

株式会社テクノプロジェクト様

運用設計導入コース (第2回)

運用課題の分析

事前配布資料

運用設計ラーニング

2022-12-22

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



日本MSP協会 特別会員



- ・ **ADSLキャリア**で開局業務、ISP運用
- ・ **Sler**で官庁系サービスのサーバ保守
- ・ **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

アジェンダ

- ・ 導入1. 学習ガイド
- ・ 導入2. 運用あるある
- ・ 本編 当日配布します
- ・ まとめ

導入1

学習ガイド

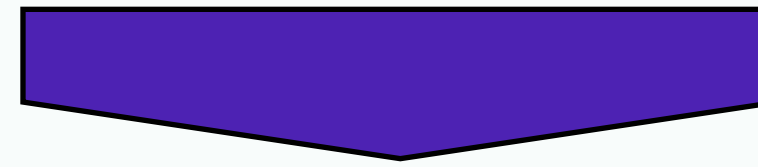
導入1

学習ガイド

OpsLearnの目的と全体像

OpsLearnの目的

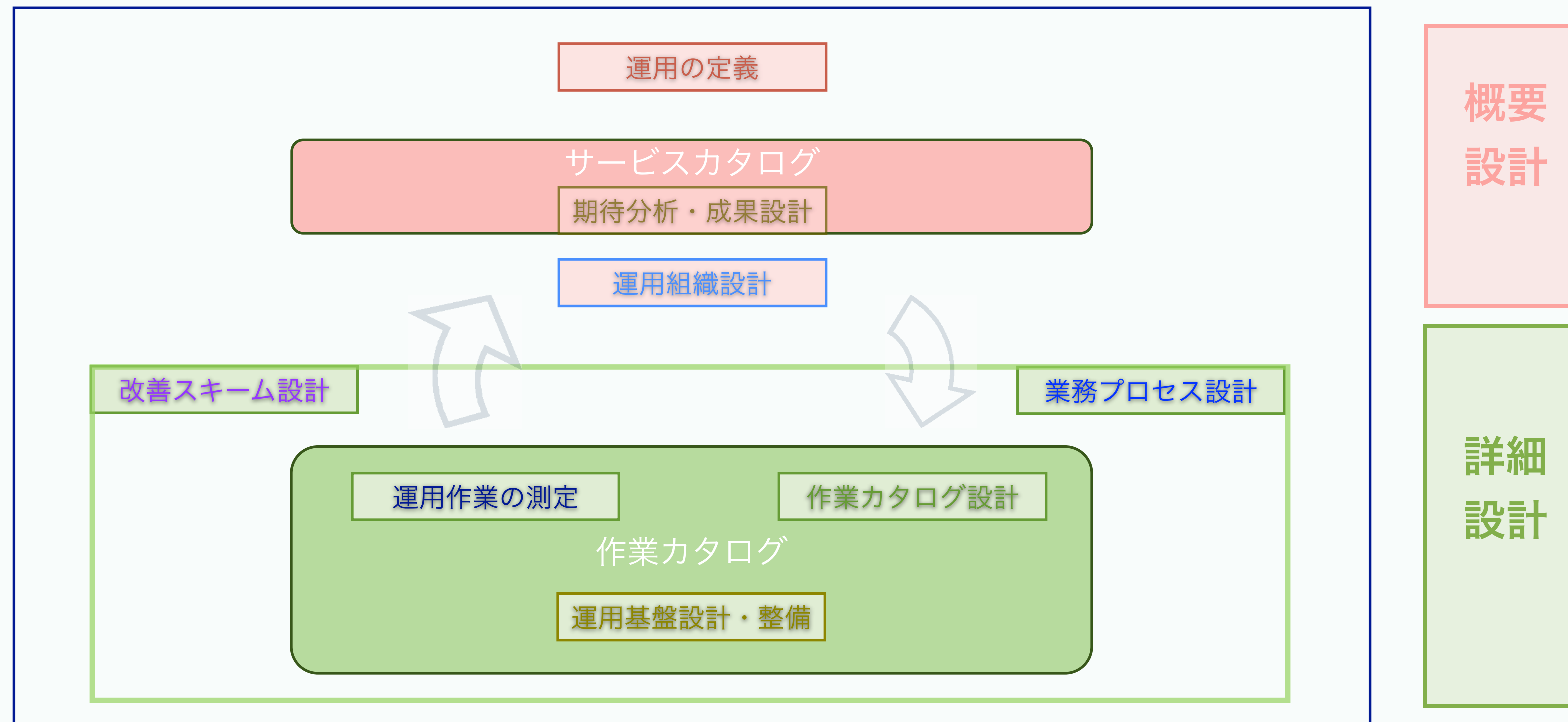
従来、現場ごとの個別事情によりやり方が異なるため、標準化が難しいと言われてきた「運用」



