

「運用エンジニア」のためのエンジニアリング入門 2

初級エンジニアの成長ステップ

(ジュニア/レギュラー)

運用設計ラーニング

この講義の対象者

初級エンジニア（ジュニア/レギュラー）

- ・ 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

初級エンジニアを指導・育成する立場の方

- ・ ジュニア/レギュラーエンジニアの指導・育成をする先輩や上司

悩み: 初級エンジニアの成長ステップ

- ・ ジュニア/レギュラーエンジニアには**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: 初級エンジニアの成長ステップ

正しい「ジュニア/レギュラーエンジニア」になり、次のステージに進む

- ・ジュニア/レギュラーエンジニアとして、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。

ジュニア/レギュラーエンジニアを指導・育成する立場(先輩や上司)の方

- ・指導・育成対象のジュニア/レギュラー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・育成対象のジュニア/レギュラーエンジニアに、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

- ・ ジュニア/レギュラーエンジニアに対する**期待が明確**になる。
- ・ ジュニア/レギュラーエンジニアに**必要な考え方を理解**できる。
- ・ ジュニア/レギュラーエンジニアに**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

エンジニアの成長サイクル

初級エンジニアの成長ステップ

ジュニアエンジニアの成長ステップ

レギュラーエンジニアの成長ステップ

この講座のロードマップ

スタートとゴール

スタート

初級エンジニア(ジュニア、レギュラー)としての成長や育成に悩んでいる

着実かつ再現性のある成長・育成とはどのようなものを理解する

エンジニアの成長サイクル

主体的に初級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

初級エンジニアの成長ステップ

ゴール

目標や自信を持って初級エンジニア(ジュニア、レギュラー)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

エンジニアの成長サイクル

Step2

ジュニアエンジニアの成長ステップ

概要

レベル1 導入期

レベル2 確立期

Step3

レギュラーエンジニアの成長ステップ

概要

レベル1 導入期

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. エンジニアの成長サイクル
- Step2. ジュニアエンジニアの成長ステップ
- Step3. レギュラーエンジニアの成長ステップ

Step1. エンジニアの成長サイクル

ナビゲーション

Step1

エンジニアの成長サイクル

Step2

ジュニアエンジニアの成長ステップ

概要

レベル1 導入期

レベル2 確立期

Step3

レギュラーエンジニアの成長ステップ

概要

レベル1 導入期

レベル2 確立期

エンジニアの成長サイクル

「エンジニアの成長」は
どのように進むと考えますか？

エンジニアの成長サイクル

エンジニアとしての「成長」がイメージできない、実感できない

勉強する気がおきない

エンジニアとしての成長の仕方がわからないので、モチベーションが沸かず、勉強する気が起きない

学んだことをどう実務に活かすかわからない

勉強はしているが、それをどのように実務に繋げられるかわからない

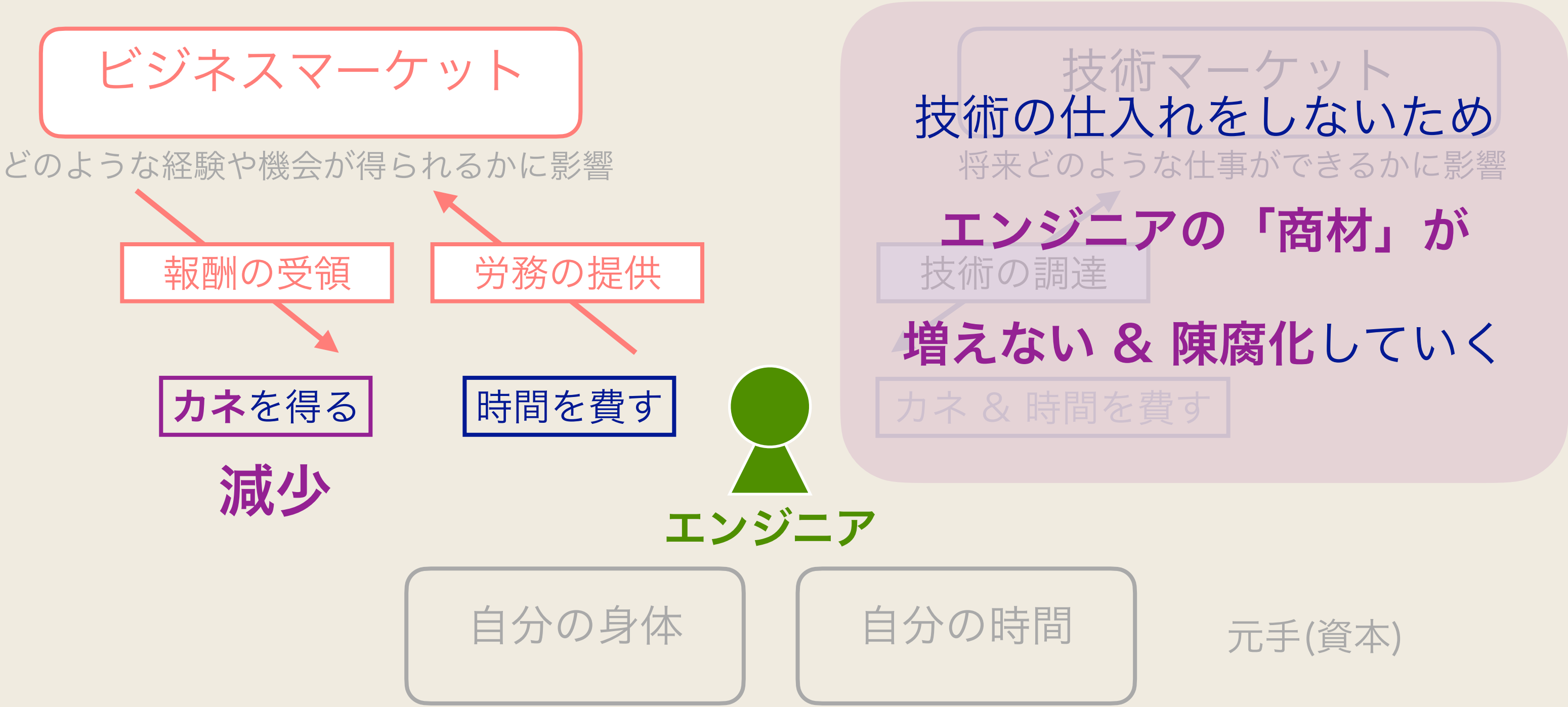
エンジニアらしい学び方や成長の仕方がわからない

勉強はしているが、それがどのように「エンジニア」としての成長につながるかわからない

エンジニアの成長サイクル

勉強する気がおきない、とどうなるか？

エンジニアの商材とも言える「技術」が増加しない&陳腐化し、報酬が減少していく

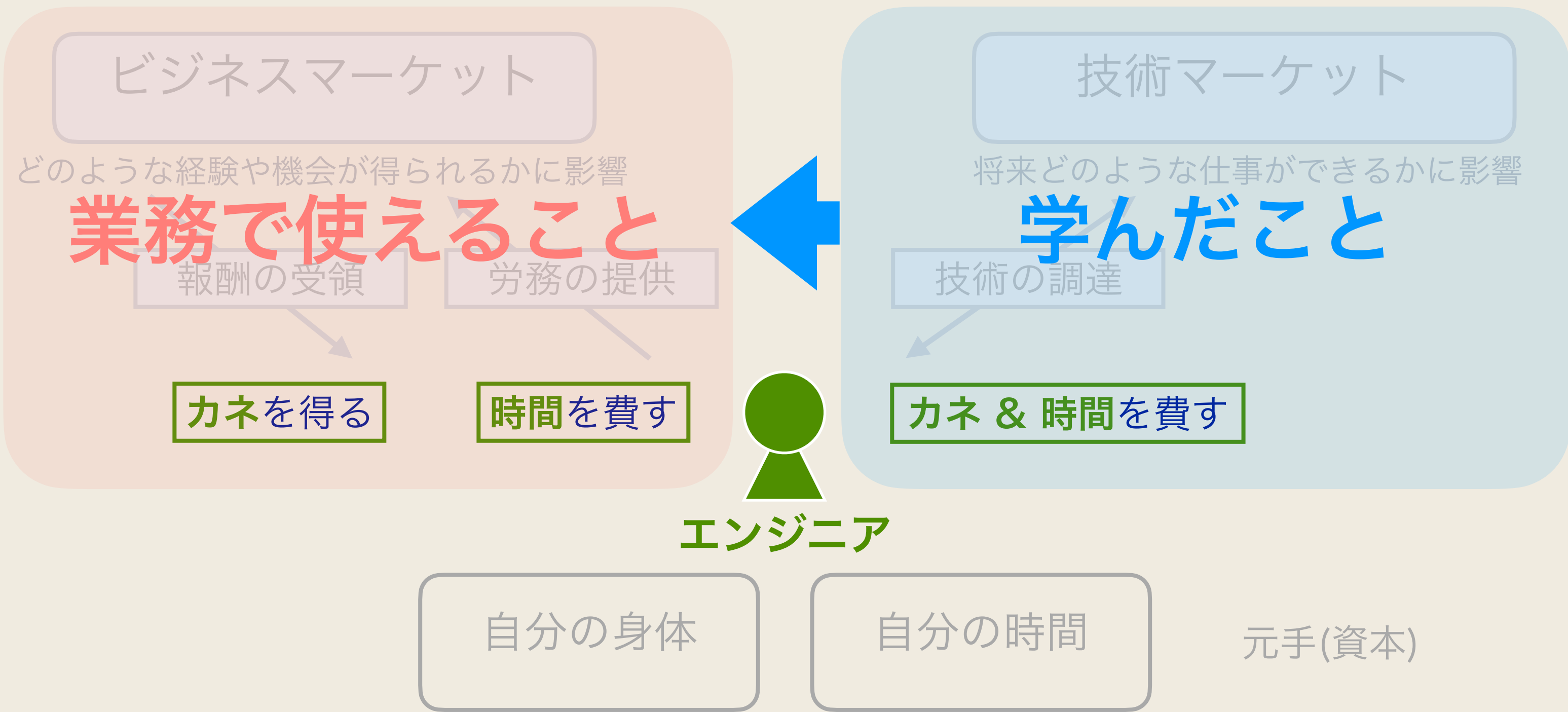


エンジニアは現役である限り、技術の調達と水準の維持をし続けなければ生き残れない

エンジニアの成長サイクル

学んだことをどう実務に活かすかわからない

エンジニアの学びは、原則として業務で使えないと意味がない

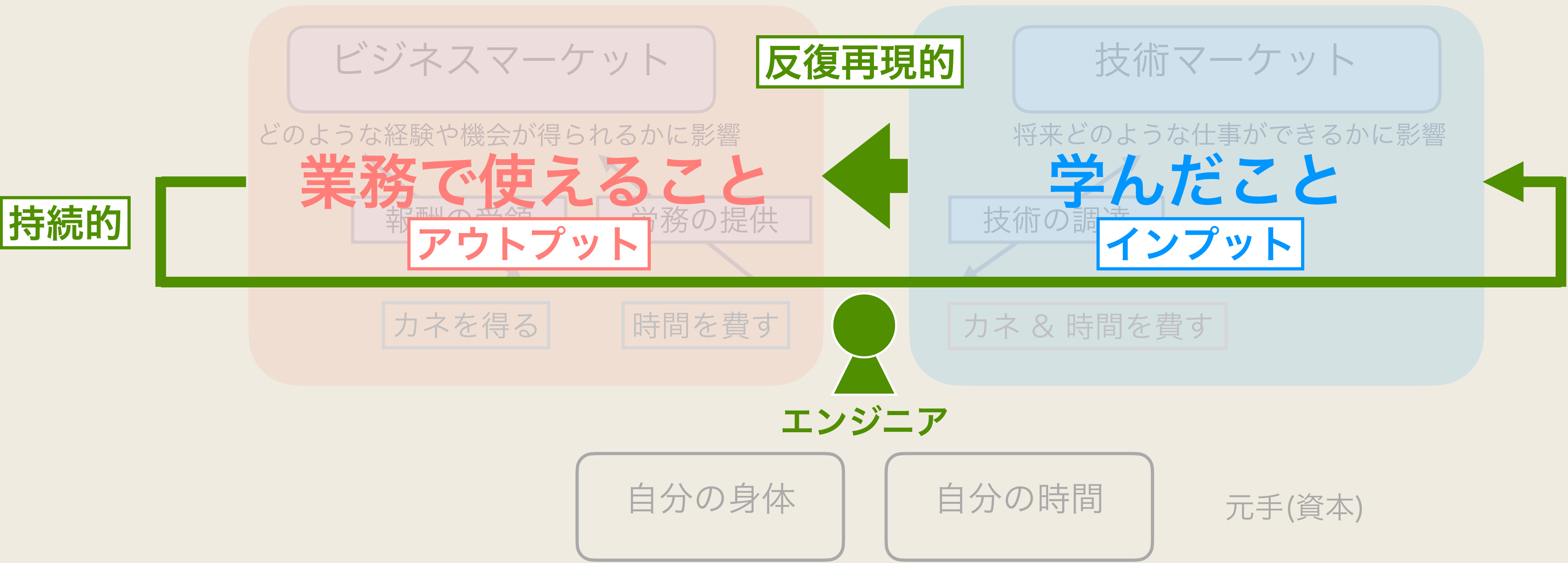


まず「業務で使う」があって、次に「何を学ぶ」が来るのが原則。(基礎理論などは業務に直結しなくても必須)

