

芯

# Opsから定義するDev. 価値あるエンタープライズDevOpsとは？

株式会社 戦略スタッフ・サービス

EXIN Executive Designer for DevOps Certification

三井伸行

# 自己紹介

年齢:61歳      ソフトウェア産業従事歴 37年

## 主たる実績(直近10年)

- ソフトウェア生産技術力向上の為のコンサルテーション
- Agile開発の導入指導・プロジェクト支援(プロジェクトレスキューも含む)
  - 基幹系、生産管理、等
- TMS(Toyota Management System)の導入指導(ホワイトカラーの現場改善)
  - TPS : Toyota Production System:Lean , Agile開発の源流
- 働きかた改善の指導(コーチ)
- W/Fプロジェクト現場の見える化支援
- DevOpsの普及活動と導入支援

## 資格

- 米国ScrumAlliance.org 認定の公認スクラムマスター
- TMS&TPS検定協会 認定TMS(Toyota Management System)講師
- EXIN Agile Scrum Foundation 認定講師
- EXIN Executive Designer for DevOps Certification



- DevOps & CALMSモデルとは
- 何故Opsから定義するのか？
- DevOps2.0
- 軽量化されたITSM
- DevOps成熟度モデル (DevOps Maturity Model)
- EXIN DevOps Master 認定
- Related certification programs

# AGENDA

# DevOpsとは何か？

開発側

運用側

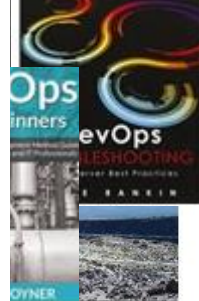
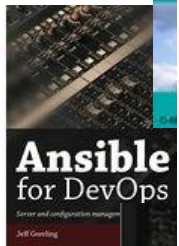
- Development + Operations の造語
- 2009年パトリック・デュボア氏は開発と運用を結び付ける手段としてDevOpsを提唱し、最初のDevOpsカンファレンス「DevOpsDays」を開催している。
- 2009年に「Velocity 2009」において、**flickr** のエンジニアにより初めて公の場で用いられた。
- このプレゼンテーションでは「開発と運用が協力することで、1日に10回以上のペースでリリースが可能になる」という発表とともにDevOpsという単語が用いられた。
- アジャイルとリーンの原則がソフトウェア・サプライチェーン全体に適用される。
- ALM(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。



# DevOpsの目的（ゴール）

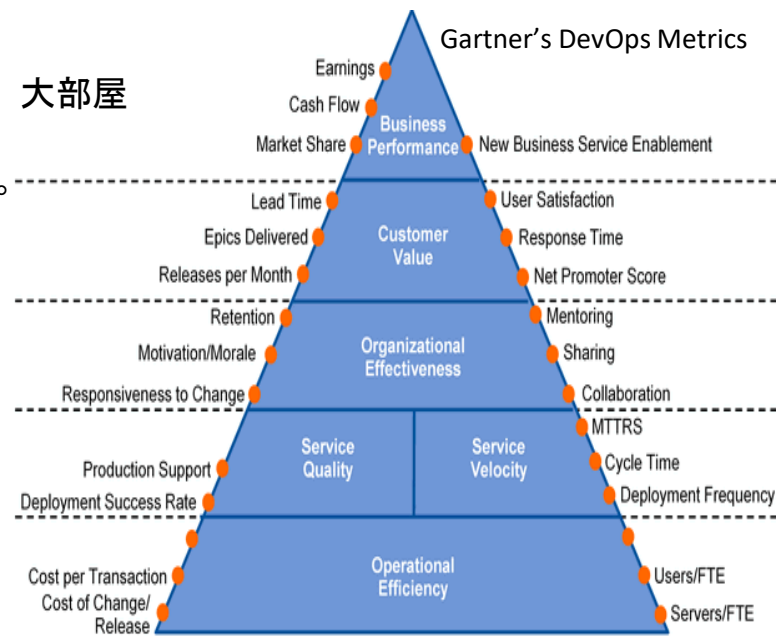
## • ジャストインタイムでITサービスを提供するサプライチェーン

- プロセスの流水化
  - 開発プロセス、テスト・プロセスと運用プロセスの同期化
- プロセスの成熟化(改善)
  - 開発プロセス: 自律したアジャイル・チーム
  - テストプロセス: TPI NEXT
  - 運用プロセス: ISO20000
- 全プロセスの見える化と現地現物での機敏な意思決定 → 大部屋
- 全体最適
- ソフトウェアを作る、提供する。 → ITサービスを提供する。

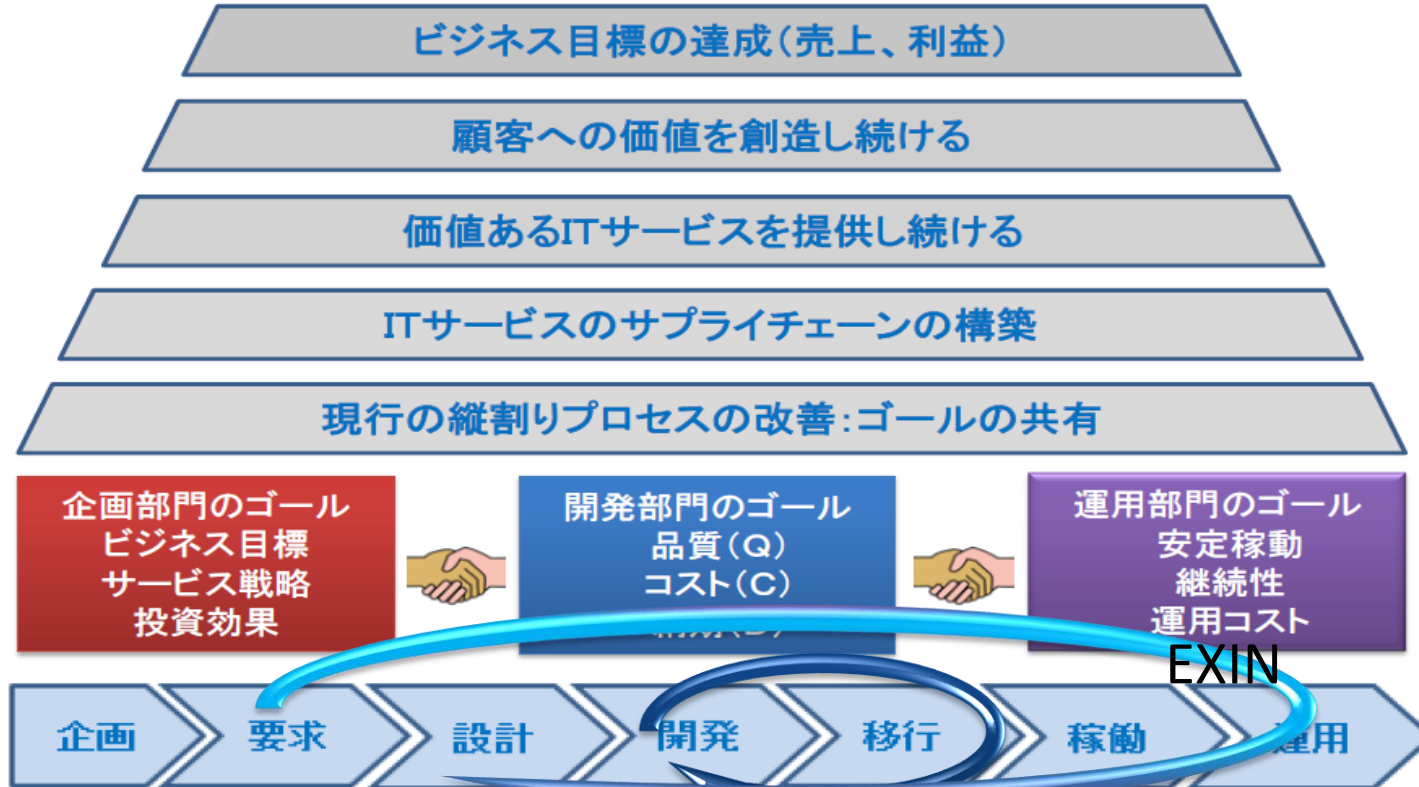
## • ビジネス成果での評価

- ビジネスへの貢献
  - 売上増加、利益増加

## • 事業継続性、事業の機敏性、適応力



# 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

# DevOpsの効果

- IT部門管理者
  - システム開発・運用のムダを省いてコストを削減する。
  - ビジネス・オーナー(部門)から評価されるJITでのITサービスを提供できる。
  - BCP(事業継続性)を考慮したITの仕組みが作れる。
  - ITプロジェクトの企画から開発・運用そしてEOLまでのALM(ライフサイクル)の管理体制(プロセス)が確立できる。
- 人財開発管理者
  - 全体最適を考えられる自律した社員・チームの育成ができる。
    - T型社員の育成
  - 作業品質を高めるワークスタイルを作る。
    - 当たり前品質から魅力的品質へ
  - ワークライフバランス、テレワーク(リモート)等の多様化した働き方を受け入れられるIT部門社員の育成。
- 経営者
  - IT投資の確実な回収が期待できる。
  - ビジネスの足を引っ張らないIT部門の運営ができる。
  - ビジネスに貢献するIT部門を作る。