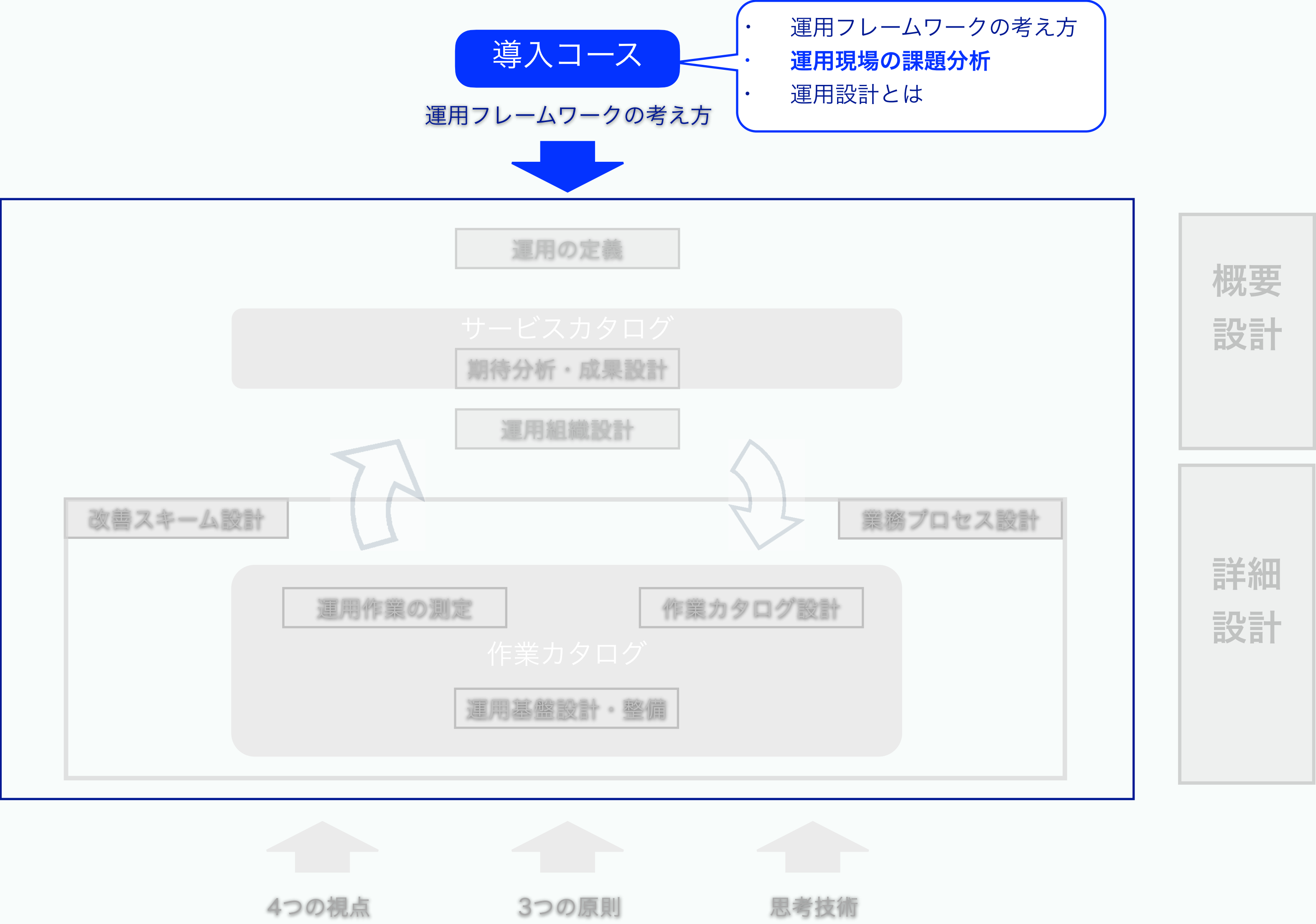
「運用現場視点による運用設計のための方法論」により
みなさんの運用現場における「実践的な運用設計」への取り組みを促進すること