エンジニアの成長サイクル

エンジニアらしい学び方や成長の仕方がわからない

エンジニアの学びや成長では、エンジニアらしく**反復再現的、持続的に学び・成長**する

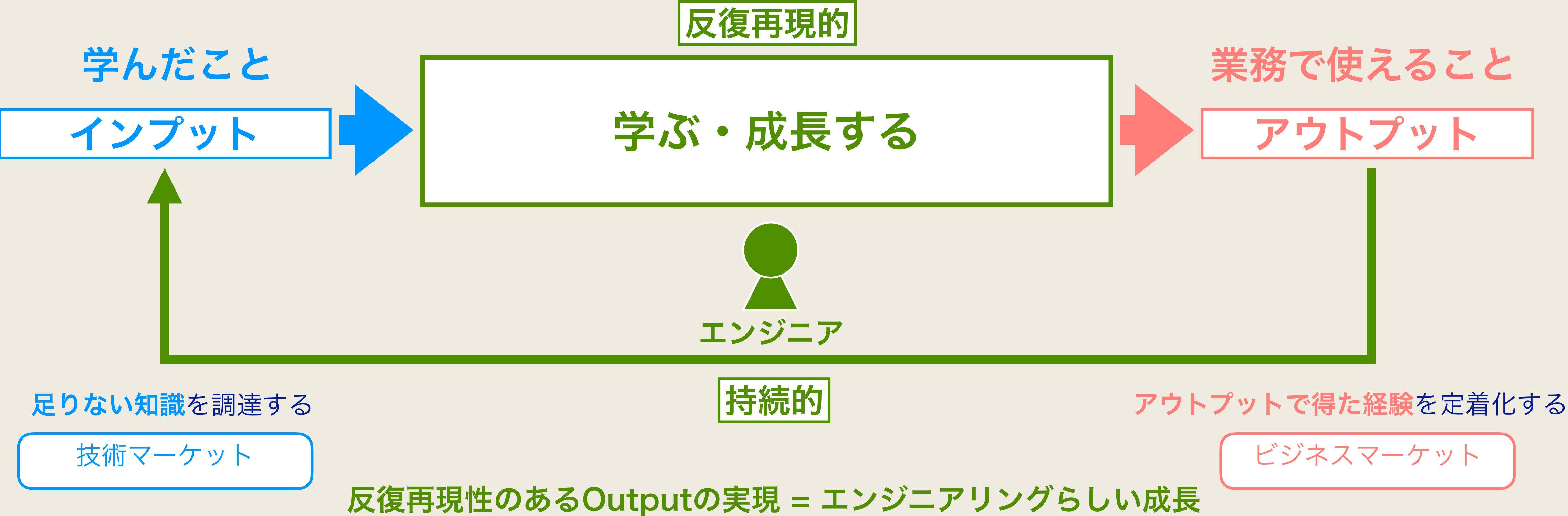


エンジニアにとっては「学びも仕事のうち」であるため、**学ぶ行為自体にも反復再現性や持続性が期待される**

エンジニアの成長サイクル

エンジニアらしい学び方や成長の仕方

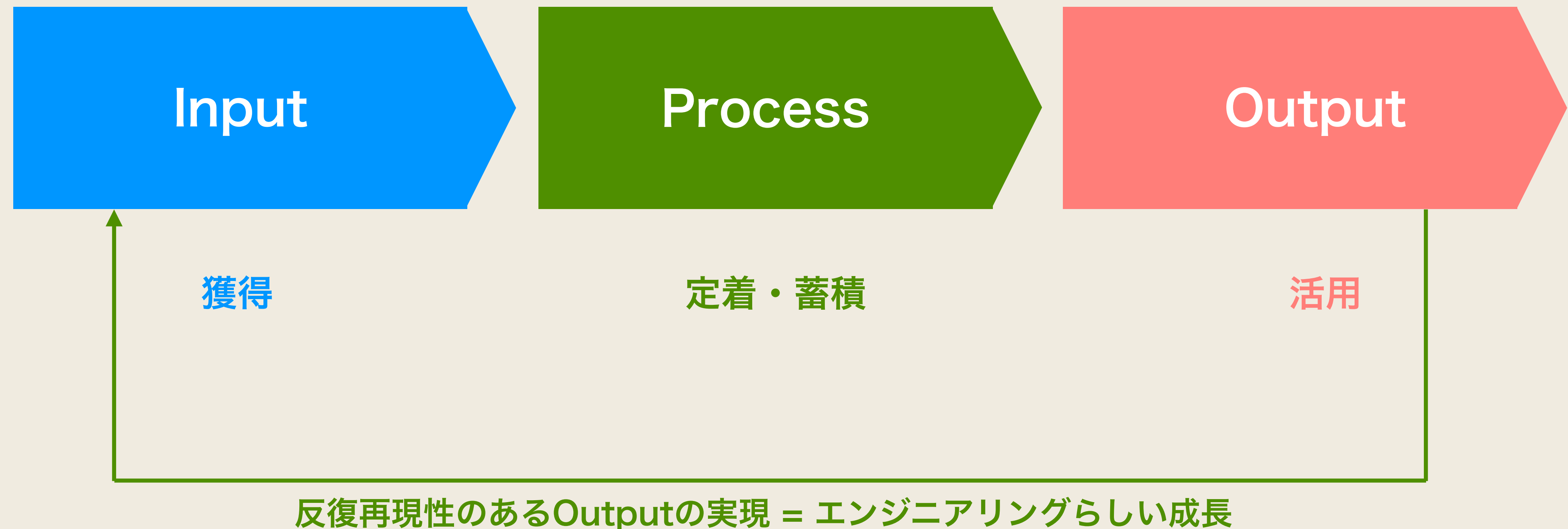
エンジニアは業務に直結する知識を反復再現的、持続的に学び・成長する



エンジニアの成長サイクル

エンジニアらしい学び方や成長のモデル

エンジニアは成果だけでなく、自分自身の成長についても**反復再現的、持続的に実現する**



エンジニアの成長サイクル

成長サイクルにおける2つの柱「知識」と「経験」

エンジニアの成長には、「知識」と「経験」の2つの柱が重要

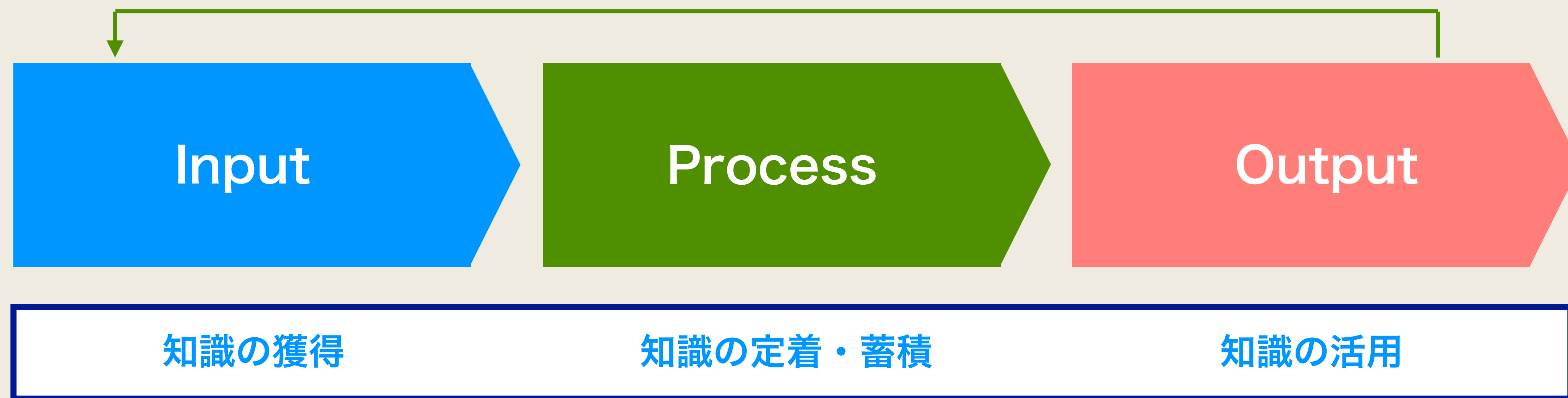


「知識だけ」では現実と乖離した机上の空論になりやすく、「経験だけ」では体系化されず応用が効かなくなりやすい

エンジニアの成長サイクル

ジュニアエンジニアの成長サイクル

ジュニアエンジニアは、まず「知識」から学ぶ



経験の獲得

経験の定着・蓄積

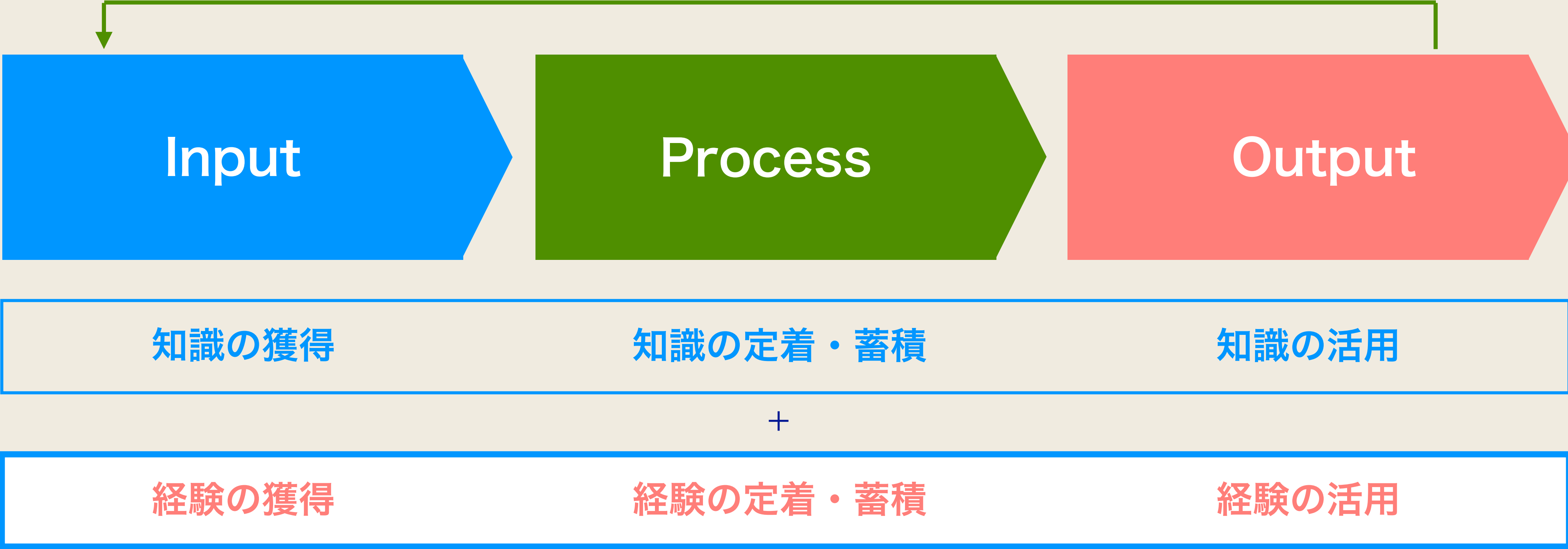
経験の活用

今後得られる「経験」を体系化するためには、まず適切な「知識」が必要

エンジニアの成長サイクル

レギュラーエンジニアの成長サイクル

レギュラーエンジニアは、「経験」からも学ぶ

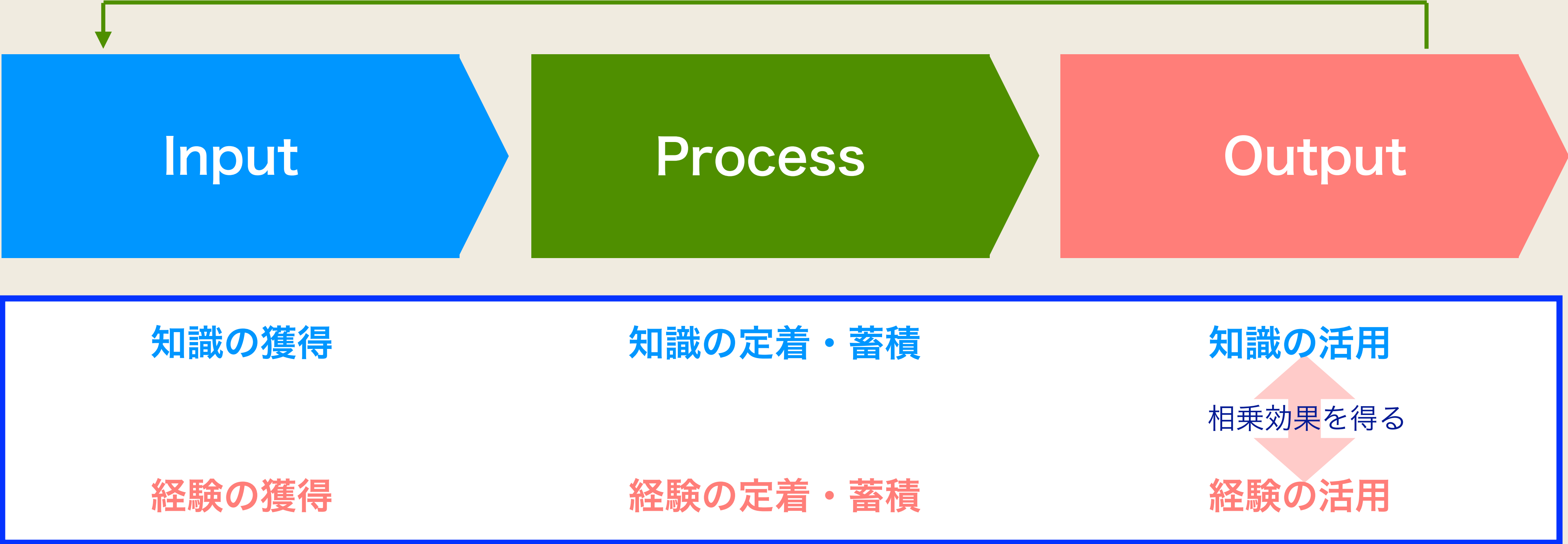


持っている「知識」に、現実から得られた「経験」を肉付けをしていく

エンジニアの成長サイクル

ミドルエンジニアの成長サイクル

ミドルエンジニアは、「知識」と「経験」を掛け算して活用する

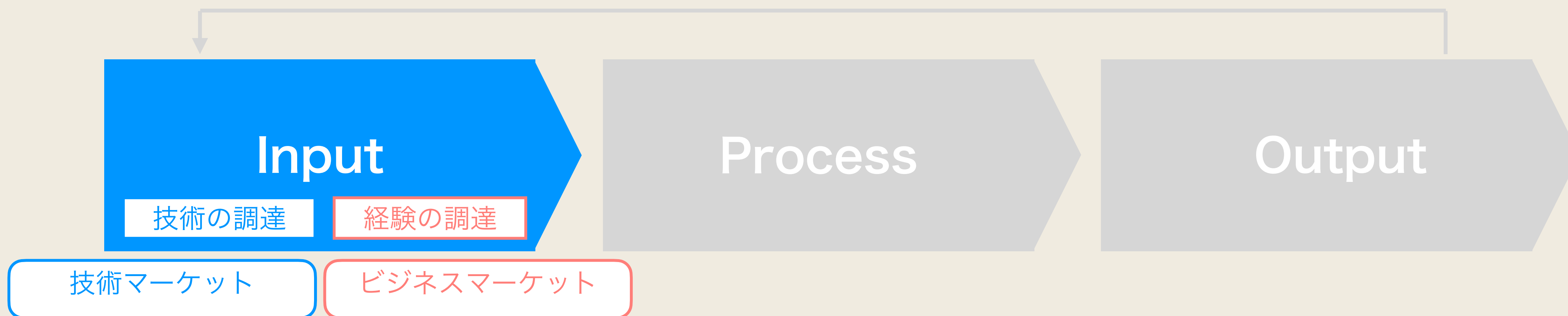


持っている「知識」と「経験」を双方向で掛け合わせて、新たな知識や経験を**効果的に身に付けていく**

エンジニアの成長サイクル

留意点1. 「インプットの質」と「アウトプットの質」

アウトプットの質がインプットの質以上になることは無い



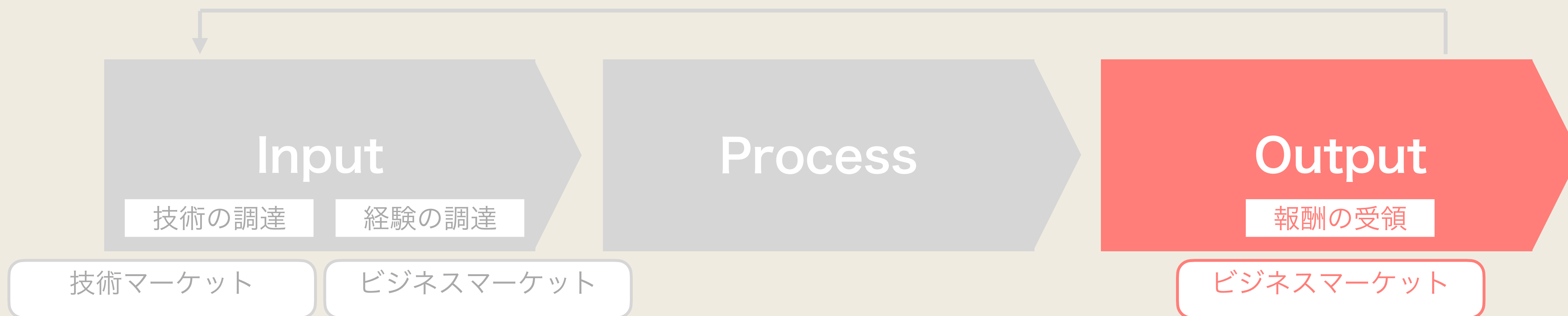
インプットの質が低ければ、当然にアウトプットの質は更に低くなる

アウトプットの質を高めるためには、**まずインプットの質を上げる**必要がある

エンジニアの成長サイクル

留意点2. 「アウトプット」と「ビジネスマーケット」

アウトプットがそのまま**ビジネスマーケットの報酬に直結**する



インプットの質が高ければ、結果として報酬が高くなる可能性が向上する

エンジニアの学びは結果的に「報酬の増額」に結実しやすい、と言える
(やる気の一つの源泉)

エンジニアの成長サイクル

留意点3. プロセスの役割と習熟効果

プロセスは、InputとOutputを最短で繋ぐことが求められる

プロセスの役割



学習を重ね、累積の経験量の増加とともに、無駄が減り効率化も進む

プロセスの習熟効果

累積経験が少ない&回り道をするライバルに比べて
同一内容の学習や習得に必要な時間が大幅に低減していく

良く学ぶ人ほど
成長効率が良くなる

エンジニアの成長サイクル

まとめ

エンジニアは成果だけでなく、自分自身の成長についても**反復再現的、持続的に実現する**



ジュニアエンジニア:

まず「知識」から学ぶ

レギュラーエンジニア:

「経験」からも学ぶ

ミドルエンジニア:

「知識」と「経験」を掛け算して活用する

アウトプットの質が

インプットの質以上になることは無い

エンジニアの学びは

結果的に「報酬の増額」に結実しやすい

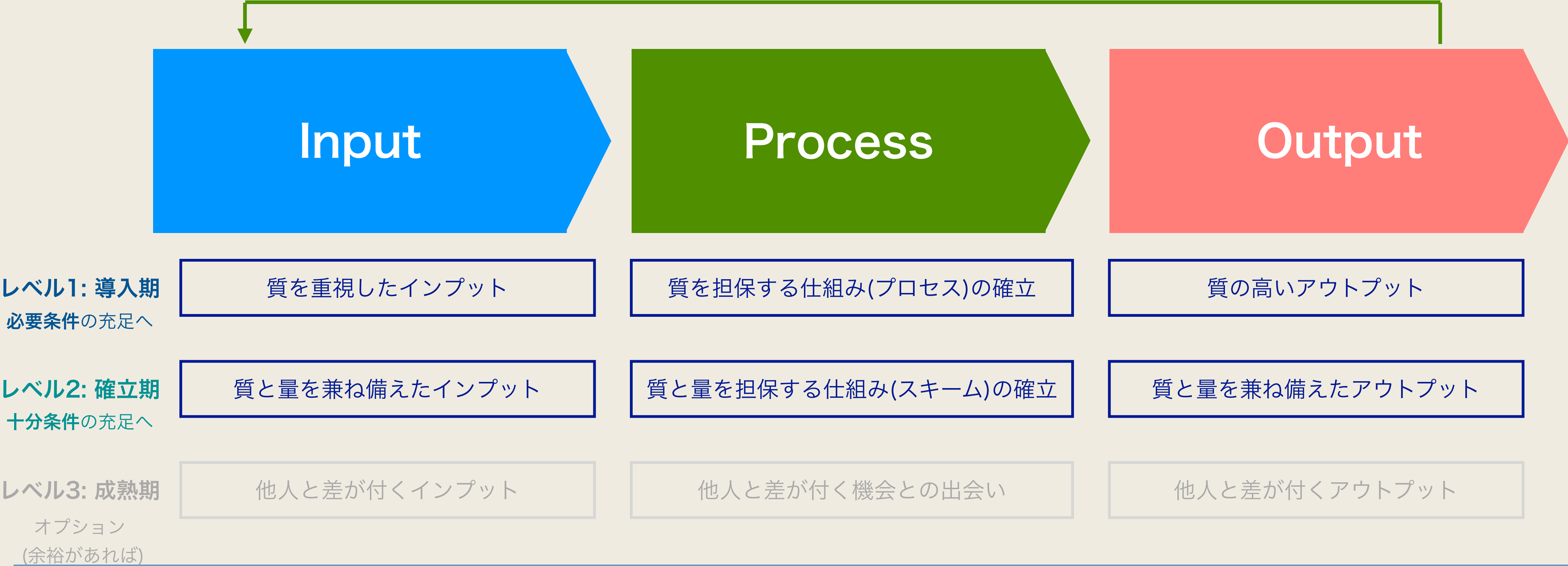
エンジニアの学びでは

良く学ぶ人ほど**成長効率が良くなる**

エンジニアの成長サイクル

各エンジニアステージ内での成長マトリクス

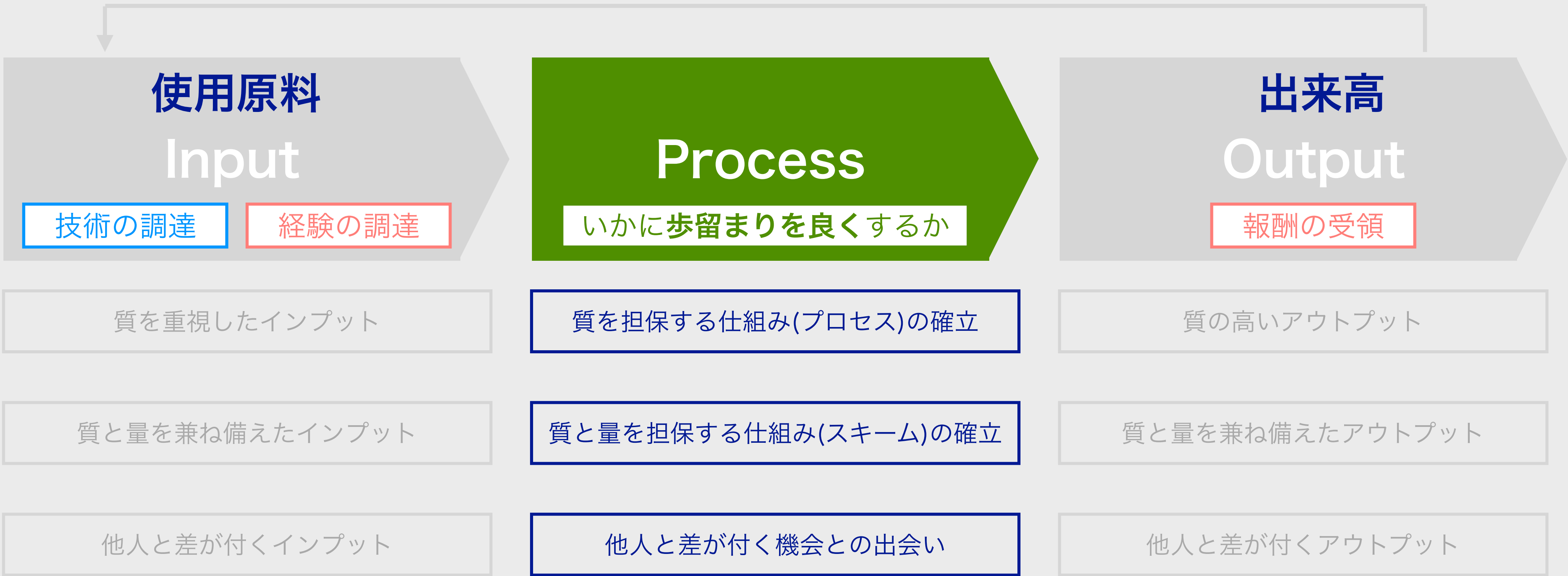
本講座では、以下の**2レベル x 3工程**で、各エンジニアの成長ステップを解説します



参考: 「歩留まり」という考え方

「歩留まり」とは

加工に際し、**使用原料**に対する製品の**出来高の割合**。(生産効率を示す)



Step2. ジュニアエンジニアの成長ステップ

ジュニアエンジニアの成長ステップ

入職から

「ジュニアエンジニア」として一定のレベルになるまでに必要な過程
はどのようなものになるのでしょうか？

ステップバイステップで考えてみてください。

Step2. ジュニアエンジニアの成長ステップ

「ジュニアエンジニア」の概要

ナビゲーション

