

「運用エンジニア」のためのエンジニアリング入門 3

ミドルエンジニアの成長ステップ

運用設計ラーニング

この講義の対象者

ミドルエンジニア

- ・ サービス運用に関わるエンジニアを目指す人
- ・ サービス運用に関わるエンジニアになりたての人
- ・ 基礎スキルの一部が不足していると感じているエンジニア

ミドルエンジニアを指導・育成する立場の方

- ・ ミドルエンジニアの指導・育成をする先輩や上司

悩み: ミドルエンジニアの成長ステップ

- ・ ミドルエンジニアには**どんなことが期待されているのか?**
- ・ **何をどのように積み上げていけば、期待に応えることができるようになるのか?**
- ・ 次のステップに進むためには、**どんな考え方、知識、経験が必要なのか?**

理想: ミドルエンジニアの成長ステップ

正しい「ミドルエンジニア」になり、次のステージに進む

- ・ ミドルエンジニアとして、不足している部分、伸ばすべき部分について、育成する行動をするようになる。
- ・ 次のステージを意識して、可能な限り準備をしておく。

ミドルエンジニアを指導・育成する立場(先輩や上司)の方

- ・ 指導・育成対象のミドルエンジニアに不足している部分、伸ばすべき部分を考慮し、適切に指導・育成できる。
- ・ 指導・育成対象のミドルエンジニアに、次のステージを意識して、可能な限り準備させておくことができる。

この講座で得られるもの

- ・ ミドルエンジニアに対する**期待が明確**になる。
- ・ ミドルエンジニアに**必要な考え方を理解**できる。
- ・ ミドルエンジニアに**必要な知識、経験の方向性が明確**になる。

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

なぜエンジニアの成長ステップを学ぶのか？

学ぶ理由

将来への不安を解消し、**自信を持って着実に成長**していくため

- ・ 現在・未来のエンジニア像を把握・予測し、将来への不安を減らすため
- ・ 現在・未来のライフステージに適した成長戦略を考えるため
- ・ 次のステップへの準備を常にしておくことで、チャンスを掴みやすくするため

ミドルエンジニア以上でも、**不足している部分があれば補完**するため

この講義で学習すること

講義の要素マップ

ミドルエンジニアの概要

ミドルエンジニアの成長ステップ

レベル1. 導入期

レベル2. 確立期

この講座のロードマップ

スタートとゴール

スタート

ミドルエンジニアとしての成長や育成に悩んでいる

ミドルエンジニアとはどのような立場・役割なのか
を理解する

ミドルエンジニアの概要

主体的にミドルエンジニアとしての成長を切り開くための
ステップや考え方を理解する

ミドルエンジニアの
成長ステップ

ゴール

目標や自信を持ってミドルエンジニアとしての成長や育成を進めることができる

この講座のロードマップ

3つのステップ

Step1

ミドルエンジニアの概要

Step2

レベル1. 導入期

Step3

レベル2. 確立期

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. ミドルエンジニアの概要
- Step2. レベル1. 導入期
- Step3. レベル2. 確立期

Step 1. 「ミドルエンジニア」の概要

はじめに

学習ガイド

本編

まとめ

ステップアップ

ナビゲーション

Step1

ミドルエンジニアの概要

Step2

レベル1. 導入期

Step3

レベル2. 確立期

はじめに

学習ガイド

本編

まとめ

ステップアップ

ミドルエンジニアの概要

ミドルエンジニアは
初級エンジニア(ジュニア/レギュラー)と
どんな点が異なるのでしょうか？

大きなものを3つ考えてみてください。

ミドルエンジニアの概要

ミドルエンジニアに何が求められるのかわからない

ミドルエンジニアになったが、何から手を付けていいのかわからない

まず最初に何が求められるのかわからない

「ミドルエンジニアとして及第点」だと認められるための基準がわからない

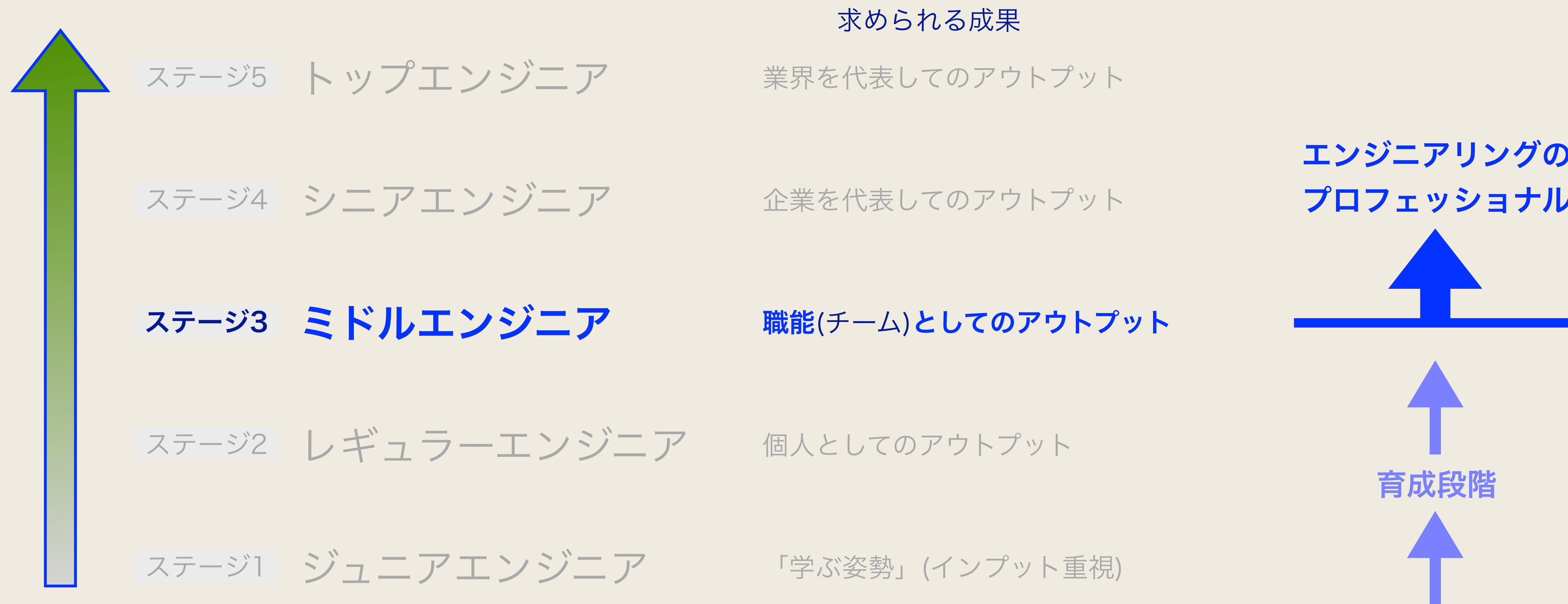
どこまでやればミドルエンジニアとして認められるのかわからない

シニアエンジニアになるために、ミドル時代にやるべきことを知りたい

ミドルエンジニアの時代にどんな知識やスキルを身に付けておけば、将来有利なのか知りたい

ミドルエンジニアの概要

ミドルエンジニアの位置付け



ミドルエンジニアの概要

ミドルエンジニアの希少性と育成の難易度

		希少性	育成の難易度
↑	ステージ5 トップエンジニア	業界に数十人程度	極めて困難
	ステージ4 シニアエンジニア	企業に数人~数十人	困難
	ステージ3 ミドルエンジニア	ボリュームゾーン (最も需要が多い)	可能
	ステージ2 レギュラーエンジニア	通過ポジションなので比較的少ない	比較的容易
	ステージ1 ジュニアエンジニア	一時的なポジションなので少ない	容易

ミドルエンジニアの概要

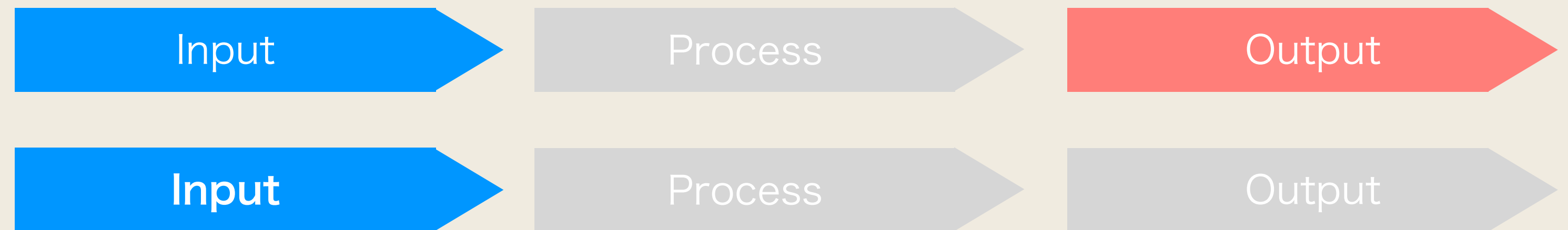
ミドルエンジニアと初級エンジニアの違い

レギュラーまでは、**インプットありき**でアウトプット

レギュラーエンジニア

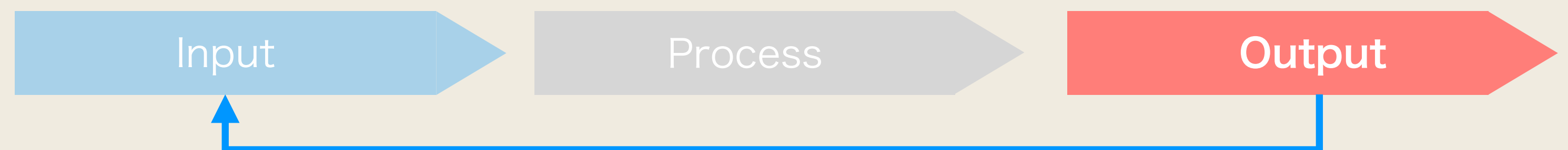
初級エンジニア

ジュニアエンジニア



ミドルからは、**アウトプットありき**でインプット

ミドルエンジニア



職能として**結果**を出さなければならない (=報酬)

ミドルエンジニアの概要

「職能」とは

- ・ **職務を果たす能力。**
- ・ 社会・企業などの大きな枠組みの中で、その**職業・職務の果たす役割。**
- ・ **もののはたらき。**あるものの中で果たす役割。

(出典: スーパー大辞林)

職能給

従業員の**職務遂行能力**に基づいて決められる**給与**

職能制

企業で、資材調達・生産・販売など**各職能を単位とする組織編成** (⇔事業部制)

職能組織

企業経営において、人事部・製造部・販売部といった、仕事の種類や**専門**に基づいて編成した**組織**。
ファンクショナル-オーガニゼーション。(=機能別組織)

ミドルエンジニアの概要

ミドルエンジニアの「職能」とは

職能組織

企業経営において、仕事の種類や**専門**に基づいて編成した**組織**。
ファンクショナル-オーガニゼーション。(=機能別組織)

ミドルエンジニアの職能 エンジニアリングの**専門領域**に基づいて割り当てられる**役割**

(職能組織に属する場合)

組織の**主戦力**として、組織に求められる「**機能**」を実現し続ける**役割**

(事業部組織に属する or 単独の場合)

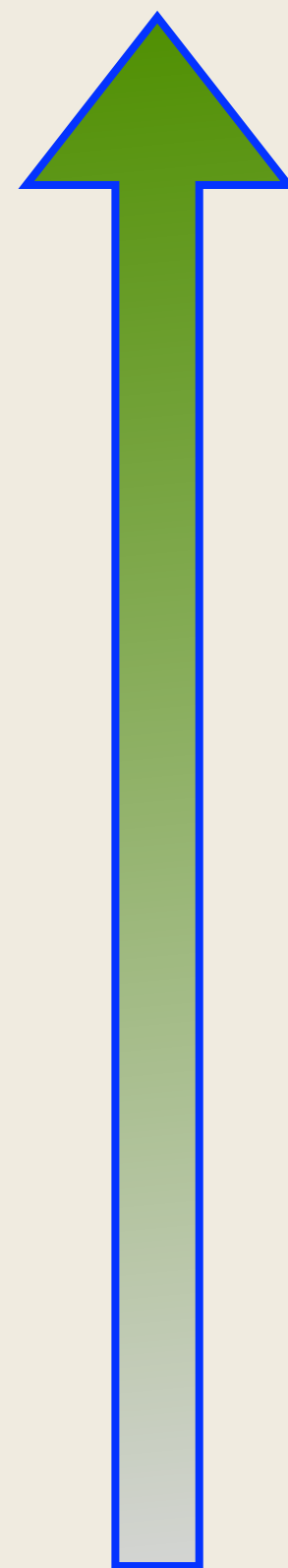
役割に求められる「**機能**」を実現し続ける**役割**

ミドルエンジニア = ファンクションの実体

「組織の歯車」というよりは「組織のエンジン」というイメージ

ミドルエンジニアの概要

ミドルエンジニアの「職能」の特徴



ステージ5 トップエンジニア

ステージ4 シニアエンジニア

ステージ3 **ミドルエンジニア**

ステージ2 レギュラーエンジニア

ステージ1 ジュニアエンジニア

専門領域が抽象的になっていく
ことが多い

ボリュームゾーン (最も需要が多い)

**最も専門領域
が具体的**

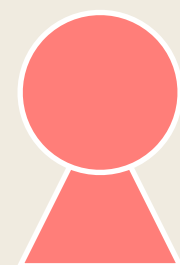
専門性が薄い

ミドルエンジニアの概要

「機能」を実現し続ける役割

ビジネスモデルの

設計者



経営層



要求



ユーザ

ビジネスモデルの

受益者



成果

機能

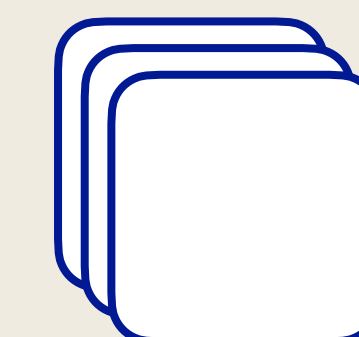
ミドルエンジニアは、
人的リソースや各種リソースを活用して
求められる「機能」を実現し続ける



ミドルエンジニア

主戦力

ビジネスモデルの
実践者



各種リソース



後輩

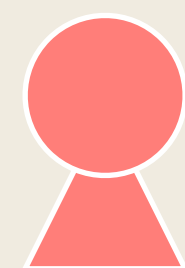


常駐さん

ミドルエンジニアの概要

求められる機能を「実現し続け」ないと？

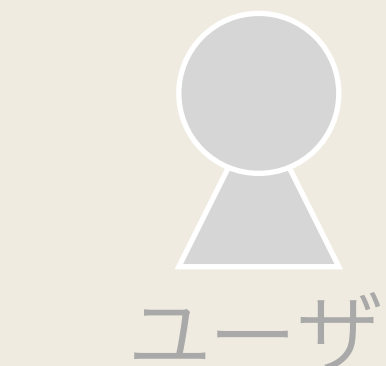
ビジネスモデルの
設計者



経営層

機能を実現していない
なら不要

要求



ユーザ

ビジネスモデルの
受益者

成果

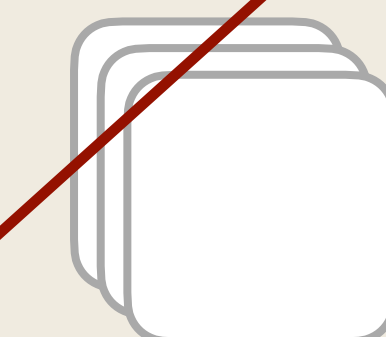
機能

働いている人やリソースが不要
とされてしまう

主戦力

ビジネスモデルの
実践者

ミドルエンジニア



各種リソース



後輩

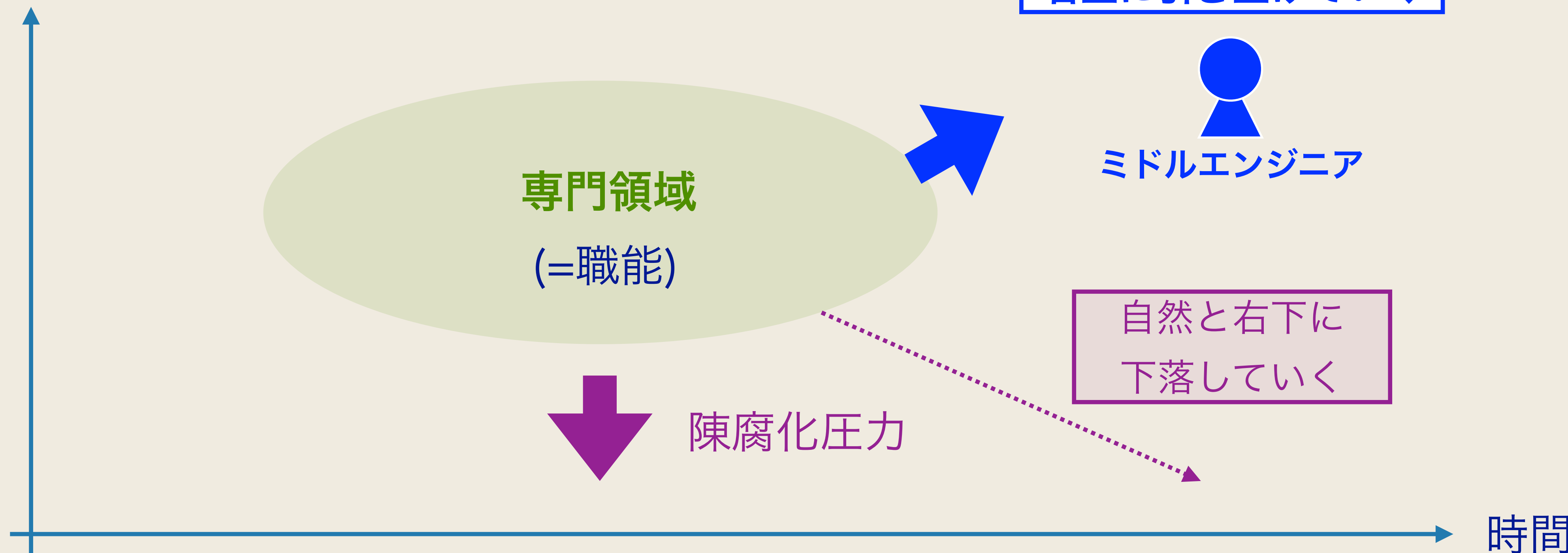


常駐さん

ミドルエンジニアの概要

求められる機能を「実現し続ける」とは

ビジネス価値



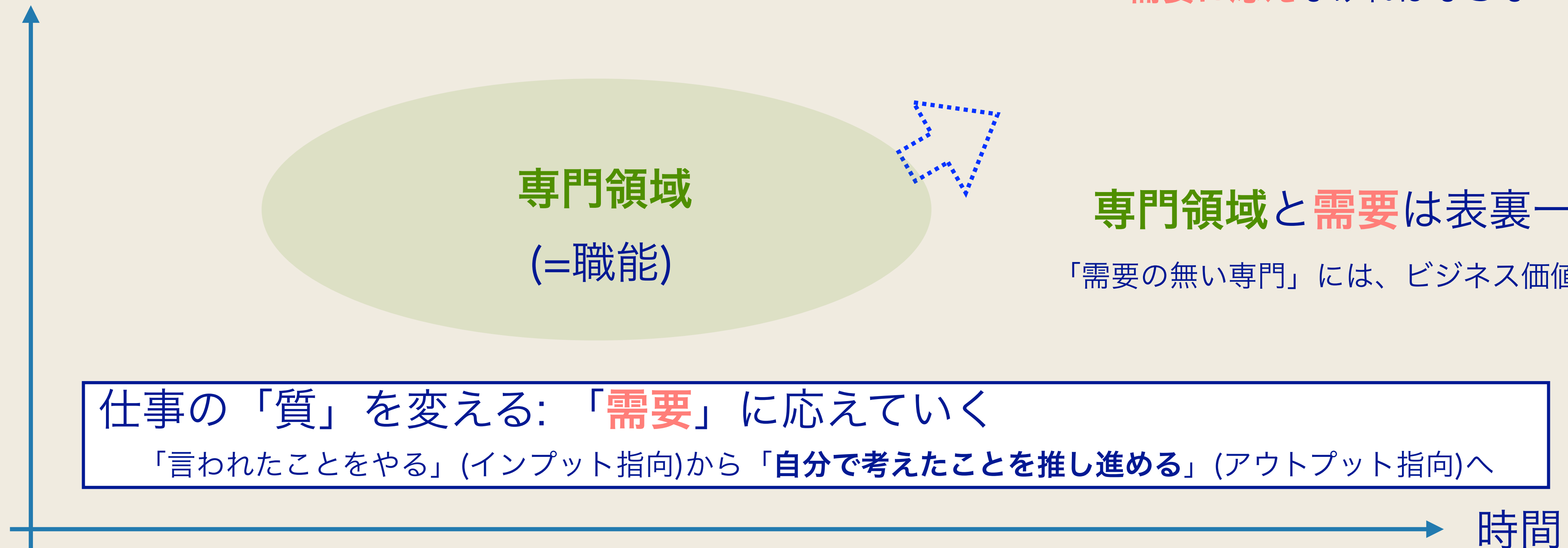
ミドルエンジニアの概要

求められる機能を「実現し続ける」ためにミドルエンジニアがやるべきこと

ビジネス価値

ビジネス価値=需要に応えること

プロフェッショナルとして結果を出すには
需要に応えなければならない



専門領域
(=職能)