- ▶ サービスの安定
社会基盤に相応しい安定運用。 → 「安定した運用」の実現
- ▶ 業務負荷の平準化
個々人ががんばりすぎなくともうまく業務が回る運用現場。 → 「楽な運用」の実現
- ▶ 運用に対する評価の適正化
適正な利潤を生む現場と、適切に評価される要員。 → 「稼ぐ運用」の実現

OpsLearnの全体像



運用設計導入コースの位置付け



導入1

学習ガイド

運用設計導入コースの概要

運用設計導入コースの概要

「運用設計」について本格的に学ぶ前に
運用業務や運用組織を客観的に把握して最適化するために必要な
「運用フレームワーク」の考え方を身に付けていきましょう。

サービスやシステムの運用に関与する幅広い人々(マネージャー、エンジニア、セールスパーソン)向けのコースです。

講義1

運用フレームワークの考え方

適切な運用設計を行うために、客観的かつ合理的に運用業務や運用組織を把握・整備する
「運用フレームワーク」という考え方について解説します。

講義2

運用課題の分析

第一回の「運用フレームワーク」の考え方に基づいて、
運用現場が抱える課題を分析していきます。

講義3

運用設計とは

第一回の「運用フレームワーク」の考え方に基づき、第二回の「運用課題の分析」で見えてきた運用課題の
根本原因を解消するための、あるべき「運用設計」について解説します。

運用設計導入コースの学習ポイント

導入コースは「抽象」的な内容が多くなります。

抽象 モデル

事物や表象を、ある性質・共通性・本質に着目し、それを抽（ひ）き出して把握すること。その際、他の不要な性質を排除する。

具体 実装(コード)

人間の感覚でとらえられるものであること。形や内容を備えていること。

空間的範囲	適用範囲が広い、適用対象が多い	適用範囲が狭い、適用対象が少ない
時間的範囲	寿命が長い、陳腐化が遅い	寿命が短い、陳腐化が早い
外部影響	影響を受けにくい、影響を吸収しやすい	影響を受けやすい、影響を吸収しにくい
実装との距離感	実装と遠い	実装に近い