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。



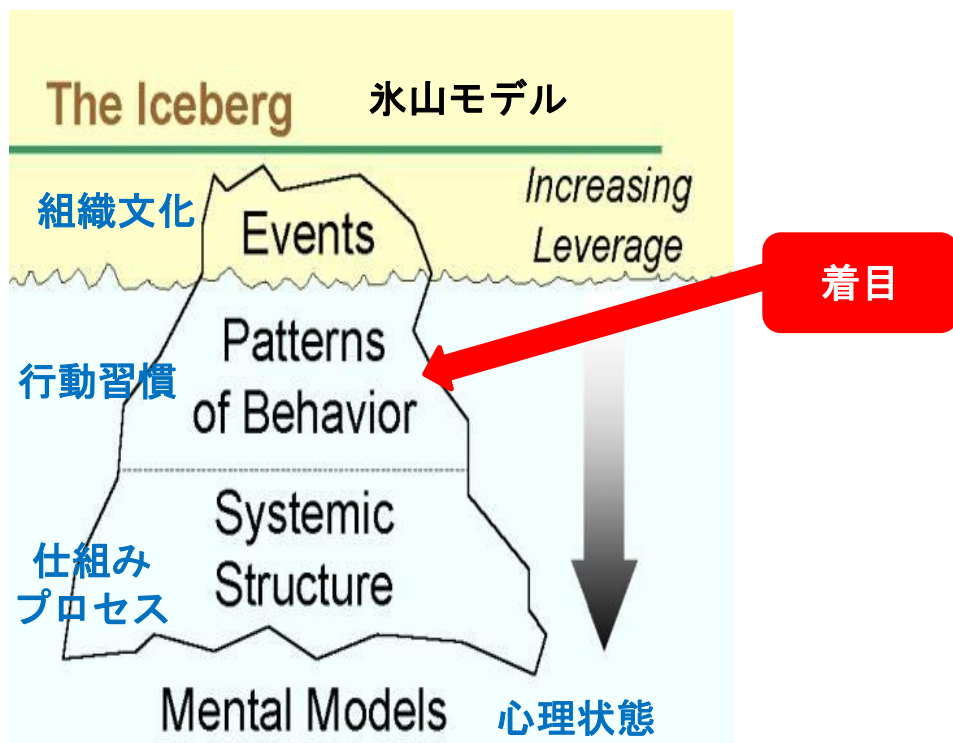
# どう変わるか？変えなければならないか？

	ウォーターフォールを中心とした従来の方式	アジャイル、DevOpsの新しい方式
価値観	計画重視	リソース重視、適応性重視
スピード	遅い	早い
働き方	仕事はまとめた方が効率が良い 残業を認める	仕事は一つづつこなした方が効率が良い 残業を認めない（定時間労働）
時間の使い方	仕事の目的を達するまでの時間	区切られた時間内で仕事の目的を達成 →タイムボックス
計画	計画重視 全期間にわたる計画立案 （計画通りにコトは運ぶ）	計画作り重視（朝令暮改） 当面1ヶ月は詳細に、後は粗い計画 （計画通りにはコトは運ばない）
プロジェクト	しっかりと計画を立て、理論的に進める。	反復的（イタラティブ）に進める フィードバックが重要
リスク	プロジェクト後半に増大	プロジェクト前半に集中
役割	専門家（分業化）	多能工化（T型人才）
進捗管理	管理指標	現地現物管理（働くプログラムのみ）
見える化	作業が終わらないと見えない	ほぼ何時でも見える化（透明性）
評価基準	計画に対して	ビジネス価値（ビジネスの成果）
要求	管理可能、100%定義可能	管理不能、定義不能（標的は動く）
設計	機能中心設計	プロセス中心設計
開発	基本は個人（専門家）	チームの連帯責任
コード	属人化する	属人性の排除
品質	管理強化（当たり前品質）	向上可能性あり（魅力的品質）
ドキュメント	多種多様、管理基準	MRI（最低必要限）使用目的の明確化
デプロイ&リリース	半自動	自動化（デプロイメント・パイプライン）
運用	分権独立	オペレーションの一体化
運用管理	ITIL	軽量化したITサービス管理 & ISO20000
リーダーシップ	統率型（指示 & 命令）	サーバントリーダーシップ（ファシリテーション）

# DevOps 成功の最大要因は、行動習慣

## TMS (Toyota Management System)

は世界で唯一、行動習慣を変える実践的な手法としてテンプレート（改善塾）を提供



# When Toyota met e-commerce: Lean at Amazon



**Marc Onetto**

Former senior vice president of worldwide operations  
Amazon.com

<http://www.mckinsey.com/business-functions/operations/our-insights/when-toyota-met-e-commerce-lean-at-amazon>

(2014年)

“Amazonはトヨタに学び勝組になり、  
日本は海外に学び負組になる。皮肉で  
は済まされない…国家の危機です。”

– 賢者の独り言

# 何故Opsから定義するのか？

- ITサービスは開発しただけでは、利益を生まない
  - 運用して初めて利益を生む
- 運用コストが肥大化したITサービスを守る意義があるか？
- 運用コストは開発時の制約事項にふくめなければならない。
- 運用のインシデントの40%が開発フェーズの手抜きに起因する。 昨年のitSMF Japan Expoの発表より