専門領域と需要は表裏一体

「需要の無い専門」には、ビジネス価値が無い

仕事の「質」を変える: 「需要」に応じていく

「言われたことをやる」(インプット指向)から「自分で考えたことを推し進める」(アウトプット指向)へ

時間

「やりたいこと」をやるなら、その需要を創り出さないといけない

需要がないものは「仕事」にはならない

ミドルエンジニアの概要

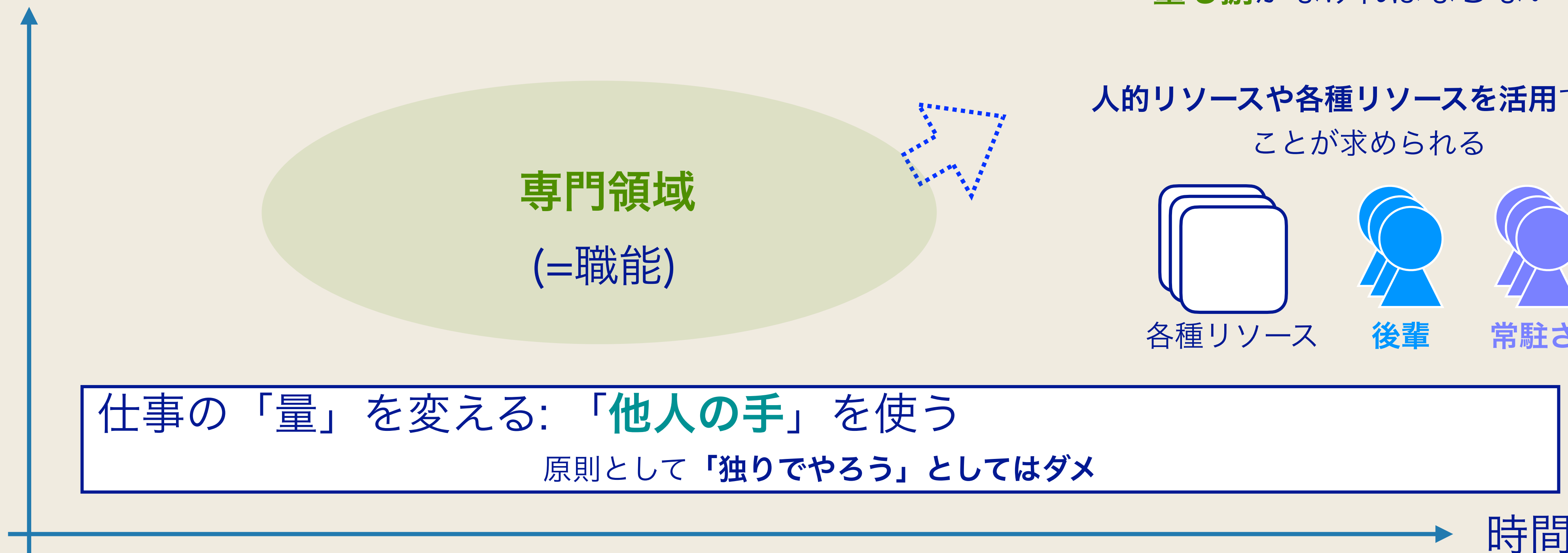
求められる機能を「実現し続ける」ためにミドルエンジニアがやるべきこと

ビジネス価値

個人プレイでは早々に限界

プロフェッショナルとして結果を出すには

量も捌かなければならない



人的リソースや各種リソースを活用できることが求められる



各種リソース



後輩



常駐さん

仕事の「量」を変える: 「他人の手」を使う
原則として「独りでやるう」としてはダメ

ミドルエンジニアには、特に「コミュニケーション」が求められる

「コミュ障」が許されたのはIT黎明期だけ

ミドルエンジニアの概要

ミドルエンジニアになったが、何から手を付けていいのかわからない

需要に応えるために必要な事項や将来像を明確にし、上司や周囲を説得できることが求められる

仕事の「質」を変える: 「**需要**」に応じていく

需要に応じて、自律的に周囲を説得して、職能の強化を主導できる

説得力の確立

相手を納得させるだけの力。その力のある話し方や論理の展開のしかた。

レベル1 [導入期]

需要に応え、自律的に「職能を強化」し続ける

これができるまでは
ローミドル扱い

ミドルエンジニアの概要

「ミドルエンジニアとして及第点」だと認められるための基準がわからない

エンジニアとしての成果を、個人名ではなく「**職能名**」(チーム名)で出すことが求められる

仕事の「量」を変える: 「**他人の手**」を使う

チームの主力として、**チーム成果を主導**するとミドルエンジニアとして**及第点**

個人仕事からの脱却

レベル2 [確立期]

「**個人の仕事**」から「**職能の仕事**」へ切り替える

これができると
「及第点」扱い

ミドルエンジニアの概要

シニアエンジニアになるために、ミドル時代にやるべきことを知りたい

ミドルエンジニアとしては必須ではないが、**将来の伸び代に大きく影響する活動**がある

他のエンジニアとの**差別化を図るための準備**をする

他のミドルエンジニアがあまりやらないような**インプット**

他のミドルエンジニアがあまりやらないような**プロセス**

他のミドルエンジニアがあまりやらないような**アウトプット**

レベル3 [成熟期]

シニアへの伸び代を準備しておく

ハイミドル

ミドルエンジニアの概要

ミドルエンジニアとは

ミドルエンジニア

得意領域が明確なエンジニアで、**職能(チーム)の主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

職能(チームや機能)としてのアウトプット (職能の範囲で成果責任を負う)

育成すべき力

- ・ **説得力**
- ・ **個人仕事からの脱却**
- ・ **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

ミドルエンジニアの概要

ミドルエンジニアの導入期

レベル1 [導入期]

需要に応え、自律的に「職能を強化」し続ける

ローミドル

必要条件の充足へ

需要に応えるために必要な事項や将来像を明確にし、上司や周囲を説得できることが求められる

「言われたことをやる」(インプット指向)から「自分で考えたことを推し進める」(アウトプット指向)へ

フェイズ1: 「需要」を知る

アウトプット指向への転換

フェイズ2: 「需要」に最適化していく

説得力の確立

フェイズ3: 「需要」を創出していく

プロとして
「やりたい事」ができるようにする

仕事の「質」を変える: 「需要」に応じていく

ミドルエンジニアの概要

ミドルエンジニアの確立期

レベル2 [確立期]

「個人の仕事」から「職能の仕事」へ切り替える

真のミドル

十分条件の充足へ

エンジニアとしての成果を、個人名ではなく「職能名」(チーム名)で出すことが求められる

「独りでやれることをやる」(個人指向)から「チームとしてやれることやる」(チーム指向)へ

フェイズ1: 職能としての成果を出す

チーム指向への転換

フェイズ2: ミドルの仕事に集中する

個人仕事からの脱却

フェイズ3: 職能全体を育てる

組織として
「やりたい事」ができるようにする

仕事の「量」を変える: 「他人の手」を使う

ミドルエンジニアの概要

ミドルエンジニアの内面変化

自ら「エンジニアリングのプロフェッショナル」と認知するようになる

良い意味での「矜持」が生まれ、適切にバランスを取りながら、プロとして大事なところにはこだわるようになる

「他の人にできない・やらない何か」を探す、見付けるようになる

「自分ならではの」の得意・特異な専門領域を探し続け、環境変化に合わせてその領域を変更・更新し続ける

「エンジニアリング」の影響・結果を、強く意識するようになる

自身の「エンジニアリング」による短期的・中長期的な影響(良い影響、悪い影響)を意識し、常に改良していくようになる

自分の「得意」を伸ばす、活かすために行動するようになる

「自分ならではの」の得意・特異な専門領域を、ビジネス価値に合わせて自律的に伸ばし、活用していくようになる

ミドルエンジニアの概要

初級エンジニア と ミドルエンジニア の違い

「育成」から「**実戦**」へ

「上司や先輩から指導されながら働く」から「**主戦力として自律的に働く**」へ

「インプット指向」から「**アウトプット指向**」へ

「言われたことをやる」から「**自分で考えたことを推し進める**」へ

「個人指向」から「**職能(チーム)指向**」へ

「独りでやれることをやる」から「**職能(チーム)としてやれることやる**」へ

ミドルエンジニアの概要

ミドルエンジニア 3つのレベル

レベル1 [導入期]

需要に応え、自律的に「職能を強化」し続ける

需要に応じて、自律的に周囲を説得して、職能の強化を主導できる

ローミドル

必要条件の充足へ

説得力の確立

レベル2 [確立期]

「個人の仕事」から「職能の仕事」へ切り替える

チームの主力として、チーム成果を主導するとミドルエンジニアとして及第点

真のミドル

十分条件の充足へ

個人仕事からの脱却

レベル3 [成熟期]

シニア以降の伸び代を準備しておく

他のエンジニアとの差別化を図るための準備をする

ハイミドル

オプション
(余裕があれば)

Step2. レベル1. 導入期

はじめに

学習ガイド

本編

まとめ

ステップアップ

ナビゲーション

Step1

ミドルエンジニアの概要

Step2

レベル1. 導入期

- フェイズ1 「需要」を知ろう
- フェイズ2 「需要」に最適化していく
- フェイズ3 「需要」を創出していく

Step3

レベル2. 確立期

Step2. レベル1. 導入期

ミドルエンジニアの導入期

レベル1 [導入期]

需要に応え、自律的に「職能を強化」し続ける

ローミドル

必要条件の充足へ

需要に応えるために必要な事項や将来像を明確にし、上司や周囲を説得できることが求められる

「言われたことをやる」(インプット指向) -> 「自分で考えたことを推し進める」(アウトプット指向)

フェイズ1: 「需要」を知る

アウトプット指向への転換

フェイズ2: 「需要」に最適化していく

説得力の確立

フェイズ3: 「需要」を創出していく

プロとして
「やりたい事」ができるようにする

仕事の「質」を変える: 「需要」に応じていく

Step2. レベル1. 導入期

以前の講義で、「ミドルエンジニアは、現場の主戦力である」と説明しましたが、
現場においてミドルエンジニアに求められる役割に、
どのようなものがあると考えますか？

大きなものを3つ考えてみてください。

需要に応え、自律的に「職能を強化」し続ける

Step2. レベル1. 導入期

フェイズ1 「需要」を知るう

ナビゲーション

Step1

ミドルエンジニアの概要

Step2

レベル1. 導入期

フェイズ1 「需要」を知ろう

フェイズ2 「需要」に最適化していく

フェイズ3 「需要」を創出していく

Step3

レベル2. 確立期

フェイズ1 「需要」を知ろう

「需要って何?」という人



エンジニア仕事で
「需要」って考えたことがない

「需要」ってどこから知るの?

「需要」を成果にどう繋げるか
わからない

仕事の「質」を変える: 「需要」に応じていく

フェイズ1 「需要」を知ろう

エンジニア仕事で「需要」って考えたことがない

仕事の「質」を変える:
「需要」に応じていく

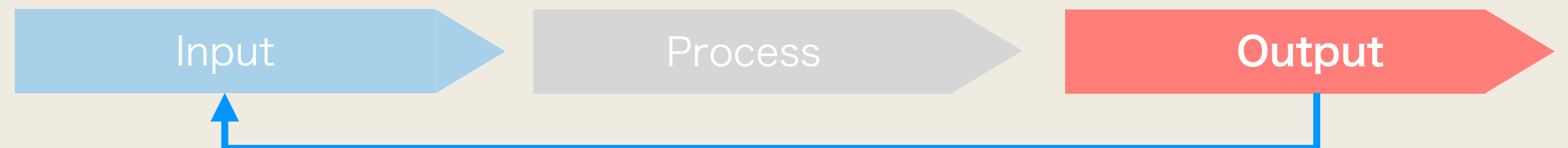
「需要」とは

市場に現われる商品への、**対価を支払うことを前提とする欲望、**
またはその**総量。**

(出典: 精選版 日本国語大辞典)