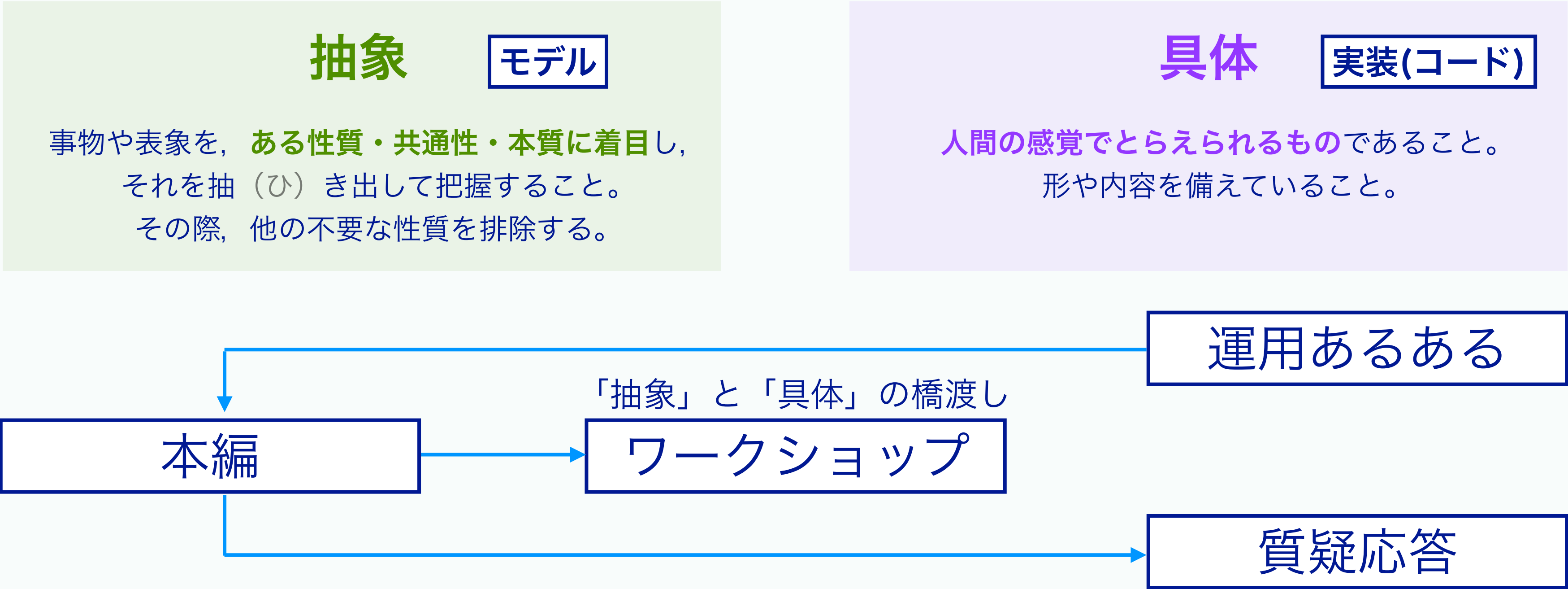
課題の本当の意味での解決には「抽象」的な分析が必須となります。

IT業界でも「具体的な話」を望む方が多いですが、ITは本質的に「抽象化(モデル化)」することに価値の比重があり、適切に抽象化されたモデルを「各現場に合わせて最適に具体化(実装)」することではじめて大きな効果をもたらします。

(自分たちの前提条件や環境と合致しない「具体的な話」を実践することは、むしろ弊害を招くことが多いです。)

運用設計導入コースの講座の構成

本編の「抽象」的な内容を「運用あるある」や質疑応答で「具体」的に補完します。



「抽象」的な分析や議論に慣れていきましょう。

あらゆる設計・プログラミングでは極めて重要な能力ですが、できる人があまり多くないのが現実です。

運用設計導入コース (第1回)

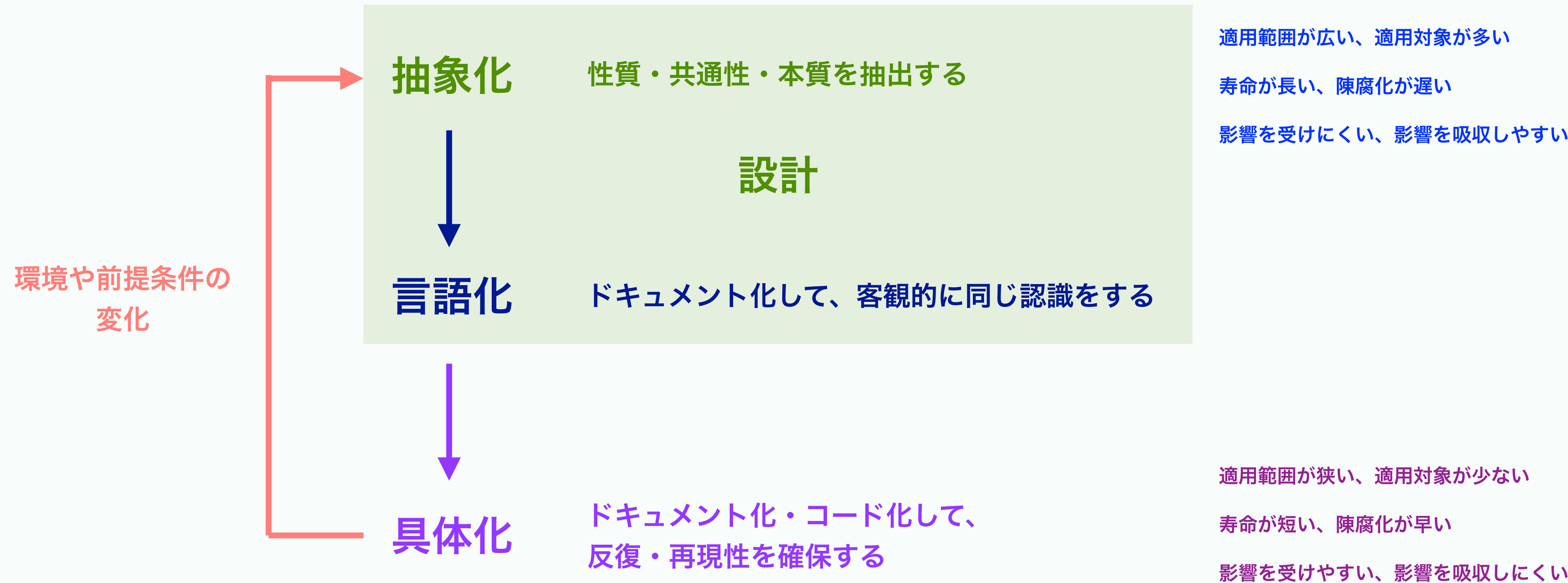
運用フレームワークという考え方 (補講)

