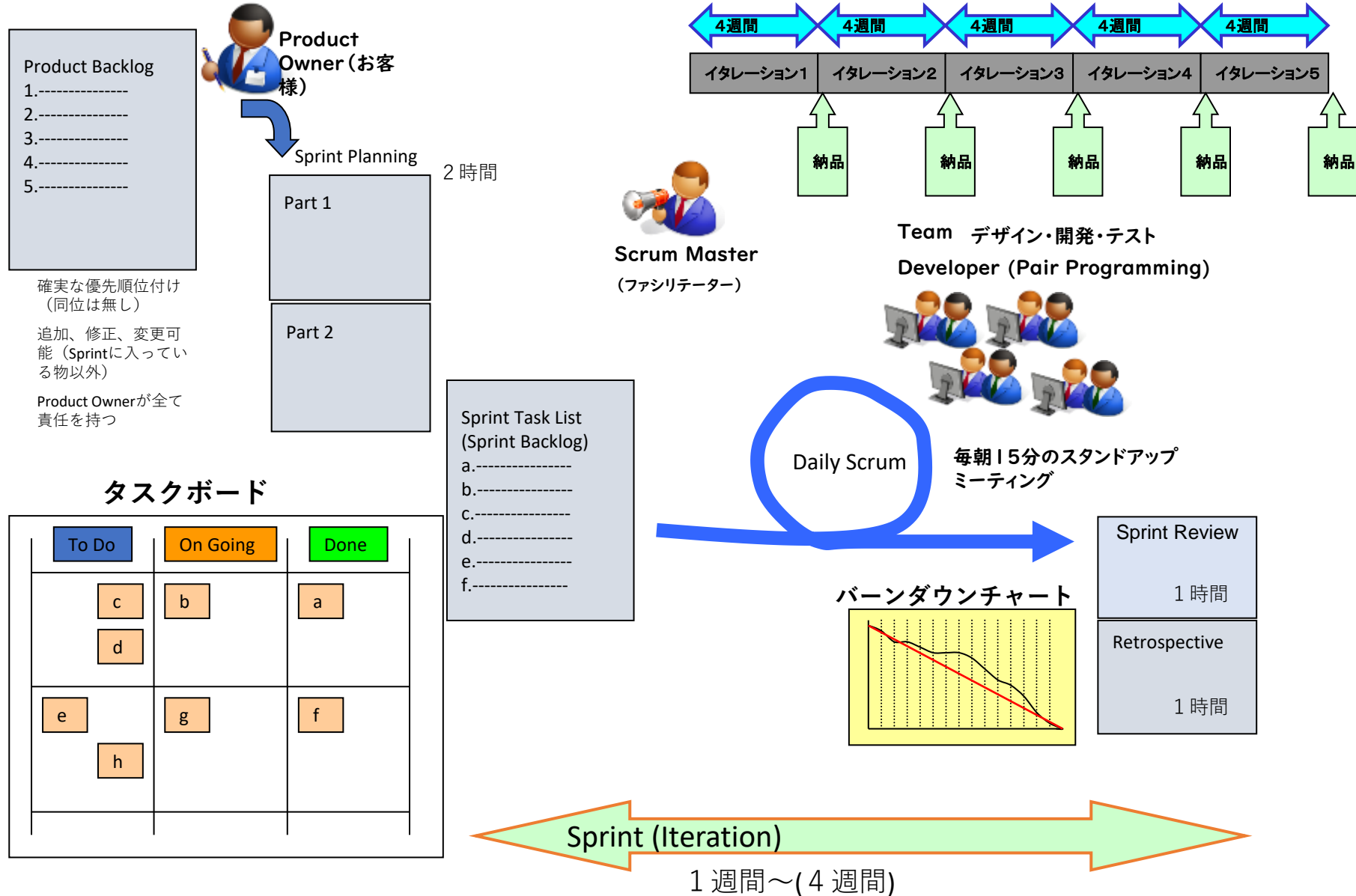
やる気のホルモン = ドーパミン

ドーパミンは、ご褒美によって放出される。

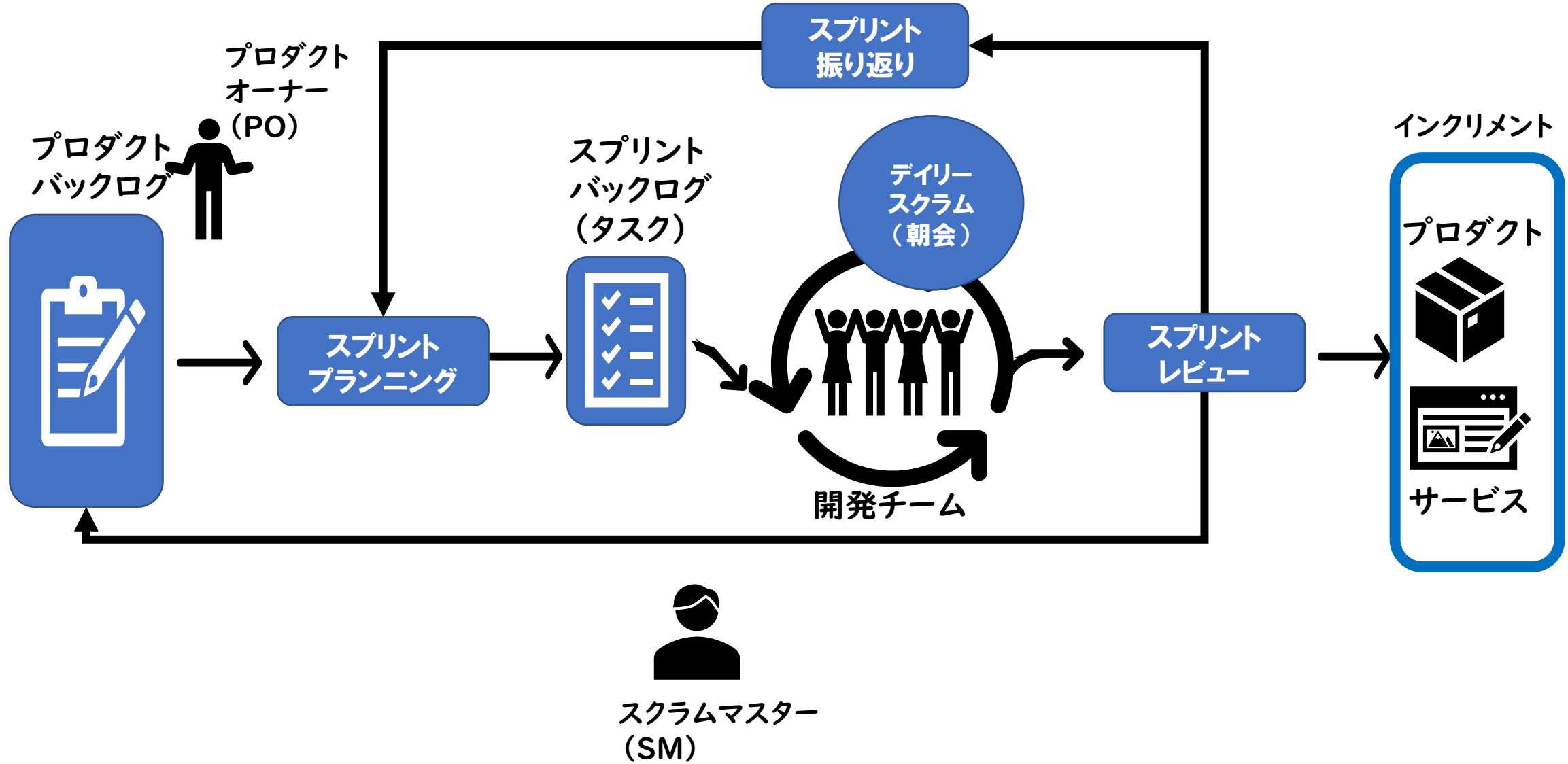
やる気はご褒美の事を考えるだけで出る。しかし、裏切られると一瞬で低下する。

ご褒美の60秒ルール = ご褒美は直ぐに貰える事が重要。楽しく想像できる事が重要。

スクラムの登場人物とプロセス

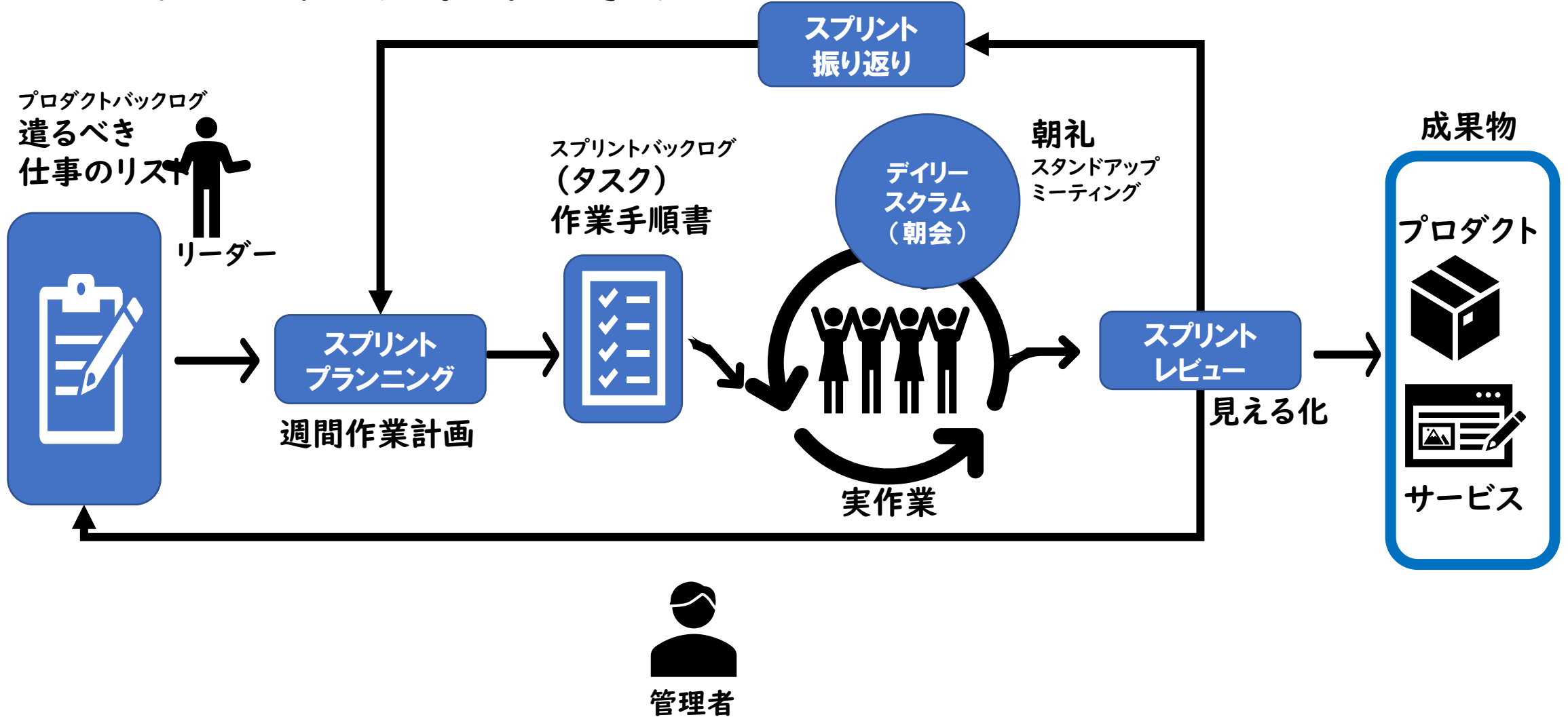


スクラム フレームワーク

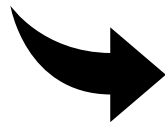


スクラム フレームワーク を開発以外の仕事場に適用

一週間の仕事だけでなく
働き方、仲間(チーム)、環境などを



スクラムの1週間



月	火	水	木	金
スタンドアップ	スタンドアップ	スタンドアップ	スタンドアップ	スタンドアップ
計画会議	To Do Doing Done			
タスク分解	タスクボード			
				振り返り KPT

