

Internet Week 2015 T1 クラウドネイティブ時代のインフラエンジニア

クラウドネイティブ時代のエンジニアスキル

運用設計ラボ合同会社

シニアアーキテクト 波田野 裕一

2015-11-17

アウトライン

クラウドネイティブ時代は、ITが本当に「道具」(手段)になる時代だと言われ、従来の道具が重要な「手段指向」からよりビジネスに近い「目的指向」に変化してきています。今後インフラエンジニアに求められる考え方やスキルについて解説していきます。

1. クラウドネイティブ時代のエンジニアを取り巻く「変化」

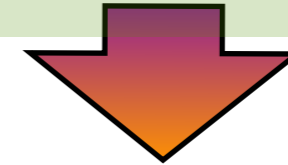
1. クラウドネイティブ時代のエンジニアを取り巻く「変化」

1. 「技術だけ」では難しくなる

激変するインフラ環境

1 番目のセッションの再掲

クラウド時代のインフラは、そもそもサーバやインスタンス(それらを接続するネットワーク)を意識する機会が激減していく。



以下の業務価値についてコモディティ化が進む。

- ・ サーバやOSの管理やその保守
- ・ サーバやOSの初期導入作業、変更作業
- ・ サーバやOSの安定運用、信頼性向上
- ・ サーバやOSの高拡張性設計

(アプライアンスを含む。)

どこかで臨界点を超えると、ネットワークエンジニアやサーバエンジニアは「普通のインフラエンジニアでいられなくなる」時がくる。

インフラエンジニアの「専門性」の変化 (before)

- かつては、特定のハードウェアやその中で動作する特定のミドルウェアやアプリケーションを安定して動かすことに専門性があった。
- 自分の管理する個々のハードウェア、ソフトウェアに対してつくり込みをして、徐々に愛情を育んでいった。
- インフラ環境の可用性、安全性、信頼性を重視し、どちらかというに変化を伴う要望に対して否定的であった。

従来: 手段(プロダクト)指向

手段 の提供(コストセンター扱い)
道具のお守り (利用目的は気にしない)

インフラエンジニアの「専門性」の変化 (after)

1 番目のセッションの再掲

- ・ IT資産は本当の意味で随時調達となる。
 - ・ 作り込みから使い捨てへ
- ・ IT資産を使って「何をしたいのか」が常に問われる。
 - ・ 手段重視から目的重視へ
- ・ 変化の激しい時代に常に求められるのはスピード感。
 - ・ 確実性よりも迅速性

今後: 目的(課題解決)指向へ

目的 の実現(プロフィットセンター扱い)

道具は単なる「手段」

インフラエンジニアの「専門性」の変化 やり方が変わる

- ・ 作り込みから使い捨てへ
- ・ 手段重視から目的重視へ
- ・ 歴史的経緯よりも合理性
- ・ 確実性よりも迅速性
- ・ 「運用でカバー」から《設計で保証》へ
- ・ 主観性よりも客観性
- ・ テクニカルよりもエンジニアリング
- ・ やりっぱなしモデル(作りっぱなし)からサイクルモデル(スクラッチ&ビルド)へ

アメリカでは「ペットモデル」から「家畜モデル」への変化との表現も出てきています。

インフラエンジニアの「専門性」の変化

今後: 目的(課題解決)指向へ

目的の実現(プロフィットセンター扱い)

