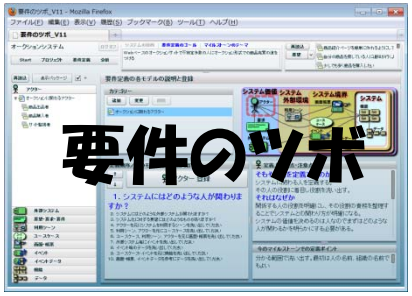
要件定義のコツ「実際にどう行うのか」

JavaFesta2010

Who am I

- ㈱バリューソース
  - 代表取締役　社長
  - 神崎　善司
  - zkanzaki@vsa.co.jp
  - はてな:good_way
  - twitter:zenzengood



- 普段は
  - 要件定義のコンサルテーション
  - オブジェクト指向やUMLのセミナー開催
  - 要件定義支援ツール「要件のツボ」開発

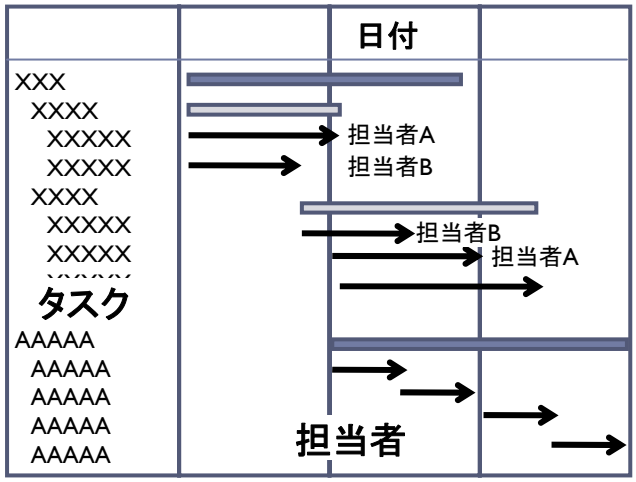
  - 「上流工程勉強会　札幌」主催:　第2回　11月12日　「包括フレームワークの開発プロセスの分析」

- 要件定義との関係は
  - 20年ほど前からオブジェクト指向を中心にシステム開発全般のコンサルティングを行う
  - 10年ほど前から上流工程を中心としたコンサルティングを行う
  - その間一貫してモデル中心で開発支援を行う
  - その経験を活かしてモデルを使った要件定義の手法をまとめる
  - 要件定義を支援するツールを開発