Step1

エンジニアの成長サイクル

Step2

ジュニアエンジニアの成長ステップ

概要

レベル1 導入期

レベル2 確立期

Step3

レギュラーエンジニアの成長ステップ

概要

レベル1 導入期

レベル2 確立期

「ジュニアエンジニア」の概要

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

- ・ **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
- ・ **読解力**

人との繋がり

相談できる先達を探す

「ジュニアエンジニア」の概要

ジュニアエンジニアに何が求められるのかわからない

はじめて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

「ジュニア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

どこまでやればジュニアエンジニアとして認められるのかわからない

エンジニアとして困らないために、**ジュニア時代にやるべきこと**を知りたい

ジュニア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

「ジュニアエンジニア」の概要

はじめて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

エンジニアとして活動するために「言語化された知識を学ぶ力」が最初に求められる

「学び方」を知らない人や「学び続ける力」が無い人は、エンジニアとして生き残ることが難しくなっていく

最初は、質の良い「知識」を確実に獲得する術を身に付ける

学習能力の確立

レベル1 [導入期]

「知識」から学ぶ力を固める

これができるまでは
「員数外」扱い

「ジュニアエンジニア」の概要

「ジュニア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

学んだ「知識」を、**実際の業務に紐付けられる能力**が求められる

実務の現場では「実務家」が求められるのであり、ただの「物知り」は必要ない人として扱われてしまう

先輩の**フォローを受けながら実務**ができるとジュニアエンジニアとして**及第点**

読解力の確立

文章や言葉を正確に読み取り、**自分の知識や経験に紐付けて**適切に理解し、評価・判断する

レベル2 [確立期]

「知識」を「実務」に繋がられるようになる

これができると
「及第点」扱い

「ジュニアエンジニア」の概要

エンジニアとして困らないために、ジュニア時代にやるべきことを知りたい

ジュニア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

他のジュニアエンジニアがあまりやらないような**インプット**

他のジュニアエンジニアがあまりやらないような**プロセス**

他のジュニア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]

レギュラー以降の伸び代を準備しておく

昇格待ち相当

「ジュニアエンジニア」の概要

ジュニアエンジニアとしての必要条件、十分条件、オプション

レベル1 [導入期]

「知識」から学ぶ力を固める

員数外扱い

必要条件の充足へ

エンジニアとして活動するために「言語化された知識を学ぶ力」が最初に求められる

学習能力の確立

レベル2 [確立期]

「知識」を「実務」に繋がられるようになる

及第点相当

十分条件の充足へ

先輩のフォローを受けながら実務ができる

学んだ「知識」を、実際の業務に紐付けられる能力が求められる

読解力の確立

レベル3 [成熟期]

レギュラー以降の伸び代を準備しておく

昇格待ち相当

オプション
(余裕があれば)

ジュニア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

「ジュニアエンジニア」の概要

ジュニアエンジニアの成長マトリクス

	Input	Process	Output
レベル1: 導入期 必要条件の充足へ	質を重視したインプット 最初は、 質の良い「知識」 を確実に獲得する術を身に付ける	質を担保する仕組み(プロセス)の確立	質を重視したアウトプット 学習能力の確立
レベル2: 確立期 十分条件の充足へ	質と量を兼ね備えたインプット 先輩の フォローを受けながら実務 ができるとジュニアエンジニアとして 及第点	質と量を担保する仕組み(プロセス)の確立	質と量を兼ね備えたアウトプット 読解力の確立
レベル3: 成熟期 オプション (余裕があれば)	他人と差が付くインプット 他のエンジニアとの差別化を図るための準備をする	他人と差が付く機会との出会い	他人と差が付くアウトプット

「ジュニアエンジニア」の概要

ジュニアエンジニアの仕事は、**インプットがメイン** (学習能力の確立)