道具は単なる「手段」へ

- ・ 作り込みから使い捨てへ
- ・ 手段重視から目的重視へ
- ・ 確実性よりも迅速性

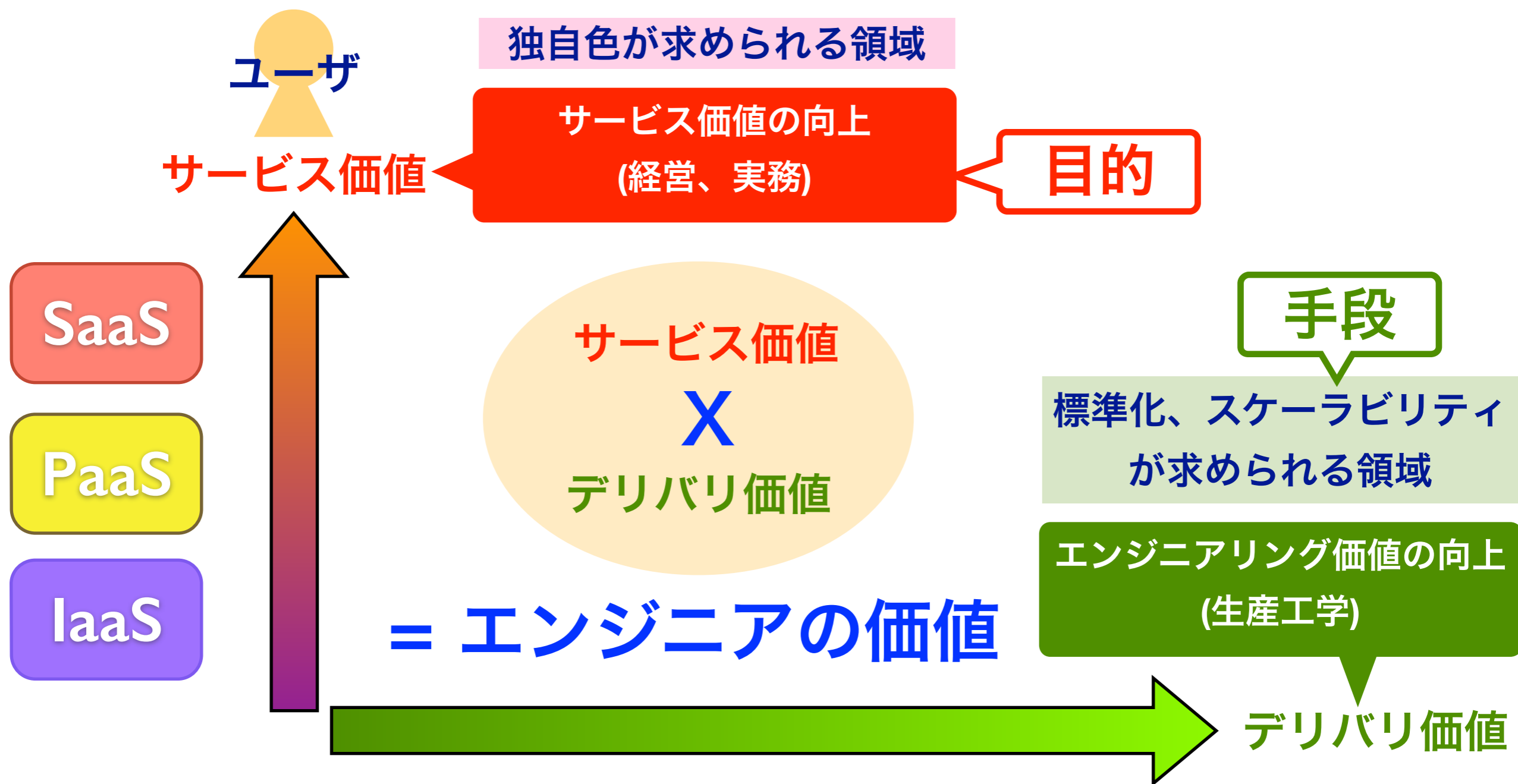
目的(課題解決)指向とは

ユーザの「課題」を解決すること

- ・ エンドユーザの課題を解決すること
- ・ 社内ユーザの課題を解決すること
- ・ 誰かの課題を解決することを(間接的に)支援すること

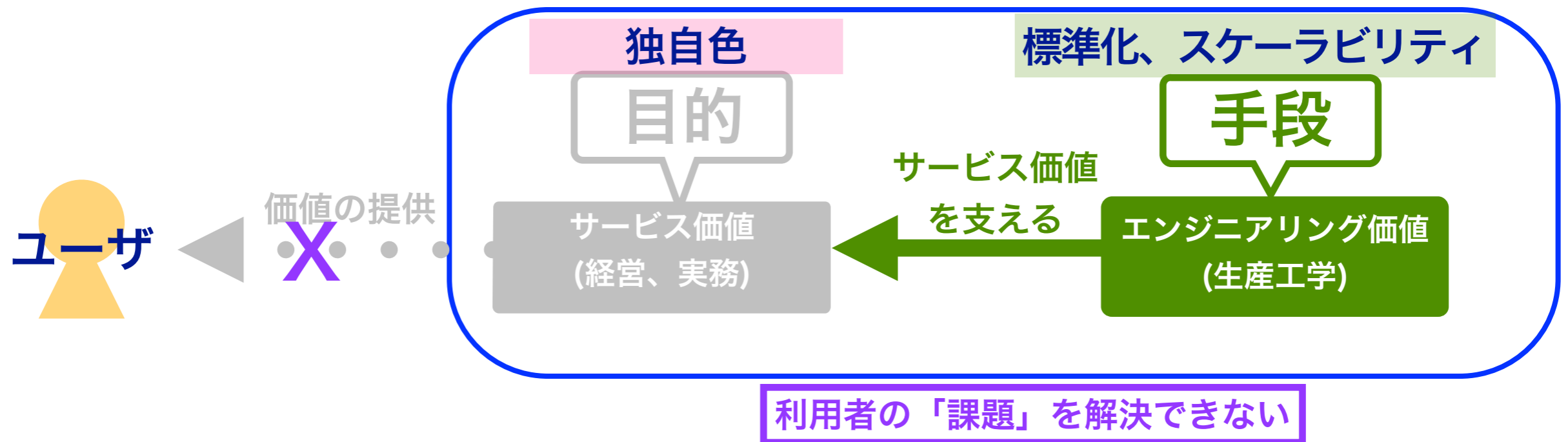
エンジニアの全ての業務は、課題を解決するための
「サービス」であると捉えることができる

目的指向と手段指向のバランスが重要



でも手段より目的が大事

どちらの価値も重要だが……



サービス価値が無いと「手段」の存在意義がない

1. クラウドネイティブ時代のエンジニアを取り巻く「変化」

2. スキルの客観化が進む

スキルの客観化 (クラウドの特徴)

スキルの囲い込み(秘伝)はできない時代に

- ・ 従来は、特定のハードウェアやミドルウェアやアプリケーションの経験を積むには、それらを保有する現場での経験や経験者からのスキル転送が必要だった。
- ・ クラウドネイティブ時代は、誰でも同じ基盤に触れることができるようになった。