運用設計ラーニング

2022-11-30

<https://www.opslearn.jp/>

補講: 設計とは「抽象化から始まる活動」



「抽象的 = よくわからない」認識のままでは、設計に着手することはできない。

補講: 運用フレームワークの実装イメージ

原則として、1つの運用フレームワーク = 1つの「運用ダッシュボード」



運用フレームワークの全ての要素は、運用ダッシュボードに掲載される。

補講: 運用フレームワークのカタチ

皆さんの運用現場に、運用ダッシュボードはありますか？



導入1

学習ガイド

今回の学習ポイント

今回の学習ポイント

インプット

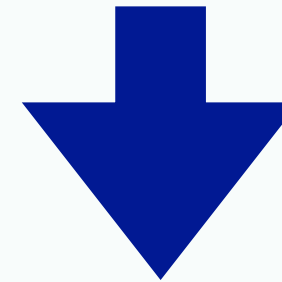
本講義で意識してほしいこと

- ・ 運用現場に多くの課題があり、現場がなかなか楽になっていかないのはなぜなのか。
- ・ 運用現場においてAsIsの認識のギャップが起こりやすいのはなぜか。

一次アウトプット

本講義から持ち帰ってほしい事

- ・ 「運用のAsIs」を把握する上での考え方
- ・ 「運用のAsIs」を分析する上での考え方
- ・ 「運用のAsIs」を共有する上での考え方



- ・ 運用のAsIsを把握・分析・共有するための仕組み作り
- ・ 自分達の「運用のAsIs」は今現在どうなっているか？

最終アウトプット

導入2

運用あるある ～ 講師の経験から

前史: データセンターに泊り込みで構築作業をする日々

状況

- ・ **サーバエンジニアとして、サーバの構築と運用を担当。**
 - ・ 数千台のサーバを2人チームx5で運用。毎月100台近く構築。
 - ・ 昼間はミーティングばかりで構築する時間は取れない。
 - ・ データセンターに泊まり込みで構築することも多かった。

監視スタッフとのコミュニケーション

- ・ **監視センターはデータセンター内で稼動。(24時間365日稼動)**
 - ・ スタッフと話すことが多かったので、たまに相談を受けたりしていた。
 - ・ 自分も構築と監視の一部を担っていた(監視専任エンジニアがいなかった)ので、できる範囲で対応したり、他の担当者にフィードバックしたりしていた。

監視スタッフと良好な関係にあった (当時の認識)

監視システム・監視センターの設計・運用主担当になる

状況

- ・ 監視システムも監視センターも主担当の社員が不在。
 - ・ 監視システムの構築プロジェクトが解散。インフラ担当がバラバラに登録・更新。
 - ・ 監視センターはアルバイト利用から、アウトソースに移行。