＜昨年の振り返り＞

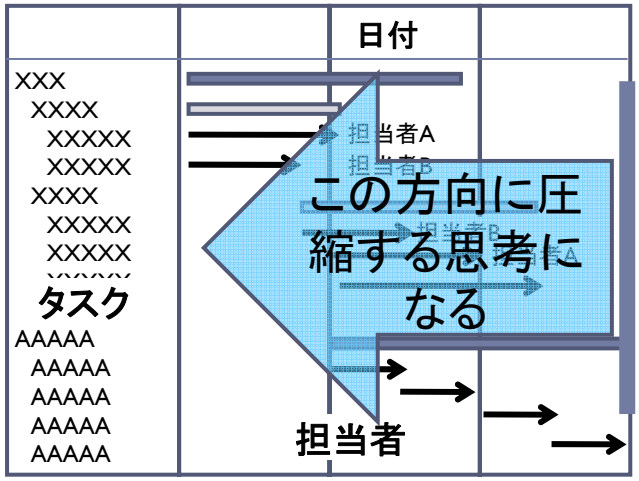
要件定義の進め方と枠組み

ガントチャートが示したいこと



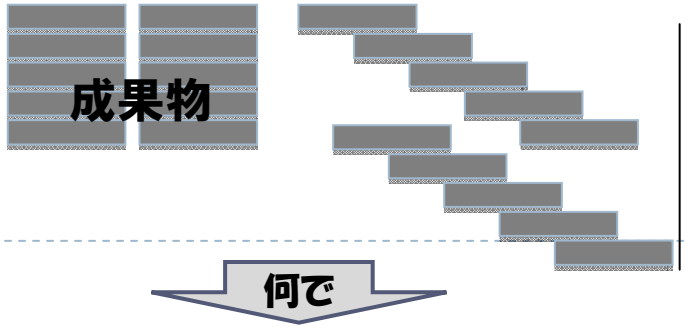
- 以下の前提が成り立つときガントチャートはすばらしいツールである。
  - タスクの洗い出しが正確である
  - 洗い出したタスクを予定時間内に終わらせられる
- ガントチャートで示したいこと
  - 「**この期間と人で大丈夫です！**」

ガントチャートが作業に与える影響

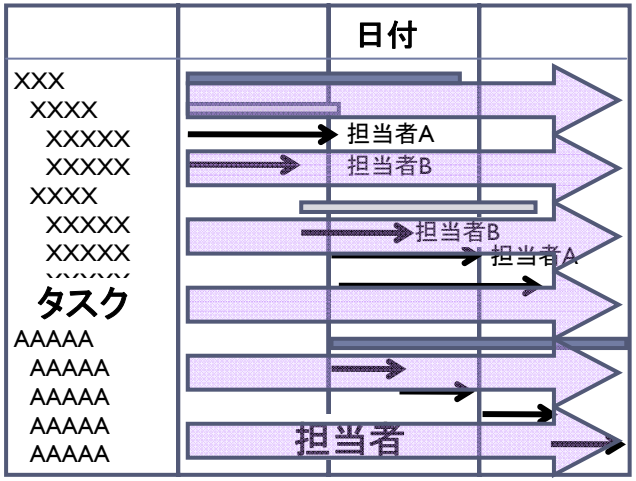


- ガントチャートを作ると…
  - 大抵タスクをつなげると納期に収まらない
- だから…
  - 納期に納めるために並行性を高める

- 結局…
  - ガントチャートは成果物が決まっているという幻想を抱かせる



ガントチャートからくる弊害

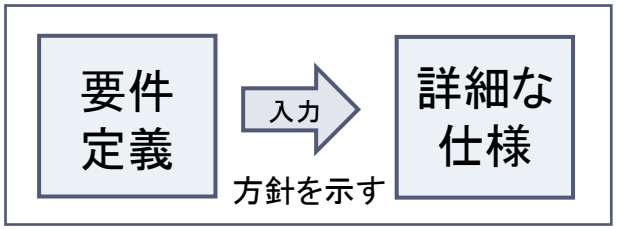


- 何が起こっているのか…
  - 早い段階から並行的に個人作業を始める
    - システムの全体的なことについて打ち合わせする機会がない
    - 誰も全体を見ていない
  - 並行作業を実現するためには成果物ベースで管理するのが容易
    - 個々の成果物の完成で進捗を測る