**DevOpsのスピード感を実現するには、  
従来のITSMでは無く、軽量化されたITSMが必要である。**

# 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



# 全球运维大会

## 2016

### DevOps 2.0: 重塑运维价值



北京站

GOPS

GOPS2016  
Beijing

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 Data Center Alliance

主办单位: 高校运维社区 HealthOps Community, OCPA OpenOps Alliance

协办单位: IQ24 学院, 中国软件网, XIN, 中国软件网

高级赞助: 腾讯云, 金山云

钻石赞助: CD, 网宿科技, broader, 阿里云, SpeedCloud 迅达云

金赞助: 博云, 中企云联, LinkedBio 贝康, hli 光耀无限, 云杉网络, P, 用友网络

银牌赞助: 日志易, 用友网络, Steadyops, 齐治科技, NSFOCUS, 北京信奥, 精英, 用友网络, Vercloud 日康科技

<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

Gold Sponsors  
Silver Sponsors  
Bronze Sponsors  
Community Sponsors

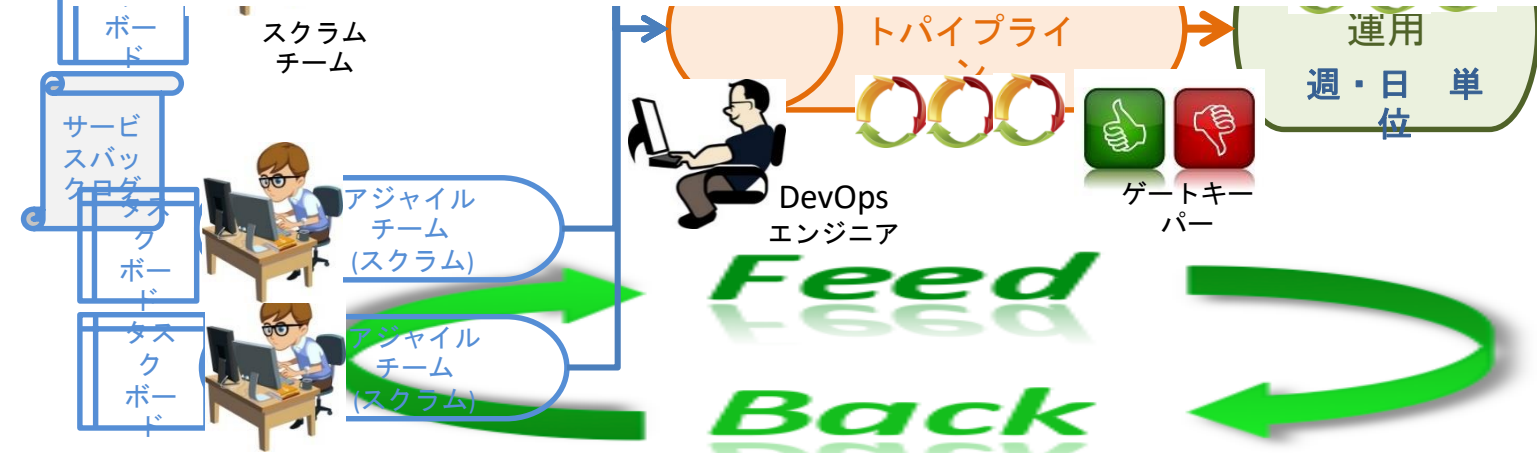


Patrick Debois  
DevOps 之父  
Friday, Mar 17, Saturday, Mar 18, 2017



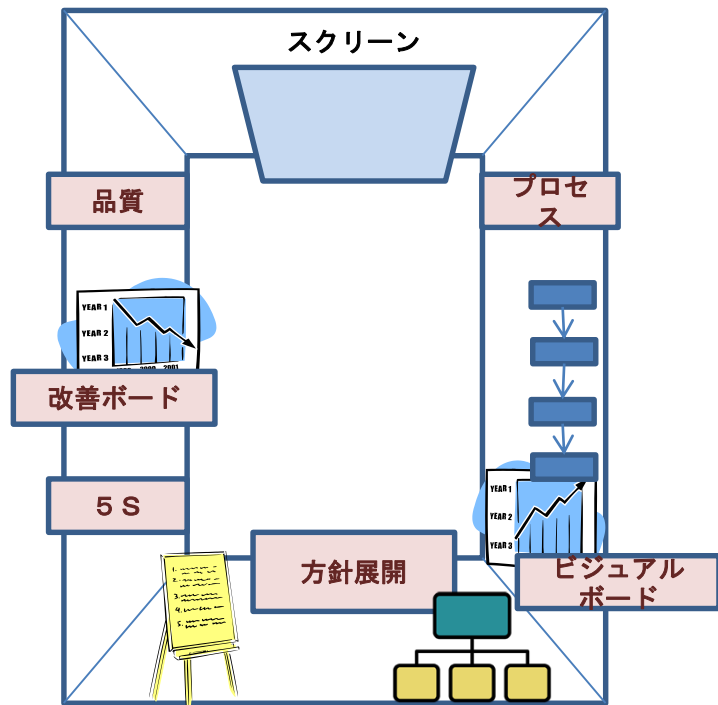


# 計画から運用までの全体最適による流水化されたプロセスの構築



# 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（RPAの導入、自動化）が重要
- ◆ ...



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

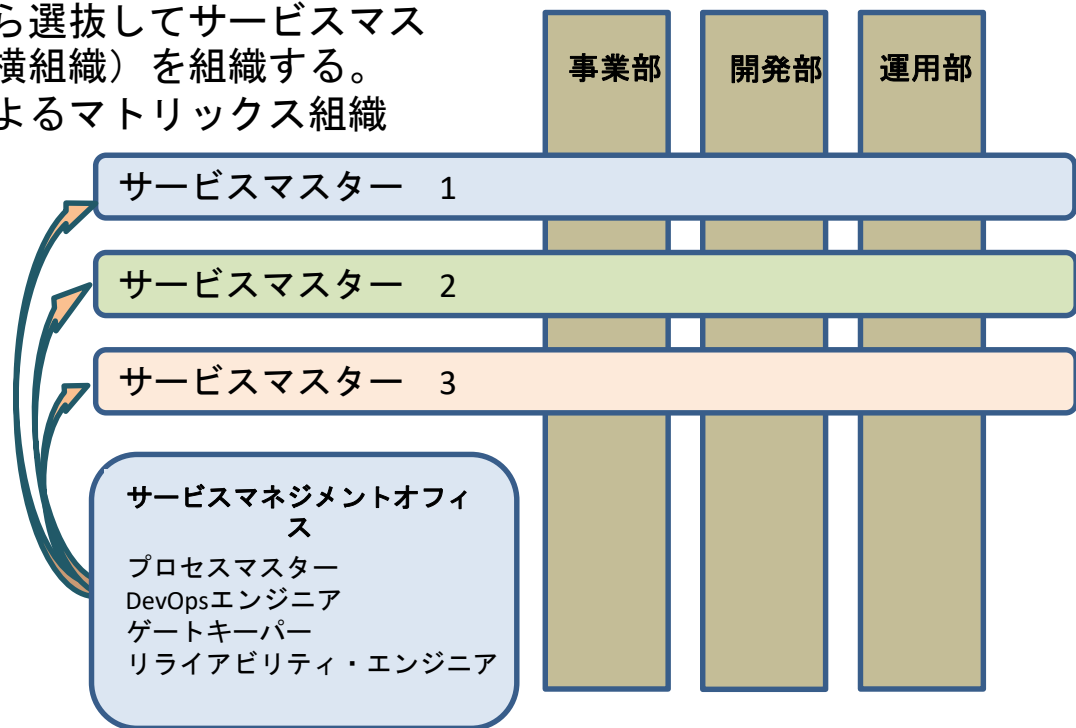
同じ部屋で仕事を行う事でコミュニケーションが活発化しアイデアが沢山出るようになる