見える化（透明性の担保）のアイデア

プロダクトバックログリスト

タスクボード

バーンダウンチャート

振り返り（KPT）ボード

チームの規約（作業標準）

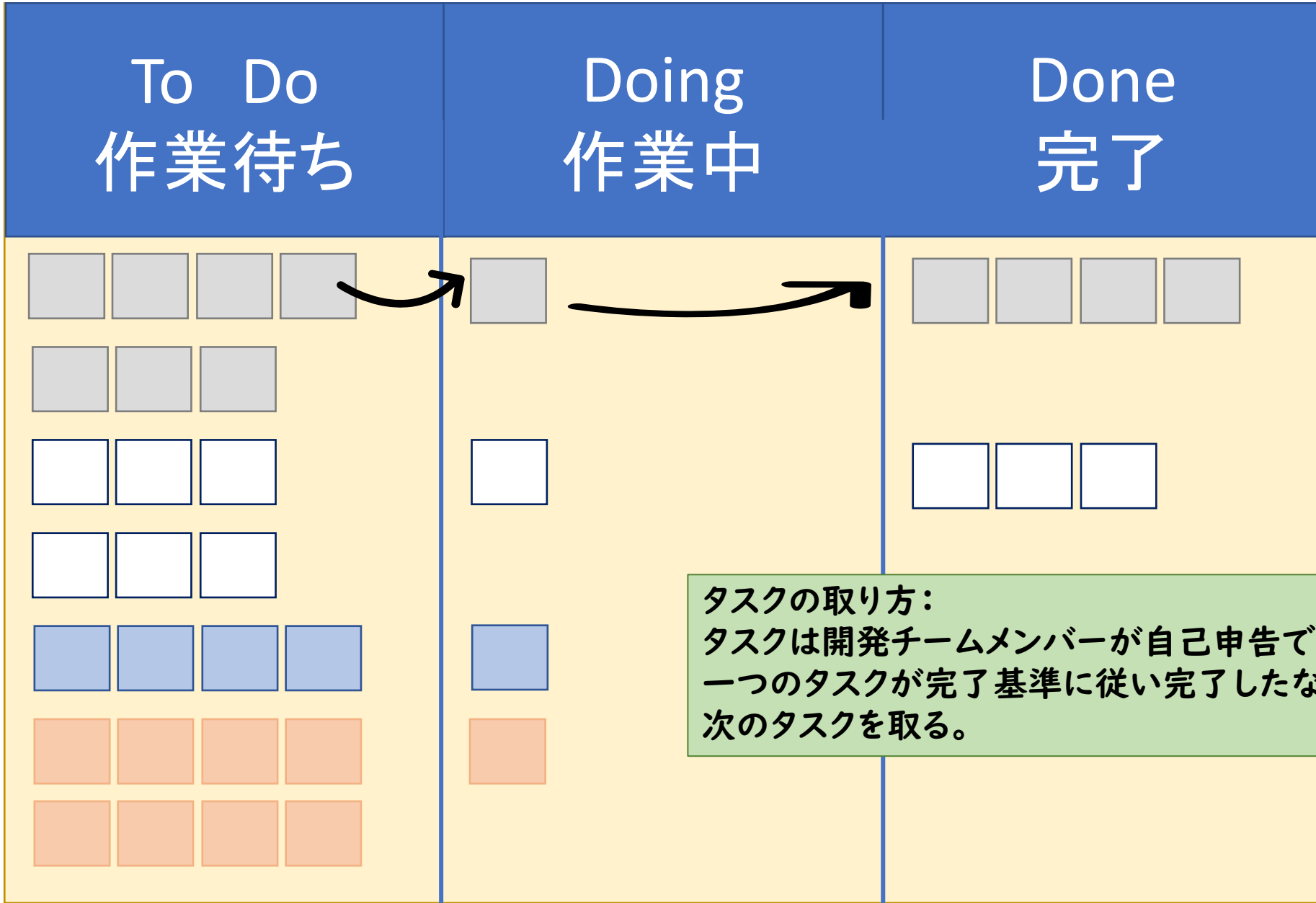
チームの目標（スプリントゴール）

+

チームのアイデアで自由に作成

- お助けボード
- 不安ボード
- ニコニコカレンダー
- 連絡ボード など

タスクボードの運用



タスクチケット

ID : 001		
Date:2/01/2019		
タスクの記述 (作業内容)		
見積り 時間	作業者名	実績 時間

タスクの取り方:
 タスクは開発チームメンバーが自己申告で取る。
 一つのタスクが完了基準に従い完了したならば、
 次のタスクを取る。

アジャイルを支えるプラクティス

ビジネスプラクティス

計画それ自体よりも計画作り(計画ゲーム)
ベロシティー
フィードバック
小さな(頻繁な)リリース
受け入れテスト
チーム全体

チームプラクティス

真のチーム(仲間)
同じ場所、同じ仲間
持続可能なペース
共同所有
継続的インテグレーション
スタンドアップミーティング
振り返り

テクニカルプラクティス

シンプルな設計
テスト駆動開発(TDD)
ペアプログラミング
リファクタリング
ACDM

アジャイル・マインドセット

信念、信条、価値観、共通の目的

RFPによる一般入札案件（基幹業務系システム刷新）の事例

業種： 全国展開の建築（リハウス）＆保守メンテ、建物販売
売上：28 億 8600 万（16 期実績） 29 億 3000 万（17 期実績）
従業員数：421 名（2011 年 9 月現在）

提案の範囲

今回ご提案をお願いするシステムの全体概要および調達するシステムのスコープは以下の通りです。
情報システムの「設計（基本設計、詳細設計、移行設計、運用設計）」、「開発・調達」、「テスト及び移行」、「初期サポート」、「運用保守への引継ぎ」に関する具体的な実現方法をご提案下さい。

- 現行業務システム実装済機能：263
- 新規システム要求機能：197
- RFP記載の機能要求明細：262

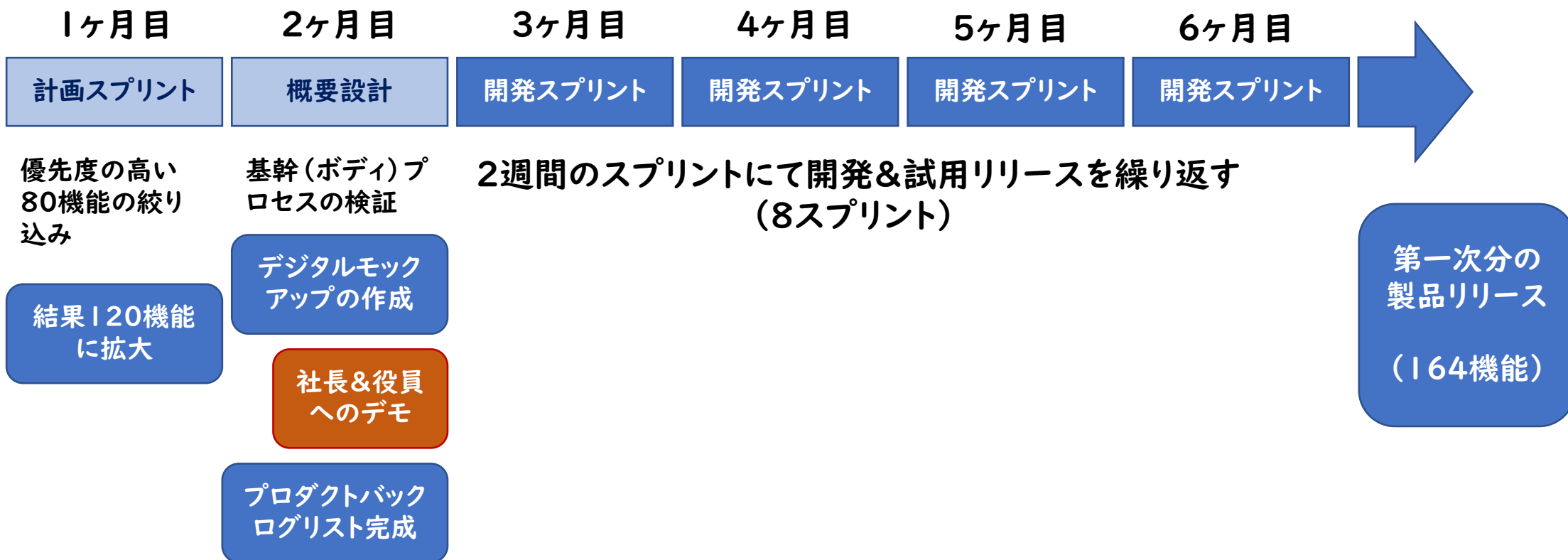
WFで見積もると
開発期間：3年間
金額：3億円～4億円

提案概要

期間：6ヶ月間のスクラムで80機能の実装をコミット
（開発：2週間のスプリント、8スプリント）
金額：7,000万円
開発チーム：8名 完全リモート開発（東京ー大垣）
進捗会議；透明性（24時間×7日）を担保するので、
月次進捗会議、ドキュメントは実施しない。

