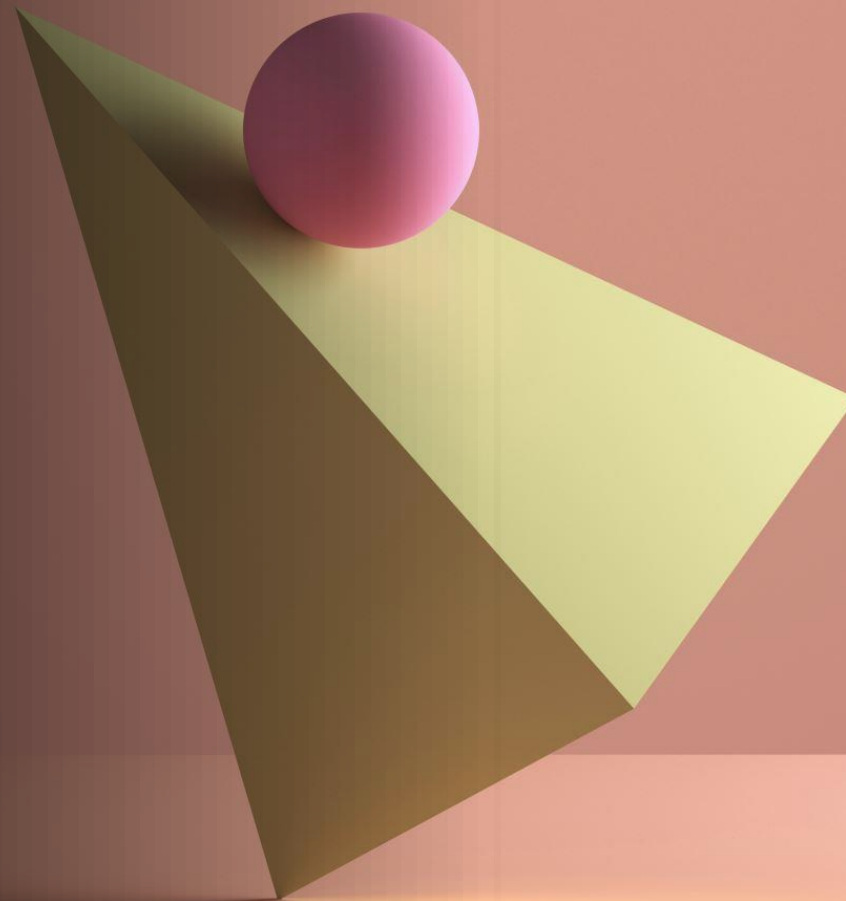




既存システム の可視化 支援サービス

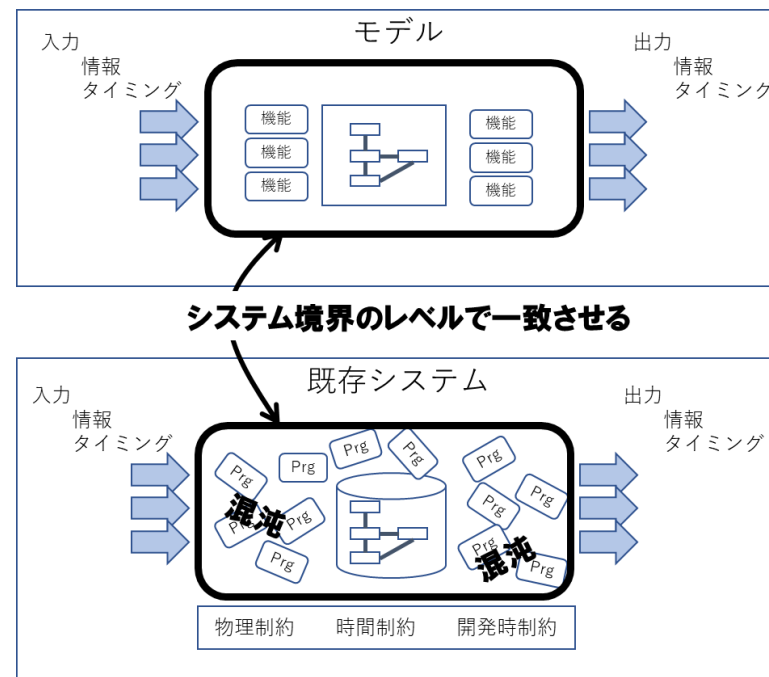


既存システムの可視化

- 可視化の方針

- 可視化の目的は新規開発に向けた現状調査としての可視化と、保守開発の精度向上を目的とした可視化に分かれる

可視化の目的		
	ToBe開発に向けた可視化	保守開発に向けた可視化
方針	ToBe検討に必要最低限の整理を行う	保守品質を上げるために行う網羅的に各々の関係者の詳細な資料と紐づけられる内容で整理する
説明	ToBeの要求に応じてAsIsの扱いが異なるので、後で調べられるようにインデックス的に整理する	システム化関係者が共有できる要件レベルで全体俯瞰できるように整理する



既存システムを可視化するための視点

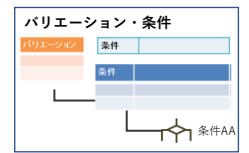
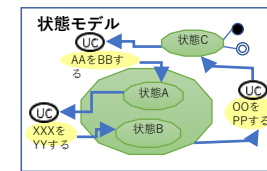
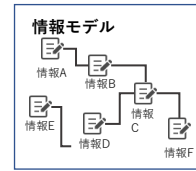
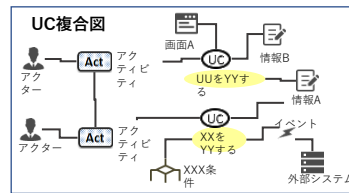
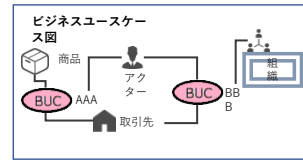
