超上流から始めるシステム設計とは！
～ビジネス創出・要求・要件・設計そして実装まで一気通貫を目指そう～

2024年2月22日

ITエンジニア／コンサルタント
赤　俊哉
E-mail:toshiya.seki28496@gmail.com
Facebook・Linkedin・X

自己紹介

元ホームレス寸前から下請けプログラマー、SEとしてIT業界の最下層に入る。

ベンダー企業のSE/プログラマー、ユーザー企業システム担当、利用部門、
ベンダー企業PMといったあらゆる立場からIT／情報システムに関わり
“叩き上げシステム屋”としてキャリアを築く。

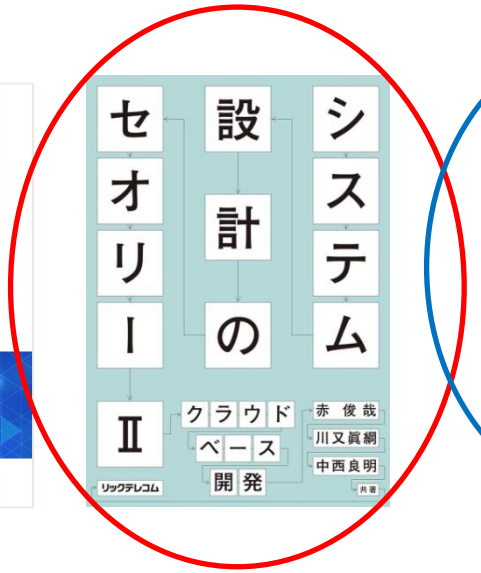
様々な悲惨な？経験から上流工程の重要性について痛感するようになる。
さらにデータ中心でビジネス、そしてシステムを作り上げていくことが低迷する
日本のIT復活の処方箋であると考えるようになる。

生まれ変わったとしても情報システムにかかわる仕事をしたいと考えている。

得意分野

- （超）上流工程全般
  - ✓ ビジネス/IT企画策定
  - ✓ ビジネス／業務分析
  - ✓ 要求分析／要求定義
  - ✓ 要件定義
  - ✓ 広義のシステム設計（IT設計・ビジネス／業務設計）
- データマネジメント全般
  - ✓ データ中心型ビジネスアプローチ（DxBA　通称DOBA）
- プロジェクトマネジメント支援

著作について



- 著書：「SE職場の真実」（日経BP）
  「システム設計のセオリー」
  「要件定義のセオリー」　（リックテレコム）
- 共著：「データ経営が日本を変える！」（JUAS出版）
  「システム設計のセオリーⅡ－クラウドベース開発」（リックテレコム）
- 監修：「データマネジメントの実態と最新動向2024」

- 連載記事に「どんづまりから見上げた空」（日経xTECH）があるほか、執筆記事多数。

Agenda

1. データ中心でいこう！
2. ビジネスとITの一体化を目指そう！
3. 全てはコミュニケーション！
4. DxBA（通称DOBA）の実践
5. 超上流から「設計」を始めよう！

Agenda

1. データ中心でいこう！
2. ビジネスとITの一体化を目指そう！
3. 全てはコミュニケーション！
4. DxBA（通称DOBA）の実践
5. 超上流から「設計」を始めよう！

最初から・・・素朴な疑問

- **「DXやれ！」といわれてシステム開発を行う際に**
  データを重要視しないように見えるのは何故？

✓ データ分析してビジネスへ反映、はいいけど、そもそもシステム作る時には
データをあんまり意識してない？

✓ DXではデータモデル（ビジネス地図）の話は話題にならない？

✓ DXだろうが例外なく、データのあり方をきちんと検討しない仕組みは確実に
破綻するのでは？　その場しのぎでいい？

何故データが軽視されているか？

- **「データ（ドリブン）経営」とは、データをいじってなんとかするもの？**

✓データの本質的な意味を理解せずにデータ分析しても・・・

✓あらゆるところから一生懸命データかき集めても・・・

✓何故仕組み（システム）を作る時からデータに着目しないのか？

データ（特にデータモデル）関係者にも責任がある！

- **「データ経営が日本を変える！」**
  - ✓ Ⅷ章 大切なデータを生かすデータモデルとは～データモデルの価値と課題
  - ✓ Ⅷ．９ データモデルなんて嫌い？

JUAS　Webサイトより全文Pdfにてダウンロードいただけます。
https://juas.or.jp/library/research_rpt/various/#data

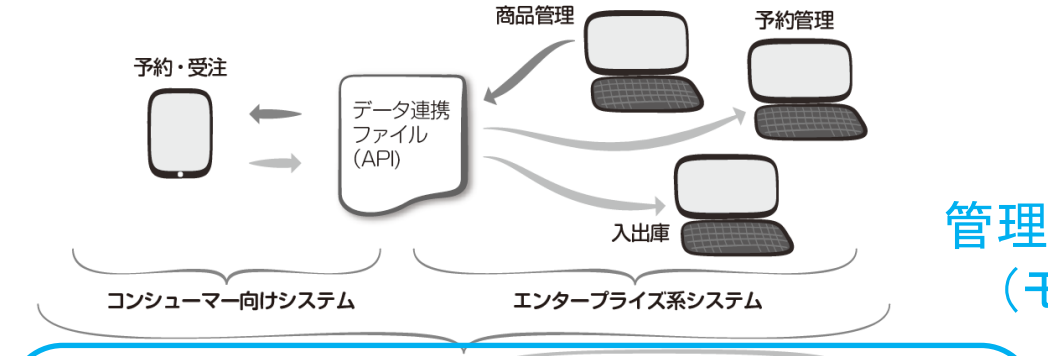
Amazon　Kindle版はこちら。
https://www.amazon.co.jp/dp/B0BB2D9B5X

作る時からデータの繋がりを確保することが重要！

- **データの繋がりを意識して開発しよう！**
  - ✓ データの繋がりをすべてデータの観点から管理する
  - ✓ 同じくデータの繋がりをすべてプロセスの観点から管理する

＊データの観点

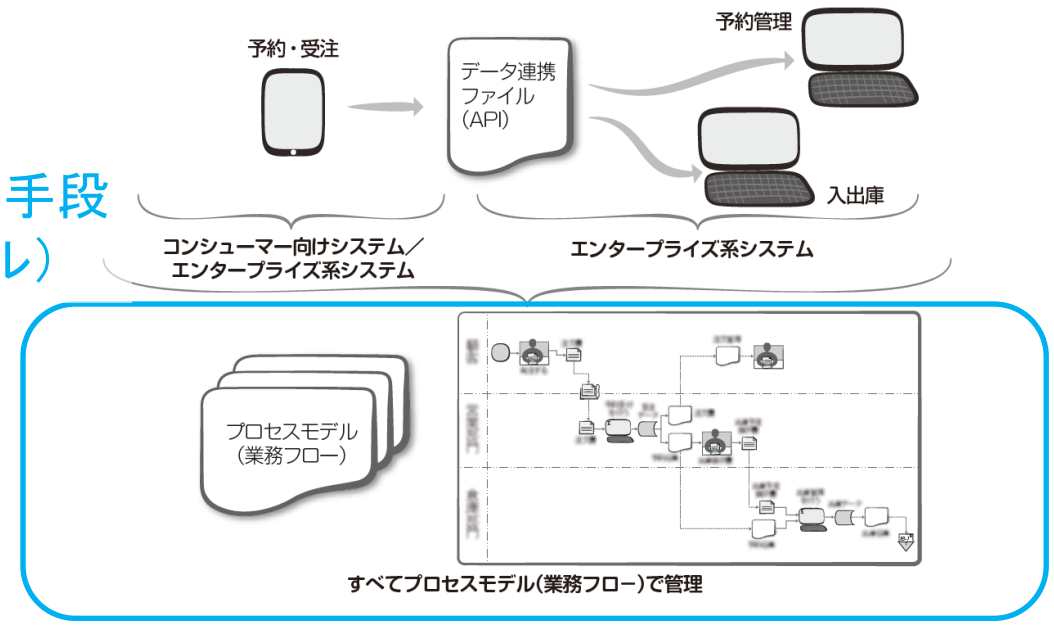
図3-6　「データのつながり」を確保する



管理する手段
（モデル）

＊プロセスの観点

図4-7　「データのつながり」を確保する



「システム設計のセオリーⅡ　クラウドベース開発」より

情報システムの使命を果たすために

- **情報システムの使命**

「適切な時に、適切な人（場所）が、適切な（品質）の情報を

ｉｎｐｕｔ（入力）することにより、必要な時に、必要な人（場所）へ、

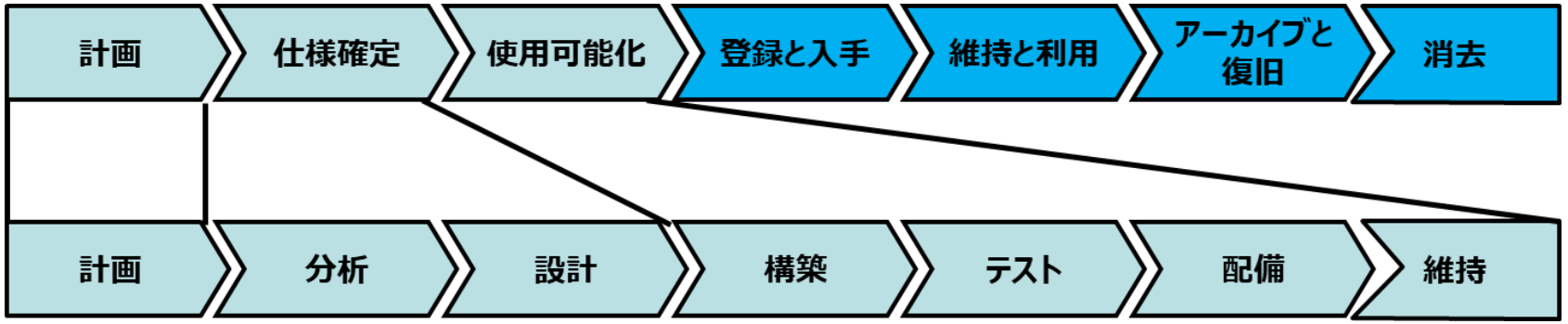
必要な（品質）の情報を　ｏｕｔｐｕｔ（出力）する」 為に存在する

「システム設計のセオリー」より

- ✓ ビジネス ／業務と情報システムを繋ぐのは「データ」であり「プロセス」だけではどうにもならない
- ✓ データライフサイクルはシステムライフサイクルより長い⇒ビジネスをデータ中心で考えることは必然
- ✓ **後述するデータ中心型ビジネスアプローチ（DxBA）が有効**

<データライフサイクル>

<システムライフサイクル>



「データマネジメント知識体系ガイド　第一版」より

（ちょっと脱線）今流行の市民開発の課題

- ノーコード/ローコード開発ツールの普及により市民開発がより簡易に可能になった！
- ✓ 企業、自治体問わず日本中でアプリが開発された！
- ✓ それ自体は素晴らしいこと！

- でも課題もでてきた
- ✓ シャドーIT？
- ✓ それ以前にデータの氾濫を防ぐ手がない！
- ✓ 簡易な開発ツールで開発する時こそまずデータ中心で考える必要がある！

（さらに脱線）日本にはそもそもデータ標準がたりない！

- 「避難所情報が合わない」、被災自治体と自衛隊が直面したデータ集約の壁

✓ https://xtech.nikkei.com/atcl/nxt/column/18/02745/021300001/

- 欧州ではデータスペースが大注目されている！

✓ データスペースとは、国境や分野の壁を越えた新しい経済空間、社会活動の空間のこと

✓ 日本は圧倒的に出遅れている！データが繋がらなければ世界から相手にされなくなる！

✓ 企業内だけのデータ標準化ではもう駄目！

✓ 社外含むデータの標準化が必要

⇒今こそ後述するデータ中心型ビジネスアプローチ(DxBA　通称DOBA)実践の時！

Agenda

1. データ中心でいこう！
2. ビジネスとITの一体化を目指そう！
3. 全てはコミュニケーション！
4. DxBA（通称DOBA）の実践
5. 超上流から「設計」を始めよう！

