

「運用エンジニア」のためのエンジニアリング入門 1

運用エンジニアのキャリアを考えよう

運用設計ラーニング

この講義の対象者

ジュニア～ミドルの運用エンジニア

- ・ キャリアについて**明確な展望が持っていない方**
- ・ 今後のキャリアに**迷いや悩みを持っている方**
- ・ キャリアについて明確な展望を持っているが、**参考になる情報を知りたい方**

運用エンジニアを指導・育成する立場(先輩や上司)の方

- ・ 後輩や部下の指導・育成について**明確な展望が持っていない方**
- ・ 現在、後輩や部下の指導・育成に**迷いや悩みを持っている方**
- ・ 後輩や部下の指導・育成に自信を持っているが、**参考になる情報を知りたい方**

悩み: 運用エンジニアのキャリア

- ・ 運用エンジニアとしてこのまま続けていていいのか？
- ・ 今後、どのようなキャリアを想定して、どんなスキルや経験を積んでいけばいいのか？
- ・ どのようなスキルや経験を積めば、どのような展望が開けるのか？

理想: 運用エンジニアのキャリア

ジュニア～ミドルの運用エンジニア

- ・自分のキャリアについて**明確な展望**を持っている。
- ・自分のキャリア展望における**現在の位置**を把握している。
- ・今後のキャリアに**不足している知識・経験**、**伸ばしていくべき知識・経験**を理解している。

運用エンジニアを指導・育成する立場(先輩や上司)の方

- ・後輩や部下に**現在不足している知識・経験**を把握し、適切に助言や指導ができる。
- ・後輩や部下の**伸びしろ**を把握し、**業務に合わせて適切に育成**できる。
- ・後輩や部下の**キャリアの展望**と**業務に合わせた育成計画**を立案・遂行できる。

この講座で得られるもの

- ・ **運用エンジニアの特性とキャリアパスの全体像**を把握できる。
- ・ 運用エンジニアの主戦場である「**運用**」と「**エンジニアリング**」に対する考え方を理解できる。
- ・ 上記2点から、**自分や部下・後輩が進むべき先が見通せる**ようになる。

講師自己紹介

はじめに

学習ガイド

本編

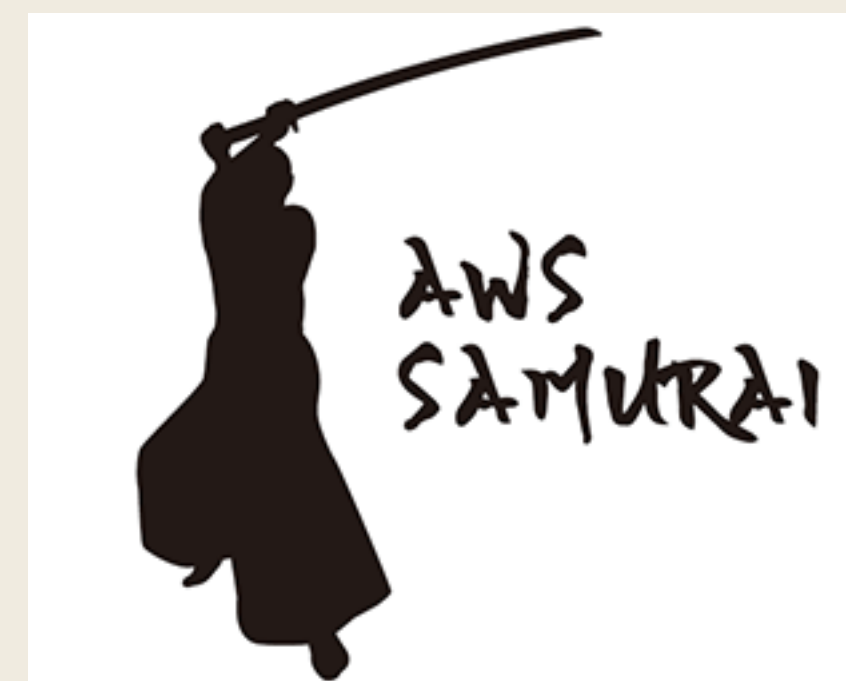
まとめ

ステップアップ

運用設計ラボ合同会社

シニアアーキテクト

波田野 裕一



AWS Samurai 2017 (個人)
AWS Samurai 2020 (CLI専門支部)



AWS Community Hero

- ・ 2000年 **ADSLキャリア**で開局業務、ISP運用
- ・ 2002年 **Sler**で官庁系サービスのサーバ保守
- ・ 2003年 **ASP**で基盤設計・構築・運用
 - ・ ジョブ基盤の設計構築 (ユーザ影響: 数千万人)
 - ・ 監視基盤の設計構築 (監視対象: 数万ポイント)
 - ・ 監視センタの構築運用設計 (アラート: 数百万通/年)
- ・ 2013年 **運用設計ラボ**を設立



エンジニア経歴



東京めたりっく通信入社

東京で最初に商用ADSLサービスを開始した会社

1年7ヶ月

電話局作業、ATM運用、IPネットワーク運用
いずれも未経験ながら、並行して担当していた。

退職で引き継ぎが必要になったため、
自動化した業務の全面手動化を行なった。

ASP 入社

大規模運用がやりたくて転職

9年9ヶ月

物理サーバー数千台 (毎月100台程度増える)

10人(5チーム)で構築・運用管理 (各チーム 数百～1千台程度)

監視システムと監視チームの設計・構築・運用

アラートが数百万/年だったのを数万/年に削減

ジョブ基盤の設計・構築・運用

影響範囲が数千万人の基盤の完全リプレイス

Telecom-ISACと運用研究

個人の資格で1年間

通信キャリア8社と運用組織について議論

NTT東西、KDDI、ソフトバンク3社、富士通



学習ガイド

はじめに

学習ガイド

本編

まとめ

ステップアップ

なぜ運用エンジニアリングを学ぶのか？

学ぶ理由

現在の運用で困っていることを根本から解決するため

- ・ 現在の運用から非合理・非論理的な設計・運用を排除し、トラブルや想定外事象を減らしていくため。
- ・ 運用エンジニアの業務から雑用を減らし、エンジニアらしい仕事に集中するため。
- ・ 運用エンジニアとしての将来像を描くため。

この講義で学習すること

講義の要素マップ

運用エンジニアの根本に関わる概念

エンジニアリングとは

運用エンジニアの役割と特徴

「運用」とは

「エンジニア」とは

運用エンジニアのキャリア

エンジニアとしてのステージ

ヒトとしてのステージ

この講座のロードマップ

スタートとゴール

スタート

日々の運用業務に苦勞し、今後のキャリアに悩んでいる



苦勞や悩みを解消するために必要な考え方を理解する



運用に主体的に関与して自分のキャリアを切り開くための
考え方を理解する



ゴール

日々の苦勞を解消し、今後のキャリアを切り開くための**行動に着手**できる

エンジニアリングとは

運用エンジニアの役割と特徴

運用エンジニアのキャリア

この講座のロードマップ

3つのステップ

Step1

エンジニアリングとは

Step2

運用エンジニアの役割と特徴

「運用」とは

「エンジニア」とは

Step3

運用エンジニアのステージ

エンジニアとしてのステージ

ライフステージ

講座の構成

講座の構成は、基本的に以下の5つのステップになっています

疑問

各セクションのテーマに関する「問い」です

あるある

講師が現場で見掛けた話や、経験した話をします

考察

「あるある」に対する**講師の考察**を紹介します

解決へ

「考察」を基に、**みなさんの課題解決へのヒント**を示します

まとめ

各セクションの**まとめ**です

アジェンダ

- Step1. エンジニアリングとは
- Step2. 運用エンジニアの役割と特徴
- Step3. 運用エンジニアのステージ

Step1. エンジニアリングとは

～「エンジニアリング」とは何か本気で考えてみよう～

ナビゲーション

運用エンジニアの根本に関わる概念

エンジニアリングとは

運用エンジニアの役割と特徴

「運用」とは

「エンジニア」とは

運用エンジニアのキャリア

エンジニアとしてのステージ

ヒトとしてのステージ



質問: 「エンジニアリング」 とは？

「エンジニアリング」
と聞いて何を思い描きますか？

こんな「エンジニアリング」に困ったことはありませんか

独り仕事

チーム仕事を嫌がり、独りで仕事をしようとする

神秘主義

超絶技が良いことで、他人が理解できることは価値が低いと考える

斜に構えて説明しない

「わかってくれる人がわかれば良い」と説明責任を軽視する

技術しか興味ない

ビジネス(目的)より技術(手段)を重視する

安定性は後回し

実績がある安定したものよりも、先進性や独自性を優先する

熱しやすく冷めやすい

プロトタイプは熱心に作るが、本番環境の継続性にあまり興味が無い

新しい事にしか興味無い

ビジネスよりも、技術の新しさや知見を賞賛されることを重視する

これらはエンジニアリング
ではなく
悪い意味での
「職人技」

これらはエンジニアリング
ではなく
悪い意味での
「研究」

こんな「エンジニアリング」の弊害

悪い意味での「職人技」の弊害

職人技とは、身につけた熟練した技術によって、手作業で物を作り出す技のこと

独り仕事

チーム仕事を嫌がり、独りで仕事をしようとする

業務がブラックボックス化、既得権益化し、運用現場が硬直化する

神秘主義

超絶技が良いことで、他人が理解できることは価値が低いと考える

保守や引き継ぎができず、**長期的に負債化、ボトルネック化する**

斜に構えて説明しない

「わかってくれる人がわかれば良い」と説明責任を軽視する

周囲からは理解されず、**関与する上司、後輩などの評価も低下させる**

技術しか興味ない

ビジネス(目的)より技術(手段)を重視する

ビジネスから乖離した振る舞いで評価が低下し、厄介者扱いされる

こんな「エンジニアリング」の弊害

悪い意味での「研究」の弊害

研究とは、新しい知識や技術を発見し、その新たな活用方法を見つけるための活動のこと

安定性は後回し

実績がある安定したものよりも、先進性や独自性を優先する
運用業務が不安定になり、保守や引き継ぎも困難になる

熱しやすく冷めやすい

プロトタイプは熱心に作るが、本番環境の継続性にあまり興味が無い
保守・廃止できない環境の乱立や不安定化により現場が混乱に陥る

新しい事にしか興味無い

ビジネスよりも、技術の新しさや知見を賞賛されることを重視する
現実から乖離した振る舞いで評価が低下し、厄介者扱いされる

「エンジニアリング」の歴史

「エンジニアリング」がなぜ誕生したか調べてみました

「エンジニアリング」という言葉の誕生

産業革命開始の約10年前が初出

イギリスで工業が急速に多様化した時代 (蒸気機関の黎明期)

産業革命の特色は「大量生産の実現」

産業革命により、**大量の製品**を**一定の効率**で**安定的・持続的に生産**できるようになった

産業革命の礎となった「エンジニアリング」

適切なQCD(品質、コスト、納期)で

エンジニアリングは

反復再現的な製造方法で

工業製品の大量生産を実現した

持続可能な形で

大切なことは、**エンジニアリングによって「製品」**というビジネス上の価値が生まれること

ITにおける「エンジニアリング」

ITの「エンジニアリング」は何を目指すのか考えてみました

ITにおける「エンジニアリング」は、工業「エンジニアリング」の応用

その本質(大量の生産・処理)に違いは無い 対象領域が物理から論理に変わるだけ

IT(情報技術)の特色は「大量の情報処理の実現」

ITにより、大量の情報を一定の効率で安定的・持続的に処理できるようになった

ITにおける「エンジニアリング」

適切なQCD(品質、コスト、納期)で

ITのエンジニアリングは

反復再現的な処理方法で

情報の大量処理を実現する

持続可能な形で

大切なことは、エンジニアリングによって「情報」というビジネス上の価値が生まれること

ITにおける「エンジニアリング」

ITエンジニアリングが産み出す「情報」

人が作り出す情報 (例)

- ・ 要求定義
- ・ 概要設計書
- ・ アーキテクチャ
- ・ 詳細設計書
- ・ 手順書
- ・ ソースコード

システムが作り出す情報 (例)

- ・ 実行ファイル
- ・ インタフェース
- ・ マスターデータ
- ・ テンポラリデータ
- ・ パケット
- ・ ログ

大切なことは、**エンジニアリングによって「情報」というビジネス上の価値が生まれること**

「エンジニアリング」を正しく理解する

ITにおける「エンジニアリング」とは何か？

適切なQCDで

ビジネス視点でバランスを取る

口よりデータに語らせる

反復再現的な処理方法で

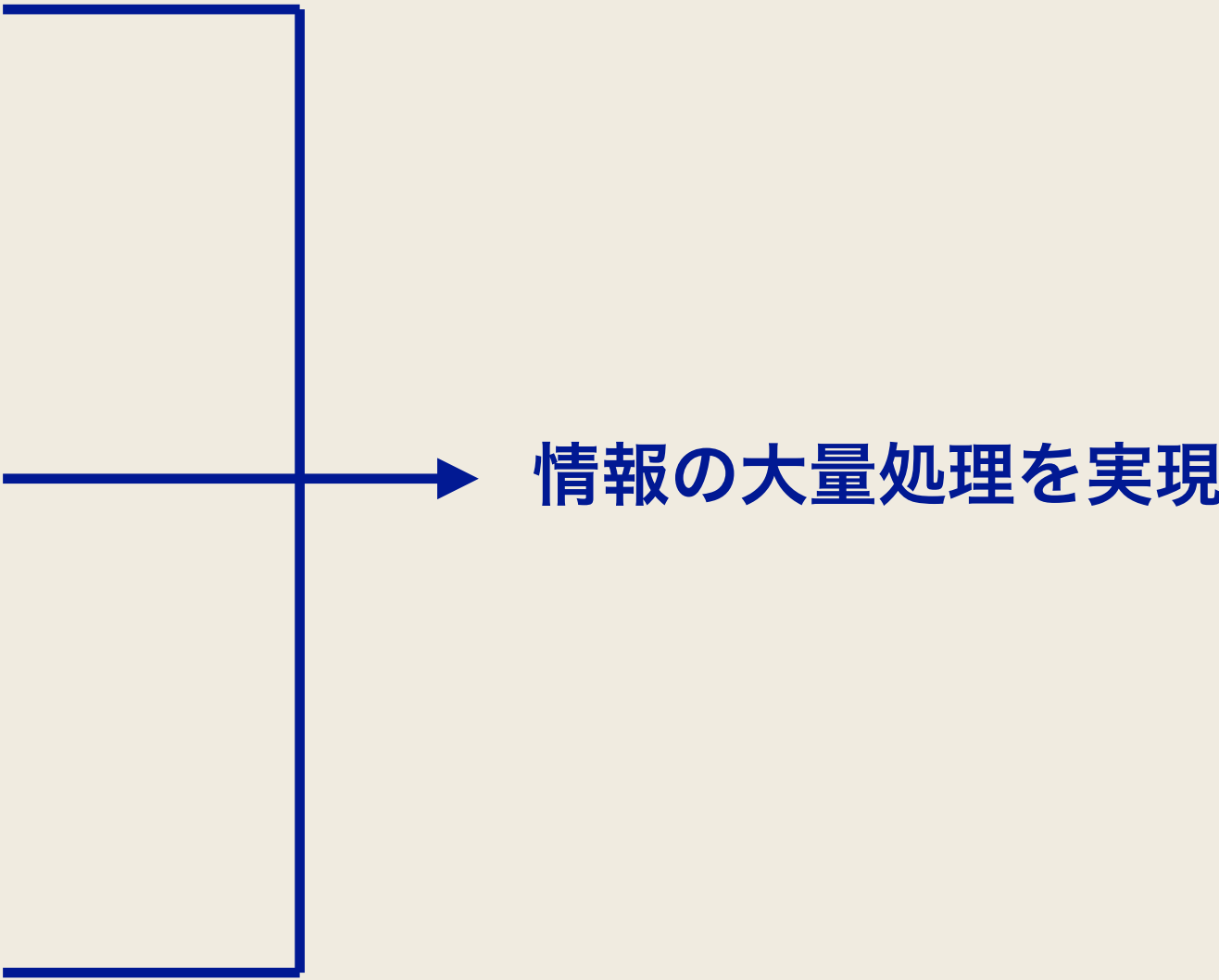
チームで活動する

業務を言語化・客観化する

持続可能な形で

引き継ぎを重視する

陳腐化の回避を重視する



大切なことは、エンジニアリングによって「情報」というビジネス上の価値が生まれること

「エンジニアリング」を正しく理解する

「職人技」との違い

エンジニアリング

職人技

適切なQCDで

ビジネス視点でバランス

Q(品質)に重点

口よりデータ

経験を重視

反復再現的な処理方法で

チームで活動

(主に)個人で活動

業務を言語化・客観化

主観を研ぎ澄ます

持続可能な形で

引き継ぎを重視する

引き継ぎは無頓着

陳腐化の回避を重視する

陳腐化も比較的無頓着

職人技とは、身につけた熟練した技術によって、手作業で物を作り出す技のこと

「エンジニアリング」を正しく理解する

「研究」との違い

適切なQCDで

反復再現的な処理方法で

持続可能な形で

エンジニアリング

ビジネス視点でバランス

口よりデータ

チームで活動

業務を言語化・客観化

引き継ぎを重視する

陳腐化の回避を重視する

研究

Q(品質)に重点

(口よりデータ)