- ・ クラウド上の構築スキルについては、出題(特定の要件)に対してどんな設定(JSON)ファイルを書いてくるか、で客観化が可能。

スキルの客観化 (事業者/コミュニティによる共有)

技術的なトピックはコミュニティで広く共有

- ・ クラウド事業者が積極的に技術的なトピックやノウハウを公開。有償サポートの有無を問わず公式情報を得やすい環境になった。
- ・ 技術の変化が激しいため、書籍などの出版物は内容が周回遅れになる傾向があり、コミュニティ内で得られる情報が最速かつ正確である傾向が強まっている。(基礎を固めるには書籍はやはり貴重)
- ・ 情報を発信する人に情報が集まる傾向に拍車。(「囲い込む」よりも「アウトプットをしてブラッシュアップしてもらおう」方が効率的に)

スキルの客観化（ドキュメント能力）

技術を「より適切に活かす」ための客観化

- ・ 技術の囲い込み(秘伝)が難しい時代には、その「技術を扱える」と自体の価値は早期に陳腐化してしまう。(誰にでもできるようになってしまう)
- ・ 技術を使える人に何をしてもらうか、技術を駆使して何を実現するか、を客観的に伝えられる人の価値が高くなる。(生産ラインでの熟練工よりも、生産ラインの設計者が求められる時代)
- ・ あらゆる知識や要求を客観化(ドキュメンテーション)できる人の価値が高くなる。(特にクラウド時代は意図しておかないと失なわれる情報も膨大になっていく。)