監視の破綻が明らかに見えていた

- ・ サーバの急激な増加によって、監視登録数が激増。
- ・ 監視対象の急増により、監視対応表が肥大化。
- ・ 監視登録の基準が無いので、アラートも激増。

監視システム

監視センター

システム担当者

自分から監視(システム+センター)の主担当を希望して異動

監視の主担当として、監視の再設計と課題対応に取り組む

状況

- ・ **監視システムと監視センターの両方を抜本的に見直す立場**
 - ・ 監視システム全ての主担当としてもらった。
 - ・ 監視センターのアウトソース契約を変更して、業務設計責任を取り戻した。
 - ・ 監視に関わる全ての人が、仕事相手になった。

全てが混沌としている状態からスタート

- ・ **勝ち目は見えていた**
 - ・ 監視システム～監視センターの監視プロセスの整備
 - ・ 監視アラートのルーティング整備
 - ・ 監視対応のルール整理

監視システム

システム担当者

監視センター

「監視」を抽象化(モデル化)して、実装を進めていった

監視センターの現場とのすりあわせ

状況

- ・ **監視センターとその設計者は拠点が別**
 - ・ 設計者(講師)は、本社勤務。
 - ・ 監視センターは、データセンター内に設置。
 - ・ 基本的に監視リーダーと、チャットかメールでコミュニケーション。(たまに電話)

現場のことは現場にしかわからない

- ・ **週1回、監視センターでリーダーと課題ミーティングを実施**
 - ・ 課題管理表を共有し、双方が課題を持ち寄る。
 - ・ 毎回2時間程度議論をして、双方持ち帰り、課題解消を進める。
 - ・ 時間があれば、監視スタッフと雑談をするようにした。

現場の声を聞きながら、課題の解決を進めていった (当時の認識)

同僚には「現地に行く必要が無い」と批判的な人もいた。

課題の対象

監視システムの構築・運用に関する課題

- ・ 一次情報の半数が監視センターにある。
- ・ 監視の現実は、監視センターの方がよくわかっている。

監視センター業務の設計に関する課題

- ・ 既存業務の洗い出しからする必要があった。
- ・ アンドキュメンテッドな業務や対応も多かった。

監視センターの悩みごと

- ・ アラートが多過ぎる、ミスが頻発、無茶な依頼が多い、など。
- ・ 監視スタッフのキャリアパスの悩み、なども。

現場の声を聞きながら、課題の解決を進めていった（当時の認識）

導入2 運用あるある

講師が「運用あるある」で経験したこと

経験したこと1. 「改善される側」の改善疲れ

現場に話を聞きに行けば、**すぐ情報が集まるし、改善のフィードバックも得られる**と思っていた。

実際には

- ・ リーダーからは情報が得られるが、シフトメンバーからはなかなか情報が来なかった。
- ・ 過去の「運用改善」で疲弊した経験から、課題への取り組みに前向きでない。(またか、という空気感)
- ・ 「運用改善」の結果についても、否定的な情報を上げずに、現場でなんとかする、という空気。

現場は「改善」に懐疑的で、協力する**リスクや工数に敏感**だった。

経験したこと2. 良かれとおもってやったことが逆効果

自分は監視経験も長いため「こうすれば現場を楽しになって、喜ばれるに違いない」
とっていた。

実際には

- ・ 業務フローが自分の把握していた(つもり)とは異なり、改善後に監視スタッフが運用でカバーしていた。
- ・ 業務はきれいになったが、監視センターの負荷は上昇していた。
- ・ 見た目の課題は解決したが、「現場が望んでいた解決」にはなっていなかった。

「思い込み」で突っ走り、現場に多大な迷惑を掛けた。

経験したこと3. 現場とすぐ乖離する

自分が業務設計を主導し、監視経験も長いため「現場をちゃんと把握している」
とっていた。

実際には

- ・ 週1の訪問時に、シフトスタッフからはじめて聞く現実が少なくなかった。
- ・ シフトスタッフとの雑談の中で、監視リーダーも把握してない現状が見えてくる事も少なくなかった。
- ・ 他の業務で忙しく課題への対応が止まっている間に、状況は変わり、数週間前の現実、今現在の現実ではなくなっていることも珍しくなかった。