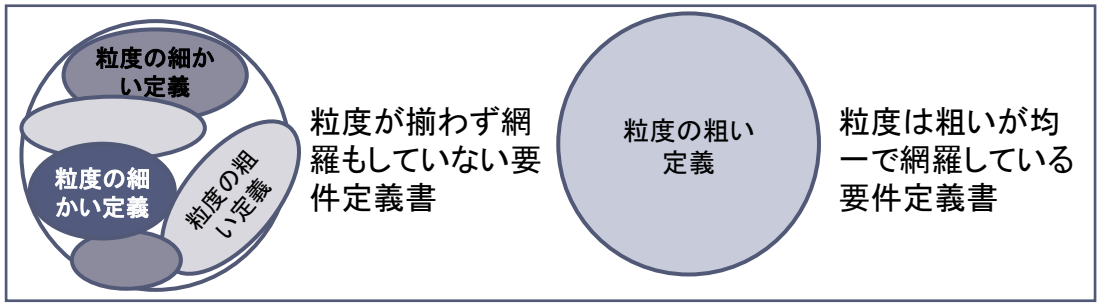
どうすればいい？

要件定義で本来大事なこと

- 正確な情報よりもマクロ視点での方針を重視する
  - 有効なシステムを作るための方向性を示す要件が大事

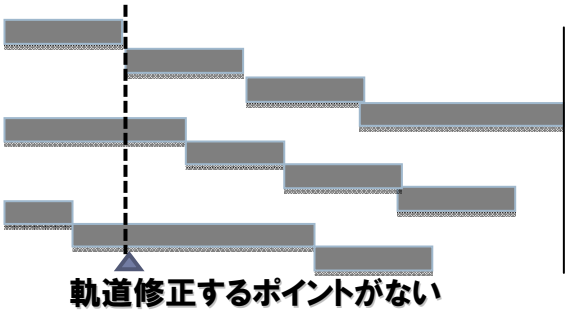


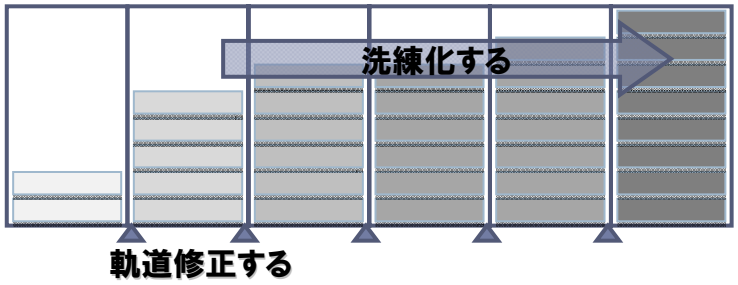
- 粒度の揃わない資料よりも粒度が粗くても精度が均一なドキュメントの方が役に立つ