ミドルからは、**アウトプットありき**でインプット

ミドルエンジニア



アウトプット指向
への転換

Step2.
需要に応じてインプットする

Step1.
需要を掌握する

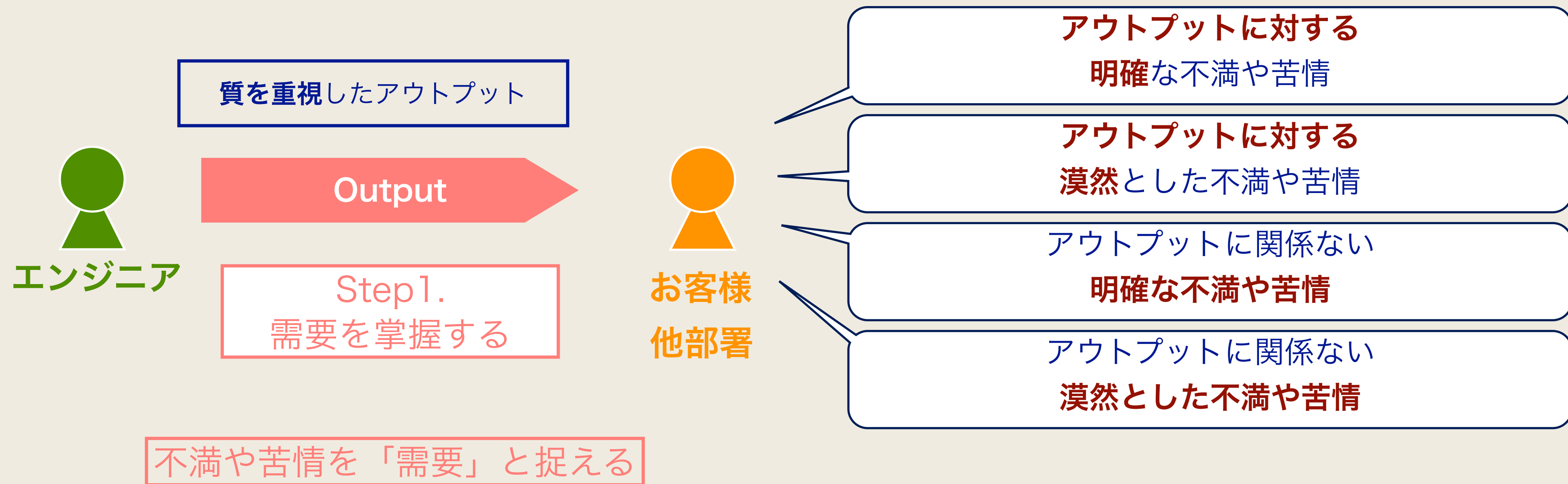
ビジネス価値=需要に応えること

フェイズ1 「需要」を知ろう

「需要」ってどこから知るの？

仕事の「質」を変える:
「需要」に応えていく

一番わかりやすい「需要」は、ユーザーからの不満や苦情



アウトプットに対する不満・苦情を「職能の伸び代」とポジティブに捉える

フェイズ1 「需要」を知ろう

「需要」を成果にどう繋げるかわからない

仕事の「質」を変える:
「需要」に応じていく

まず、ユーザーからの不満や苦情を緩和・解消できないか考えてみる



「どんな不満や苦情があるか」
を把握することに取り組む

「不満・苦情」を言語化することで、
自分なら最短でどう成果に繋がるか考え易くする

フェイズ1 「需要」を知ろう

ミドルエンジニア最初のステップ

仕事の「質」を変える:
「需要」に応じていく

まず、ユーザーからの不満や苦情に敏感になる

不満や苦情を「需要」と捉える

アウトプットに対する不満・苦情を「職能の伸び代」とポジティブに捉える

積極的に不満や苦情を集める

「どんな不満や苦情があるか」を把握することに取り組む

不満や苦情をどう成果に繋げるか

「不満・苦情」を言語化することで、自分なら最短でどう成果に繋がるか考え易くする

フェイズ1 「需要」を知ろう

ミドルエンジニアは、不満や苦情を「需要」と捉える

仕事の「質」を変える:
「需要」に応じていく

アウトプットに対する不満や苦情から学ぶ

不満や苦情を「需要」と捉える

アウトプットに対する不満・苦情を「職能の伸び代」とポジティブに捉える

ユーザーからの不満や苦情をストレスに感じないようにする

ユーザーからの不満や苦情を受けたときに、詳しく聞いておく

ユーザーからの不満や苦情を受けたときは、蓄積しておく

フェイズ1 「需要」を知ろう

ミドルエンジニアは、積極的に不満や苦情を集める

仕事の「質」を変える:
「需要」に応じていく

不満や苦情(クレーム)に関する知見のインプットを行う

積極的に不満や苦情を集める

「どんな不満や苦情があるか」を把握することに取り組む

不満やクレームに関する知見を書籍などから学んでおく

こちらからユーザーの不満や苦情を探しに行く

ユーザーの不満や苦情を共通点を探し、学びを深めていく

フェイズ1 「需要」を知ろう

ミドルエンジニアは、不満や苦情をどう成果に繋げるか考える

仕事の「質」を変える:
「需要」に応じていく

不満や苦情(クレーム)に関する知見のインプットを行う

不満や苦情をどう成果に繋げるか

「不満・苦情」を言語化することで、自分なら最短でどう成果に繋がるか考え易くする

不満や苦情を細分化・構造化していく

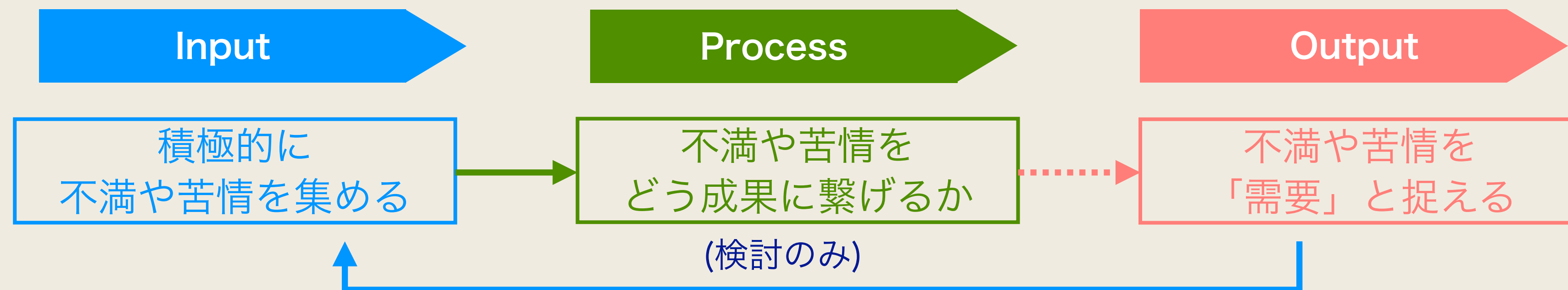
解消の余地が有る不満や苦情への対応策を考えてみる

解消の余地が無さそうな不満や苦情への対応策を考えてみる

フェイズ1 「需要」を知ろう

仕事の「質」を変える:
「需要」に応じていく

一番わかりやすい「需要」は、ユーザーからの不満や苦情



「どんな不満や苦情があるか」
を知る

「不満・苦情」を
言語化する

不満・苦情を
ポジティブに捉える

まず、ユーザーからの不満や苦情を緩和・解消できないか考えてみる

Step2. レベル1. 導入期

フェイズ2 「需要」に最適化していく

ナビゲーション

Step1

ミドルエンジニアの概要

Step2

レベル1. 導入期

フェイズ1 「需要」を知ろう

フェイズ2 「需要」に最適化していく

フェイズ3 「需要」を創出していく

Step3

レベル2. 確立期

フェイズ2 「需要」 に最適化していく

「最適化って何?」という人



「最適化」ってどんなこと?

どうすれば「最適化」になるのか
わからない

どう「最適化」していくのか
わからない

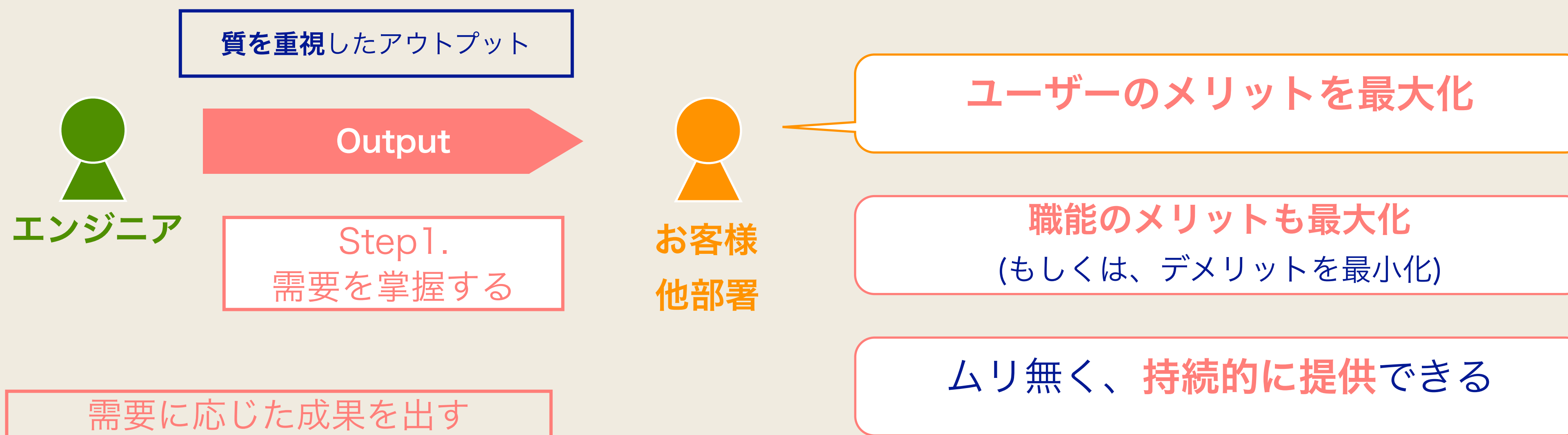
仕事の「質」を変える: 「**需要**」に依えていく

フェイズ2 「需要」に最適化していく

「最適化」ってどんなこと?