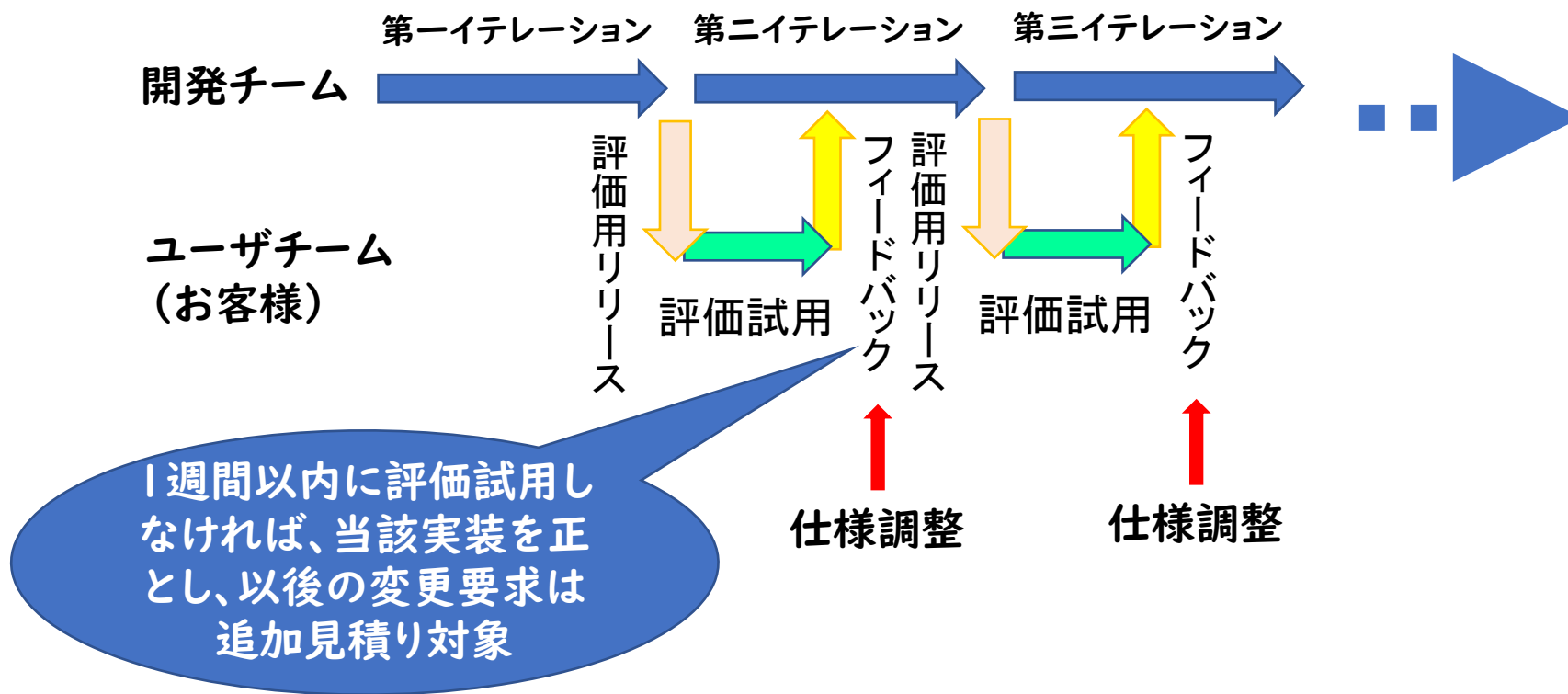
実装実績：164 機能
変更要求対応：412 件

スクラムプロジェクトプラン



スプリント サイクル

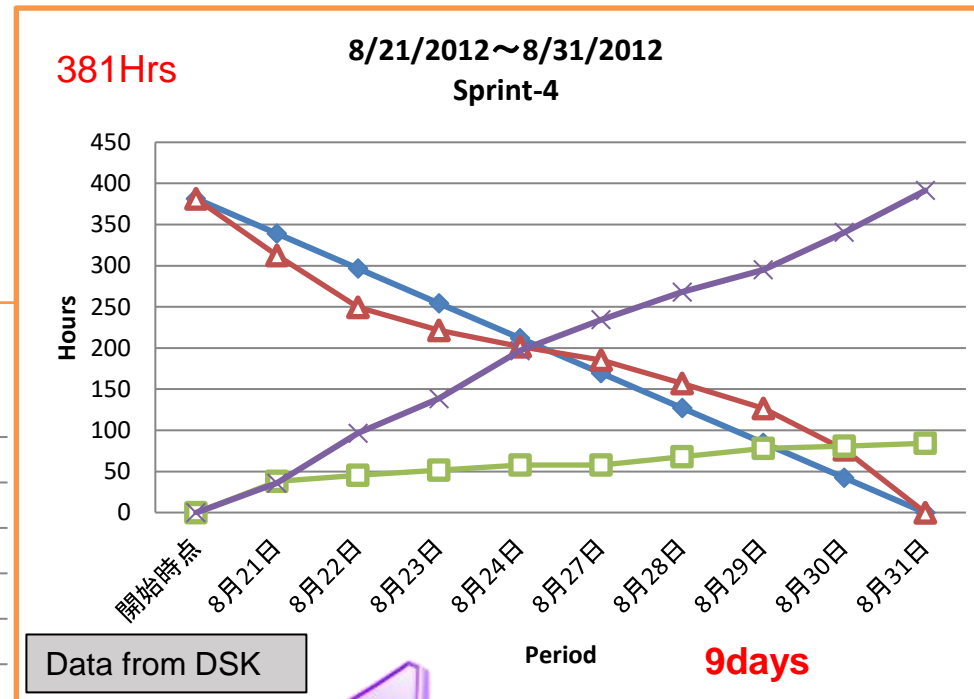
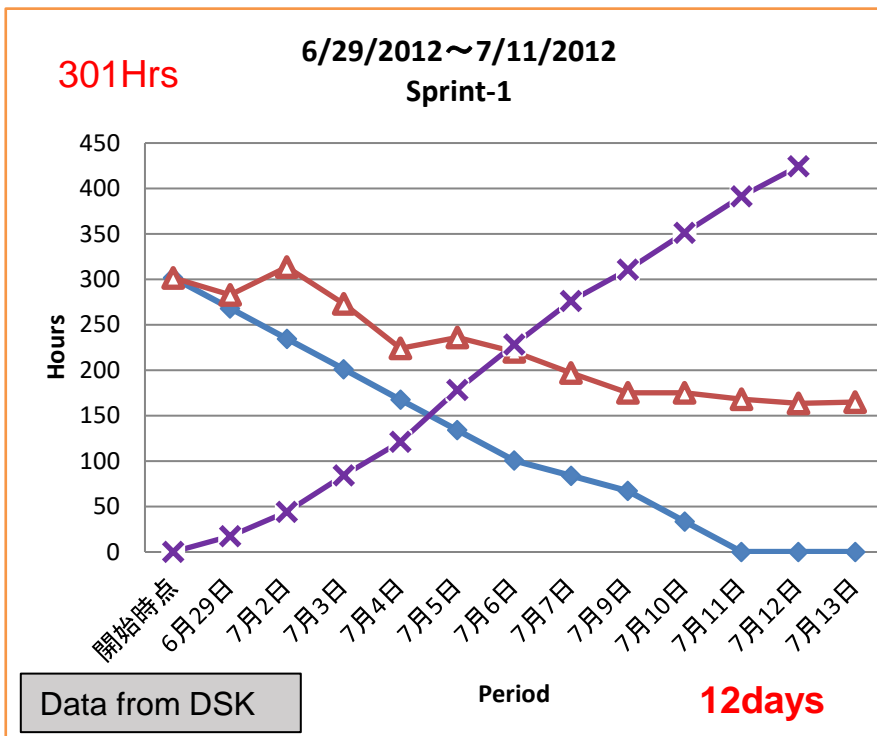
- スプリント期間は2週間サイクル (お客様の特質を見て判断)
- スプリント途中に直近のリリース結果のレビュー会を開催し、その結果を実施中のスプリントへ可能な限り反映する
(スクラムでは禁止されているが、スプリント途中でのタスク変更要求を受け付ける事)



チームの成長（カイゼン）

同一プロジェクト、同一のチーム、同一の手法（プラクティス）で実施しても
何故、生産性が変わるのか？

- △ △ △ : Actual
- ◆ ◆ ◆ : Plan
- × × × : Actual (Accumulation)
- □ □ : Correspond to Feedback



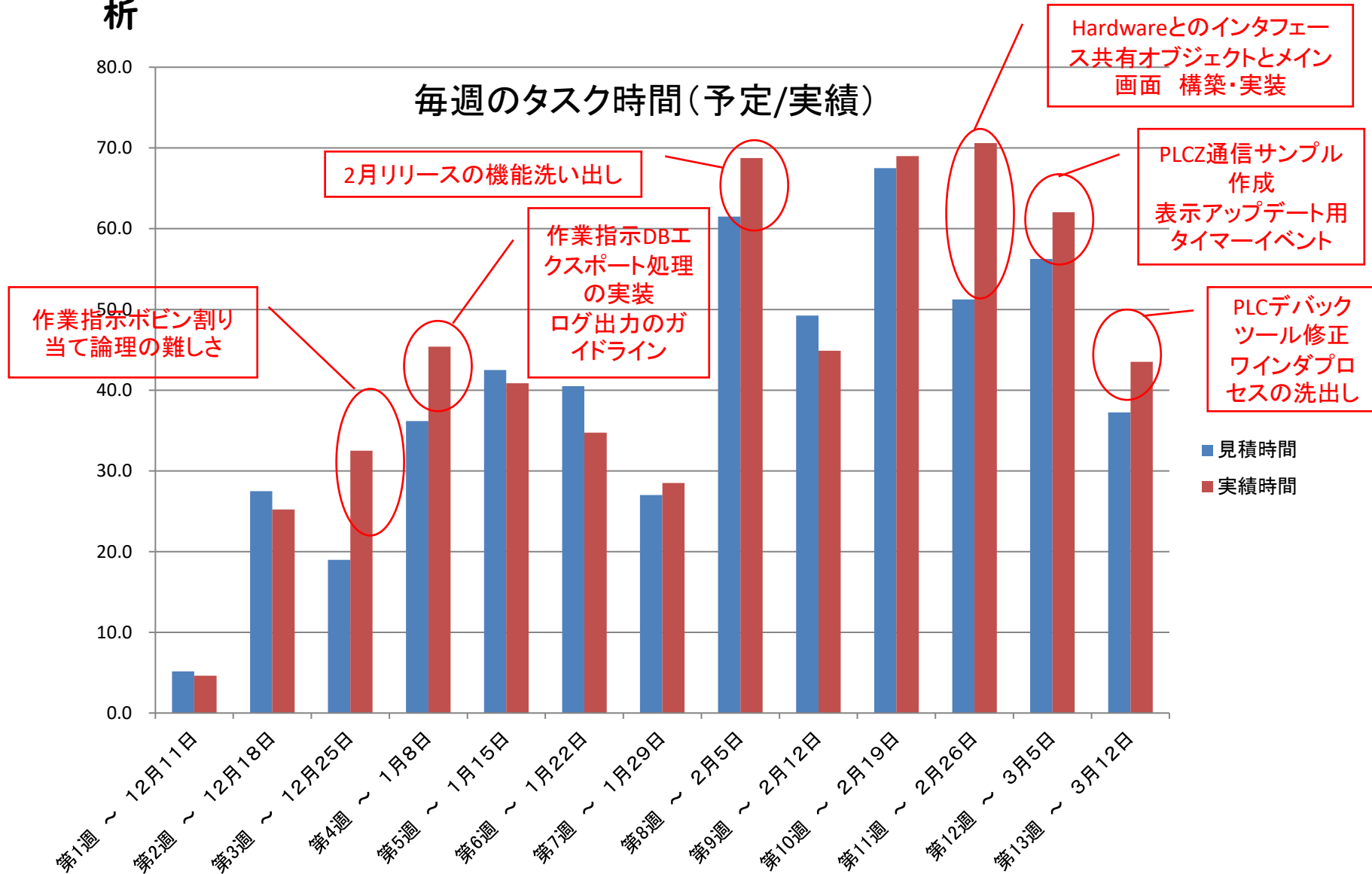
↑

生産性の向上
3スプリントで168.7%
向上

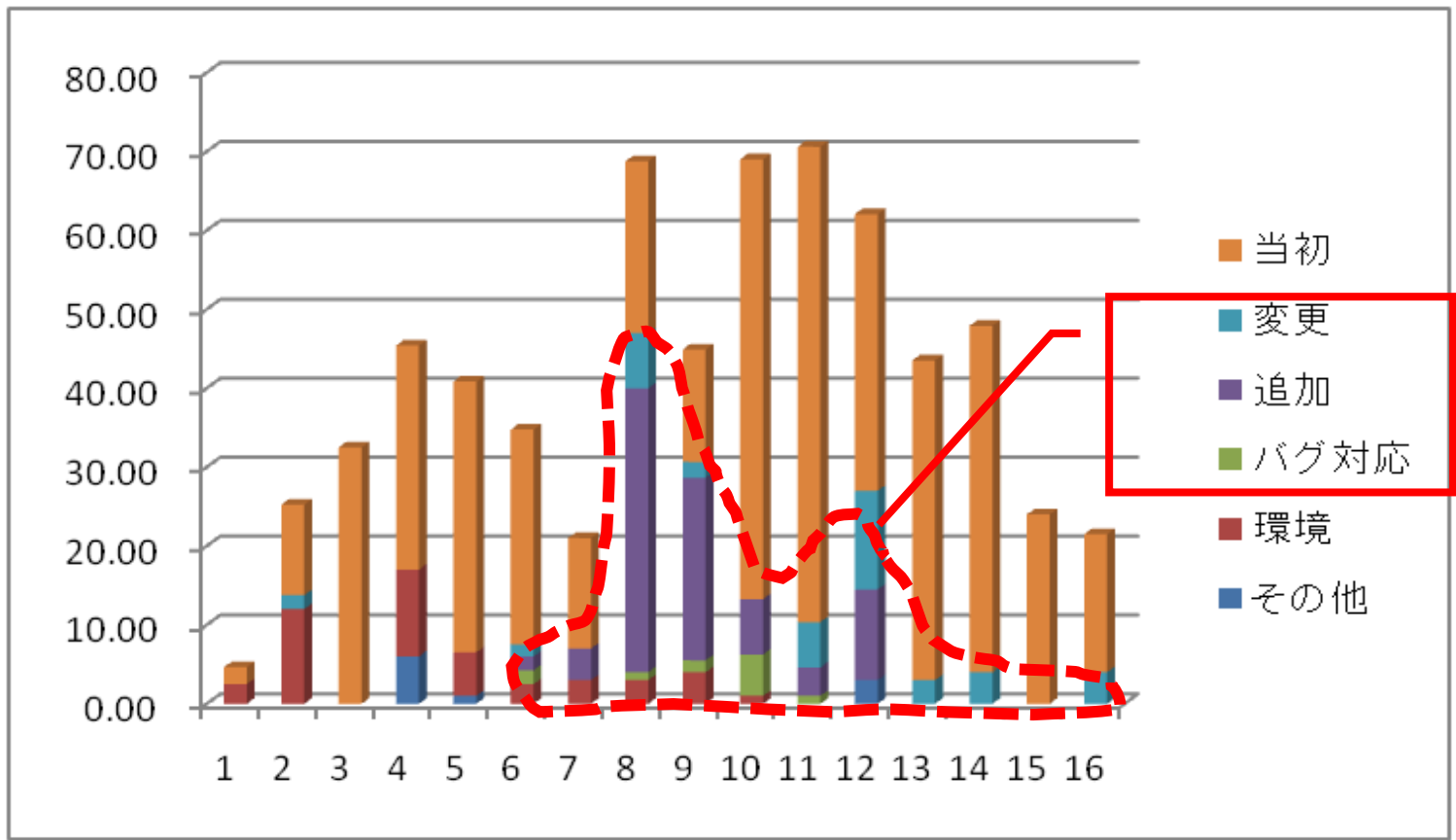
8スプリント後で、
250%向上

アジャイル開発では、何を管理すれば良いのか？

ベロシティー（スプリント期間内での作業タスク数）と作業時間（見積と実績）分析



毎週の作業要因別実績時間



タスクの粒度と平準化 (実績)

	見積総 時間	実績総 時間	タスク 総数	見積粒 度 (H)	実績粒 度 (H)
第1週～第7週 イテレーション ン ユーザー操作系 が主な機能 1 回目	197.8	211.9	134	1.476	1.581
第8週～第16週 イテレーション ン 機械制御系 が主な機能 2 回目	418.4	452.2	295	1.418	1.533