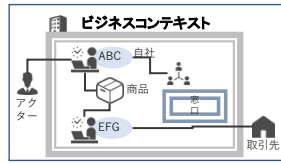
- 既存システムの可視化は、その目的によって2つに分かれる
 - ToBe開発に向けた可視化
 - 保守開発に向けた可視化
- システムの可視化は大きく3つに分かれる
 - 業務を理解する
 - システムの機能を理解する
 - ビジネスルールを整理する
- プロジェクトでヒアリング可能な人によって進め方が変わる
 - 対象システムを開発した人がメンバーに居る
 - 対象システムの使い方を理解した人がいる
 - 対象システムの業務を理解した人がいる

可視化の目的		
	ToBe開発に向けた可視化	保守開発に向けた可視化
方針	ToBe検討に必要最低限の整理を行う	保守品質を上げるために行う 網羅的に各々の関係者の詳細な資料と紐づけられる内容で整理する
説明	ToBeの要求に応じてAsIsの扱いが異なるので、後で調べられるようにインデックス的に整理する	システム化関係者が共有できる要件レベルで全体俯瞰できるように整理する

把握対象			
対象	業務を理解する	システムの機能を理解する	ビジネスルールを整理する
説明	業務とBUCを洗い出し、関わるビジネス要素を整理し対象を理解する	UCを洗い出し入出力を整理することで、システムの機能を把握する	バリエーションとUCを洗い出し、そこに結びつくるビジネスルールを整理する