仕事の「質」を変える:
「需要」に応じていく

職能(提供側)とユーザー(受益側)の双方にとっての全体最適



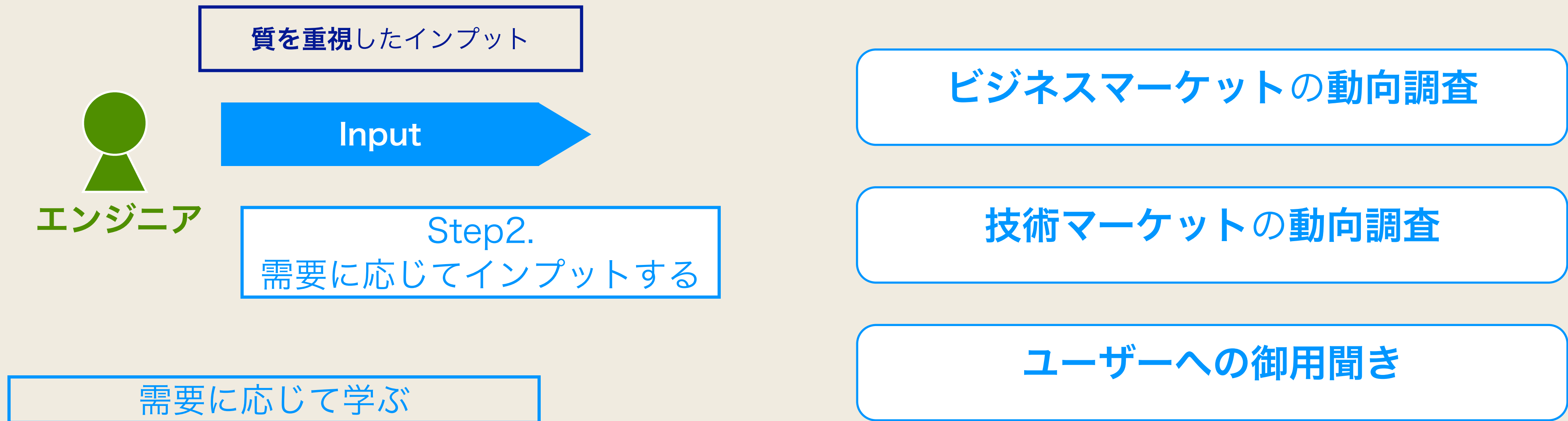
上司を説得して需要に最適なアウトプットをしていく

フェイズ2 「需要」に最適化していく

どうすれば「最適化」になるのかわからない

仕事の「質」を変える:
「需要」に応じていく

マーケットから概要情報を、ユーザーから詳細情報を得る



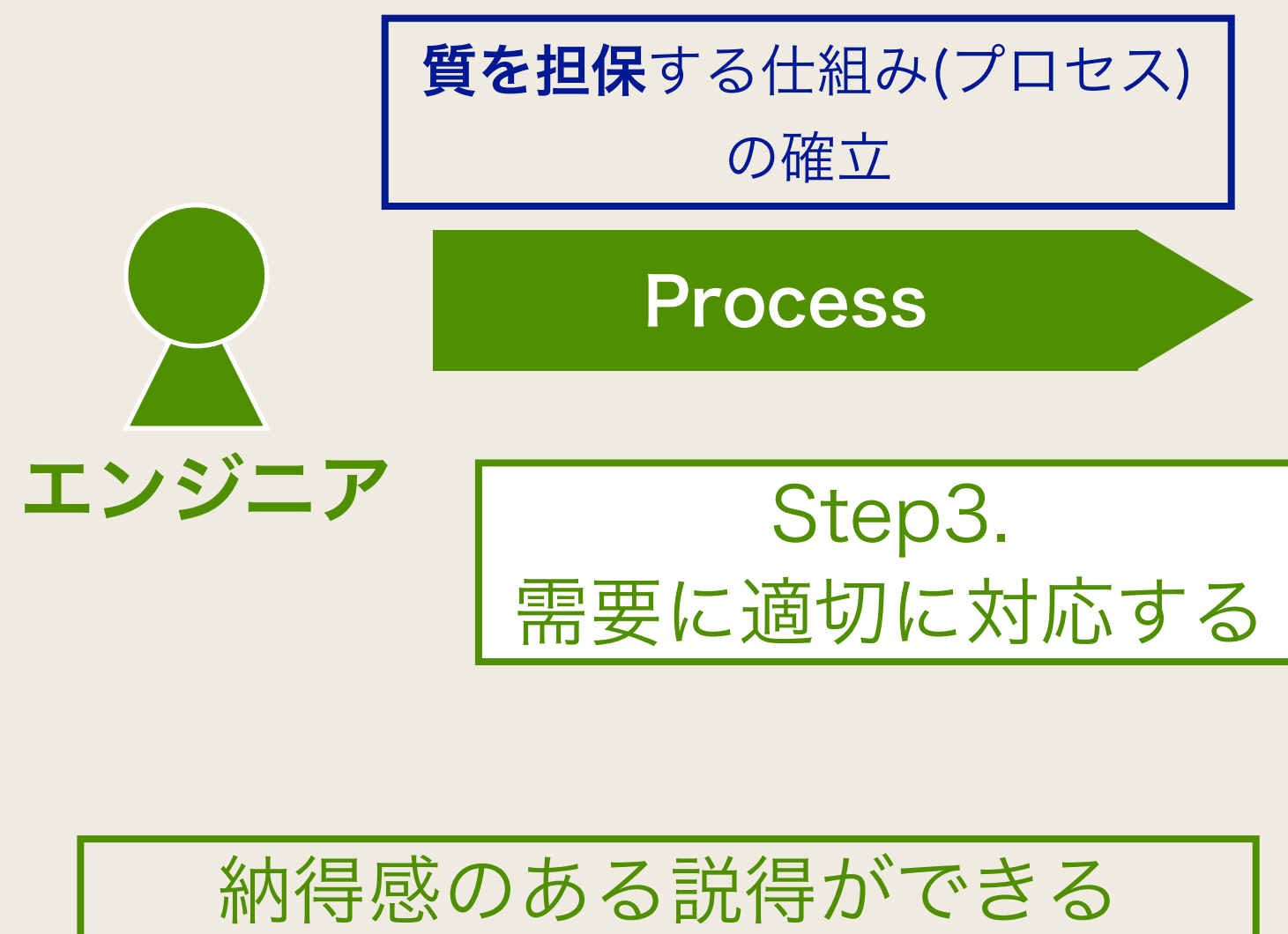
上司を説得するための材料として、**需要ベースのインプット**をしていく

フェイズ2 「需要」に最適化していく

どう「最適化」していくのかわからない

仕事の「質」を変える:
「需要」に応じていく

エンジニアらしく「最適化」を探求し、実現していく



客観的、論理的、合理的、持続的

ToBeとAsIsの明確化

ヌケやモレが無いようにする

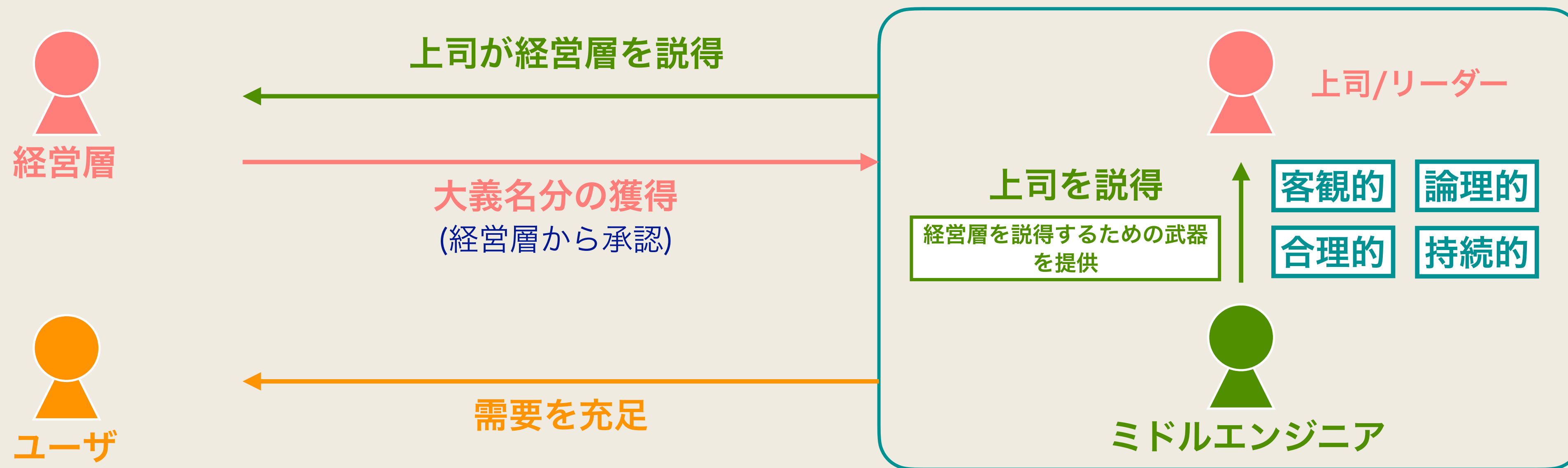
上司が不安にならないように、ヌケ・モレ無く論理的に説得できるようにする

フェイズ2 「需要」に最適化していく

最適化には説得力が必要

仕事の「質」を変える:
「需要」に応じていく

エンジニアの説得力 = エンジニアリングを言語化する力



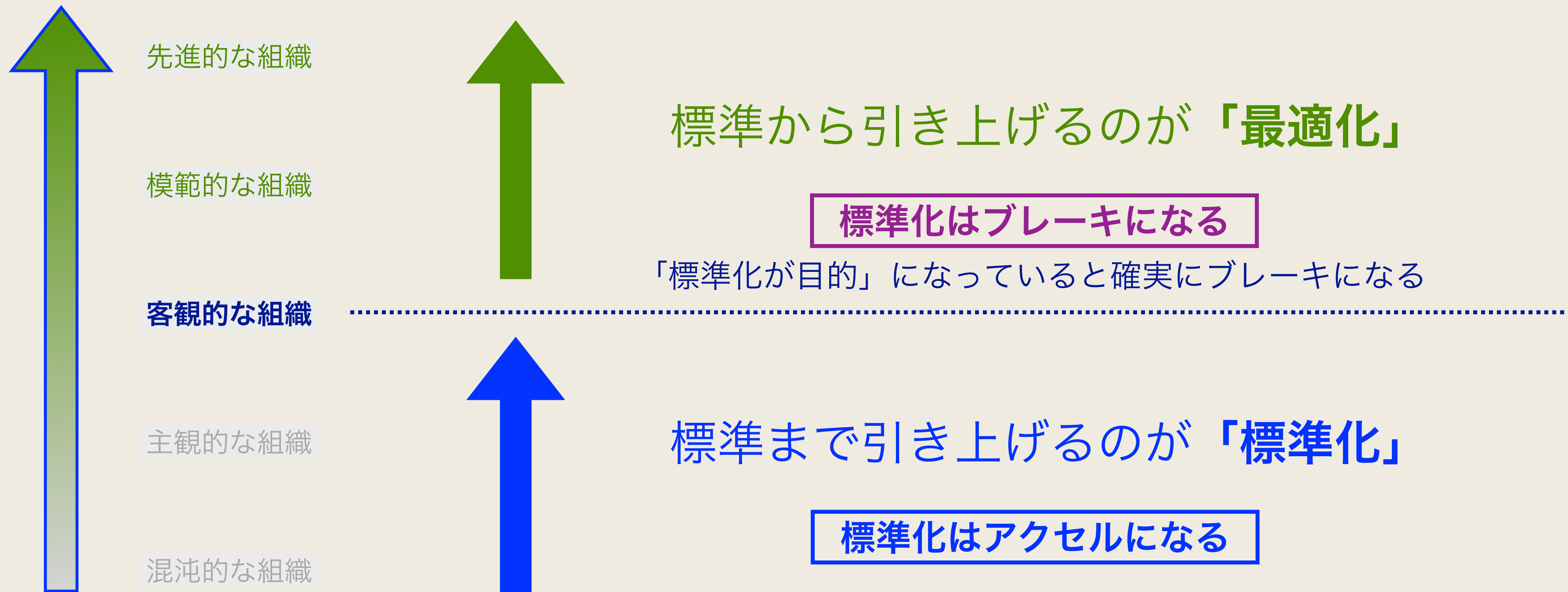
上司を説得できないのはエンジニアリング力不足

フェイズ2 「需要」に最適化していく

「標準化」と「最適化」の使い分け

仕事の「質」を変える:
「需要」に応じていく

「標準化」には先進的・模範的な組織を退化させるデメリットがある



フェイズ2 「需要」に最適化していく

ミドルとして最初の役割はアウトプットの「需要への最適化」

仕事の「質」を変える:
「需要」に応じていく

「アウトプットが需要に最適化」されるように業務を作り替え続けよう

需要に応じた成果を出す

上司を説得して**需要に最適なアウトプット**をしていく

需要に応じて学ぶ

上司を説得するための材料として、**需要ベースのインプット**をしていく

納得感のある説得ができる

上司が不安にならないように、**ヌケ・モレ無く論理的に説得**できるようにする

フェイズ2 「需要」に最適化していく

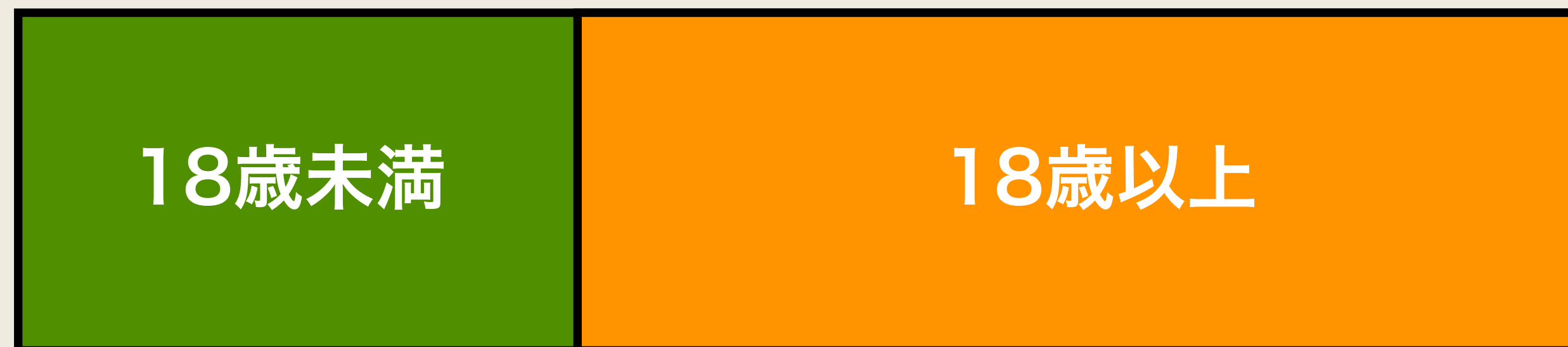
説得力をマシマシする「MECE」

仕事の「質」を変える:
「需要」に応じていく

MECE (Mutually Exclusive, Collectively Exhaustive)
ダブりがなく モレもない

ロジカルシンキングの
基本的な概念

相互に排他的、共同で網羅的であることを表現する思考法



0歳

例: 全体集合は「現在生きている人」

最高齢

MECEで説明できていると、上司や経営層が不安にならない

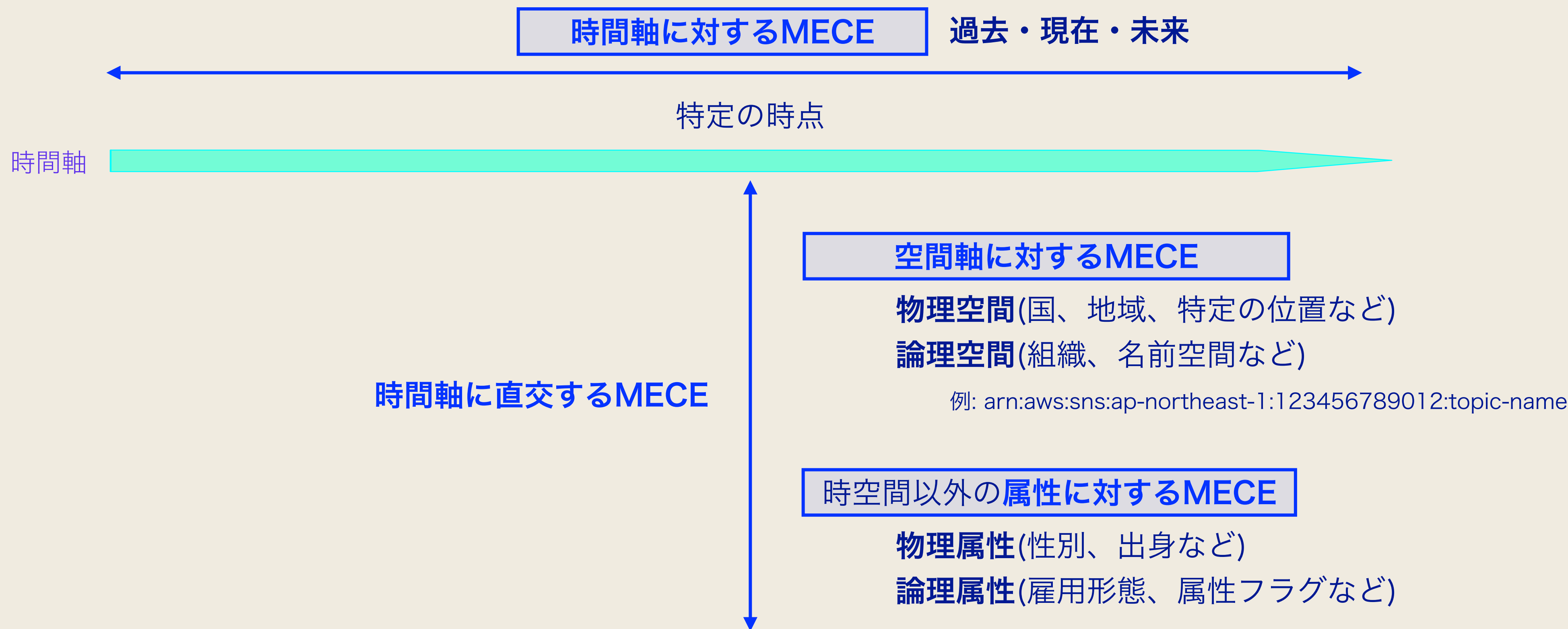
MECEで設計・実装できていると、設計や実装に漏れがない

フェイズ2 「需要」 に最適化していく

運用設計における3種類のMECE

仕事の「質」を変える:
「需要」に応じていく

運用設計では、**時間軸**、**空間軸**、**それ以外(属性軸)**の3軸でMECEを使うことが多い



フェイズ2 「需要」に最適化していく

時間軸における「MECE」の例

仕事の「質」を変える:
「需要」に応じていく

2値

Before/Afterモデル

良い悪いという意味は弱い

ToBe/AsIsモデル

ToBeの方がAsIsよりも良い

一次元
(3値以上)

Input/Process/Outputモデル

組織内部におけるプロセスモデル

Inbound/Process/Outboundモデル

組織外部に対するプロセスモデル

Plan/Do/Seeモデル

講師はよく使います

Plan/Do/Check/Action

上手く回っていることは稀?

Create/Read/Update/Deleteモデル

Web API界隈から有名に

フェイズ2 「需要」 に最適化していく

時間軸(2値): Before/Afterモデル

仕事の「質」を変える:
「需要」に応じていく

特定の時点の**以前(Before)**と**以後(After)**の**2値**によるMECEモデル

特定の時点を境にした変化を表現するものであり、BeforeとAfterのどちらかが良い悪いという意味は弱い。



運用現場での
現実

改善後(After)だけ示して、**改善前(Before)**を示せないことがほとんど。

BeforeとAfterの差分が示せないと、客観的な改善効果は誰にもわからない。

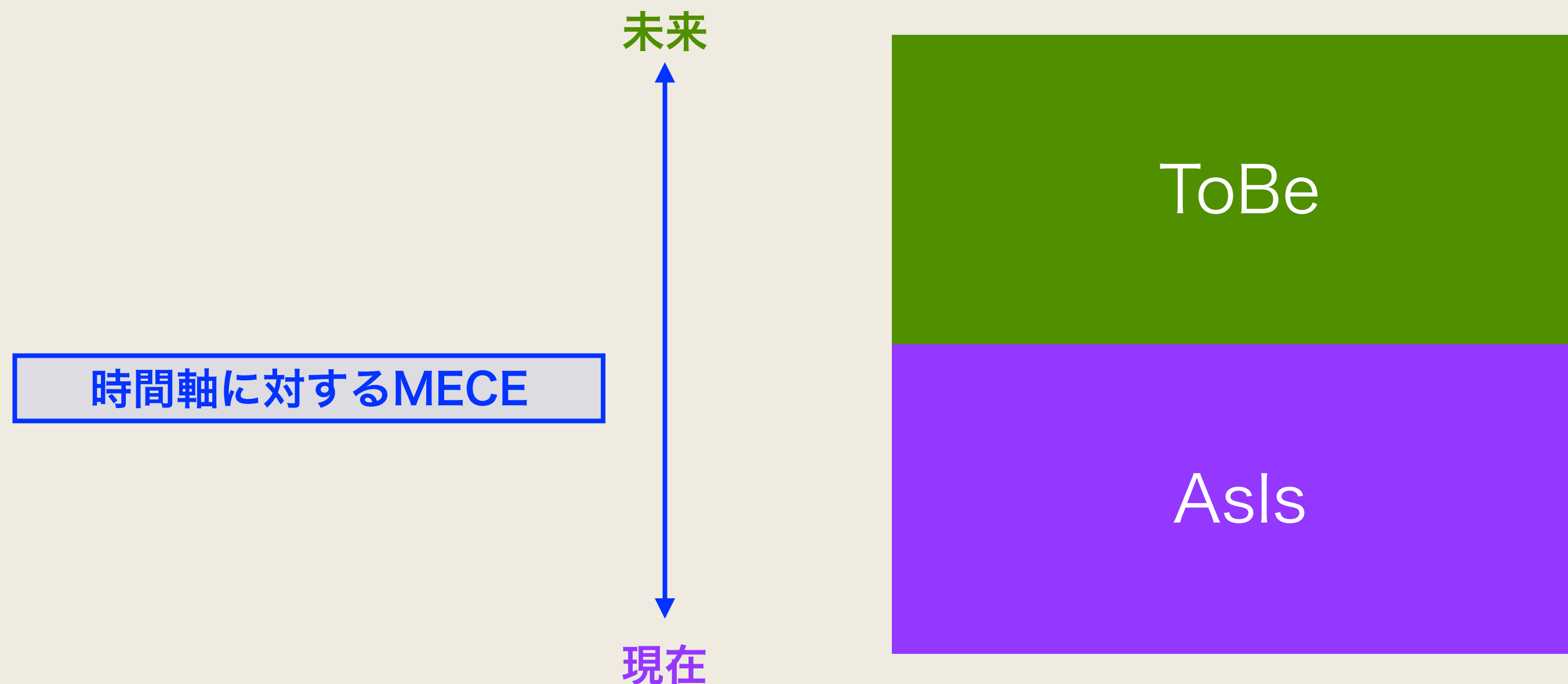
フェイズ2 「需要」 に最適化していく

時間軸(2値): ToBe/AsIsモデル

仕事の「質」を変える:
「需要」に応じていく

特定の時点の現在の姿(AsIs)とあるべき姿(ToBe)の2値によるMECEモデル

現在と理想を比較するものであり、AsIsよりもToBeの方が良いという意味が強い。



運用現場での
現実

現状(AsIs)を把握できているだけでも稀で理想(ToBe)を示せることはほぼ皆無。

AsIsで現状を把握し、ToBeでゴール像を示せないと、適切な運用設計も運用改善も不可能。

フェイズ2 「需要」 に最適化していく

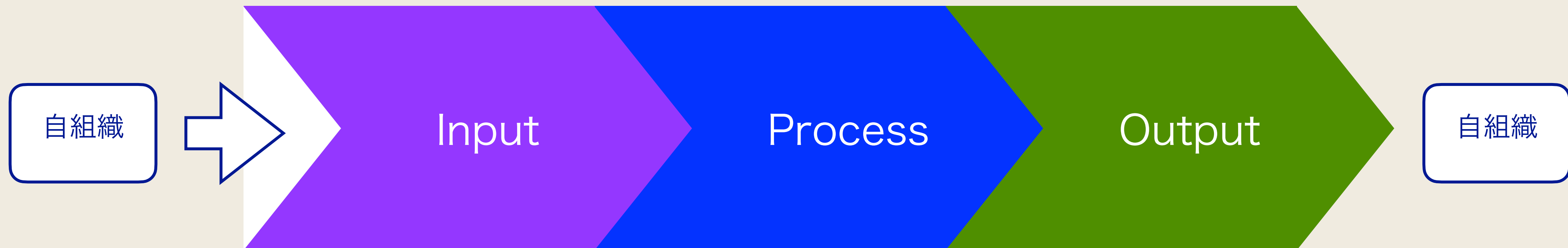
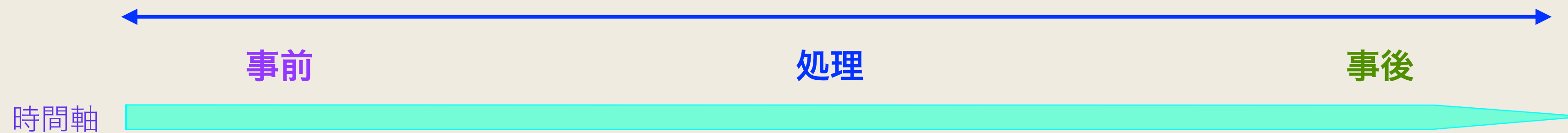
時間軸(一次元): Input/Process/Outputモデル

仕事の「質」を変える:
「需要」に応じていく

内部プロセスの処理に関する一次元によるMECEモデル

組織内部におけるプロセスモデル。外部組織への影響を意識する必要が無い。

時間軸に対するMECE



運用現場での
現実

Processだけ手順化・実装して、InputとOutputが曖昧であることが多い。

InputとOutputへの要求仕様に従ってProcessを設計しないと、正しいプロセスを実現できない。

フェイズ2 「需要」に最適化していく

時間軸(一次元): Create/Read/Update/Deleteモデル

仕事の「質」を変える:
「需要」に応じていく

プロセスのライフサイクルに関する一次元によるMECEモデル

ライフサイクルにおける作業や自動化のリスクと、自動化の寿命を示すモデル



フェイズ2 「需要」に最適化していく

補助的に使う「MECE」の例

仕事の「質」を変える:
「需要」に応じていく

時空以外の軸
(補助的)

Stock/Flowモデル

動態(蓄積型か遷移型)によるMECEモデル

Policy/Entityモデル

主客(ポリシーか実体)によるMECEモデル

Plan/Actual Resultモデル

予実(計画か実績)によるMECEモデル

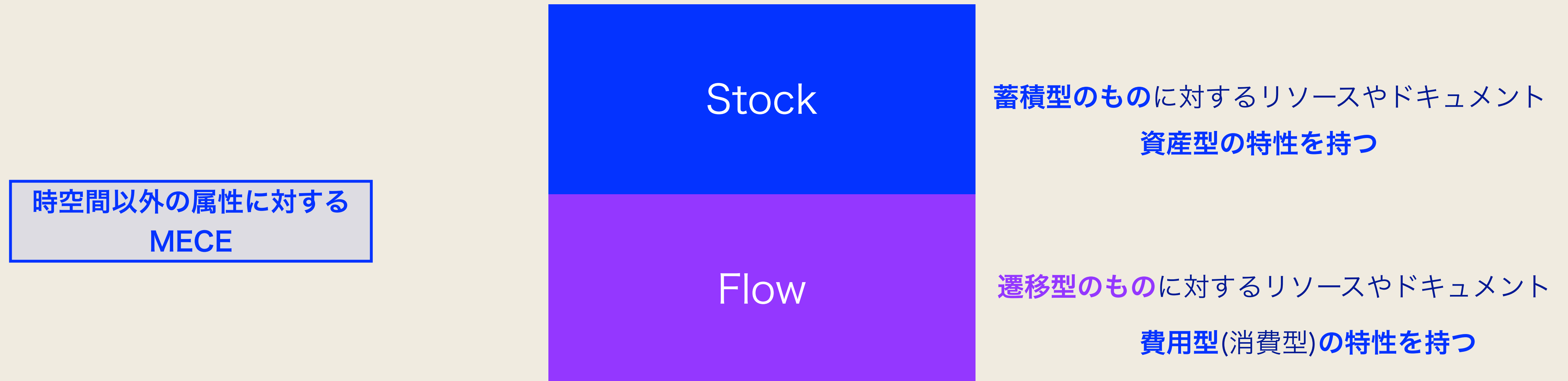
フェイズ2 「需要」に最適化していく

動態軸(2値): Stock/Flowモデル

仕事の「質」を変える:
「需要」に応じていく

動態(蓄積型か遷移型)の2値によるMECEモデル

運用設計上で、補助的に利用するMECEモデル



蓄積型に工数を掛けて、遷移型に工数を掛けないと、保守し続けられる事が多い。

フェイズ2 「需要」に最適化していく

説得&承認が得られたら「職能を強化」する

仕事の「質」を変える:
「需要」に応じていく

承認 = 大義名分がある状態で進めることができる

どう「職能」を強化していくか

基本に忠実に (情報科学、基本書、公式ドキュメント)

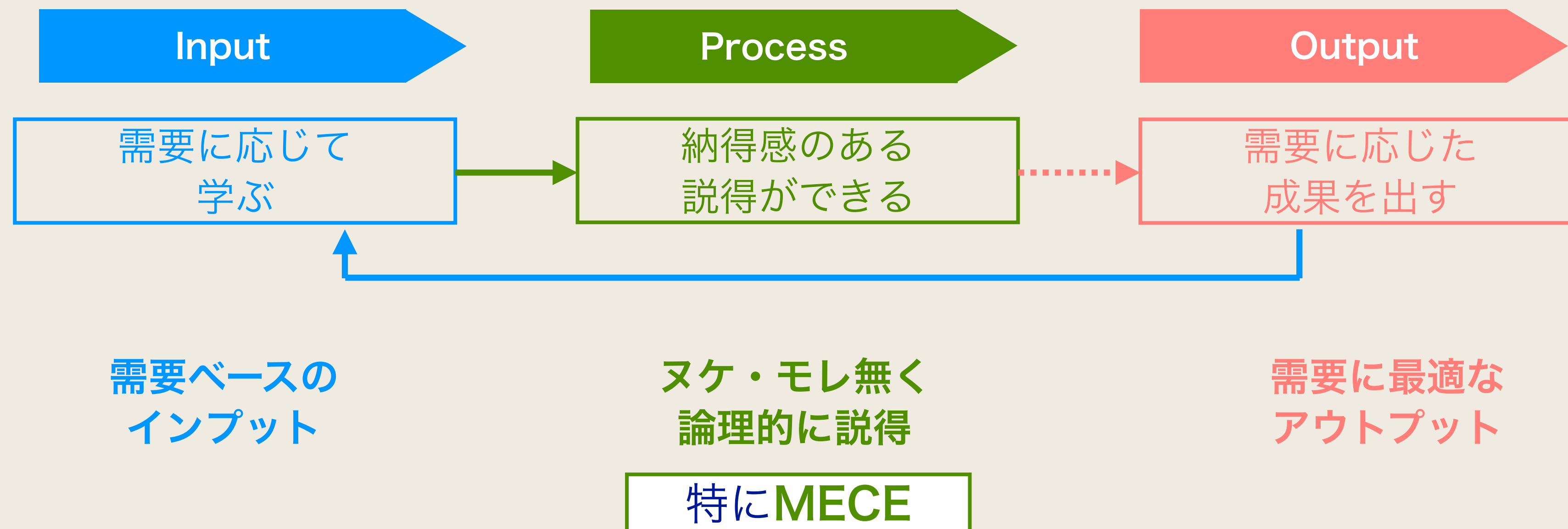
エンジニアリングでも急がば回れ

ビジネスの基礎も必要になってくる (経営学、簿記、法律、デザインなど)

フェイズ2 「需要」に最適化していく

仕事の「質」を変える:
「需要」に応じていく

ミドルエンジニアが中心となって需要に対応していく



「アウトプットが需要に最適化」されるように業務を作り替え続ける

Step2. レベル1. 導入期

フェイズ3 「需要」を創出していく

ナビゲーション

Step1

ミドルエンジニアの概要

Step2

レベル1. 導入期

フェイズ1 「需要」を知ろう
フェイズ2 「需要」に最適化していく
フェイズ3 「需要」を創出していく

Step3

レベル2. 確立期

フェイズ3 「需要」を創出していく

「需要の創出って何?」という人



「需要の創出」ってどんなこと?