システム設計（というよりセオリーシリーズ）の基本方針

システム開発を成功させるために

- 「最小の労力で、最大の効果を実現する」
- 「成果物は、より少なく、しかしより良く」

を目指す！

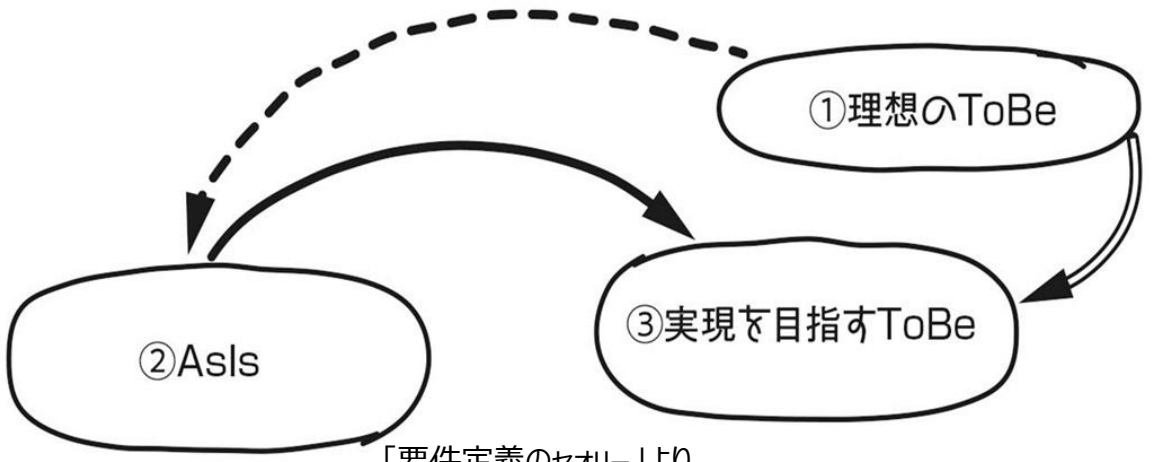
“やることはやってるはずなのにうまくいかない”
“やることが多すぎてやりたくても時間がない”
“なんか似たような作業をしている”
“何の役にたつのかわからないドキュメントを作ることが目的になっている”
“ドキュメントが役に立たない”

開発の方向性によりアプローチは異なる！

- ToBe(あるべき姿）指向か
- AsIs（今ある姿）指向か

✓ 「改革／DX」と「更改／改善」ではアプローチは異なる

- ①理想のToBe
  ⇒「目指すべき姿」
- ②AsIs
  ⇒「今ある姿」
- ③実現を目指すToBe
  ⇒「実現すべき姿」



「要件定義のセオリー」より

ToBeとAsIs

- 改革／DX

✓ ①目指すべき理想のToBeを考え抜き、「要求」としてまとめる

✓ ②AsIs 分析を通じて現状、制約、前提を加味する

✓ ③実現を目指すための新しいToBeを作成し、「要件」としてまとめる

- 更改／改善

✓ ①まずAsIs分析にて現状把握に努める

✓ ②現状を踏まえて理想のToBeを考え抜き、「要求」としてまとめる

✓ ③実現を目指すための新しいToBeを作成し、「要件」としてまとめる

ここで改めて・・・「要求」について考えてみよう！

- 例えどんな作り方をしようが、
  (例えば「ウオータフォール型開発」だろうが、
  「アジャイル型開発」だろうが、)

「何がしたいか」＝要求
わからなければ始まらない！

✓ DXだろうがなんだろうが要求がシステム開発の起点
✓ というよりあらゆるモノ、コトは全て要求が起点

要求を”分析する”ために！

- デジタル時代の要求はどうやって明らかにすべき？

✓ デジタルとビジネスがどんどん一体化している！
✓ ビジネスの源流まで遡らないと真の要求はわからない
✓ 要求を分析する際には従来通りだけではなく一工夫が必要？

✓ 要求分析で行うべきはどんな時代になろうとも「目指すべき姿」を明らかにすること

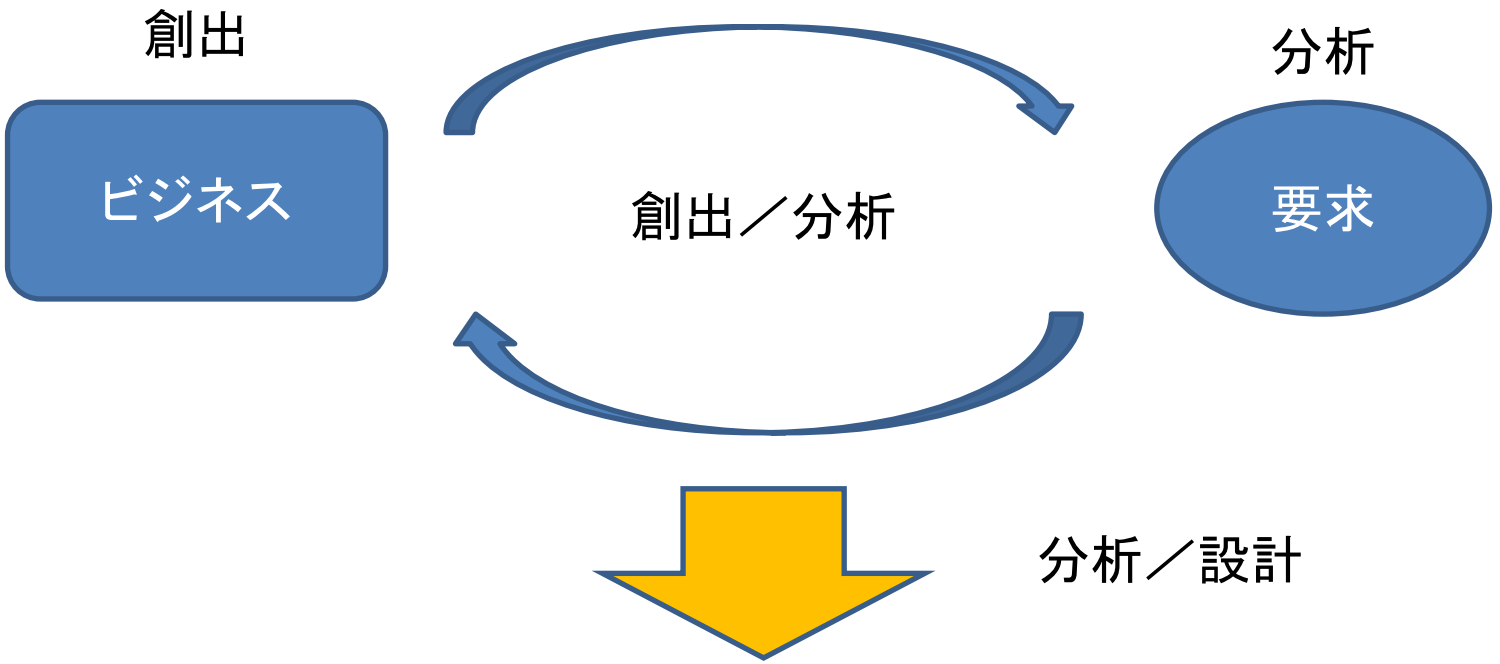
要求分析のパターン再確認

- デジタル時代の要求分析2パターン

✓ ビジネス創出と要求分析同時併行パターン
✓ （従来通りの）ビジネスありき⇒要求分析パターン

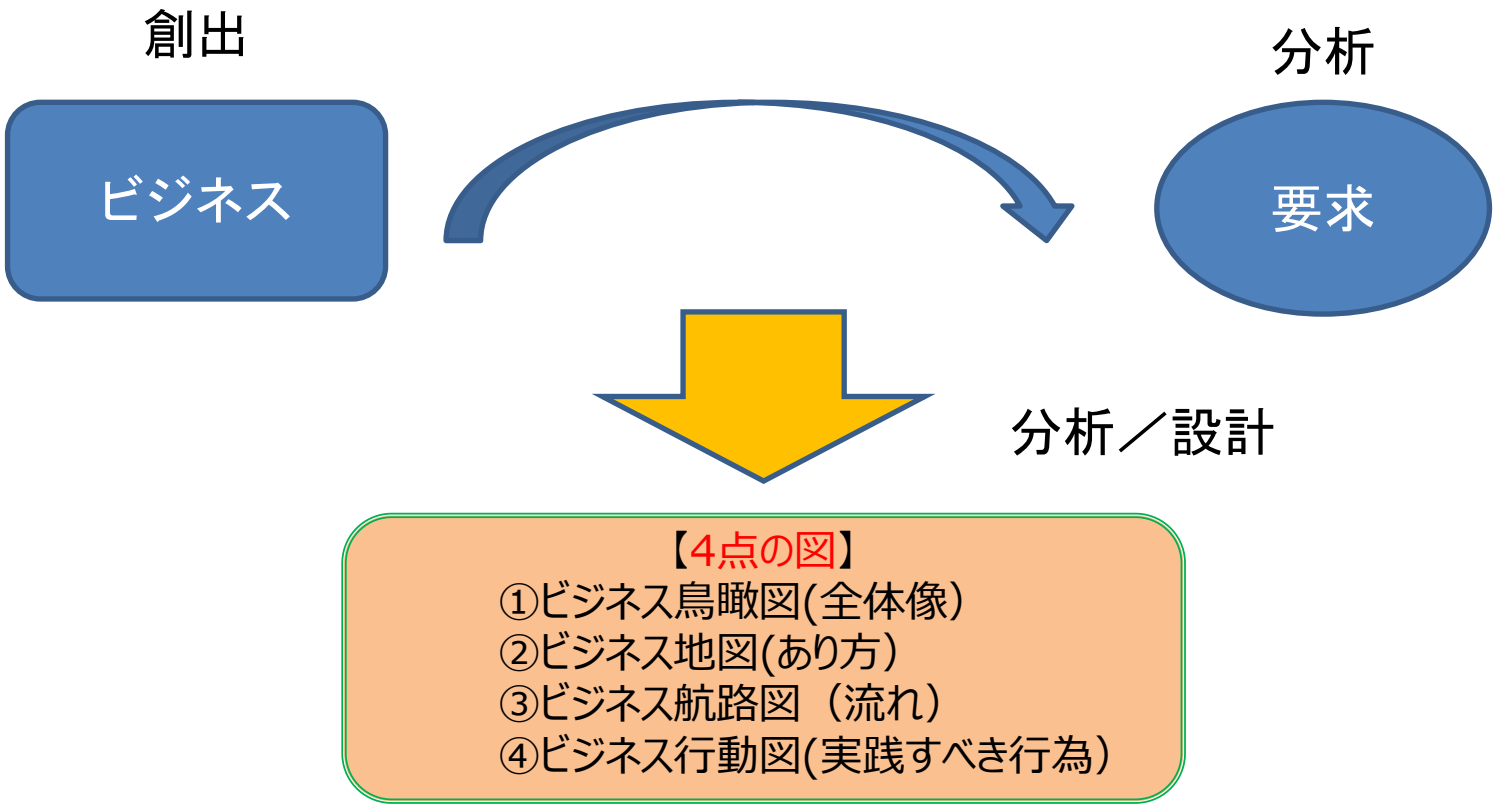
✓ 「ビジネス分析と要求分析一体」パターンと従来通りの「ビジネス⇒要求分析」パターン、いずれにおいても目指すべきビジネスの姿をきちんと把握することに意味がある
✓ いずれの場合においても後述する「4点の図」の作成は有効！

パターン1：ビジネス創出と併行して要求分析



【4点の図】
①ビジネス鳥瞰図(全体像)
②ビジネス地図(あり方)
③ビジネス航路図（流れ）
④ビジネス行動図(実践すべき行為)

パターン2：ビジネスありきで要求分析(従来パターン)



★　この場合でも目指すべきビジネスの姿をきちんと把握することに意味がある。

要求分析に限らない課題

- いつもの課題に直面！

✓「ビジネスとITがわかる人がいない」

✓「両方わかる”デジタル人材“を育てよう！」

・・・・・でも育てるのはいいけど今いないのはどうする？

✓ いないのであればビジネスがわかる人とITがわかる人が一緒になってビジネス及びITシステムを作り上げるしかない！

✓ ではどうやって？　私なりの回答は後程！

デジタル時代の要求とは

. 確実に要求は多様化している！

✓ ステークホルダーより要望を抽出し要求を洗い出していく
たしかにそんな従来からのアプローチでも大丈夫なシステム開発もたくさんある
✓ しかし、デジタル時代の要求はそれだけでは不十分
従来通りの分析をするだけでは要求は明らかにならない場合が多い

✓ 技術の進化に伴い、ビジネスの要求とITの要求が一体化していることが多くなってきた
✓ ビジネス創出と併行して「要求を作り上げる」必要がある
✓ 従来のやり方だけでは通用しないことを認識する必要がある

ここで基本にかえって用語の整理

- 要望・要求・要件とは
  - ✓ 要望とは、やりたいことの元ネタである意見、考え等を指す
  - ✓ 思いつきのレベルであることが多い
    - －要望同士は相反したり矛盾しているものが混在している状態
  - ✓ 要求とは、やりたいこと,目指すべきことを指す **「目指すべき姿」**
    - －要求同士の整合性が取れている状態
  - ✓ 要件とは、実現すべきことを指す **「実現すべき姿」**
    - －当然、要求同様整合性の確保は必須

✓ 要求・要件は与えられるものではなく、定義する、作り上げるものである

「要件定義のセオリー」より

要望、要求から要件へ

- 要望は取捨選択、整理、変換を経て要求になる
- 要求は制約、前提を踏まえた上で取捨選択、整理、変換を経て要件になる

✓ 制約とは、枠を設けて、自分の思うままに活動をしたり、物事を成り立たせたりできないようにすることを指す
「動かせない、変更できないもの」

✓ 前提とは、何かを成り立たせるために必要な事柄を指す
「動かす余地があるもの」

代表的な制約

- 代表的な制約（例）

✓「人」:

✓「物」:

✓「金」:

✓「時間」:

✓「技術的困難」:

✓「技術的制約」:

✓「現状との乖離」:

要件定義とはなんなんだろうか

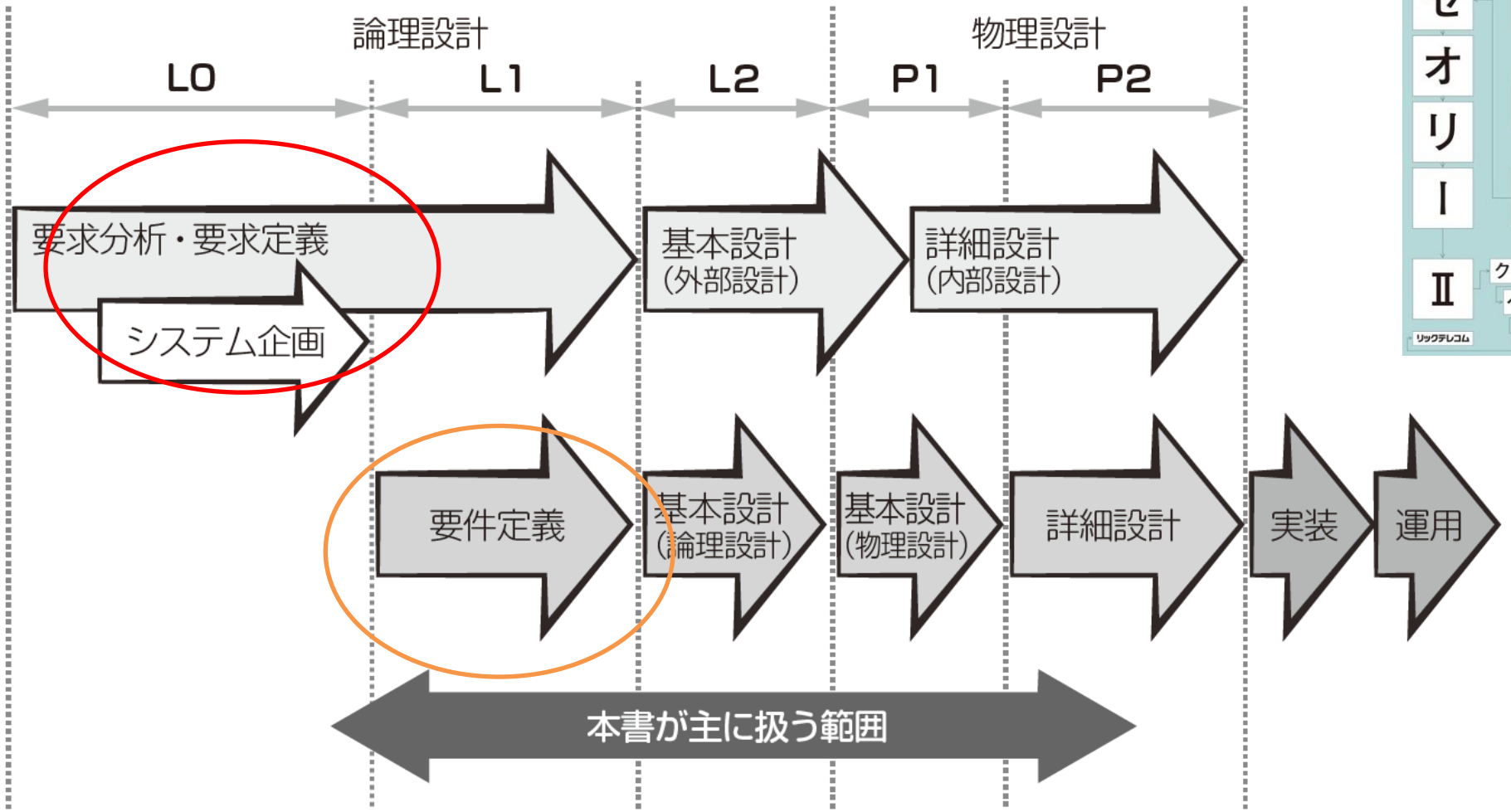
- 「何をしたいか」がわからなければ
  「どうしてよいか」はわからない

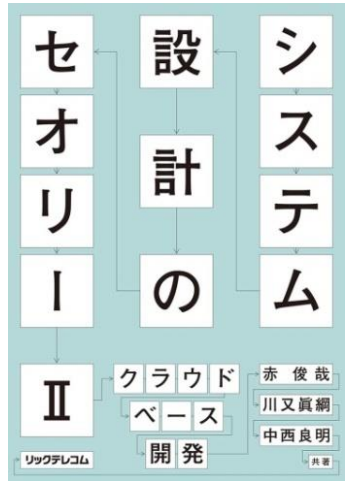
✓ 「何をしたいか」 ⇒ 要求 **「目指すべき姿」**
✓ 「どうしてよいか」 ⇒ 設計・製造・実装
✓ この2つをつなぐのが要件、それを明らかにするのが要件定義
“「実現すべき姿」を明らかにする”

✓ デジタル時代はビジネス創出からIT、システムに対する要求は発生する
✓ ビジネスが変われば当然要求も変化していく
✓ 要求の変化に伴い要件も変化していく

開発手法による違い：ウオーターフォール型の場合

図1-6　ウォーターフォール型開発の場合

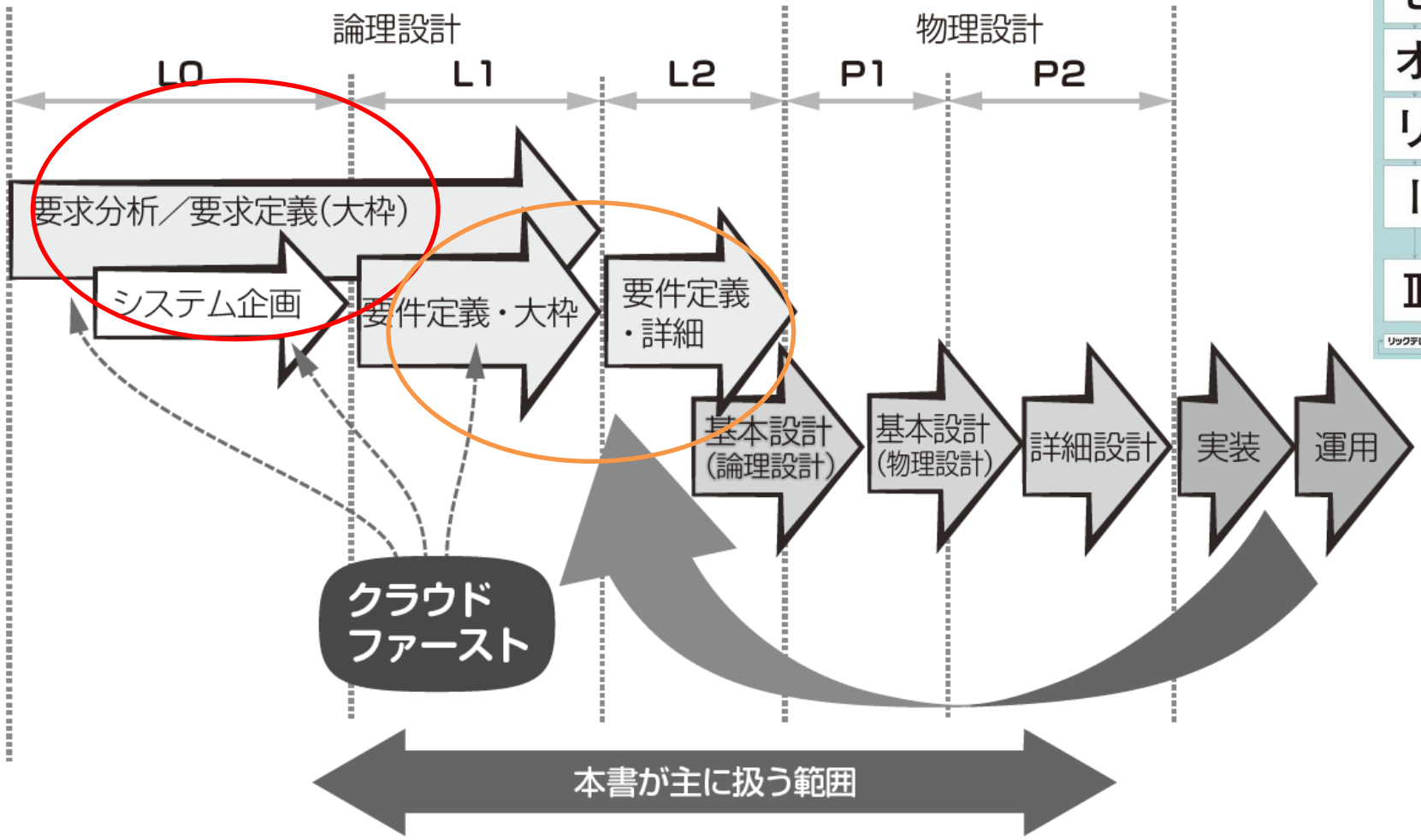


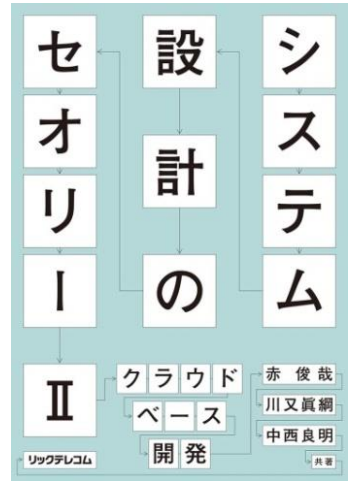


「システム設計のセオリーⅡ　クラウドベース開発」より

開発手法による違い：アジャイル型の場合

図1-7　アジャイル型開発の場合





「システム設計のセオリーⅡ　クラウドベース開発」より

要件定義をやめる？

- 「要件定義をやめよう」の真意、普通にやると金と時間が無駄になるだけ
https://xtech.nikkei.com/atcl/nxt/column/18/00166/091900134/
- ミスリードするという意見を受け、改めて「要件定義をやめよう」の真意を説明する
https://xtech.nikkei.com/atcl/nxt/column/18/00166/100300135/URL

✓ 要望をそのまま要件化した不毛な要件定義書作りはやめるべき
✓ 「本当にやりたいこと」すなわち「目指すべき姿」から「実現すべき姿」へのシームレスな移行を可能にする必要がある
✓ 「実現すべき姿」を明らかにするために、余計なことはやらずに必要なことをやるべき
✓ 本当にやめる必要はなさそう？

要求・要件のトレーサビリティを確保しよう！

- 要求から要件へのトレースを可能にしておくことは重要

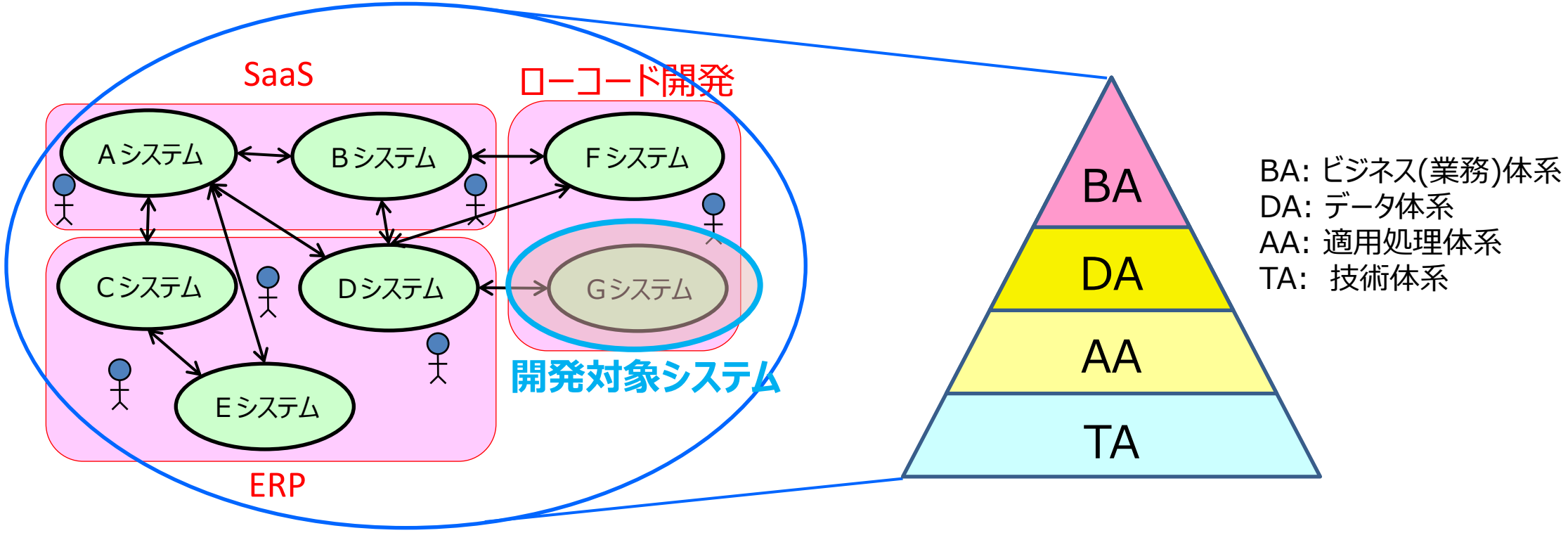
✓ 何故要件となったか説明できるようにしておく。

✓ 何故要件とならなかったかについても説明できるようにしておく。

✓「目指すべき姿」から「実現すべき姿」にいかに遷移したかわかるようにしておく

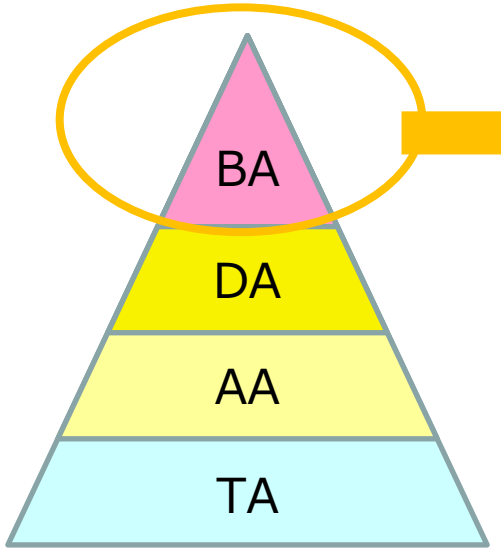
エンタープライズアーキテクチャ（EA）を意識する！

- **エンタープライズアーキテクチャ（EA）は全社の「広義の」システム設計図！**
  - ✓ **開発プロジェクト単体だけを体系化しても意味がない！プログラム視点を持つことが重要！**
  - ✓ **個別のシステム設計を行う際に全体を意識する必要がある！**