「自分は現場を把握している」は半分幻想だった。

まとめ: 経験したこと

1. 「改善される側」の改善疲れ

現場は「改善」に懐疑的で、協力するリスクや工数に敏感だった。

2. 良かれとおもってやったことが逆効果

「思い込み」で突っ走り、現場に多大な迷惑を掛けた。

3. 現場とすぐ乖離する

「自分は現場を把握している」は半分幻想だった。

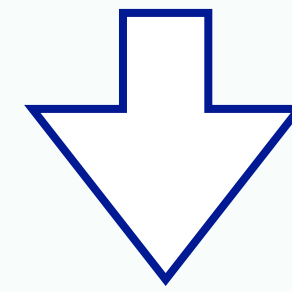
導入2 運用あるある

講師が「運用あるある」で気付いたこと

気付いたこと1. 「現場からの信頼」が課題解決の大前提

最初は「信頼してもらえない」が当たり前

- ・メンバー**全員とOne on One**で面談してから空気が変わった。
 - ・「この人なら、どうやらちゃんと改善してくれそうだ。」と思ってくれた感触。
- ・何かあったら、スタッフから**24時間、電話でもチャットでも直接連絡**をもらうようにした。
 - ・監視センターのラストリゾート(何かあったら、必ず対処してくれる人)を講師が担うようにした。



「きちんと話を聞いて活かす。」 「改善実績を上げる。」ことが大事

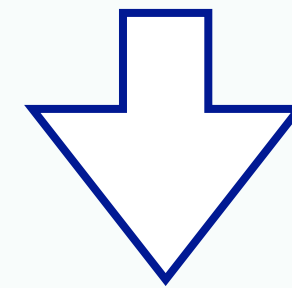
- ・改善施策の**ネガティブな面**について、フィードバックが得られるようになった。
- ・「良くなった？」って聞くよりも先に、「良くなりました」と向こうから言ってくるようになった。

信頼してもらうためには、時間と実績が必要。

気付いたこと2. 「思い込み」 駆動の課題解決は厳禁

自分に良いアイデアがあると、すぐに実施したくなる。
目の前に「課題」があると、どうしても解決したくなる。

- ・ アイデアや解決策が浮んでも一旦冷却して、現場や第三者にレビューしてもらうようにした。
- ・ でも、思い付くと、つい実装してしまうことは度々あり、いくつかは失敗してしまった...



「客観的・複眼的に課題を分析する。」ことが大事

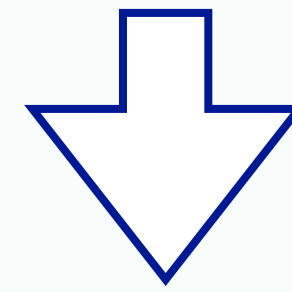
- ・ 設計者が「良い解決案」だと直感しても、実際の課題を解決できるとは限らない。
- ・ 一番大事なのは、設計者と現場の双方が「勝ち目がある。」と思うこと。

現場の課題を解決しない「改善」や「自動化」には全く価値が無い。

気付いたこと3. 「現場との乖離」が前提

「現場の人」と「現場の人ではない人」の間の乖離は不可避

- ・ 前提条件や環境の変化によって、**認識の乖離は容易に発生**する。
- ・ 同じ部屋にいても、「実際にやっている人」以外は、**時間の経過により現状認識に乖離が拡大**する。



「**設計者と現場の乖離は起こる前提**」で考えることが大事

- ・ いかに認識の乖離を最小化するか？ (現場とのコミュニケーションの質)
- ・ 仮に乖離があったとしても影響を最小化するためにどうするか？ (設計上のバッファの考え方)
- ・ 「設計と現実の乖離」であれば、現場にアラートを上げてもらうことはできる。