(主にチームで活動)

事実や事象を言語化・客観化

論文を重視

新規性が不可欠

研究とは、新しい知識や技術を発見し、その新たな活用方法を見つけるための活動のこと

「エンジニアリング」を正しく理解する

「研究」との違い

エンジニアリングであるもの (例)

- ・ ビジネスが主、技術が従
- ・ チームで協調した実践活動
- ・ 言語化されて客観的
- ・ 持続性重視(陳腐化回避)
- ・ 持続性重視(引き継ぎ重視)

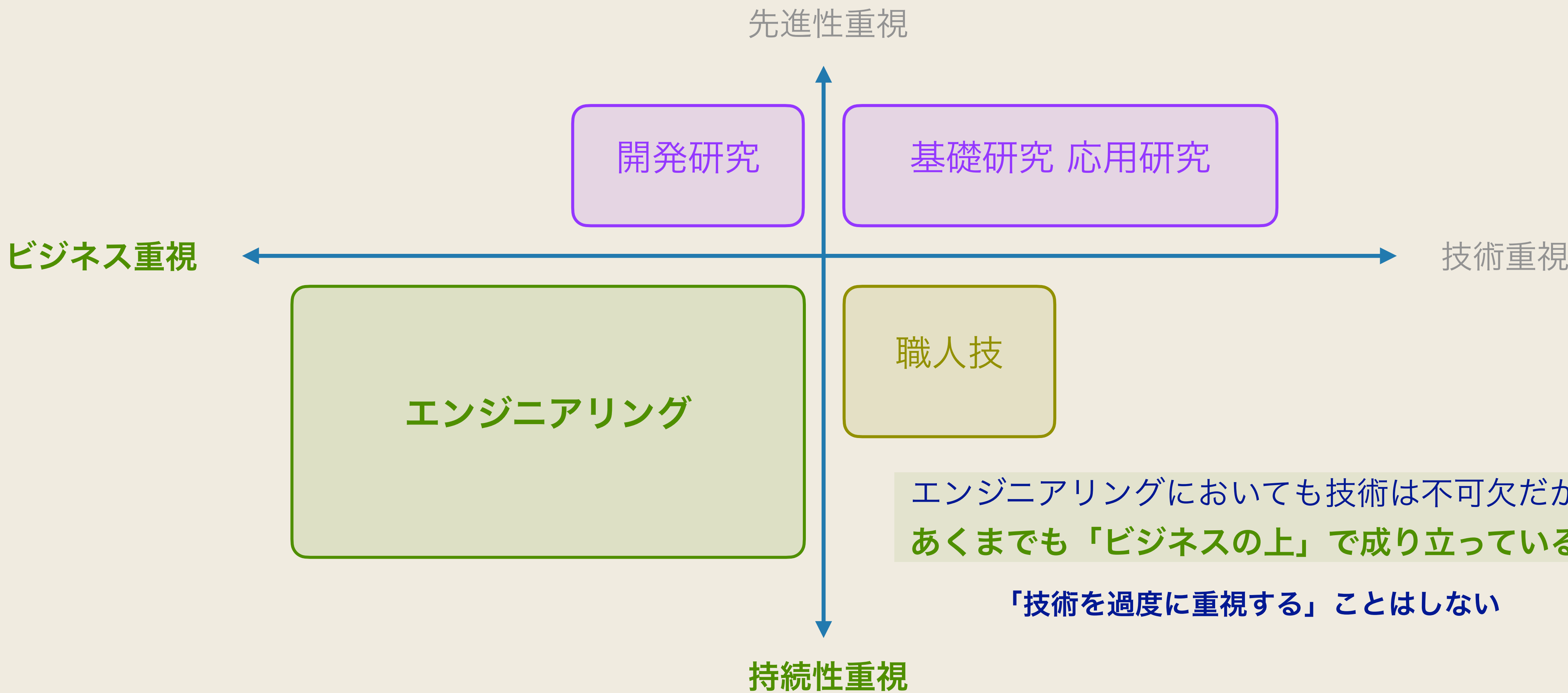
エンジニアリングでないもの (例)

- ・ ビジネスより技術優先
- ・ 組織から乖離した個人活動
- ・ 言語化されず主観的
- ・ 持続性軽視(陳腐化受容)
- ・ 持続性軽視(引き継ぎしない)

大切なことは、**エンジニアリングによって「情報」というビジネス上の価値が生まれること**

「エンジニアリング」の位置付け

エンジニアリングは「ビジネス」と「持続性」を重視する



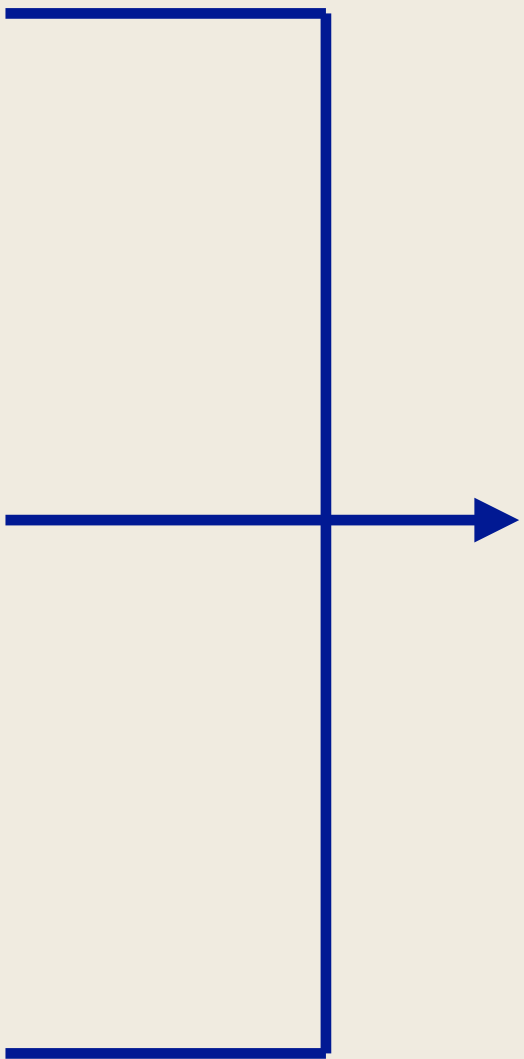
まとめ: 「エンジニアリング」とは

ITにおける「エンジニアリング」とは

適切なQCDで

反復再現的な処理方法で

持続可能な形で



情報の大量処理を実現すること

「職人技」とは混同しないこと

「研究」とも混同しないこと

エンジニアリングは「ビジネス」と「持続性」を重視する

Step2. 運用エンジニアの役割と特徴

ナビゲーション

運用エンジニアの根本に関わる概念

エンジニアリングとは

運用エンジニアの役割と特徴

「運用」とは

「エンジニア」とは

運用エンジニアのキャリア

エンジニアとしてのステージ

ヒトとしてのステージ



Step2. 運用エンジニアの役割と特徴

「運用」とは

ナビゲーション

運用エンジニアの根本に関わる概念

エンジニアリングとは

運用エンジニアの役割と特徴

「運用」とは

1. 企画 vs. 開発 vs. 運用
 2. ビジネスと「運用」
 3. 「運用」の意味
- まとめ

運用エンジニアのキャリア

エンジニアとしてのステージ

ヒトとしてのステージ



「運用」とは

みなさんの現場における
「運用」とは
どのようなことを言うのでしょうか？

キーワードを3つ挙げ、そのキーワードを含めた
「現在の現場における運用とは、～することである」
という一文を作成してみましょう

Step2. 運用エンジニアの役割と特徴

「運用」とは

1. 企画 vs. 開発 vs. 運用

ナビゲーション

運用エンジニアの根本に関わる概念

エンジニアリングとは

運用エンジニアの役割と特徴

「運用」とは

1. 企画 vs. 開発 vs. 運用
 2. ビジネスと「運用」
 3. 「運用」の意味
- まとめ

運用エンジニアのキャリア

エンジニアとしてのステージ

ヒトとしてのステージ



企画 vs. 開発 vs. 運用

よくある「運用でよろしく」

これは「企画業務」
ではないので
「運用業務」でしょう



企画担当



開発担当

これは「開発業務」
ではないので
「運用業務」でしょう

開発終了し
担当がいなし
システム

は?!

この時点で外堀は埋まってる



運用担当

それは「運用業務」じゃない
と思うんだけどなあ...

システムの内容も知らないし
専門性も遠いけどやるはめに

運用技術を磨いてきたはずなのに「なんでも屋」さんになっている...

企画 vs. 開発 vs. 運用

「運用」の特徴を説明できずに苦勞することが多い

企画と開発は、名前から仕事内容を想像しやすい

企画: 需要・要求調査、企画資料作成、稟議、など

開発: プロダクトの設計、進捗管理、コーディング、など

運用は、名前から仕事内容を想像しにくい

「運用はいつも忙しそうだが何をやっているかわからない」と言われてしまう

企画 vs. 開発 vs. 運用

企画や開発と「運用」の違いを考えてみました



企画担当



開発担当

企画からリリースまでが仕事

プロジェクト型業務

成果に区切り(リリース)があるので
特徴が目につきやすい

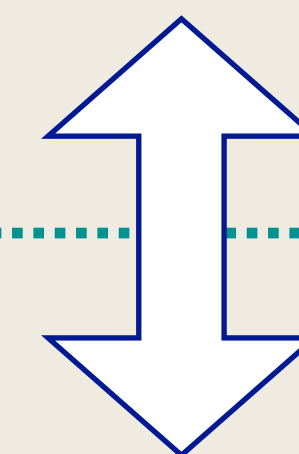
全く同じプロジェクト
全く同じリリース
は存在しない

毎回やり方が違う

毎回結果が違う

非反復的な業務

非再現的な業務



運用担当

リリースからサービス廃止までが仕事

非プロジェクト型業務

成果に区切りが無いので
特徴が目につきにくい

毎回やり方が違う運用
は維持できない

毎回結果が違う運用
は評価されない

反復的な業務

再現的な業務

+

「24時間365日」の活動
が期待されている

企画 vs. 開発 vs. 運用

「運用」の役割は何か考えてみました

運用は「**企業やサービスの事業継続**」を実現する役割を担っている

なぜ「**運用**」には業務の「**反復再現性**」と「**24365**」が求められるのか？



リリースからサービス廃止までが仕事
非プロジェクト型業務

成果に区切りが無いので
特徴が目につきにくい

毎回やり方が違う運用
は**維持できない**

毎回結果が違う運用
は**評価されない**

反復的な業務
再現的な業務

+

「**24時間365日**」の活動
が期待されている

企画 vs. 開発 vs. 運用

運用の「特徴」を理解する

非プロジェクト型業務

毎回やり方が違う運用
は維持できない

毎回結果が違う運用
は評価されない

リリースからサービス廃止
までが仕事

反復的な業務

再現的な業務

+

「24時間365日」の活動
が期待されている

「企業やサービスの事業継続」
を実現する役割

エンジニアリングと
非常に相性が良い

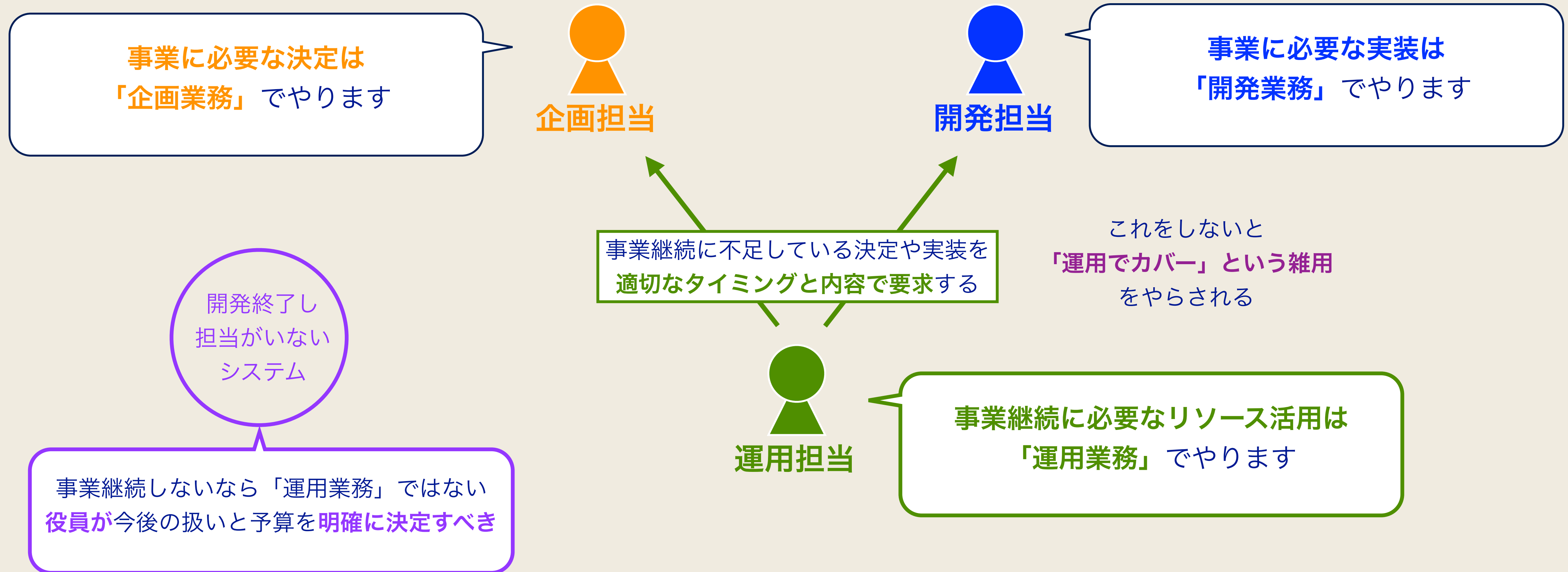
サービスやシステムに
常に最も近い



運用担当

企画 vs. 開発 vs. 運用

正しい分業と協力



企画 vs. 開発 vs. 運用



事業に必要な決定

企画からリリースまでが仕事
プロジェクト型業務

非反復的な業務
非再現的な業務



事業に必要な実装

リリースからサービス廃止までが仕事
非プロジェクト型業務

反復的な業務
再現的な業務



事業継続に必要なリソース活用

「企業やサービスの事業継続」
を実現する役割

サービスやシステムに
常に最も近い

エンジニアリングと
非常に相性が良い

Step2. 運用エンジニアの役割と特徴

「運用」とは

2. ビジネスと「運用」

ナビゲーション

運用エンジニアの根本に関わる概念

エンジニアリングとは

運用エンジニアの役割と特徴

「運用」とは

1. 企画 vs. 開発 vs. 運用
 2. ビジネスと「運用」
 3. 「運用」の意味
- まとめ

運用エンジニアのキャリア

エンジニアとしてのステージ

ヒトとしてのステージ



ビジネスと「運用」

「ビジネスに貢献していない」と言われて困っていませんか？



経営層

運用はコストセンターなので、
予算は少ないほど良い

運用の仕事には、付加価値がない

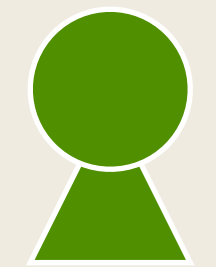


他部署

運用はいつも忙しそうだが、
何をやっているのかわからない

運用に対する批判・圧力

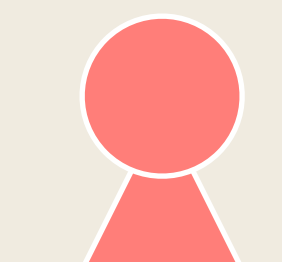
ビジネスに貢献していない
もっと予算と人を減らせ
もっと仕事を増やせ



運用担当

ビジネスと「運用」

「なぜビジネスに運用が必要なのか」を誰も説明しない



経営層

運用はコストセンターなので、
予算は少ないほど良い



他部署

運用の仕事には、付加価値がない

運用はいつも忙しそうだが、
何をやっているのかわからない

そんなに文句言うなら
「運用」いらなくないですか？



運用担当

ビジネスと「運用」

ビジネスに必要な要素を調べてみました

ビジネスモデル

利益を生み出す仕組み。

特に、情報技術やインターネットを利用して、消費者や取引先とのアクセス手段・商品や行為の選択・決済・配送まで一連の経済行為をシステム化し、さらにそれをモデル化したものを指す場合が多い。