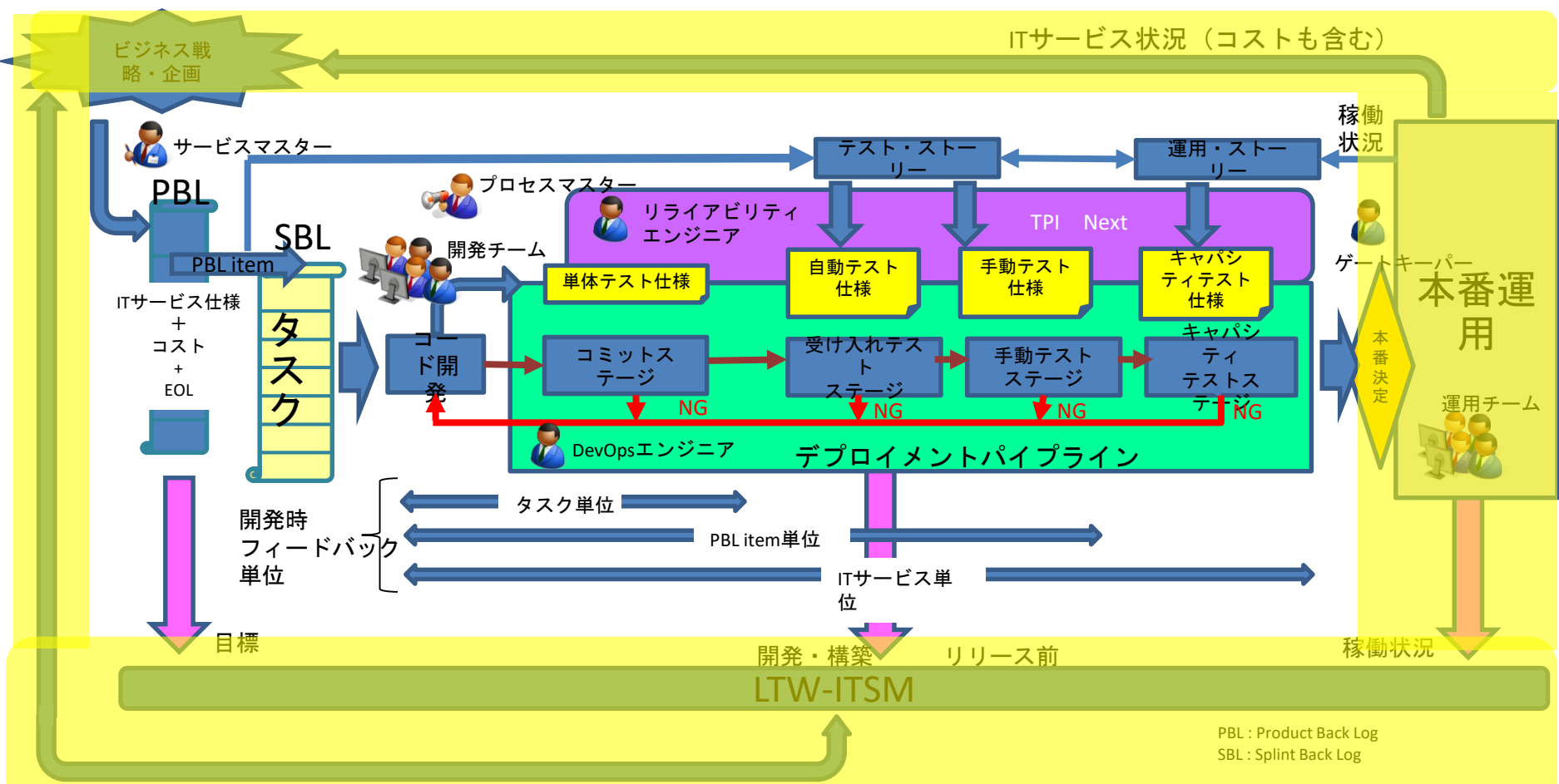
# DevOps組織（一例）

## 大規模かつ複雑な組織へ対応したマトリックス構造

各部門（縦組織）から選抜してサービスマスター配下のチーム（横組織）を組織する。  
トヨタの主査制度によるマトリックス組織

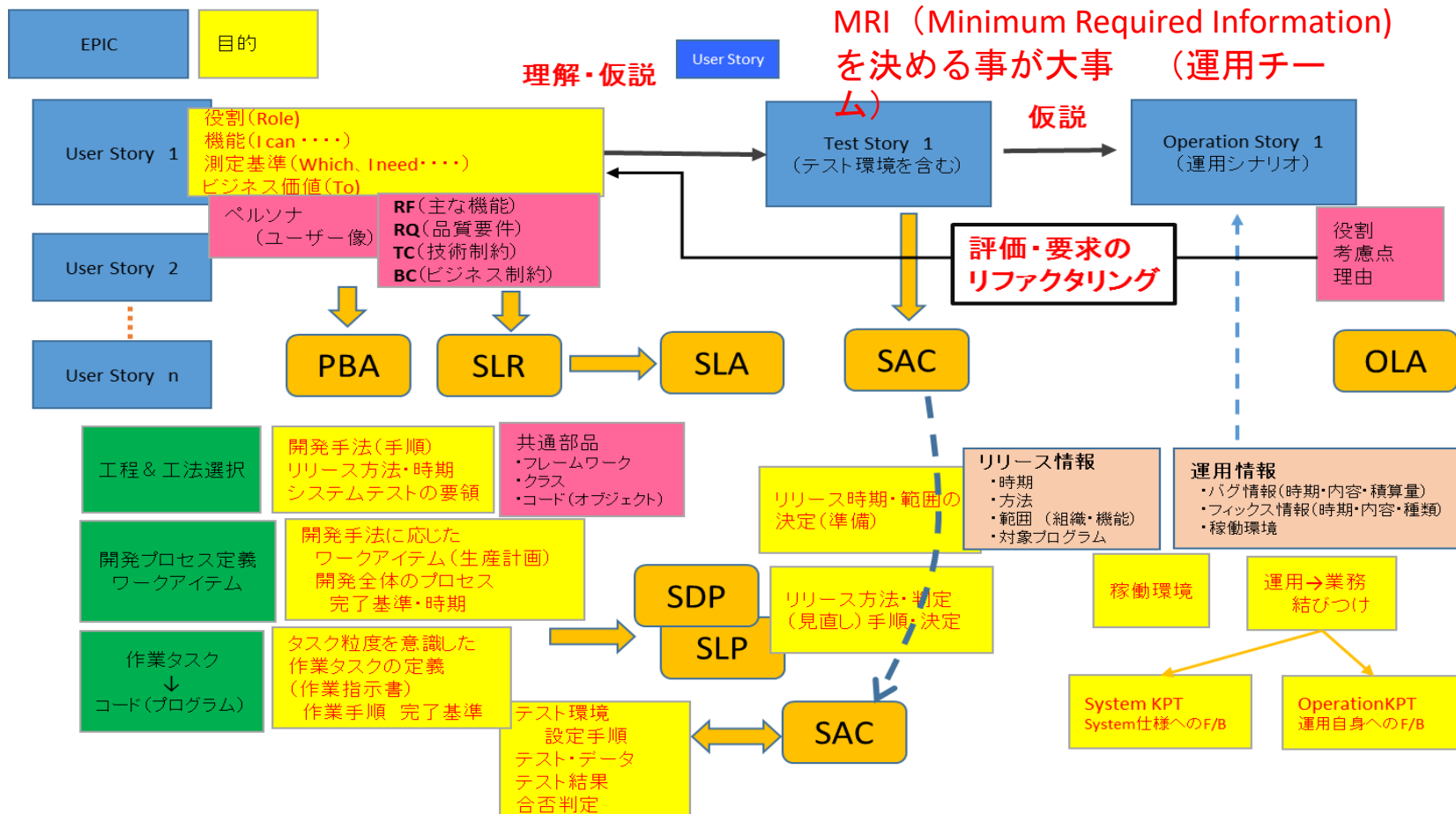


# 軽量化されたITSMの範囲



PBL : Product Back Log  
SBL : Splint Back Log

# 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



各作業現場では、正しく作業が行なわれているならば、それなりの情報が創出されている。

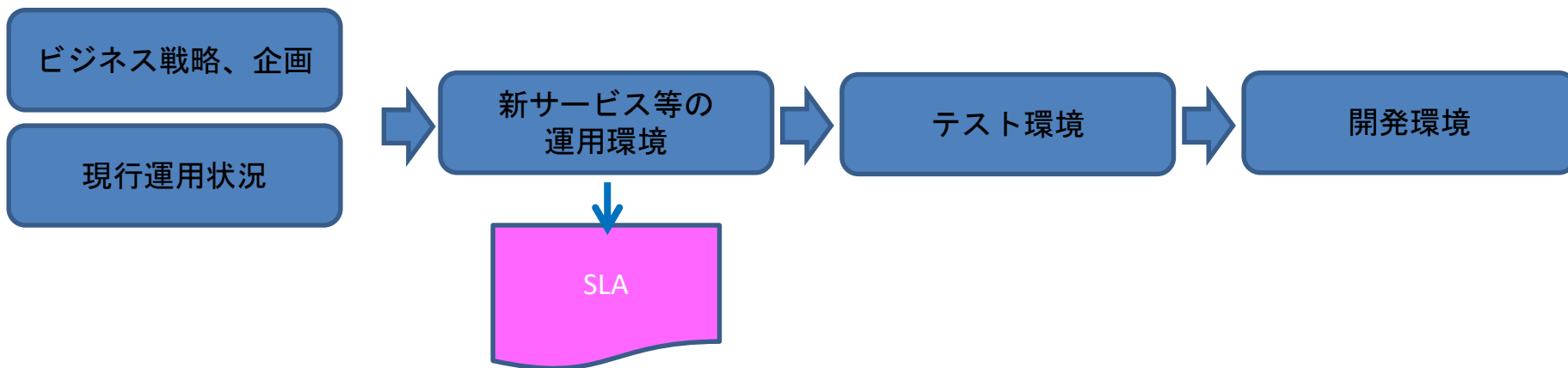
# Opsから要望する情報

- ビジネスの継続性を保証するために必要なもの
  - プロジェクトチャーターとして
    - 本番環境の定義、サービスの定義、サービス時間、機能性、サービス可用性、信頼性、セキュリティ、サービスの継続性、非機能要件 . . .
    - PBA (Patterns of Business Activity) 、SLR(Service Level Requirement)、SLA(Service Level Agreement)、 . . . 同等
  - サービスの受け入れ基準として
    - 非機能要件、運用リスク、緊急時・異常時対応、負荷パフォーマンス、セキュリティ、モニタリングと測定、運用コスト、 . . .
    - SAC (Service Acceptance Criteria)
  - サービス定義として
    - サービスのプログラム、オブジェクトのセット、テストの手法と仕組み、リリース計画、パイロットの基準と期間、サービスの保証期間 . . .
    - SDP(Service Design Package)、SLP(Service Level Package)