日本のEAに対する理解の致命的な間違いを正す

- **《大問題》 ビジネス アーキテクチャが“プロセスアーキテクチャ”になってしまっている！**
  - ✓ ビジネス/業務体系（BA）は“ビジネス”を対象するはずなのにプロセスのことしか考えない人が数多くいる
  - ✓ これは明らかな旧来からのプロセス指向アプローチであり、DXなんて到底無理
  - ✓ ビジネスはプロセスだけでなく**データ**とプロセスの組み合わせで成り立っている
  - ✓ 必然的にビジネスアーキテクチャ（BA）おいてビジネスの全体像把握とともにビジネス実現に必要なデータ、プロセス、そしてデータとプロセスの関係性までを明らかにして体系化していく必要がある



○ビジネスアーキテクチャ
×プロセスアーキテクチャ
★①全体像②データ③プロセス
④データとプロセスの関係性 の体系化は必須

Agenda

1. データ中心でいこう！
2. ビジネスとITの一体化を目指そう！
3. 全てはコミュニケーション！
4. DxBA（通称DOBA）の実践
5. 超上流から「設計」を始めよう！

“広義のシステム設計”含む上流工程とは

- 上流工程とは

✓「システムのプロと業務のプロとの間で相互翻訳作業を行い開発対象のシステム構想を作り上げる工程である」

✓「システム屋と業務屋とのコミュニケーションを円滑にして、システムで実現すべき姿を固めていく工程」

「要件定義のセオリー」より

ともいえる

✓ 立場の異なる人同士の共同作業であるという認識を持つ必要がある！

例えば要件定義においても・・・・・

- 確定した要件すなわち「実現すべき姿」については、
  ステークホルダー間で合意を得る必要がある

✓ ステークホルダーには、ビジネスを俯瞰する人、現場で業務を指揮する人、個々の業務に従事する人、そしてIT技術者など、様々な立場の人がいる

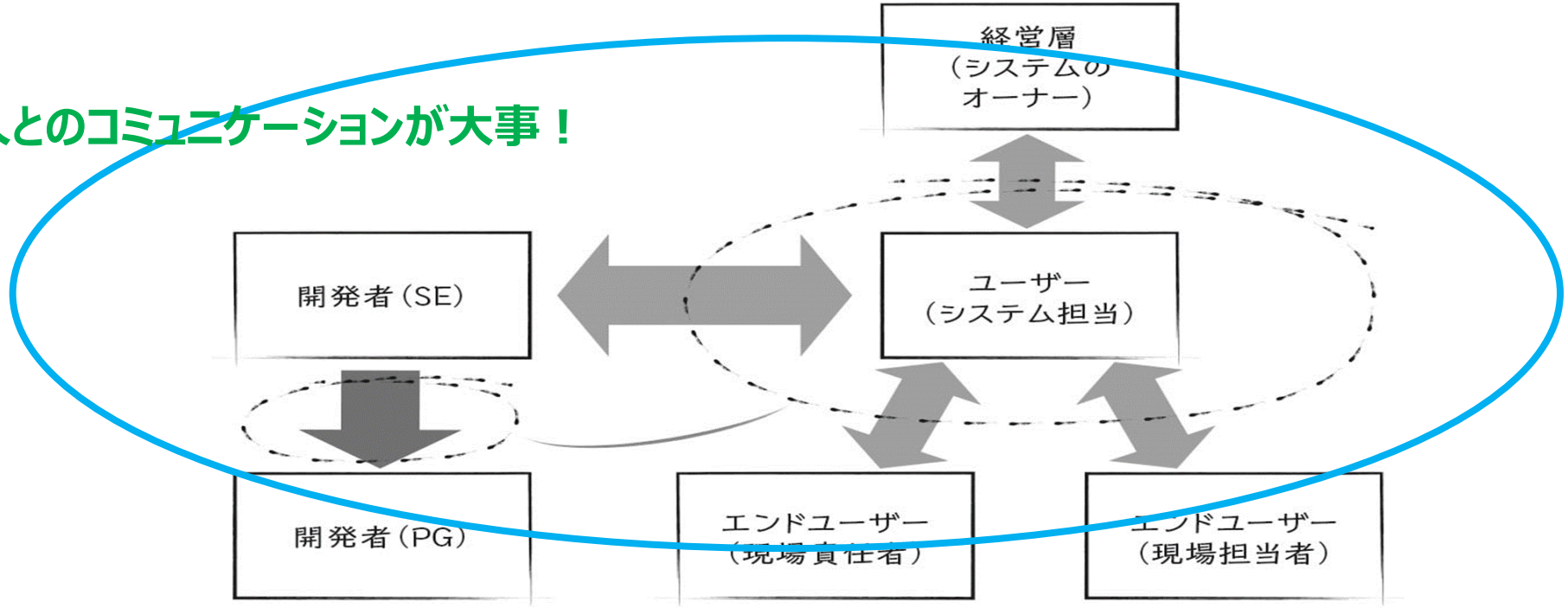
✓ 「コミュニケーションが大事！」

コミュニケーションが大事

- とにかく「コミュニケーションが大事！」

✓ まず「コミュニケーションの価値は受け手が決める！」という当たり前のことを、肝に銘じる必要がある

✓ 送り手の独りよがりでは、コミュニケーションは成立しない



「要件定義のセオリー」より

コミュニケーションが苦手な人へ

- 人と話すのは楽しいよ！

✓ 直接話すのが苦手なら違うコミュニケーション方法を考える
✓ 今ならいろいろある。 びくびくしない。自分の軸を持つ
✓ その上で立場、意見の違う人と目的、方向性を共有する

- 特に上流工程であればあるほどビジネス（業務）屋と
  システム（IT）屋の「共同作業」は必須になるので受け身では駄目

✓ 「早く決めてください」では決まらない
✓ 「決めてくれない」はNG・・何故決められないのか考える
✓ システムは基本形がないものであり、責任を負ってまでOKを出す勇気が出ないかも・・・いろんな人の顔がちらつく？

ステークホルダーとの関係性

- ステークホルダーとは、共通認識を持ち続ける必要がある

✓ とはいえ組織人はそれぞれ自分の立場がある

✓ 問題が発生した場合には、やはり疑似的でもよいので
トップダウンが有効。 難しければミドルアップを経由

✓ もろもろを考慮すると「目指すべき姿」「実現すべき姿」の
確認はなんらかの図式と補足説明として自然言語で書いた
ものの組み合わせをコミュニケーションツールとしてうまく使うの
が有効ではないか

落としどころを模索する！

- 経営的価値から提案する

✓ 実現が難しい要求に対してもビジネス（経営）要求を満たす代替え案として要件を作り上げる

✓ 最初から「できません」ではダメ・・・

✓ わかりやすい図式を駆使して合意に持っていく

(例えば）要件定義で必要なこととは

- ビジネス、業務寄りの表現でITシステムで実現すべき姿を明らかにする
- 立場の異なる人に対して説明可能な成果物を作成
- 次工程である「設計工程」のインプットとして有用な成果物の作成
- 次工程である設計が可能な状態にする
- 実現するための見積を可能にする

✓ 技術者ではない人にも理解ができ、合意に至ることが可能な表現を用いる必要がある
✓ 後工程の技術者がわかる、活用できる表現でなければならない

結局のところ・・・

- 要件定義はいらなくならない！

✓ 技術が進化し、開発のやり方が変わろうと要件定義の重要性が変わるわけではない

✓ 実現すべき姿を明らかにすることの意味は不変！

✓ 要件定義を成功に導くためには要求を明確にしておくことがポイントになる！

✓ 要求分析の精度が高めれば要件定義はうまくいく

✓ 必然的にシステム開発全体を成功に導くことになる

そのためには言葉だけではなく！

- 図式を作成し、活用、共有する！

✓ 言葉（言語）だけでは難しい

✓ 誰が見ても同じ理解を得るように、抽象的なイメージをまずポンチ絵等でわかりやすく表現する

✓ 要求はあくまでユーザーから出てくるものだから、それを分析、整理する際には、ビジネス表現、つまりユーザーが理解できる表現である必要がある

✓ さらに後工程でも有効な、強靭かつシンプルな「図式」の作成を目指す

✓ 後述する「４点の図」は上記を可能とする

ここでもコミュニケーションが大事？

- 超上流含めた広義のシステム設計は何故行う必要があるのか。
  - ✓ コミュニケーションのため、につきる
  - ✓ 自分が必要として自分で作り、作り続けて、自分で使い、使い続けるのであれば設計なんていらない
  - ✓ しかし、そんなシステムはほぼないので必要になるということ
  - ✓ 早い段階からビジネス（業務）屋とIT（システム）屋の共通認識の形成を意識
  - ✓ 開発工程全般に渡って、だけではなくシステムライフサイクル全般のコミュニケーションを意識
  - ✓ さらに全社アーキテクチャ（EA）を意識して行うと尚良い

✓開発工程全般、開発保守含めたシステムライフサイクル全般、他システムとの連携含めたEA全般において立場の違う者同士のコミュニケーションを可能とする必要がある

Agenda

1. データ中心でいこう！
2. ビジネスとITの一体化を目指そう！
3. 全てはコミュニケーション！
4. DxBA（通称DOBA）の実践
5. 超上流から「設計」を始めよう！

データ中心型ビジネスアプローチ（DxBA　通称DOBA）とは！

- **「データ経営が日本を変える！」**

✓ JUASシステム高度化プロジェクトにて提唱された考え方

✓ 従来の設計手法としてのDOAよりさらにビジネス視点からデータ中心を指向する考え方

JUAS　Webサイトより全文Pdfにてダウンロードいただけます。
https://juas.or.jp/library/research_rpt/various/#data

Amazon　Kindle版はこちら。
https://www.amazon.co.jp/dp/B0BB2D9B5X

「全体を把握する設計図の3条件」への挑戦

- 日経BP谷島氏が提唱した「全体を把握する設計図の3条件」への挑戦

「40年取材をして分かった、日本企業に「設計図」はない」
https://xtech.nikkei.com/atcl/nxt/column/18/00166/101700136/