「需要を創出」して何が嬉しいのか
わからない

どうすれば「需要を創出」できるのか
わからない

仕事の「質」を変える: 「**需要**」に応じていく

フェイズ3 「需要」を創出していく

「需要の創出」ってどんなこと？

仕事の「質」を変える:
「需要」に応じていく

「需要」とは

市場に現われる商品への、**対価を支払うことを前提とする欲望、**
またはその**総量。**

(出典: 精選版 日本国語大辞典)

「創出」とは

はじめてつくること。**新しく作り出すこと。**

(出典: 精選版 日本国語大辞典)



ユーザーが**対価を払いたくなるもの**を、**新しく創り出すこと**

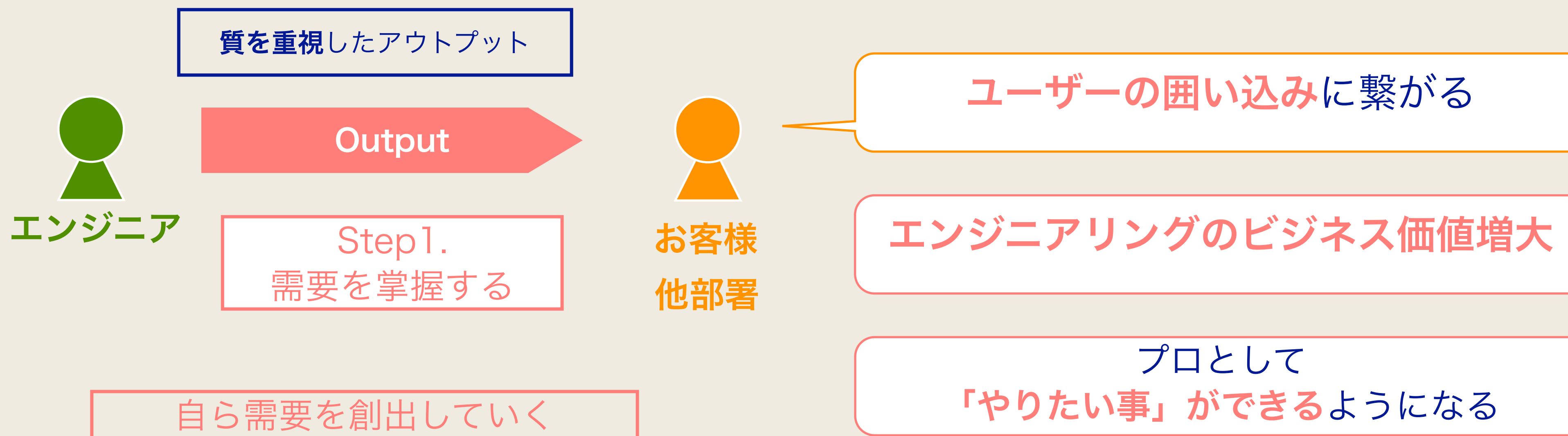
「受け身の運用」から「**攻めの運用**」への転換

フェイズ3 「需要」を創出していく

「需要を創出」して何が嬉しいのかわからない

仕事の「質」を変える:
「需要」に応じていく

経営層もユーザーも喜び、エンジニアも好きなことができるようになる



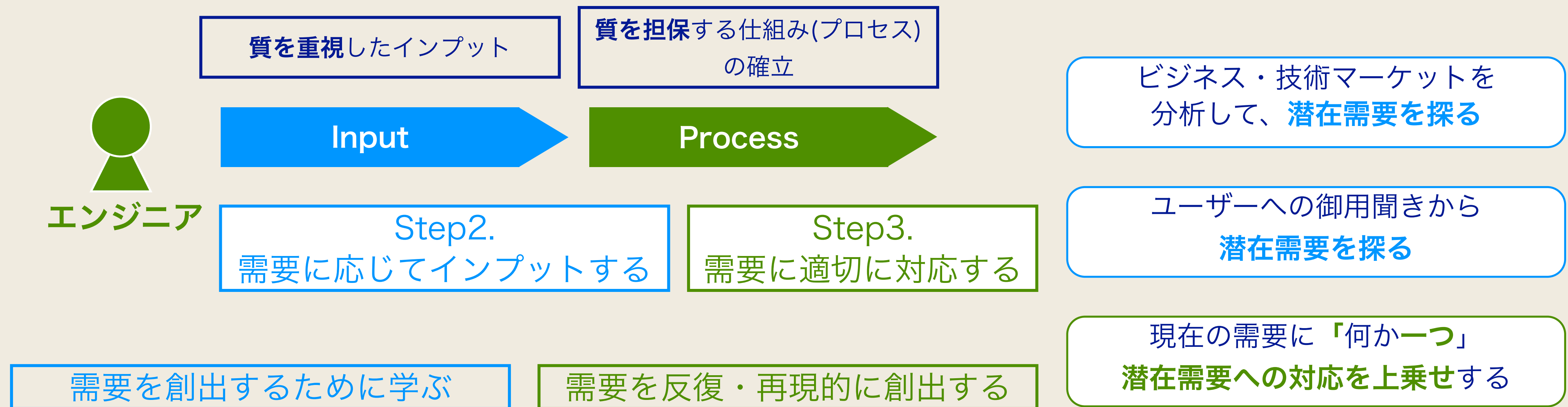
利用者が気付いていない**潜在的な需要**を掘り起こせるようになる

フェイズ3 「需要」を創出していく

どうすれば「需要を創出」できるのかわからない

仕事の「質」を変える:
「需要」に応じていく

潜在需要をマーケットとユーザーの双方から探り出し、何か一つ実現してみる



新たな需要を創出するための
インプットをする

「需要」への付加価値として
「潜在需要への対応」を上乗せし続ける

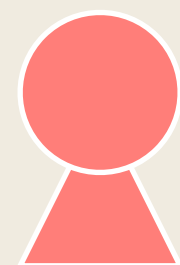
フェイズ3 「需要」を創出していく

「需要の創出」が「やりたいことができる」に繋がる

仕事の「質」を変える:
「需要」に応じていく

ビジネスモデルの

設計者



経営層

後追いで要求



機能

潜在需要から「やりたい事」
を見つける

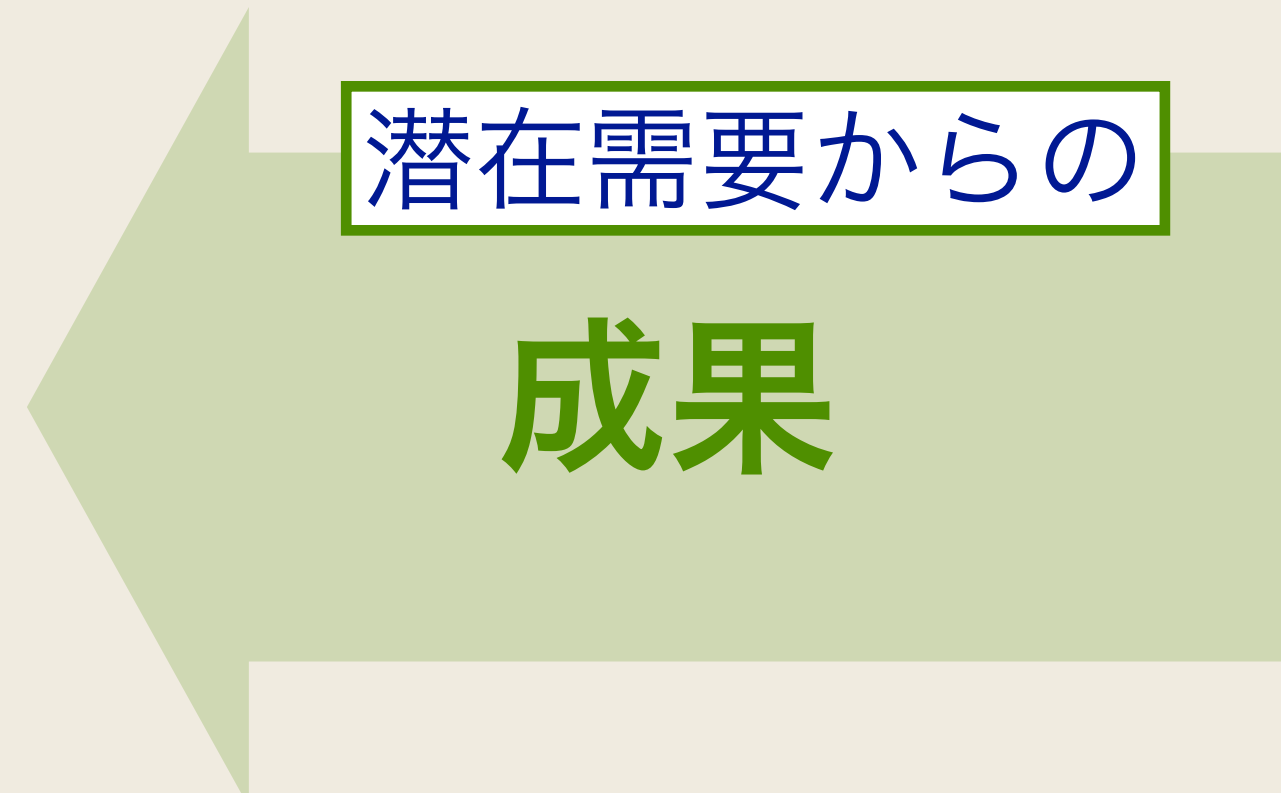


エンジニア

ビジネスモデルの
実践者

潜在需要からの

成果



潜在需要が強ければ

「やりたいこと」の優先度と成功率が
高くなる



ユーザ

ビジネスモデルの

受益者

フェイズ3 「需要」を創出していく

ミドルとして「やりたいこと」をユーザーに創出していく

仕事の「質」を変える:
「需要」に応えていく

ユーザーを誘引して「自分がやりたいこと」ができるようにしていこう

自ら需要を創出していく

ユーザーが気付いていない**潜在的な需要**を掘り起こせるようになる

需要を創出するために学ぶ

新たな需要を創出するためのインプットをする

需要を反復・再現的に創出する

「需要」への付加価値として「**潜在需要への対応**」を上乗せし続ける

フェイズ3 「需要」を創出していく

受動運用から能動運用へ

仕事の「質」を変える:
「需要」に応じていく

受動型の運用

依頼元から言われるままの運用業務は
複雑かつ多様になりやすい (フルカスタマイズ運用)

- ・ 複雑で多様な業務の増加は、高負荷状態になりやすく、それが恒常化しやすい。
- ・ 複雑で多様な業務の増加は、教育コストが増大し属人化しやすくなる。
- ・ 複雑な業務は、その費用対効果を説明することが難しい。
- ・ 自分達の専門性が曖昧になりやすく育成やキャリア設計が難しい。

成果を言語化できない

「運用はいつも忙しそうだが
何をしているのかわからない」
と言われる



需要をテコに
変えていく

能動型の運用

自分達の専門性を基に運用業務を決めるため
シンプルに設計しやすい。

- ・ シンプルな業務を組み合わせることで、多様な顧客の要求に応えることができる。(UNIXの考え方)
- ・ シンプルな業務は、属人化しにくく、負荷上昇への対応もしやすい。
- ・ シンプルな業務は、費用対効果の説明が容易である。
- ・ 専門性に基いた業務であるため、教育やキャリア設計が容易になる。

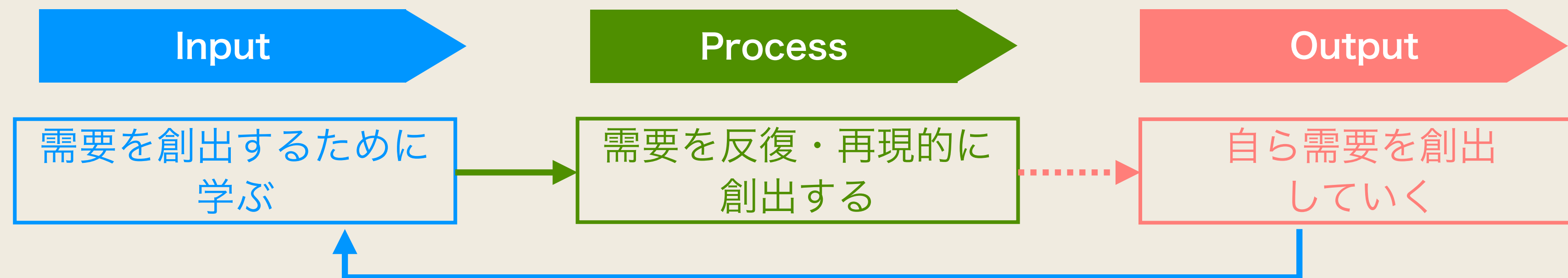
成果を言語化できる

「何をしてもらえるのか」
「こう頼めばこんな成果が提供される」
経営層や他組織の期待・評価を明確にできる

フェイズ3 「需要」を創出していく

仕事の「質」を変える:
「需要」に応じていく

ミドルエンジニアは「やりたいこと」ができる立場



新たな需要を創出するための
インプット

付加価値としての
「潜在需要への対応」

潜在的需要を
掘り起こす

ユーザーを誘引して「自分がやりたいこと」ができるようにしていこう

Step2. レベル1. 導入期

まとめ

はじめに

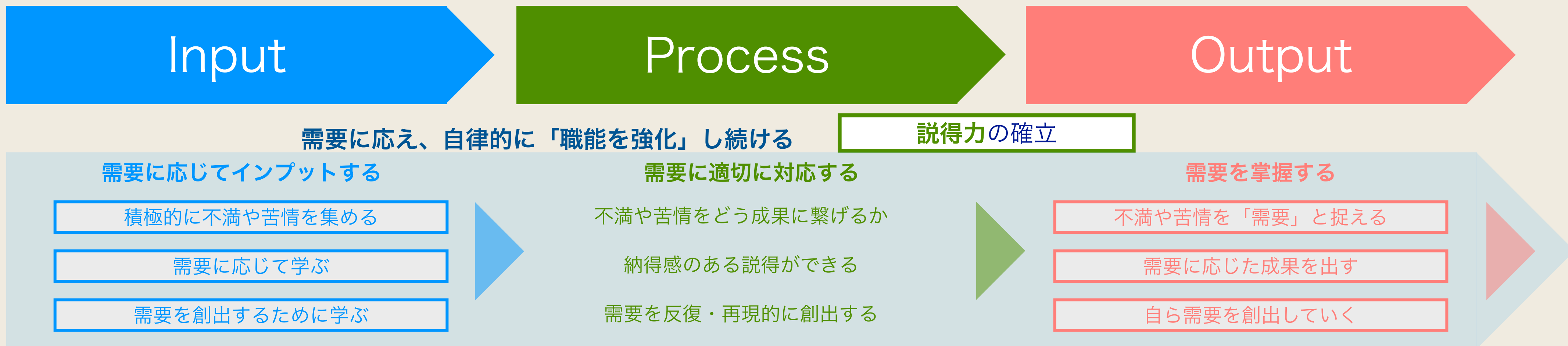
学習ガイド

本編

まとめ

ステップアップ

まとめ: レベル1. 導入期



仕事の「質」を変える: 「需要」に応じていく

課題1 解答 (例)

以前の講義で、「ミドルエンジニアは、現場の主戦力である」と説明しましたが、
現場においてミドルエンジニアに求められる役割に、
どのようなものがあると考えますか？

大きなものを3つ考えてみてください。

需要の掌握・創出

上司や経営層の説得

職能の強化

需要に応え、自律的に「職能を強化」し続ける

Step3. レベル2. 確立期

はじめに

学習ガイド

本編

まとめ

ステップアップ

ナビゲーション

Step1

ミドルエンジニアの概要

Step2

レベル1. 導入期

Step3

レベル2. 確立期

フェイズ1: 職能としての成果を出す
フェイズ2: ミドルの仕事に集中する
フェイズ3: 職能全体を育てる

Step3. レベル2. 確立期

ミドルエンジニアの確立期

レベル2 [確立期]