(出典: スーパー大辞林)

ビジネスモデルの4要素

ユーザー

Target

誰に

提供価値

Value

どんな価値を

運用能力

Capability

どう実現して

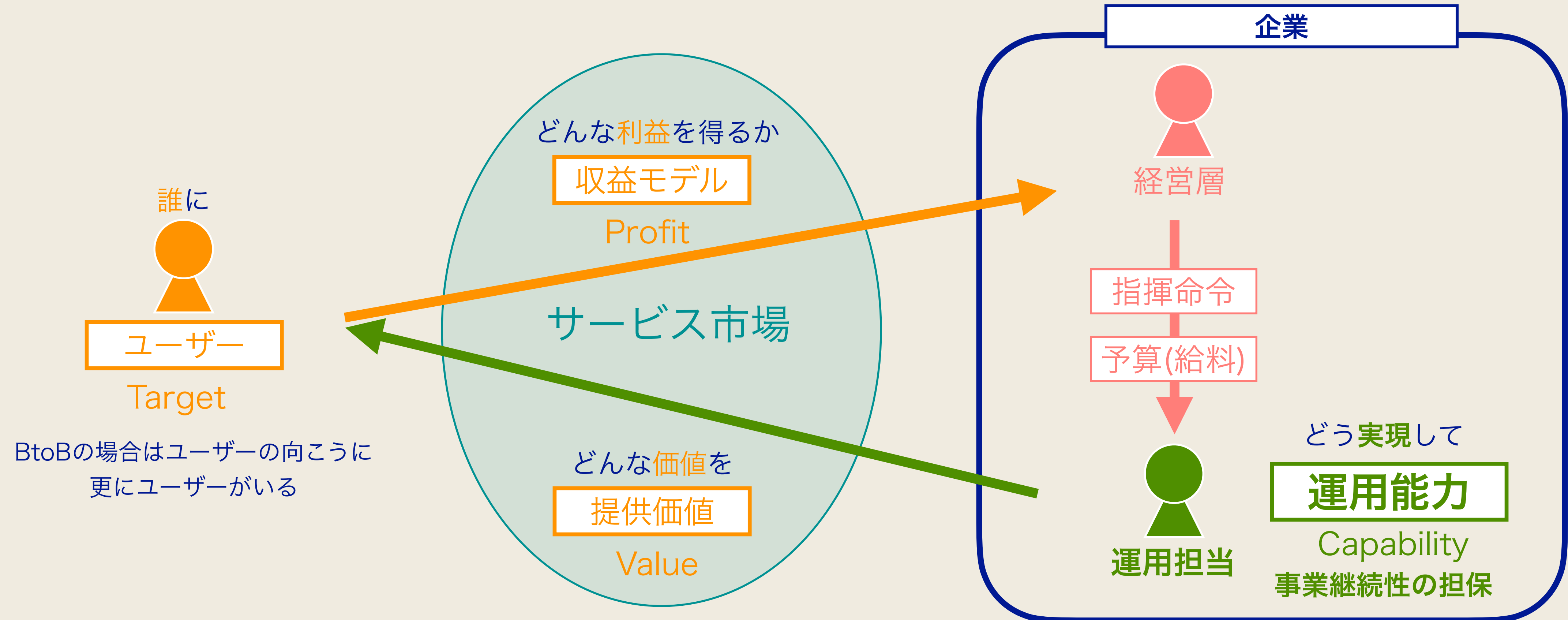
収益モデル

Profit

どんな利益を得るか

ビジネスと「運用」

ビジネスモデルにおける「運用」の役割を考えてみました



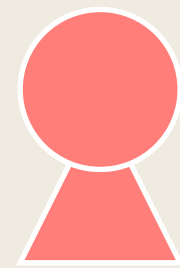
誰に、どんな価値を、どう実現して、どんな利益を得るか

ビジネスと「運用」

ビジネスモデルから考える「運用」の役割

ビジネスモデルの

設計者



経営層



ユーザ

ビジネスモデルの

受益者

予算(給料)

運用能力

Capability



運用担当

ビジネスモデルの

実践者

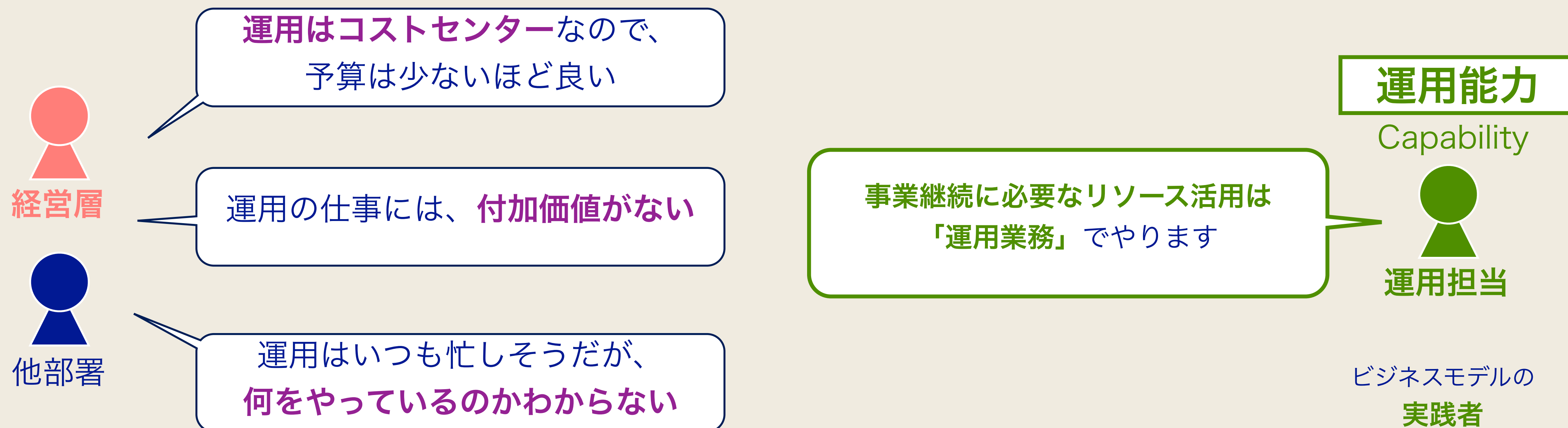
提供価値

Value

事業継続に必要なリソース活用は
「運用業務」でやります

ビジネスと「運用」

「なぜビジネスに運用が必要なのか」を運用側から示す



「予算が無ければ、ビジネスモデルの実践や事業継続は難しいです」と言える運用になることが大事

ビジネスと「運用」

ビジネスモデルの
設計者



経営層



ユーザ

ビジネスモデルの
受益者

予算(給料)

運用能力

Capability



運用担当

ビジネスモデルの
実践者

提供価値

Value

事業継続に必要なリソース活用は
「運用業務」でやります

コストセンターではない

提供価値を産んでいる

忙しいのは重要な仕事だから

Step2. 運用エンジニアの役割と特徴

「運用」とは

3. 「運用」の意味

ナビゲーション

運用エンジニアの根本に関わる概念

エンジニアリングとは

運用エンジニアの役割と特徴

「運用」とは

1. 企画 vs. 開発 vs. 運用
 2. ビジネスと「運用」
 3. 「運用」の意味
- まとめ

運用エンジニアのキャリア

エンジニアとしてのステージ

ヒトとしてのステージ



「運用」の意味

「運用」とは？



お客様
他部署

開発担当がいなくなったので、
運用で面倒見てください

サーバ(インスタンス)の面倒見ているなら
中で動いている**プログラム**の面倒も見てください

サーバー除却の予算が無いから
運用費から回してください

お金払って(費用配賦して)いるから
xxもやってくれますよね
(xxは料金や予算の想定外)

黙っていると
「どんなことでも」
やられる可能性がある



運用担当

運用技術を磨いてきたはずなのに「なんでも屋」さんにされている

「運用」の意味

「運用」という意味や範囲が人によって違うのではないか



お客様
他部署

開発担当がいなくなったので、
運用で面倒見てください

サーバ(インスタンス)の面倒見ているなら
中で動いている**プログラム**の面倒も見てください

サーバー除却の予算が無いから
運用費から回してください

お金払って(費用配賦して)いるから
xxもやってくれますよね
(xxは料金や予算の想定外)



運用担当

それ「運用」じゃない
と思うんですよえ...

言う方は、本当の「運用」の範囲を認識できていない可能性がある

「運用」の意味

「運用」の意味を調べてみました

- ・ うまく機能を働かせ用いること、活用。
- ・ そのもののもつ機能を生かして用いること。活用。

(出典: スーパー大辞林)

「何かを活用」しない活動は「運用」とは言えない。

「運用」の意味

日々の運用では「何を活用」しているのか、考えてみました

- ・ うまく機能を働かせ用いること、活用。
- ・ そのもののもつ機能を生かして用いること。活用。

(出典: スーパー大辞林)

活用の対象

運用現場の**人材**、**物理/情報資産**、**予算**を活用しています

ヒト

人的リソース

(正社員、派遣社員、アルバイト)

モノ

物的リソース

(拠点、ネットワーク、サーバなど)

情報リソース

(システム情報、運用情報、顧客情報)

カネ

予算リソース

(現金、信用)

運用業務の遂行には、実に多大な**リソース**が必要です。

「運用」の意味

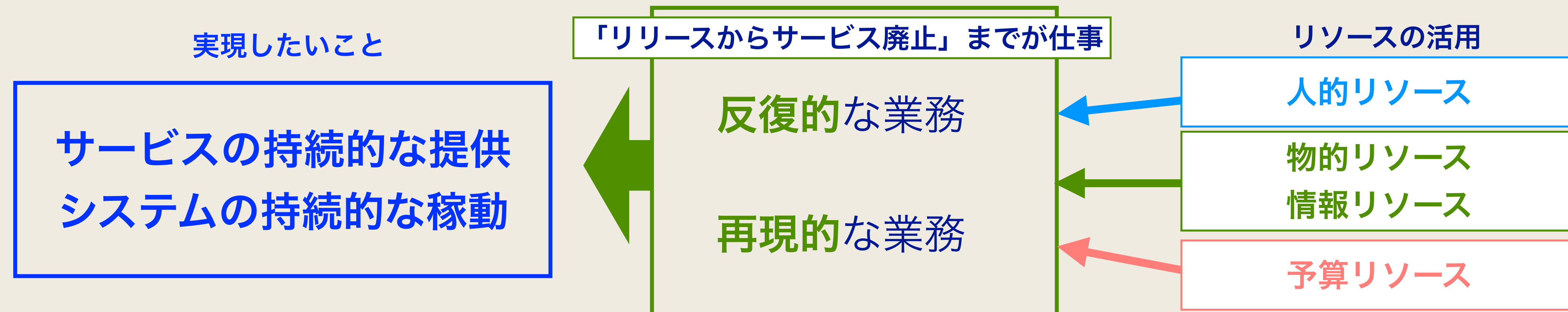
運用は「リソースを活用」して何を実現しようとしているのか、考えてみました

- ・ うまく機能を働かせ用いること、活用。
- ・ そのもののもつ機能を生かして用いること。活用。

(出典: スーパー大辞林)

活用の方向

サービスを提供し続けシステムを稼動し続けようと努力しています



「運用」の意味

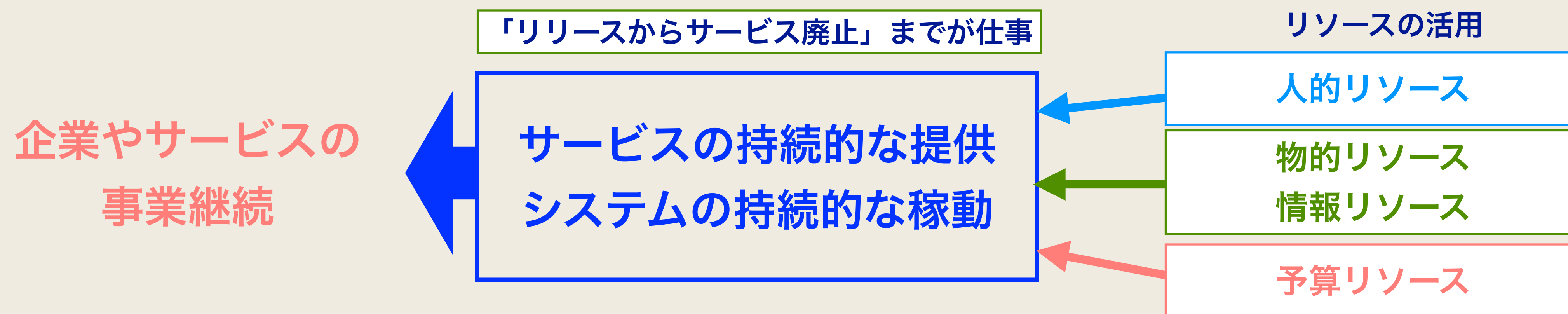
運用は「リソースの活用」を何のためにしているのか、考えてみました

- ・ うまく機能を働かせ用いること、活用。
- ・ そのもののもつ機能を生かして用いること。活用。

(出典: スーパー大辞林)

活用の目的

「**企業やサービスの事業継続**」を実現するためにリソースの活用をしています



「運用」の意味

日々の「運用」の意味を整理してみましょう

何のために **企業やサービスの事業継続を実現**するために

何を **運用組織のリソースを活用して**

どうする **サービスの提供・システムの稼動をし続けている**

「運用」の意味

自分視点での「運用」を定義してみましょう (例)

何のために **対価や評価を得る**ために

何を活用して **運用組織のリソースを**活用して

どうすること **外部に対して継続的に何らかのサービスを提供し続ける**こと

「運用」の意味

運用の定義 (例)

運用組織のリソースを活用し、対価や評価を得ることを目的に、外部に対して、継続的に何らかのサービスを提供し続けること。

連載：現場視点からの運用方法論

第3回 明日の運用現場のために - 運用フレームワークという視点

<https://thinkit.co.jp/story/2010/12/16/1934?page=0%2C2>

運用業務は「**企業やサービスの事業継続**」を実現するために活動している

運用業務の継続

が

ビジネスの継続

「何かを活用する」活動が「運用」である

「言われたことをその通りにやる」のが運用なわけではない

参考: 日本語の「運用」と英語の"operation"

「運用」="operation"ではない

「運用」

- ・ そのもののもつ機能を生かして用いること。活用。

(出典: スーパー大辞林)

"operation"

- ・ [MEDICAL] the process of cutting into someone's body to **repair** or **remove a part that is damage**
- ・ [WORK/ACTIVITIES] the **work or activities** done by a business or organization, or the process of doing this work

壊れたものを治す

単なる作業

(出典: Longman Dictionary of Contemporary English)