✓ 第1は誰でも読めること
✓ 第2は自分で描けること
✓ 第3は一定の書式や記載のルールがあること

✓ データ中心型ビジネスアプローチ（DxBA　通称DOBA）の実践方法である4点の図（後述）は上記の3条件を満たすことを前提として描く！

ビジネス体系及び要求を表す4点の図

✓ **「ビジネス鳥瞰図」**

ポンチ絵でもいいので**全体像をつかむ**ために描く。
ビジョン、目的、方向性がわかるようにするとともに、この時点で明確な制約、前提があれば書き出しておく。

✓ **「ビジネス地図」**

最大の価値を生み出すビジネス、業務の**あり方**を描く。

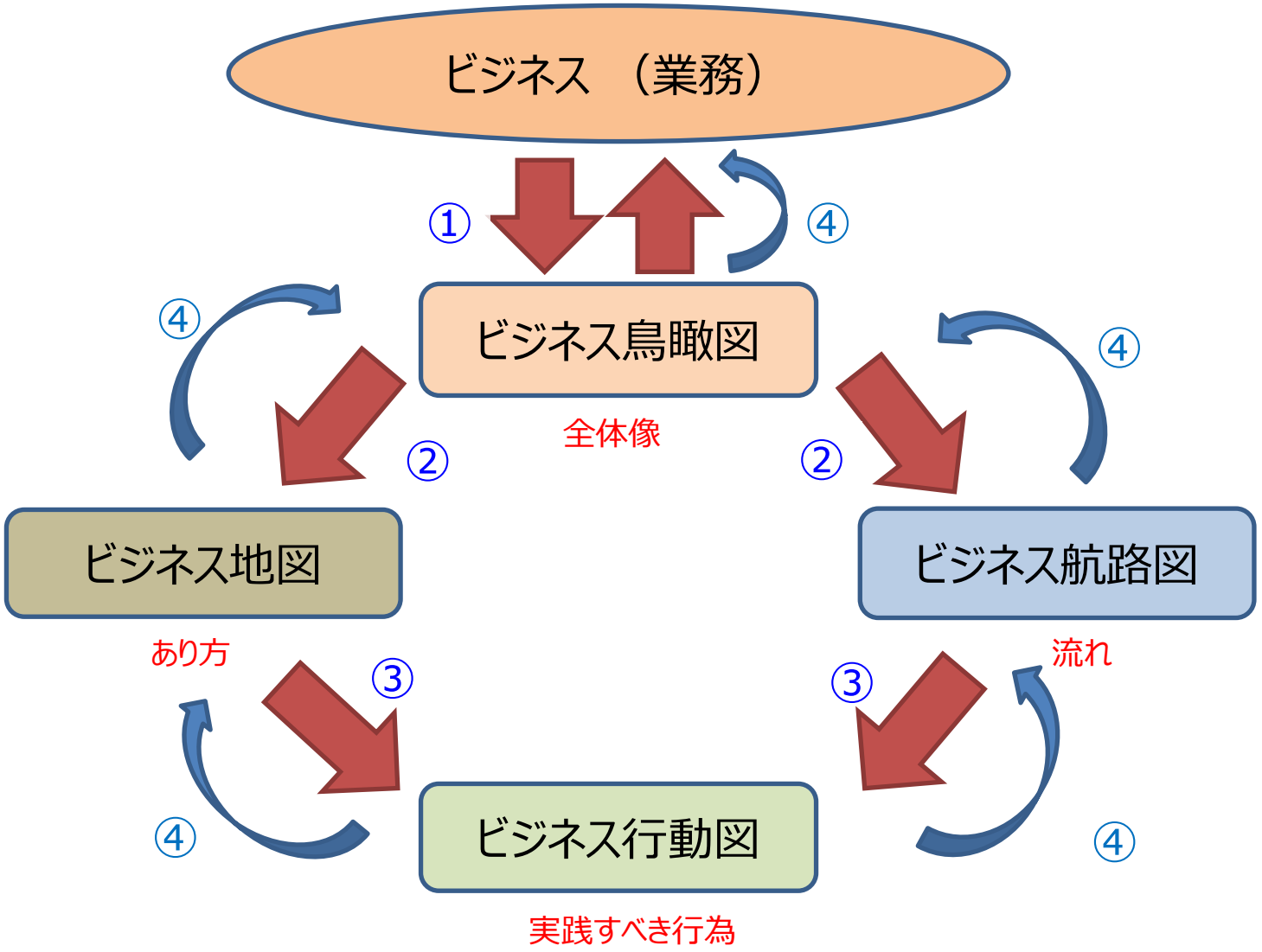
✓ **「ビジネス航路図」**

ビジネス地図を最適に巡ることにより効果を出す道筋を示すためにビジネス、業務の**流れ**を描く。

✓ **「ビジネス行動図」**

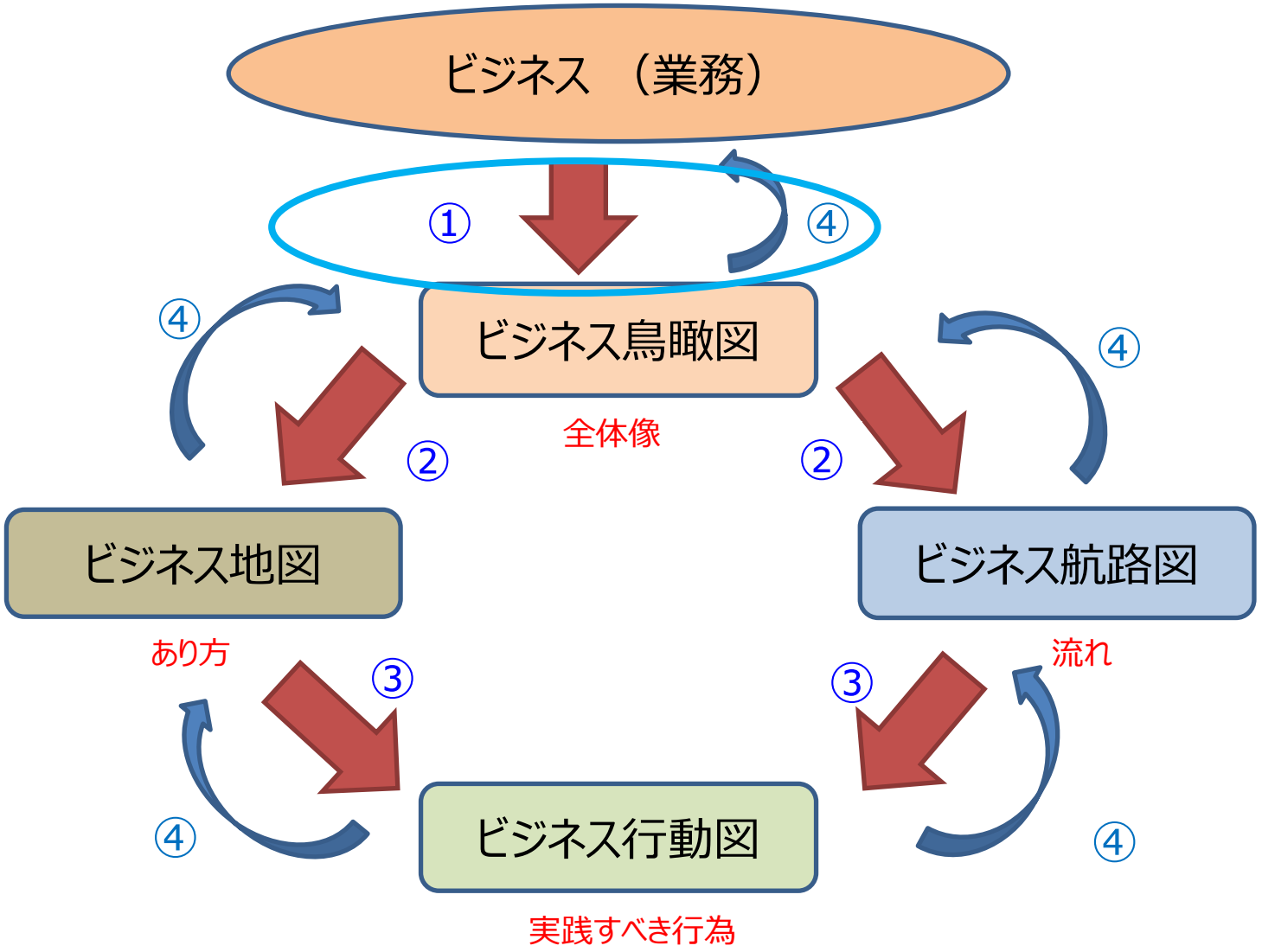
ビジネス航路図で辿り着いたビジネス地図のある地点で**実践すべき行為**を表すために描く。

４点の図の作成手順（P20　パターン１＝ビジネス創出と並行）



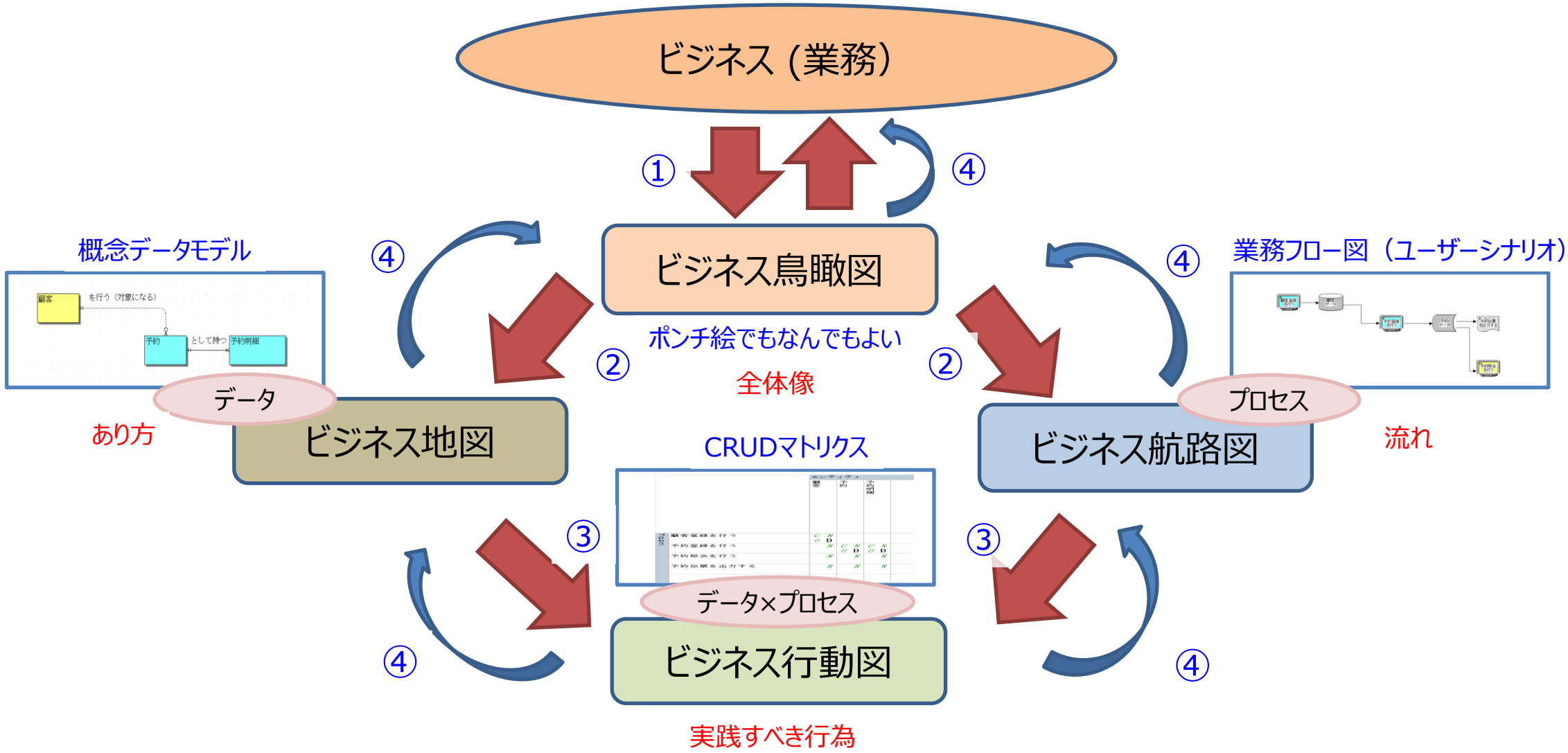
①ビジネス創出とともに**ビジネス鳥瞰図**を作成する。
　作成時、ビジネスもより有効な形に修正
　を加える。
②ビジネス鳥瞰図を基に、
　ビジネス、業務のあり方：データに関すること
　ビジネス、業務の流れ：プロセスに関すること
　に分類し、
　ビジネス地図、**ビジネス航路図**を作成する。
③ビジネス地図、ビジネス航路図を基に
　実践すべき行為を明らかにして**ビジネス行動図**
　を作成する。
④4点の図を併行してブラッシュアップを繰り返して
　仕上げる。
　ビジネス鳥瞰図の内容よりビジネスについても
　調整を行う。

４点の図の作成手順（P21　パターン2＝ビジネスありき）



①ビジネスを基に~~創出とともに~~**ビジネス鳥瞰図**を作成する。
~~作成時、ビジネスもより有効な形に修正を加える。~~
②ビジネス鳥瞰図を基に、
ビジネス、業務のあり方：データに関すること
ビジネス、業務の流れ：プロセスに関すること
に分類し、
ビジネス地図、**ビジネス航路図**を作成する。
③ビジネス地図、ビジネス航路図を基に
実践すべき行為を明らかにして**ビジネス行動図**を作成する。
④4点の図を併行してブラッシュアップを繰り返して仕上げる。
ビジネス鳥瞰図の内容よりビジネスについても調整を行う。

作成手順の概要(図例付き)