レベル1 [導入期]

「知識」から学ぶ力を固める

必要条件の充足へ

最初は、**質の良い「知識」**を確実に獲得する術を身に付ける

学習能力の確立

レベル2 [確立期]

「知識」を「実務」に繋がられるようになる

十分条件の充足へ

先輩の**フォロー**を受けながら**実務**ができるとジュニアエンジニアとして**及第点**

読解力の確立

レベル3 [成熟期]

レギュラー以降の伸び代を準備しておく

オプション (余裕があれば)

他のエンジニアとの差別化を図るための準備をする

Step2. ジュニアエンジニアの成長ステップ

レベル1. 導入期

ナビゲーション

Step1

エンジニアの成長サイクル

Step2

ジュニアエンジニアの成長ステップ

概要

レベル1 導入期

レベル2 確立期

Step3

レギュラーエンジニアの成長ステップ

概要

レベル1 導入期

レベル2 確立期

レベル1. 導入期

ジュニアエンジニアの最初の戸惑い

「何を学べばいいのか」わからない

「どのように学べばいいのか」わからない

「どこまで学ばなければいけないのか」わからない

ジュニアエンジニアのインプットとアウトプットをどう繋げていくのかわからない

学習したこと(インプット)をどうまとめるのか知りたい

学習をまとめた成果物をどこに置いておけばよいのか知りたい

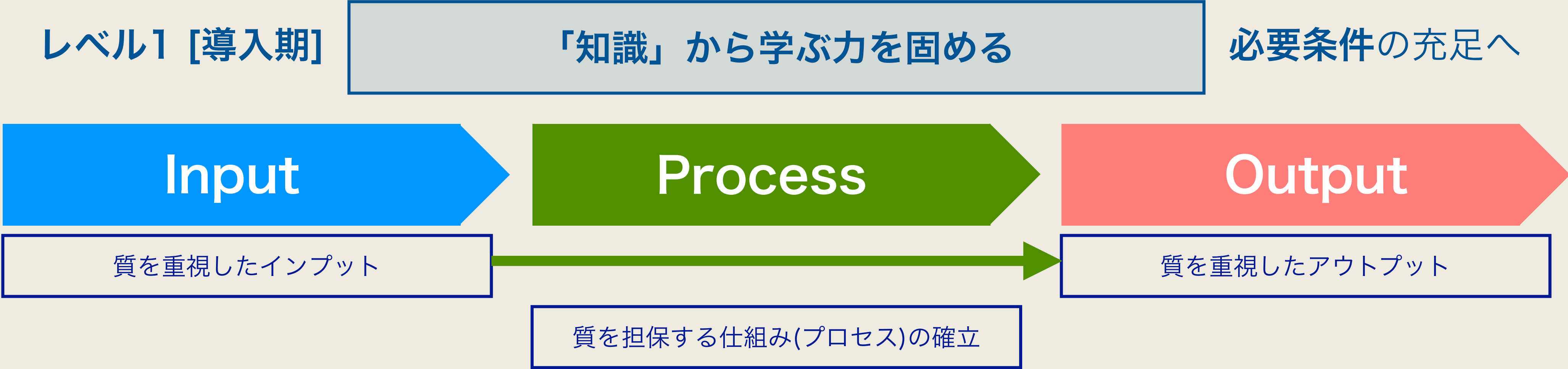
ジュニアエンジニアにどんなアウトプットが求められるのかわからない

ジュニアエンジニアがアウトプットすることの意味を知りたい

ジュニアエンジニアがアウトプットした先になにがあるのか知りたい

レベル1. 導入期

ジュニアエンジニア 導入期の成長工程



エンジニアとして活動するために「言語化された知識を学ぶ力」が最初に求められる

学習能力の確立

レベル1. 導入期

レベル1のインプットでの悩みを整理してみました

レベル1 [導入期]

「知識」から学ぶ力を固める

Input

学習能力の確立

タイプ重視で学ぼう

質を重視したインプット

「何を学ばいいのか」 わからない

学習の対象領域を明確にするとよい

タイプの良い学習領域

「どのように学ばいいのか」 わからない

学習の手段を明確にするとよい

タイプの良い学習手段

「どこまで学ばなければいけないのか」 わからない

学習の範囲を明確にするとよい

タイプの良い学習範囲

質を重視して
効率良くインプットする

= タイパが良い

レベル1. 導入期

レベル1のアウトプットでの悩みを整理してみました

レベル1 [導入期]

「知識」から学ぶ力を固める

Output

学習能力の確立

資格試験に合格しよう

質を重視したアウトプット

ジュニアエンジニアにどんなアウトプットが求められるのかわからない

客観的な成果を明確にするとよい

資格試験の客観的成果

ジュニアエンジニアがアウトプットすることの意味を知りたい

客観的な意味を明確にするとよい

資格試験の意味

専門組織が
効率良く実力を評価

ジュニアエンジニアがアウトプットした先になにがあるのか知りたい

客観的な展望を示すとよい

資格試験後の展望

結果が明瞭

レベル1. 導入期

レベル1のプロセスでの悩みを整理してみました

レベル1 [導入期]

「知識」から学ぶ力を固める

Process

学習能力の確立

メモや図を書く習慣を付けよう

質を担保する仕組み(プロセス)の確立



ジュニアエンジニアのインプットとアウトプットをどう繋げていくのかわからない

繋げ方を明確にするとよい

学習と合格を最短で繋ぐ

学習したこと(インプット)をどうまとめるのか知りたい

まとめる手法を明確にするとよい

学習の歩留まりを良くする

エンジニアは
「歩留まりを重視」する

学習をまとめた成果物をどこに置いておけばよいのか知りたい

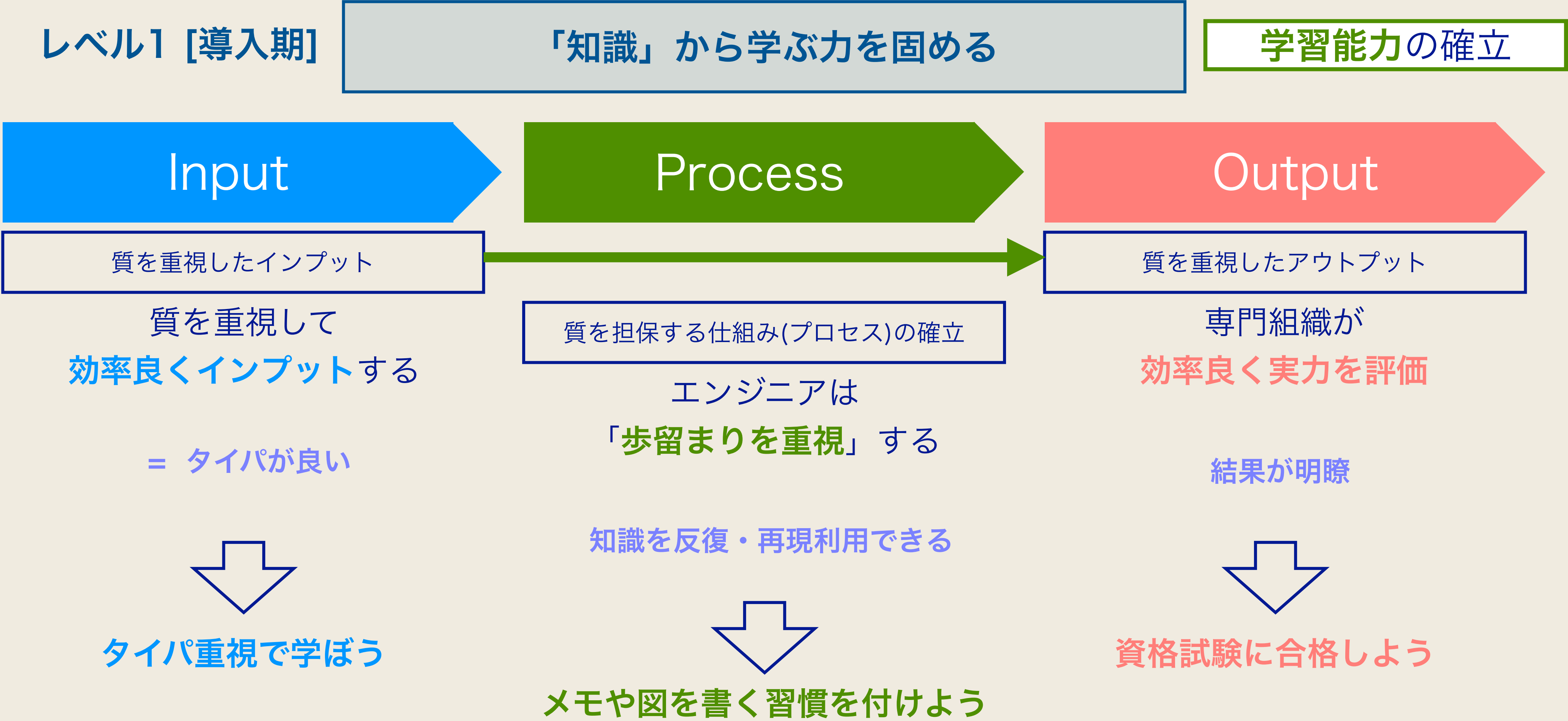
成果物の置き方を明確にするとよい

試験後の反復・再現利用

知識を反復・再現利用できる

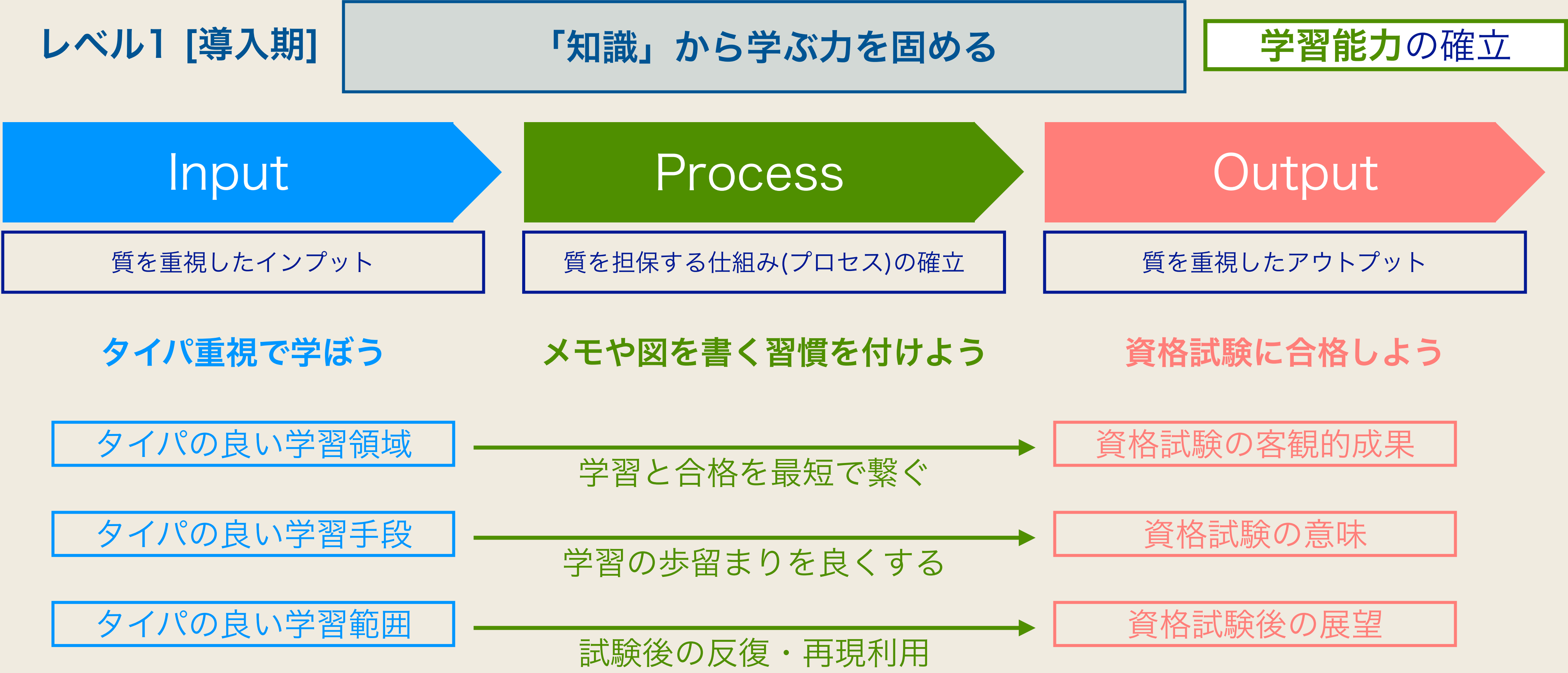
レベル1. 導入期

ジュニアエンジニア 導入期の成長工程 (まとめ)