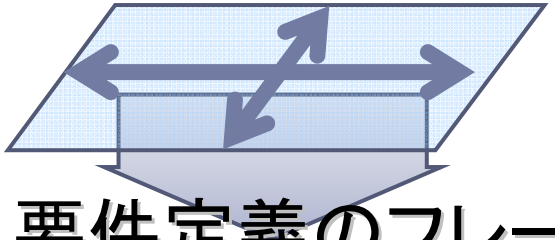


- 一人では方向性は決められない
  - 関係者が集まって検討するから方向性が決められる

状況に応じた進め方をしましょう！



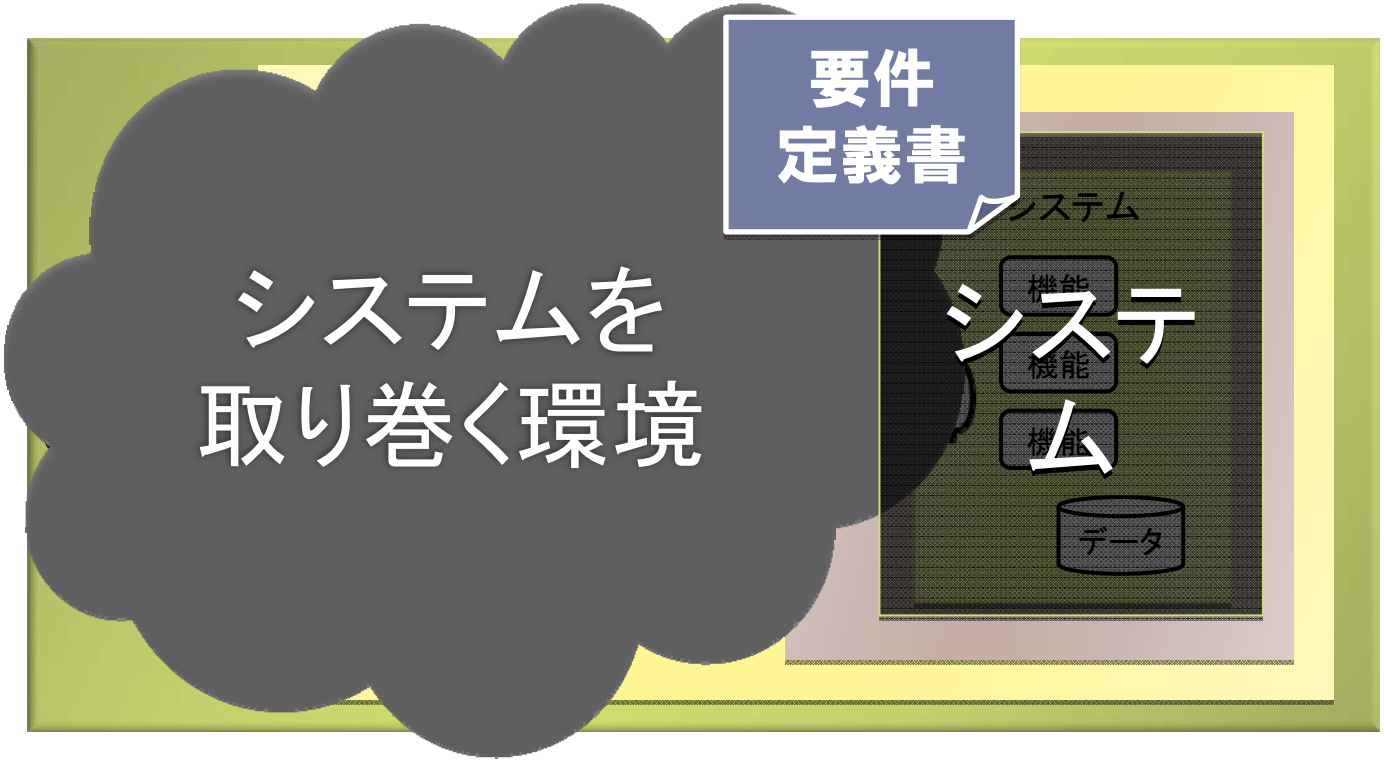




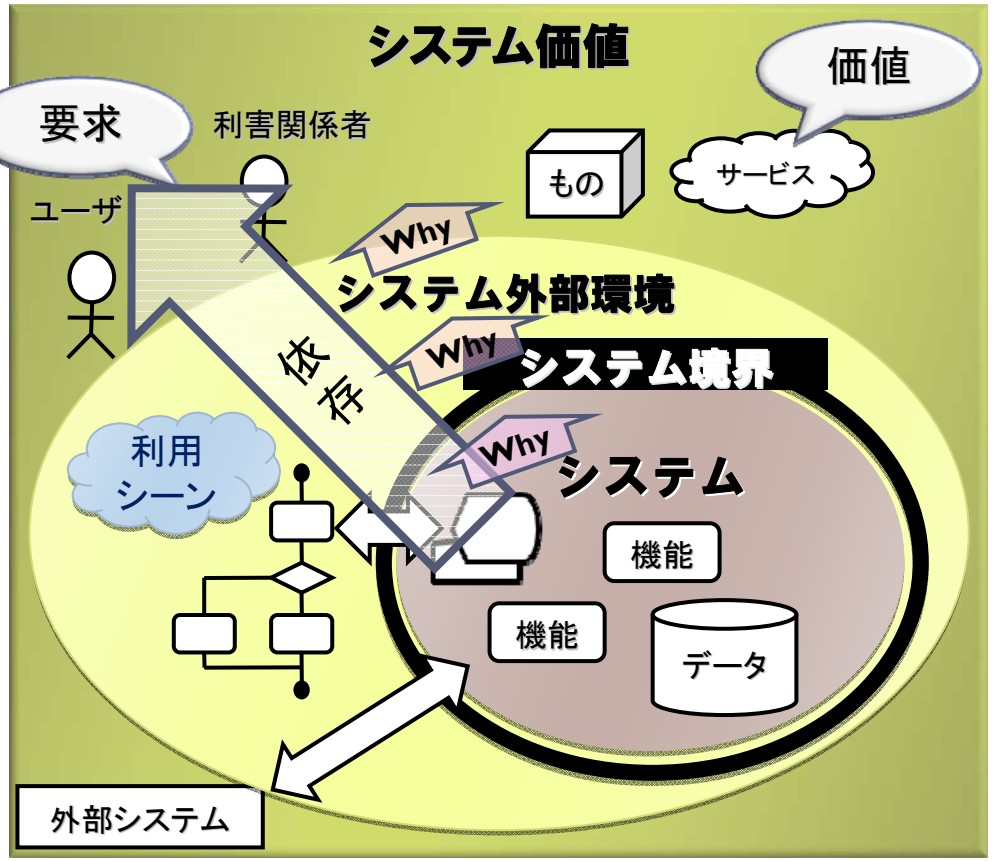
要件定義のフレームワークが必要

- 状況の変化に合わせて進め方を変える
  - 成果物は徐々に洗練化させる
  - 軌道修正しながら進める
    - ガントチャートは軌道修正のポイントが無い
- だから…
  - パラメータ「時間」に軸足をおく
  - 時間の枠の中で成果物を調整する
  - 状況に合わせて軌道修正して進める
- 状況に適合するためには…
  - 軌道修正するための状況把握が必要
  - そのためには **全体を広く** とらえ徐々に深掘りすることが必要
- 「全体を広く」とらえるためには…
  - 要件定義の構造を定め　全体の構造を俯瞰する