全く異なった機能実装
イテレーションでも同じ
粒度で平準化している

経験事例の報告（エンジニアの声）

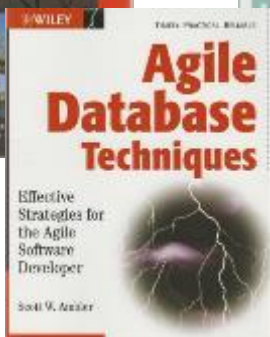
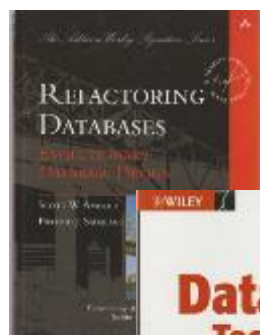
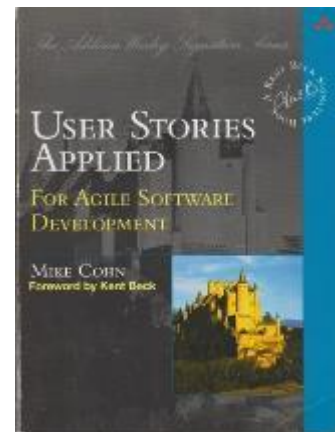
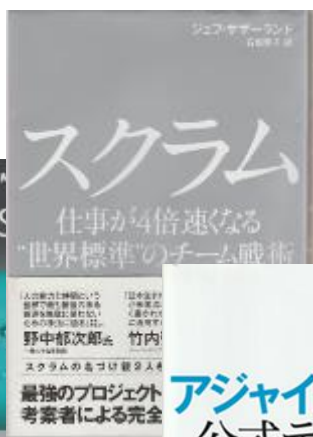
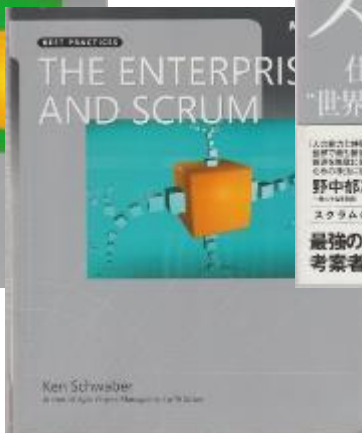
エンジニアAさん：アジャイル初体験、ウォーターフォール開発経験約15年（主任クラス）37才

- > 情報処理技術者 1種 > UML LI
- > 業務SE（物流／生産管理／公共サービス）8年 > ホスト汎用機テクニカルSE 7年
- > 今回 .NETは初めての経験 > JAVA (web) 3年
- > PL/SQL 2年 > VB.NET（2週間：本アジャイル開発で初）
- > COBOL、C++ etc（細かい開発は多数）

アジャイル開発の効果：コーディングの生産性は変わらないが、**開発作業内で手戻りが無く、結果的に早く完了できる。**

体験した感想：**とにかく頭が疲れる。集中する。**

- ・品質が高くなる。
 - ・技術的な問題や、方式で**悩む時間が少ないので効率は良い。**
 - ・気を抜く暇がないので、**稼働率は高い**
 - ・二人でやっているのでも、生産性は倍まではいかない。ただし品質が高いので、改修やテスト時の修正工数は少なくなる
 - ・ペアプロ／クロスファンクションにより ソースコードレベルで情報を共有するため、自然に **可読性／ロジックのシンプルさが感じられる実装となる。**
 - ・随時に動かしながら機能拡張をするため、**潜在バグ／デグレードのリスクは低い。**
 - ・実装が不慣れな要員がいても、ペアの組み合わせにより品質の高い実装が可能となる。
 - ・作業の完了が、視覚的に理解できる。実装の成果がすぐに見れる。
 - ・スタンドアップミーティング／振り返り／タスクの割り振りによりメンバー全員が全体の作業を見渡せる。司会を持ち回りすることにより参加意識が強調される。
- 人に見られているのに、適当な（動けばいい）コーディングはできない。**
- ・悩んでいる時間が少ない。（随時相談／調査）
 - ・具体的な目標を随時持つことができる。
 - ・タスク担当を明確にすることにより、責任範囲の**当事者意識を持つことができる。**



DevOpsとは



- Development + Operations の造語
- 2009年パトリック・デュボア氏は開発と運用を結び付ける手段としてDevOpsを提唱し、最初のDevOpsカンファレンス「DevOpsDays」を開催している。
- 2009年に「Velocity 2009」において、**flickr**のエンジニアにより初めて公の場で用いられた。

このプレゼンテーションでは「開発と運用が協力することで、1日に10回以上のペースでリリースが可能になる」という発表とともにDevOpsという単語が用いられた。

- **アジャイルとリーンの原則がソフトウェア・サプライチェーン全体に適用される。**
- ALM (アプリケーション・ライフサイクル) の視点でのITサービス管理プロセス
- ビジネス主導のITサービスプロセス改善
- DevOpsとは、人、文化の変化

DevOps is not Tool, DevOps is People, People use Tool. (Carnegie Mellon SEI)

JITでのITサービスの提供

企業におけるITの価値の本質に立ち返る活動

ITの価値の本質 = ビジネスをタイムリーに支援する。
ビジネス・スピードを牽引する。
ビジネス領域を拡大させる。
見えなかったモノを見える様にする。

IBMで継続的デリバリ関連製品のマネジメントリーダーを務めるEric Minick氏は、DevOpsの解釈は人それぞれ違う、とブログに記した。

- ・DevOpsはビジネスの成功を支援するために存在する
- ・適用範囲は広いが、中心となるのはITである
- ・基本はアジャイルとリーンである
- ・文化が非常に重要である
- ・フィードバックがイノベーションの原動力である
- ・自動化を活用する

CALMS モデル

- Culture - コラボレーションとコミュニケーションを推進するための変革を負う
- Automation - バリューチェーン内から手作業のステップをなくす
- Lean - リーン原則の採用により、サイクル周期を高める
- Metrics - すべてについて計測し、データをサイクルの洗練化に活用する
- Sharing - 他者が学べるように、経験や成功の可否を共有する

頭字語であるCAMS (Culture, Automation, Measurement, Sharing)は、John Willis氏とDamon Edwards氏が2010年、カリフォルニアのマウンテンビューで、米国内で始めて開催されたDevopsdaysの中で提唱したものです。後にリーンを意味する“L”がJez Humble氏によって加えられました。

DevOpsの最終的なゴール



縦割りの企業情報システム & ITプロセスの仕組み

全球运维大会

2016

DevOps 2.0: 重塑运维价值

北京站




全球运维大会·北京站
2016年12月16-17日
北京国际会议中心

- 指导单位:  数据中心联盟
- 主办单位:  高效运维社区  开放运维联盟
- 协办单位:  1024学院  CIO时代  XIN 
- 顶级赞助:  腾讯云  金山云
- 钻石赞助:  世纪互联  broadcom  联云世纪  SpeedyCloud 迅达云
- 金赞助:  博云  七牛云  LinkedSea 灵犀  光幕无限  云杉网络  PHEGDA 天玑物联 
- 牌赞助:  日志易  联云世纪  Steady 稳顺  齐治科技  NSFOCUS  北航咨询  天奇精英  彦博教育  VeryCloud 云腾网络

<http://gops2016-beijing.eventdove.com/>



GOPS 特邀讲师



Mr. Toda

TPS 认证学院主任

《DevOps & TPS —
For success your Business》

我在主会场等你

DevOpsDays 即将首次登陆中国

The conference that brings development and operations together.

Beijing 2017 - Welcome

Patrick Debois
DevOps 之父
Friday, Mar 17, 2017
Saturday, Mar 18, 2017

Gold Sponsors
Silver Sponsors
Bronze Sponsors
Community Sponsors

Dates: Friday, Mar 17, 2017 - Saturday, Mar 18, 2017

Sponsors: Get in touch

Contact: Get in touch





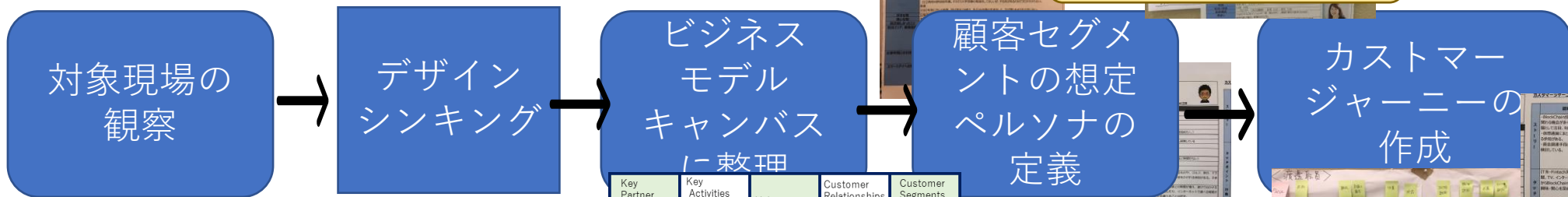
DevOps 2.0 とは

	DevOps 1.0	DevOps 2.0
カバーする領域	開発からデプロイ・リリース迄	ビジネス企画から運用・EOL迄
主な着目点	頻繁なリリース、スピード SoE (Web)が主体	素早いリリース、事業の継続性 SoE, SoR双方
適用する概念、 手法	アジャイル開発、継続的インテ グレーション、継続的デリバリー、 リーンの考え方	規律あるアジャイル開発、継続 的インテグレーション、継続的デ リバリー、軽量化したITサービ スマネジメント、ISO20000、 ALM、TPSの考え方

Process

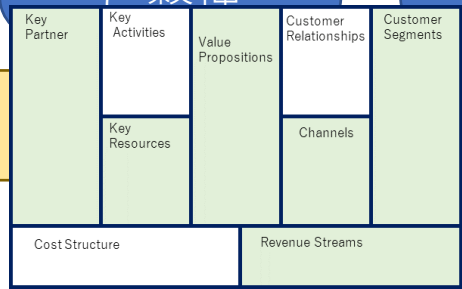


T-Methodのプロセス



ITサービスのイメージ作成 (ゴールイメージ)

ペルソナのイメージ作成 (ライフシーン)



ITシステムとの接点 (MOT) の定義

ボディープロセスの定義

オルタナティブ & オプションプロセスの定義

各プロセスをユーザーストーリーに救済

優先順位を付けてプロダクトバックログリストを完成

ハイレベルな機能定義と想定プロセスをユーザーストーリーに整理

バリューストリームマップ(Value Stream Mapping)



定義されたバリューストリームマップの一つ一つのプロセス (箱) で必要な完了条件を整理します。また同時にそれぞれのプロセス (箱) で実現すべき機能要件 (どんな事をしなければいけませんか?) を洗い出します。

次その実現すべき機能をユーザーストーリーのフォーマットを利用して一つ一つ整理します。

User Story

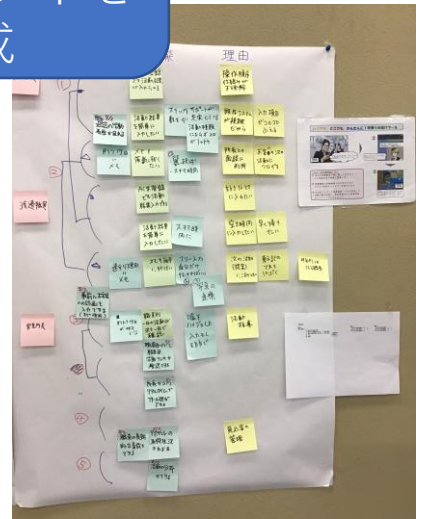
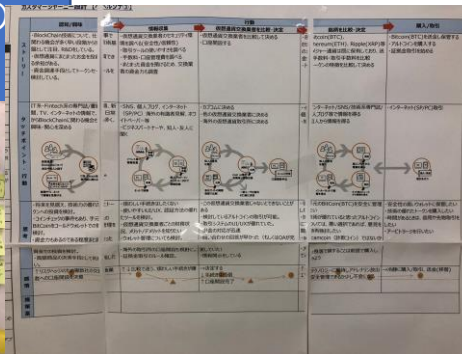
As a Role
I can
Which I need
To

