

エンジニアのためのドキュメンテーション入門 3

ドキュメントの創り方

～ 「書く」だけがドキュメンテーションではない

運用設計ラーニング

この講義の対象者

ジュニア～ミドルのエンジニア

- ・ドキュメントの書き方や保守の仕方について**知りたい方**
- ・ドキュメントの作成や保守に**迷いや悩みを持っている方**
- ・ドキュメント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**

エンジニアを指導・育成する立場(先輩や上司)の方

- ・後輩や部下のドキュメント指導・育成について**明確な展望が持っていない方**
- ・現在、後輩や部下のドキュメント指導・育成に**迷いや悩みを持っている方**
- ・後輩や部下のドキュメント指導・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**

この講座で得られるもの

- ・ ドキュメントの品質を高めるために必要な活動を理解することができる
- ・ ドキュメントのパフォーマンスをどう最大化するか知ることができる
- ・ ドキュメントの過不足を把握するためにやるべきことを知ることができる

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

この講座のロードマップ

序. 「無駄を避ける」ドキュメンテーション

Step1

1. ドキュメントの品質を担保する

「人が書く」=コストがかかる

ドキュメントのライフサイクル

誰が書くべきか

Step2

2. ドキュメントのパフォーマンスを最大化する

「最小の工数」で「最大の効果」

記述対象を「書きやすい」状態にする

「人手」以外に書かせる

Step3

3. ドキュメントの過不足を明確にする

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

序. 「無駄を避ける」ドキュメンテーション

～ いかに、不要・低品質なドキュメントを作らないで済みますか

ナビゲーション

序. 「無駄を避ける」ドキュメンテーション

Step1

1. ドキュメントの品質を担保する

「人が書く」=コストがかかる

ドキュメントのライフサイクル

誰が書くべきか

Step2

2. ドキュメントのパフォーマンスを最大化する

「最小の工数」で「最大の効果」

記述対象を「書きやすい」状態にする

「人手」以外に書かせる

Step3

3. ドキュメントの過不足を明確にする



「無駄を避ける」ドキュメンテーション

「効率の良い」ドキュメンテーション
が、できないのはなぜでしょうか？

「無駄を避ける」ドキュメンテーション

効率の悪いドキュメンテーション (例)

「どこに資料があるかわからない」ので、見付けられない、新たに作成している

「責任を取りたくない」ので、自分の作成した資料はできるだけ見せたくない

「オレオレ資料」が、多数存在するが、偏りや重複が激しい

新人が入る毎に、個々人の「用語集」が作られている

「無駄を避ける」ドキュメンテーション

なぜドキュメンテーションの効率が悪いのか？

人のリソースは有限であることを意識していない (重複の容認)

「責任」と「品質」が比例することを意識していない (オレオレ資料の容認)

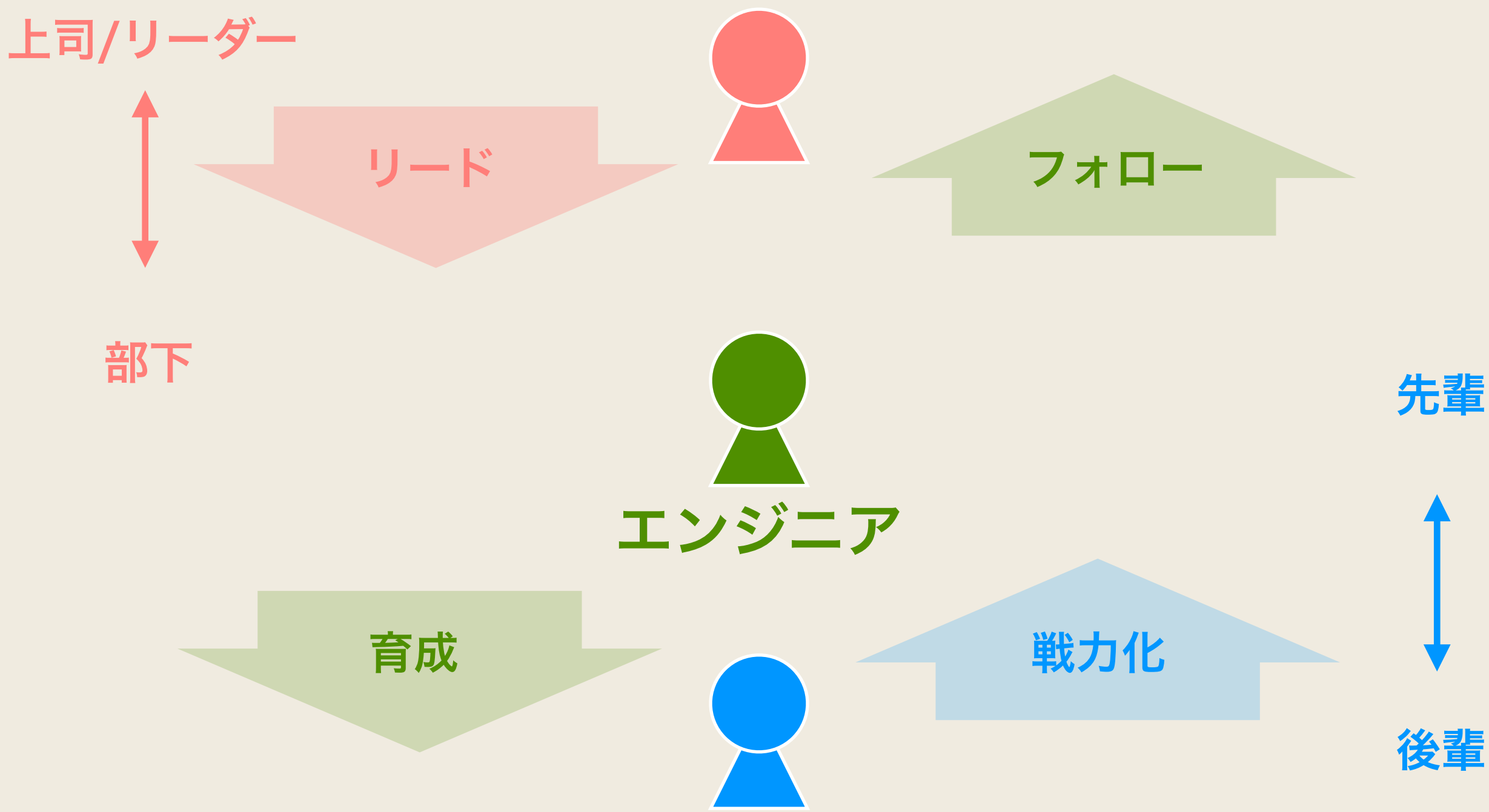
ドキュメントの過不足を意識していない (「とりあえず作る」の容認)

「無駄を避ける」ドキュメンテーション

復習: エンジニアと上司-部下、先輩-後輩の関係

エンジニアリング入門 第1回 Step2. 「エンジニアと上司-部下、先輩-後輩の関係」から

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



上司を上手に使う

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

上司や後輩に上手に使われる

上司や後輩との仕事がスムーズになる

後輩を上手に戦力化する

後輩の戦力化により自分の仕事が楽になる

「無駄を避ける」ドキュメンテーション

ドキュメンテーションで「人的リソースの最大活用」ができていない

人のリソースは有限であることを意識していない (重複の容認)



品質がバラバラな
ドキュメントが
散在・重複する

リソースの発散

書いたり書かなかったりする

ドキュメントの指導はしない

書いたり書かなかったりする

自分以外でも書けるドキュメントに注力したりする

書いたり書かなかったりする

あまり品質の高いドキュメントは書けない

「無駄を避ける」ドキュメンテーション

ドキュメンテーションで「責任」と「品質」を制御できていない

「責任」と「品質」が比例することを意識していない (オレオレ資料の容認)



他人の目に触れるものだから、
わかりやすく、長く使えるものを作ろう

有責任
高品質

組織として
ロスが少ない



自分しか使わないものだから、
自分がわかれば良い、長く使わないだろう

引き継ぎ不要の業務であれば妥当な場合もある

無責任
低品質

組織として
ロスが多い

「無駄を避ける」ドキュメンテーション

ドキュメンテーションで「過不足」が把握できていない

ドキュメントの過不足を意識していない（「とりあえず作る」の容認）



「どこに資料があるかわからない」ので、
見付けられない、新たに作成している

過剰なドキュメント

不足しているドキュメント

どんなドキュメントが必要で
現在どのドキュメントがどのような品質で存在するのか？

「無駄を避ける」ドキュメンテーション

ドキュメンテーションの成果を最大化するために

人のリソースは有限

いかに、人のリソースを浪費せずにドキュメントを作り続けるか？

「責任」と「品質」は比例

いかに、ドキュメントの品質を、組織的に担保するか？

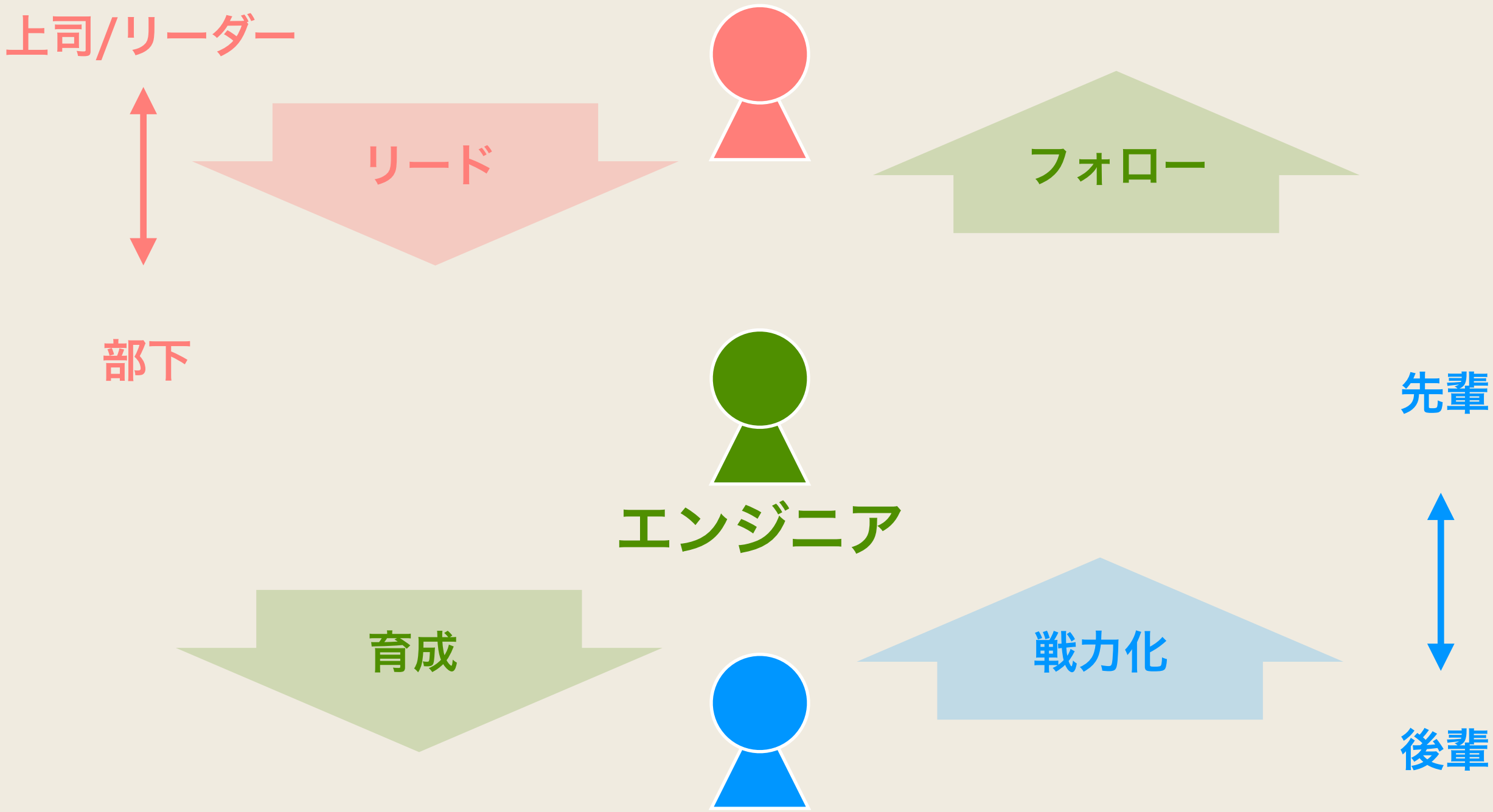
ドキュメントの過不足を明確にする

いかに、ドキュメントの過不足を把握するか？

「無駄を避ける」ドキュメンテーション

ドキュメンテーションにおける人的リソースの最大活用

役割分担して、ドキュメント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

上司でなければ書けないドキュメントを書いてもらう

上司や後輩に上手く使われる

自分にしか書けないドキュメントだけを書く

後輩を上手く戦力化する

自分でなくても書けるドキュメントを書いてもらう

「無駄を避ける」ドキュメンテーション

ドキュメンテーションの責任と品質

組織のドキュメントは、原則として「責任」を明確にし、「品質」を担保する



他人の目に触れるものだから、
わかりやすく、長く使えるものを作ろう

有責任
高品質

組織として
ロスが少ない

管理責任 (= 権限移譲)
品質責任 (= 重点評価)