「**設計でカバーする**」という意識を設計者も現場も持つべき。

まとめ: 気付いたこと

1. 「現場からの信頼」が課題解決の大前提

現場は「改善」に懐疑的で、協力するリスクや工数に敏感だった。
「きちんと話を聞いて活かす。」「改善実績を上げる。」ことが大事。

2. 「思い込み」駆動の課題解決は厳禁

「思い込み」で突っ走り、現場に多大な迷惑を掛けた。
「客観的・複眼的に課題を分析する。」ことが大事。

3. 「現場との乖離」が前提

「自分は現場を把握している」は半分幻想だった。
「設計者と現場の乖離は起こる前提」で考えることが大事。

導入2 運用あるある

見えてきた「運用現場における悩み」

よくある「運用現場における悩み」

- ✓ 業務が多岐に渡り、全てを把握することが困難になっている。
- ✓ ドキュメントが整備されていない。あっても更新されていない。
- ✓ どんなドキュメントが必要なのがわからない。書き方がわからない。
- ✓ 一部の人間にしかできない業務があり、業務が集中している。
- ✓ 属人化が進み、ノウハウの継承ができていない。
- ✓ 異動により現場が混乱することが多い。

- ✓ 人が育たない。優秀な人が入ってこない、定着しない。
- ✓ がんばっても評価されない。
- ✓ トラブルが多く、前向きな改善に着手する余裕がない。
- ✓ ツールが使いにくい、改修にはコストと期間が必要なため我慢して使っている。
- ✓ 新規のツールを設計したいが、どんな要求があるのか現場でもわかっていない。

- ✓ サービス設計導入時の検討漏れや実装が間にあわない部分を「運用でカバーする」など設計側のその場しのぎの影響を直接受ける。
- ✓ 依頼されてから動き出すまでのリードタイムが長い。
- ✓ 声の大きいユーザが強く、必要以上のサポートを強いられる。
- ✓ コスト削減要求が強いが、どう効率化すべきなのかが見えない。

よくある「運用現場における悩み」

- ✓ 業務が多岐に渡り、全てを把握することが困難になっている。
- ✓ ドキュメントが整備されていない。あっても更新されていない。
- ✓ どんなドキュメントが必要なかがわからない。書き方がわからない。
- ✓ 一部の人間にしかできない業務があり、業務が集中している。
- ✓ 属人化が進み、ノウハウの継承ができていない。
- ✓ 異動により現場が混乱することが多い。

- ✓ 人が育たない。優秀な人が入ってこない、定着しない。
- ✓ がんばっても評価されない。
- ✓ トラブルが多く、前向きな改善に着手する余裕がない。
- ✓ ツールが使いにくい、改修にはコストと期間が必要なため我慢して使っている。
- ✓ 新規のツールを設計したいが、どんな要求があるのか現場でもわかっていない。

- ✓ サービス設計導入時の検討漏れや実装が間にあわない部分を「運用でカバーする」など設計側のその場しのぎの影響を直接受ける。
- ✓ 依頼されてから動き出すまでのリードタイムが長い。
- ✓ 声の大きいユーザが強く、必要以上のサポートを強いられる。
- ✓ コスト削減要求が強いが、どう効率化すべきなのかが見えない。

実は「運用現場の悩み」は共通

- ・ **多くの現場が似たようなことで悩んでいる**
 - ・ 実は自分のところだけじゃない。
- ・ **多くの悩みに共通の要素**
 - ・ 複雑そうに見える悩みを解きほぐしていく必要がある。

参考: 運用現場における現実

多くの運用現場の人々は、「自分たちだけが苦勞し悩んで」いて、それが「自分たちの努力不足のため」と考えているようです。

しかし、上記の現場の声にあるように、実は多くの現場が似たような悩みを抱えています。程度の差はあれど、大企業、ベンチャー、上位レイヤーのWebサービス企業、下位レイヤーのインフラ企業を問わず、国内のIT基盤運用現場においては、非常に似た悩みを抱えて同じように苦しんでいるのが実情ではないでしょうか。

連載：現場視点からの運用方法論

第1回 見えない「運用」 - 疲弊する運用現場

<http://thinkit.co.jp/story/2010/12/02/1903>