「オペレーション」は「運用」の一部だが「オペレーションが運用の全て」ではない

翻訳による「運用」="operation"を
現場でも等価と決め付けて実践すると、、、

誤って失う価値があるかもしれない

Step2. 運用エンジニアの役割と特徴

「運用」とは

まとめ

ナビゲーション

運用エンジニアの根本に関わる概念

エンジニアリングとは

運用エンジニアの役割と特徴

「運用」とは

1. 企画 vs. 開発 vs. 運用
2. ビジネスと「運用」
3. 「運用」の意味

まとめ

運用エンジニアのキャリア

エンジニアとしてのステージ

ヒトとしてのステージ



まとめ: 「運用」とは

運用組織のリソースを活用し、対価や評価を得ることを目的に、外部に対して、継続的に何らかのサービスを提供し続けること。



運用担当

事業継続に必要なリソース活用は
「運用業務」で行います

「企業やサービスの事業継続」
を実現する役割

運用能力

Capability

リリースからサービス廃止までが仕事

非プロジェクト型業務

サービスやシステムに
常に最も近い

ビジネスモデルの
実践者

反復的な業務
再現的な業務

エンジニアリングと
非常に相性が良い

Step2. 運用エンジニアの役割と特徴

「エンジニア」とは

ナビゲーション

運用エンジニアの根本に関わる概念

エンジニアリングとは

運用エンジニアの役割と特徴

「エンジニア」とは

1. 「エンジニアリングする人」がエンジニア
 2. 「エンジニア」というビジネス
 3. エンジニアと上司-部下、先輩-後輩の関係
- まとめ

運用エンジニアのキャリア

エンジニアとしてのステージ

ヒトとしてのステージ



「エンジニア」とは

「エンジニア」とは
どのような人達のことを言うのでしょうか？

キーワードを1つ挙げ、そのキーワードを含めた
「エンジニアとは、～する人である」
という一文を作成してみましょう

Step2. 運用エンジニアの役割と特徴

「エンジニア」とは

1. 「エンジニアリングする人」がエンジニア

ナビゲーション

運用エンジニアの根本に関わる概念

エンジニアリングとは

運用エンジニアの役割と特徴

「エンジニア」とは

1. 「エンジニアリングする人」がエンジニア
 2. 「エンジニア」というビジネス
 3. エンジニアと上司-部下、先輩-後輩の関係
- まとめ

運用エンジニアのキャリア

エンジニアとしてのステージ

ヒトとしてのステージ



「エンジニアリングする人」がエンジニア

「エンジニア」と名乗りながら、こんな人はいませんか？

主観的な物言いをする人

客観的なデータや根拠を示さずに「自分はこう思う」としか言わない・言えない

合理的な説明ができない人

説明に合理性や一貫性が無く、自分の言う事の根拠の弱さや矛盾に気付けない

引き継ぎができない人

「現物を見ればわかる」と言って、きちんとした引き継ぎをしない・できない

ものごとを客観化できない人

自分の立場や視点に強く捉われ、一步離れて物事を捉えることができない

再現性の無い仕事をする人

仕事の品質や成果のバラツキが大きく、過去の成果を再現することができない

持続性のある仕事ができない人

短期的な仕事は上手だが、長期的な視野で課題に対応することはできない

「エンジニアリングする人」がエンジニア

「エンジニアリングできない人」がエンジニアの仕事をする弊害

主観的な物言いをする人

主観的な人同士の衝突、客観的な人との意思疎通困難により現場が疲弊する

合理的な説明ができない人

「雰囲気」で物事を進めるため、様々なトラブルや課題を頻発させる

引き継ぎができない人

仕様や意図が不明のツールやプロセスが山積み、現場が混乱・硬直化する

ものごとを客観化できない人

現場の現実や実情と乖離した設計や実装を行い、現場を混乱・硬直化させる

再現性の無い仕事をする人

重要な仕事を任せることができないため、フォローやレビューの負担が大きい

持続性のある仕事ができない人

短期的には良い仕事をしているように見えるが、長期的な負債を産みやすい

「エンジニアリングする人」がエンジニア

「エンジニアではないふるまい」を整理してみました

主観的な物言いをする人

合理的な説明ができない人

引き継ぎができない人

ものごとを客観化できない人

再現性の無い仕事をする人

持続性のある仕事ができない人

主観的: 意見や分析を主観で語る

非合理的: 感覚や雰囲気で物事を語る

非論理的: 論理的な説明ができない

非持続的: やりっぱなし

「エンジニアリングする人」がエンジニア

「エンジニアではないふるまい」が運用にもたらすものを考えてみました

主観的、非合理的、非論理的、非持続的な活動や業務により
日々**運用課題が山積**していくことになる

1. 人が理解しやすい運用ではなくなる
2. システムが扱いやすい運用ではなくなる
3. 論理破綻・矛盾による無駄・無意味が多い運用になる

- ・ ドキュメント化工数が膨大になる
- ・ 中途・新人の戦力化に時間がかかる
- ・ 環境変化への対応に遅れがちになる
- ・ Whyが失われると硬直化に繋がる
- ・ ツール製品を導入しても効果が出にくい (連携しにくい)

主観的: 意見や分析を主観で語る

非合理的: 感覚や雰囲気で物事を語る

非論理的: 論理的な説明ができない

非持続的: やりっぱなし

運用課題を(エンジニアリングによってではなく)**根性や気合で「解決」していく**ことになる

「エンジニアリングする人」がエンジニア

「エンジニア」に期待されるふるまいを抽出してみました

客観的: 意見や分析を客観で語る

合理的: 物事を原理原則や事実で語る

論理的: 論理的な説明ができる

持続的: 持続的な設計や実装ができる

主観的: 意見や分析を主観で語る

非合理的: 感覚や雰囲気物事を語る

非論理的: 論理的な説明ができない

非持続的: やりっぱなし

「エンジニアリングする人」がエンジニア

「エンジニア」に期待されるエンジニアリング的行動

ITにおける「エンジニアリング」

客観的: 意見や分析を客観で語る

合理的: 物事を原理原則や事実で語る

論理的: 論理的な説明ができる

持続的: 持続的な設計や実装ができる

適切なQCD
で計測・判断

**反復再現的な
処理方法**
を設計・実装

持続可能な形
で実現・継続

情報の大量処理
を実現

「エンジニアリングする人」がエンジニア

「エンジニア」に求められるエンジニアリング的な「美学」

主観的な言葉で語るのではなく

ロジック・モデル・データに語らせる

客観的

実装に重きを置くのではなく

理論やモデル(アーキテクチャ)に重きを置く

合理的

論理的

近視眼的な視点で見るとはではなく

長期的・俯瞰的な視点で物事を捉える

持続的

(参考) 不確定未来の予測を諦めるのではなく

サイクル(流行サイクル/ライフサイクルなど)観点で変化を予測して対応する

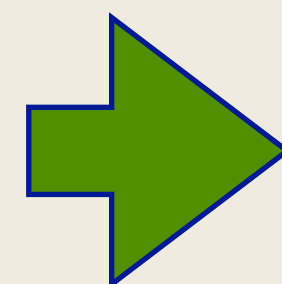
「エンジニアリングする人」がエンジニア

「エンジニアらしいふるまい」が運用にもたらすもの

人材面(育成や引き継ぎ)、システム面(自動化)で大きな変化が起きる

「エンジニアではないふるまい」による運用

1. 人が理解しやすい運用ではなくなる
2. システムが扱いやすい運用ではなくなる
3. 論理破綻・矛盾による無駄・無意味が多い運用になる



「エンジニアらしいふるまい」による運用

1. 人が理解しやすい運用になる
2. システムが扱いやすい運用になる
3. 論理的に正しいことが検証された運用になる

人材がスケールする

業務がスケールする

工数を最大活用できる

- ・ ドキュメント化工数が膨大になる
- ・ 中途・新人の戦力化に時間がかかる
- ・ 環境変化への対応に遅れがちになる
- ・ Whyが失われると硬直化に繋がる
- ・ ツール製品を導入しても効果が出にくい (連携しにくい)

- ・ ドキュメント化工数を必要最小限にできる
- ・ 中途・新人の戦力化に時間がかからない
- ・ 環境変化への対応が比較的容易にできる
- ・ Whyが失われにくいので硬直化を避けやすい
- ・ ツール製品を導入したときに効果が出やすい (連携しやすい)

運用課題を根性や気合で「解決」する必要がなくなる

「エンジニアリングする人」がエンジニア

「エンジニアらしいふるまい」

客観的: 意見や分析を客観で語る

合理的: 物事を原理原則や事実で語る

論理的: 論理的な説明ができる

持続的: 持続的な設計や実装ができる

「エンジニアらしいふるまい」による運用

1. 人が理解しやすい運用になる
2. システムが扱いやすい運用になる
3. 論理的に正しいことが検証された運用になる

運用課題を根性や気合で「解決」する必要がなくなる

Step2. 運用エンジニアの役割と特徴

「エンジニア」とは

2. 「エンジニア」というビジネス

ナビゲーション

運用エンジニアの根本に関わる概念

エンジニアリングとは

運用エンジニアの役割と特徴

「エンジニア」とは

1. 「エンジニアリングする人」がエンジニア
 2. 「エンジニア」というビジネス
 3. エンジニアと上司-部下、先輩-後輩の関係
- まとめ

運用エンジニアのキャリア

エンジニアとしてのステージ

ヒトとしてのステージ



「エンジニア」というビジネス

「ビジネスに興味の無いエンジニア」という存在

ビジネスを気にしない、興味のない自称エンジニア

「エンジニアなので、**ビジネスの成功・失敗は自分には関係ない**」と考えている

「手段のためには目的を選ばず」と公言する人

「自分の興味のある技術(手段)を試せるなら、**ビジネスがどうなろうと知らない**」と公言し、実践している

「エンジニア」というビジネス

給料がどこから出ているか考えていない (極端な実例)

ビジネスを気にしない、興味の無い自称エンジニア

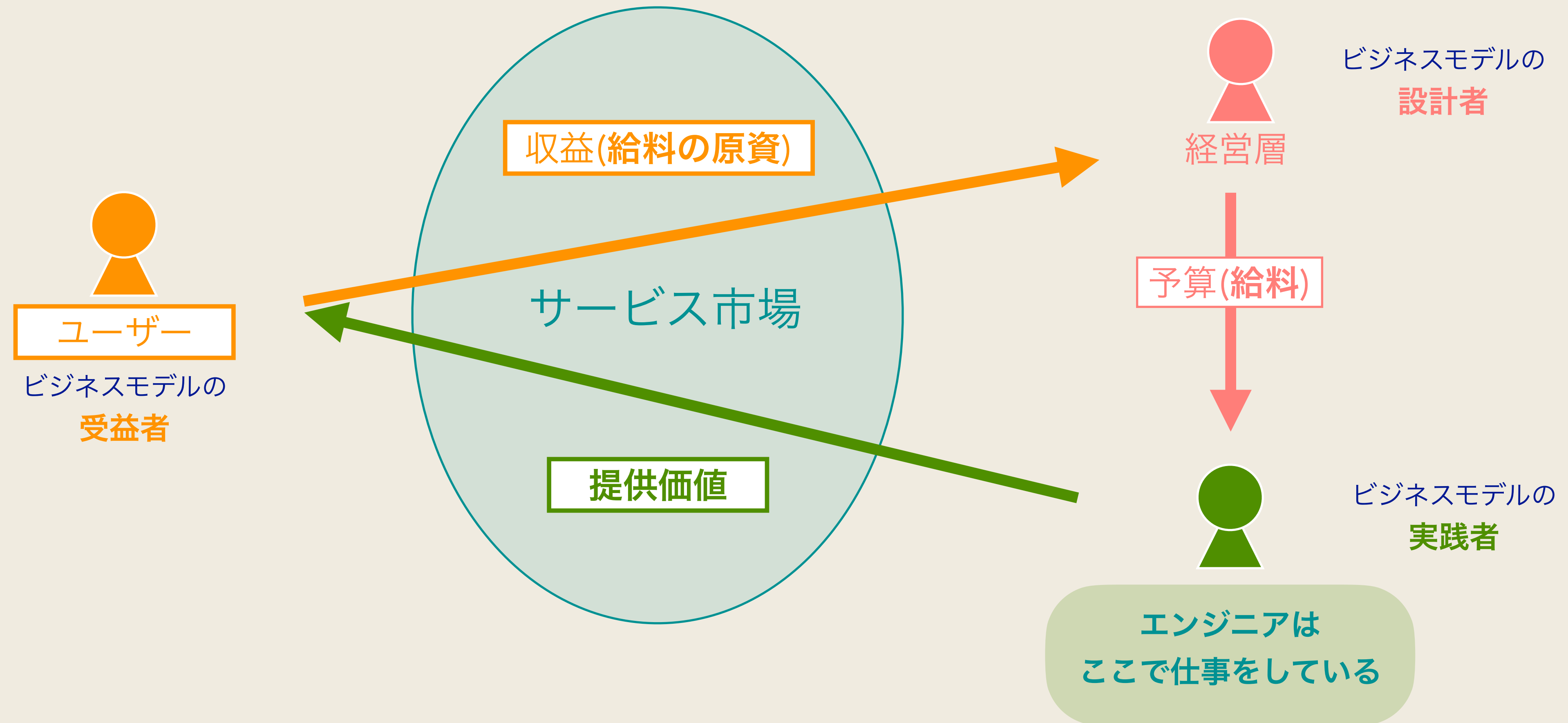
ビジネスの成功か失敗に関わらず給料はもらえるし、がんばっていれば当然昇給するものだと思っている

「手段のためには目的を選ばず」と公言する人

自分の技術なら当然に良い給料はもらえるはずで、それをビジネスに繋げるのは自分の仕事じゃないと思っている

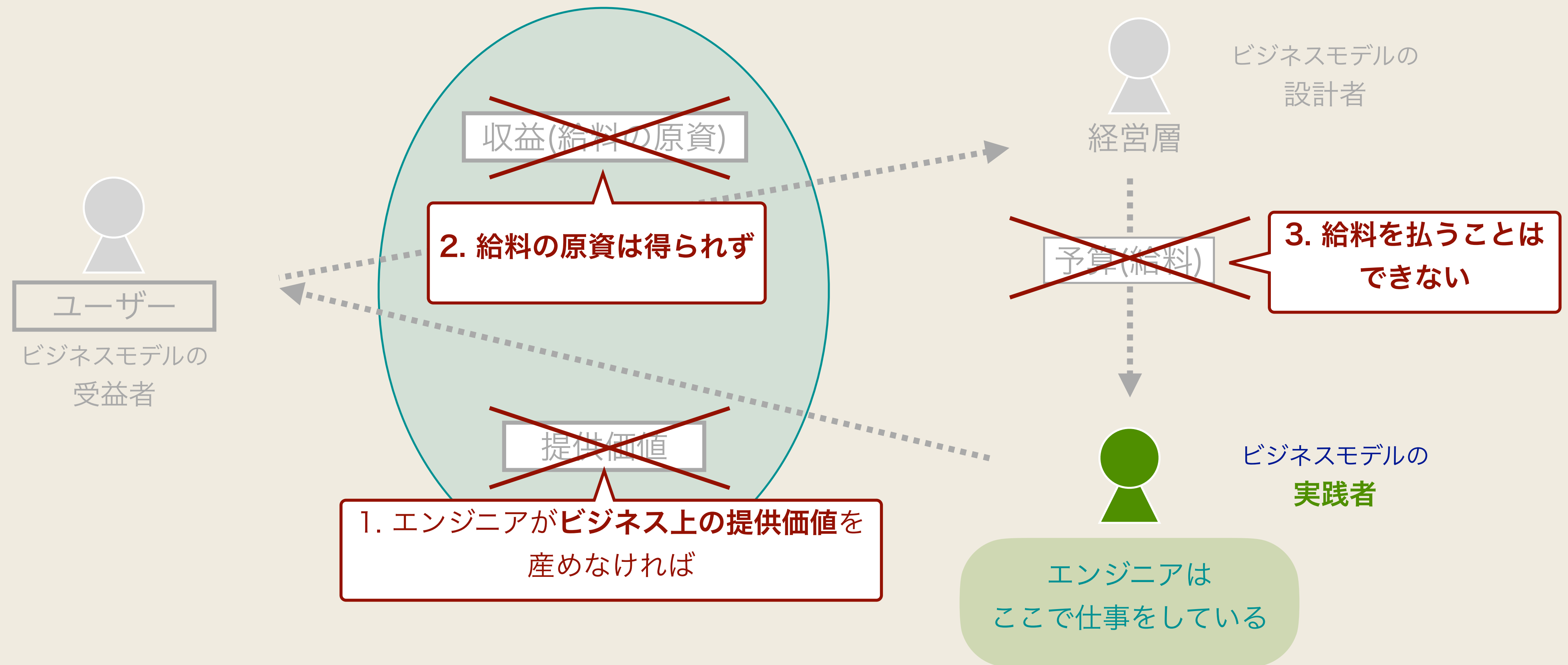
「エンジニア」というビジネス

ビジネスとエンジニアの関係について整理してみました



「エンジニア」というビジネス

エンジニアの給料はどこから生まれるか考えてみました (実体験)