「無駄を避ける」ドキュメンテーション

ドキュメントの過不足把握

組織のドキュメントは「ドキュメントツリー」を明確にし、「過不足」を把握する



存在するドキュメントの
品質を高める

存在しないドキュメントを
新規に作成する

適正なドキュメント

過剰なドキュメント

不足しているドキュメント

どんなドキュメントが必要で
現在どのドキュメントがどのような品質で存在するのか？

「無駄を避ける」ドキュメンテーション

組織的なドキュメンテーションの重要性

「ドキュメント単体で上手に書ける事」は**必須条件**

しかし、ドキュメント単体で上手に書けても効果は限定的

ミクロの観点
(第1回、第2回)

ドキュメントを「**組織知**」にしていく必要がある

複数のドキュメントが、有機的に連携し、相乗効果を上げ続けていく

マクロの観点
(第3回 = 今回)

「**組織知**」は、**エンジニアを継続的に育てる**

職人的(暗黙知)教育から、エンジニアリング(形式知)教育への発展

フィードバック効果

「無駄を避ける」ドキュメンテーション

役割分担して、ドキュメントのパフォーマンスが最大になるように動く

組織のドキュメントは、原則として「責任」を明確にし、「品質」を担保する

組織のドキュメントは「ドキュメントツリー」を明確にし、「過不足」を明確にする

アジェンダ

- ・ Step1. ドキュメントの品質を担保する
- ・ Step2. ドキュメントのパフォーマンスを最大化する
- ・ Step3. ドキュメントの過不足を明確にする

1. ドキュメントの品質を担保する

ナビゲーション

	序. 「無駄を避ける」ドキュメンテーション
Step1	1. ドキュメントの品質を担保する 「人が書く」=コストがかかる ドキュメントのライフサイクル 誰が書くべきか
Step2	2. ドキュメントのパフォーマンスを最大化する 「最小の工数」で「最大の効果」 記述対象を「書きやすい」状態にする 「人手」以外に書かせる
Step3	3. ドキュメントの過不足を明確にする



1. ドキュメントの品質を担保する

「人が書く」=コストがかかる

ナビゲーション

	序. 「無駄を避ける」ドキュメンテーション		
Step1	1. ドキュメントの品質を担保する		
	「人が書く」=コストがかかる	ドキュメントのライフサイクル	誰が書くべきか
Step2	2. ドキュメントのパフォーマンスを最大化する		
	「最小の工数」で「最大の効果」	記述対象を「書きやすい」状態にする	「人手」以外に書かせる
Step3	3. ドキュメントの過不足を明確にする		



「人が書く」=コストがかかる

「人が書く」=コストがかかる

事業コストに占める人件費の割合は、極めて大きい



品質がバラバラな
ドキュメントが
散在・重複する

リソースの発散

手間を掛けたのに
低品質、使われない

は、許されない....

「人が書く」=コストがかかる

その人の時間単価を意識する

「スキルがあるから」「書きたいから」でドキュメントを書くわけではない



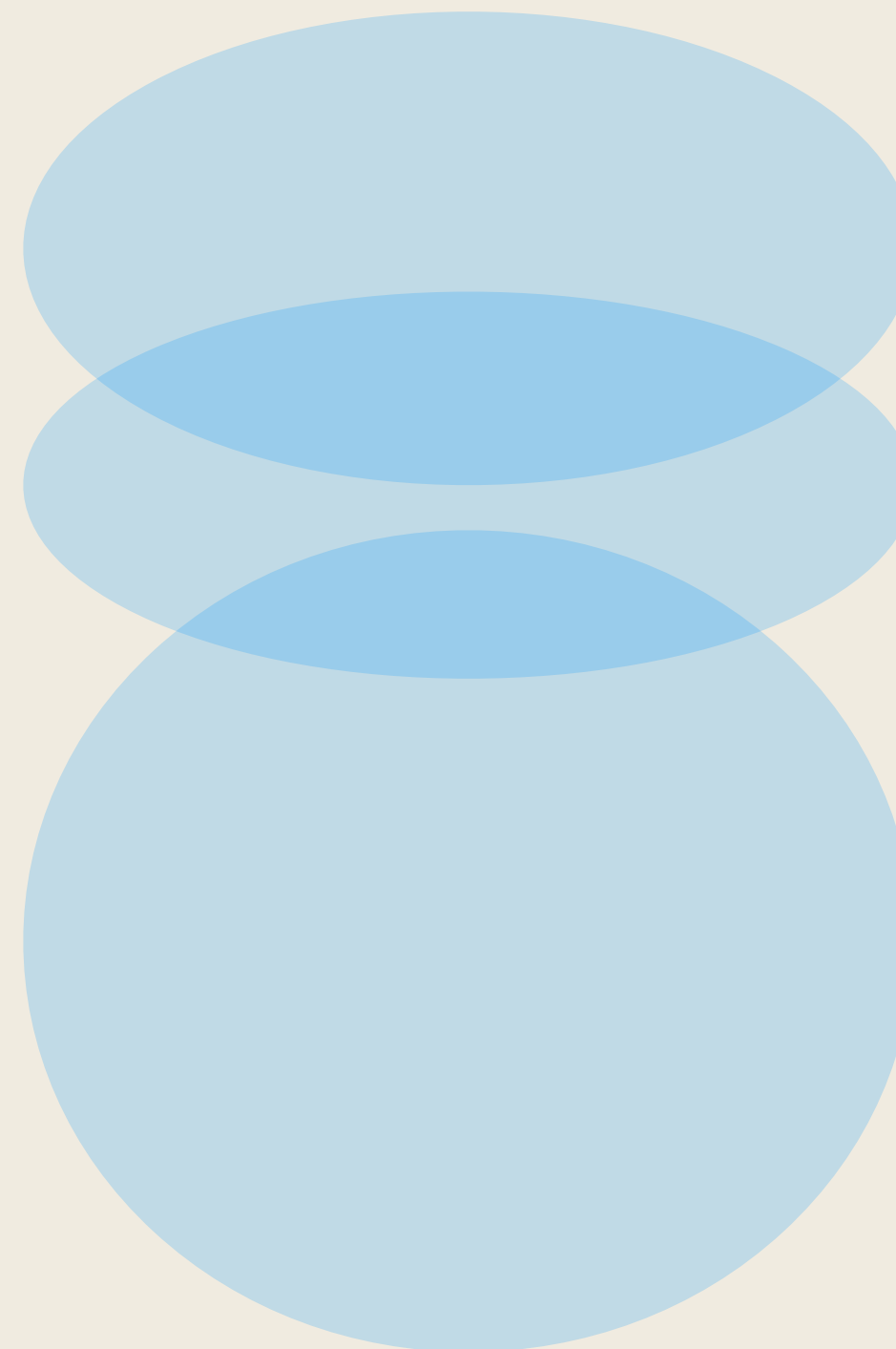
上司/リーダー



エンジニア



部下・後輩



高い時間単価

単価に見合うドキュメントを書くべき (内容&品質)

比較的高い時間単価

単価に見合うドキュメントを書くべき (内容&品質)

比較的低い時間単価

積極的にドキュメントを書いてもらうべき (品質は重要)

「人が書く」=コストがかかる

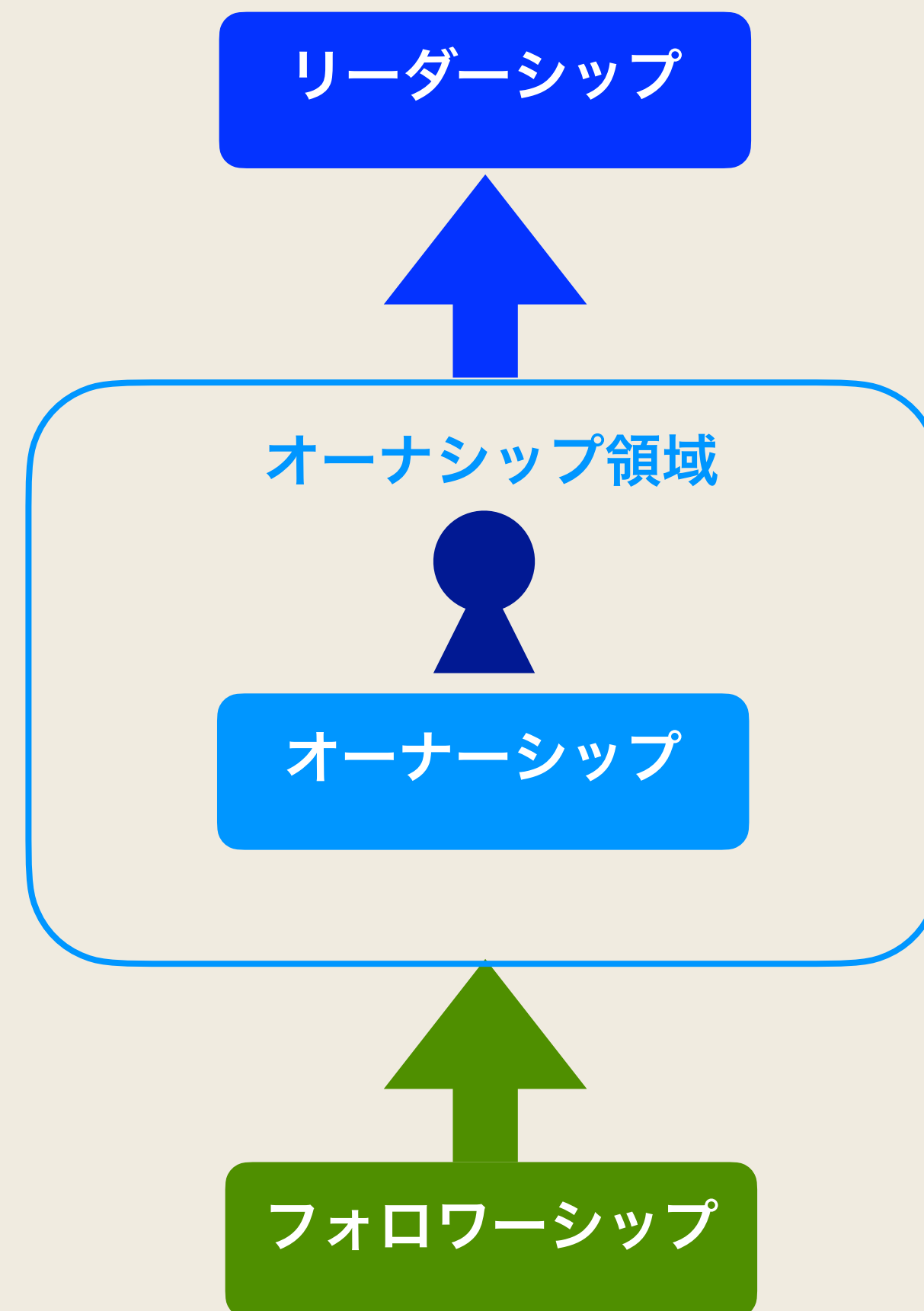
責任と権限移譲: *シップという考え方

役割間の関係に応じて、それぞれが以下のマインドセットを持つ必要がある



「人が書く」=コストがかかる

責任と権限移譲: *シップという考え方



Step3. リーダーシップ

イメージは部活の主将や部長

リーダーシップの発揮の仕方を明確に定める。(サーバント型など)
リーダーとして「自分が何ができるか」を考え続けることが重要。

Step2. オーナーシップ

イメージは部活の2年生(レギュラー)