参考: Internet Week 2011 T3 「運用ドキュメント2011 ～ 手軽にスピーディに継続的に保守するためのツール入門」

1. クラウドネイティブ時代のエンジニアを取り巻く「変化」

3. 「自前主義」では難しくなる

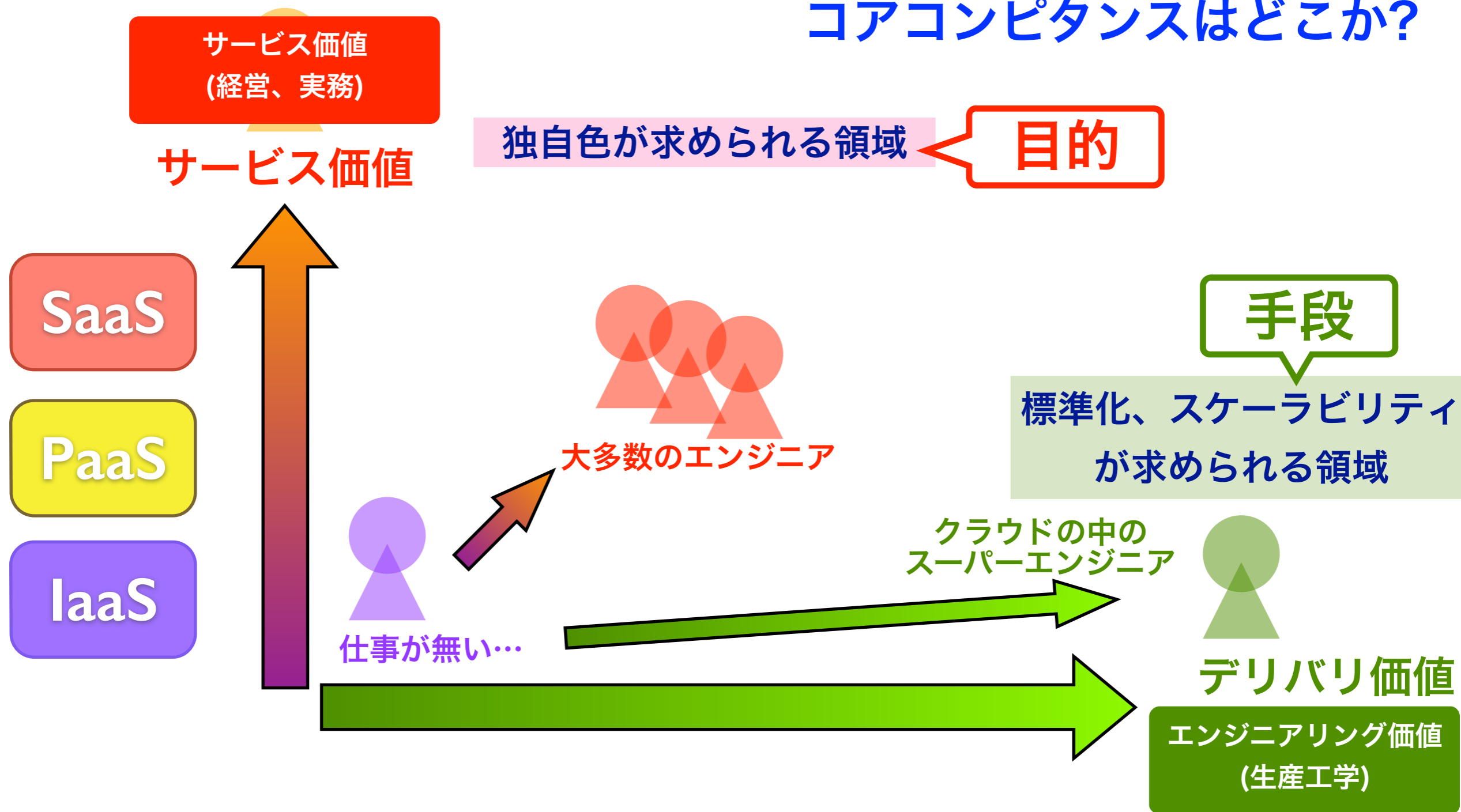
組織レベルでの脱自前主義

- ・ インフラを数百ノード規模で構築運用をしても価値が認められにくくなった。
- ・ 全ての業務プロセスを自前で実施するよりも、**自組織のコアコンピタンス以外のプロセスは専門家に任せる方が、スピード・コスト・品質ともに勝るケースが増えてきている。**
- ・ **非コアコンピタンスをマネージドサービスに任せ、自組織に有効なリソースをコアコンピタンスなプロセスに集中する時代へ。**

そのインフラ構築・運用は自組織のコアコンピタンスか？

これからのインフラエンジニア

コアコンピタンスはどこか？



個人レベルでの脱自前主義

- ・ 今後は、どんなエンジニアも幅広くかつある程度深い知識が求められる。(「ここは専門外」という決め付けがしづらくなってきた。)
- ・ イノベーションは、ある程度知識的な距離が離れているもの同士の融合により生まれる、と言われている。
- ・ 一方で、独りで把握できる技術領域にはやはり限界があるため、技術の動向や価値を把握するためには、相互にキュレーションできる仲間が必要。(特にセキュリティ関連は迅速な対応が求められる)。

2. クラウドネイティブ時代のエンジニアに求められる「スキル」

2. クラウドネイティブ時代のエンジニアに求められる「スキル」 エンジニアに求められるスキル

- ・ 会計スキル (カネのスキル)
- ・ 経営スキル (ヒトのスキル)
- ・ 技術スキル (モノのスキル)