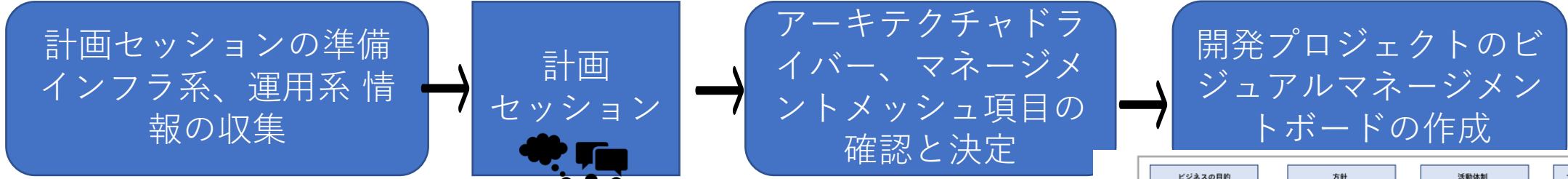
全てがユーザーストーリーに書き出せましたら、要件定義の完了です。

ユーザーストーリーは各々が独立している事

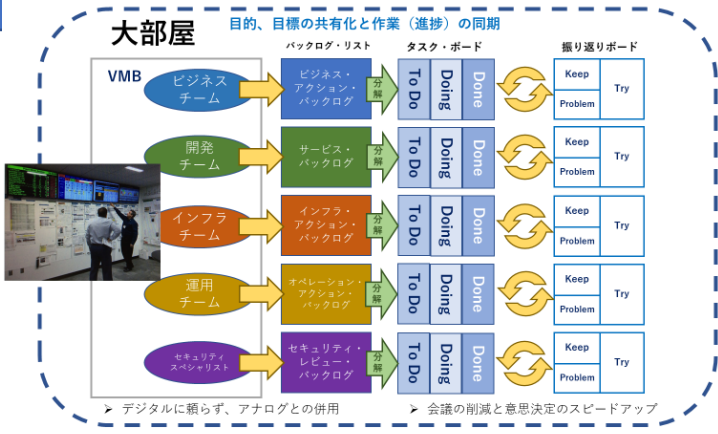
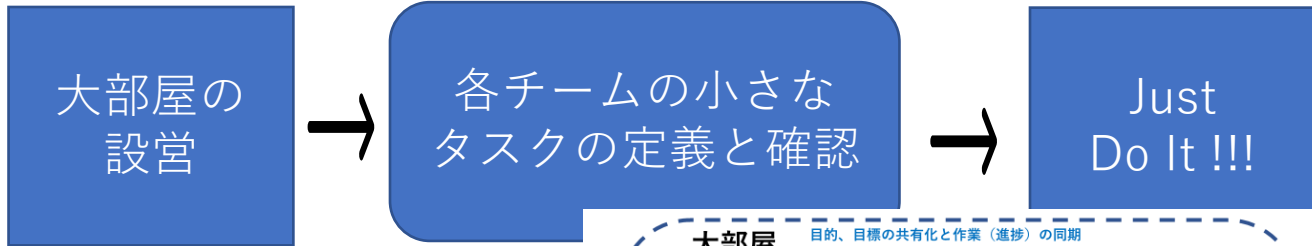
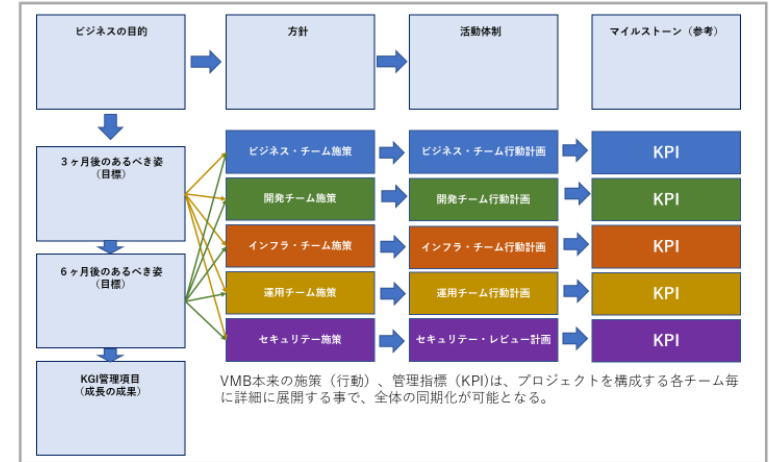
- Negotiable (交渉可能)
ユーザーストーリーを使って、ユーザーと開発者が会話できる結果をまとめる。
- Value to Users/Customers (価値を提供)
一つ一つのユーザーストーリーがユーザーにとって価値を提供する。
- Estimatable (見積可能)
開発者が作業を見積れる事。
- Small (小さい、簡潔)
- Testable (テスト可能)

ID:	日付: 2015/06/26
要求者名:	記入者名:
部門名:	
役割 (As a Role):	
機能 (I/We can):	
環境・条件 (Which I need):	
効果・価値 (To):	

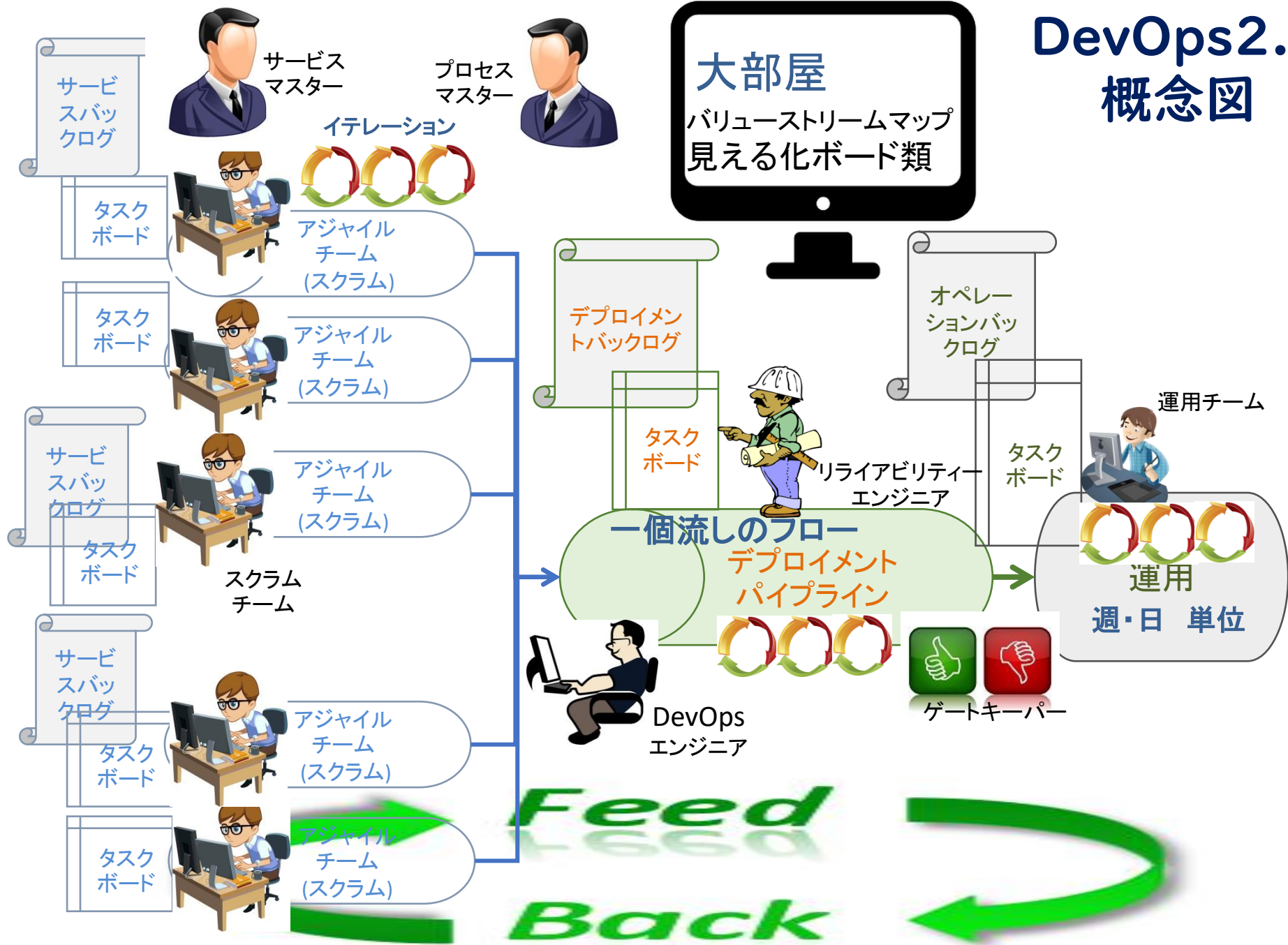




プランニング・セッション

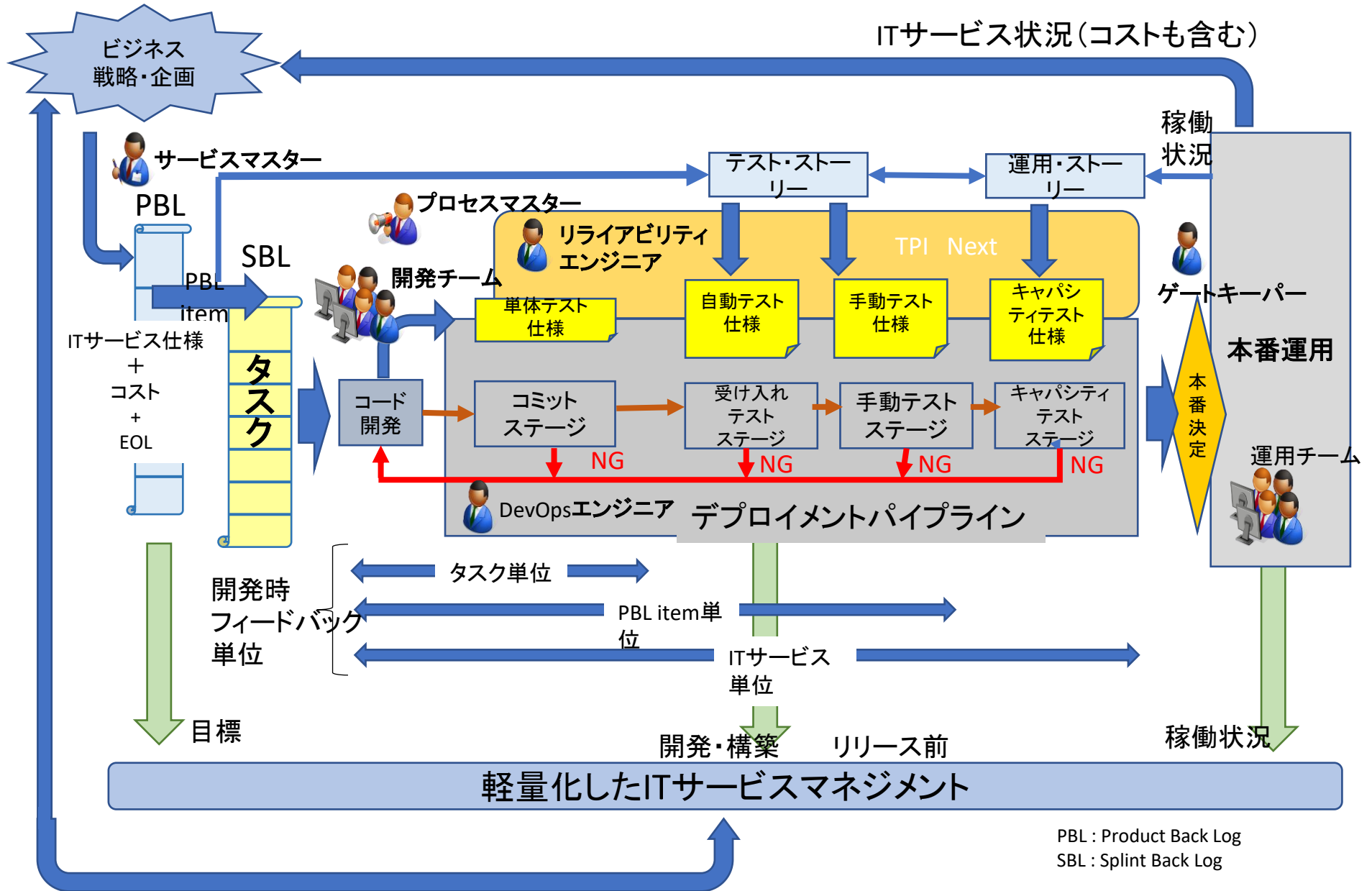


DevOps2.0 概念図



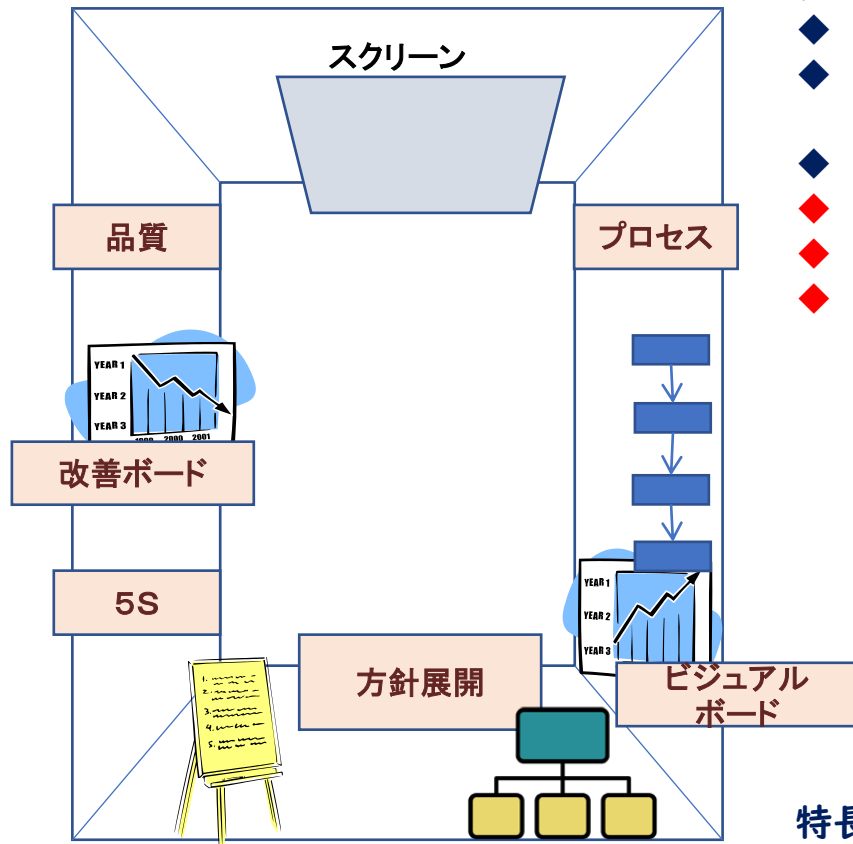
計画から運用までの全体最適による流水化されたプロセスの構築

デプロイメントパイプラインの概略フロー



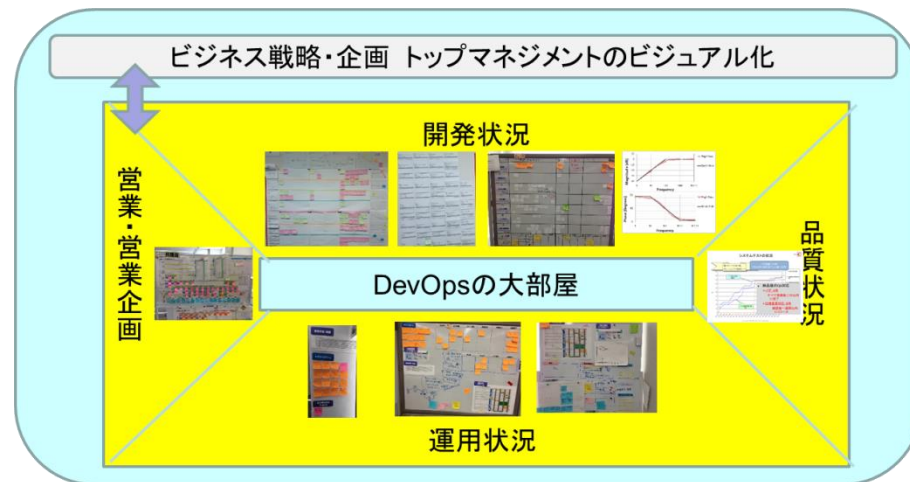
大部屋システム

大部屋の概略図



狙い:

- ◆ 人と職場(チーム)の活性化を促進する効果大きい
- ◆ プロジェクト全体が俯瞰(見える化)できる
「見える化」とは、異常が瞬時に解る、気づく事
- ◆ Scrum of Scrumの実施会場(場所)
- ◆ DevOpsの全プロセスの見える化(バリューストリームマップ)
- ◆ プロセス間の繋ぎ(リレーのバトンタッチゾーン)が重要
- ◆ プロセス間の同期化

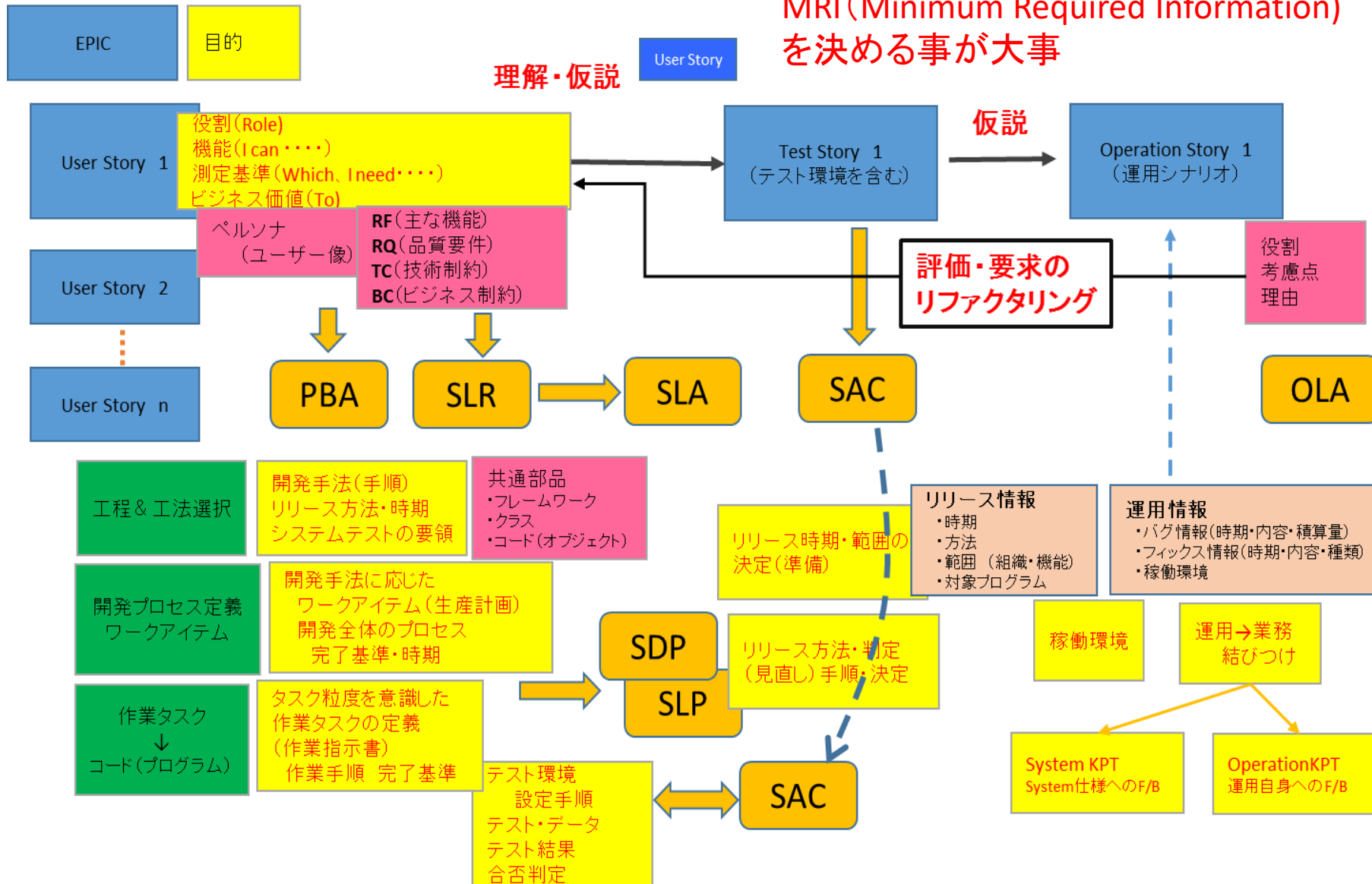


特長:

大きなプロジェクトや他部署との**連携したチーム**など**組織横断的**な事業に取り組む場合には、『目で見える管理』を大部屋化して行う
大部屋とは、関係者が一つの部屋に集まって課題を**見える化**し、**問題解決を進めていくやり方**
同じ部屋で仕事を行う事で**コミュニケーションが活発化**し**アイデアが沢山**でる様になる

軽量化されたITサービスマネジメントを適用した全体概要図

MRI (Minimum Required Information) を決める事が大事



各作業現場では、正しく作業を行うために、それなりの情報が創出されている。

DevOps成熟度モデル (DevOps Maturity Model)

ステージ1：自己管理 (Analysis Ability)

アジャイル&DevOpsの知識・技能を習得し、自己管理の為の分析が働く開発チーム

ステージ2：プロセスの見える化 (End-to-end Traceability)

DevOpsチームの活動の「見える化」、活動のログを改善に繋げる。

ステージ3：プロセスの安定化 (Stabilized Development & Operation Process)

お客様、システムの要求品質を理解し、その品質を満たす。開発スピード(ベロシティー)を巡行速度で維持し、定期的にリリースして安定運用できる。

ステージ4：学習する組織 (Learning Organization)

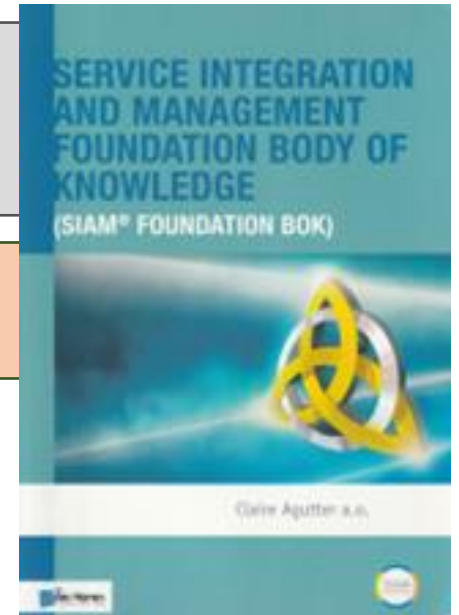
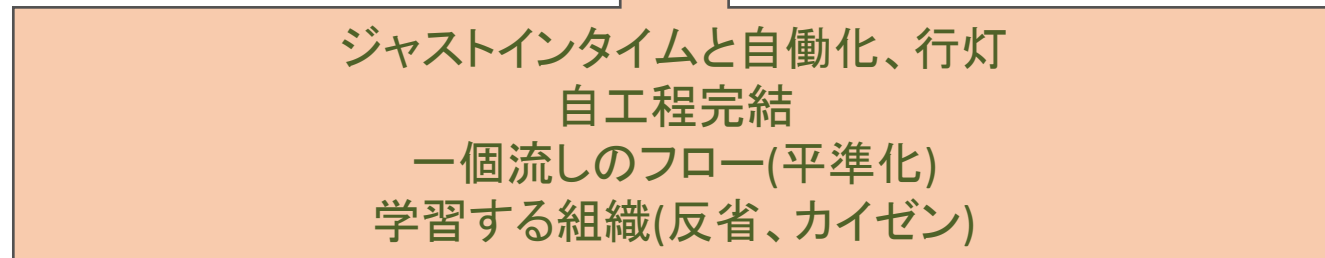
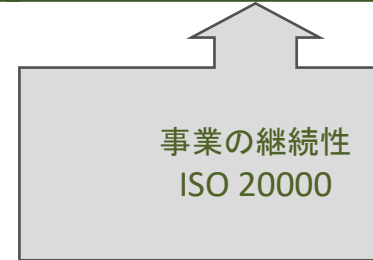
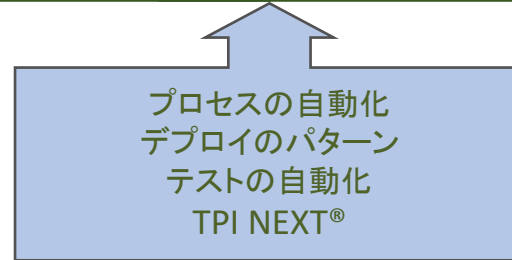
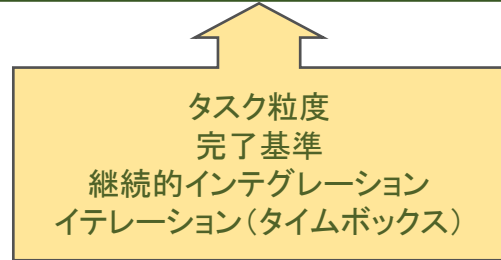
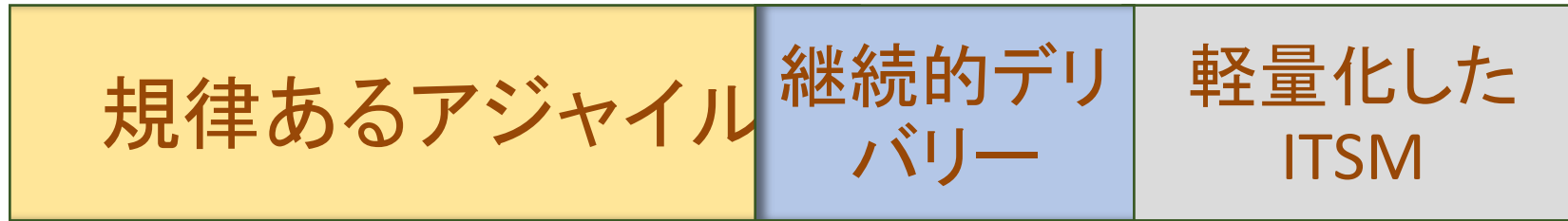
品質、生産性、納期(リードタイム)を意識し改善が進む学習するDevOpsチームで自らモチベーションを維持してワークライフバランスが取れる。

ステージ5：自律した組織 (Anticipated ROI & Failure Tolerant Organization)