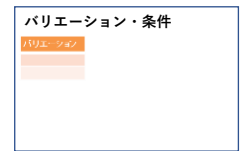
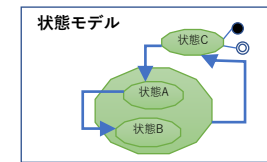
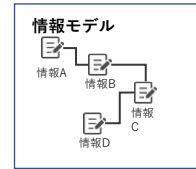
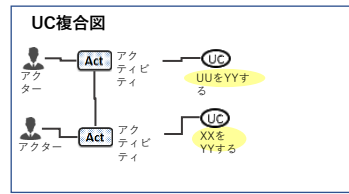
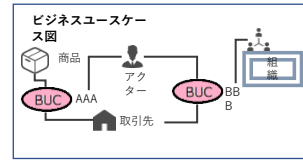
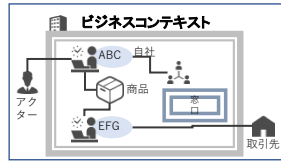
プロジェクト状況			
状況	対象システムを開発した人がメンバーに居る	対象システムの使い方を理解した人がいる	対象システムの業務を理解した人がいる
説明	開発者が比較的信頼がおける 開発時の資料 を主として活用し、補助的に他の資料を使う	入出力（画面、イベント）に関わる情報 を集め、ボトムアップでUCとしてまとめる	トップダウンで 業務、BUC の階層化で分析を進める

既存システムを目的別可視化のイメージ



- ToBeの要求に応じてAsIsの扱いが異なるので、後で調べられるようにインデックス的に整理する

ToBeに向けた可視化



- 網羅的に業務とビジネス要素を整理
- ToBeの議論の広がりのために業務を網羅的に出す

- 典型的なBUCを洗い出す
- BUCと関わるビジネス要素との関係を把握する

- BUCとビジネス要素との関係から仕事の手順としてアクティビティを出す
- システム化するところにUCを追記する

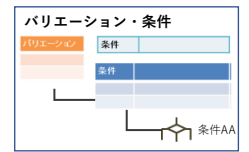
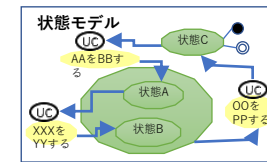
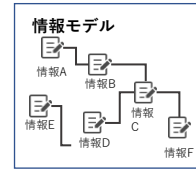
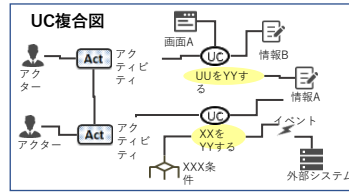
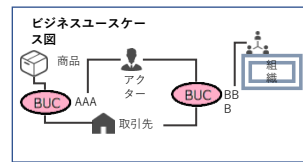
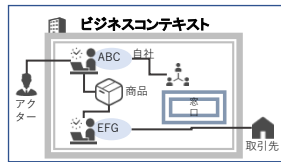
- UC複合図までを作成する中で把握した、情報を表す言葉を構造化する

- UC複合図までを作成する中で把握した、状態を表す言葉を構造化する
- 遷移にUCを当てはめない

- ビジネス要素を中心にバリエーションを整理する

- システム化関係者が共有できる要件レベルで全体俯瞰できるように整理する
- 企画、サポートなどの各部門の詳細な資料につながる接点をもつ

保守開発に向けた可視化



- 網羅的に業務とビジネス要素を整理

- 網羅的にBUCを記述する
- BUCとビジネス要素との関係を明示し、関係部署との接点を明確にする

- BUCとビジネス要素との関係から仕事の手順としてアクティビティを出し、現場作業と突き合わせて整理する
- システム化するところにUCを追記し、画面、イベント、情報を接続する
- 状態モデル、条件などにつなげて精度を向上させる

- UC複合図までを作成する中で把握した、情報を表す言葉を構造化する
- UCにつなげて網羅的に情報を整理する

- UC複合図までを作成する中で把握した、状態を表す言葉を構造化する
- 遷移にUCを当てはめ、UCの精度を向上させる
- 情報からID管理するものを取り上げ状態の