「個人の仕事」から「職能の仕事」へ切り替える

真のミドル

十分条件の充足へ

エンジニアとしての成果を、個人名ではなく「職能名」(チーム名)で出すことが求められる

「独りでやれることをやる」(個人指向)から「チームとしてやれることやる」(チーム指向)へ

フェイズ1: 職能としての成果を出す

チーム指向への転換

フェイズ2: ミドルの仕事に集中する

個人仕事からの脱却

フェイズ3: 職能全体を育てる

組織として
「やりたい事」ができるようにする

仕事の「量」を変える: 「他人の手」を使う

Step3. レベル2. 確立期

現場においてミドルエンジニアに求められる役割を担うために、
どのような能力やスキルが必要になると考えますか？

需要の掌握・創出

上司や経営層の説得

職能の強化

大きなものを3つ考えてみてください。

「個人の仕事」から「職能の仕事」へ切り替える

Step3. レベル2. 確立期

フェイズ1: 職能としての成果を出す

ナビゲーション

Step1

ミドルエンジニアの概要

Step2

レベル1. 導入期

Step3

レベル2. 確立期

フェイズ1: 職能としての成果を出す

フェイズ2: ミドルの仕事に集中する

フェイズ3: 職能全体を育てる

フェイズ1: 職能としての成果を出す

「職能としての成果って何?」という人



個人の成果ではなぜだめなのか
わからない

なにが「職能としての成果」になるのか
わからない

どうすれば「職能としての成果」が生まれるのか
わからない

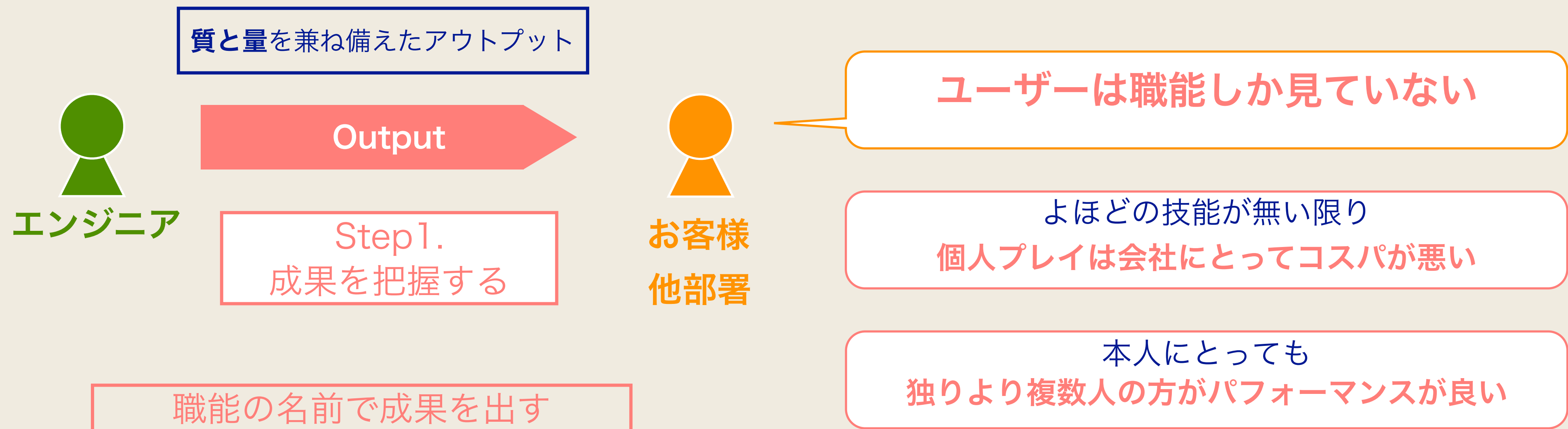
仕事の「量」を変える: 「他人の手」を使う

フェイズ1: 職能としての成果を出す

個人の成果ではなぜだめなのかわからない

仕事の「量」を変える:
「他人の手」を使う

ユーザー・会社・本人から見て「個人プレイ」の成果はコスパが悪い



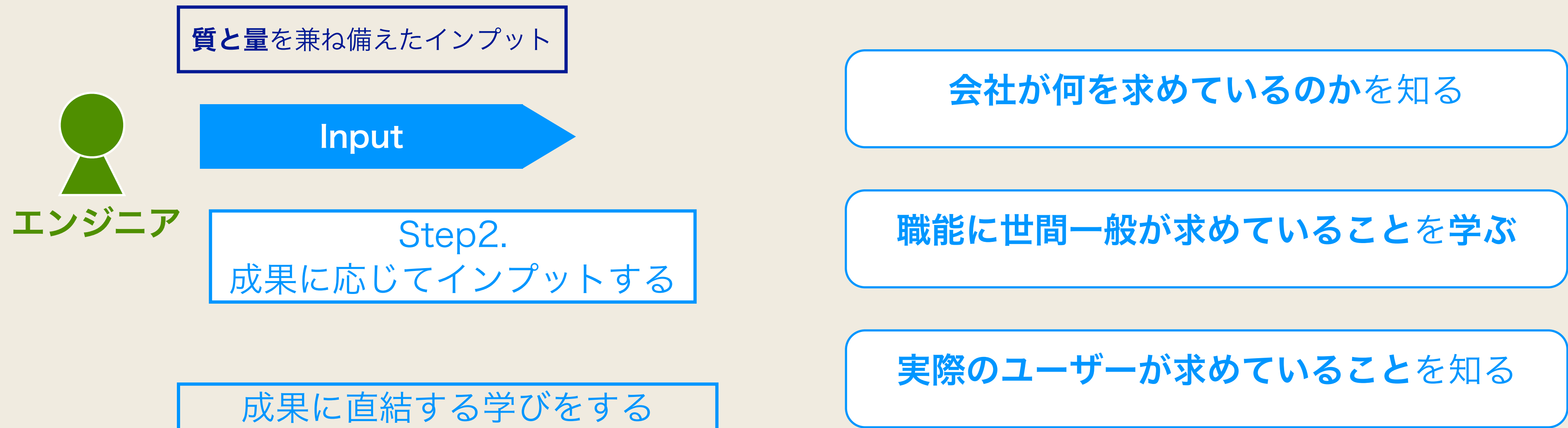
個人成果ではなく職能集団としての成果が求められる

フェイズ1: 職能としての成果を出す

なにが「職能としての成果」になるのかわからない

仕事の「量」を変える:
「他人の手」を使う

会社、世間一般、実際のユーザーの順で、どんな成果を求めているのかを調査する



職能(チーム)の「成果」責任を負うために必要なインプットをしていく

フェイズ1: 職能としての成果を出す

どうすれば「職能としての成果」が生まれるのかわからない

仕事の「量」を変える:
「他人の手」を使う

ミドルエンジニアが、職能全体の活動や将来を言語化し、主導していく



「成果に必要な構成要素」を言語化することが成果に繋がる

フェイズ1: 職能としての成果を出す

「個人の活躍」から脱却しよう

仕事の「量」を変える:
「他人の手」を使う

「個人での活躍」から「**職能の成果を導く人**」になろう

職能の名前で成果を出す

個人成果ではなく**職能集団としての成果**が求められる

成果に直結する学びをする

職能(チーム)の「**成果**」を負うために必要なインプットをしていく

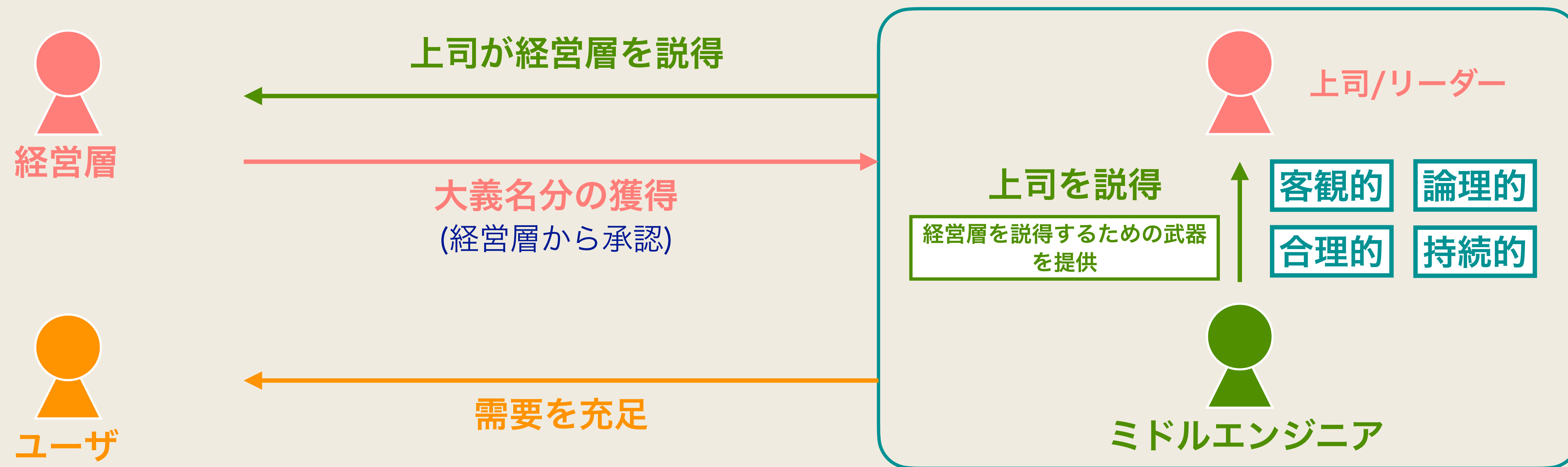
学びと職能の成果を最短で繋ぐ

「**成果の構成要素**」を言語化することが成果に繋がる

フェイズ1: 職能としての成果を出す

ミドルが「職能の実質的な意思決定」を行う

上司を「意思決定プロキシ」として、実質的には意思決定を主導する

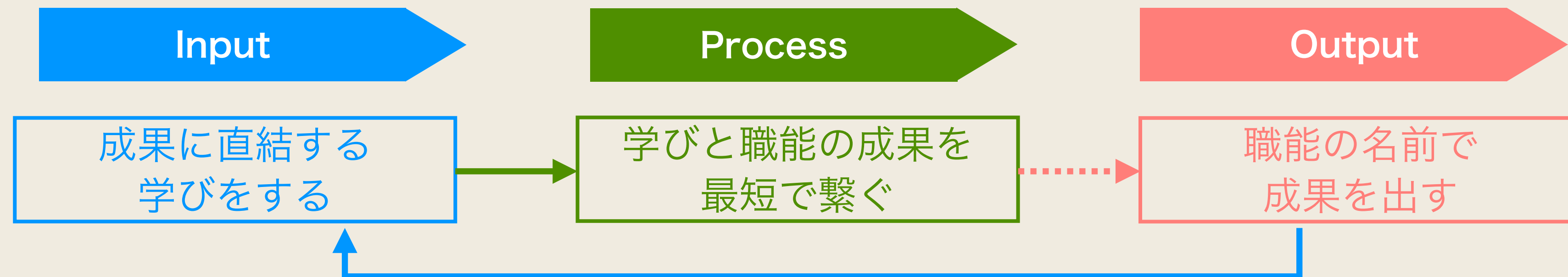


ミドルエンジニアは、専門領域について上司より詳しい

フェイズ1: 職能としての成果を出す

仕事の「量」を変える:
「他人の手」を使う

「職能で成果」を出す == エンジニアのプロ意識



職能の「成果」責任のために
必要なインプット

「成果の構成要素」
を言語化

職能集団
としての成果

「個人での活躍」から「職能の成果を導く人」になろう

Step3. レベル2. 確立期

フェイズ2: ミドルの仕事に集中する

ナビゲーション

Step1

ミドルエンジニアの概要

Step2

レベル1. 導入期

Step3

レベル2. 確立期

フェイズ1: 職能としての成果を出す
フェイズ2: ミドルの仕事に集中する
フェイズ3: 職能全体を育てる

フェイズ2: ミドルの仕事に集中する

「ミドルの仕事ってどういうもの？」という人



これまでの仕事のやり方と何が違うのか
わからない

「ミドルの仕事に集中する」
とはどういうことか？

どうすれば「ミドルとしての成果」が生まれるのか
わからない

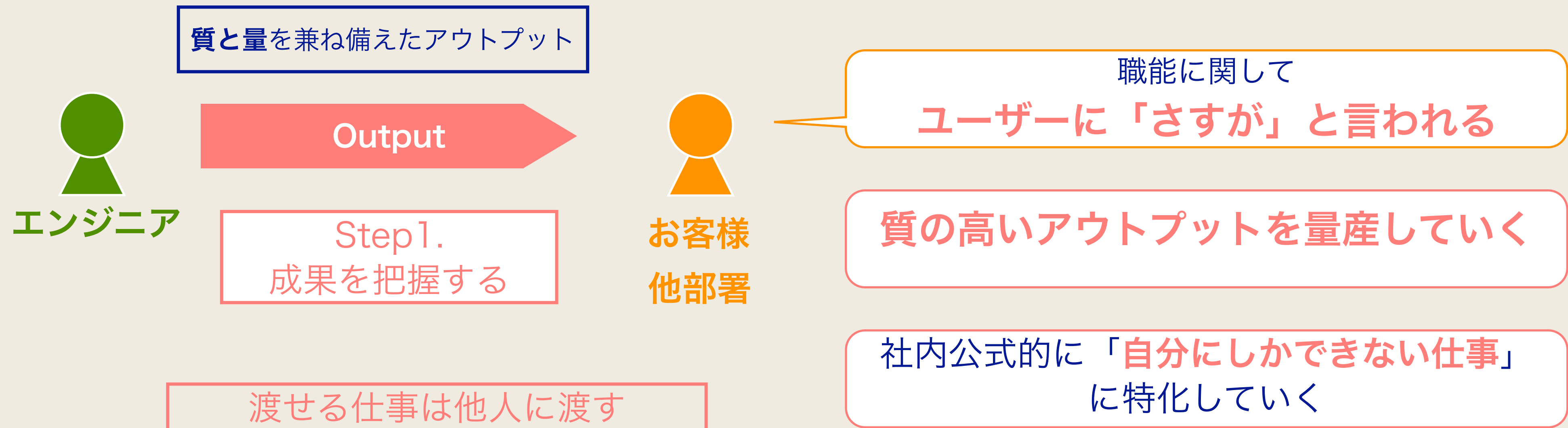
仕事の「量」を変える: 「他人の手」を使う

フェイズ2: ミドルの仕事に集中する

これまでの仕事のやり方と何が違うのかわからない

仕事の「量」を変える:
「他人の手」を使う

プロフェッショナルとして、自分にしかできない仕事を公式化して、量産していく



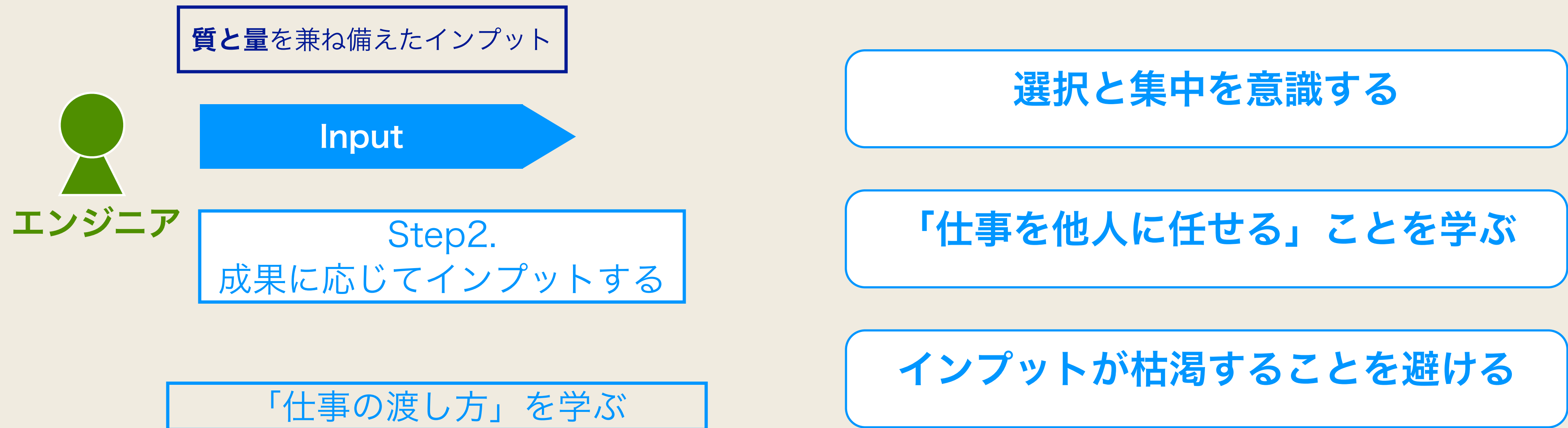
アウトプットとして「人に渡せる仕事」を作っていく

フェイズ2: ミドルの仕事に集中する

「ミドルの仕事に集中する」とはどのようなことか?

仕事の「量」を変える:
「他人の手」を使う

「自分以外でもできる仕事」を把握し、他人への渡し方を学ぶ



「他人に渡せる仕事」とはどんな仕事を学び、どのように渡せば良いかを理解する必要がある

フェイズ2: ミドルの仕事に集中する

どうすれば「ミドルとしての成果」が生まれるのかわからない

仕事の「量」を変える:
「他人の手」を使う

「自分以外でもできる仕事」を他人に渡し、ミドルにしかできない仕事をする



ミドルエンジニアの仕事は原則として「他人に渡すこと」が前提になってくる

フェイズ2: ミドルの仕事に集中する

ミドルにしかできない仕事

仕事の「量」を変える:
「**他人の手**」を使う



ミドルエンジニア

ミドルエンジニアとしての
専門領域での仕事

客観的

論理的

合理的

持続的

ミドルエンジニアの専門領域から、
初級エンジニアの仕事への切り出し

初級エンジニアの
アウトプットの価値を増大



初級エンジニア

初級エンジニアの仕事の基本設計と再設計
(改善ではない)

初級エンジニアの
負担を軽減

後輩に仕事を渡せないのはエンジニアリング力不足

フェイズ2: ミドルの仕事に集中する

属人化と非属人化

属人化すべき領域

望まれた属人化

善い属人化

「専門性」や「個性」が期待される領域

- ・ 専門性や個性が活きる (期待が明確であれば成果も出やすい)
- ・ 立ち上がりが早い (比較的短期で成果が出やすい)
- ・ 非再現的な仕事に最適 (全く新しい仕事を任せられる)