# 後工程引取りでの環境設定

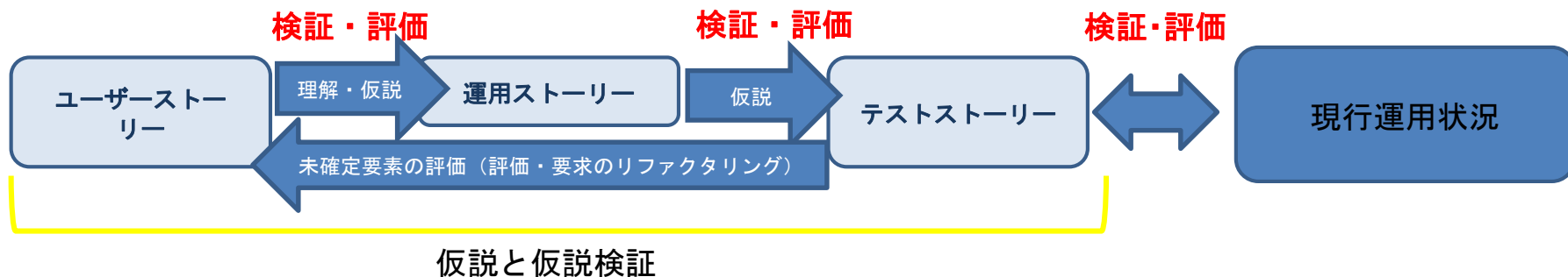
(運用起点での早期制約策定)

- 企画、計画段階の早期に現在の本番運用状況を鑑みて、ビジネス戦略、企画から想定される、新サービス等の運用環境を定義します。
  - 新しい運用環境に依存する、セキュリティ、パフォーマンス、キャパシティ等といった非機能要件に該当する項目を早期に制約として認識できる。
- 新サービス等の運用環境が決まれば、それからテスト環境を決め、最後に開発環境を定義する。

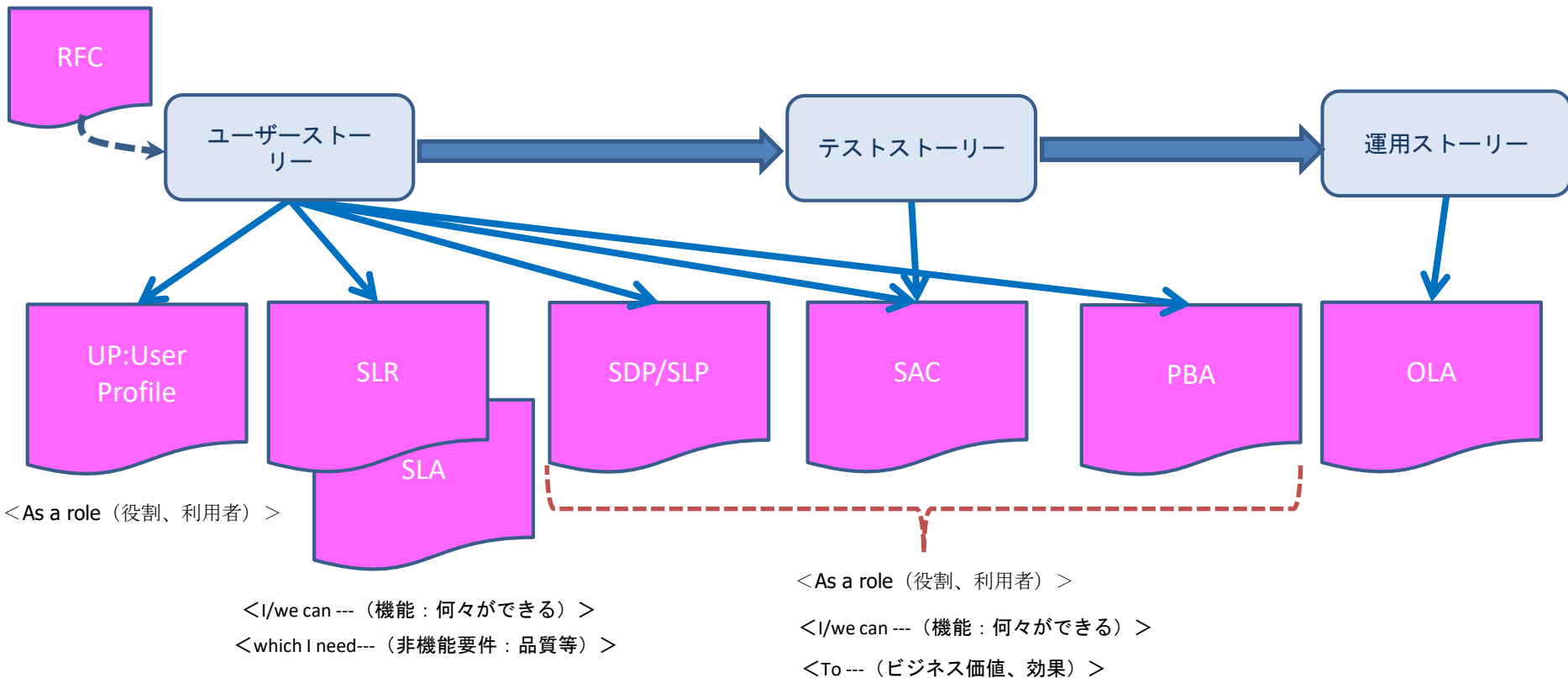


# 仕様要求の早期評価

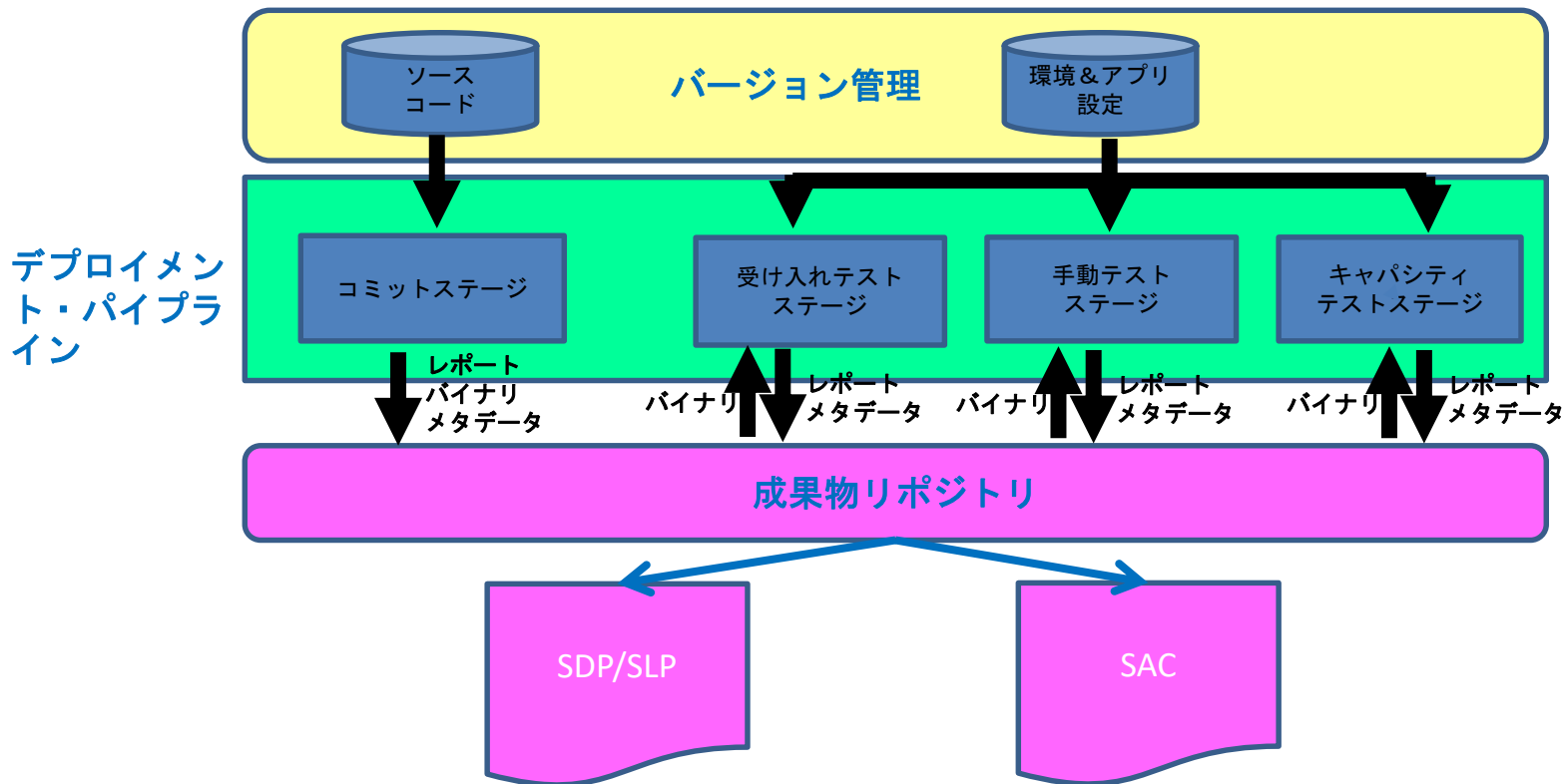
- ユーザーストーリーが明確になった時点で、そのサービス（プログラム）が実現された時の受け入れテストとなる、テストストーリー、と運用ストーリーを作成します。
- 運用ストーリーを現在の運用チームのスキル、運用環境のキャパシティ等で評価して、能力を超えたり、予測したコストを上回る様な場合には、もととなった、ユーザーストーリーをリファクタリングします。