レベル1. 導入期

ジュニアエンジニア導入期の全体像



レベル1. 導入期

学習能力の確立

タイパ重視で学ぼう

質を重視して
効率良くインプットする

= タイパが良い

学習の対象領域を明確にするとよい

タイパの良い学習領域

知識には3つの領域がある

明確に分けて理解する

社会知識

社会一般の知識
情報が豊富。広く使える。

専門知識

専門領域の知識
情報が限定的。業界業種で使える。

業務知識

企業内の知識
情報が希少。企業内のみで使える。

学習の手段を明確にするとよい

タイパの良い学習手段

初期学習は、資格試験が最も効率が良い

プロが作成したシラバスと問題

社会知識の資格試験

日商や各種資格団体

専門知識の資格試験

情報処理試験
ベンダー資格試験

学習の範囲を明確にするとよい

タイパの良い学習範囲

資格対策本や基本書の範囲を学習する

プロが言語化された必須項目を凝縮

資格試験の対策本

最初期の学習に最適

基本書

当初は資格試験の副読本として使う
合格後にきちんと学習すると効果大

レベル1. 導入期

学習能力の確立

資格試験に合格しよう

専門組織が
効率良く実力を評価

結果が明瞭

客観的な成果を明確にするとよい

資格試験の客観的成果

最初は「合格」にこだわる

最も客観的な成果として見える

合格まで

最短での合格を目指そう

合格後

合格で満足せずに復習して実力を補強しよう

上位資格へ

実力が合格に追い付いたら上位の資格を狙おう

客観的な意味を明確にするとよい

資格試験の意味

合格した資格の範囲の知識を証明できる

プロが求める知識を獲得した証明となる

無知ではない証明

その知識領域について「無知ではない」
ことを示せる

基礎が固まっている証明

ちゃんと復習していれば、「基礎が固
まっている証明」ともなる

客観的な展望を示すとよい

資格試験後の展望

合格してからがスタート

実務に向けた基礎体力ができたばかり

合格の直接効果

実務を学ぶときに手戻りやロスが少ない

長期的な効果

基礎が抜けていることによる怪我を
回避し続けることができる

レベル1. 導入期

学習能力の確立

メモや図を書く習慣を付けよう

エンジニアは
「歩留まりを重視」する

知識を反復・再現利用できる

繋げ方を明確にするとよい

学習と合格を最短で繋ぐ

最短での「合格」にこだわる

最初のエンジニアリング対象とする

メモを残す

ドキュメントレベル1

「自分のためのメモを残す」を実践しよう

復習しやすさを意識する

いつでも復習できることを重視してメモを
作成し、まめに復習しよう

メモのセルフレビュー

自分のメモでわかりにくいところは、どん
どん修正して、読みやすくしていこう

まとめる手法を明確にするとよい

学習の歩留まりを良くする

タイプのよいInputとOutputの接続

最初の「歩留まり」を意識する

編集ツールに習熟する

文書のためのエディタと、
図表のためのプレゼンテーションツールに
習熟しよう

再利用できるメモの作成

軽量マークアップ言語に習熟しよう

成果物の置き方を明確にするとよい

試験後の反復・再現利用

失われず、いつでも使えることにこだわる

「可用性」を意識して学習する

バージョン管理の導入

学習にも分散バージョン管理ツール
(DVCS)を活用しよう

リポジトリの配置と活用

リポジトリのクローンを、学習環境に
配置して、バージョン管理しながら
学習を進めよう

レベル1. 導入期

Input

Process

Output

レベル1: 導入期
必要条件の充足へ

タイプ重視で学ぼう

タイプの良い学習領域

タイプの良い学習手段

タイプの良い学習範囲

「知識」から学ぶ力を固める
メモや図を書く習慣を付けよう

学習と合格を最短で繋ぐ

学習の歩留まりを良くする

試験後の反復・再現利用

資格試験に合格しよう

資格試験の客観的成果

資格試験の意味

資格試験後の展望

学習能力の確立

Step2. ジュニアエンジニアの成長ステップ

レベル2. 確立期

ナビゲーション

Step1

エンジニアの成長サイクル

Step2

ジュニアエンジニアの成長ステップ

概要

レベル1 導入期

レベル2 確立期

Step3

レギュラーエンジニアの成長ステップ

概要

レベル1 導入期

レベル2 確立期

レベル2. 確立期

ジュニアエンジニアの成長途上の戸惑い

インプットしたことを忘れてしまわないか不安になる

インプットする時間が足りない

「レギュラーエンジニア」が見えてくるためのインプットを知りたい

学習や考察した内容を忘れても困らないためにどうすれば良いのか知りたい

ジュニアエンジニアのレベルで効率良く、深く理解するにはどうすればいいのか？

インプットとアウトプットを最短でつなげるにはどうすればいいのか

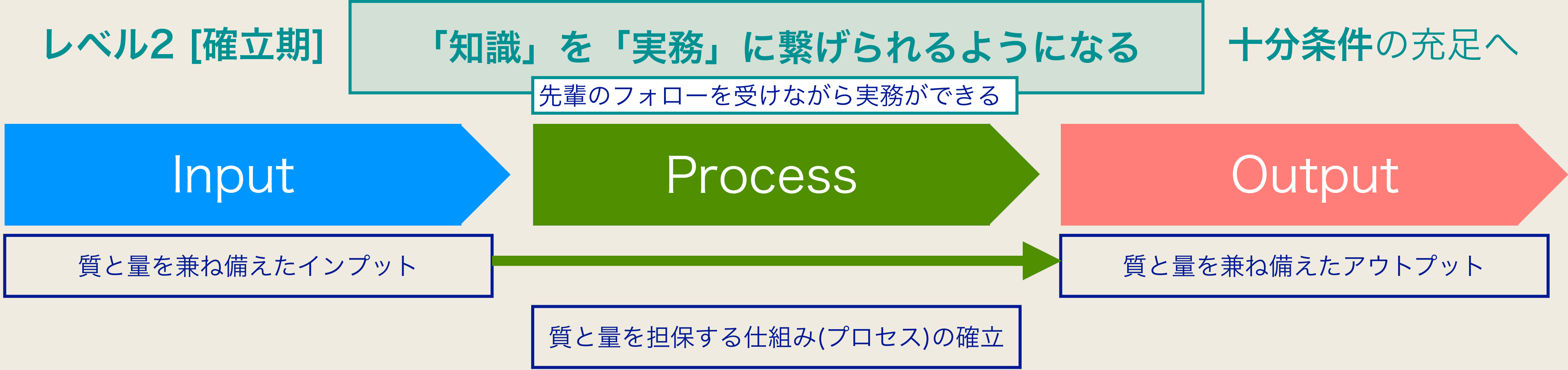
学んだことをどう活かせばいいのか知りたい

他のジュニアエンジニアと同じことをしていていいのか焦る

「レギュラーエンジニア」が見えてくるためのアウトプットを知りたい

レベル2. 確立期

ジュニアエンジニア 確立期の成長工程



学んだ「知識」を、実際の業務に紐付けられる能力が求められる。

読解力の確立

レベル2. 確立期

レベル2のインプットでの悩みを整理してみました

レベル2 [確立期]

「知識」を「実務」に繋がられるようになる

先輩のフォローを受けながら実務ができる

Input

読解力の確立

読解力を付けよう

質と量を兼ね備えたインプット

インプットしたことを忘れてしまわないか不安になる

忘れることを前提にするとよい

忘れてもインプット

インプットする時間が足りない

なぜ時間が足りないのかを明確にするとよい

時間が無くてもインプット

質を維持しつつ
量をこなしていく

「レギュラーエンジニア」が見えてくるためのインプットを知りたい

受動的に学ぶだけでは気付けないことがある

能動的なインプット

= 読書量が大幅に増える

レベル2. 確立期

レベル2のアウトプットでの悩みを整理してみました

レベル2 [確立期]

「知識」を「実務」に繋がられるようになる

先輩のフォローを受けながら実務ができる

Output

読解力の確立

知識を実務に繋げていこう

質と量を兼ね備えたアウトプット



学んだことをどう活かせばいいのか知りたい

学んだことをアウトプットに直結するとよい

正確なアウトプット

他のジュニアエンジニアと同じことをしていていいのか焦る

常に期限前にアウトプットするとよい

期限を厳守してアウトプット

学んだ知識が
本当に役に立つのか実践

「レギュラーエンジニア」が見えてくるためのアウトプットを知りたい

実務を意識したアウトプットをするとよい

実務に直結するアウトプット

徐々に実践が必要

レベル2. 確立期

レベル2のプロセスでの悩みを整理してみました

レベル2 [確立期]

「知識」を「実務」に繋がられるようになる

先輩のフォローを受けながら実務ができる

Process

読解力の確立

学習プロセスを構築しよう

質と量を担保する仕組み(プロセス)の確立



学習や考察した内容を忘れても困らないためにどうすれば良いのか知りたい

忘れても困らないようにするとよい

暗記ではなく理解

ジュニアエンジニアのレベルで効率良く、深く理解するにはどうすればいいのか？

「知識を積み上げる」感覚を大切にするとよい

業務の歩留まりを良くする

エンジニアは
「歩留まりを重視」する

インプットとアウトプットを最短でつなげるにはどうすればいいのか

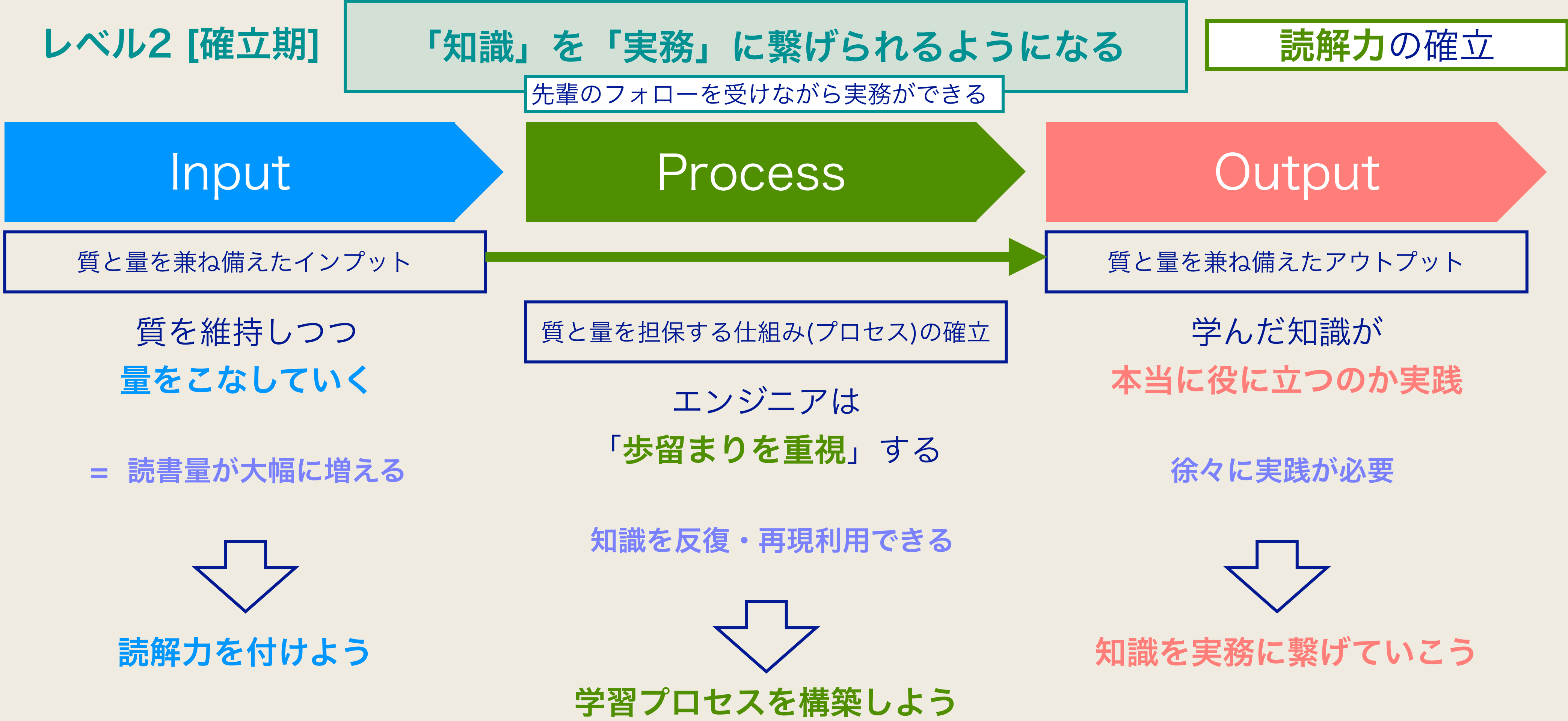
実務を前提に具体的にイメージするとよい

実務を前提に学ぶ

知識を反復・再現利用できる

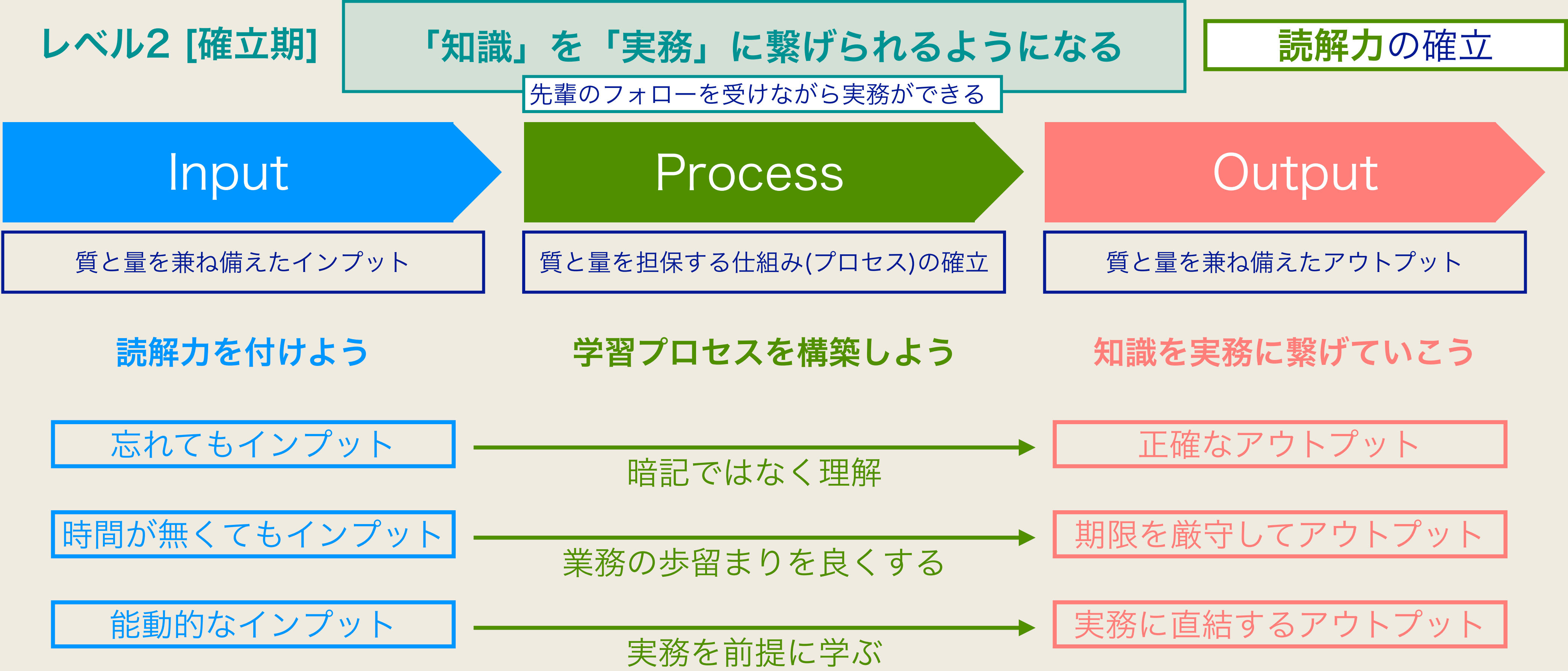
レベル2. 確立期

ジュニアエンジニア 確立期の成長工程 (まとめ)