- ビジネス要素を中心に網羅的にバリエーションを整理する
- 条件も洗い出しUCにつなぐ
- 条件につながるバリエーションを全て出す

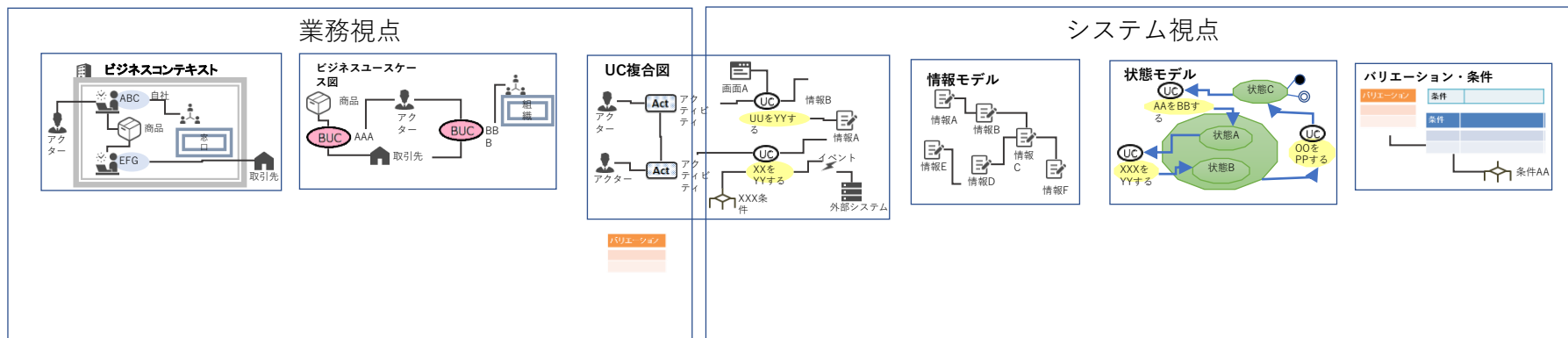
プロジェクトの状況別の進め方

- 3つの可視化対象

- 業務視点 ToBeの業務を考えるために必要な業務
- システム視点 現在のシステム化の範囲を理解する
- ビジネスルール視点 ビジネスを駆動し整合させるルールを明らかにする