完全に自律し、問題解決適応力のあるDevOpsチームで事業継続性を担保できる。

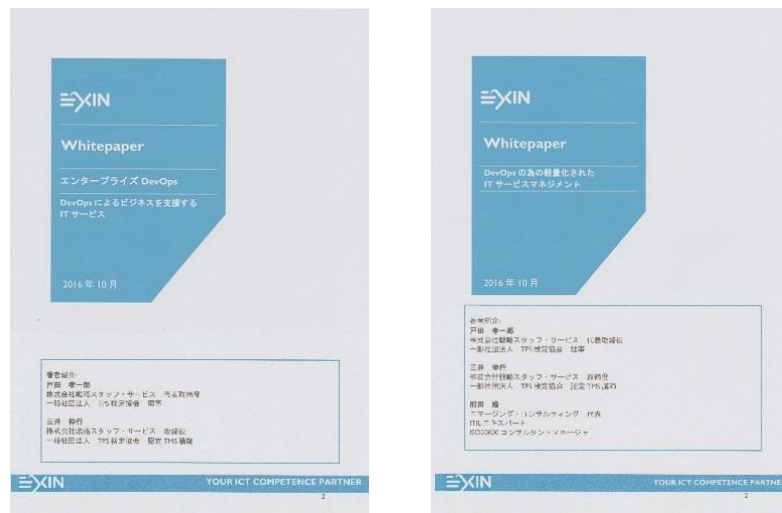
DevOps2.0に必要な知識体系

Process



EXINのサイトから入手できます。

ホワイトペーパー



エンタープライズDevOps

DevOpsによるビジネスを支援するITサービス

DevOpsの為に軽量化された ITサービスマネジメント

<https://www.exin.com/JP/ja/home/>

メールマガジン連載;『全体最適へのDevOps(Agile, TPI, ITIL/ITSM)』

EXINでは計10回、システムの全体最適をテーマにメルマガを6月17日より配信

- 第1回: ITIL側から見たAgileの不安
- 第2回: 計画・企画段階のAgileとITIL
- 第3回: イテレーション期間のITILから診た評価
- 第4回: テストにおける当たり前品質とSAC
- 第5回: リリース時の評価とSDPとSLP
- 第6回: 運用へのScrumの適用
- 第7回: 研究と改善(品質は現場でつくられる)(運用状況によるフィールドバック)
- 第8回: 全体最適へ(DevOps)軽量化したITSM (1)全体像
- 第9回: 全体最適へ(DevOps)軽量化したITSM (2)適応させるロードマップ
- 第10回: 全体最適へ(DevOps)軽量化したITSM (3)理想となる姿とそれに向けたチェック項目、バーンアップ・チャート

<https://www.exin.com/JP/ja/news-japan/2016/dommg>

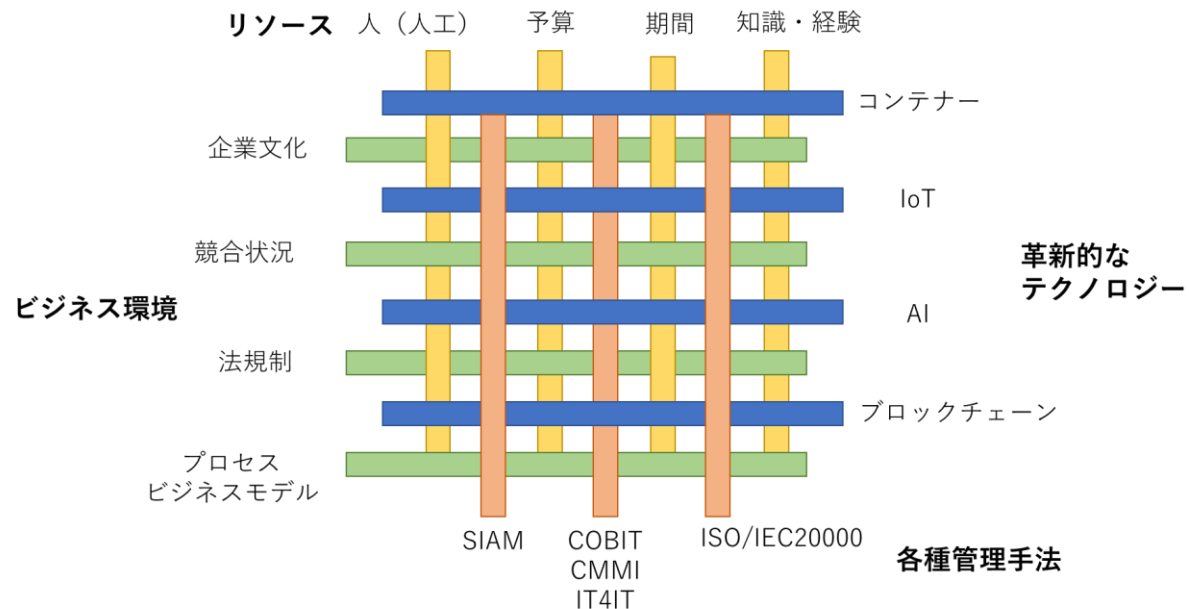
ライフサイクル

ソフトウェアは無形固定資産か???

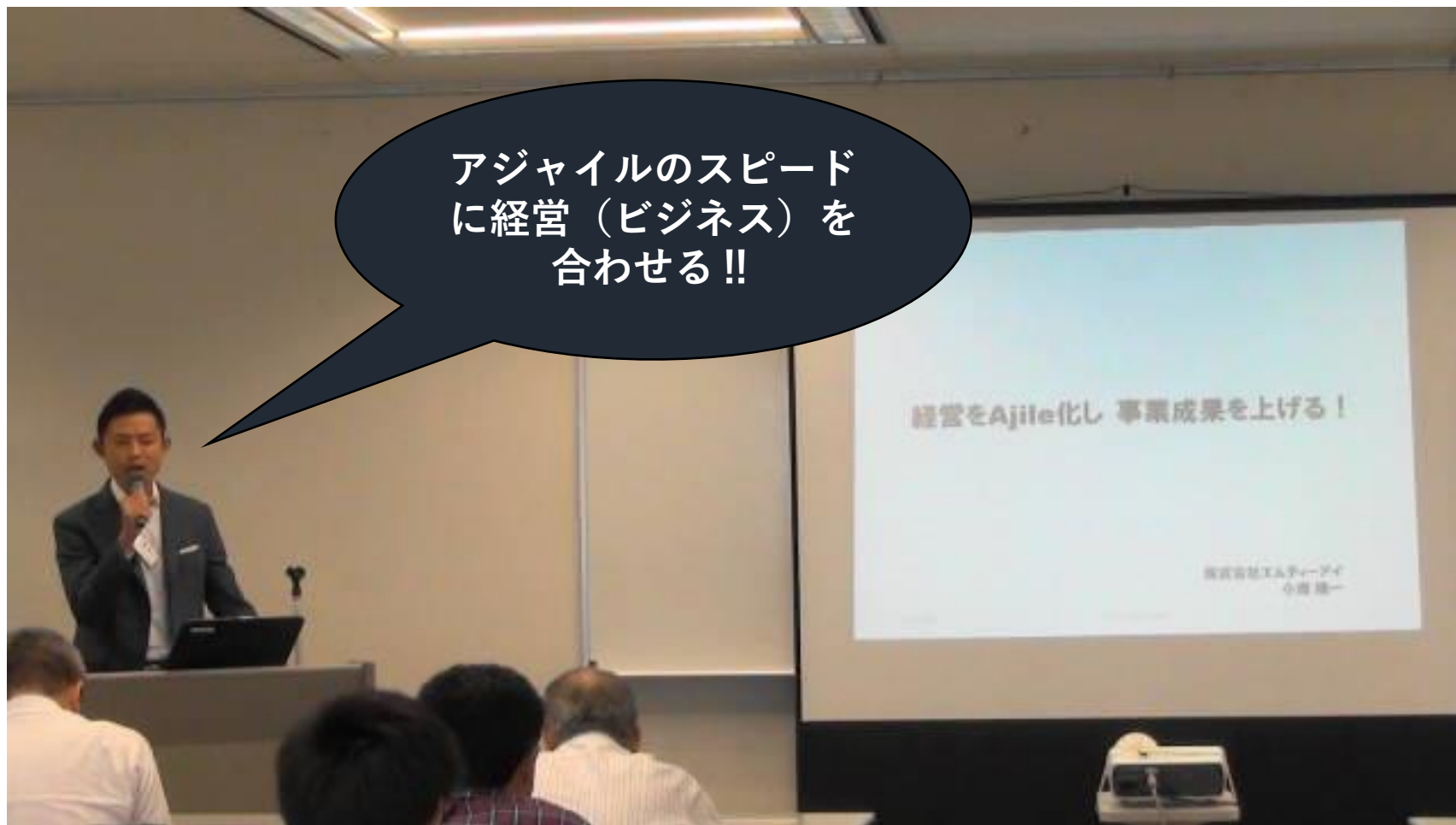
IT技術には正味期限がある。(素早い陳腐化、ムーアの法則)

サービスは環境や経験によって変質する。

ソフトウェアを機能の実現と見るか? サービスの実現とみるか?



成功したアジャイルプロジェクトの事例紹介



ビジネススピードを高めた事例 (DevOps2.1の事例)

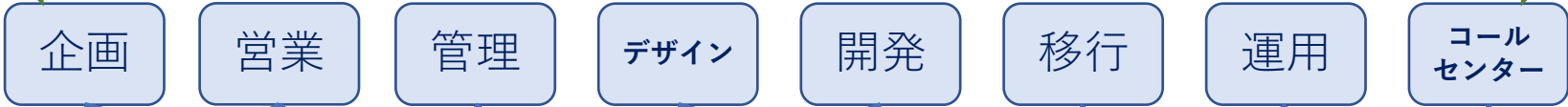
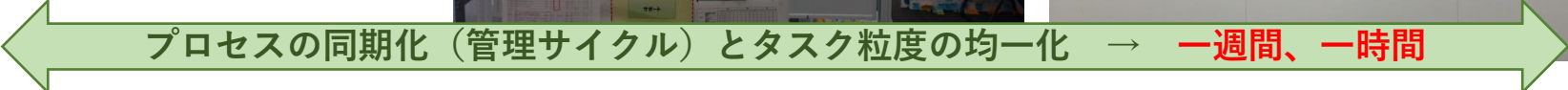
2009年 アジャイル開発を開始
2011年秋 アジャイル開発は成功裏に導入できたが、、、
課題噴出： 開発工程はボトルネックでは無かった。
2012年4月 ビジネスプロセスの全工程の見直しと整流化のプロジェクト開始
2013年10月DevOpsの導入 (全プロセスの見える化、同期化と大部屋化実現)

事業規模が
2年で3倍以上

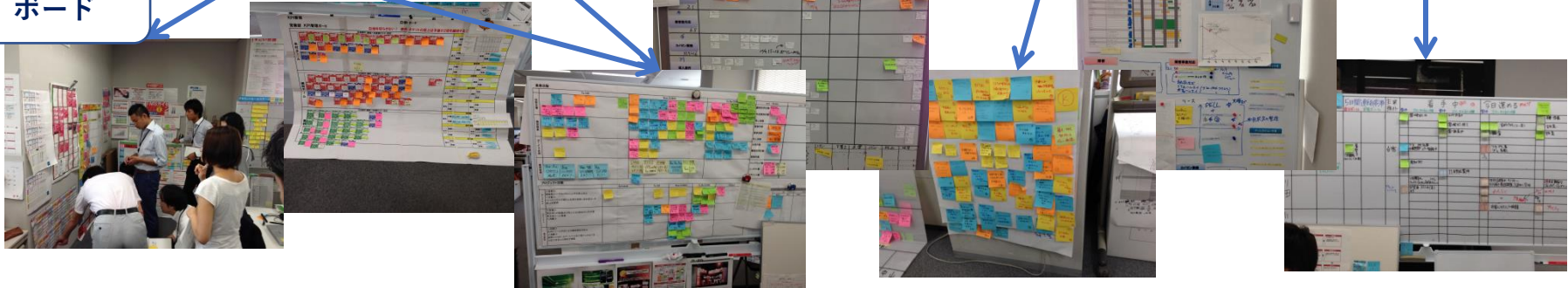
非製造業で初の
大部屋の実現



MTI (株) PS事業部



ビジュアル
ボード



ケーススタディー

Bit Coin POC project

VeriSM & TMS (世界初のDX管理モデル) を活用した
プロジェクトの事例 (グローバルに発表)