レベル2. 確立期

ジュニアエンジニア確立期の全体像



レベル2. 確立期

読解力とは

読解

文章を読み，その内容を理解すること。

(出典: スーパー大辞林)

読解力

語彙力

言葉の意味を正確に読み取る力

解釈力

文章の構造を把握する力

速読力

文章を時間内に読み終える力

評価力

読み取った文章を評価し、判断する力

レベル2. 確立期

エンジニアの「読解力」の育成

エンジニアらしい表現 (参考)

語彙力

テクニカルタームを正確に理解する

ライブラリを強化する

解釈力

物事を構造(集合やツリー)で理解する

パーサーを強化する

速読力

読書量を増やし、文章に慣れていく

スループットを上げる

評価力

場数を踏んで、成功や失敗を重ねていく

「学習」する (AIっぽく)

レベル2. 確立期

Input

Process

Output

「知識」から学ぶ力を固める

学習能力の確立

レベル1: 導入期
必要条件の充足へ

タイパ重視で学ぼう

タイパの良い学習領域

タイパの良い学習手段

タイパの良い学習範囲

メモや図を書く習慣を付けよう

学習と合格を最短で繋ぐ

学習の歩留まりを良くする

試験後の反復・再現利用

資格試験に合格しよう

資格試験の客観的成果

資格試験の意味

資格試験後の展望

「知識」を「実務」に繋がられるようになる

レベル2: 確立期
十分条件の充足へ

多読で読解力を付けよう

忘れてもインプット

時間が無くてインプット

能動的なインプット

自分なりの学習プロセスを構築しよう

暗記ではなく理解

業務の歩留まりを良くする

実務を前提に学ぶ

知識を実務に繋げていこう

正確なアウトプット

期限を厳守してアウトプット

実務に直結するアウトプット

読解力の確立

Step2. ジュニアエンジニアの成長ステップ

まとめ

まとめ: ジュニアエンジニアの成長ステップ

Input

Process

Output

レベル1: 導入期
必要条件の充足へ

質を重視したインプット

タイプ重視で学ぼう

質を担保する仕組み(プロセス)の確立

メモや図を書く習慣を付けよう

学習能力の確立

質を重視したアウトプット

資格試験に合格しよう

レベル2: 確立期
十分条件の充足へ

質と量を兼ね備えたインプット

多読で読解力を付けよう

質と量を担保する仕組み(プロセス)の確立

自分なりの学習プロセスを構築しよう

読解力の確立

質と量を兼ね備えたアウトプット

知識を実務に繋げていこう

レベル3: 成熟期
オプション
(余裕があれば)

他人と差が付くインプット

資格試験を合格目的以外にも活用しよう

他人と差が付く機会との出会い

相談できる先達を見つけよう

他人と差が付くアウトプット

自分が困ったことを記録しておこう

まとめ: ジュニアエンジニアの成長ステップ

Input

Process

Output

レベル1: 導入期
必要条件の充足へ

「知識」から学ぶ力を固める

タイプ重視で学ぼう

タイプの良い学習領域

タイプの良い学習手段

タイプの良い学習範囲

学習能力の確立

メモや図を書く習慣を付けよう

学習と合格を最短で繋ぐ

学習の歩留まりを良くする

試験後の反復・再現利用

資格試験に合格しよう

資格試験の客観的成果

資格試験の意味

資格試験後の展望

レベル2: 確立期
十分条件の充足へ

「知識」を「実務」に繋がられるようになる

多読で読解力を付けよう

忘れてもインプット

時間が無くてインプット

能動的なインプット

読解力の確立

自分なりの学習プロセスを構築しよう

暗記ではなく理解

業務の歩留まりを良くする

実務を前提に学ぶ

知識を実務に繋げていこう

正確なアウトプット

期限を厳守してアウトプット

実務に直結するアウトプット

レベル3: 成熟期
オプション
(余裕があれば)

レギュラー以降の伸び代を準備しておく

資格試験を合格目的以外にも活用しよう

相談できる先達を見つけよう

自分が困ったことを記録しておこう

Step3. レギュラーエンジニアの成長ステップ

レギュラーエンジニアの成長ステップ

ジュニアエンジニアから「レギュラーエンジニア」に昇格後
一人前と認められるために必要な過程
はどのようなものになるのでしょうか？

ステップバイステップで考えてみてください。

Step3. レギュラーエンジニアの成長ステップ

「レギュラーエンジニア」の概要

ナビゲーション

Step1

エンジニアの成長サイクル

Step2

ジュニアエンジニアの成長ステップ

概要

レベル1 導入期

レベル2 確立期

Step3

レギュラーエンジニアの成長ステップ

概要

レベル1 導入期

レベル2 確立期

「レギュラーエンジニア」の概要

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

「レギュラーエンジニア」の概要

レギュラーエンジニアに何が求められるのかわからない

ようやくレギュラー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

「レギュラー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

どこまでやればレギュラーエンジニアとして認められるのかわからない

エンジニアとして困らないために、**レギュラー時代にやるべきこと**を知りたい

レギュラー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

「レギュラーエンジニア」の概要

ようやくレギュラー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

借り物の「知識」に自分の「経験」を肉付けして「自分独自の知識」にしていく必要がある

知識や経験について「自分の言葉」で語れるようになることが求められる

上司や先輩の指示通りの仕事が一人でできるようになる

借り物ではない「自分の言葉」で報告を(口頭、ドキュメントで)できる。

論述力の確立

レベル1 [導入期]

知識を「借り物」から「自分のもの」にしていく

まだ
「半人前」扱い

「レギュラーエンジニア」の概要

「レギュラー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

自分の専門領域(できること)と、専門外(できないこと)の線引きができるようになる必要がある

「なんでもできる」は、明確な専門性が何も無くて「プロとしてはなにもできない」のと同じ事

専門外の事を、「専門家としてはできない」と言えるようになったら一人前

「チーム内の個人」として、専門的な活動や成果が認知されるようになる

知識と経験の両立

知識と経験を両立していく過程で、自分の得意領域と境界線(できる、できない)が見えてくる

レベル2 [確立期]

「できること」と「できないこと」を明確にする

これができると
「一人前」扱い

「レギュラーエンジニア」の概要

エンジニアとして困らないために、レギュラー時代にやるべきことを知りたい

レギュラー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

他のレギュラーエンジニアがあまりやらないような**インプット**

他のレギュラーエンジニアがあまりやらないような**プロセス**

他のレギュラー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]

ミドル以降の伸び代を準備しておく

昇格待ち相当

「レギュラーエンジニア」の概要

レギュラーエンジニアとしての必要条件、十分条件、オプション

レベル1 [導入期]

知識を「借り物」から「自分のもの」にしていく

まだ半人前

必要条件の充足へ

借り物の「知識」に自分の「経験」を肉付けして「自分独自の知識」にしていく

論述力の確立

レベル2 [確立期]

「できること」と「できないこと」を明確にする

一人前相当

十分条件の充足へ

自分の専門領域(できること)と、専門外(できないこと)の線引きができるようになる

知識と経験の両立

レベル3 [成熟期]

ミドル以降の伸び代を準備しておく

昇格待ち相当

オプション
(余裕があれば)

レギュラー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

「レギュラーエンジニア」の概要

レギュラーエンジニアの成長マトリクス

	Input	Process	Output
レベル1: 導入期 必要条件の充足へ	質を重視したインプット	質を担保する仕組み(プロセス)の確立	質を重視したアウトプット
	上司や先輩の指示通りの仕事が一人で行えるようになる		論述力の確立
レベル2: 確立期 十分条件の充足へ	質と量を兼ね備えたインプット	質と量を担保する仕組み(プロセス)の確立	質と量を兼ね備えたアウトプット
	専門外の事を、「専門家としてはできない」と言えるようになったら一人前		知識と経験の両立
レベル3: 成熟期 オプション (余裕があれば)	他人と差が付くインプット	他人と差が付く機会との出会い	他人と差が付くアウトプット
	他のエンジニアとの差別化を図るための準備をする		

「レギュラーエンジニア」の概要

レギュラーエンジニアは、**正規の戦力**として扱われる

レベル1 [導入期]

知識を「借り物」から「自分のもの」にしていく

上司や先輩の指示通りの仕事が一人のできるようになる

必要条件の充足へ

論述力の確立

レベル2 [確立期]

「できること」と「できないこと」を明確にする

専門外の事を、「専門家としてはできない」と言えるようになったら一人前

十分条件の充足へ

知識と経験の両立

レベル3 [成熟期]

ミドル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

オプション (余裕があれば)

Step3. レギュラーエンジニアの成長ステップ

レベル1. 導入期

ナビゲーション

Step1

エンジニアの成長サイクル

Step2

ジュニアエンジニアの成長ステップ

概要

レベル1 導入期

レベル2 確立期

Step3

レギュラーエンジニアの成長ステップ

概要

レベル1 導入期

レベル2 確立期

レベル1. 導入期

レギュラーエンジニアの最初の戸惑い

レギュラーエンジニアにどんなインプットが求められるのかわからない

ジュニアエンジニアのインプットと何が違うのか知りたい

レギュラーエンジニアとして「どこまで学ばなければいけないのか」わからない

レギュラーエンジニアのインプットとアウトプットをどう繋げていくのかわからない

ジュニアエンジニアのプロセスと何が違うのか知りたい

エンジニアらしく反復再現性を高めていく方法を知りたい

レギュラーエンジニアにどんなアウトプットが求められるのかわからない

ジュニアエンジニアのアウトプットと何が違うのか知りたい

レギュラーエンジニアがアウトプットした先になにがあるのか知りたい

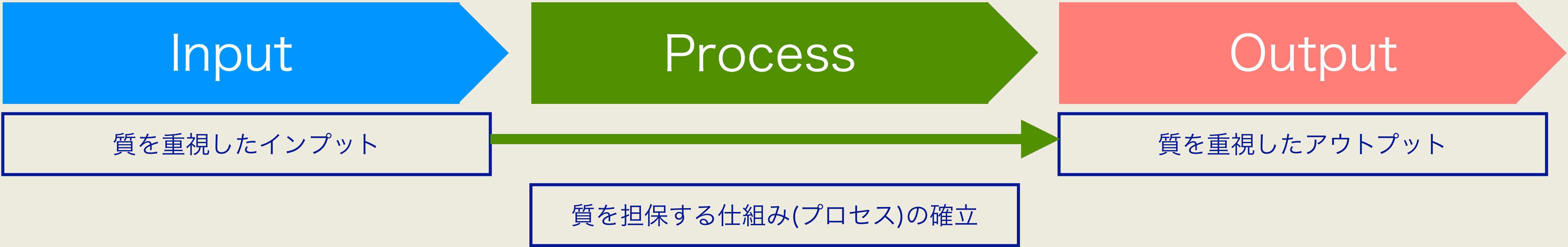
レベル1. 導入期

レギュラーエンジニア 導入期の成長工程

レベル1 [導入期]

知識を「借り物」から「自分のもの」にしていく

論述力の確立



借り物の「知識」に自分の「経験」を肉付けして「自分独自の知識」にしていく必要がある

レベル1. 導入期

レベル1のインプットでの悩みを整理してみました

レベル1 [導入期] 知識を「借り物」から「自分のもの」にしていく

Input

論述力の確立

自分の経験や失敗から学ぼう

質を重視したインプット



レギュラーエンジニアにどんなインプットが求められるのかわからない

「知識」だけが対象ではなくなる

ジュニアエンジニアのインプットと何が違うのか知りたい

「失敗した経験」が価値になる

レギュラーエンジニアとして「どこまで学ばなければいけないのか」わからない

自律的な学びが求められるようになる

経験から学びを得る

失敗から学びを得る

学びの間口を広げる

「自分の経験」という
良質なインプットで
知識を強化する

成功・失敗の両方から学ぶ

レベル1. 導入期

レベル1のアウトプットでの悩みを整理してみました

レベル1 [導入期]

知識を「借り物」から「自分のもの」にしていく

Output

論述力の確立

成功も失敗もアウトプットしよう

質を重視したアウトプット



レギュラーエンジニアにどんなアウトプットが求められるのかわからない

経験に裏打ちされた**個人の成果**が求められる

自分の名前で成果を出す

ジュニアエンジニアのアウトプットと何が違うのか知りたい

失敗した事実もアウトプット対象となる

失敗もアウトプットする

一人前のエンジニアとしての
「あたりまえ」ができる

レギュラーエンジニアがアウトプットした先になにがあるのか知りたい

エンジニアならできることが当たり前ができる

適量の成功と失敗を経験する

成功も失敗も経験している

レベル1. 導入期

レベル1のプロセスでの悩みを整理してみました

レベル1 [導入期]