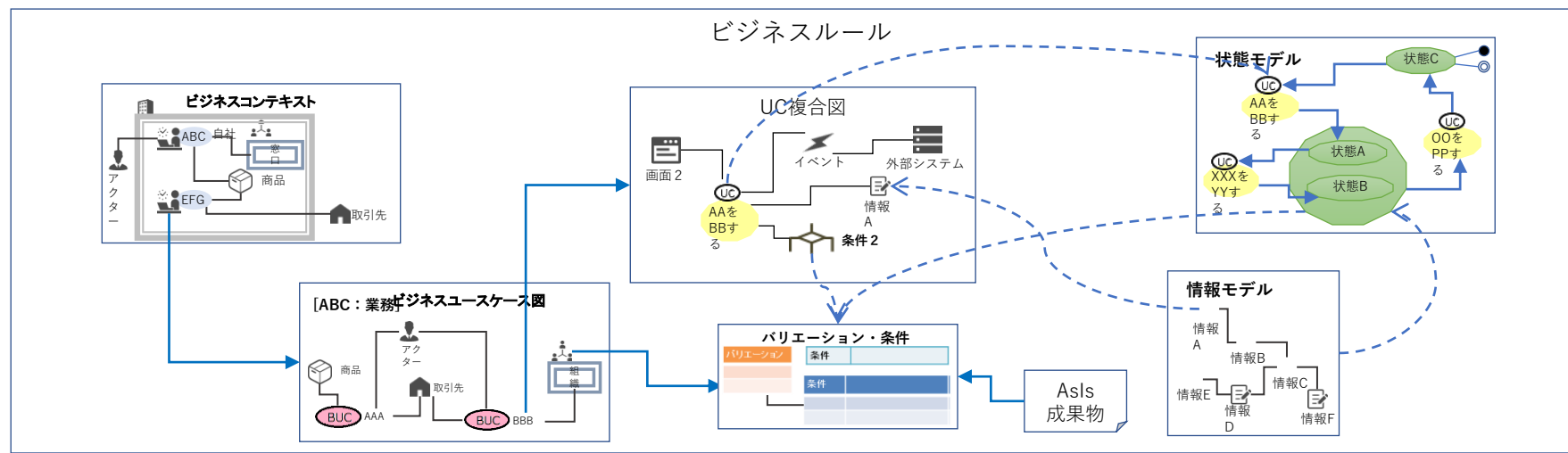
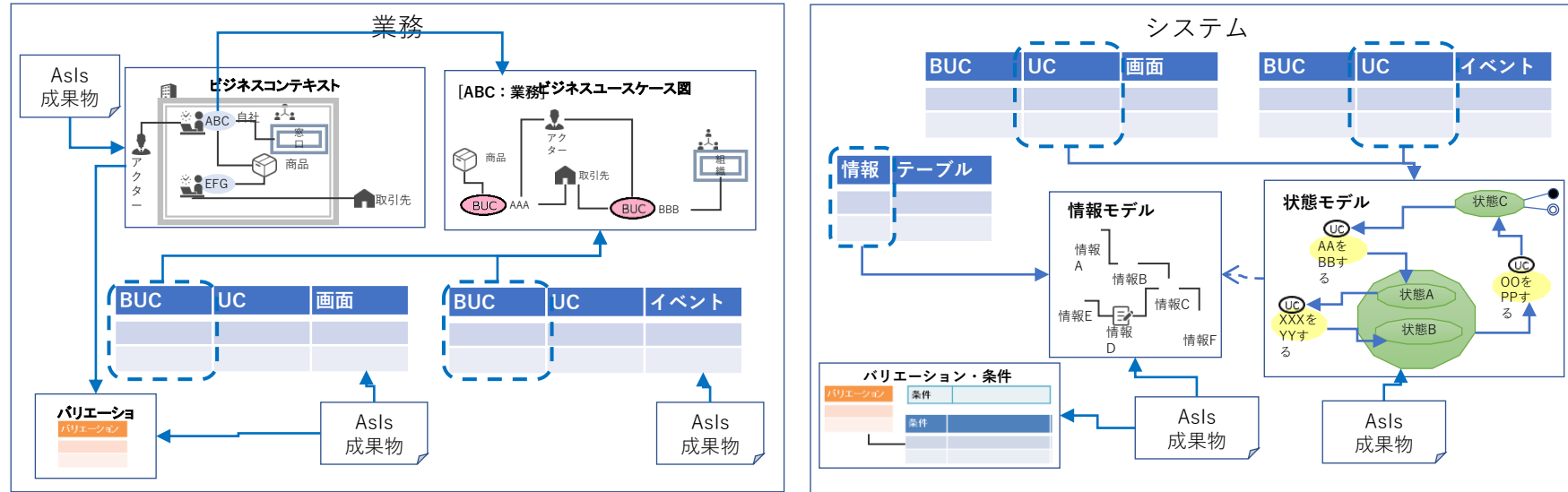
- プロジェクトでヒアリング可能な相手による進め方

- 可視化は業務視点とシステム視点に分かれるがヒアリング可能な状況により各々の進め方が異なる
- ヒアリング可能人
 - 業務を理解する人がいる
 - いる トップダウン
 - いない UCからのボトムアップ
 - 対象システムの使い方を理解した人がいる
 - いる 入出力を整理しそれを中心に確認しながらUCを整理する
 - いない 入出力を仮説を立ててUCを整理する
 - 開発・保守している人がいる
 - いる 開発・保守の資料を主にして整理する
 - いない 入出力中心に整理し、開発・保守資料を補足資料と位置付ける



既存システム把握対象

- 業務を洗い出しシステムが使われる環境を明らかにする
- システムを入出力からユースケースを導き出し表現する
- 業務とシステムをつなぎシステムの使われ方とシステムの役割を把握する
- 業務とシステムをビジネスルールでつなぎ、仕様化への土台とする