# 各種ストーリーからの管理情報を得る



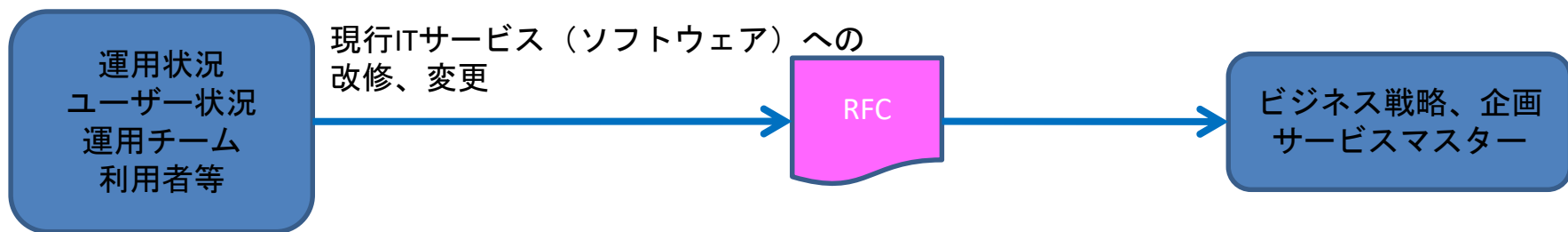
# 開発状況から管理情報を得る



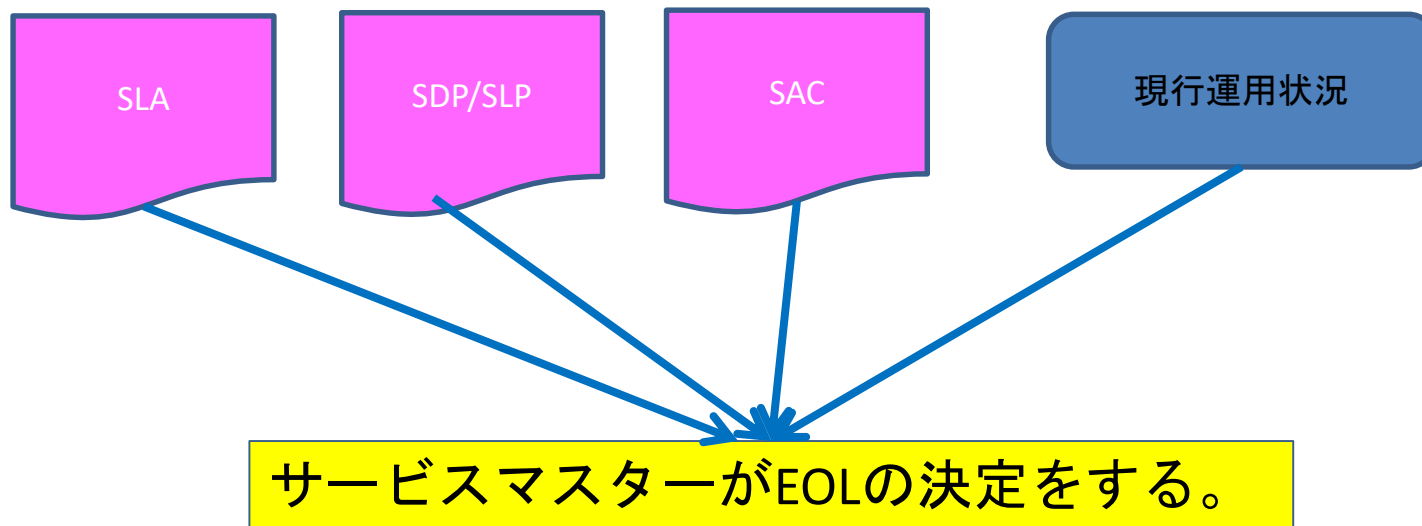


# 運用現場、ユーザーからの変更・修正要望

- 稼働中のITサービス（ソフトウェア）への改修、変更、といった要望はRFCに起票して企画、開発へのフィードバックする。



# EOL (End of Life)の決定



# 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



規律あるアジャイル

継続的デリ  
バリー

軽量化した  
ITSM

↑  
タスク粒度  
完了基準  
継続的インテグレーション  
イテレーション(タイムボックス)

↑  
プロセスの自動化  
デプロイのパターン  
テストの自動化  
TPI NEXT®

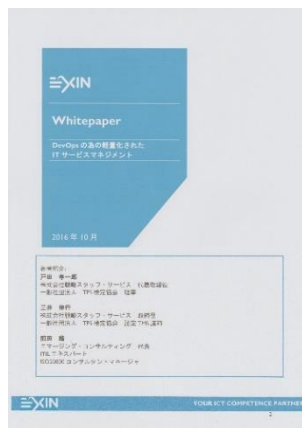
↑  
事業の継続性  
ISO 20000

TMS・TPS / リーン

↑  
ジャストインタイムと自動化、行灯  
自工程完結  
一個流しのフロー(平準化)  
学習する組織(反省、カイゼン)

# 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/>

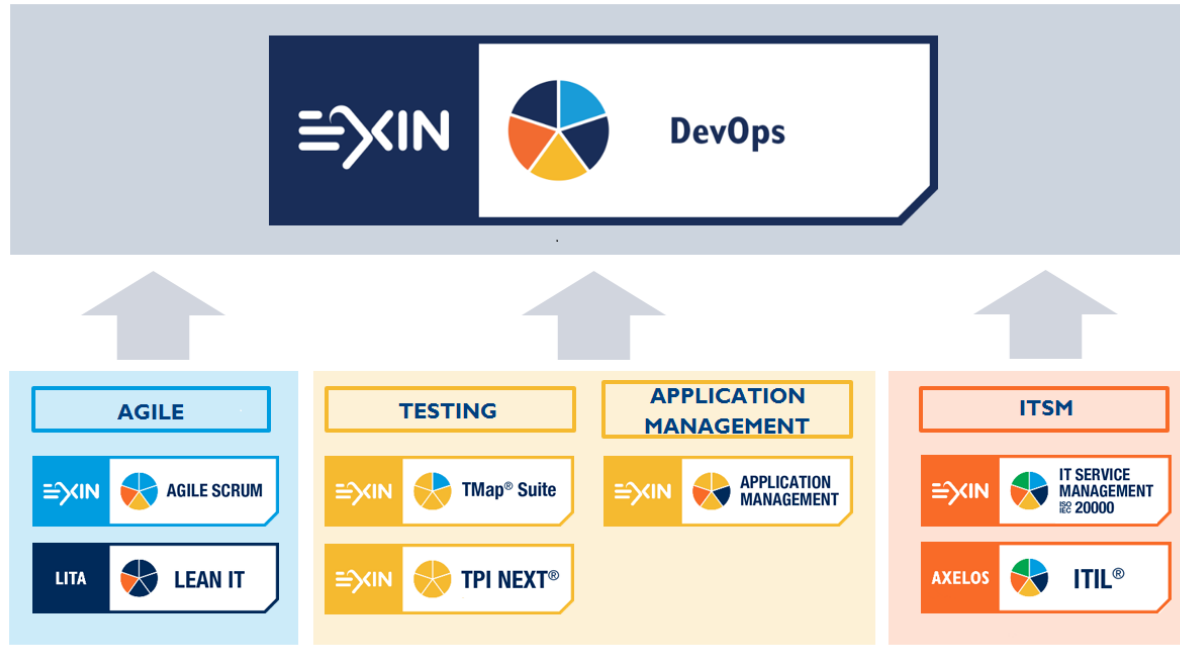
## メールマガジン連載；『全体最適へのDevOps（Agile, TPI, ITIL/ITSM）』