オーナーシップを発揮すべき「オーナーシップ領域」を明確に定める。
オーナーとして何が求められるか、を自分で考えるのが最初の一步。

Step1. フォロワーシップ

イメージは部活の新入部員

リーダーや他のメンバーへのフォロワーシップを育成する。
リーダーや他のメンバーが何を求めているか、を理解するのが最初の一步。

「人が書く」=コストがかかる

責任と権限移譲: オーナシップという考え方

オーナーシップの無いドキュメントは、品質担保することができない

オーナーシップ領域



オーナーシップ

Step2. オーナーシップ

イメージは部活の2年生(レギュラー)

オーナーシップを発揮すべき「オーナーシップ領域」を明確に定める。
オーナーとして何が求められるか、を自分で考えるのが最初の一步。

オーナーシップが有る: ドキュメント品質を常に維持する意識が働く

オーナーシップが無い: 基本的に他人任せ & 特に困らなければ放置 (当然の行動)

「人が書く」=コストがかかる

「人目に晒す」ことの効果

短時間でドキュメント品質を高めるには、多くの人の目に触れ、指摘してもらうことが有効

ドキュメント作成に対する責任・緊張

人の目に触れる、チェックしてもらうということで、一定の責任と、緊張が品質の向上に繋がる

多様な目からのチェック

多様な人からの指摘により、自分だけでは気付きづらい観点からの指摘を得られる

スキル移転効果

指摘を通して、指摘した人の知見(スキルや経験)の一部を、獲得することができる

空間軸と時間軸の2つの幅で広いことが大事。

- ・ 空間軸: チーム内向け、組織内向け、ユーザ向け
- ・ 時間軸: 過去の経緯、現在の最新版、未来へ伝言

査読効果

教育効果

「人が書く」=コストがかかる

時間単価を意識した分担

組織内部で、**時間単価を意識**して、ドキュメンテーションを**分担**する



上司/リーダー

上司でなければ書けないドキュメントを書く



エンジニア

自分にしか書けないドキュメントだけを書く



部下・後輩

自分でなくても書けるドキュメントを書いてもらう

「人が書く」=コストがかかる

ドキュメントのオーナーを移譲する

自分でなくても書けるドキュメントは積極的に**オーナー移譲**する



上司でなければ書けないドキュメントを書いてもらう



自分にしか書けないドキュメントだけを書く



自分でなくても書けるドキュメントを書いてもらう

上司でなくても書けるドキュメント

正式にオーナー移譲

ただし、結果責任は負う

自分でなくても書けるドキュメント

正式にオーナー移譲

ただし、結果責任は負う

部下や後輩でも書けるドキュメント

「人が書く」=コストがかかる

ドキュメントを人の目に晒す

必然的に「ドキュメントが人の目に晒される」状態を定常化する

ドキュメント作成を、**正式な主要業務とする** (一定品質の要求)

ボランティアベース、ボトムアップでの活動では、限界がある & 整合性が取れない & オレオレ化しやすい

ドキュメントを、**公式の掲載場所に必ず配置する** (成果物の活用)

オレオレドキュメントは公式業務と認めず、工数を使うなら、正式かつ関係者全員が見えるところに配置する

ドキュメンテーション活動を、**主要な評価対象とする** (活動誘因の喚起)

ドキュメンテーション活動に適切に貢献した人を、適切に評価する (ボランティア活動にしない)

最初は、抵抗や心理的不安が大きいですが、慣れてくると成果物を人目に晒すことが全然気にならなくなる

「人が書く」=コストがかかる

人手のリソースは有限 かつ 貴重。ロスの大いいドキュメンテーションはNG。

オーナーシップを明確にすることで、ドキュメントの陳腐化回避、品質向上に繋がる

人目に晒すことが、ドキュメント品質の向上、スキルの向上に繋がる

1. ドキュメントの品質を担保する

ドキュメントのライフサイクル

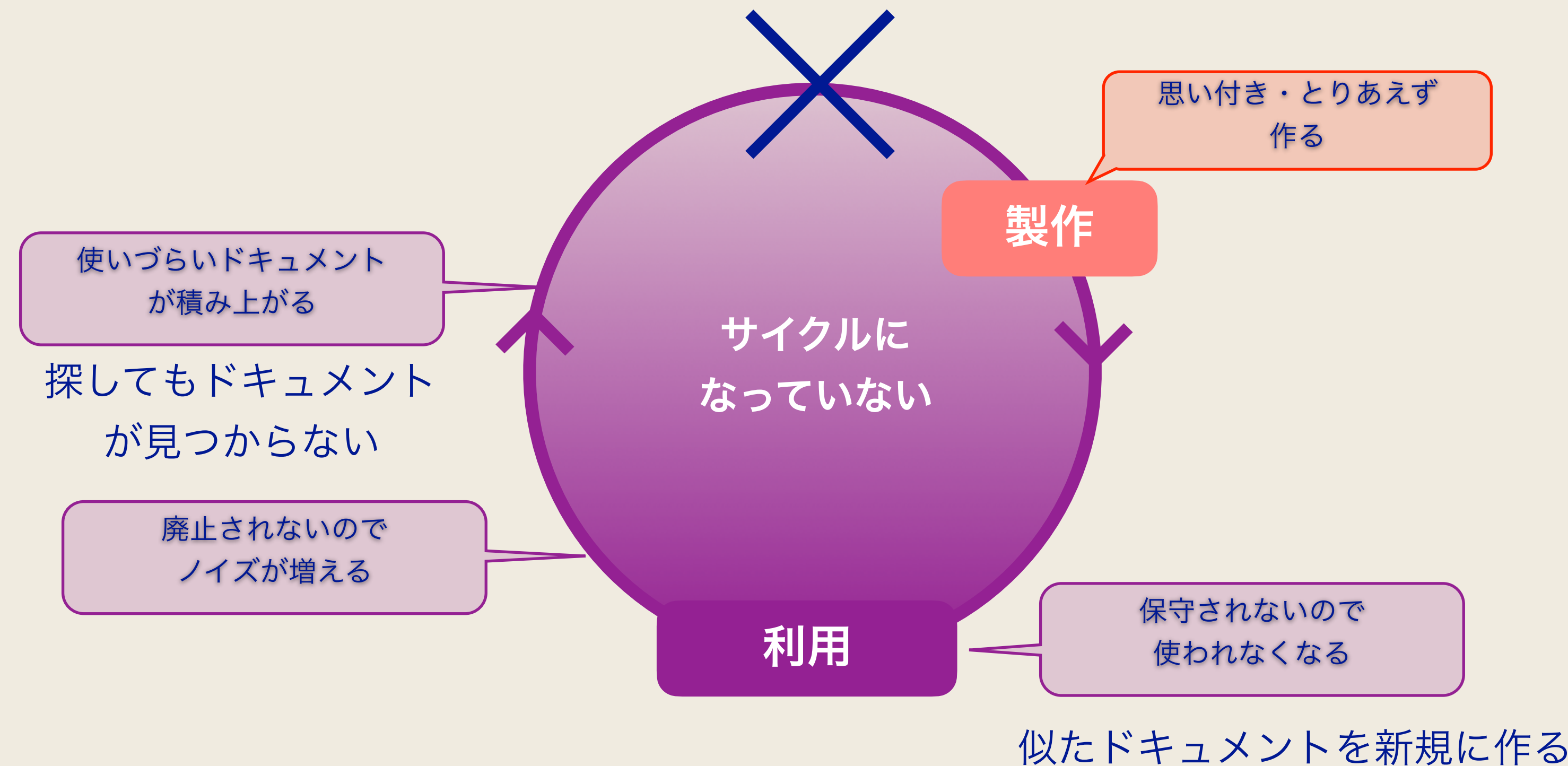
ナビゲーション

	序. 「無駄を避ける」ドキュメンテーション
Step1	1. ドキュメントの品質を担保する 「人が書く」=コストがかかる ドキュメントのライフサイクル 誰が書くべきか
Step2	2. ドキュメントのパフォーマンスを最大化する 「最小の工数」で「最大の効果」 記述対象を「書きやすい」状態にする 「人手」以外に書かせる
Step3	3. ドキュメントの過不足を明確にする