要件定義にはシステムを取り巻く環境を含める



要件定義の枠組み



1.【システムの価値】を捉える

- 対象業務に関わる人を洗い出す
- 関係する外部システムを洗い出す
- 要求を捉える

2.【システム外部環境】を捉える

- 対象業務の流れを捉える
- 対象業務に関わる概念を明らかにする
- システムの利用シーンを明らかにする

3.【システム境界】を捉える

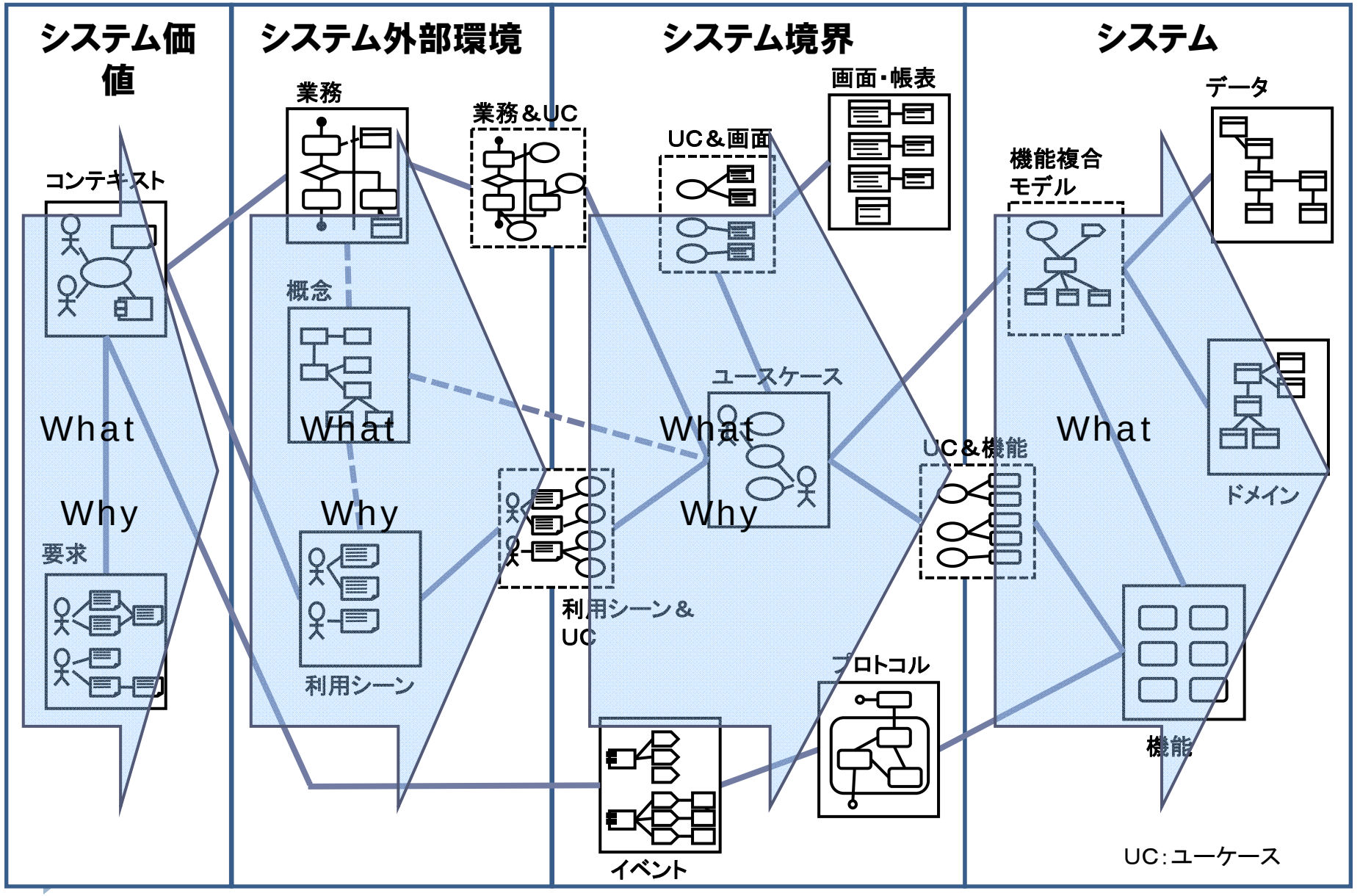
- ユーザインターフェースを明らかにする
- 外部システムとのインターフェースを明らかにする