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>

# EXIN DevOps Master 認定



受験の対象者

- ❑ EXIN Agile Scrum Foundation
- ❑ TPI Next® or TMap Suite®
- ❑ EXIN IT Service Management Foundation
- ❑ LITA Lean IT Foundation
- ❑ EXIN Application Management Foundation

# DevOps Master開発の経緯

2015年夏： EXINジャパンよりDevOpsの認定プログラムの提案

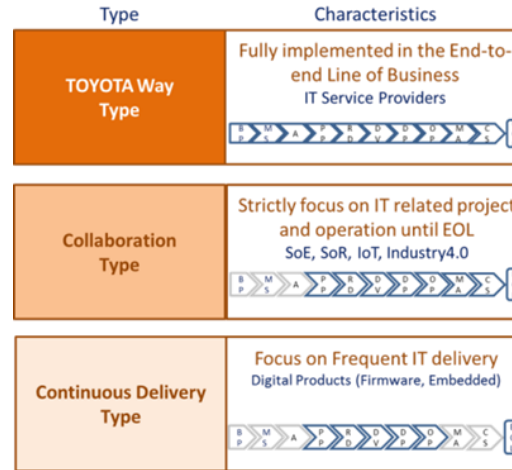
2016年1月： EXIN社内に開発プロジェクト設置(プロダクト・マネージャー)  
2016年2月： グローバル・チームにて検討開始  
DevOpsの定義、範囲を決定(日本案)

- ◆ 企業システムにおける成功事例を経験していた事
- ◆ DevOpsには、リーンの組織文化が重要な成功要因である事  
リーン概念の発祥と発展は日本が常に世界をリード = TPS (トヨタ生産方式)

2016年3月： DevOps全体プロセスの定義と基本的な知識領域の決定  
2016年5月： 認定試験問題の開発と受験者の勉強為の参考資料選定  
ホワイトペーパー作成 (Success with Enterprise DevOps)  
2016年7月： 日本国内にて試験問題のβテスト実施(35名のボランティア)  
2016年8月： EXINよりグローバル(EU & US)に発表&リリース  
2016年10月： 中国でリリース  
2017年3月21日： 日本でリリース

# DevOps Master 認定プログラムの狙い

企業における種々なDevOps導入形態に対応する。



実務的(プラクティカル)な知識を検定する。

ビジネスを支援する。

ビジネス・スピードに対応

事業継続性の担保



# 認定試験について

試験の形式: コンピューター、もしくは試験用紙。 四択の設問

設問数: 50

合格点: 65%

テキスト・参考書: 持ち込み不可

試験時間: 120分 (2時間)

## 受験の為のトレーニング概要

DevOps及びアジャイル開発、ISO20000の基本的知識を有している事が前提です。  
これらの基礎知識は、eラーニングとか書籍でカバーできます。


### トレーニングの基本的パターン

DevOps及びアジャイル開発、ISO20000の基礎知識を既に習得されている人対象(2日間)

1日の理論的知識習得の講義

1日の実践的知識確認の講義/ワークショップ

# Certification landscape

		DevOps Institute	DASA
知識領域	数冊の本、 ホワイトペーパー	DOI教材	DASA教材
背景	eCFマネジメント領域、ガバナンス～ITSM	Agile+ITSM	Agile+Operation
スコープ	Complete application or service life cycle, organizational and technical aspects	DevOps環境の役割 組織と技術, F; ビジネスと組織 A; 技術	F: 145spec, 50% Agile/Lean, A; 態度とリーダーシップ
Multiple languages	✓	✗	✓
評価	MC試験、実践課題	MC試験	MC試験
Competente Framework	✓	✗	✓
Balance of development, deployment, and operation	✓	✗	✗

## 参照文献－1

### **The Phoenix Project**

–A Novel about IT, DevOps, and helping your business.

Gene Kim, Kevin Behr, George Spafford. ISBN9784822285357  
(邦訳版) The DevOps 逆転だ！究極の継承デリバリー 日経BP社

### **Continuous Delivery**

–Reliable Software Releases through Build, Test, and Deployment automation.

Jez Humble, David Farley ISBN9780321601919

### **Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale**

Jennifer Davis, Katherine Daniels ISBN9781491926307

### **DevOps: A Software Architect's perspective**

(SEI Series in Software Engineering)

Len Bass, Ingo Weber, Liming Zhu. ISBN9780134049847  
(邦訳本) DevOps教科書 日経BP社

### **The DevOps 2.0 Toolkit:**

–Automating the Continuous Deployment Pipeline with Containerized Microservices.

Viktor Fracic. ISBN9781523917440

### **Architecting Software Intensive Systems (ACDM)**

–A Practitioner's Guide

Anthony J. Lattanze. ISBN9781420045697  
(邦訳本) アーキテクチャ中心設計手法(ACDM) 翔泳社

### **DevOps Automation Cookbook**

Michael Duffy. ISBN9781784392826

### **The Visible Ops Handbook**

–Implementing ITIL in 4 Practical and Auditable steps

Kevin Behr, Gene Kim, George Spafford. ISBN9780975568613

## 参照文献－2

### **TPI NEXT: Business Driven Test Process Improvement**

Alexander van Ewijk, Bert Linker, Marcel van Oosterwijk, Ben Visser, Gerrit de Vries, Loek Wilhelmus. Tik Marselis.

ISBN9789072194978

(邦訳版) TPI NEXT® ビジネス主導のテストプロセス改善 株式会社トリフォリオ

### **The TOYOTA Way**

Jeffrey K. Liker ISBN9780071392310

### **The DevOps Handbook**

Gene Kim, Jez Humble, Patrick Debois, John Willis ISDN9781942788003

(邦訳本) 今夏出版予定 日経BP社

### **EXIN White paper**

#### **Success with Enterprise DevOps**

Koichiro Toda, Nobuyuki Mitsui

(邦訳名) 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

### **Light weighted IT Service Management**

Koichiro Toda, Takashi Maeda, Nobuyuki Mitsui

(邦訳名) DevOpsの為に軽量化されたITサービスマネジメント

# 試験のシラバス

## 1. DevOps 概要 28%

- 1.1 DevOpsとは?
- 1.2 組織文化
- 1.3 原則と手法

## 2. 計画、要求定義、設計局面 18%

- 2.1 アプリケーション/サービス・ライフサイクル管理
- 2.2 プロジェクト憲章(スコープ)と見える化
- 2.3 インフラストラクチャーとアーキテクチャー設計
- 2.4 サービスレベル要求(SLR)とサービスレベルアグリーメント

(SLA)

2.5 テスト戦略の導入; ユーザー・ストーリー、テスト・ストーリー、運用ストーリー

## 3. 開発と展開局面 30%

- 3.1 継続的インテグレーションと継続的デリバリー
- 3.2 デプロイメント・パイプライン
- 3.3 継続的デプロイメント
- 3.4 自工程完結、リズム、一個流し
- 3.5 自動化ツールとテストिंग