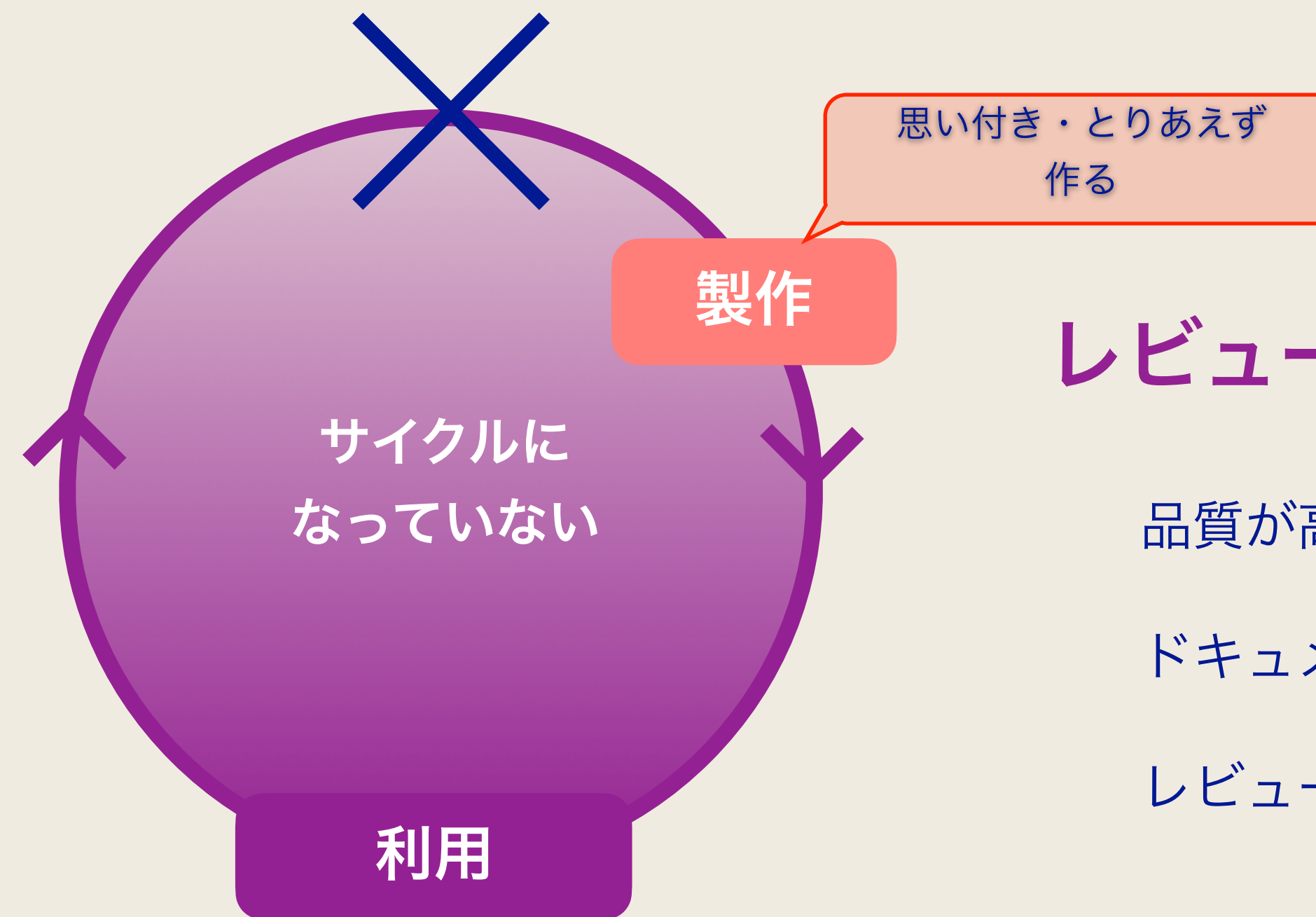
ドキュメントのライフサイクル

ライフサイクルが意識されていないドキュメンテーション



ドキュメントのライフサイクル

レビューされないドキュメント



レビューが行われない

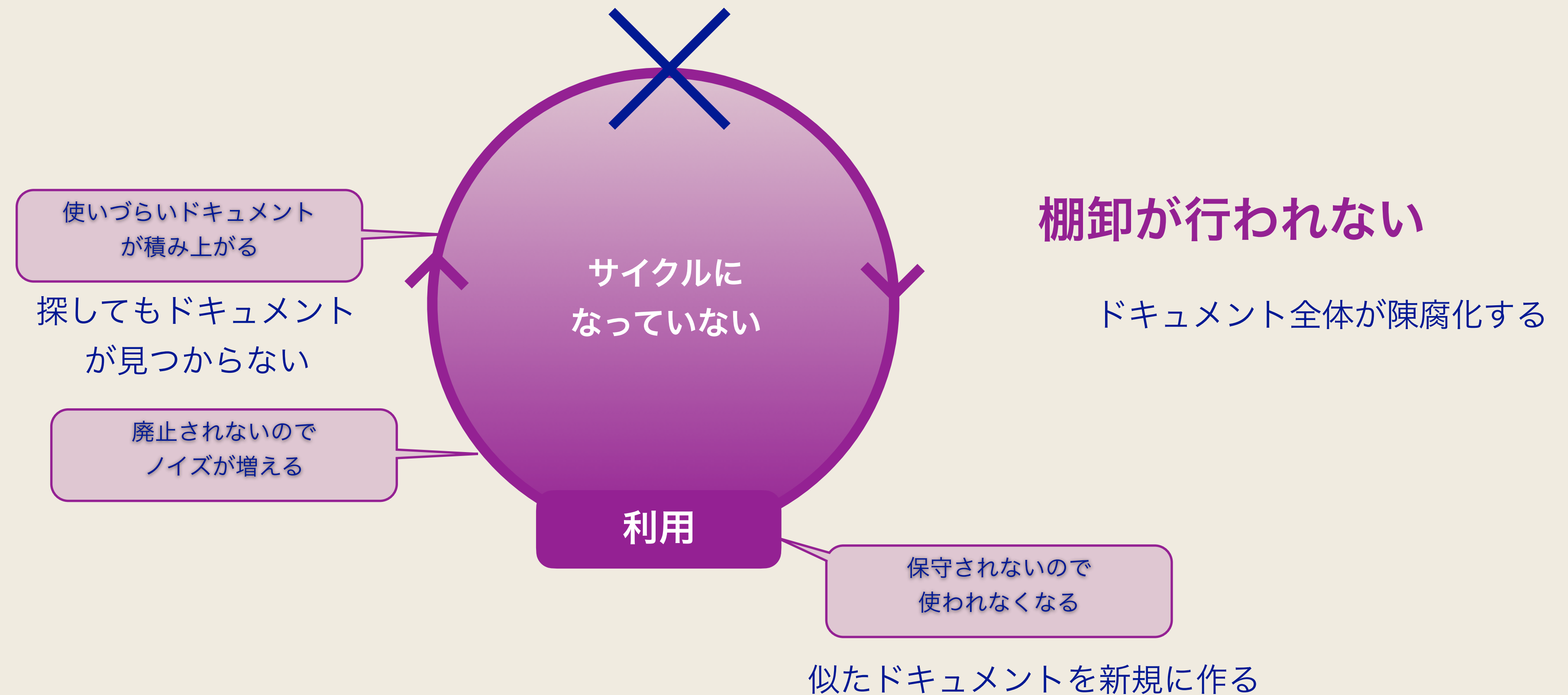
品質が高くない

ドキュメントスキルが向上しない

レビュースキルが向上しない

ドキュメントのライフサイクル

棚卸されないドキュメント

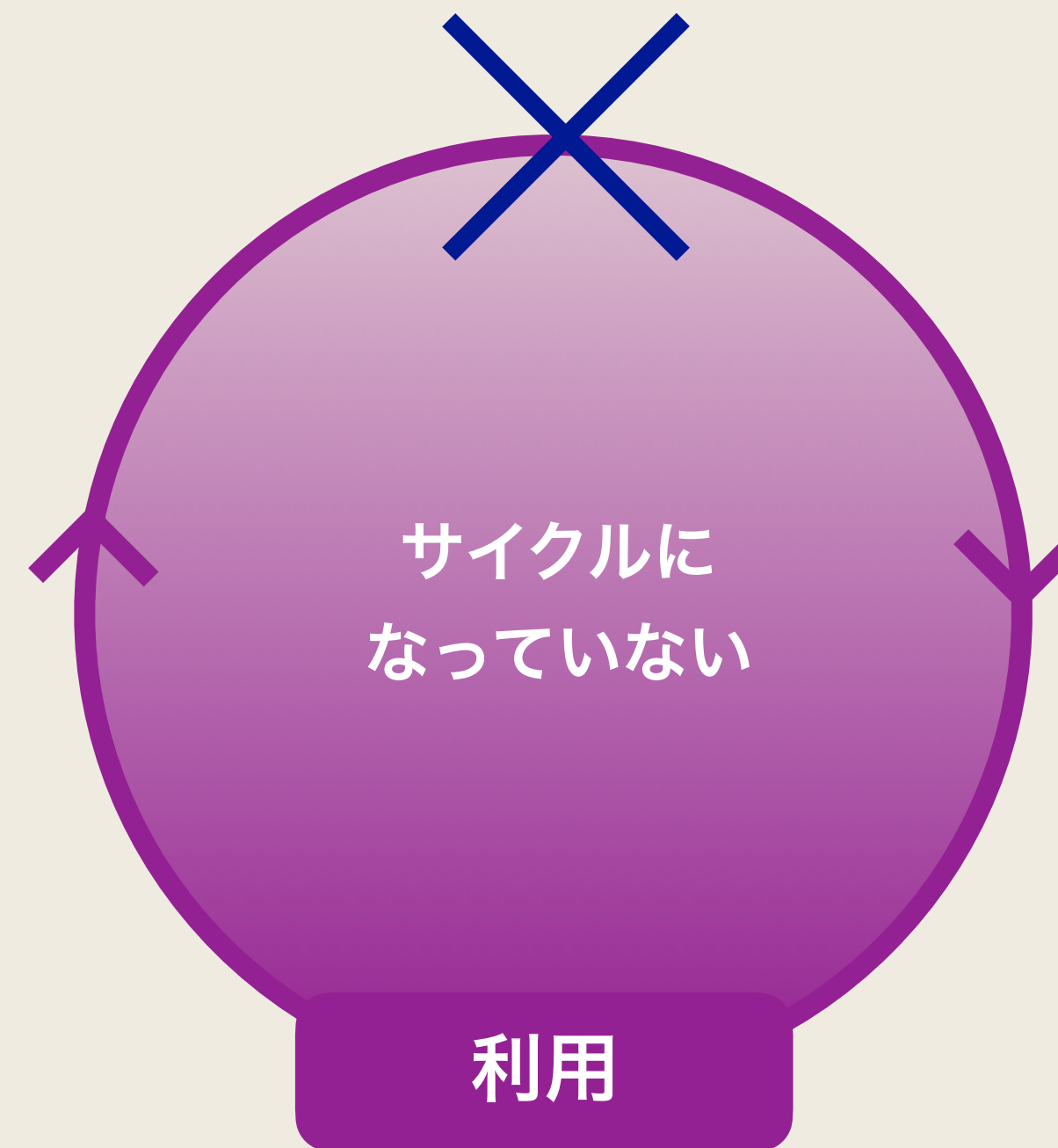


ドキュメントのライフサイクル

まとめ: ライフサイクルが意識されていないドキュメンテーションの末路

棚卸が行われない

ドキュメント全体が陳腐化する



レビューが行われない

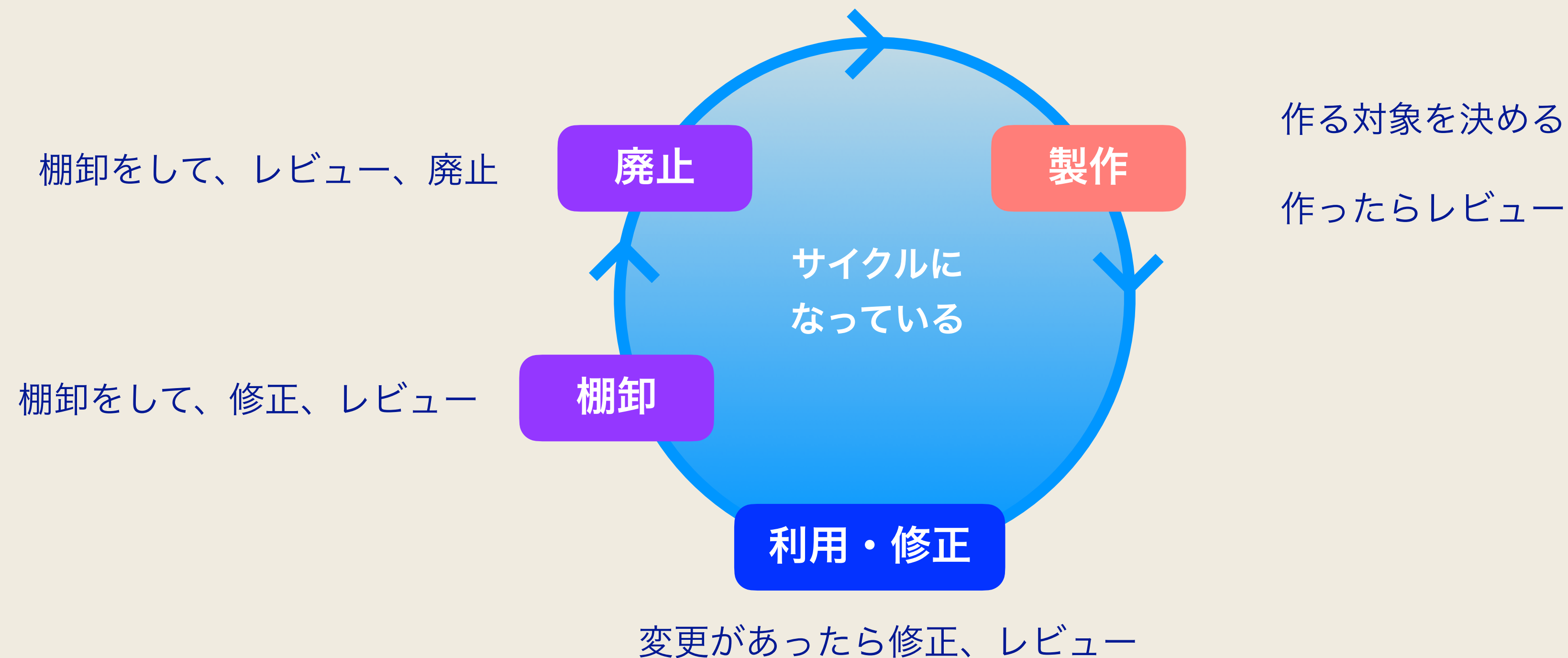
品質が高くない

ドキュメントスキルが向上しない

レビュースキルが向上しない

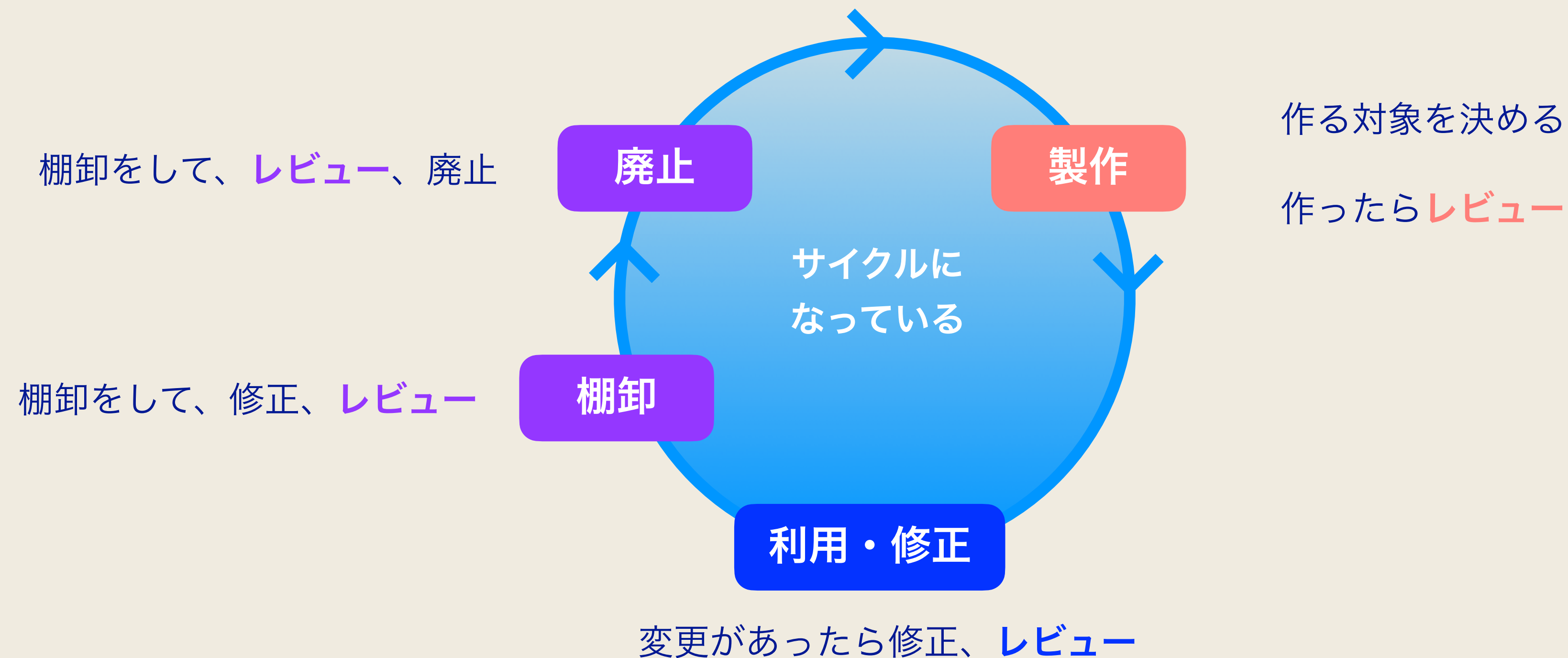
ドキュメントのライフサイクル

ドキュメントのライフサイクル



ドキュメントのライフサイクル

レビューの定着化



ドキュメントのライフサイクル

レビュー: 基本モデル (セルフレビュー)

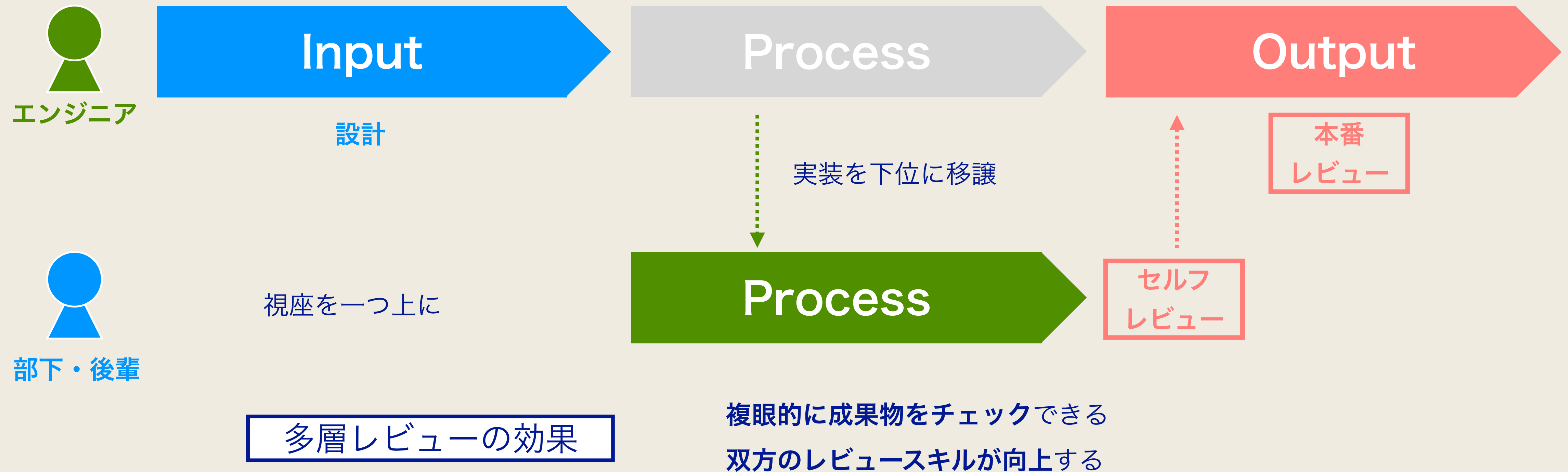


セルフレビューの効果

自分の成果物を客観化して見るようになる
自身のレビュースキルが向上する
他人の成果物のレビュースキルができるようになる

ドキュメントのライフサイクル

レビュー: 階層モデル (多層レビュー)



ドキュメントのライフサイクル

棚卸しの実施

定期的に、棚卸しを実施する

最低でも年1回(できれば四半期に1回)は実施する。(量が多い場合は四半期に1/4の範囲で実施など)

棚卸し基準を定めておく (陳腐化回避、ノイズ比率低減)

一定の期間以上使われていないものは廃止、一定の期間以上更新されていないものはレビューする、など

棚卸しは、組織全体の取り組みとして行う

個人の任せっきりにしたり、ボランティア活動として行う棚卸しは、実効性が低い。

ドキュメントのライフサイクル

ドキュメントのライフサイクルを意識することが品質担保の第一歩

レビューを重ねることが、ドキュメント品質の向上、スキルの向上に繋がる

棚卸しをすることが、ドキュメント全体の陳腐化回避、品質向上に繋がる

1. ドキュメントの品質を担保する

誰が書くべきか

ナビゲーション

	序. 「無駄を避ける」ドキュメンテーション
Step1	1. ドキュメントの品質を担保する 「人が書く」=コストがかかる ドキュメントのライフサイクル 誰が書くべきか
Step2	2. ドキュメントのパフォーマンスを最大化する 「最小の工数」で「最大の効果」 記述対象を「書きやすい」状態にする 「人手」以外に書かせる
Step3	3. ドキュメントの過不足を明確にする



誰が書くべきか

誰々でなければ書けないドキュメントの意味

「職責」を意味する場合と、「スキル」を意味する場合がある



上司/リーダー

上司の職責として書かなければいけないドキュメントを書く

個人スキルとして他人が書けないドキュメントを書く



エンジニア

エンジニアの職責として書かなければいけないドキュメントを書く

個人スキルとして他人が書けないドキュメントを書く