既存システムへのアプローチ

- 可視化の目的
 - ToBe開発に向けた可視化
 - ToBe検討に必要な最低限の整理を行う
 - 保守開発に向けた可視化
 - 保守品質を上げるために行う
- プロジェクトの状況
 - 対象システムを開発した人がメンバーに居る
 - 対象システムの使い方を理解した人がいる
 - 対象システムの業務を理解した人がいる
- 可視化対象
 - 業務を理解する
 - 業務とBUCを洗い出し、関わるビジネス要素を整理し対象を理解する
 - システムの機能を理解する
 - ユースケースを洗い出し入出力を整理する
 - ビジネスルールを整理する
 - バリエーションとユースケースを洗い出し、そこに結びつくるビジネスルールを整理する
- アプローチ方法
 - 攻め方
 - 入出力から整理する
 - 業務からトップダウンで整理する

新規の要件定義と既存システムの可視化の違い

- RDRAはシステムを可視化する手法なので、新規の要件定義にも既存システムの可視化にも使用できるが、新規の要件定義と既存システムの可視化では表現方法が異なる。さらに既存システムの可視化も保守開発用と、再構築に向けたAsIsの調査では異なる
- その違いはHow（手段）に関わる部分をどこまで表現するかに依存する
- 以下にその違いを全体方針と手段、その例で示す

	領域	新規の要件定義	既存システムの可視化	
			(再構築に向けた可視化)	(保守開発用の可視化)
全体方針		手段を含めずに何を行うかを明確にする	システムに埋め込まれた、ビジネスルールを明らかにする	保守開発に役に立つために現状の可視化
手段	関心事	アーキテクチャが確定していないので、Howの要素は極力排除する	アーキテクチャが変わることが予想され、システム機能も変わる可能性があるため、ビジネスルールの洗い出しに注力する	仕様の可視化に関心があり、仕様に手段が関係する場合はその部分も含める
	扱い	手段ではなく、その手段を選択する元になる要求を非機能要求として明示する	手段に関わる部分は時間の無駄になるので扱わない	システム連携方法やバッチなどHowに関わる部分が仕様に影響する場合はHowも表現する
手段の代表例	システム間の連携方法	システム間の連携方式はアーキテクチャとして方針を決めるので、手段となる連携方式は表現せず「イベント」として抽象化する	システム間の連携は相手の仕様（タイミングや手段）を明らかにする	システム間連携は変更が入りやすいところなので、連携方式（ファイル、APIなど）を表現してもよい
	バッチ	バッチにするか否かは、手段の選択なので要件定義の中では、「ユースケース」として表現する。バッチ処理の要因となりうる、時間的制約（締め日、時間）や外部システムとの制約（連携タイミング、データ量）を表現する	バッチからビジネスルールを抜き出す	バッチが締め時間などのビジネスルールに関わっている場合は、バッチとして表現する。その場合は「バッチ」アイコンで表現する
	画面などの入出力	必要な入出力を洗い出す。振る舞いはUI標準などに任せる	人が関わる入出力でビジネスルールがあるものだけを扱う	物理画面につながる起点となる画面を中心にまとめる

既存システムの調査アプローチ

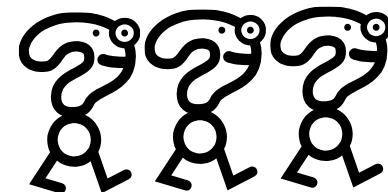


何を調べたいのか

- 何を調べたいのか
 - 「現在のシステムはこうなっている」
 - 「次のシステムの方向性はこうだ！」
 - 「だから次のシステムはこうする」
- 必要なことを判断出来る情報が重要！
 - 判断するためには「何を何のために」が分かる必要がある
- つまり つじつまのあう説明ができる

目指すべきは

- 実装（プログラム）に左右されずに仕様を明らかにする
- システムにとって大事なことは
 - 何ができればいいのか
 - どう整合しているのか
 - 主要な機能は？
 - 主要な情報は？
 - 機能と情報の関係は？
- そこで…
 - 仕様を整理するためにRDRAを活用する