「エンジニア」というビジネス

「エンジニアはビジネスパーソン」という意識が必要

エンジニアは、**エンジニアリングをビジネス**にして報酬を得ている**プロフェッショナル**



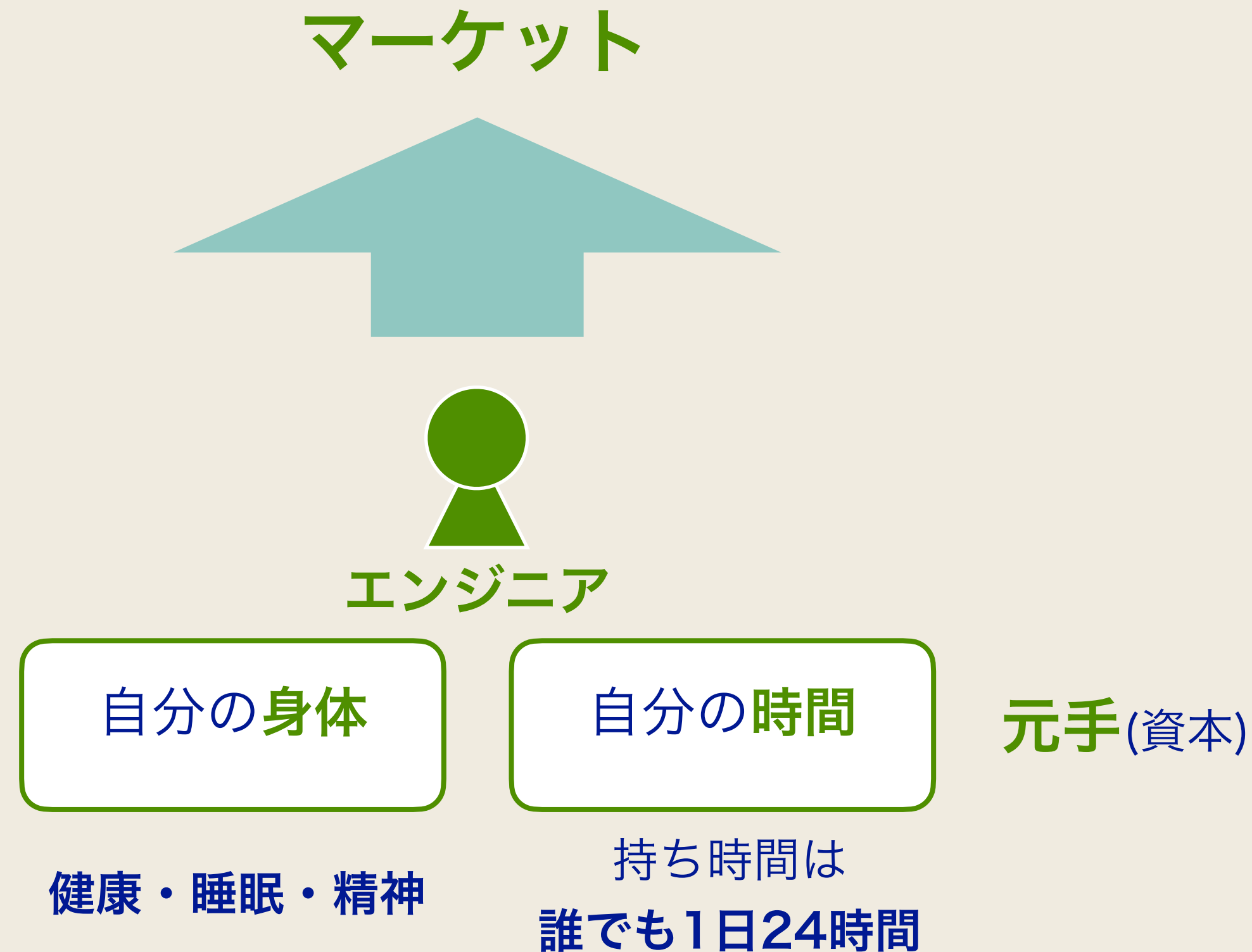
**ビジネス観点での技術選定やモデル(アーキテクチャ)を
作れない人は居場所が無くなっている**

特にクラウド界限ではビジネス観点で技術選定ができない人は姿を消しつつある

「エンジニア」というビジネス

エンジニアの元手は「身体」と「時間」

エンジニアは、自分の身体と時間を「マーケット」に投じて収益を得ている



「エンジニア」というビジネス

エンジニアと「技術マーケット」(調達)

エンジニアは、**仕事や興味に応じて「技術マーケット」から技術を調達する**

技術調達時の考慮ポイント(例)

- ・ **習得の必要性** (すぐに必要か、なぜ必要か)
- ・ **市場性** (情報があるか、商用事例があるか)
- ・ **信頼性** (実績があるか、スジが良いか)
- ・ **将来性** (陳腐化が早くないか、無駄にならないか)
- ・ **習得に掛かる時間・費用**
- ・ 個人的な技術的嗜好、優先順位など

技術マーケット

書籍、動画、blog
eラーニング、講座など

技術の調達

カネ & 時間を費す

エンジニア

自分の**身体**

自分の**時間**

元手(資本)

調達する技術によって、**将来どのような仕事ができるか**が決まってくる

「エンジニア」というビジネス

エンジニアと「ビジネスマーケット」(参入)

エンジニアは、得意領域や需要に応じて「ビジネスマーケット」に参入している

ビジネスマーケット

報酬の受領

カネを得る

労務の提供

時間を費す

エンジニア

自分の身体

自分の時間

元手(資本)

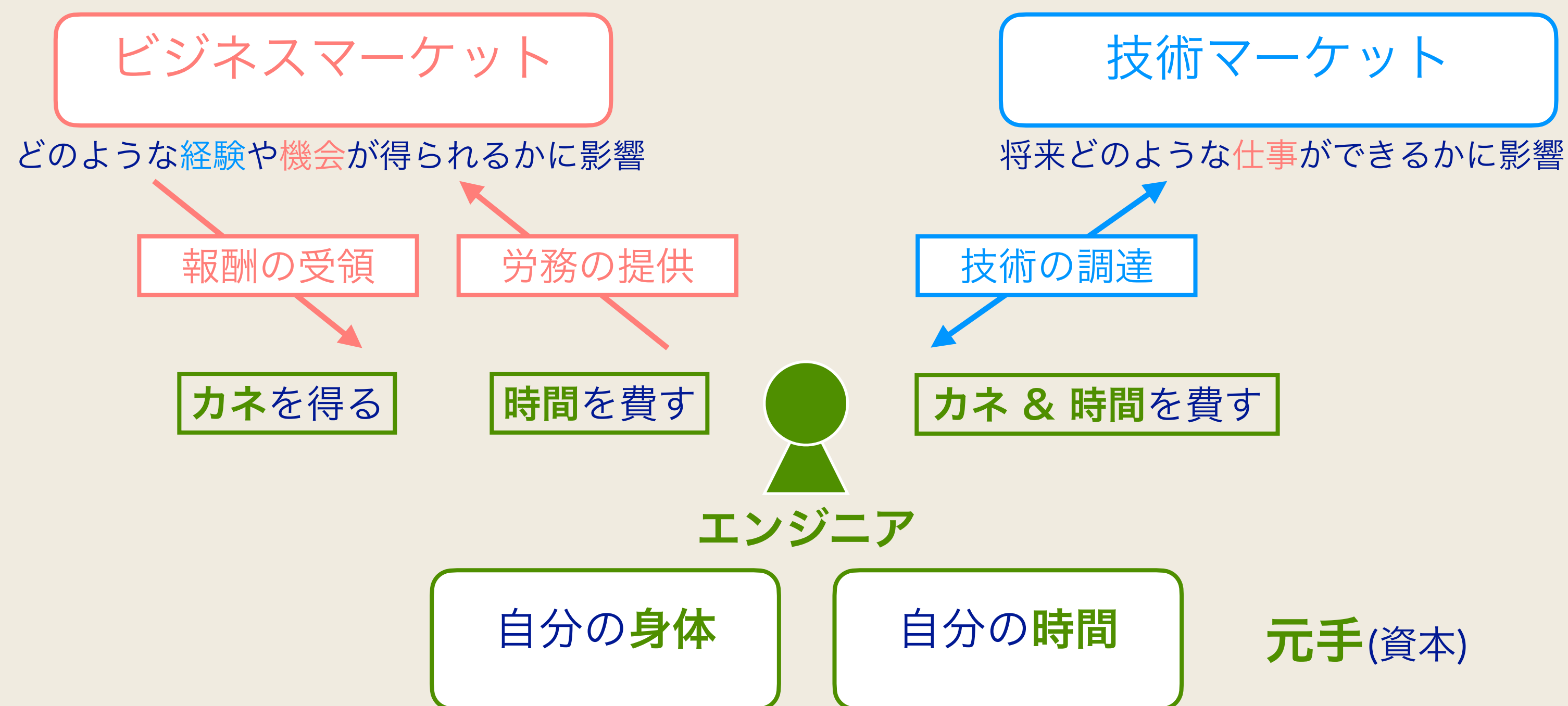
参入時の考慮ポイント(例)

- ・ **需要** (仕事があるか、引き合いは多いか)
- ・ **信頼性** (荒れていないか、客筋が良いか)
- ・ **将来性** (陳腐化が早くないか、縮小していかないか)
- ・ **流動性** (変化は早いか、参入障壁はあるか)
- ・ **参入の機会、参入に掛かる時間**
- ・ 個人的な仕事の嗜好、優先順位など

参入するマーケットによって、どのような経験や機会が得られるかが決まってくる

「エンジニア」というビジネス

「エンジニアリング」をビジネスにして報酬を得ているプロフェッショナル



参入しているマーケットや調達すべき技術を意識することが、プロフェッショナルなエンジニアリングに繋がる

Step2. 運用エンジニアの役割と特徴

「エンジニア」とは

3. エンジニアと上司-部下、先輩-後輩の関係

ナビゲーション

運用エンジニアの根本に関わる概念

エンジニアリングとは

運用エンジニアの役割と特徴

「エンジニア」とは

1. 「エンジニアリングする人」がエンジニア
 2. 「エンジニア」というビジネス
 3. エンジニアと上司-部下、先輩-後輩の関係
- まとめ

運用エンジニアのキャリア

エンジニアとしてのステージ

ヒトとしてのステージ



エンジニアと上司-部下、先輩-後輩の関係

「上司や後輩に振り回されずに仕事をしたい」という人

チームの中であっても独りで仕事をしたい人

「自分は技術に自信があるので、**できるだけ人と関わらずに仕事をしたい**」と考えている

上司やリーダーの「リーダシップ」に対して冷淡な人

「上司やリーダーが何やら言っているけど、**自分の仕事には口を出さないで欲しい**」と考えている

後輩や部下の育成に対して興味の無い人

「後輩や部下の育成は**面倒なので、関わりたくない**」と考えている

エンジニアと上司-部下、先輩-後輩の関係

本当の意味での「求められている成果」を理解していない (極端な実例)

チームの中であっても独りで仕事をしたい人

本人がやりたい「成果」とチームが求める「成果」が異なることで**チームが機能不全となり、チームも本人も低い評価になってしまう**

上司やリーダーの「リーダシップ」に対して冷淡な人

リーダシップが機能しないため「チームの評価」が低下してしまい、**一緒に働く人全員の評価を下げてしまう**

後輩や部下の育成に対して興味の無い人

いつまで経っても「チームとしての仕事の質とスピード」が向上しないため、**チームと個人の評価が上がらない**

エンジニアと上司-部下、先輩-後輩の関係

エンジニアとチームの成果の関係を考えてみました

ビジネスモデルの
設計者



経営層



ユーザ

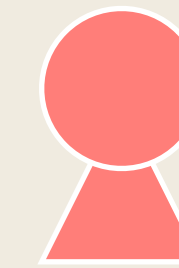
ビジネスモデルの
受益者

要求

成果

チームのパフォーマンスを最大に発揮すれば
大きな成果を産むことができる

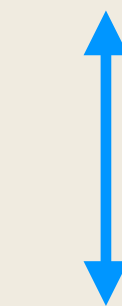
上司/リーダー



部下

エンジニア

先輩

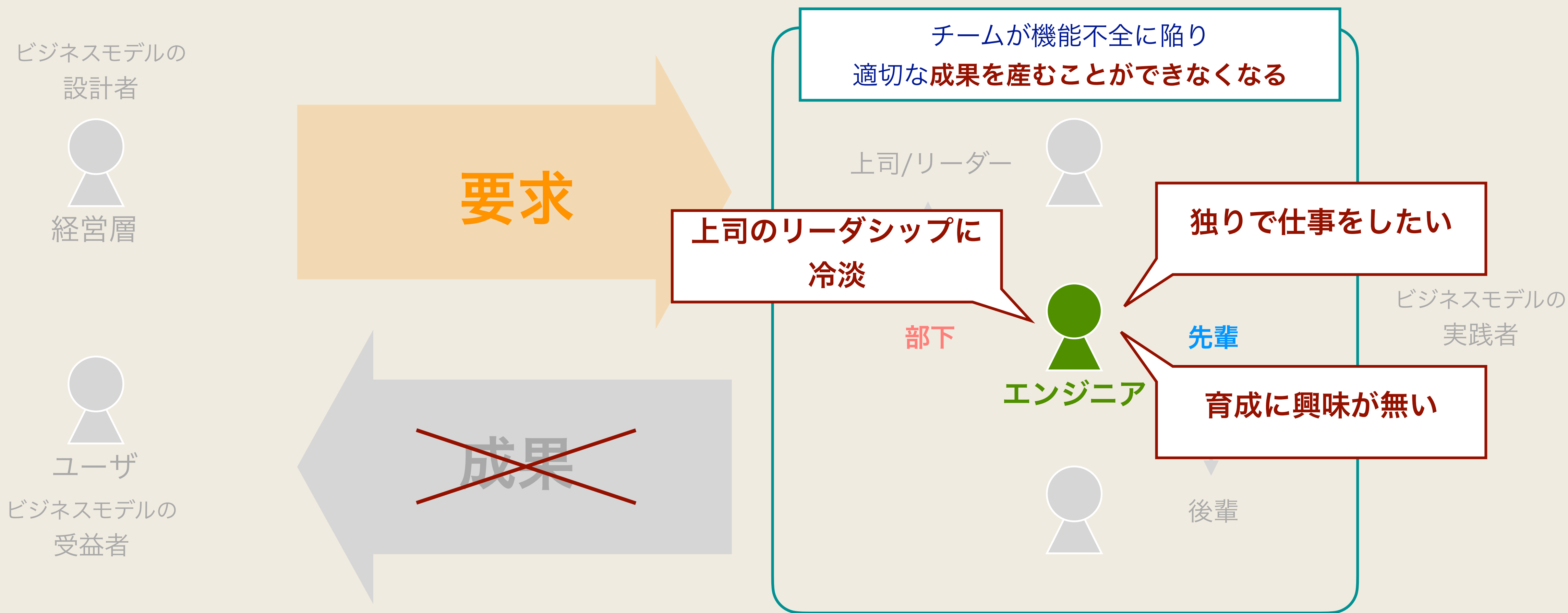


後輩

ビジネスモデルの
実践者

エンジニアと上司-部下、先輩-後輩の関係

エンジニアが「求められている成果」を理解していないと...