部下・後輩

書けるドキュメントを書いてもらう

誰が書くべきか

職責として書かなければいけないドキュメント

「職責」に基づくドキュメントはトップダウンで決まり、移譲できない



上司の職責として書かなければいけないドキュメントを書く

方針的なドキュメント (業務の全体的な骨格を決めるもの。基本設計、基本規約など)



エンジニアの職責として書かなければいけないドキュメントを書く

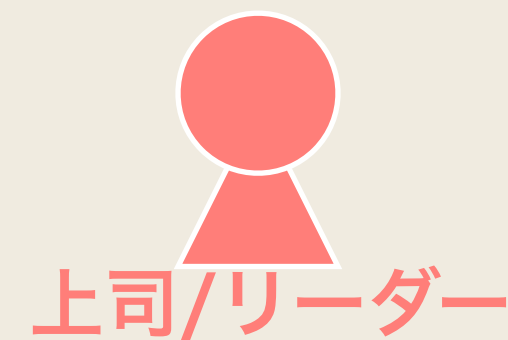
専門職としてのドキュメント (業務の主要な骨格を決めるもの。個別の設計、規約、課題など)



誰が書くべきか

スキルに基づいて書かなければいけないドキュメント

「スキル」に基づくドキュメントは積極的に移譲すべき



上司/リーダー

人数が少なく、工数が貴重

個人スキルとして他人が書けないドキュメントを書く

育成してオーナー移譲

ただし、結果責任は負う



エンジニア

人数が限定的で、工数が貴重

個人スキルとして他人が書けないドキュメントを書く

育成してオーナー移譲

ただし、結果責任は負う



部下・後輩

人数が比較的多く、工数を活用できる

書けるドキュメントを書いてもらう

誰が書くべきか

それぞれの職責に求められるドキュメントを書く

自分でなくても書けるドキュメントは積極的に**オーナー移譲**する



上司/リーダー

職責によるドキュメント

職責として書かなければいけないドキュメントを書く



エンジニア

職責として書かなければいけないドキュメントを書く



部下・後輩

職責に関係なく書けるドキュメントを書く

個人スキルによるドキュメント

上司でなくても書けるドキュメント

正式にオーナー移譲

ただし、結果責任は負う

自分でなくても書けるドキュメント

正式にオーナー移譲

ただし、結果責任は負う

部下や後輩でも書けるドキュメント

誰が書くべきか

職責とドキュメント量

職責が上の人ほど積極的にドキュメントを書くべき（相対的に人数が少ない）



上司/リーダー

書くべきドキュメントは相対的に少ないが影響が大きい

スキルの移転とオーナーの移譲による効果はかなり大きい



エンジニア

書くべきドキュメントは広範囲でボリュームがある

スキルの移転とオーナーの移譲による効果大きい



部下・後輩

書けるドキュメントを書いてもらう

スキルの向上とオーナーの譲受により、できることが広がる

誰が書くべきか

対象の理解とドキュメント

ドキュメントは、「本当にわかっている人」にしか書けない（一般的に、一番時間を使っている人）

一般的に、**上司より、エンジニアの方が詳しい**

上司と部下はあくまでも役割分担。上司の方が常に詳しい、正しいわけではない。（喧嘩腰ではなく謙虚に）

一般的に、**エンジニアより、上司の方が、ドキュメント品質は高い**

現場の「言語化」はエンジニアにしかできないが、その言語化が適切かどうかは上司が判断し、指導育成する。

昇格・異動による引き継ぎがスキル移転のチャンス

引き継ぎ後しばらくすると知識量は逆転する。その前にドキュメントを通じて、後任のエンジニアにスキル移転する。

誰が書くべきか

実装とレビューのスキル

原則的に、レビューできる人 = 自分で実装できる人



ドキュメントをレビューできる人 = ドキュメントを書ける人

コードをレビューできる人 = コードを書ける人 (生成AIで代替)