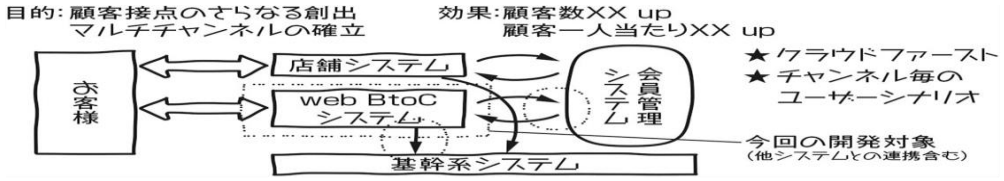


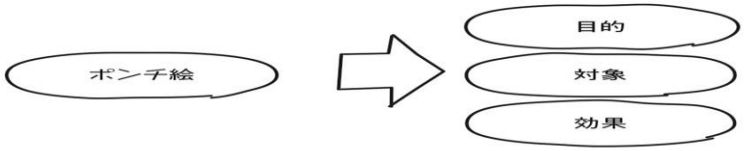
＊データ中心型ビジネスアプローチはあくまで考え方でありどんな記法で描いてもよい

ビジネス鳥瞰図

- **ビジネス鳥瞰図としての「ポンチ絵」 全体像をつかむ**

  - ✓ 目指すべきビジネスの全体像をイメージ化することに努める
  - ✓ ジャストアイデアをどんどん取り込んでいく
  - ✓ ○理想の姿　×空想の姿
  - ✓ 無理してかっこよく1つの絵にまとめようとしなくていい　要望レベルで描き始める。 他の3図描いていくうちに整合性が担保されて要求を表現した図になっていく
  - ✓ 「ビジネスモデルキャンパス」等のフレームワークを使用したり、デザイン思考を積極的に用いることも検討する
  - ✓ ビジョン、ビジネス価値(収益構造まで明確になると尚良い）を明確にすることを意識する
  - ✓ わかりやすく表現することを意識することにより「あり方」「流れ」を抽出可能とする

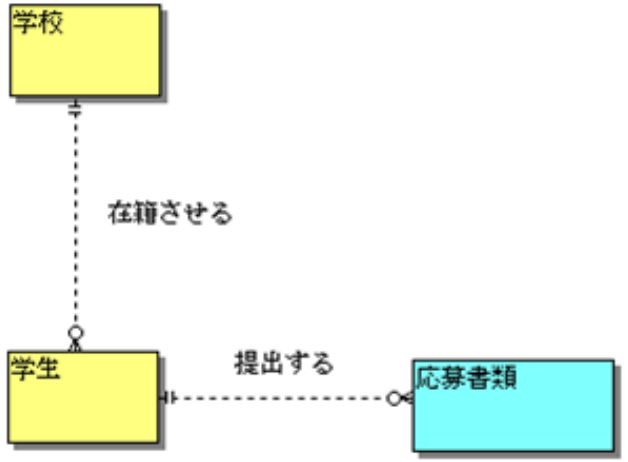




ビジネス地図

- **ビジネス地図としての「概念データモデル」　あり方**

✓ 実体と呼ばれるエンティティという「箱」（下記の図であれば「学校」「学生」「応募書類」）と、実体同士の関係を表すリレーションシップという鳥の足付きの「線」（下記の図であれば「学校」と「学生」の間にひかれた線、「学生」と「応募書類」の間にひかれた線）であり方を表現する



エンティティの名称、リレーションシップの定義はきちんと日本語で表記する
（線の元エンティティ：主語）は（線の先エンティティ：目的語）を（矢印線：述語）する、という表記になる

左図例：・「学校」は「学生」を「在籍させる」
・「学生」は「応募書類」を「提出する」

この繰り返しでモデルができ上がる

・わかりやすさ優先　過度な抽象化はおこなわない
・リソースのあり方、外部連携（主にデータ連携）は押さえる必要あり

ビジネス航路図

- **ビジネス航路図としての「業務フロー図」 流れ**

✓ 横書きで描く場合、左から右へ、縦書きで描く場合、上から下に
ビジネス、業務の流れを表す

✓ プロセスをわかりやすくアイコンで表現し、アイコン同士を線で繋いでいく

✓ 個々のプロセスは「～する」「～を行う」という動詞句で表現

✓ プロセスが操作するデータを記述し、プロセスオーナーにデータへの責任を認識させる

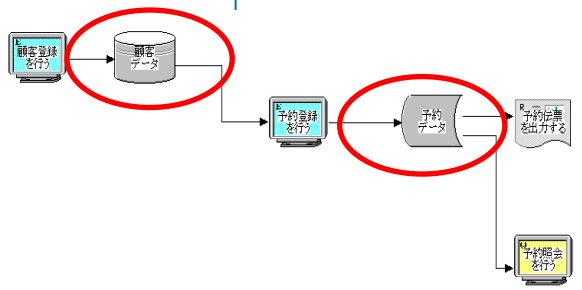
✓ 個々のプロセスに対しては5W2Hにて定義を記述

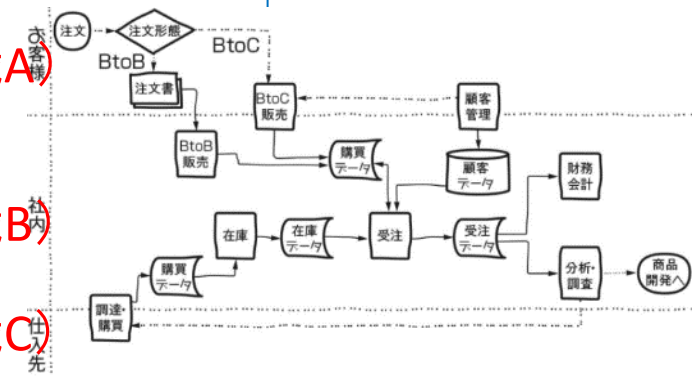
「When（いつ）」「Where（どこで）」「Who（だれが）」「What（なにを）」「Why（なぜ）」
「How（どのように）」「How Many（どれ位）」

✓ ビジネス、業務規模が大きい場合は階層化する

✓ 役割(組織）を明確にしてスイムレーンで分ける

✓ リソース管理、外部連携（主にデータ連携）は押さえておく





役割（組織A）

役割（組織B）

役割（組織C）

ビジネス行動図

- **ビジネス行動図としての「CRUDマトリクス」 実践すべき行為**

✓ 概念データモデルに記述されたエンティティと業務フロー図の個々のプロセスを対象に実践すべき行為を表す

✓ 業務フロー図に記述された「XXXデータ」のアイコンとプロセスを参考に記述

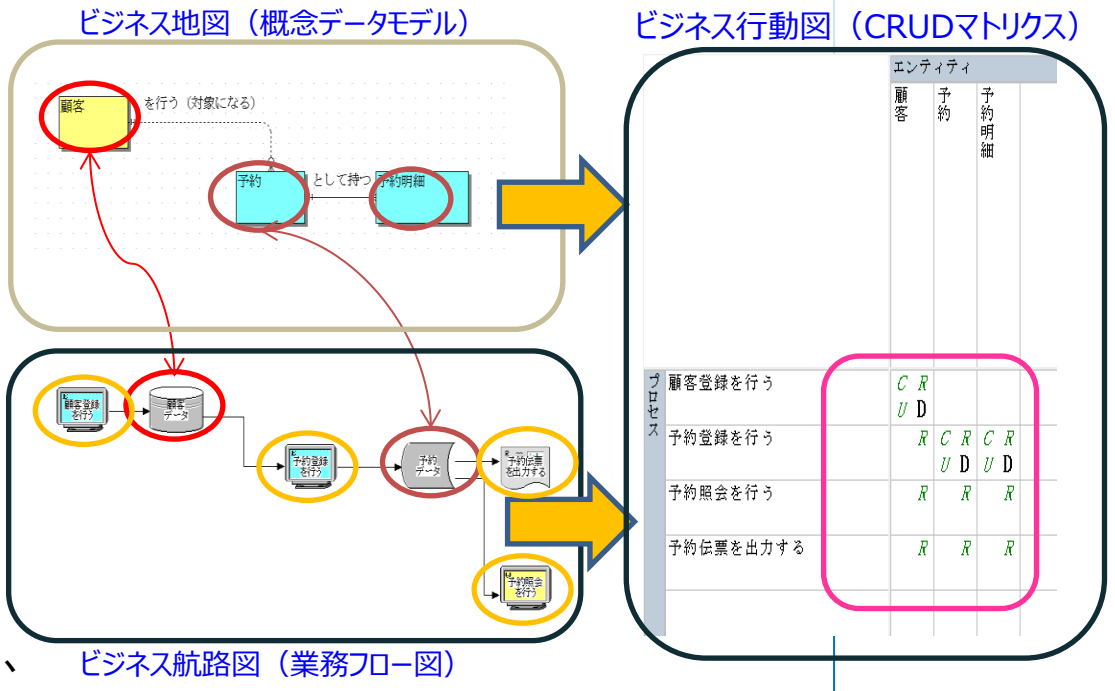
✓ データとプロセスが交差する欄に該当するデータ操作の文字（C,R,U,D）を記述