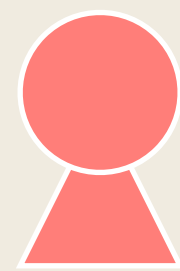


エンジニアと上司-部下、先輩-後輩の関係

エンジニアにとって理想の「上司との関係」を整理してみました

ビジネスモデルの

設計者



経営層



ユーザ

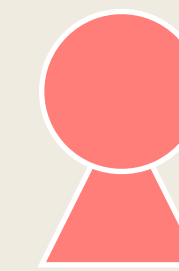
ビジネスモデルの

受益者

要求

成果

上司/リーダー



上司を上手く使う

司令塔

リード

フォロー

実働部隊

上司に上手く使われる



エンジニア

ビジネスモデルの

実践者

役割としての「司令塔」と「実働部隊」
を分担し

チームのパフォーマンスを最大化する

アンチパターン

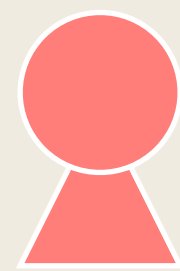
上司が理解も協力もしてくれなくなる

エンジニアと上司-部下、先輩-後輩の関係

エンジニアにとって理想の「後輩との関係」を整理してみました

ビジネスモデルの

設計者



経営層



ユーザ

ビジネスモデルの

受益者

要求

成果

拡大

エンジニアが後輩を育成し
戦力化することで
チームのパフォーマンスが強化される

エンジニア

後輩に(教科書として)
上手に使われる

育成

戦力化

後輩を上手く
戦力化する

後輩

副産物

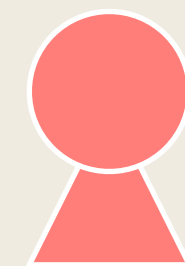
「人に教えること」は
教える側にとって深い学習機会になる

ビジネスモデルの
実践者

エンジニアと上司-部下、先輩-後輩の関係

「フォロワーシップ」という考え方

自発的に、リーダーや他のメンバーに働きかける行動を生み出すマインド



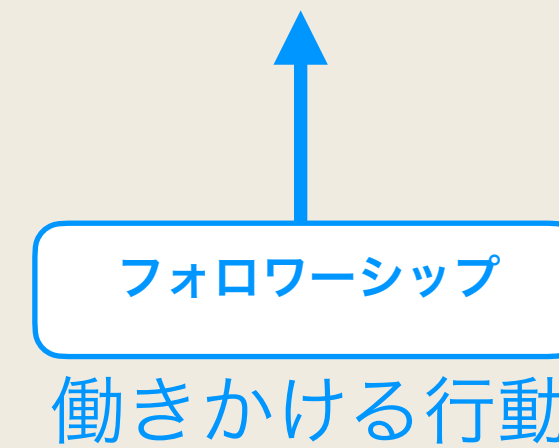
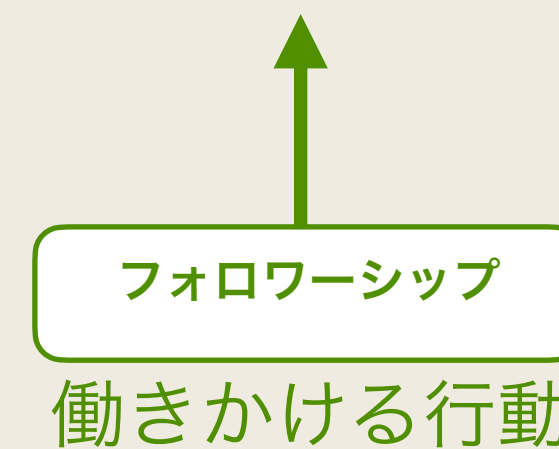
上司/リーダー



エンジニア



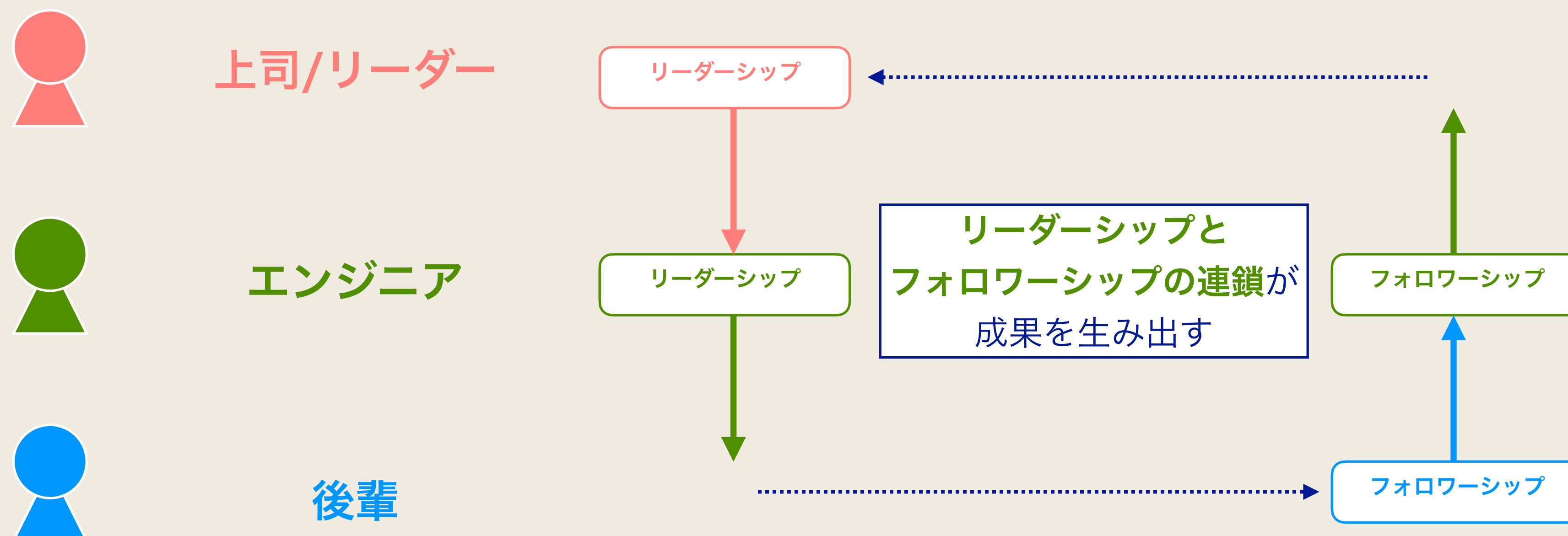
後輩



エンジニアと上司-部下、先輩-後輩の関係

「リーダーシップ」と「フォロワーシップ」

フォロワーシップを前提としたリーダーシップと自発的なフォロワーシップが
相互作用することで連鎖的に成果が生まれやすくなる

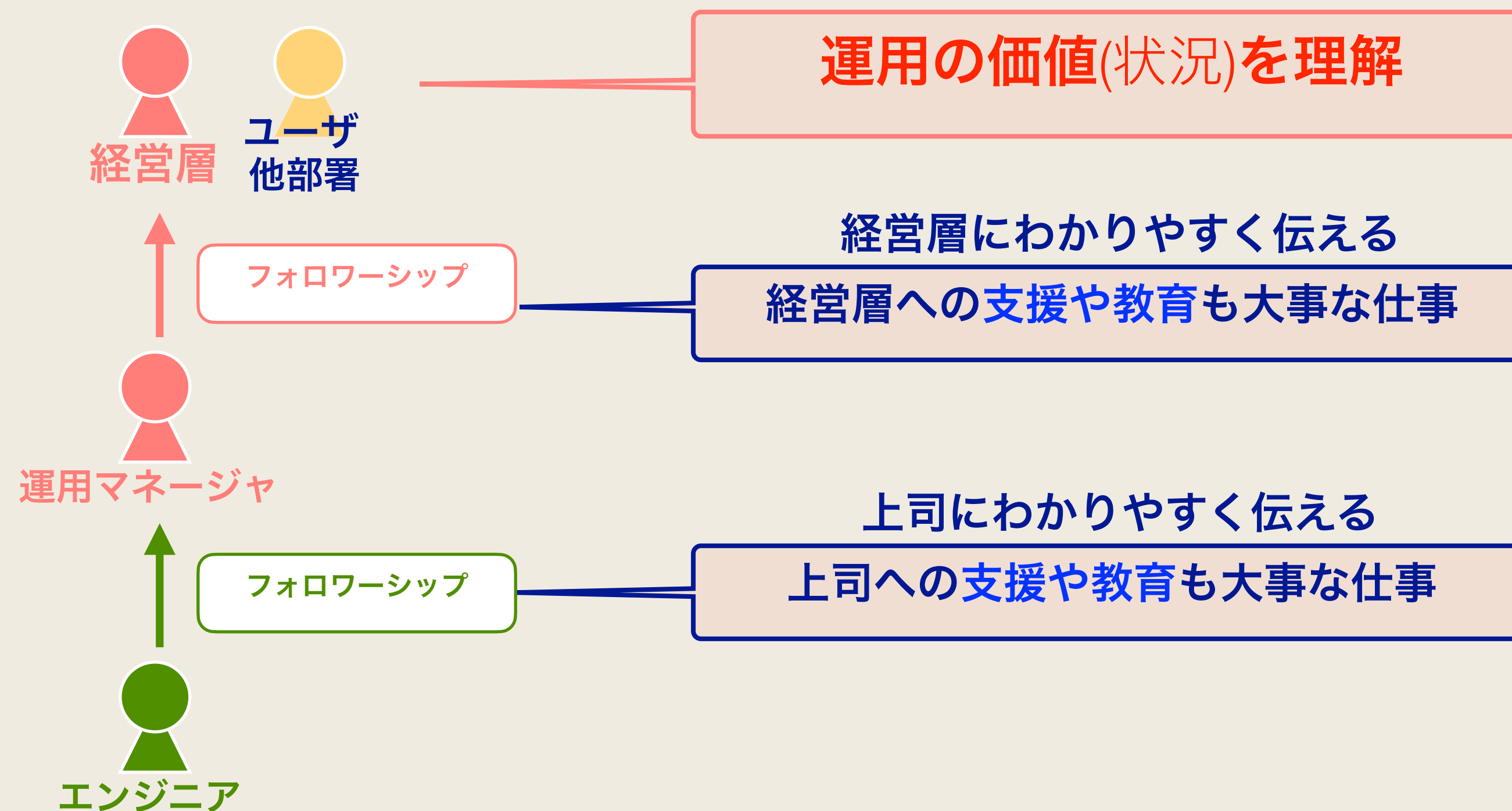


組織が最大成果を挙げるためにお互いを活用する意識

エンジニアと上司-部下、先輩-後輩の関係

「フォロワーシップ」で現場の実情を経営層に伝える

現場からの「フォロワーシップ」で現場の課題感を経営層に伝えることも重要



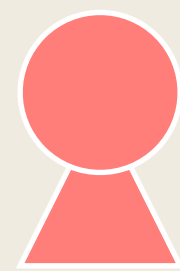
「上司やお客様を教育する」という視点も重要になってきている

エンジニアと上司-部下、先輩-後輩の関係

「上司を上手く使う」「部下に上手く使われる」という視点

ビジネスモデルの

設計者



経営層



ユーザ

ビジネスモデルの

受益者

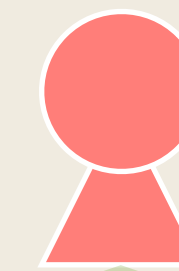
提案

要求

成果

現場から提案して成果を産むこともできる
(上司は、提案プロキシの役割)