知識を「借り物」から「自分のもの」にしていく

Process

論述力の確立

相手にわかりやすく説明しよう

質を担保する仕組み(プロセス)の確立



レギュラーエンジニアのインプットとアウトプットをどう繋げていくのかわからない

「経験」を言語化することが成果に繋がる

経験と成果を最短で繋ぐ

ジュニアエンジニアのプロセスと何が違うのか知りたい

責任ある説明を求められるようになる

「説明責任」を意識する

エンジニアは

「歩留まりを重視」する

エンジニアらしく反復再現性を高めていく方法を知りたい

「経験」を言語化することで再現性が高まる

知識と経験の反復・再現利用

知識と経験を反復・再現利用できる

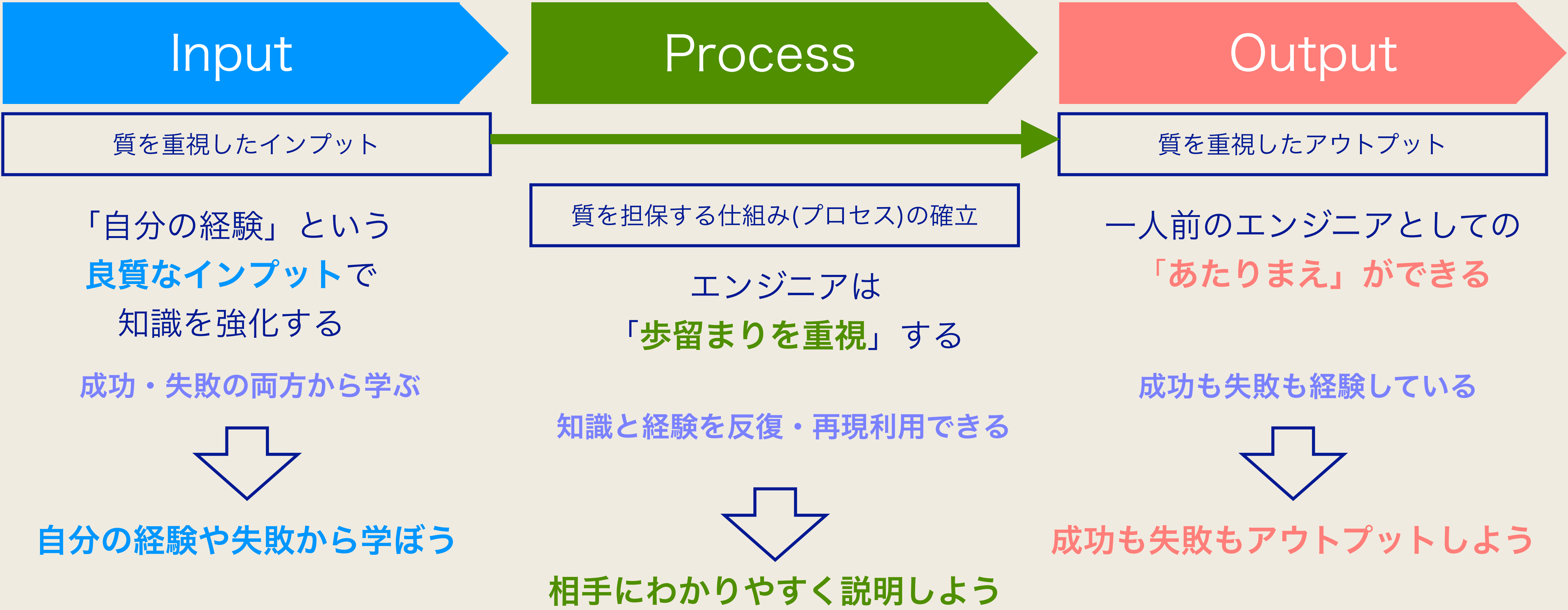
レベル1. 導入期

レギュラーエンジニア 導入期の成長工程 (まとめ)

レベル1 [導入期]

知識を「借り物」から「自分のもの」にしていく

論述力の確立



レベル1. 導入期

レギュラーエンジニア導入期の全体像

レベル1 [導入期]

知識を「借り物」から「自分のもの」にしていく

論述力の確立

Input

質を重視したインプット

自分の経験や失敗から学ぼう

経験から学びを得る

失敗から学びを得る

学びの間口を広げる

Process

質を担保する仕組み(プロセス)の確立

相手にわかりやすく説明しよう

経験と成果を最短で繋ぐ

「説明責任」を意識する

知識と経験の反復・再現利用

Output

質を重視したアウトプット

成功も失敗もアウトプットしよう

自分の名前で成果を出す

失敗もアウトプットする

適量の成功と失敗を経験する

レベル1. 導入期

論述力とは

論述

意見や考えを筋道立てて述べること。また、その述べたもの。

(出典: スーパー大辞林)

論述力

語彙力

言葉の意味を正確に読み取る力

論理力

正しい論理構造(筋道)を構築できる力

速記力

文章を時間内に記述し終える力

要約力

作成する文章を適切な分量に要約し、伝える力

レベル1. 導入期

エンジニアの「論述力」の育成

エンジニアらしい表現 (参考)

語彙力

テクニカルタームを正確に使用する

ライブラリを強化する

論理力

物事を構造(集合やツリー)で表現する

ロジックを強化する

速記力

文書の作成量を増やし、文書作成に慣れていく

スループットを上げる

要約力

上司や先輩のレビューを受けて強化していく

ダイジェストする

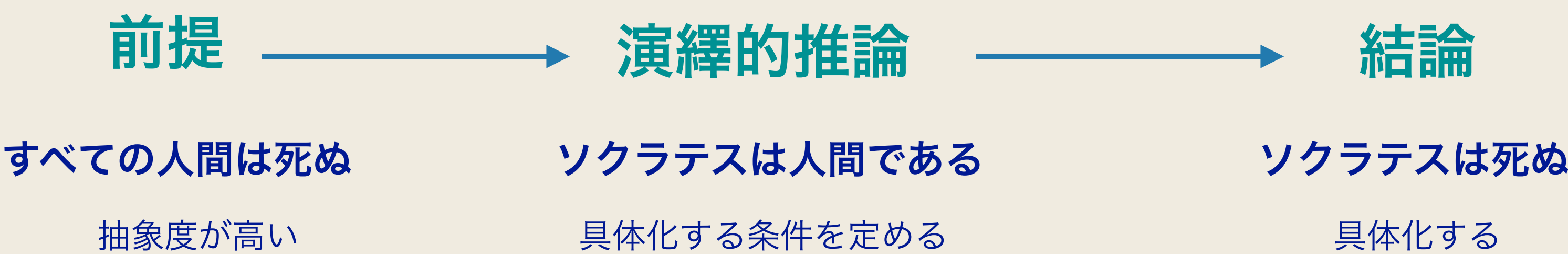
レベル1. 導入期

エンジニアリングに必要な「演繹」という考え方

演繹

諸前提から論理の規則にしたがって
必然的に結論を導き出すこと。

演繹の基本構造



演繹では、前提が同一であれば、必ず結論は同じになる (必然的)

レベル1. 導入期

「演繹」と「帰納」

演繹

前提

すべての人間は死ぬ

抽象度が高い

演繹的推論

ソクラテスは人間である

具体化する条件を定める

結論

ソクラテスは死ぬ

具体化する

演繹では、前提が同一であれば、必ず結論は同じになる (必然的)

帰納

結論

ユーザーは申請スキルが低い。(仕方ない)

ユーザーに申請ガイドが必要。(改善すべき)

抽象化する

帰納的推論

Aさんの推論

Bさんの推論

抽象化する条件をさがす

具体例

ユーザ1は申請に時間が掛かる。

ユーザ2は申請にミスが多い。

ユーザ3は申請方法を理解していない。

具体度が高い

帰納では、結論は同じになるとは限らない (蓋然的)

レベル1. 導入期

「演繹」と「帰納」の使いどころ

演繹

前提



演繹的推論



結論

ほとんど全てのビジネス文章は、**演繹で記述**する

前提が変わらなければ結論が同一であるため。(反復・再現性)

帰納

結論



帰納的推論



具体例

文章のうち「分析」や「調査」のセクションのみ「帰納」を使う

分析文書や、調査文書も、その文書構造自体は演繹になる

エンジニアの書く文書は、演繹的推論形式である

レベル1. 導入期

Input

Process

Output

知識を「借り物」から「自分のもの」にしていく

自分の経験や失敗から学ぼう

経験から学びを得る

失敗から学びを得る

学びの間口を広げる

相手にわかりやすく説明しよう

経験と成果を最短で繋ぐ

「説明責任」を意識する

知識と経験の反復・再現利用

成功も失敗もアウトプットしよう

自分の名前で成果を出す

失敗もアウトプットする

適量の成功と失敗を経験する

論述力の確立

レベル1: 導入期
必要条件の充足へ

Step3. レギュラーエンジニアの成長ステップ

レベル2. 確立期

ナビゲーション

Step1

エンジニアの成長サイクル

Step2

ジュニアエンジニアの成長ステップ

概要

レベル1 導入期

レベル2 確立期

Step3

レギュラーエンジニアの成長ステップ

概要

レベル1 導入期

レベル2 確立期

レベル2. 確立期

レギュラーエンジニアの成長途上の戸惑い

不得意なことのインプットに時間がかかってしまうがどうすればいいのか？

得意なことのインプットで差別化するにはどうすればいいのか？

「ミドルエンジニア」が見えてくるためのインプットを知りたい

ドキュメントを書くように言われることが増えたがどうすればいいのか？

レギュラーエンジニアのレベルで効率良く、深く理解するにはどうすればいいのか？

インプットとアウトプットを良質につなげるにはどうすればいいのか

不得意なことをどうアウトプットしていけばいいのか？

得意なことをどうアウトプットしていけばいいのか？

「ミドルエンジニア」が見えてくるためのアウトプットを知りたい

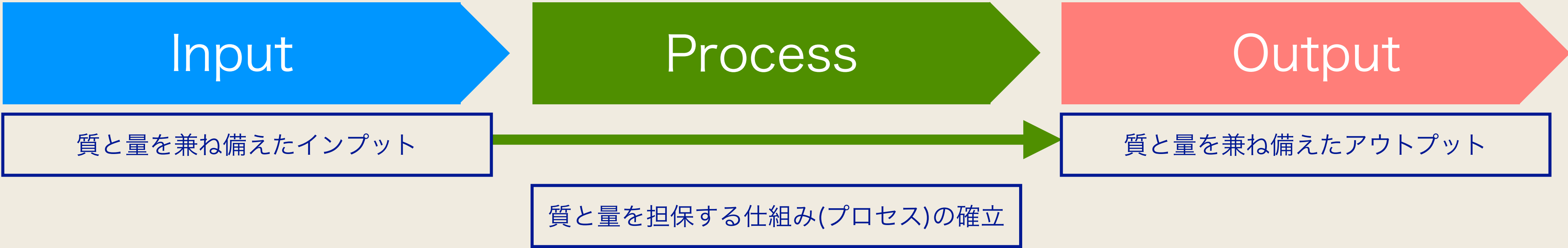
レベル2. 確立期

レギュラーエンジニア 確立期の成長工程

レベル2 [確立期]

「できること」と「できないこと」を明確にする

知識と経験の両立



自分の専門領域(できること)と、専門外(できないこと)の線引きができるようになる必要がある

レベル2. 確立期

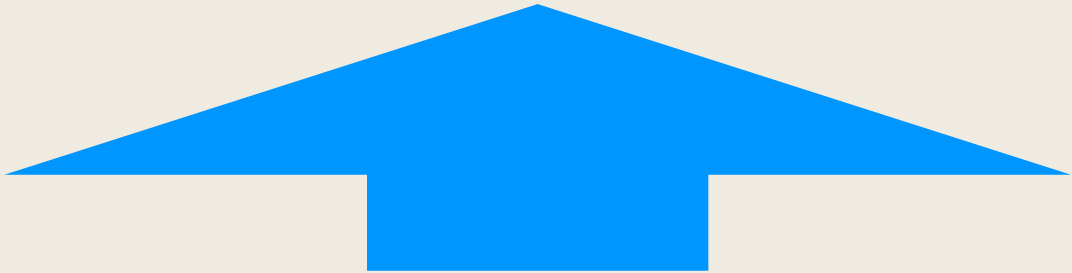
レベル2のインプットでの悩みを整理してみました

レベル2 [確立期] 「できること」と「できないこと」を明確にする

Input

知識と経験の両立 タイパの悪いインプットに時間を使おう

質と量を兼ね備えたインプット



不得意なことのインプットに時間がかかってしまうがどうすればいいのか？

やらずに済むならやらない選択をするとよい

不得意を自覚する

得意なことのインプットで差別化するにはどうすればいいのか？

他人がやらないところまでやると差が付く

得意を自覚する

業務影響の無い範囲で
インプットとの相性を測る

「ミドルエンジニア」が見えてくるためのインプットを知りたい

インプットの相性から自分の特性がわかる

自分の特性を見極める

自分の特性でのタイパを知る

レベル2. 確立期

レベル2のアウトプットでの悩みを整理してみました

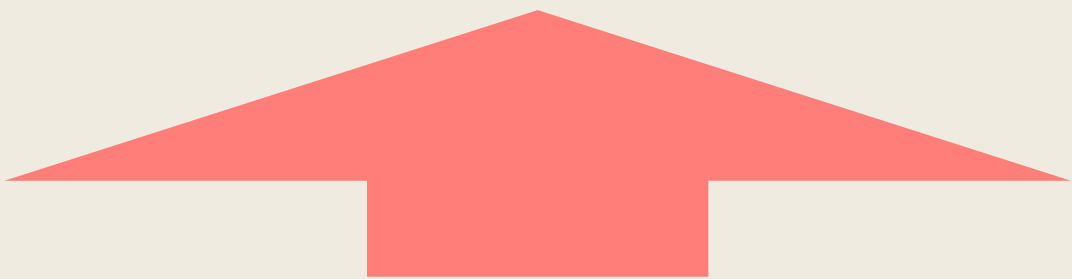
レベル2 [確立期] 「できること」と「できないこと」を明確にする

Output

知識と経験の両立

得意と不得意を意識してアウトプットしよう

質と量を兼ね備えたアウトプット



不得意なことをどうアウトプットしていけばいいのか？

不得意なことは無理なくアウトプットするとよい

不得意は穏便にアウトプット

得意なことをどうアウトプットしていけばいいのか？

得意なことは積極的にアウトプットするとよい

得意は強気でアウトプット

業務影響の無い範囲で
アウトプットとの相性を測る

「ミドルエンジニア」が見えてくるためのアウトプットを知りたい

自分の相性や強みを明確にしていく

自分の特性を磨く

自分の特性での
パフォーマンスの出し方を知る

レベル2. 確立期

レベル2のプロセスでの悩みを整理してみました

レベル2 [確立期]