なんらかの操作、行為が明らかになっていれば内容を記述しておく

✓ データ(エンティティ）のデータライフサイクルがもれがなくなる状態まで記述・・・全てのデータに対してCRUDが定義されるまで

✓ もれている場合、データ、プロセスの追加を検討し、各図に反映後、行動図（CRUD)に反映

✓ 外部連携（主にデータ連携）API使用が明らかな場合はAPIをデータとして定義して、CRUDに取り込んでおく



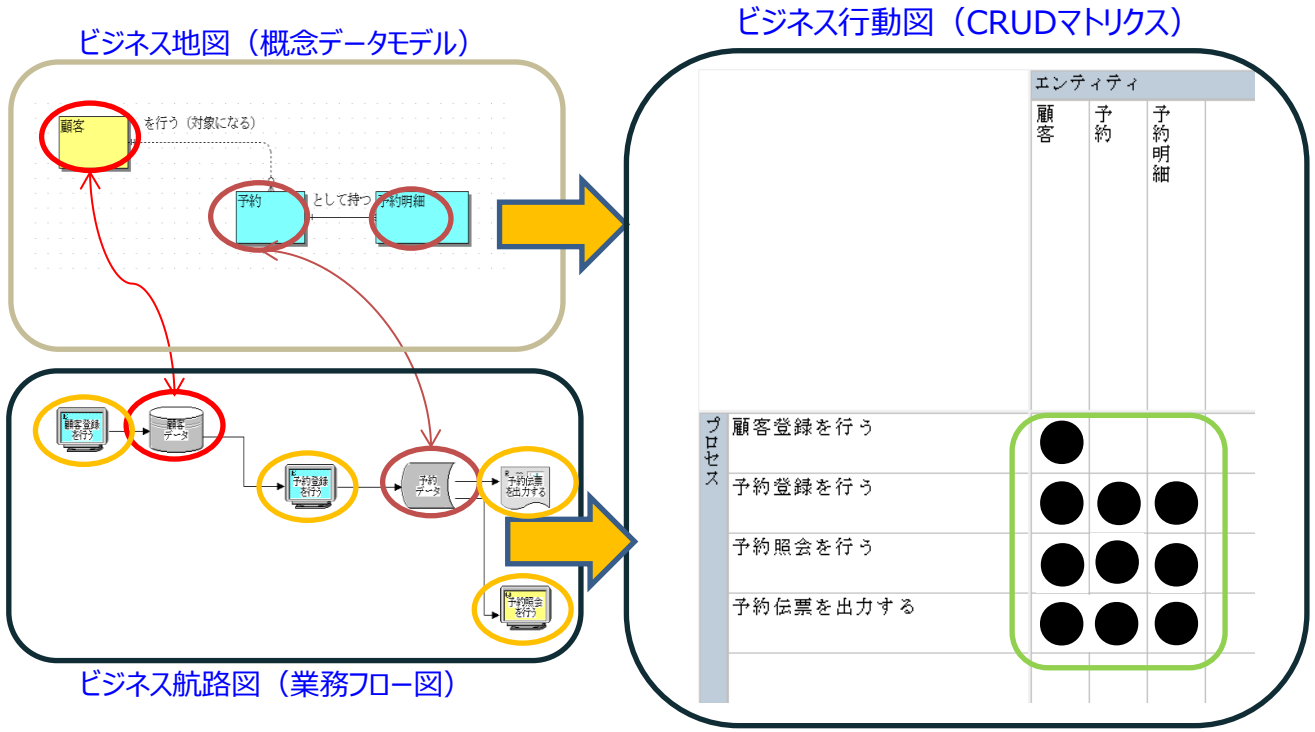
ビジネス行動図その2

- **ビジネス行動図としての「CRUDマトリクス」簡易版**

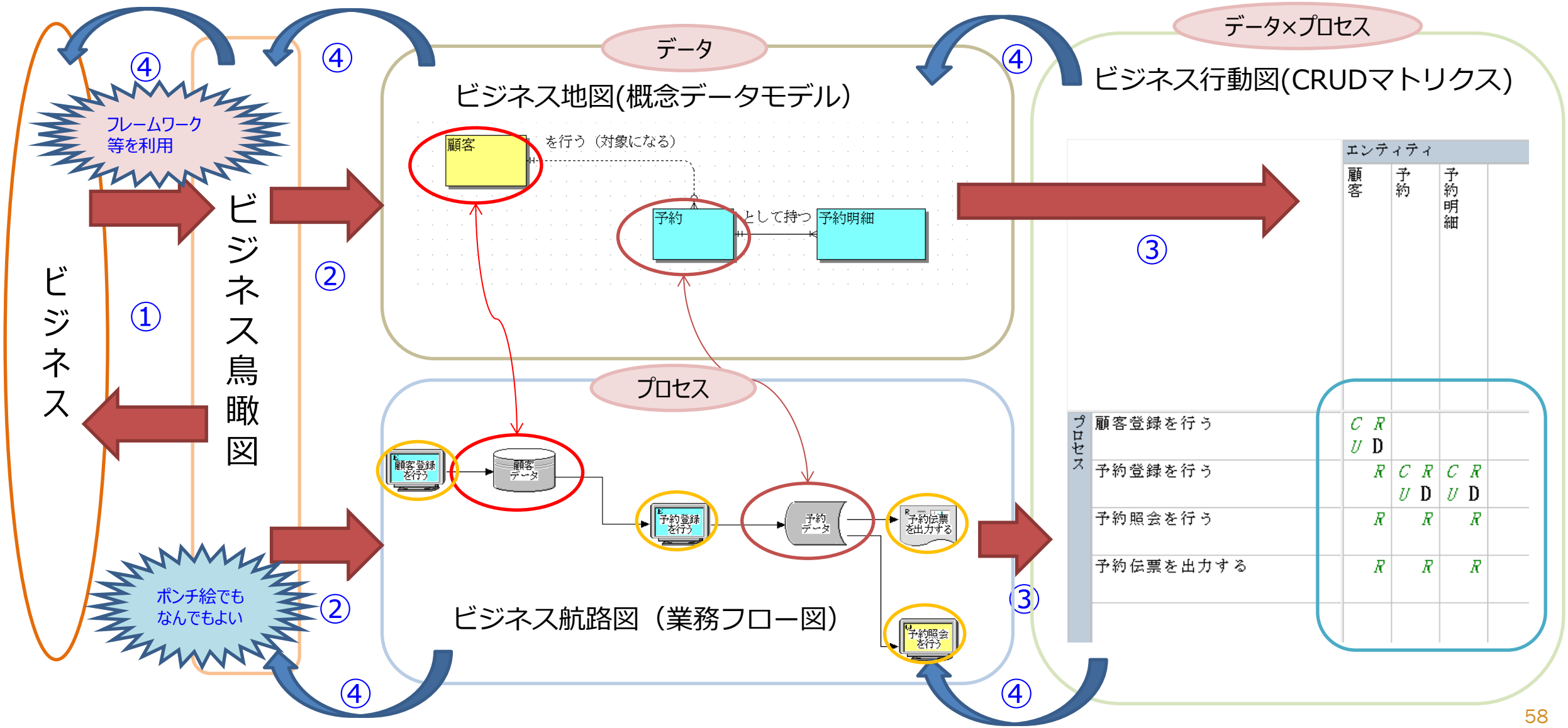
✓ この段階ではデータとプロセスに対する実践すべき行為さえ把握できれば“正式な”CRUDマトリクスでなくてもよい

✓ 下図では●：なんらかのデータ操作がありうると思われる場合マークで表現

✓ この段階ではデータとプロセスの関係の有無がわかればよい



４点の図の具体的な関係性



４点の図と「システム設計」にて作成したモデルとの関係性

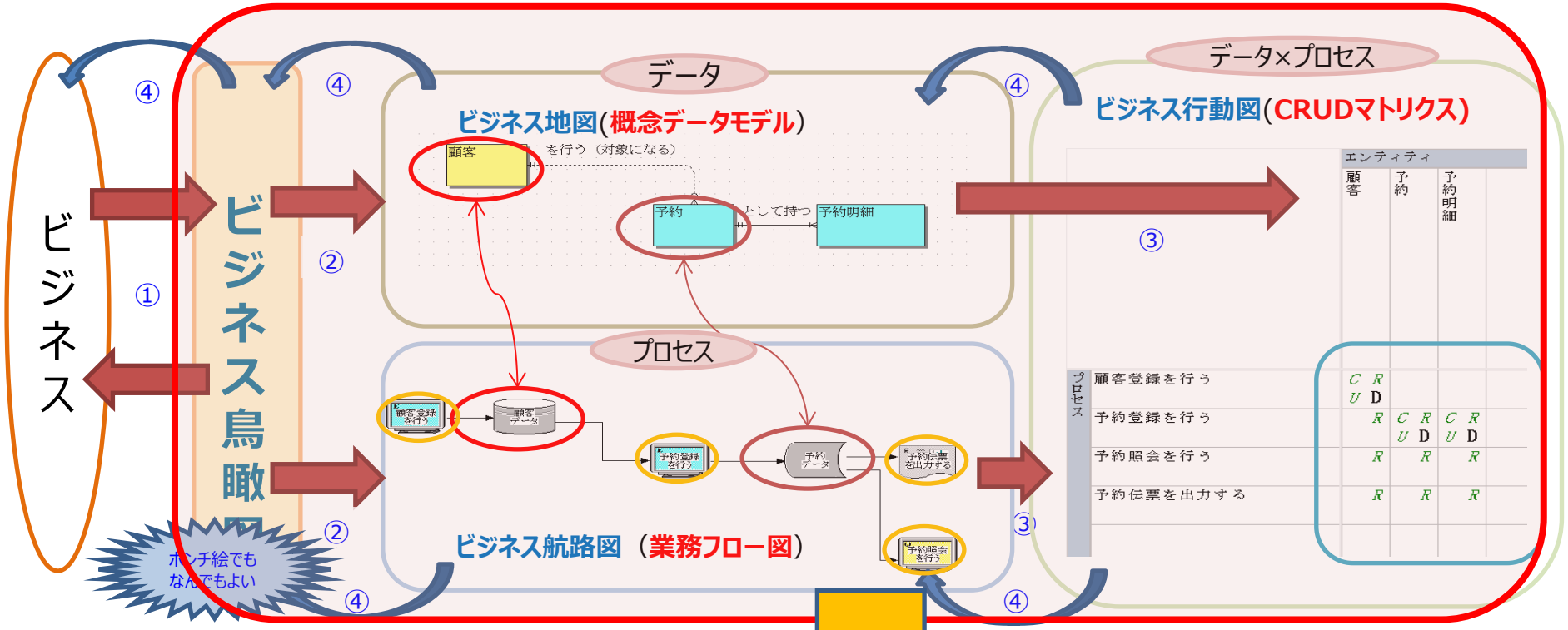
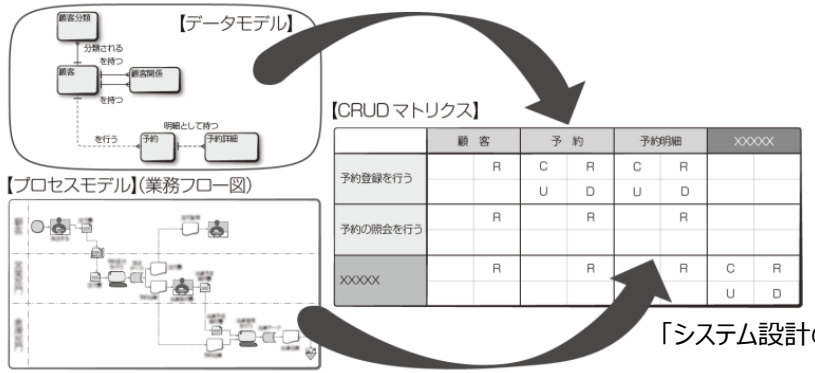


図5-1　CRUDマトリクスのイメージ



4点の図は後工程であるシステム設計で
そのまま有効な“モデル”として継承できる！

「システム設計のセオリーⅡ　クラウドベース開発」より

４点の図を作成するために

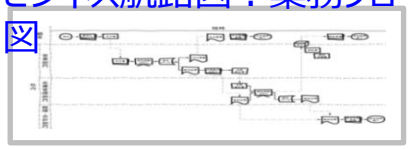
- **「自分達、皆で」脳みそを絞って考え抜き作成する**
  - ✓ アナログの力を最大限に活用して手で書いては消して、書いては消して、を繰り返して、仕上げていく
  - ✓ この試行錯誤の繰り返しが重要　最初はおかしくてもいい。どんどん直していく。その課程が大事
  - ✓ できればホワイトボード等を数枚用意して立場の違う関係者同士で囲んでワイガヤ
  - ✓ 右脳と左脳をフル回転させて描いていく
  - ✓ ビジネス鳥観図、ビジネス地図、ビジネス航路図、ビジネス行動図を並行して書いていき、ビジネスと4点の図の整合性を保ちつつ一体化していく
  - ✓ 誰でも描ける、読めることを目指し、なるべくシンプルでわかりやすく描く

ビジネス地図：概念データモデル

ビジネス鳥瞰図：ポンチ絵

ビジネス行動図：CRUDマトリクス

ビジネス航路図：業務フロー図



図作成時のお作法

- **テクノロジーの進化・ビジネス（業務）の変化は様々な立場の人間に影響を与える**
  そのため、今まで以上に影響を受ける立場の違う者同士がわかりあえることを最優先に考える
  - ✓ 誰でも描ける、読めることを目指し、なるべくシンプルでわかりやすいものを選ぶ
  - ✓ 標準化よりわかりやすさを優先して「手が届く共通言語(図式）」を目指す
  - ✓ 自分達の企業組織にあった（身の丈にあった）最適な記法を選択する
  - ✓ 鳥瞰図は設計工程にて全体図への継承を可能とすることを意識する
  - ✓ 地図、航路図、行動図の3図は設計工程への継承を可能とするために一定の記法ルールがあるものを選択する
  - ✓ 一つの企業組織においては全社でなるべく一つの記法に統一する
  - ✓ 作成した図が自分達の企業組織のビジネス、業務の可視化を実現し、かつ、それを元に関係者が**ワイガヤ**のコミュニケーションを可能にするもの（方法論、記法）を選択する

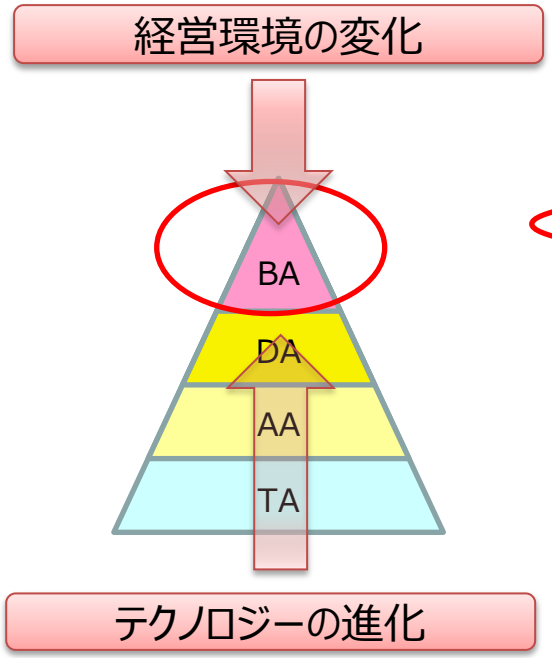
Agenda

1. データ中心でいこう！
2. ビジネスとITの一体化を目指そう！
3. 全てはコミュニケーション！
4. DxBA（通称DOBA）の実践
5. 超上流から「設計」を始めよう！

変化と進化に対応し続けるエンタープライズアーキテクチャ

- **「経営環境の変化」「テクノロジーの進化」によりビジネス／業務は影響を受ける！**

✓ クラウド、ERP、SaaSサービス、ノーコード／ローコード開発ツール、AI、IoT、NoSQL等のTA（技術体系）、AA（適用処理体系）、DA（データ体系）の進化によりBA(ビジネス（業務）体系)は激変していく！



BA: ビジネス(業務)体系（Business Architecture）
DA: データ体系（Data Architecture）
AA: 適用処理体系（ Application Architecture）
TA: 技術体系（ Technology Architecture）

変革に必要な抽象とは！

- JIIMA機関誌「IM」2023年7･8月号
  奥平氏取材／執筆
  「データ中心型ビジネスアプローチ（DOBA）で拓く「日本型DX」の姿
  ＝ITエンジニア・赤 俊哉 氏の眼から見える「DXの現在地」とは？！＝」
  https://www.jiima.or.jp/wp-content/uploads/im-pdf/2023_7_8.pdf

✓ 『「制約」を前提に物事を考えていても、「変革」にはつながらないからだ。特に「カイゼン」が盛んな日本企業では「具体論」を基調に議論を進める傾向が強く、「変革」に必要な「抽象論」が疎かになってしまっている。現状のDXが、「改善」の域を脱し切れていないのもそのためだ。逆に「カイゼン」に長けた日本企業が「ゴール」へと目線を高めていけば、そのアドバンテージを活かした「日本型DX」の姿が見えてくる』

✓ 日本企業は事例好きだけど、抽象化は嫌い？
✓ 日本人は事例を基に抽象化して、具体化してくことが苦手？
✓ そもそもITに向いていない？

ではどうする？

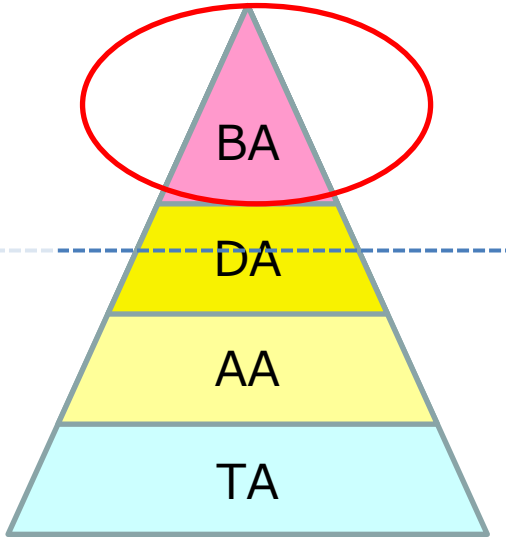
- ビジネス創出との接点である超上流工程から
  要求分析とともに広義の設計を始めること！