上司/リーダー



提案

部下



エンジニア

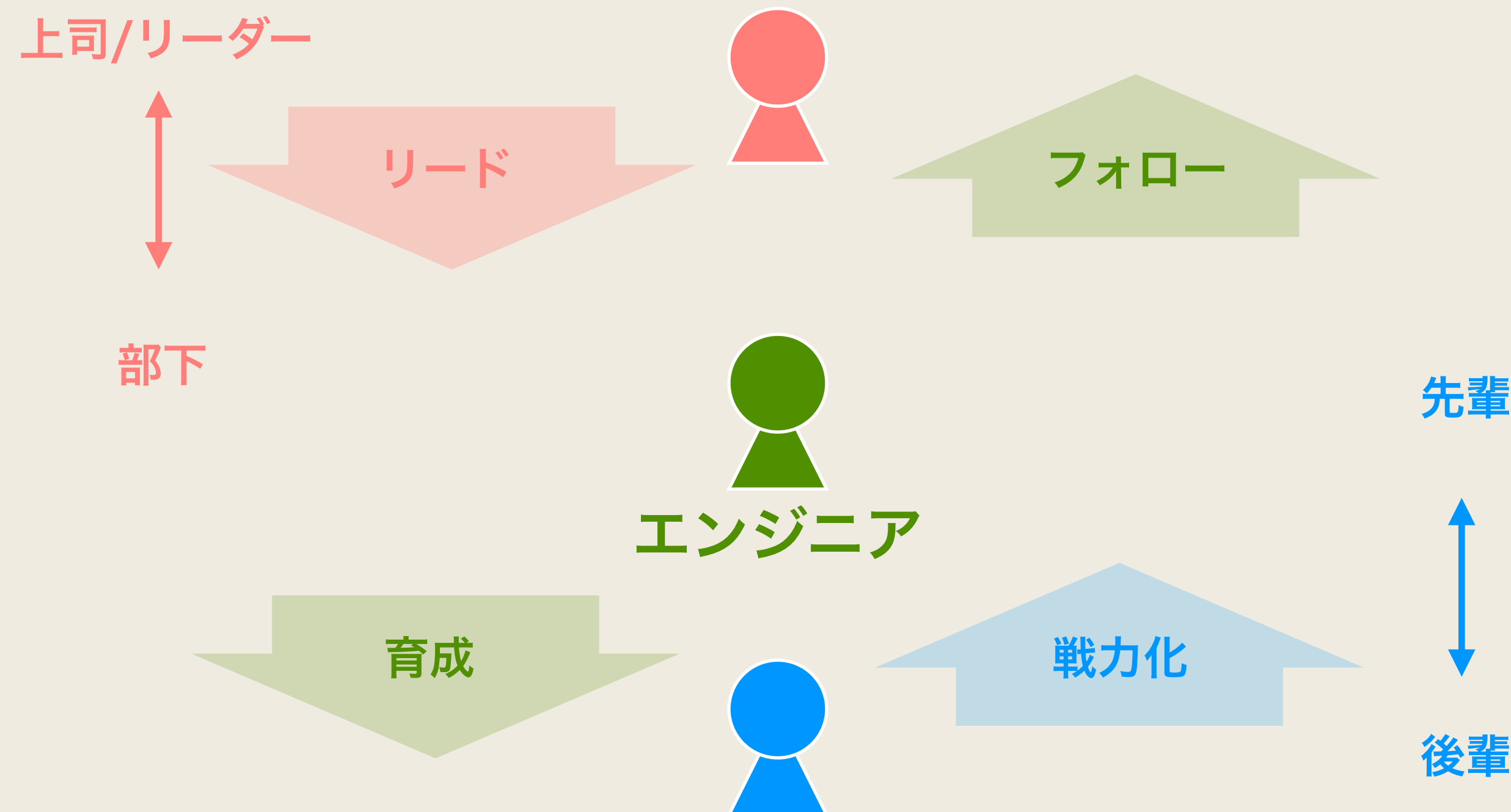
ビジネスモデルの
実践者

上司が提案を通せるかどうかは
エンジニアが渡す「武器」(資料や根拠)次第

上司が無事帰ってくる武器を渡すのが
エンジニアの腕前

エンジニアと上司-部下、先輩-後輩の関係

業務上の役割分担と割り切って、**全体のパフォーマンスが最大**になるように動く



上司を上手く使う

上司への提案により自分のやりたい仕事を得やすくなる

上司や後輩に上手く使われる

上司や後輩との仕事がスムーズになる

後輩を上手く戦力化する

後輩の戦力化により自分の仕事が楽になる

Step2. 運用エンジニアの役割と特徴

「エンジニア」とは

まとめ

ナビゲーション

運用エンジニアの根本に関わる概念

エンジニアリングとは

運用エンジニアの役割と特徴

「エンジニア」とは

1. 「エンジニアリングする人」がエンジニア
2. 「エンジニア」というビジネス
3. エンジニアと上司-部下、先輩-後輩の関係

まとめ

運用エンジニアのキャリア

エンジニアとしてのステージ

ヒトとしてのステージ



まとめ: 「エンジニア」とは

「エンジニアリング」をビジネスにして報酬を得ているプロフェッショナル

客観的: 意見や分析を客観で語る

上司を上手く使う

上司への提案により自分のやりたい仕事を得やすくなる

合理的: 物事を原理原則や事実で語る

上司や後輩に上手く使われる

上司や後輩との仕事がスムーズになる

論理的: 論理的な説明ができる

持続的: 持続的な設計や実装ができる

後輩を上手く戦力化する

後輩の戦力化により自分の仕事が楽になる

Step2. 運用エンジニアの役割と特徴

まとめ

まとめ: 運用エンジニアの役割と特徴

運用組織のリソースを活用し、対価や評価を得ることを目的に、
外部に対して、継続的に何らかのサービスを提供し続ける



事業継続に必要なリソース活用は
「運用業務」で行います

「企業やサービスの事業継続」
を実現する役割

サービスやシステムに
常に最も近い

エンジニアリングと
非常に相性が良い

客観的: 意見や分析を客観で語る

合理的: 物事を原理原則や事実で語る

論理的: 論理的な説明ができる

持続的: 持続的な設計や実装ができる

上司を上手く使う

上司や後輩に上手く使われる

後輩を上手く戦力化する

Step3. 運用エンジニアのキャリア

ナビゲーション

運用エンジニアの根本に関わる概念

エンジニアリングとは

運用エンジニアの役割と特徴

「運用」とは

「エンジニア」とは

運用エンジニアのキャリア

エンジニアとしてのステージ

ヒトとしてのステージ



運用エンジニアのキャリア

運用業務に関わるエンジニアとしてのキャリアを考える上で、
**現在から10年後までの間に
どのようなキャリアパスが考えられるでしょうか？**

自由な切り口で記述してみましょう

Step3. 運用エンジニアのキャリア

1. エンジニアとしてのステージ

ナビゲーション

運用エンジニアの根本に関わる概念

エンジニアリングとは

運用エンジニアの役割と特徴

「運用」とは

「エンジニア」とは

運用エンジニアのキャリア

エンジニアとしてのステージ

ヒトとしてのステージ



エンジニアとしてのステージ

「エンジニアとしての将来」が見えなくて困っていませんか？



中堅エンジニア

自分の今後のキャリアが見通せない

キャリアモデルになる先輩や上司がいない、見当らない

今の仕事だけに集中していて大丈夫なのか不安がある



若手エンジニア

先輩のキャリアを参考にしたいが教えてくれない

エンジニアとしてのステージ

「エンジニアとしての将来」が見えないと何に困るか考えてみました (実例)

1. 「今、何をしておくと将来楽になるのか」がわからない
2. 備えができていないため、いずれは能力や収入が頭打ち・右肩下がりになる
3. (人によっては) モチベーションが沸かない、将来が不安になる

エンジニアとしてのステージ

エンジニアとして求められる成果を「ステージ」として整理してみました

求められる成果

ステージ5

トップエンジニア

業界を代表してのアウトプット

ステージ4

シニアエンジニア

企業を代表してのアウトプット

ステージ3

ミドルエンジニア

機能(チームや職能)としてのアウトプット

ステージ2

レギュラーエンジニア

個人としてのアウトプット

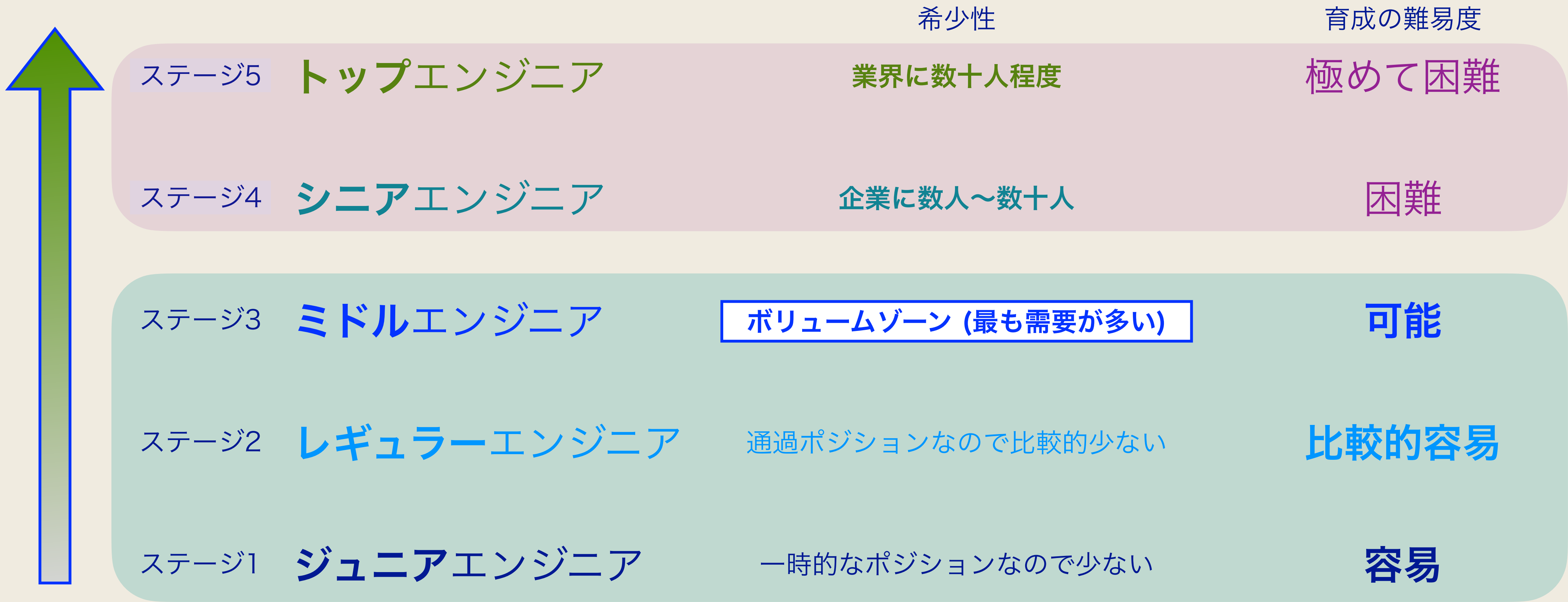
ステージ1

ジュニアエンジニア

「学ぶ姿勢」(インプット重視)

エンジニアとしてのステージ

エンジニアとして希少性と育成の難易度を整理してみました



エンジニアとしてのステージ

ステージ1「ジュニアエンジニア」

ジュニアエンジニア

エンジニアとして最初のステージで、「見習い」(戦力外)に相当する

特徴	努力が評価される (給料をもらって勉強している) 会社は赤字
求められる成果	「学ぶ姿勢」(インプット重視)
育成すべき力	- 読解力 - 学習能力 資格取得による基礎固めが効果的な時期である
人との繋がり	相談できる先達を探す

エンジニアとしてのステージ

ステージ2 「レギュラーエンジニア」

レギュラーエンジニア

エンジニアとして一人前となるステージで、正規の戦力として扱われる

特徴

上司や先輩の指示通りの仕事ができるようになる（給料相当の仕事ができる）
3年未満で辞められると会社は赤字

求められる成果

個人としてのアウトプット

育成すべき力

- ・ 論述力
 - ・ 知識と経験 (バランス良く育成)
- 資格取得と実力養成の両建てが効果的な時期である

人との繋がり

相談できる同僚を増やす

エンジニアとしてのステージ

ステージ3 「ミドルエンジニア」

ミドルエンジニア

得意領域が明確なエンジニアとなるステージで、現場の主要戦力として扱われる

特徴	自分の強みを理解して社内公式的に特化していく (会社の利益の源泉となる)		
求められる成果	機能(チームや職能)としてのアウトプット		
育成すべき力	- 説得力 - 知識と経験 (バランス良く育成) 資格取得よりも実力養成に比重を移していく時期である		
人との繋がり	相談・信頼できる後輩(レギュラーエンジニア)を育てる		

エンジニアとしてのステージ

現在・未来のエンジニア像を「ステージ」で考える・予測する

自分の**キャリア**を考える時間を確保するようにしよう

現在の自分の位置を把握する

未来の自分の位置を想像する

望むキャリア(未来の位置)を**実現するために必要・不足しているものを把握**しよう

必要な知識や経験は何か？

必要な知識や経験をどのようにして得るか？

必要な**知識や経験**を着実に**積み上げ**ていこう

エンジニアとしてのステージ

チャンスは「準備ができている人」に訪れる

外部起因: 準備ができている人にしか**誘い**が来ない

準備のできていない人: **チャンス**を提供する人がいない

準備ができている人: **次々と誘い**が来る

能動起因: 準備ができている人しか**チャンス**を**獲得**できない

準備のできていない人: **重要な仕事**を任されることは無い

準備ができている人: 狙った**チャンス**を**容易に獲得**することができる

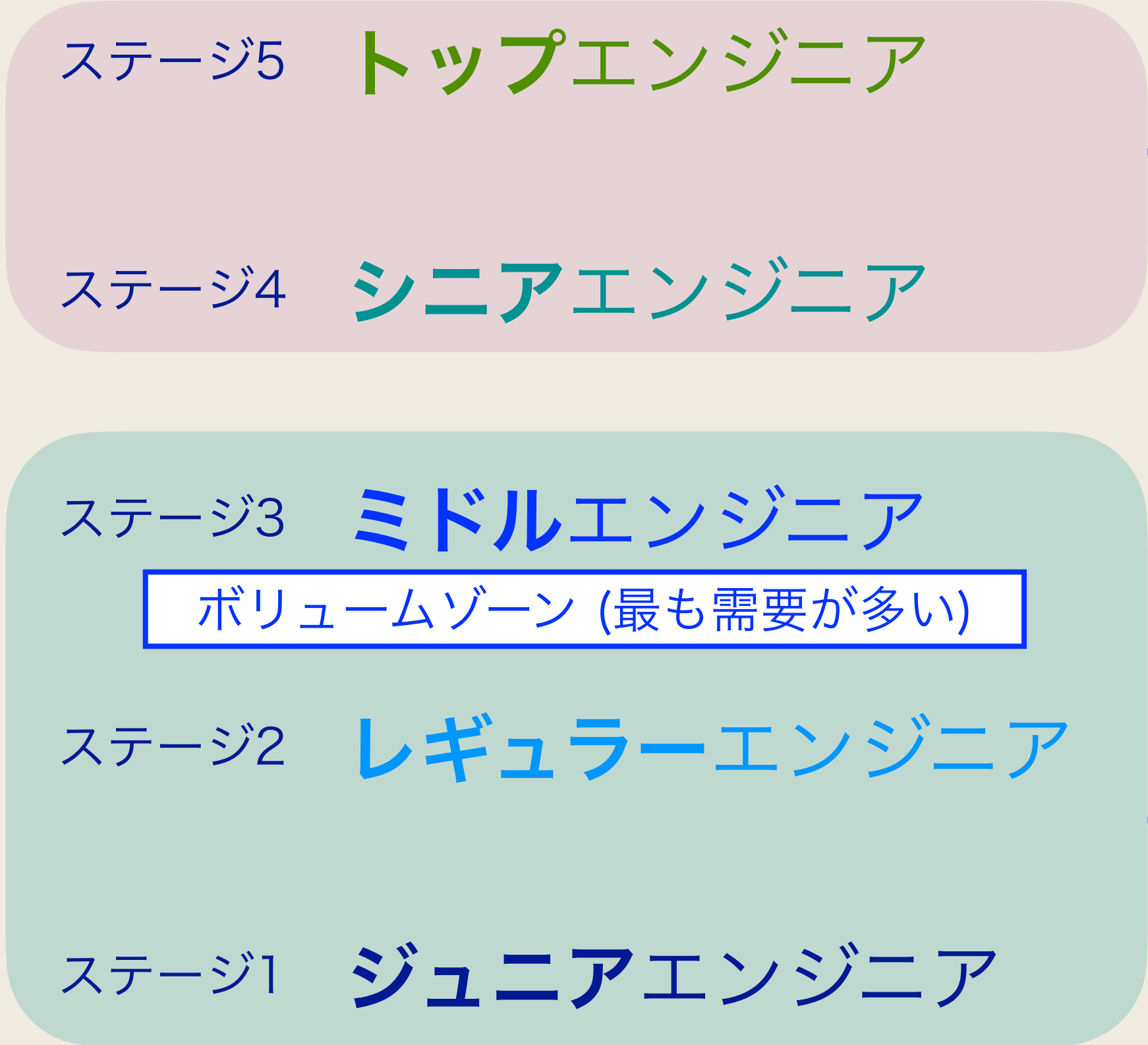
受動起因: 準備ができている人しか**チャンス**に**気付**かない

準備のできていない人: 「それが**チャンス**である」と**気付く**ことができない

準備ができている人: **チャンス**に**気付**いて**モノにする**ことができる

エンジニアとしてのステージ

キャリアパスを意識し、常に準備しておくことで、チャンスを掴みやすくなる



現在・未来のエンジニア像を
「ステージ」で考える・予測する

↓

チャンスは
「準備ができている人」に訪れる

Step3. 運用エンジニアのキャリア

2. ヒトとしてのステージ

ナビゲーション

運用エンジニアの根本に関わる概念

エンジニアリングとは

運用エンジニアの役割と特徴

「運用」とは

「エンジニア」とは

運用エンジニアのキャリア

エンジニアとしてのステージ

ヒトとしてのステージ



ヒトとしてのステージ

「ヒトとしての将来」を考えたことはありますか？

30代前半で徹夜ができなくなった...

40歳前後で老眼がはじまった...

40代後半で血圧が高くなってきた...

50代半ばで病気持ちが増えてきた...