誰が書くべきか

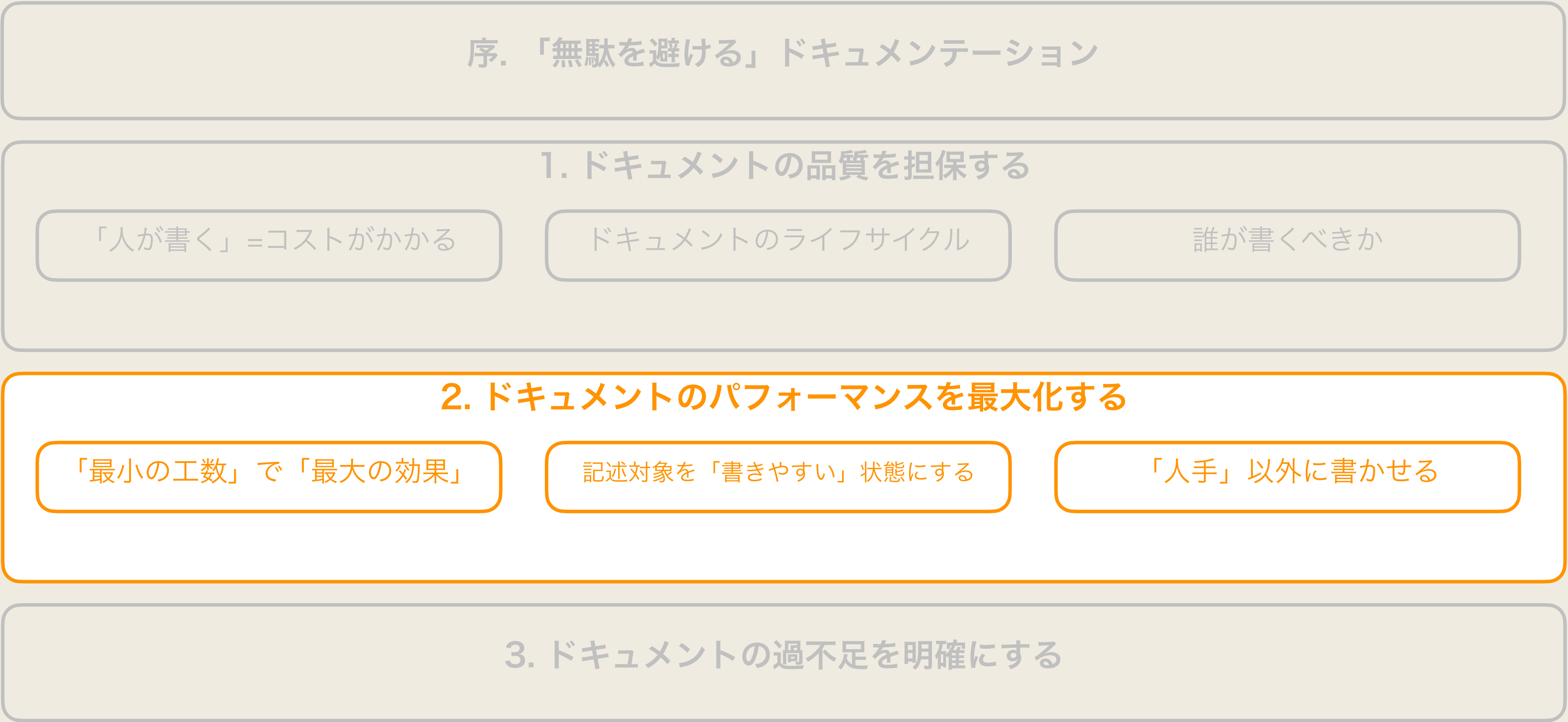
職責として書くべきドキュメントと、スキルの的に書けるドキュメントを区別する

上位職は、職責として書くべきドキュメントを積極的に書くことが求められる

下位職は、スキルの的に書けるドキュメントを積極的に増やすことが求められる

2. ドキュメントのパフォーマンスを最大化する

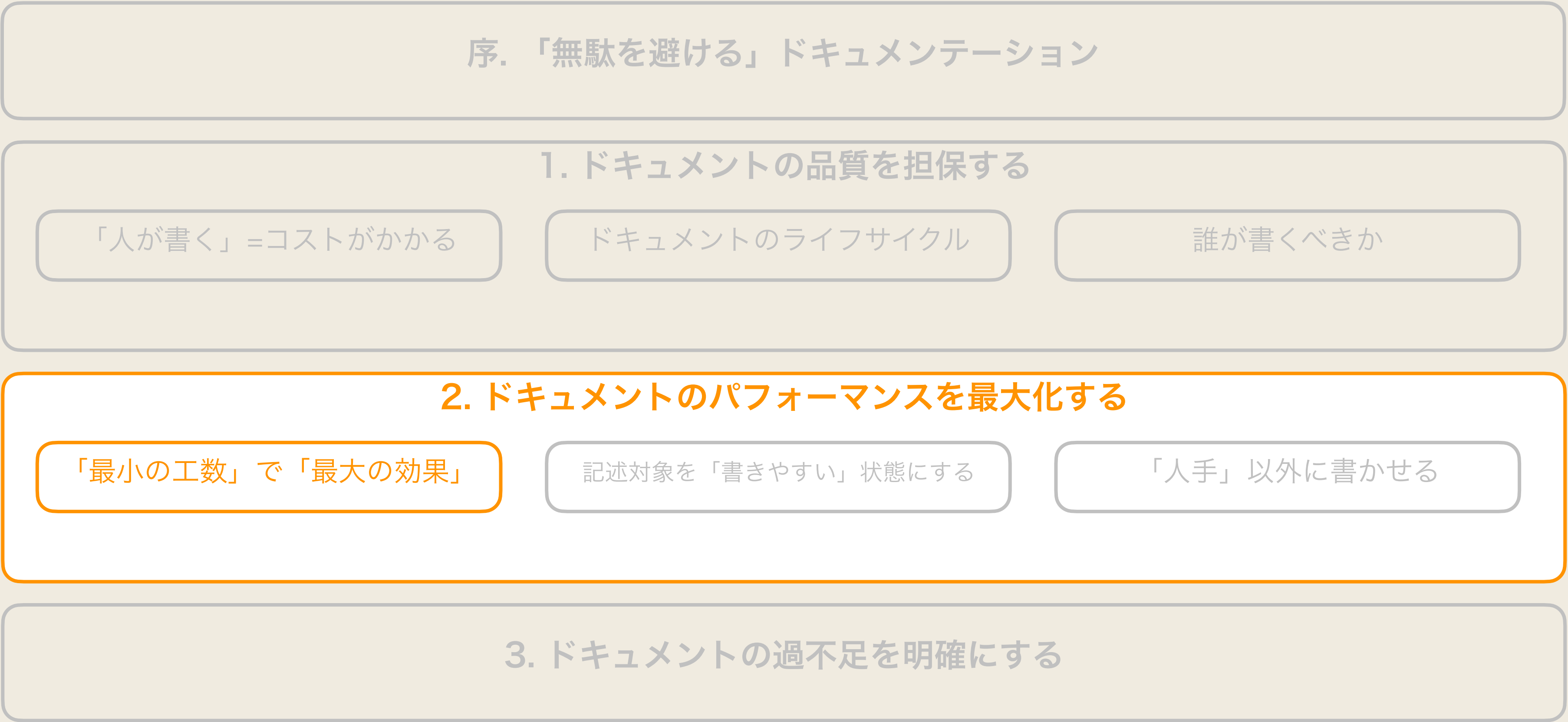
ナビゲーション



2. ドキュメントのパフォーマンスを最大化する

「最小の工数」で「最大の効果」

ナビゲーション



「最小の工数」で「最大の効果」

「工数」を回収できないドキュメンテーション



上司/リーダー



エンジニア



部下・後輩

品質がバラバラな
ドキュメントが
散在・重複する

リソースの発散

掛けた工数ほど使われない

投下した工数を回収できない

掛けた工数ほど使われない

投下した工数を回収できない

掛けた工数ほど使われない

投下した工数を回収できない

「最小の工数」で「最大の効果」

「工数」を回収できない原因

単価が高い人が、ドキュメント価値の低いドキュメントを作っている

単価の低い人に、スキル移転 & オーナー移譲することで対応できる

陳腐化が早いドキュメントに工数を掛け過ぎている

陳腐化が早いドキュメントに工数を掛け過ぎないようにする必要がある

そもそもドキュメントを過剰に作り過ぎている

ドキュメントは、必要十分を目指し、過剰に作り過ぎないようにする必要がある

「最小の工数」で「最大の効果」

陳腐化が早いドキュメント

費用性ドキュメント

作成したときに最大の価値を持ち、再利用性が低いために急激に陳腐化する

変化が早いものに対する資産性ドキュメント

変化に追随するために、保守工数が掛かり過ぎる。保守しないと内容が古くなり、急激に陳腐化する。

過剰に作り込みされたドキュメント（陳腐化）

ボリュームがあるため、保守工数が掛かり過ぎる。保守しきれないと、急激に陳腐化する。

「最小の工数」で「最大の効果」

ドキュメントの作り過ぎ

誰も使わないドキュメント

そもそも利用価値を生まないため、その工数が全て無駄になる。

自社で作る必要が無いドキュメント

ドキュメントが充実した気になるが、貴重な工数が非効率的に消費されてしまう。

過剰に作り込みされたドキュメント（工数浪費）

作成者の満足度は高いが、貴重な工数が非効率的に消費されてしまう。

「最小の工数」で「最大の効果」

工数の単価に見合う「最大の効果」を意識する



上司/リーダー

高い単価に見合うドキュメンテーション



エンジニア

高い単価に見合うドキュメンテーション



部下・後輩

適度な単価に見合うドキュメンテーション

「最小の工数」で「最大の効果」

工数に見合うドキュメンテーション

不要なドキュメントは書かない

「そもそも書かない」は論外。一方で、「書けばよい」わけではない。

「必要十分なドキュメントを書く」と「不要なドキュメントも書く」は全くの別物。

寿命が短いドキュメントに工数を掛けない (費用性)

寿命が短くても、必要最低限の工数を掛ける必要はある。(過剰に掛けないことが大事)

永く使うドキュメントに工数を掛ける (資産性)

工数効率の悪い(変化の激しいもの、寿命が近いもの)に対しては、工数は最低限に抑える。(必要な分は工数投入する)

工数効率の良い(変化が少ない、寿命が長いもの)に対して、工数を集中する。

「最小の工数」で「最大の効果」

「書かずに済みます」という選択肢

不要なドキュメントは書かない

「そもそも書かない」は論外。一方で、「書けばよい」わけではない。

「必要十分なドキュメントを書く」と「不要なドキュメントも書く」は全くの別物。

自前では書かず、外部ドキュメントへの参照で済ます (外部参照)

公開されている公式情報や、標準化団体のドキュメントを参照情報に示すくらいに留める

自前では、外部ドキュメントのダイジェストに留める (ダイジェスト/キャッシュ的ドキュメント)

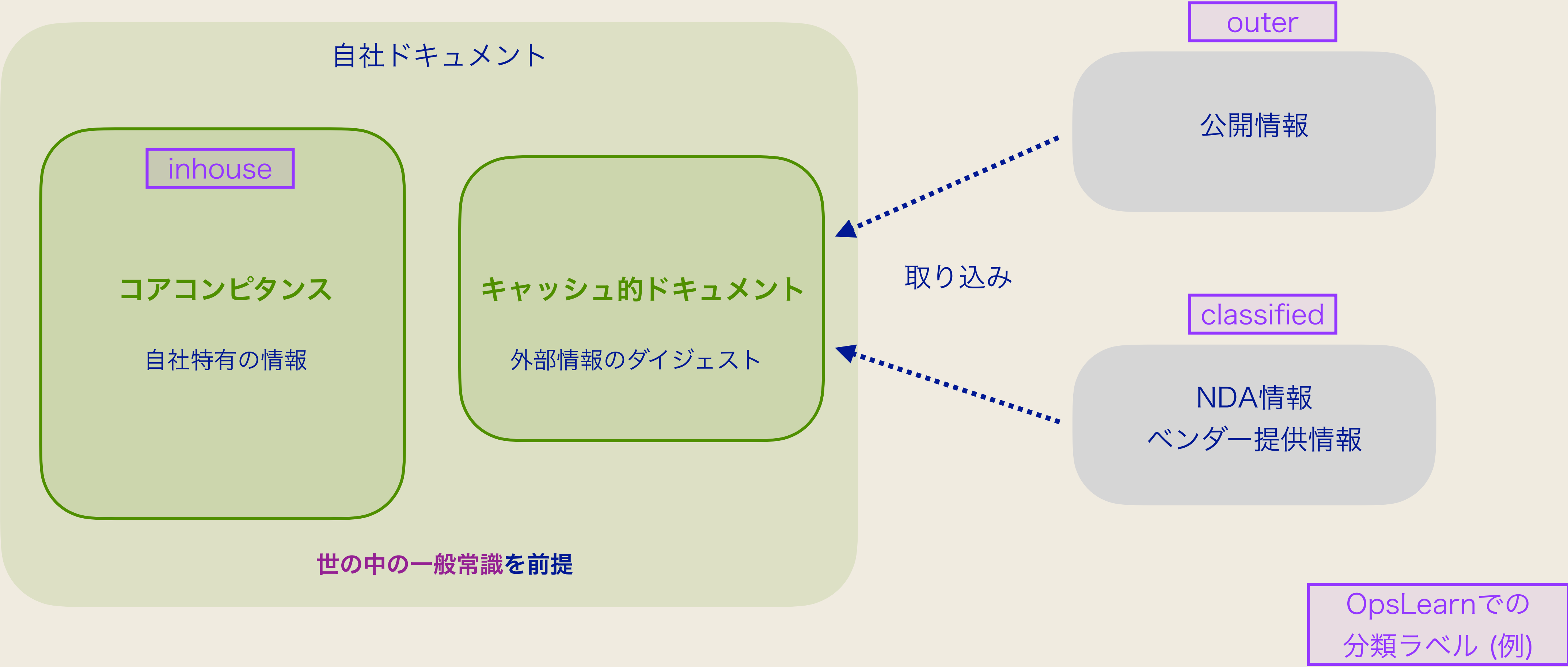
公開されている公式情報や、標準化団体のドキュメントのダイジェストを作成し、必要十分な保守をする。