2. クラウドネイティブ時代のエンジニアに求められる「スキル」

1. 会計スキル (カネのスキル)

2. クラウドネイティブ時代のエンジニアに求められる「スキル」

1. 会計スキル (カネのスキル)

よほど尖がったスーパーエンジニア以外は、**会計に関する基礎知識が不可欠になる。**

- ・ 「複式簿記」は世界最古の、紙の上で実現した、極めて完成度の高い「美しいシステム」
- ・ 自分達の業務が「売上原価」(直接配賦/プロフィットセンター)か、「販管費」(共通配賦/コストセンター)か、が評価や将来に大きな影響。
- ・ 「その作業」が売上や利益に繋がるのか、繋がらないが会計的に説明できるか、が行動決定の重要な指標の一つになっていく。
- ・ 土地以外の「資産」は、「将来的な費用が一時的に塩漬状態になっているだけ」なので、**過度に評価することは弊害につながる。**

「カネの話が全て」では無いが、「やりたいことをやる」には必要。

2. クラウドネイティブ時代のエンジニアに求められる「スキル」

1. 会計スキル (カネのスキル)へのステップ

よほど尖がったスーパーエンジニア以外は、**会計に関する基礎知識が不可欠になる。**

- ・ **ステップ1. 簿記4級で複式簿記の基礎を理解する。**
- ・ **ステップ2. フィナンシャルプランナー3級の入門書を軽く読む。**
- ・ **ステップ3. 会計クラウドサービスで個人会計をやってみる。**

「カネの話が全て」では無いが、**「やりたいことをやる」には必要。**

2. クラウドネイティブ時代のエンジニアに求められる「スキル」

2. 経営のスキル (ヒトのスキル)

2. クラウドネイティブ時代のエンジニアに求められる「スキル」

2. 経営のスキル (ヒトのスキル)