既存システムを調べるときのスタンス

- 既存のシステムを調べているんだ
 - モデルと既存のシステムの関係は？
 - 既存のシステムは綺麗ではない
 - 綺麗なモデルは現実と離れていく
 - 綺麗なモデルは結局何を表しているんだ！
- 現実とつながりながら現実の混沌に影響されずに整理する
- そのためには…
 - 前提
 - 細かなルールにこだわらなくていい
 - 既存システムの仕組みは無視していい

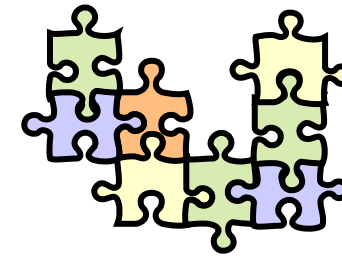
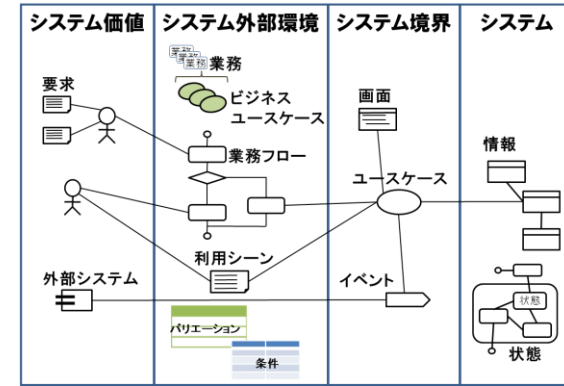
整理方針

• 進め方

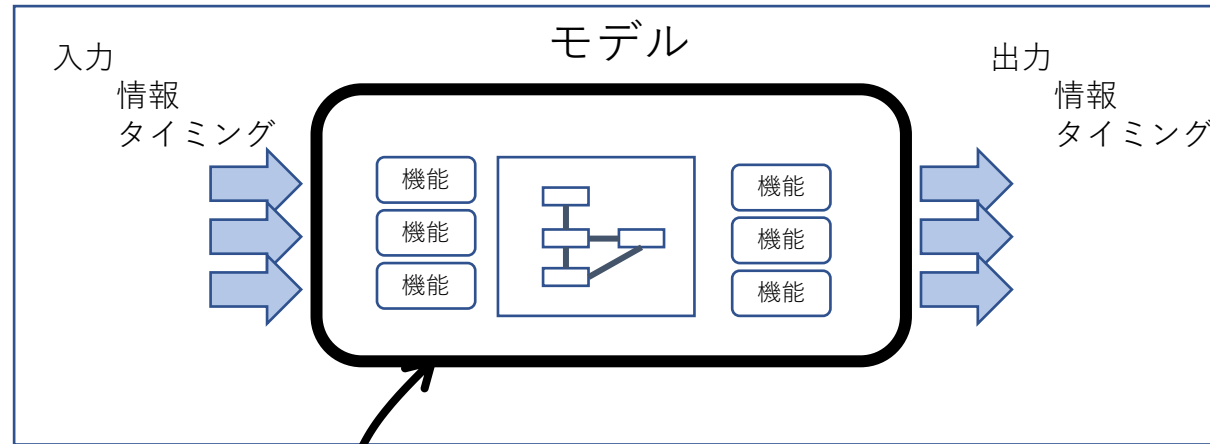
- 得られる資料、担当者の知識量により進め方は変わる
- 状況別進め方
 - 対象業務が分かる場合
 - トップダウンで業務整理から進める
 - 通常のRDRAの進め方に近い
 - 業務が見えていないが、システムの入出力は手に入る
 - ボトムアップで入出力から組み立てる
 - 集まった資料を分類 大きなまとまりを整理する
 - DBから主要なテーブルを整理する
 - 入出力からユースケースの単位を類推する
 - UCと主要なテーブルの関係を付ける
 - 状態の洗い出し
 - UCとテーブルの関係と状態を組み合わせる