自社のコアコンピタンス以外は、世の中の一般常識を前提とする (ローコンテキスト化)

世の中(専門領域)の一般常識については、自社ドキュメントには、あえて記述しない。

「最小の工数」で「最大の効果」

工数で考えるドキュメンテーションマップ



「最小の工数」で「最大の効果」

「書き手」の工数の単価に見合う「最大の効果」を意識する

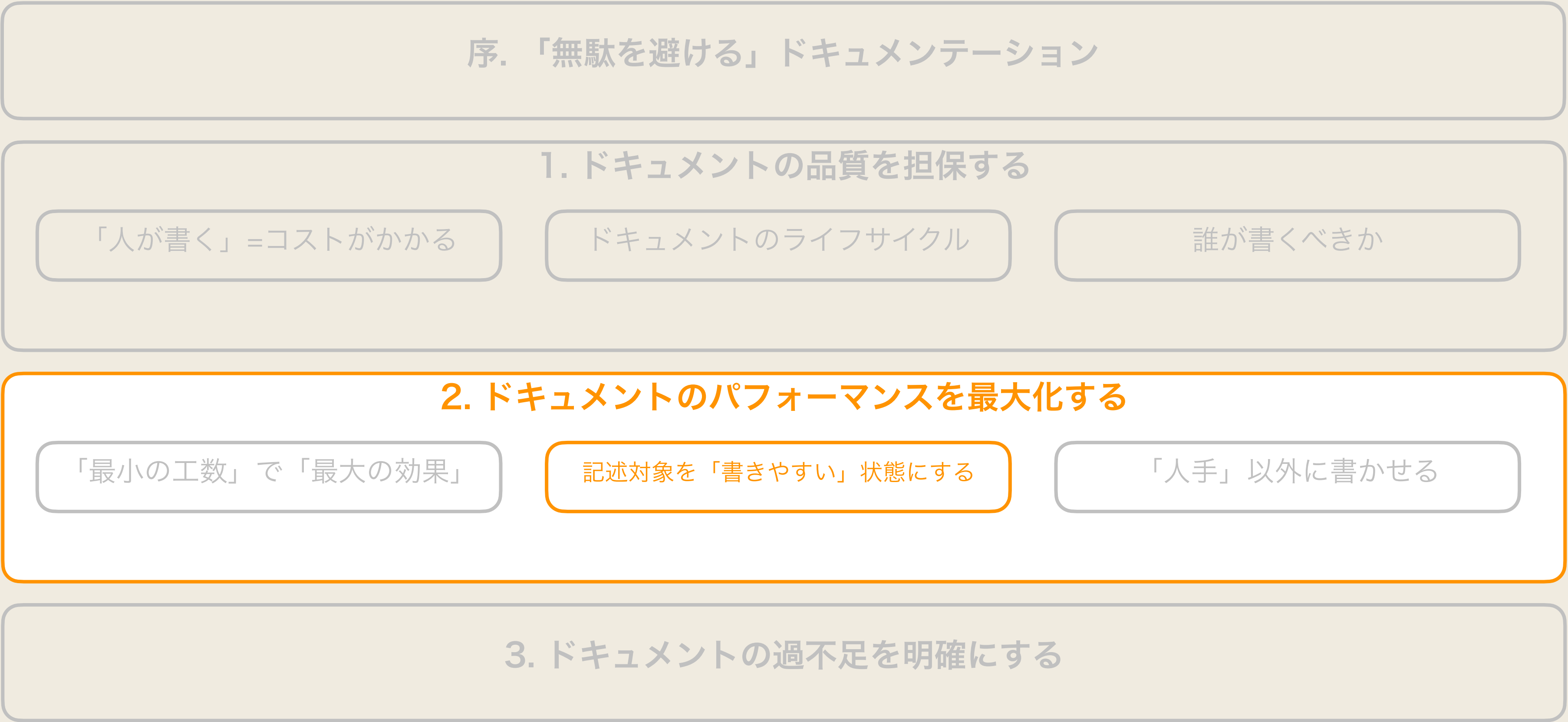
「成果物」陳腐化の早さと工数に見合うドキュメンテーションを行う

「書かずに済みます」という選択肢を常に意識する

2. ドキュメントのパフォーマンスを最大化する

記述対象を「書きやすい」状態にする

ナビゲーション



記述対象を「書きやすい」状態にする

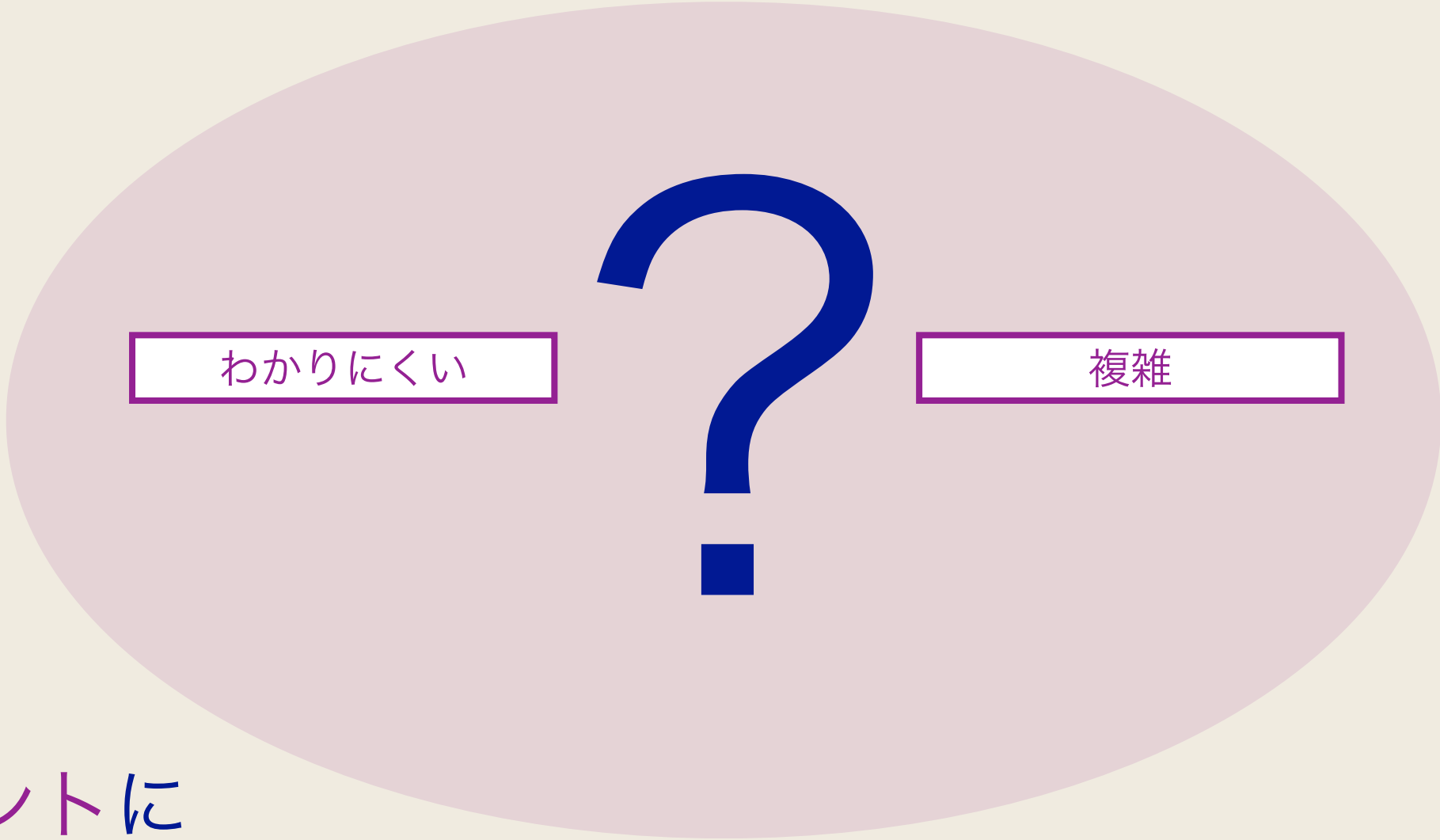
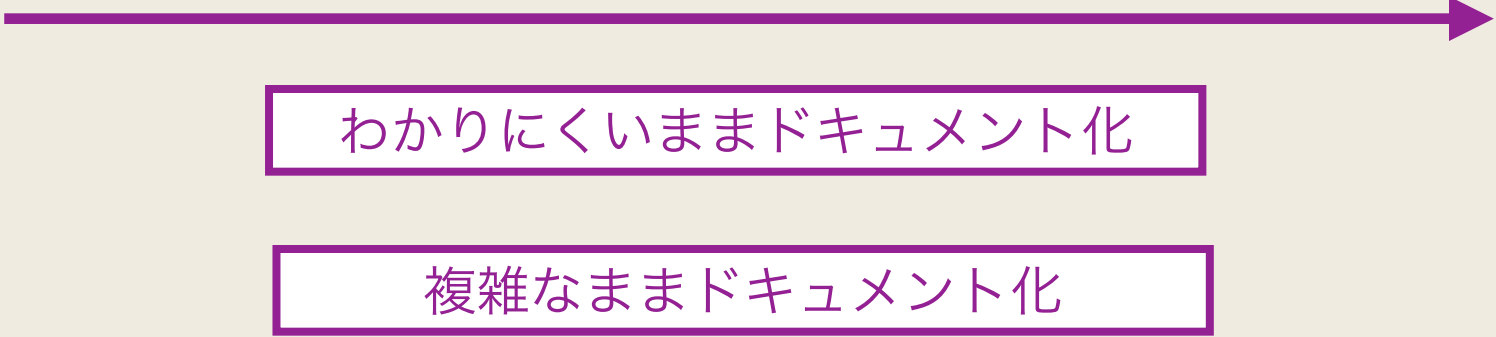
ドキュメントの対象の「そのままの状態」が前提

書かれる側はそのまま、**「書く側で頑張ろう」**としがち

対象がわかりにくいまま書こうとするからつらい

書く側

書かれる側



工数が掛かるが、**わかりにくいドキュメントに**

記述対象を「書きやすい」状態にする

前提を変えてしまうと書きやすくなる

書く側で頑張らず、「書かれる側を変えて」しまう



わかりやすいドキュメント化

シンプルにドキュメント化

書かれる側

わかりやすく変えてしまう

整理してシンプルにしてしまう

工数が掛からず、わかりやすいドキュメントに

記述対象を「書きやすい」状態にする

前提を柔軟に考える・変える

同じ工数を使うなら、効果が出易いところに注力する

書く側



こちらで頑張っても効果は出にくい

書かれる側



こちらの整理に頑張ると効果が出やすい

記述対象を「書きやすい」状態にする

業務の独自色の価値を考える

一般的に、業務の独自色が高いと、ドキュメント量は増える

独自色に意味が無い (世間のやり方に合わせる)