18ヶ月間のプロジェクト企画案 (稟議取得) を3ヶ月間で実現する方法

2018年度
IT特別賞 (組織風土改革賞) 受賞  2018年度企業情報化協会
スーパーSE100人衆登録



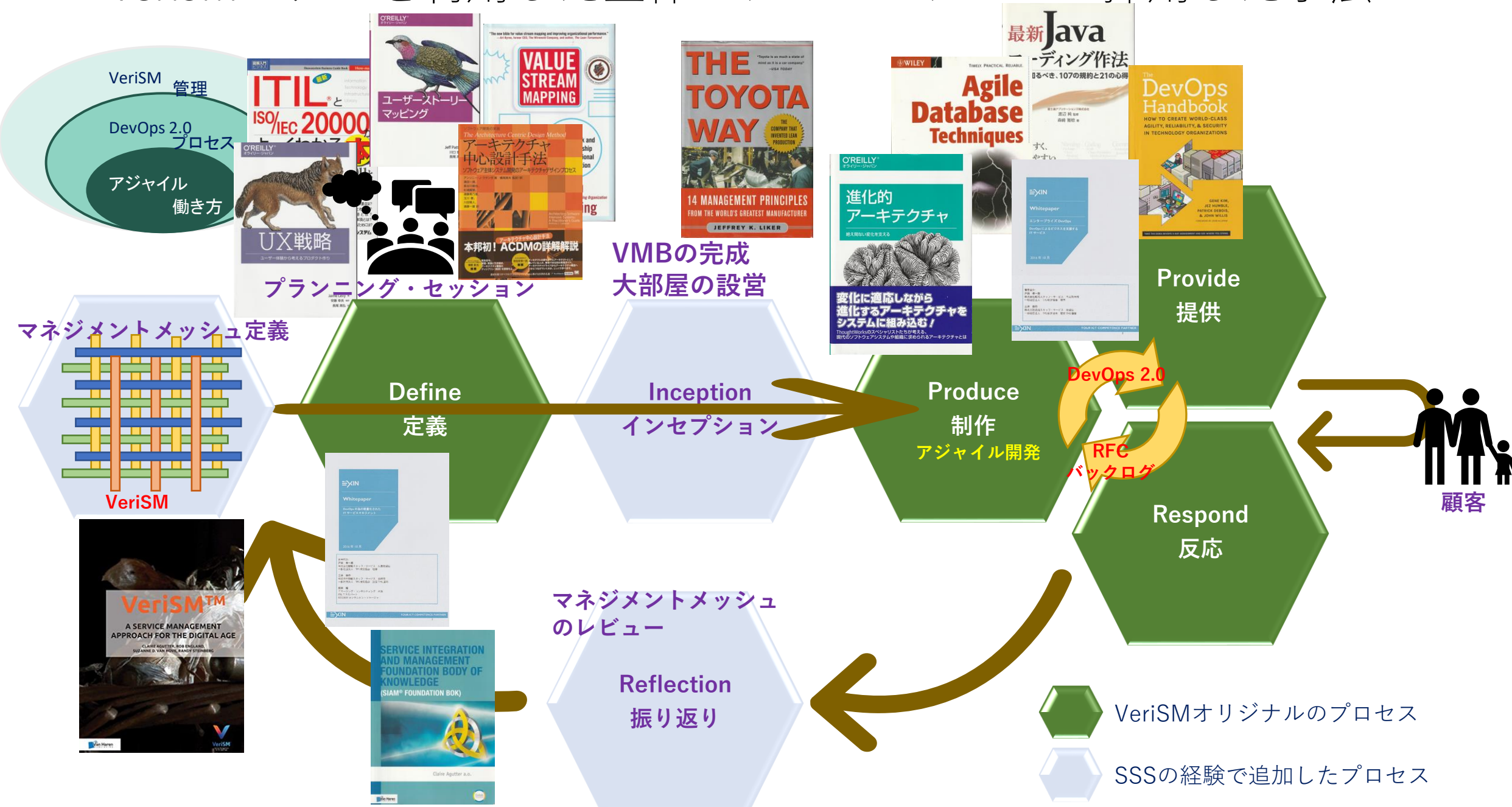
 日本証券協会グループ
公益社団法人 企業情報化協会



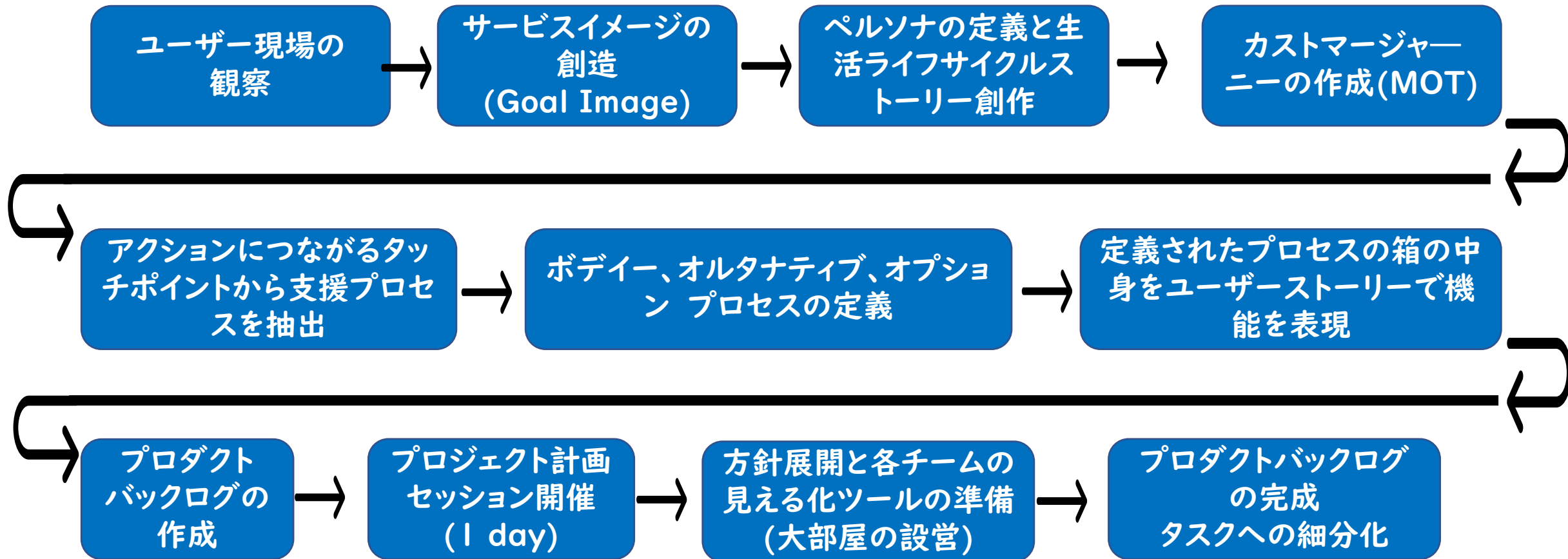
カブドットコム証券

わたしたちは  MUFGです。

VeriSMモデルを利用した全体のプロセスフローと採用した手法



Process of T-Method – Outline 新しい設計アプローチ



日本を元気に

革新コンソーシアム

Consortium for Innovation

<https://www.facebook.com/groups/jpinnova/>

国立国会図書館オンライン

<https://ci.nii.ac.jp/>

[https://ndlonline.ndl.go.jp/#!/search?lang=ja_JP
&keyword=%E4%B8%89%E4%BA%95%E4%BC%B8%E8%A1%8C&maintain=true&searchCode=SIMPLE](https://ndlonline.ndl.go.jp/#!/search?lang=ja_JP&keyword=%E4%B8%89%E4%BA%95%E4%BC%B8%E8%A1%8C&maintain=true&searchCode=SIMPLE)

日経コンピュータ連載コラム (2)

総合タイトル：続編 【現場を元気にする組織変革術】

- | | |
|-----------------|-----------------------------------|
| 1. 2018年4月12日 | 組織変革は1日にして成らず トップを変え、3段階で現場を変える |
| 2. 2018年4月26日 | やる気引き出す「心のマネジメント」、まずは鏡を |
| 3. 2018年5月10日 | 言葉では3割しか伝わらない、現場に向き変化を感じ取る |
| 4. 2018年5月24日 | 大切なのはルールよりマナー、「人の幸せ」を目指す発想法 |
| 5. 2018年6月7日 | 意思決定の遅さを変革 新事業を次々生み出すコツ |
| 6. 2018年6月21日 | 泥臭さと人を幸せにする思い リーダーに求められる行動原理 |
| 7. 2018年7月5日 | さらば縦割り組織、形だけで終わらせないマネジメント |
| 8. 2018年7月19日 | 「知る」と「分かる」は大違い マネジャーは現地現物の徹底を |
| 9. 2018年8月2日 | 誰でも必ず変革できる 行動できない人を動かす術 |
| 10. 2018年8月16日 | 慣れ親しんだ全てが変革を妨げる ウォーターフォールの現場に決別 |
| 11. 2018年8月30日 | 高めたいのはIQより「EQ」 フェイス・ツー・フェイスから始めよう |
| 12. 2018年9月13日 | アジャイル変革は「急が回れ」 上っ面の理解では失敗が連鎖する |
| 13. 2018年9月27日 | 変化の兆しを感じ取れるか 「KPT」で可視化して成長促す |
| 14. 2018年10月11日 | 運用部門に元気を取り戻す NTTデータCCSの挑戦 |
| 15. 2018年10月25日 | チェックリストが元気を奪う NTTデータCCSの気づきと改善 |
| 16. 2018年11月22日 | ベンダー常駐チームと初アジャイル 外資系保険会社が一念発起 |
| 17. 2018年12月6日 | 外資系保険会社の初アジャイル 成功への軌跡 |

日経コンピュータ連載コラム (1)

総合タイトル：前編 【現場を元気にするチーム運営術】

- | | |
|----------------|-----------------------------------|
| 1. 2017年4月13日 | 元気が無い現場に四つのサイン あいさつが元気を取り戻す第一歩 |
| 2. 2017年4月27日 | コミュニケーションが取れない組織 他人をどれだけ知っているのか |
| 3. 2017年5月11日 | 残業が減らない組織 属人的な仕事を無くしているか |
| 4. 2017年5月25日 | 多忙で夏休みを取れない職場 改善活動の停滞乗り越え取得率向上 |
| 5. 2017年6月8日 | 助け合いの無いシステム部門 「心のマネジメント」が改善のカギ |
| 6. 2017年6月22日 | アジャイル開発でチームを元気に初挑戦でも成功、カギは見える化 |
| 7. 2017年7月6日 | 発注者の信頼をどう得るのか アジャイル開発成功のカギ |
| 8. 2017年7月20日 | チームの透明性はアナログで高める 自らを律するのが成功の秘訣 |
| 9. 2017年8月3日 | 暗すぎるウォーターフォールの現場 やる気引き出す心のマネジメントを |
| 10. 2017年8月17日 | 進捗が90%から先に進まない 1時間の作業に分割しているか |

総合タイトル：後編 【現場を元気にするDevOps 2.0】

- | | |
|----------------|----------------------------------|
| 1. 2017年9月14日 | 今こそ大量生産時代の常識に決別 トヨタ由来の方法論で元気になる |
| 2. 2017年9月28日 | 自ら改善を続けるチームに変える 働き方改革への第一歩 |
| 3. 2017年10月12日 | 人を大切にするのが「強い企業」 仕事のやり方を変えて近づこう |
| 4. 2017年10月26日 | 「サイロ」現場を3段階で変革 個別最適では幸せにはなれない |
| 5. 2017年11月9日 | アジャイル開発の品質向上策 7つの視点で作業の様子を観察 |
| 6. 2017年11月23日 | 「伝達ゲーム」は開発失敗の元凶 業務プロセスの観点を貫こう |
| 7. 2017年12月7日 | ビジネスの速度が3倍に 全体最適に変革する3ステップ |
| 8. 2017年12月21日 | ウォーターフォールは時流に合わない まずは安全管理と可視化に着手 |
| 9. 2018年1月4日 | チームを育てるならアジャイルを マネジャーは支援役に徹する |
| 10. 2018年1月18日 | ツールだけでは元気は出ないDevOps 2.0に必要な7つの役割 |

日経コンピュータ連載コラム

総合タイトル： 【現場を元気に！ カブコム現場改革の軌跡】

1. 2019年1月24日
2. 2019年2月21日
3. 2019年3月21日
4. 2019年4月18日
5. 2019年5月16日

6. 2019年6月13日
7. 2019年7月11日
8. 2019年8月8日
9. 2019年9月5日
10. 2019年10月1日

創業15年目にシステム障害が多発 停滞した現場に社長自らがメス
統率型に染まったリーダーたち それで現場が明るくなる？
冷めたメンバーは必ずいる 思いを聞き出しチーム巻き込む
スランプは改革に付き物、答えは不要、やる気促す声かけを
熱い思いが空回り そんなときこそ「聞く耳」を

組織の筋トレ
の話

2泊3日の集中議論 関係者の合意を一気に取る
社長メッセージでやる気にさせる 文書作成よりも全員合意を重視
開発費を6割削減 初挑戦アジャイルで得た成果
アンチの塾生に1時間向き合う 行動する人を支えて良い流れに
悪役を演じてチーム崩壊を回避 人の感情を無視してはならない

VeriSMを適用した
DXプロジェクト
の話



参考情報 (Youtube動画)

日本を元気に・革新コンソーシア

<https://www.youtube.com/c/jpinnova>

プロダクト愛(実践者が語るプロダクトオーナーの秘訣)(第2回 現場改善会議『チーム改革のスイッチ』#10)

<https://www.youtube.com/watch?v=oXdcEcS5mlc&t=70s>

変えてはならないモノ『マイ・マニフェスト』(第3回 現場改善会議『チーム改革のスイッチ』#11)

<https://www.youtube.com/watch?v=7TtatxLs5dw&t=533s>

T-Methodについて。(『チーム改革のスイッチ』#21 第13回 現場改善会議)

<https://www.youtube.com/watch?v=E9Hr6TcQRFY>

クレスコにおけるT-Method実践例(『チーム改革のスイッチ』#22 第14回 現場改善会議)

<https://www.youtube.com/watch?v=9G8z4PWl0jw>

DX認定制度システム開発裏話#2(『チーム改革のスイッチ』#24 第16回 現場改善会議)

政府調達のアジャイル奮闘裏話

<https://www.youtube.com/watch?v=Cx5ROx8iMII>

Scrum: How to do twice as much in half the time | Jeff Sutherland | TEDxAix

<https://www.youtube.com/watch?v=s4thQcgLCqk>