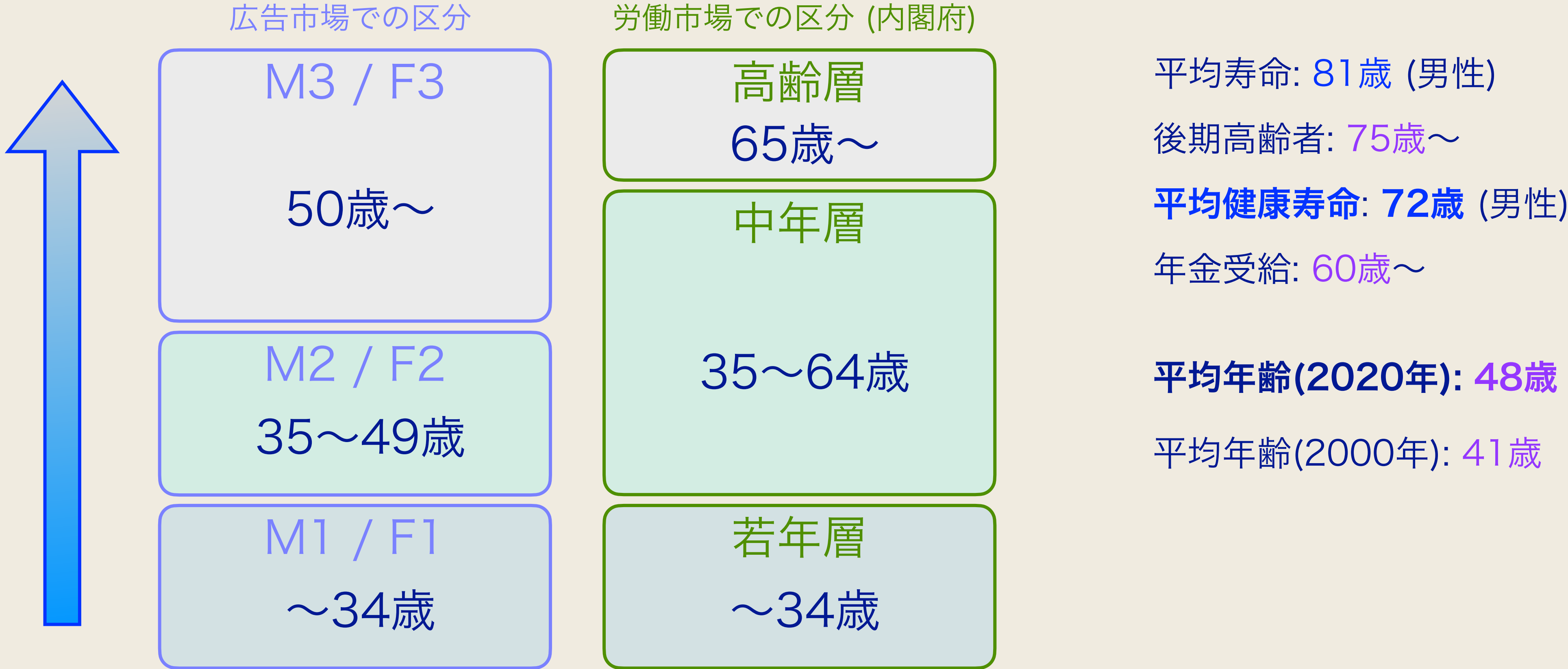
ヒトとしてのステージ

「ヒトとしての将来」を把握していないと何に困るか考えてみました (実例)

1. 徹夜などを前提に組んだ**計画の通りには進まない**
2. 備え無しにある日突然、**身体機能低下や健康不安**に襲われる
3. あっという間に年齢を重ねてしまい、**やり直しが効かなくなる**

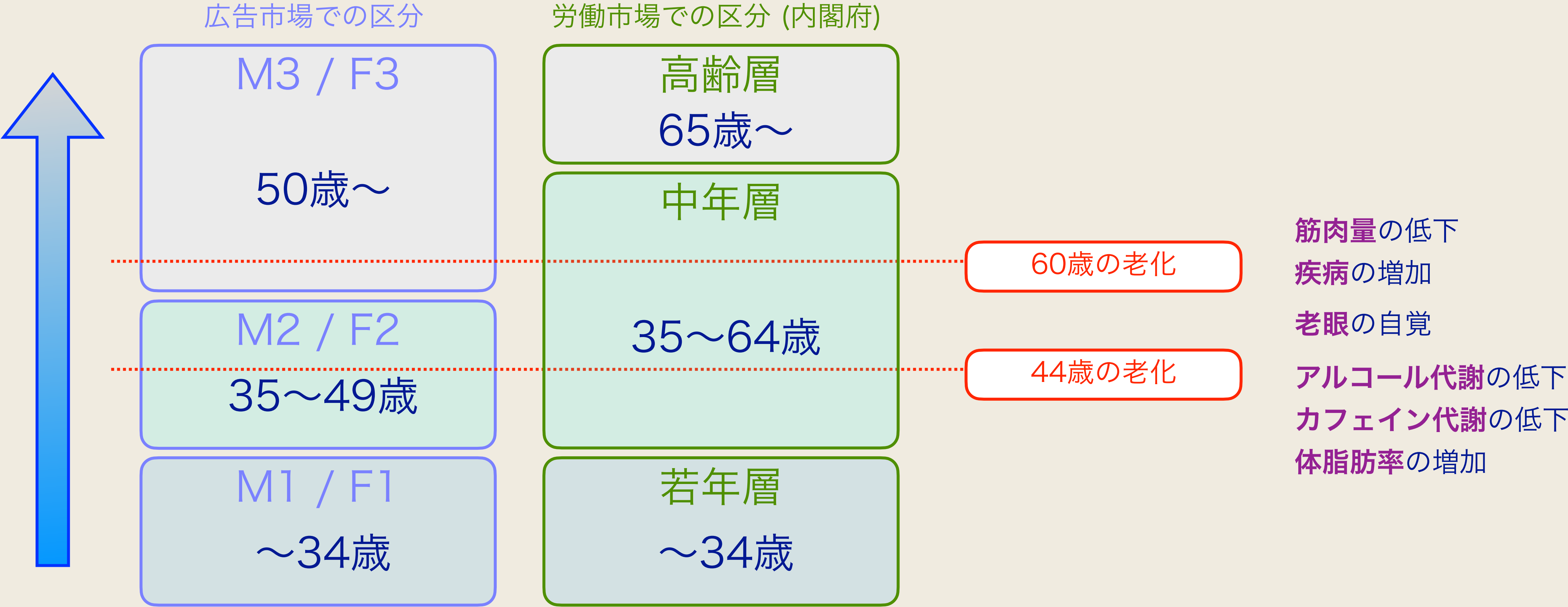
ヒトとしてのステージ

広告市場や労働市場などでの「ライフステージ」を調べてみました



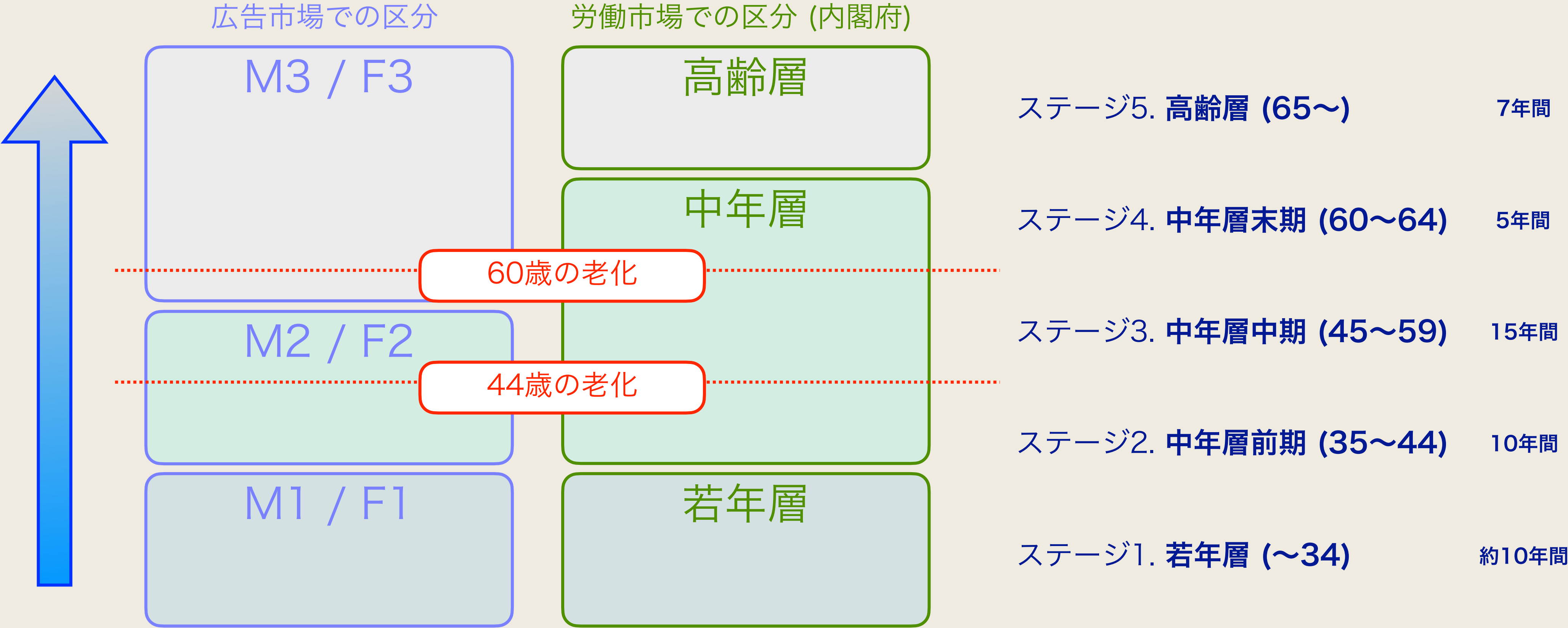
ヒトとしてのステージ

「老化」のタイミングを調べてみました



ヒトとしてのステージ

「ライフステージ」と「老化」を整理してみました



ヒトとしてのステージ

「ライフステージ」それぞれの特徴と制約を知る

ステージ5 高齢層 (65～)

ステージ4 中年層末期 (60～64)

ステージ3 中年層中期 (45～59)

ステージ2 中年層前期 (35～44)

ステージ1 若年層 (～34)

人生区分の特徴

お金、知識、経験が豊富でやれる事が多いが
時間と体力が足りない

お金、知識、経験、時間、体力の
バランスが一番良い時期

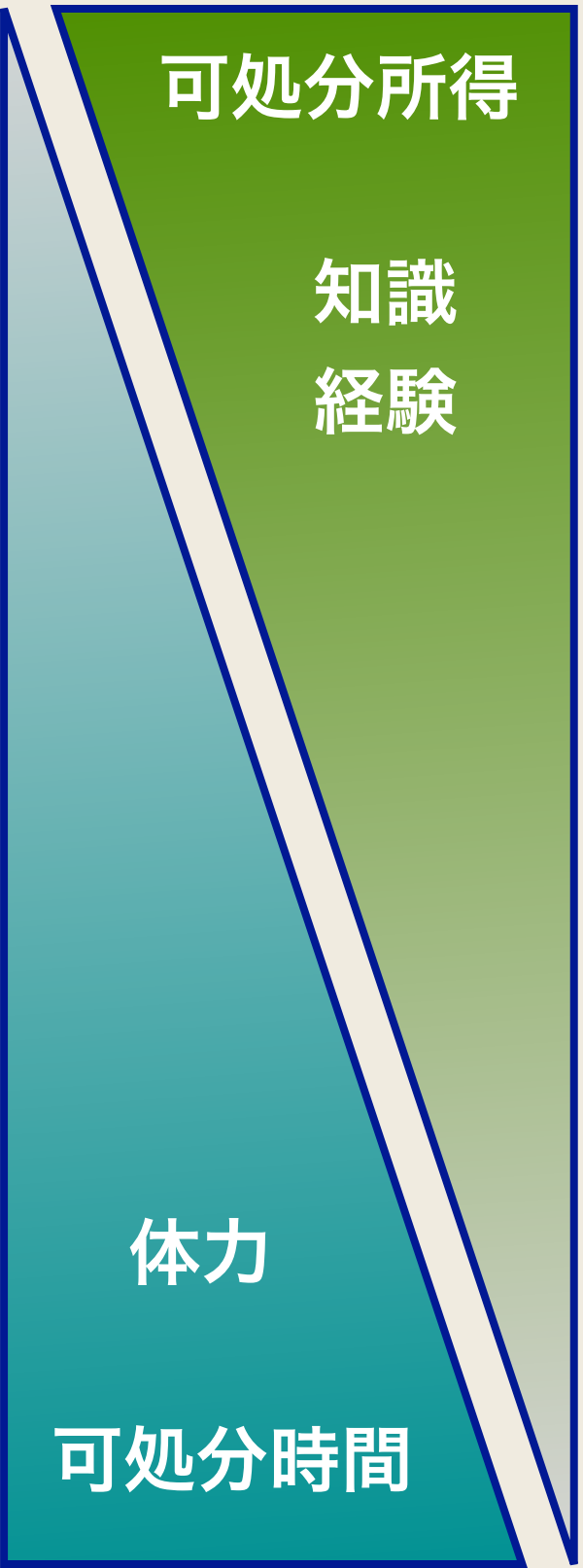
お金、知識、経験が少ないが
時間と体力は豊富

人生区分による制約

人生のリスクには慎重になる
徹夜による消耗は極めて大きい

人生のリスクには柔軟に対応できるが
徹夜できなくなる

人生のリスクを取りやすく
徹夜もできる



ヒトとしてのステージ

自分というリソースの生物上のステージに応じたエンジニアリング戦略が必要



エンジニアの人的資本を向上させるのは健康と体力

ヒトとしてのステージ

現在のライフステージの特徴や制約に応じた
エンジニアリング戦略を選択することが重要

ステージ3 中年層中期 (45～59)

お金、知識、経験が豊富でやれる事が多いが
時間と体力が足りない

ステージ2 中年層前期 (35～44)

お金、知識、経験、時間、体力の
バランスが一番良い時期

ステージ1 若年層 (～34)

お金、知識、経験が少ないが
時間と体力は豊富

Step3. 運用エンジニアのキャリア

まとめ

まとめ: 運用エンジニアのステージ

現在・未来のエンジニア像を
「ステージ」で考える・予測する



現在・未来のライフステージの
特徴や制約に応じた戦略を考える



チャンスは
「準備ができている人」に訪れる

おわりに: 運用エンジニアの活躍の場

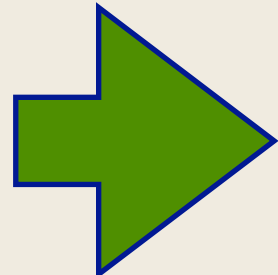
「運用」は楽しい

適切な「エンジニアリング」によって
人がわかりやすい運用、システムが処理しやすい運用を考えて実現し続ける

ありがちな運用業務

- 1. 人が理解しやすい業務ではない。
- 2. システムが扱いやすい業務ではない。
- 3. 論理破綻・矛盾による無駄・無意味が多い。

- 人材がスケールしない
- 業務がスケールしない
- 工数を最大活用できない



あるべき運用業務

- 1. 人が理解しやすい業務である。
- 2. システムが扱いやすい業務である。
- 3. 論理的に正しいことを検証している。

- 人材がスケールする
- 業務がスケールする
- 工数を最大活用できる

適切な運用設計がされていない

適切な運用設計がされている

目に見えて結果が出るし、現場の人の反応も良いので楽しい

運用エンジニアの活躍の場は広がっている

運用の無いサービスは存在しないため

ビジネス、コミュニケーション、エンジニアリングの力があれば、どこでも活躍ができる

クラウドの普及により

ハードウェアの制約が少ないエンジニアリングの機会が増えている

「運用エンジニア」という呼称でなくても

「運用に関わるエンジニアリング」の活用機会はどんどん増えている

運用のスキルは陳腐化しにくい

スキルは、その特性によって賞味期限が違う

変化の激しい技術領域のスキルは**急速に陳腐化**しやすい

運用に求められるスキル(持続性、反復再現性、サービス視点など)は
特化した技術スキルに比べて**陳腐化しにくい**

まとめ

この講義で一番大切なこと

「エンジニアリング」によって
「情報」というビジネス上の価値が生まれる

この講義の全体ふりかえり

Step1. エンジニアリングとは

適切なQCDで

反復再現的な処理方法で

持続可能な形で

情報の大量処理を実現する

Step2. 運用エンジニアの役割と特徴

運用組織のリソースを活用し、対価や評価を得ることを目的に、
外部に対して、継続的に何らかのサービスを提供し続ける

Step3. 運用エンジニアのステージ

チャンスは「(エンジニアとして、ヒトとして)準備できている人」に訪れる

もう一度確認しましょう

この講座ではどのような内容を受け取りましたか？

その中で一番**大事な事**は何でしたか？

ステップアップ

受講内容を活用して「未来像」を描こう

現在・未来のエンジニア像を
「ステージ」で考える・予測しよう



現在・未来のライフステージの
特徴や制約に応じた戦略を考えよう



チャンスは
「準備ができている人」に訪れる

受講内容を活用して「実際の行動」を行おう

「エンジニアリング」をビジネスとするプロフェッショナルになろう

客観的: 意見や分析を客観で語る

上司を上手く使う

上司への提案により自分のやりたい仕事を得やすくなる

合理的: 物事を原理原則や事実で語る

上司や後輩に上手く使われる

論理的: 論理的な説明ができる

上司や後輩との仕事がスムーズになる

持続的: 持続的な設計や実装ができる

後輩を上手く戦力化する

後輩の戦力化により自分の仕事が楽になる

この講義から更にその先へ

第1回
(今回) 運用エンジニアのキャリアを考えよう

第2回 ジュニアエンジニア 現在ジュニア、レギュラーエンジニアの方
これからレギュラーエンジニアになる方

第3回 レギュラーエンジニア ジュニア、レギュラーエンジニアを育成する方

第4回 ミドルエンジニア 現在ミドルエンジニアの方
これからミドルエンジニアになる方
ミドルエンジニアを育成する方