ドキュメント工数を非効率にし、読み手にも負担になる

ドキュメント量とドキュメント価値は反比例する

独自色に意味が有る (資産性ドキュメントの中核に位置付ける)

他社に無いアドバンテージとして重要な意味を持つ

ドキュメント量とドキュメント価値は比例する

書かれる側



独自性の高い業務

記述対象を「書きやすい」状態にする

独自性の価値を整理する

業務の独自色に意味が有るのか、無いのかについて整理する

独自色に意味が無い (世間のやり方に合わせる)

独自色を排除するだけで、ドキュメント作成・レビューがかなり楽になる

異動による引き継ぎや、新人の戦力化が容易になる。

ドキュメント量とドキュメント価値が反比例する

独自色に意味が有る (資産性ドキュメントの中核に位置付ける)

他社に無いコンテンツとなるため、企業の重要な資産となる。

ドキュメント量とドキュメント価値は比例する

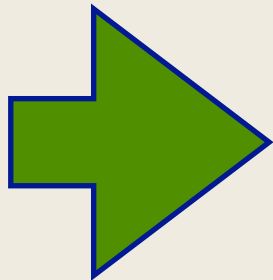
記述対象を「書きやすい」状態にする

記述対象の状態による影響

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

記述対象を「書きやすい」状態にする

理解しやすい業務による効果

あるべき運用業務

1. 人が理解しやすい業務である。

人材がスケールする

2. システムが扱いやすい業務である。

業務がスケールする

3. 論理的に正しいことを検証している。

工数を最大活用できる

- ・ 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)

記述対象を「書きやすい」状態にする

意味の無い独自色を減らすことで、ドキュメントが書きやすくなる

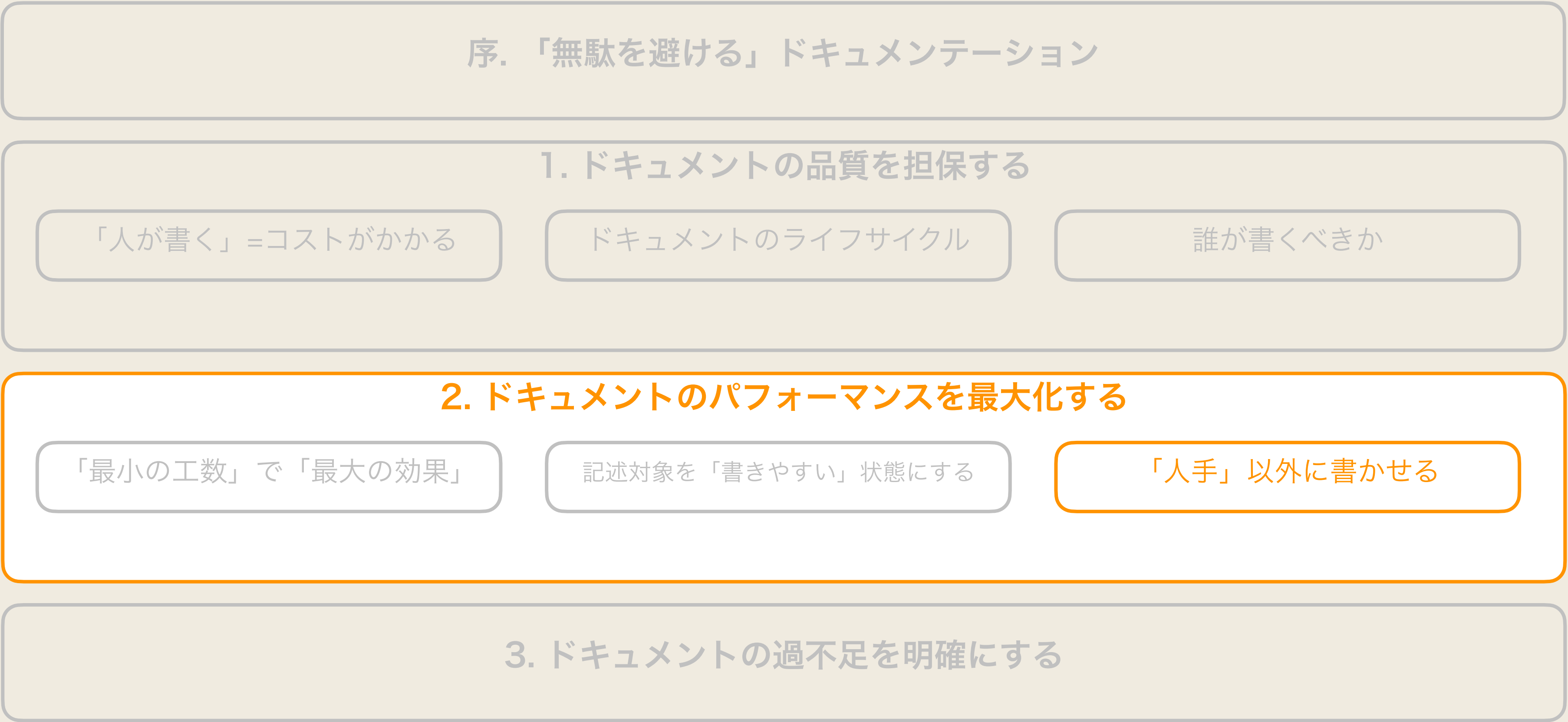
意味の有る独自色(コアコンピタンス)に集中することで、ドキュメント価値が向上する

人が理解しづらい業務を、人が理解しやすい業務に変えることで、書きやすくなる

2. ドキュメントのパフォーマンスを最大化する

「人手」以外に書かせる

ナビゲーション



「人手」以外に書かせる

人は有限のリソース / 「人が書く」=コストがかかる

人ではないもの(コンピューティングリソース)に書かせる



コンピューティング
リソース

論理が正しければ、
何万回/ページでも瞬時に生成する

ドキュメント

「人手」以外に書かせる

参考: コンピューティングリソースに何を書かせるか?

人ではないもの(コンピューティングリソース)に書かせるもの

「コンピュータが書けるもの」にしがち

ある程度の範囲までは有効であるが、早期に頭打ちになりやすい

人が書かなくてもいいもの (理想)

「人にしか書けないもの」以外を対象と考える
工夫の結果、人手を介さなくてもよくなる事が増える

全体構造は人にしか書けない

コアコンピタンスな情報は人にしか書けない

Whyは人にしか書けない

コンピューティング
リソース

論理が正しければ、
何万回/ページでも瞬時に生成する

ドキュメント

「人手」以外に書かせる

データ形式から考える

構造化データを生成する

CSV / TSV / XML / JSON

技術的には、流行り廃りがあるが、長く使えることが多い。

非構造化データを生成する

ヒアドキュメント / バイナリ

パーサーを維持できれば、どんなデータでも扱うことができる。

「人手」以外に書かせる

参考: データのライフサイクルから考える

L1: 生データ

媒体(紙、デジタル)を問わない一次データ

L2: テキスト化データ

デジタル化された一次データ。永久保存

L3: マスターデータ

5W1H情報が補完された、マスターデータ。原則として永久保存。

L4: 使い捨てデータ

L3データを加工して、自由に生成された一時データ。利用後に破棄される。

L5: 出力データ

L4データを加工して、最終成果物となったデータ。用途に応じて保存期間は決まる。

「人手」以外に書かせる

普段からドキュメント用のデータを用意・自動生成しておく

データーをドキュメント化しやすいように生成しておく

機械(コンピューター)に書かせる

データーの生成、保存、加工など定型化しておく

データーをドキュメント化しやすいように保存しておく

良いドキュメントを書くには、普段からドキュメントの材料を揃えておく必要がある

人がとりまとめる

多様な用途で使うようにする。定型化できるものはスクリプト化して反復再現的に利用する

自動化が目的ではなく、ドキュメント生成の手段が自動化されているだけ

「人手」以外に書かせる

人ではないもの(コンピューティングリソース)にもドキュメントを書かせる

記述対象の特性に合わせて**データ形式**(構造化・非構造化)を選択する

普段からドキュメント用のデータを自動生成し、**多様に活用**できるようにする

3. ドキュメントの過不足を明確にする

ナビゲーション

	序. 「無駄を避ける」ドキュメンテーション
Step1	1. ドキュメントの品質を担保する 「人が書く」=コストがかかる ドキュメントのライフサイクル 誰が書くべきか
Step2	2. ドキュメントのパフォーマンスを最大化する 「最小の工数」で「最大の効果」 記述対象を「書きやすい」状態にする 「人手」以外に書かせる
Step3	3. ドキュメントの過不足を明確にする



ドキュメントの過不足を明確にする

復習: 適切な目次は読みやすい (単体ドキュメント)

わかりやすいツリー構造

各見出しに責務が集中

目次

各見出し間がMECE

ドキュメントの過不足を明確にする

ドキュメントの上位にもツリー構造が必要

わかりやすいツリー構造

各ドキュメントに
責務が集中

ドキュメント群

各ドキュメント間が
MECE

ドキュメントの過不足を明確にする

ドキュメントの上位にもツリー構造が存在しないと

ツリー構造が存在しない

あるべきドキュメンテーションの全体像が見えない

各ドキュメントに
責務が分散

品質がバラバラな
ドキュメントが
散在・重複する

リソースの発散

各ドキュメント間が
MECEでない

どこが過剰で、どこが手薄か？

どこが重要で、どこが重要ではないのか？

ドキュメントの過不足を明確にする

ドキュメントマップを作ろう (ツリー構造)

わかりやすいツリー構造

各ドキュメントに
責務が集中

ドキュメント群

各ドキュメント間が
MECE

ドキュメントの過不足を明確にする

ドキュメントマップの役割

わかりやすいツリー構造

あるべきドキュメンテーションの全体像が見える

各ドキュメントに
責務が集中

どこが重要で、どこが重要ではないのか？

各ドキュメント間が
MECE

ドキュメント群

どこが過剰で、どこが手薄か？

どこを整備すると最小の工数で、最大の効果が出るか、見通しが付く

ドキュメントの過不足を明確にする

「決める人が必要」

わかりやすいツリー構造

各ドキュメントに
責務が集中

ドキュメント番長
(アーキテクト)

ドキュメント群

各ドキュメント間が
MECE

ドキュメントの過不足を明確にする

わかりやすいツリー構造で、ドキュメント全体を表現する

各ドキュメントに責務が集中することで、重要度・優先度がわかりやすくなる

各ドキュメント間がMECEであることで、重複や抜け漏れを防ぐ

おわりに:

書く前にできることはたくさんある

ドキュメントのライフサイクルを意識する

ドキュメントのライフサイクルを意識することで、低工数・高品質化

記述対象を「書きやすい」状態にする

書く側で頑張るよりも、書かれる側を変えてしまった方が、低工数・高品質

人手以外のリソースを活用する

論理が正しければ、何万回/ページでも瞬時に生成する & 低工数・高品質

まとめ

この講義で一番大切なこと

書くことだけがドキュメンテーション
ではない

この講義の全体ふりかえり

1. ドキュメントの品質を担保する

「人が書く」=コストがかかる

ドキュメントのライフサイクル

誰が書くべきか

2. ドキュメントのパフォーマンスを最大化する

「最小の工数」で「最大の効果」

記述対象を「書きやすい」状態にする

「人手」以外に書かせる

3. ドキュメントの過不足を明確にする

ドキュメントツリーの重要性

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

この講義から更にその先へ

- 第1回 ドキュメントの価値を知る ～ ドキュメンテーション事始め
- 第2回
(今回) ドキュメントの基本構造 ～ 楽するためのドキュメンテーション
- 第3回 **ドキュメントの創り方 ～ 「書く」だけがドキュメンテーションではない**
- 第4回 ITエンジニアのための「プロセス設計」入門