✓ 典型的なアンチパターン
「上流工程の不備をプロジェクトマネジメントでなんとかする」
✓ 残念ながらなんとかならない⇒上流に遡るしかない！

（ここで再確認）そもそもシステム設計とは

- **システム設計　＝ビジネス／業務設計＋IT設計**
  - ✓ システム設計はITの設計だけで成り立っている訳ではない
  - ✓ ビジネス／業務の設計とIT設計を両方行うのがシステム設計
  - ✓ ビジネス／業務設計をきちんと行わないシステム開発は間違いなく破綻する
  - ✓ DXだろうがなんであろうがビジネス／業務設計は必須

★システム設計
＝ビジネス（業務）設計＋IT設計
⇒設計を可能とする要求・要件の明確化が必要



ビジネス（業務）設計の範囲・・・データ・プロセス・その関係性の設計
⇒可能とする要求・要件の明確化が必要

IT設計の範囲・・・実装に関わる設計
⇒可能とする要求・要件の明確化が必要

「システム設計のセオリーⅡ　クラウドベース開発」より

まず、ビジネス／業務設計を可能とするために

- **開発対象のシステムにおけるビジネス／業務の全体像、**

  データ、プロセス、その関係性を明らかにする！

- ✓ 全体像：ビジネス/業務システム全体を鳥瞰するものが必要

  ⇒ ビジネス鳥瞰図からシステム全体図へ

- ✓ データ ： ビジネス/業務のデータ構造が把握可能なものが必要

  ⇒ ビジネス地図からデータモデル（クラス図）へ

- ✓ プロセス ： ビジネス業務のフロー、シナリオが把握可能なものが必要

  ⇒ ビジネス航路図から業務フロー（BPMN、アクティビティ図）、ユースケースへ

- ✓ 関係性：データとプロセスの関係性が把握可能なものが必要

  ⇒ビジネス行動図から***CRUDマトリクス***、クラス図へ

関係性：何故「CRUDマトリクス」なのか

- **「システム設計のセオリー」「要件定義のセオリー」**

  そして新書「システム設計のセオリーⅡクラウドベース開発」に

  おいて「CRUDマトリクス」を大きく取り上げている、何故か！

  - ✓「今どき流行らない」
  - ✓「古臭い」
  - ✓「他にもいくらでもいいやり方がありそうなもの」
  - ✓「もっと今風？のやり方を紹介してほしい」
  - ✓「知っている人すら少ないのでは」

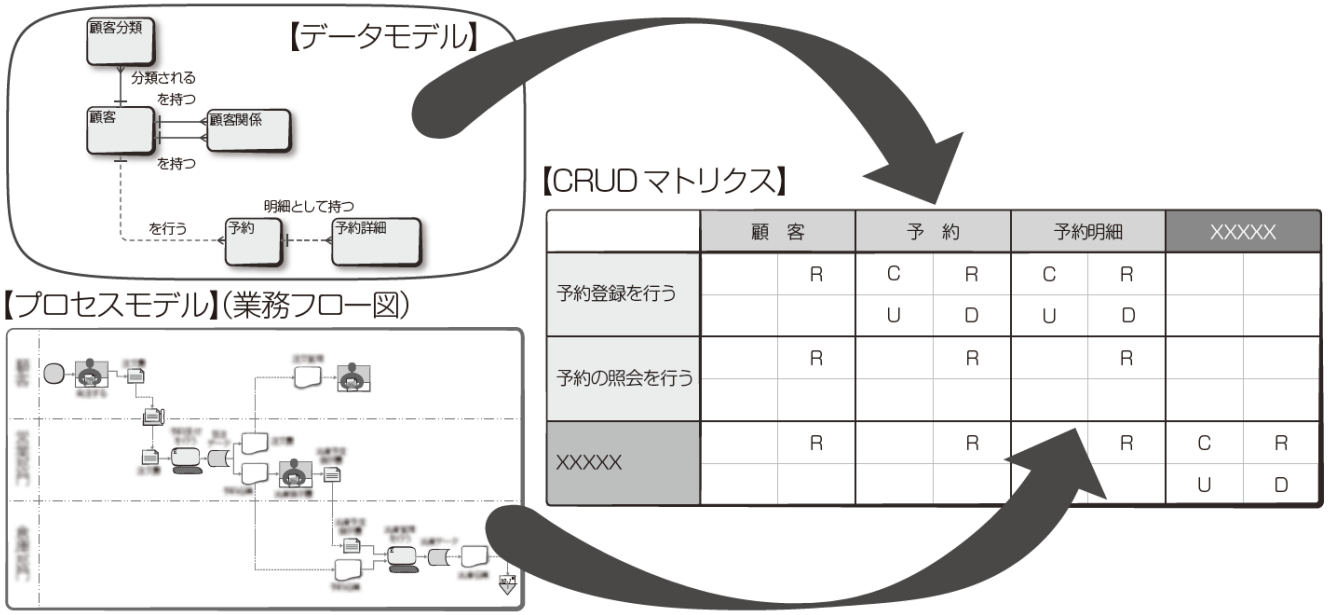
CRUDマトリクスとは：ビジネス行動図を設計に継承したもの

- データとプロセスの関係性の見える化を可能とする
- ビジネス地図とビジネス航路図の関係性を表すビジネス行動図を設計情報として継承したものでもある

＊データ
（ビジネス地図から継承）

＊プロセス
（ビジネス航路図から継承）

図5-1　CRUDマトリクスのイメージ

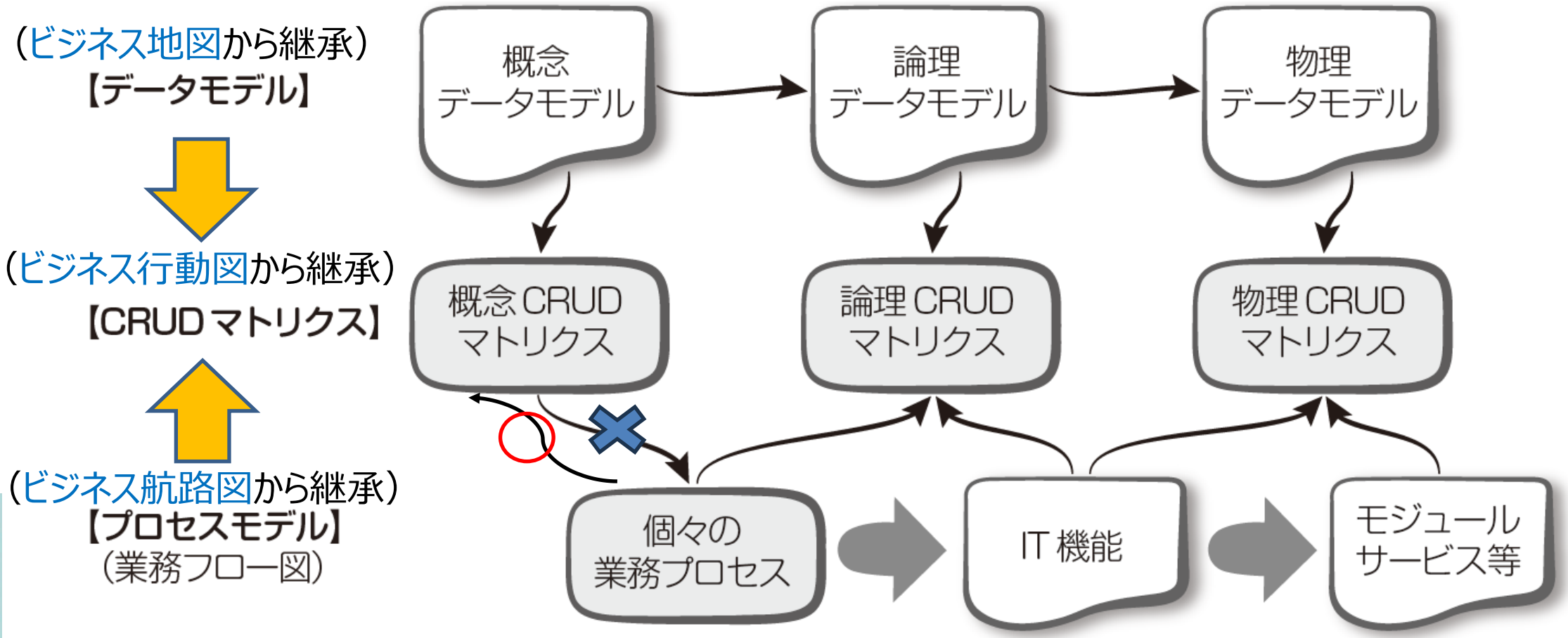


	顧客		予約		予約明細		XXXXX	
予約登録を行う		R	C	R	C	R		
			U	D	U	D		
予約の照会を行う		R		R		R		
XXXXX		R		R		R	C	R
							U	D

「システム設計のセオリーⅡ　クラウドベース開発」より

CRUD設計：データモデル･プロセスモデル･CRUDマトリクスの関係

図 5-2　概念／論理／物理 CRUD マトリクスの関係



「システム設計のセオリー Ⅱ　クラウドベース開発」より

論理CRUDマトリクスの例

＊データ×プロセスのCRUDマトリクス

図5-6　図5.2-1 論理CRUDマトリクスの一例

論理エンティティ／プロセス	顧客		住所		サービス		サービス明細	
顧客情報を登録する	C	R		R		R		R
	U	D						

＊データ×機能（プロセスの構成要素）のCRUDマトリクス

論理エンティティ／機能	顧客		住所		サービス		サービス明細	
顧客登録	C							
	U	D						
顧客検索		R						
住所検索				R				
サービス検索						R		R

「システム設計のセオリーⅡ　クラウドベース開発」より

さらにシステム設計とは

- コミュケーションであり設計書はコミュケーションツール
- ✓ システム開発におけるドキュメントは全てコミュニケーションに役立たないと意味をなさない

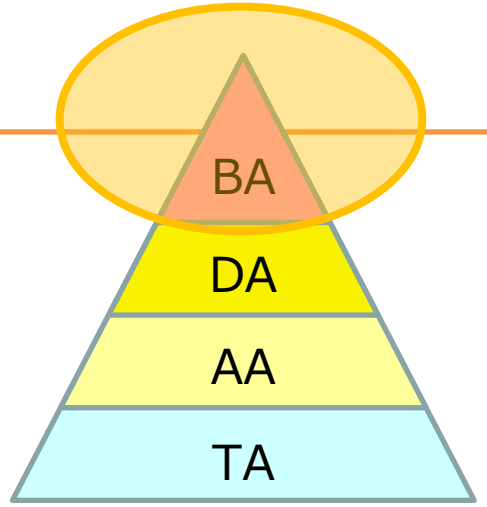
- ✓ 開発工程全体のコミュニケーション
- ✓ システムライフサイクル全体のコミュニケーション
- ✓ 他システム含めた全社システム全体のコミュニケーション

超上流工程から始めるシステム設計

- **データ中心型ビジネスアプローチの実践**

DxBA（ｘ･･･centric,driven,orientedなんでもいい）･･･通称ＤＯＢＡ

- ✓ データ中心型ビジネスアプローチは超上流工程から実装まで一気通貫を可能とする
- ✓ 実現にはビジネス／事業創出と併行してビジネス体系（ＢＡ）として**4点の図を作成する**ことにより「設計を始める」ことになる
- ✓ システム設計そして実装まで繋ぐことを可能とする”超上流工程ゼロ“からの考え方でもある



超上流工程からシステム設計へ一気通貫！

- **ビジネス創出とともに「広義の」システム設計を始めて行く**
  - ✓ ビジネス／業務の創出とともに必要なデータ、プロセス、その関係性を「4点の図」
    を描くことでビジネスとITの目指すべき姿を明らかにした上で実装へ一気通貫で繋げていく！

◆超上流

データ中心型ビジネスアプローチ（DxBA）

【4点の図】
①ビジネス鳥瞰図(全体像)
②ビジネス地図(あり方)
③ビジネス航路図（流れ）
④ビジネス行動図(実践すべき行為)

◆設計

データ:データモデル

×

プロセス：プロセスモデル
(業務フロー)

データとプロセス
の関係（CRUDﾏﾄﾘｸｽ）

終わりに

- 「鳥の目を持って地べたを這う」
  - ✓ 超上流工程からシステム設計へシームレスに繋ぐ！
  - ✓ ビジネス創出/要求分析からデータ中心で！
  - ✓ 「正しい」システム設計は、全て上記精神？
    を持って臨めば必ず成功する！
  - ✓ （と私は確信しています！）