## 4. 運用局面とスケーリング 22%

- 4.1 データ、インフラストラクチャー、環境、構成品と前提条件
- 4.2 構成管理と世代管理
- 4.3 クラウド環境、イミュータブル・インフラストラクチャー
- 4.4 事業の継続性
- 4.5 スケーリング

## 5. EOL管理 2%

- 5.1 製品/サービスのEOLの条件

Total 100%

DevOps Master 日本語試験リリース 3月21日より

アイ・ラーニング社開催の対策講座と試験受験

第一回: 3月28日(火)－29日(水)

第二回: 5月25日(木)－26日(金)

第三回: 7月27日(木)－28日(金)

第四回: 9月25日(月)－26日(火)

「DevOpsマスター認定試験コース」

<http://www.i-learning.jp/service/it/devops.html>

[https://www.i-learning.jp/products/detail.php?course\\_code=DO005](https://www.i-learning.jp/products/detail.php?course_code=DO005)

## DevOps Master試験問題のサンプル

(Q1) CTOは、DevOpsを導入する際にある種のリーンの概念を適用すると最も効果的であると考えています。  
次のリーンの概念または手法の中で、DevOpsを導入する時に最も効果的であるのはどれでしょうか？

(A)カイゼンと5S。

アジャイルやDevOpsはリーンの主要な概念に基づいている。そしてカイゼンと5Sはリーンの基本である。それらがDevOps導入に際して最も効果的である。

(B)先行カイゼン。

DevOpsでは、運用から開発へのフィードバックが要求される。先行カイゼンは、上流へのフィードバック・ループを作る。DevOpsにこの原則を適用するとフィードバック・ループを支援する。

(C)大部屋システム。

DevOpsは、異なった管理形式のプロセスを統合する。大部屋システムは、全体プロセスの見える化を支援し、DevOps導入を成功させる。

(D)一個流しと自工程完結。

DevOpsは上流工程の構築と単一のバリューストリームフローから効果が出る。一個流しはこれを可能にし、自工程完結は効率化とそのフローを作るのに役立つ。

(Q1) 解答解説:

(A) 不正解:

リーン、アジャイルとDevOpsが相互に関係しているにも関わらず、カイゼンと5SはDevOpsの開始の成功を支援する最適な組み合わせではありません。DevOps導入後、カイゼンは継続的な改善のために利用されます。そして5Sは、良い手順を維持するために使用されます。どちらもDevOps導入が成功した後になります。

(B) 不正解:

フィードバックはいつでも歓迎されます。しかしこれがDevOps導入時にリーンの最も効果的な運用を保証するものではありません。

(C) 不正解:

見える化は役立ちますが、DevOps導入時で最もインパクトの強いリーンの手法ではありません。

(D) 正解:

実行可能な、単一の、デプロイメント・パイプラインの構築はDevOps導入を成功させるための手助けとなります。DevOpsで最も重要なことは、開発から運用までの上流工程を構築することです。特に単一のデプロイメント・パイプラインのために必要です。自工程完結はこの整流化を実現する最も効果的な仕事の行動習慣です。

(参考文献: Success with Enterprise DevOps)



(Q2) 軽量化されたITSM(ITサービスマネジメント)とは何でしょうか？

- (A) 事業継続性に重点を置いたITSM
- (B) 標準として提案された新しいITILバージョン
- (C) ITILプロセスの不十分な導入
- (D) リリース管理を志向したITSM

(Q2) 解答解説:

(A) 正解:

ITILは重く、DevOpsの素早いプロセスには向いていません。軽量化されたITSMは一連の最小限必要な情報(MRI)を持った事業継続性に重点を置いたDevOpsのために再編成されました。

(参照文献: Success with Enterprise DevOps - セクション4 iii「ITサービスマネジメント」)

(B) 不正解:

まだそのようなITILバージョンは提案されていません。

(C) 不正解:

軽量化されたITSMとは簡素な不十分な導入ではありません。むしろ事業継続性と管理負荷軽減に特化した要点を掴むバージョンです。

(D) 不正解:

ITSMは、サービスマネジメント志向であり、リリース管理ではありません。ITSM概念では、リリースはサービスを補強するプロセスです。

(Q3)新しいプロダクトのためにあなたのチームはデプロイメント・パイプラインを作る必要があります。継続的インテグレーションの一部として、あなたはパイプラインの中でコミット・ステージを定義する必要があります。あなたはあなたのチームの仲間とこのステージについて議論します。

プロセスマスタ(管理者)が言いました。「完了の定義(DoD)は、コミットステージの前または最中に定義すべきだ。コミットした時にコードが完了していない場合、作業を停止すべきである。」

これは正しいでしょうか？

(A)はい、もし仕事が完了していなければ、プロセスマスタは職務を果たしていません。この問題は早急に解決すべきです。

(B)はい、顧客に付加価値を与えていないため、完了していない仕事は、コミットすべきではありません。

(C)いいえ、完了の定義とは、顧客との会議でのみ定義できます。完了の定義を待つことはあまりにも仕事を遅らせます。

(D)いいえ、デプロイメント・パイプラインでの仕事は、常に継続されるべきものです。もしコードが完了していなければ、それはただ停止する必要があります。

(Q3) 解答解説:

(A) 不正解:

プロセスマスタは、完了の定義と完了していないコードがコミットされたとき、作業を停止すべきであるという事を確実にするという職務があります。しかし、コードが完了していないコミットされた場合でもプロセスマスタが必ずしも職務を果たしていないということにはなりません。

(B) 正解:

仕事が完了していないとき、デプロイメント・パイプラインの仕事を開始することは、顧客に十分な価値をもたらしません。一個流しを考えれば、これはより価値のある仕事の流れを遅らせるでしょう。

(参考文献: Continuous Delivery, Chapter 3)

(C) 不正解:

完了の定義とは、プロジェクトにおいて合意された重要な事の一つです。それは、顧客との会議中に定義されません。コーディングを開始した時、常に完了の定義を意識すべきです。さもなければ何時コーディングを止めるかが解らないでしょう。

(D) 不正解:

コードに何か問題がある、または価値を付加できないというのは、デプロイメント・パイプラインを停止させ修復する、あるいは1個流しのパイプラインでもっと価値のあるものを得るための十分な理由です。

(Q4)開発チームはDevOpに興味を持っています。彼らは特に継続的インテグレーション(CI)に興味を抱いてました。彼らは現在3つの主要なソリューションと4つの小さなソリューションの開発と保守を担当しています。彼らはスクラム手法を用いています。それぞれのスプリント(イテレーション)は4週間で、平均的に1つのコミットされたリリースからテスト環境までにそれぞれ10~15日かかります。そして1つリリースから稼働まで1か月かかります。彼らは、CIを作成する努力や彼らの投資を支える管理のために定性的なビジネス・ケースを作りたいと望んでいます。このビジネス・ケースを最も支援するCIの具体的な利益(メリット)はどれでしょうか？

(A)1日に一回デプロイからテスト環境まで実行できればビジネス的メリットの増加と大幅な開発コストの削減ができる。

(B)チームの精神を支援する。すでに彼らはスクラムを用いており、CIはビジネスの目に見える利益を生まない。

(C)余分な費用を抑えるためにリリース速度を維持しながら、優れた統合テストでビジネスの安定性を高める。

(D)1日に一回のリリースから稼働までの実行は、ビジネスの利益(メリット)を増大させ、大幅な開発コストの削減ができる。

(Q4) 解答解説:

(A) 不正解:

テスト環境へ素早くデプロイするのは良いし CI の結果であるが、ビジネス上のメリットは作成されません。

(B) 不正解:

スクラムを使用するかどうかは関係なく、CIは、少ないコストでバグを素早く見つけたり稼働までの早いデリバリーを支援することができます。

(C) 不正解:

リリース・スピードを維持することは、DevOpsの、特にCIからの期待される効果ではありません。バグを速やかに見つけて修正し、稼働への増加されたリリースからコストを削減します。

(D) 正解:

早いリリースからの稼働はCIの主要な効果の一つです。同様にバグを素早く見つけることはバグ・フィックス費用と開発コストを削減します。  
(参考文献: Continuous Delivery, Chapter 3 - Continuous Integration)

ご清聴ありがとうございました。



ITは、新3Kの時代に!!



Copyrights©2017 SSS Corporation

YOUR ICT COMPETENCE PARTNER