属人化すべきではない領域

望まれぬ属人化

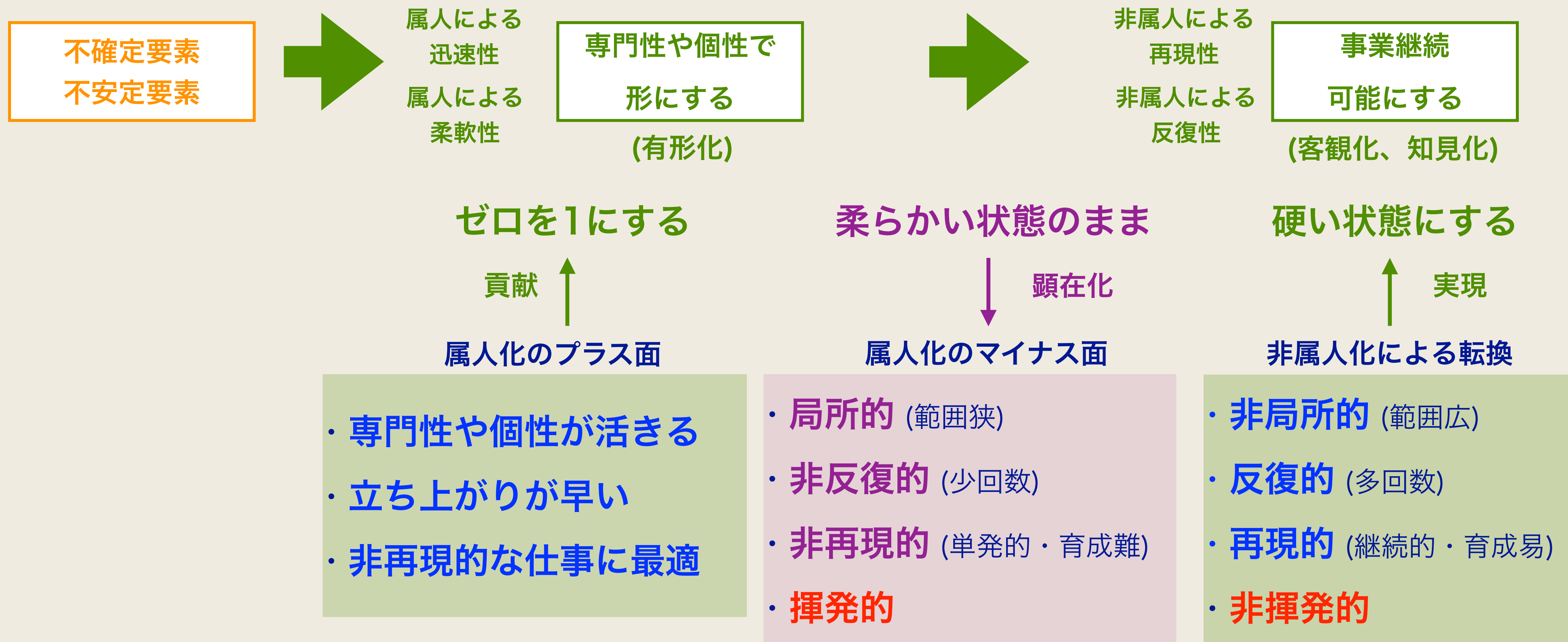
悪い属人化

「専門性」や「個性」よりも強い期待が別にある領域
(事業継続性など)

- ・ 「誰でもできる」 (手順が明確であれば成果が出やすい)
- ・ 引き継ぎが可能 (比較的短期で後任をアサインできる)
- ・ 再現的な仕事に最適 (枯れている仕事を安定して任せられる)

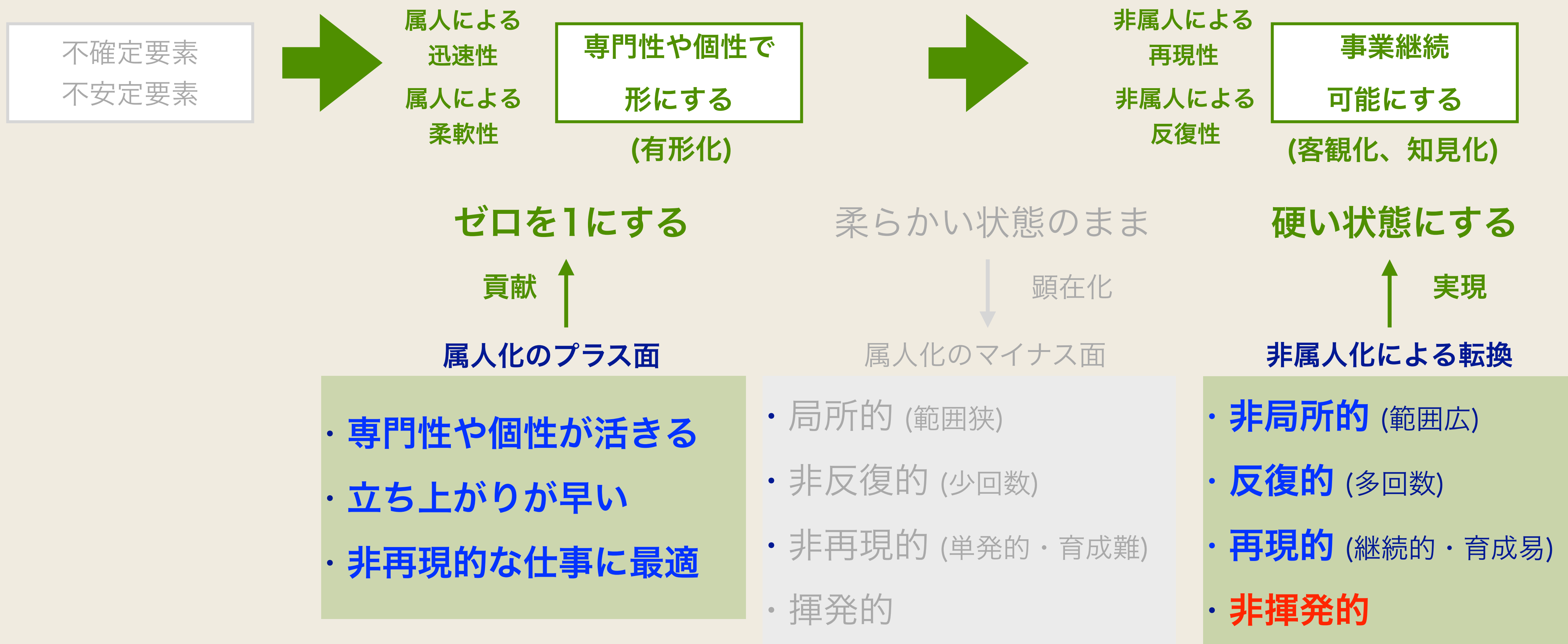
フェイズ2: ミドルの仕事に集中する

属人化と非属人化の分担



フェイズ2: ミドルの仕事に集中する

ミドルエンジニアの仕事



フェイズ2: ミドルの仕事に集中する

「他人に仕事を渡せる人」になるう

仕事の「量」を変える:
「他人の手」を使う

「仕事を抱え込む人」から「他人に仕事を渡せる人」になるう

渡せる仕事は他人に渡す

アウトプットとして「人に渡せる仕事」を作っていく

「仕事の渡し方」を学ぶ

「他人に渡せる仕事」とはどんな仕事を学び、どのように渡せば良いかを理解する

「仕事の渡し方」を身に付ける

ミドルエンジニアの仕事は原則として「他人に渡すこと」が前提になる

フェイズ2: ミドルの仕事に集中する

ミドルの2種類の仕事

仕事の「量」を変える:
「他人の手」を使う

エンジニアリング業務

不安定な業務を硬い状態にする

客観的 論理的 合理的 持続的

非属人による再現性

非属人による反復性

事業継続

可能にする

(客観化、知見化)

- ・ **非局所的** (範囲広)
- ・ **反復的** (多回数)
- ・ **再現的** (継続的・育成易)
- ・ **非揮発的**

職人的エンジニアリング業務

ゼロを1にする

客観的 論理的 合理的 持続的

属人による迅速性

属人による柔軟性

専門性や個性で

形にする

(有形化)