よほど尖がったスーパーエンジニア以外は、**経営(マネジメント)**に関する**基礎知識**が強く求められるようになる。

- ・ 安定したサービスを提供するための**サービスマネジメント手法**
- ・ 自組織やメンバーの内部的な強みや行動規範を分析・定義する**管理会計** (業績評価基準、強み分析などの分野)
- ・ 自組織の対外的な強みやユーザの動向を分析・定義する**マーケティング分析手法** (購買行動、製品戦略など)
- ・ 自組織の対外的、内部的な強みを維持・拡大していくための**企業経営理論** (人材育成、組織論)

「マネジメント」の専門能力を持つ人が日本には少ないが、「やりたいことを実現する」には必要。

2. クラウドネイティブ時代のエンジニアに求められる「スキル」

2. 経営のスキル (ヒトのスキル)へのステップ

よほど尖がったスーパーエンジニア以外は、**経営(マネジメント)**に関する**基礎知識**が強く求められるようになる。

- ・ **ステップ1. 情報処理技術者試験「サービスマネージャ」の知識を理解する。**
- ・ **ステップ2. 「管理会計」の概要を理解する。**
- ・ **ステップ3. 「企業経営」の概要を理解する。**

「マネジメント」の専門能力を持つ人が日本には少ないが、「やりたいことを実現する」には必要。

2. クラウドネイティブ時代のエンジニアに求められる「スキル」

3. 技術スキル (モノのスキル)

2. クラウドネイティブ時代のエンジニアに求められる「スキル」

3. 技術スキル (モノのスキル)

よほど尖がったスーパーエンジニア以外は、幅広い分野について、変化に迅速に付いていけることが求められるようになる。

- ・ クラウドサービスの利用経験は必須。(圧倒的なスピード感と簡便性と情報量を体感すべき)
- ・ 脱オンプレミスのための技術(特にコンテナ技術)は今後重要になる。
- ・ 落ちることが前提のシステム設計に慣れる。(リソースの使い捨て感覚。イミュータブルインフラストラクチャの実現化)
- ・ 今後は、API実装能力が構築・運用系エンジニアにも求められる。(ソフトウェア設計技術が必須に)

全てを自前で行うのは無理、だが可能な限り幅広く大量の知識を高速にインプットし、それを上回るアウトプットを行う必要がある。

2. クラウドネイティブ時代のエンジニアに求められる「スキル」

3. 技術スキル (モノのスキル)へのステップ

よほど尖がったスーパーエンジニア以外は、幅広い分野について、変化に迅速に付いていけることが求められるようになる。

- ・ ステップ1. 「クラウドサービス」のアカウントを作成して、入門ハンズオンに参加してみる。
- ・ ステップ2. VagrantやDocker-Machineでコンテナ技術を実際に動かしてみる。
- ・ ステップ3. Swaggerや、API Gatewayなどで、基本機能だけのRESTful APIを設計・実装してみる。

全てを自前で行うのは無理、だが可能な限り幅広く大量の知識を高速にインプットし、それを上回るアウトプットを行う必要がある。

まとめ

2. クラウドネイティブ時代のエンジニアに求められる「スキル」 まとめ

1. 会計スキル (カネのスキル)

よほど尖がったスーパーエンジニア以外は、**会計に関する基礎知識が不可欠になる。**

「カネの話が全て」では無いが、「やりたいことをやる」には必要。

2. 経営のスキル (ヒトのスキル)

よほど尖がったスーパーエンジニア以外は、**経営(マネジメント)に関する基礎知識が強く求められるようになる。**

「マネジメント」の専門能力を持つ人が日本には少ないが、「やりたいことを実現する」には必要。

3. 技術スキル (モノのスキル)

よほど尖がったスーパーエンジニア以外は、**幅広い分野について、変化に迅速に付いていけることが求められるようになる。**

全てを自前で行うのは無理、だが可能な限り幅広く大量の知識を高速にインプットし、それを上回るアウトプットを行う必要がある。

参考: スーパーエンジニアになる

いわゆるスーパーエンジニアかどうかは、
知っているか、経験しているか、が大きい。

※発表者はスーパーエンジニアではありません。

- まずは、とにかく「実際に触ってみる」ことが第一歩。
- 離れた分野をいくつか経験していると、差別化しやすい。
- ITについては、**抽象化能力**(設計能力)と**具体化能力**(実装能力)のバランスが重要。
- **痛い目にあった経験は特に得難い財産になる。**