分析方針

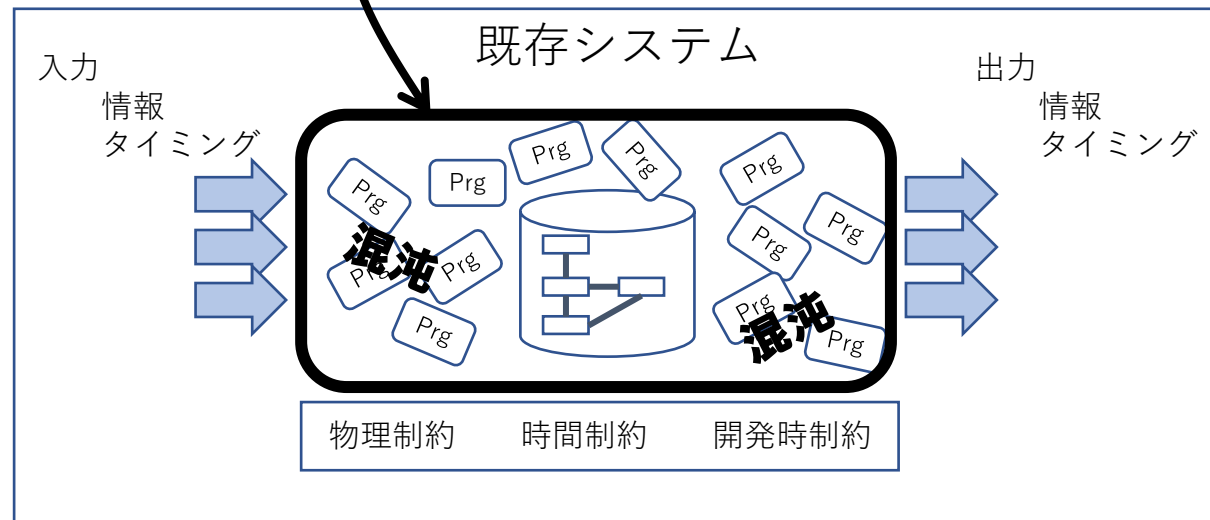
- 集めやすい情報から集める
 - RDRAで形を作る
- 抜けたピースを探す
 - RDRAでつながらない部分を探す
- 集めた情報の特徴から切り込む
 - 分類する
 - RDRAのモデルに当てはめる
 - 関係をつかむ
 - RDRAのモデルのつながりで抜けたピースを探す



現実とつながりながら現実の混沌に影響されずに**整理**する



システム境界のレベルで一致させる



認識（調査）する情報

- 入出力
 - 画面 稼働しているシステムの画面のハードコピー
 - 帳票 //
- 通信 API仕様
- データ交換
 - File Fileレイアウト
 - DB テーブル定義書
- データ
 - テーブル定義書（スキーマ定義から出力）
 - ファイルレイアウト

ボトムアップアプローチCRUD表

テーブル

分類をモデルとあわす



プログラム



支援内容と費用

- 概要
 - 可視化作業そのものはお客様に行っていただき、1 2回の打合せでGoogle SpreadSheetを使ってRDRA手法で可視化します
 - 初回に対象システムの概要と関連する資料の説明を受け、以後の進め方を提案、毎回の打合せの進め方、定義内容などをリードし、確実に1 2回の打合せでシステムの枠組みが把握できるレベルで要件を定義します
- 内容
 - 形態：ビデオ会議による打合せ
 - 方法：1 2回の打合せによる可視化支援、3ヶ月のメールフォロー
- 方法
 - 1 2回の打合せを週1回または2回で実施
 - Google SpreadSheetでシステムの可視化を支援
- 成果物
 - Google SpreadSheetによる可視化
 - 成果物例：[クラウドサービスの定義例](#)
- 費用
 - 150万円/1回（1 2回の打合せ＋3ヶ月メールフォロー）
 - 以後の追加作業は1回の打合せ毎の費用計算となり、10万円/1回の打合せ