「できること」と「できないこと」を明確にする

Process

知識と経験の両立

説明のための論理スキームを構築しよう

質と量を担保する仕組み(プロセス)の確立



ドキュメントを書くように言われることが増えたがどうすればいいのか？

「書くことの場数」を踏むとよい

不得意なものほどドキュメント化する

レギュラーエンジニアのレベルで効率良く、深く理解するにはどうすればいいのか？

人に説明できるようになるとよい

先輩や上司に説明できるようになる

エンジニアは
「歩留まりを重視」する

インプットとアウトプットを良質につなげるにはどうすればいいのか

「ロジックを考える場数」を踏むとよい

呼吸するようにロジック化する

知識や経験
を反復・再現利用できる

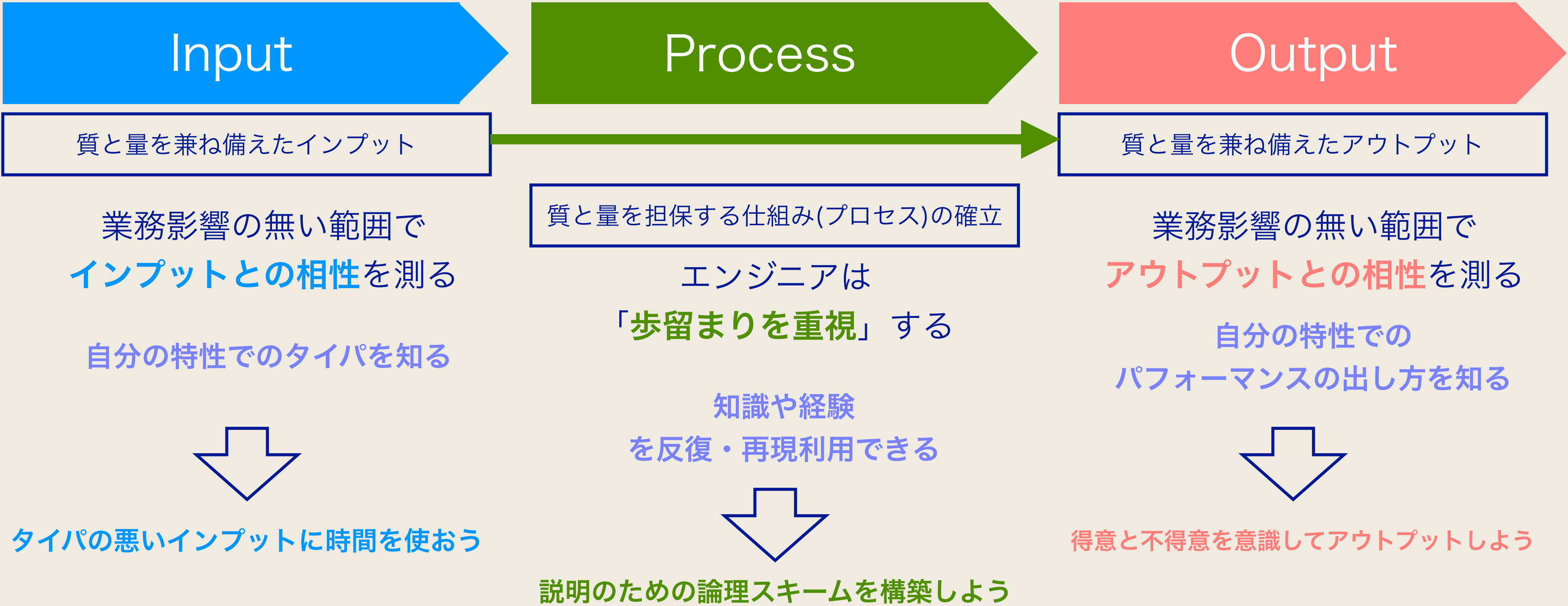
レベル2. 確立期

レギュラーエンジニア 確立期の成長工程 (まとめ)

レベル2 [確立期]

「できること」と「できないこと」を明確にする

知識と経験の両立



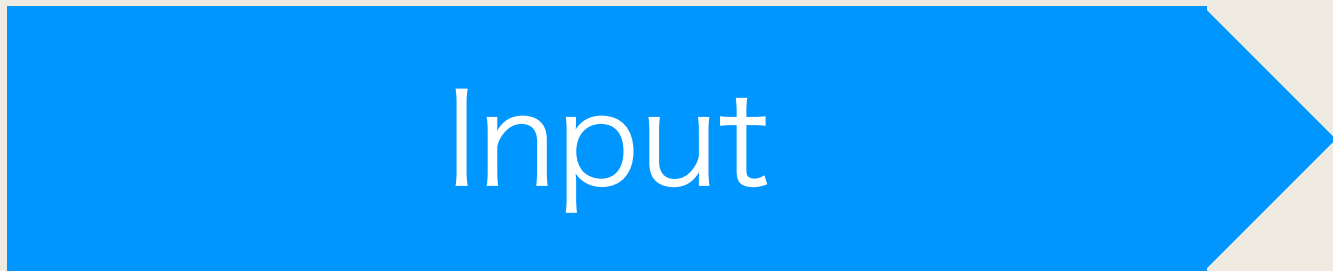
レベル2. 確立期

レギュラーエンジニア確立期の全体像

レベル2 [確立期]

「できること」と「できないこと」を明確にする

知識と経験の両立



質と量を兼ね備えたインプット

タイパの悪いインプットに時間を使おう

不得意を自覚する

得意を自覚する

自分の特性を見極める



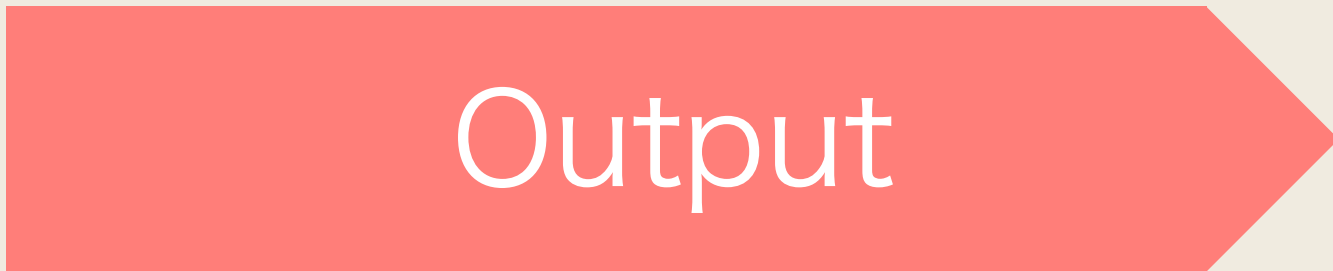
質と量を担保する仕組み(プロセス)の確立

説明のための論理スキームを構築しよう

不得意なものほどドキュメント化する

先輩や上司に説明できるようになる

呼吸するようにロジック化する



質と量を兼ね備えたアウトプット

得意と不得意を意識してアウトプットしよう

不得意は穏便にアウトプット

得意は強気でアウトプット

自分の特性を磨く

レベル2. 確立期

Input

Process

Output

知識を「借り物」から「自分のもの」にしていく

論述力の確立

自分の経験や失敗から学ぼう

相手にわかりやすく説明しよう

成功も失敗もアウトプットしよう

経験から学びを得る

経験と成果を最短で繋ぐ

自分の名前で成果を出す

失敗から学びを得る

「説明責任」を意識する

失敗もアウトプットする

学びの間口を広げる

知識と経験の反復・再現利用

大量の成功と失敗を経験する

「できること」と「できないこと」を明確にする

タイプの悪いインプットに時間を使おう

説明のための論理スキームを構築しよう

得意と不得意を意識してアウトプットしよう

不得意を自覚する

不得意なものほどドキュメント化する

不得意は穏便にアウトプット

得意を自覚する

先輩や上司に説明できるようになる

得意は強気でアウトプット

自分の特性を見極める

呼吸するようにロジック化する

自分の特性を磨く

知識と経験の両立

レベル1: 導入期

必要条件の充足へ

レベル2: 確立期

十分条件の充足へ

Step3. レギュラーエンジニアの成長ステップ

まとめ

まとめ: レギュラーエンジニアの成長ステップ

Input

Process

Output

レベル1: 導入期
必要条件の充足へ

質を重視したインプット

自分の経験や失敗から学ぼう

質を担保する仕組み(プロセス)の確立

相手にわかりやすく説明しよう

論述力の確立

質を重視したアウトプット

成功も失敗もアウトプットしよう

レベル2: 確立期
十分条件の充足へ

質と量を兼ね備えたインプット

タイパの悪いインプットに時間を使おう

質と量を担保する仕組み(プロセス)の確立

説明のための論理スキームを構築しよう

知識と経験の両立

質と量を兼ね備えたアウトプット

得意と不得意を意識してアウトプットしよう

レベル3: 成熟期
オプション
(余裕があれば)

他人と差が付くインプット

欲望の赴くままにインプットしよう

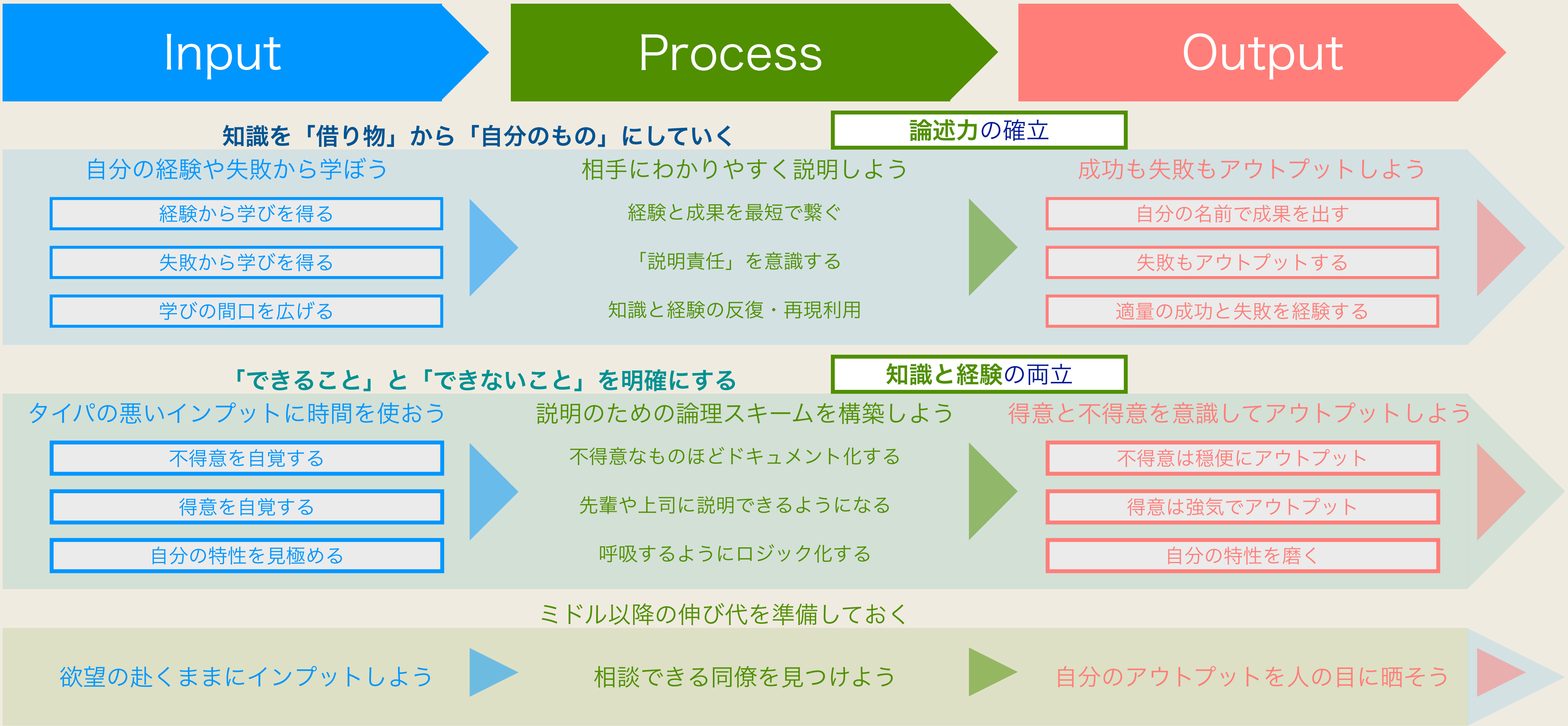
他人と差が付く機会との出会い

相談できる同僚を見つけよう

他人と差が付くアウトプット

自分のアウトプットを人の目に晒そう

まとめ: レギュラーエンジニアの成長ステップ



おわりに: ミドルエンジニアに進むその前に

おわりに: ミドルエンジニアに進むその前に

「説明責任」を果たす

ミドルエンジニアは、日々「説明責任」を果たすことが仕事になる

「できること」と「できないこと」を見極める

ミドルエンジニアは、「できること」に特化しているプロフェッショナルである

「得意」を仕事にする準備をする

ミドルエンジニアは、「得意」を磨き続けなければ生き残れない仕事である

まとめ

この講義で一番大切なこと

「初級エンジニア」は**歩留りの良い学び**で
反復再現性のある成長・育成ができる

この講義の全体ふりかえり

Step1. エンジニアの成長サイクル

エンジニアは成果だけでなく、自分自身の成長についても**反復再現的、持続的に実現する**



Step2. ジュニアエンジニアの成長ステップ

レベル1: 導入期

「知識」から学ぶ力を固める

学習能力の確立

レベル2: 確立期

「知識」を「実務」に繋がられるようになる

読解力の確立

Step3. レギュラーエンジニアの成長ステップ

レベル1: 導入期

知識を「借り物」から「自分のもの」にしていく

論述力の確立

レベル2: 確立期

「できること」と「できないこと」を明確にする

知識と経験の両立

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

エンジニアとして、自分自身の成長についても**反復再現的、持続的に実現**していこう



受講内容を活用して「実際の行動」を行おう

タイプ重視で学ぼう

メモや図を書く習慣を付けよう

資格試験に合格しよう

学習能力の確立

多読で読解力を付けよう

自分なりの学習プロセスを構築しよう

知識を実務に繋げていこう

読解力の確立

資格試験を合格目的以外にも活用しよう

相談できる先達を見つけよう

自分が困ったことを記録しておこう

全てのエンジニアに必要な基礎体力です。強みは強化しつつ、弱点を補強していきましょう

自分の経験や失敗から学ぼう

相手にわかりやすく説明しよう

成功も失敗もアウトプットしよう

論述力の確立

タイプの悪いインプットに時間を使おう

説明のための論理スキームを構築しよう

得意と不得意を意識してアウトプットしよう

知識と経験の両立

欲望の赴くままにインプットしよう

相談できる同僚を見つけよう

自分のアウトプットを人の目に晒そう

この講義から更にその先へ

第1回 運用エンジニアのキャリアを考えよう

第2回
(今回)

ジュニアエンジニア
レギュラーエンジニア

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
ジュニア、レギュラーエンジニアを育成する方

第3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