「**どれだけ地雷を踏み抜いて、どれだけ血を流したか**」は、
エンジニアとしての価値に決定的な影響…

付録: 明日のエンジニアライフのために

明日のエンジニアライフのために

サービスエンジニアリングの力を持つ

▶ 疎結合 (UNIXの哲学/カスタマイザブル)

- ▶ 一つの事を上手にやるプロダクト、新規追加の仕組み、他組織との相互接続。

▶ 永続的 (事業継続性/商用製品に非依存)

- ▶ 災害に強く、ベンダーロックインされておらず、設計自体が引き継ぎ可能。

▶ フレームワーク (科学的/エンジニアリング)

- ▶ 理論と実践。再現性(たいていの人ができること)の重視。
- ▶ 反復性(定期的な棚卸しなど継続するための仕組み)の実現。

明日のエンジニアライフのために

サービスエンジニアリングに必要な知識

- ・ 抽象化手法 (特にオブジェクト指向分析)
- ・ カネの知識
- ・ モノの知識
- ・ ヒトの知識

明日のエンジニアライフのために

オブジェクト指向分析の知識

- **ヒト/モノ**(オブジェクト)が、
- **相互にメッセージ**で会話する、
- 一連の**相互作用**を把握/表現するもの。

※個人的な意識です

抽象化能力重要

明日のエンジニアライフのために

カネの知識

- ▶ **簿記** (ITの祖先。世界最古のシステムでもある。)
- ▶ **財務会計**
- ▶ **管理会計**
- ▶ **IFRS** (国際会計基準のうち資産負債アプローチ)

明日のエンジニアライフのために

モノの知識

- ▶ 生産工学 (100年以上の歴史)
- ▶ ソフトウェア工学 (品質の考え方)
- ▶ Web アプリケーション フレームワーク
- ▶ IaaS (クラウド、VPS)
- ▶ マネージドサービス
- ▶ デスクトップ自動化 (JavaScript/VBAなど)

ヒトの知識

▶ 経営学全般

▶ 経営学自体は主観的要素が強い。

- ▶ 自分はこう思う、に対する批評や事例研究が中心
- ▶ でも、決して無駄な議論ではないものも多い。

▶ 例えば、組織運営や生産性向上など。

難しいのは、聞く側のコンディションによって、
受け取り方が変わってしまうこと (経験あり)

明日のエンジニアライフのために

サービスエンジニア(アーキテクト)のミッション

運用現場の日常活動を「継続的」に「計測」し
「振り返る」ための仕組みを構築する。

- ・ 業務の構造化 (運用の構造を明確にする。)
- ・ 期待の客観化 (運用品質を定義する。)
- ・ 実績の測定化 (運用実績をデータに語らせる。)

明日のエンジニアライフのために

(参考) 電子書籍は重要

- * 専門書PDF (オライリ / 技術評論社 / オーム社 / 達人出版会、など)
- * 一般書籍 (KoBo / Amazon Kindle など)

- ▶ 午前3時でも**思い立ったら3分で購入**完了。
- ▶ 割安で効率良くまとまっている。(知識大量消費時代の到来、急速な選別化)
- ▶ **O'reilly の電子書籍全部読んできた新卒社員**など、中年の中堅エンジニアには危険すぎる存在。
- ▶ だが、「**学ぶ年寄り**」には**永久に追い付けないんじゃないか仮説** (豊富な前提知識と勘どころ)も十分に考えられる。
- ▶ **電子書籍読まない、のは**エンジニアには**ハイリスク**すぎる選択肢
- ▶ **ただし、「積ん読」の見えない化に要注意!**

Operation Lab

運用設計

<http://www.operation-lab.co.jp/>