- ・ **専門性や個性が活きる**
- ・ **立ち上がりが早い**
- ・ **非再現的な仕事に最適**

フェイズ2: ミドルの仕事に集中する

ミドルエンジニアのエンジニアリング

仕事の「量」を変える:
「他人の手」を使う

活動に手戻りやロスが少なく、成果物が初級エンジニアにわかりやすい

ミドルエンジニアの活動において手戻りやロスが少なく、その成果物が初級エンジニアにわかりやすい

ミドルエンジニアの成果物は不可逆的な前進、発展となっている

ミドルエンジニアの成果物は、客観的、論理的、合理的、持続的であり、職能が正常に前進、発展している

ミドルエンジニアの成果物は「負債」を生まず、生まれてもすぐに解消できる

ミドルエンジニアの成果物は、負債を生む可能性が低く、仮に生まれてもすぐに解消できる仕組みになっている

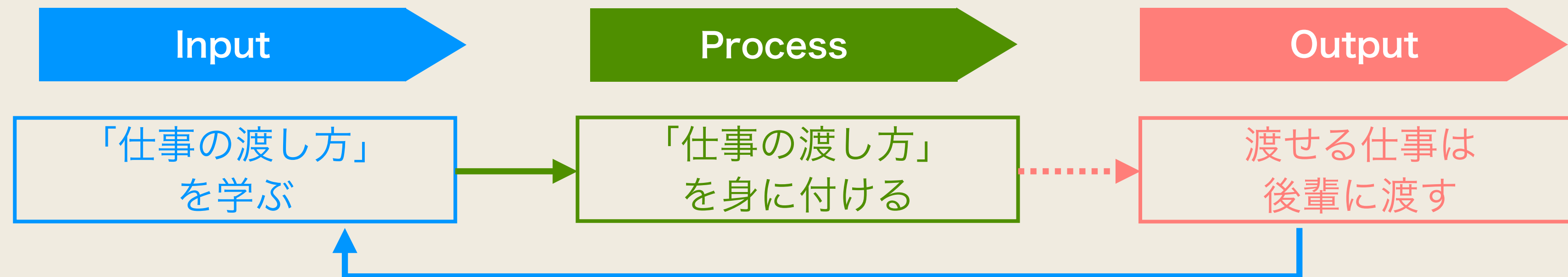
ミドルエンジニアの仕事は全て「引き継ぎ前提」で、失われる可能性が少ない

ミドルエンジニアの活動や成果物は全て引き継ぎを前提としているので、突然いなくなっても事業継続に影響は無い。

フェイズ2: ミドルの仕事に集中する

仕事の「量」を変える:
「他人の手」を使う

ミドルエンジニアは、「ミドルにしかできない仕事」をやる人



時間が圧倒的に不足する中での
インプット

「他人に渡すこと」
が前提

「人に渡せる仕事」
を作る

「仕事を抱え込む人」から「他人に仕事を渡せる人」になる

「それくらいわかるでしょう」という言葉

ドキュメントを読む人が言うのは許されます

ドキュメントを書く人は、決して言ってはいけません。

ドキュメントを良くするためのチャンスをつぶしてしまうからです

Step3. レベル2. 確立期

フェイズ3: 職能全体を育てる

ナビゲーション

Step1

ミドルエンジニアの概要

Step2

レベル1. 導入期

Step3

レベル2. 確立期

フェイズ1: 職能としての成果を出す
フェイズ2: ミドルの仕事に集中する
フェイズ3: 職能全体を育てる

フェイズ3: 職能全体を育てる

「職能としての成果って何?」という人



自分自身の成長だけではなぜだめなのか
わからない

なにが「職能を育てること」になるのか
わからない

どうすれば「組織としての成長」するのか
わからない

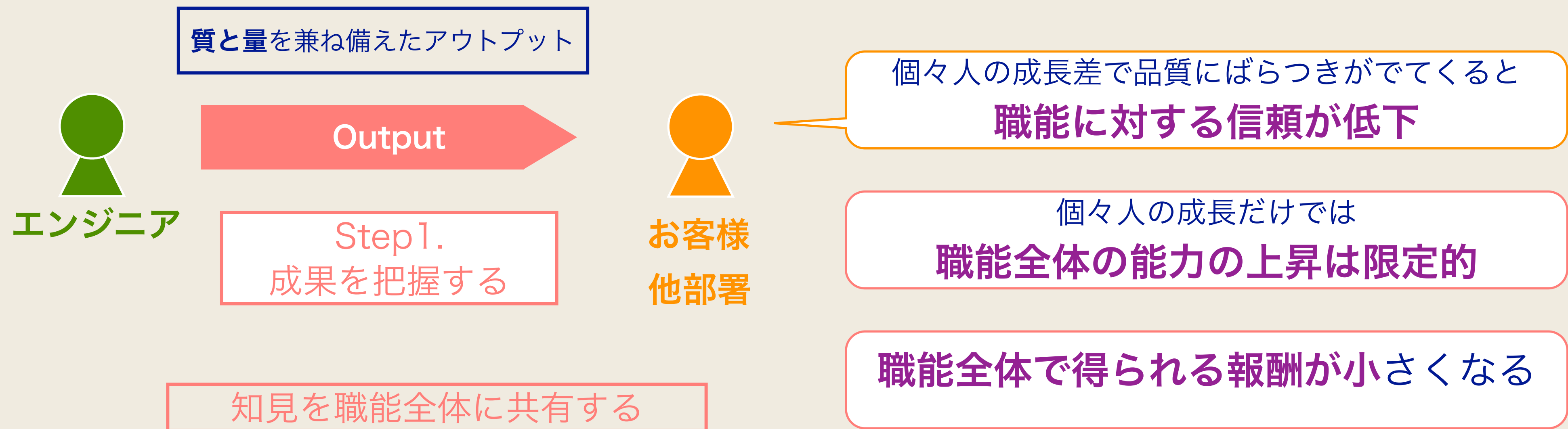
仕事の「量」を変える: 「他人の手」を使う

フェイズ3: 職能全体を育てる

自分自身の成長だけではなぜだめなのかわからない

仕事の「量」を変える:
「他人の手」を使う

個々人での成長だけではテコが効かず、**職能全体の成長・発展に繋がらない**



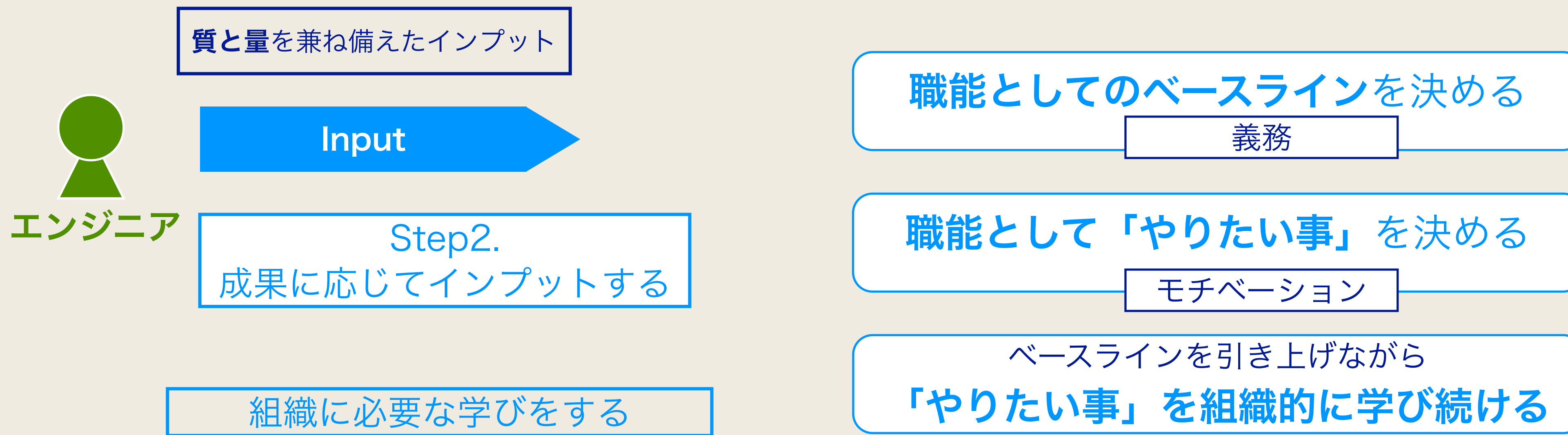
職能の能力を引き上げ続けていくことが自然とできる必要がある

フェイズ3: 職能全体を育てる

なにが「職能を育てること」になるのかわからない

仕事の「量」を変える:
「他人の手」を使う

需要の範囲で「ベースライン」と「やりたい事」の双方を軸に職能を育てる



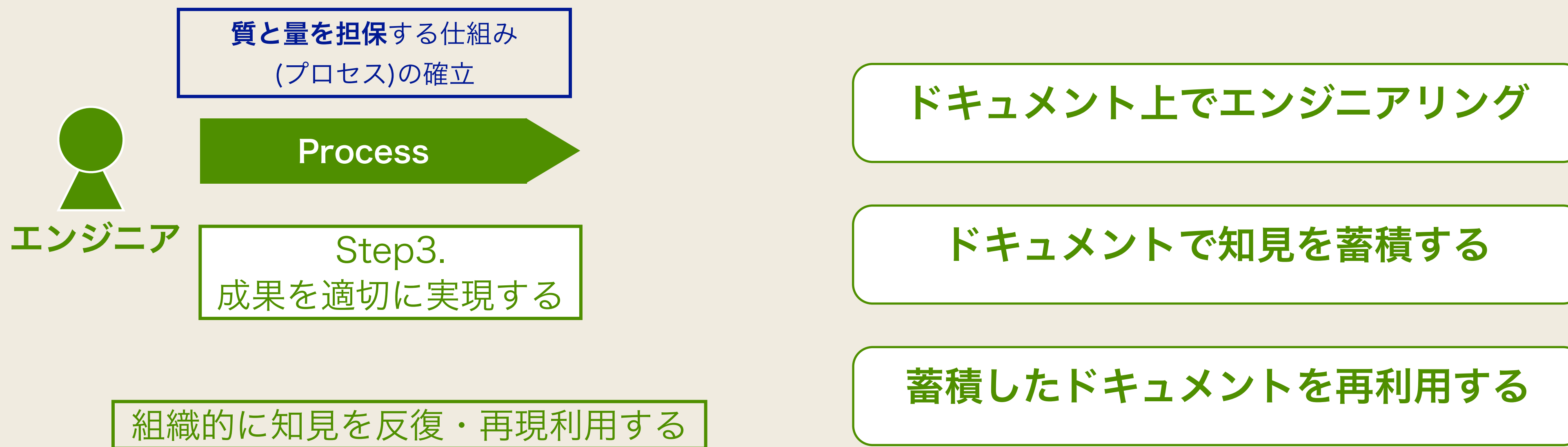
個人知ではなく「組織知」としての学びにする必要がある

フェイズ3: 職能全体を育てる

どうすれば「組織としての成長」するのかわからない

仕事の「量」を変える:
「他人の手」を使う

ミドルエンジニア側からは、ドキュメントベースのコミュニケーションが原則となる



「自分が居なくても回る」ことを意識する

フェイズ3: 職能全体を育てる

「職能を育てるミドル」になるう

仕事の「量」を変える:
「他人の手」を使う

「自分のために学ぶ人」から「**職能全体のために学ぶ人**」になるう

知見を職能全体に共有する

職能の能力を引き上げ続けていくことが自然とできる

職能全体に必要な学びをする

個人知ではなく「**組織知**」としての**学び**にする必要がある

組織的に知見を反復・再現利用する

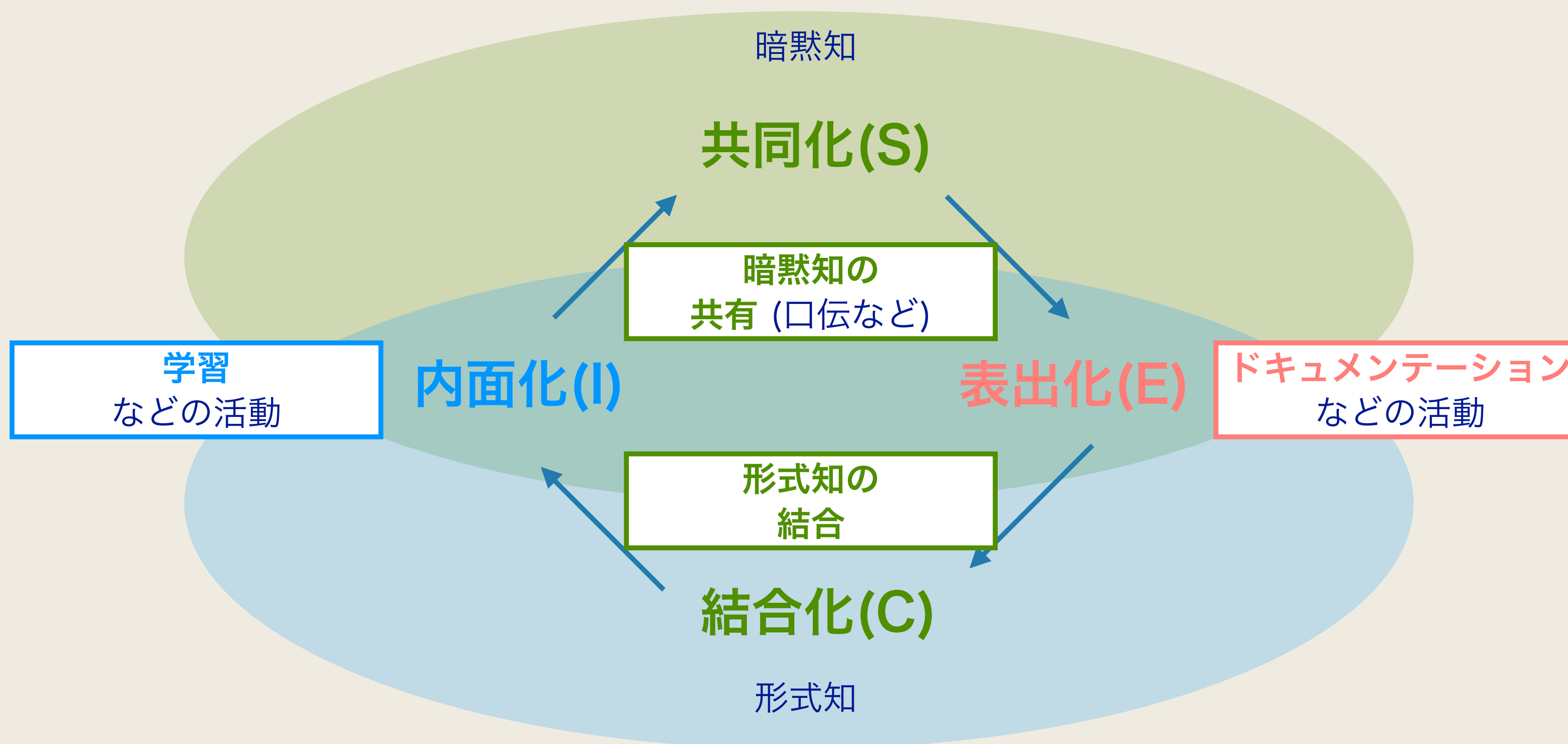
「**自分が居なくても回る**」ことを意識する

フェイズ3: 職能全体を育てる

SECIモデル

仕事の「量」を変える:
「他人の手」を使う

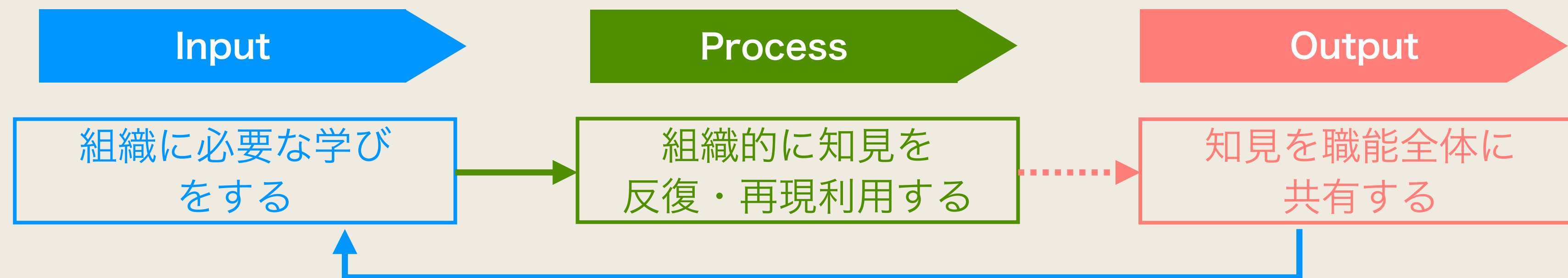
個人の知識を組織的な知識資源に転換していくプロセス



フェイズ3: 職能全体を育てる

仕事の「量」を変える:
「他人の手」を使う

ミドルエンジニアは、「職能全体の育成」を主導する人



「組織知」
としての学び

「自分が居なくても回る」こと
を意識

職能の能力を
引き上げ続けていく

「自分のために学ぶ人」から「職能全体のために学ぶ人」になる

Step3. レベル2. 確立期

まとめ

はじめに

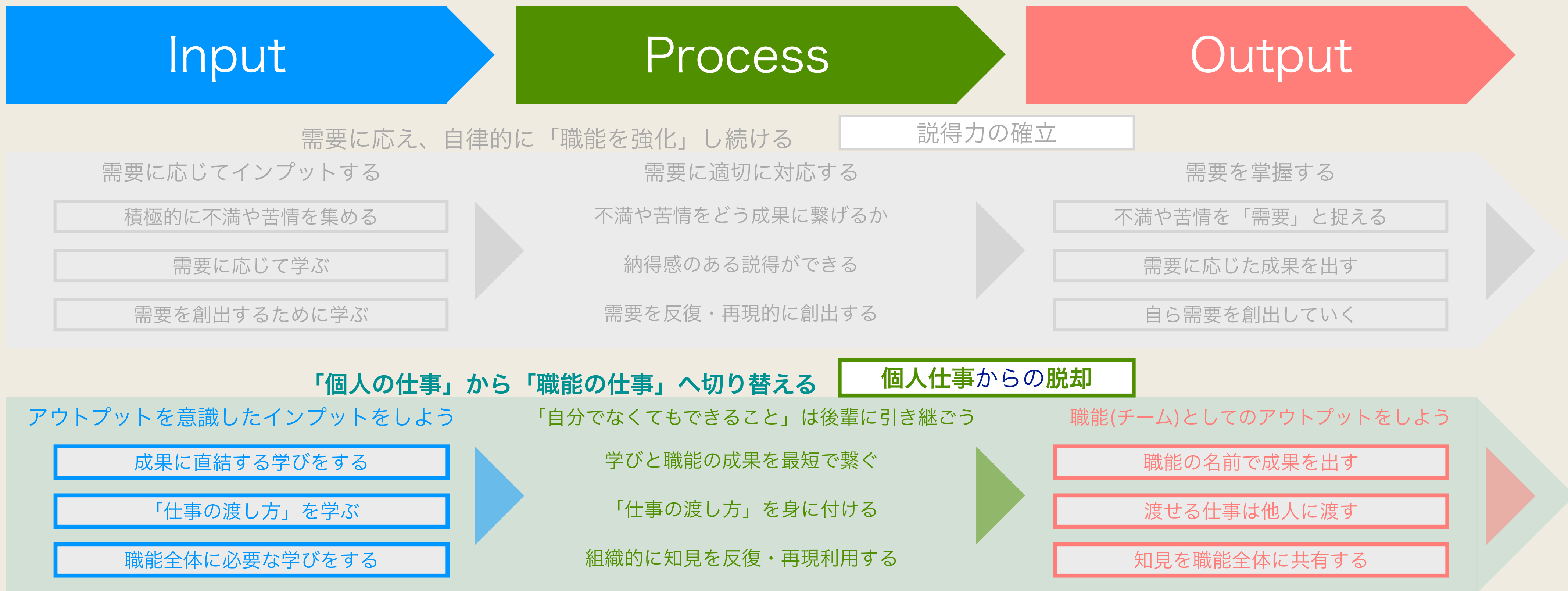
学習ガイド

本編

まとめ

ステップアップ

まとめ: レベル2. 確立期



仕事の「量」を変える: 「他人の手」を使う

課題1 解答 (例: レベル2. 確立期)

以前の講義で、「ミドルエンジニアは、現場の主戦力である」と説明しましたが、
現場においてミドルエンジニアに求められる役割に、
どのようなものがあると考えますか？

大きなものを3つ考えてみてください。

需要の掌握・創出

実質的な意思決定

業務を創る・渡す

「個人の仕事」から「職能の仕事」へ切り替える

課題2 解答 (例)

現場においてミドルエンジニアに求められる役割を担うために、
どのような能力やスキルが必要になると考えますか？

大きなものを3つ考えてみてください。

コミュニケーション
能力

説得力

ドキュメント
能力

「個人の仕事」から「職能の仕事」へ切り替える

まとめ

はじめに

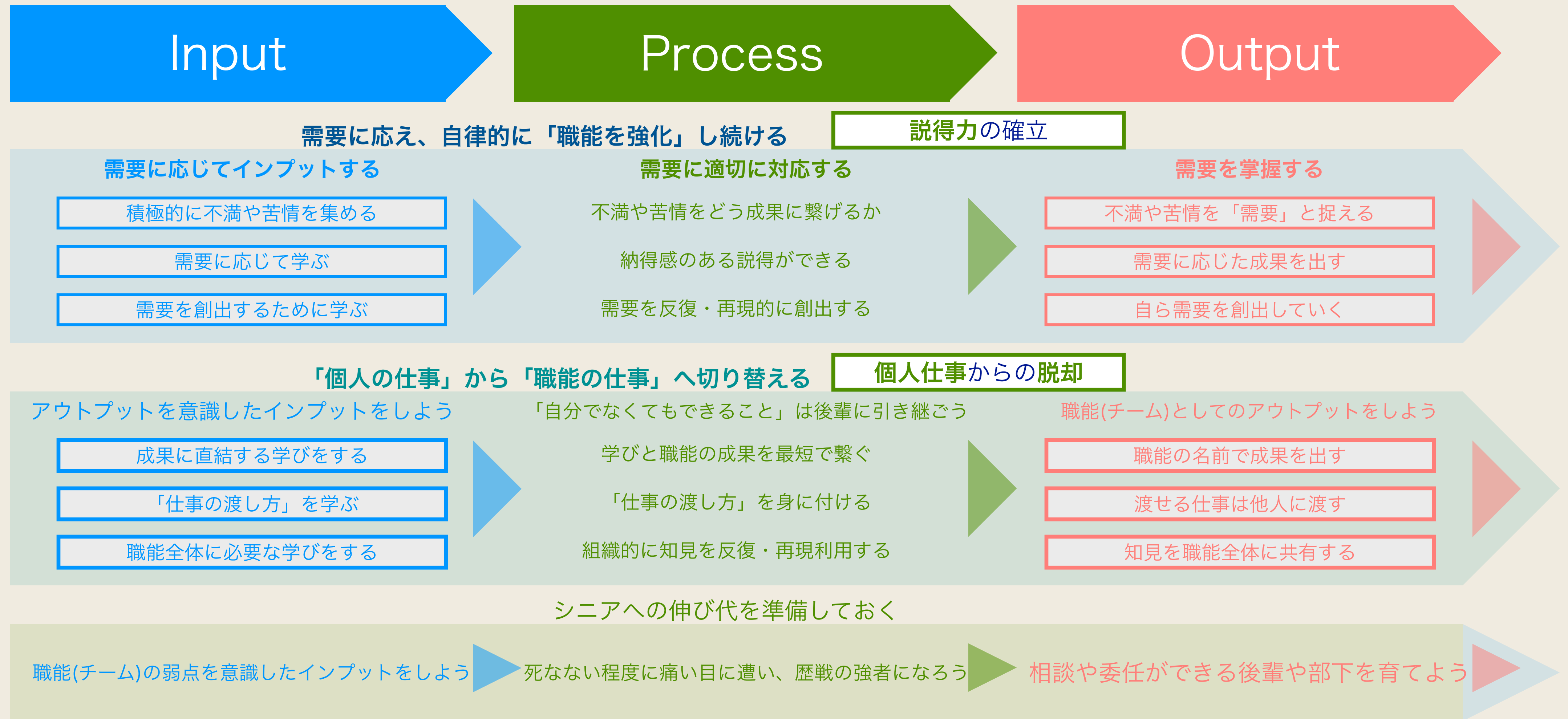
学習ガイド

本編

まとめ

ステップアップ

まとめ: ミドルエンジニアの成長ステップ



おわりに: シニアエンジニアに進むその前に

はじめに

学習ガイド

本編

まとめ

ステップアップ

おわりに: シニアエンジニアに進むその前に

自分の専門範囲で視野狭窄にならない

ミドル同士でコンフリクトを起こしやすくなる

トレードオフで語る

自分の立場に固執しすぎないことが大切になる

シニアは、サービス全体の最適を見なければならない

プロとして譲歩してはいけないところはちゃんと押さえながら
ビジネスへの最適を追求する

まとめ

はじめに

学習ガイド

本編

まとめ

ステップアップ

この講義で一番大切なこと

上司を説得できない、後輩に仕事を渡せない
のは**エンジニアリング力不足**

客観的

論理的

合理的

持続的

+

MECE

ダブリがなく

モレもない

職能として**結果**を出さなければならない (=報酬)

この講義の全体ふりかえり

Step1. ミドルエンジニアの概要

得意領域が明確なエンジニアで、**職能(チーム)の主戦力**として扱われる

Step2. ミドルエンジニアの導入期

仕事の「質」を変える: 「**需要**」に応じていく

Step3. ミドルエンジニアの確立期

仕事の「量」を変える: 「**他人の手**」を使う

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

はじめに

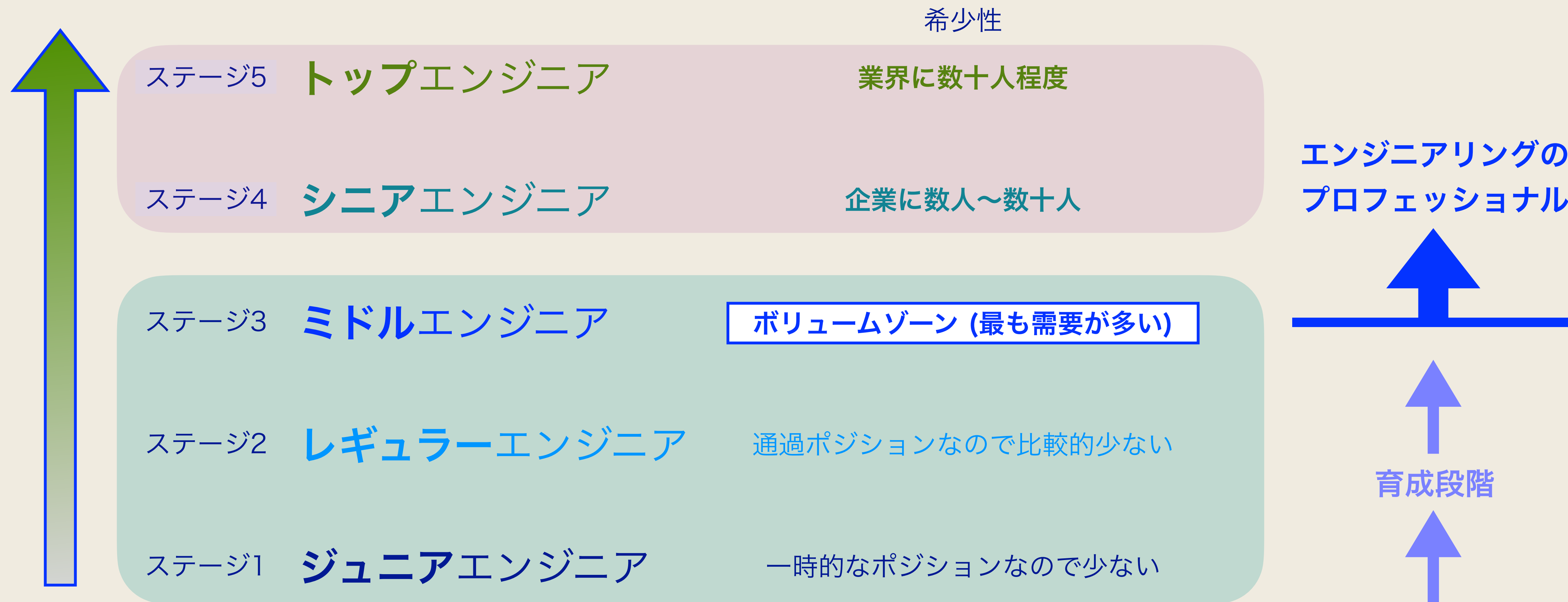
学習ガイド

本編

まとめ

ステップアップ

受講内容を活用して「未来像」を描こう



受講内容を活用して「実際の行動」を行おう

ミドルエンジニア以降は「やれる事」「活躍できる場」がどんどん増えていきます

需要に応じてインプットする

- 積極的に不満や苦情を集める
- 需要に応じて学ぶ
- 需要を創出するために学ぶ

需要に適切に対応する

- 不満や苦情をどう成果に繋げるか
- 納得感のある説得ができる
- 需要を反復・再現的に創出する

需要を掌握する

- 不満や苦情を「需要」と捉える
- 需要に応じた成果を出す
- 自ら需要を創出していく

アウトプットを意識したインプットをしよう

- 成果に直結する学びをする
- 「仕事の渡し方」を学ぶ
- 職能全体に必要な学びをする

「自分でなくてもできること」は後輩に引き継ごう

- 学びと職能の成果を最短で繋ぐ
- 「仕事の渡し方」を身に付ける
- 組織的に知見を反復・再現利用する

職能(チーム)としてのアウトプットをしよう

- 職能の名前で成果を出す
- 渡せる仕事は他人に渡す
- 知見を職能全体に共有する

シニアへの伸び代を準備しておく

職能(チーム)の弱点を意識したインプットをしよう

死なない程度に痛い目に遭い、歴戦の強者になろう

相談や委任ができる後輩や部下を育てよう

この講義から更にその先へ

以下のキーワードや、本講義(1～3回)に出てきた単語から興味があるテーマがあればぜひお声掛けください。

運用プロセス設計

ドキュメンテーション

ドキュメンテーション実践 (手順書の設計)

運用設計のためのオブジェクト指向分析・設計

運用自動化のアンチパターンとベストプラクティス

サービス監視デザイン