4.【システム】を定義する

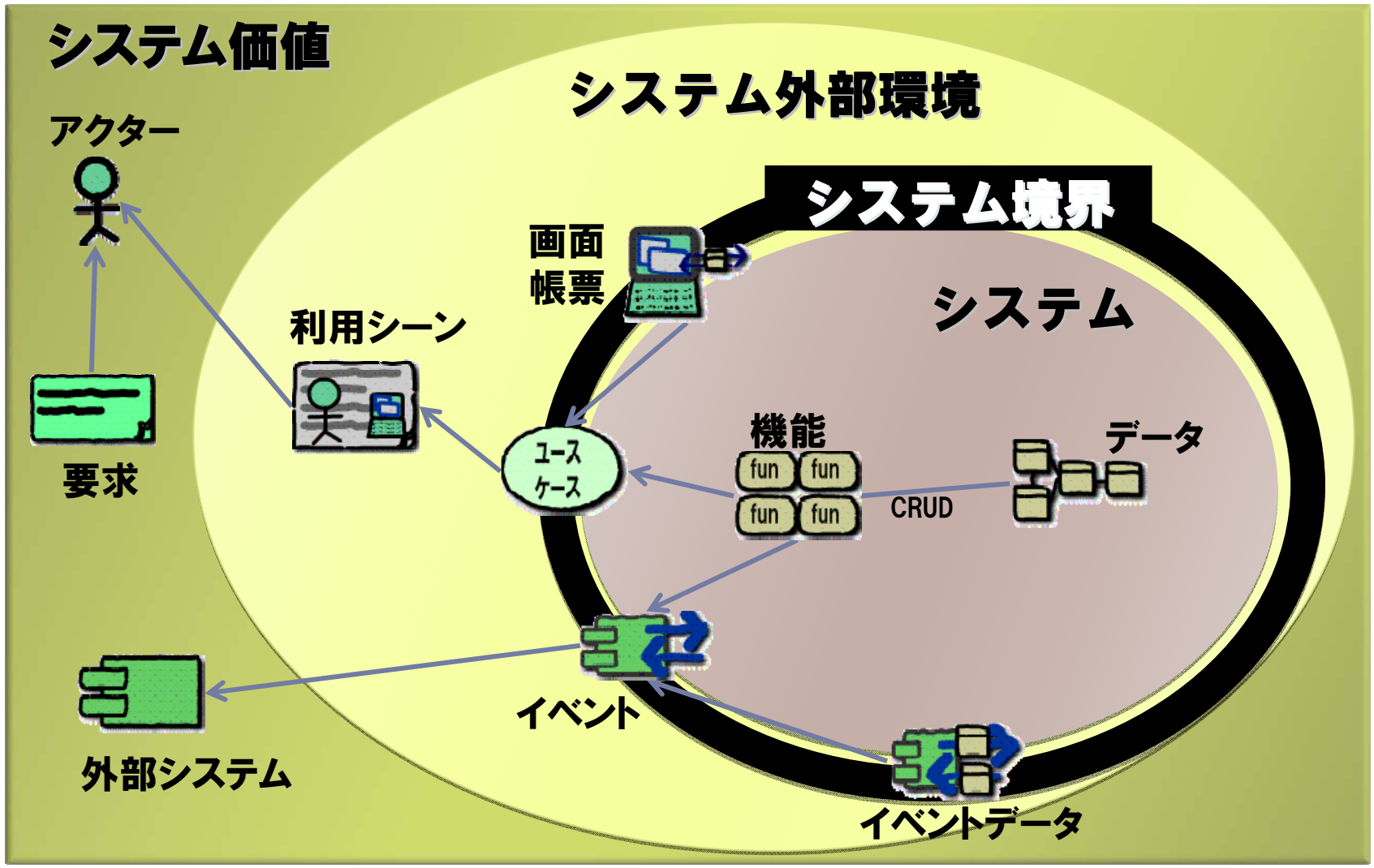
- 機能を明らかにする
- データを明らかにする
- ドメインの構造を把握する

全ての情報をつなげる　UML

関係ダイアグラム



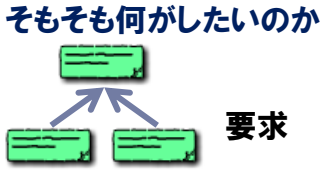
要件のツボ


システム価値
システム外部環境
アクター
システム境界
画面
帳票
システム
利用シーン
要求
ユース
ケース
機能
fun
fun
fun
fun
CRUD
データ
イベント
外部システム
イベントデータ

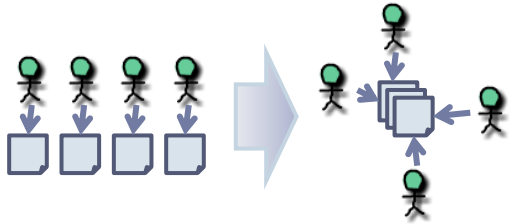
まとめると

- システムの全体像をつかみ方向性を示す
  - 全体像をつかむためには？
    - システムを取り巻く環境からシステムを説明できるようにする

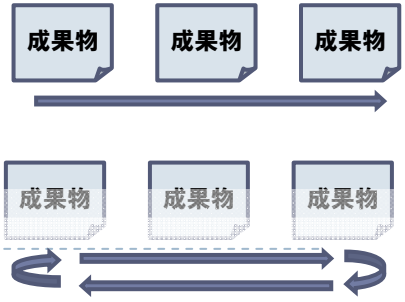
  - 方向性を示すためには？
    - 要求を整理してシステムの目的・価値につなぐ



- 個人作業からチーム作業に移行する
  - チームとして効率的に作業するためには？
    - 要件を導き出す仕組みを用意する

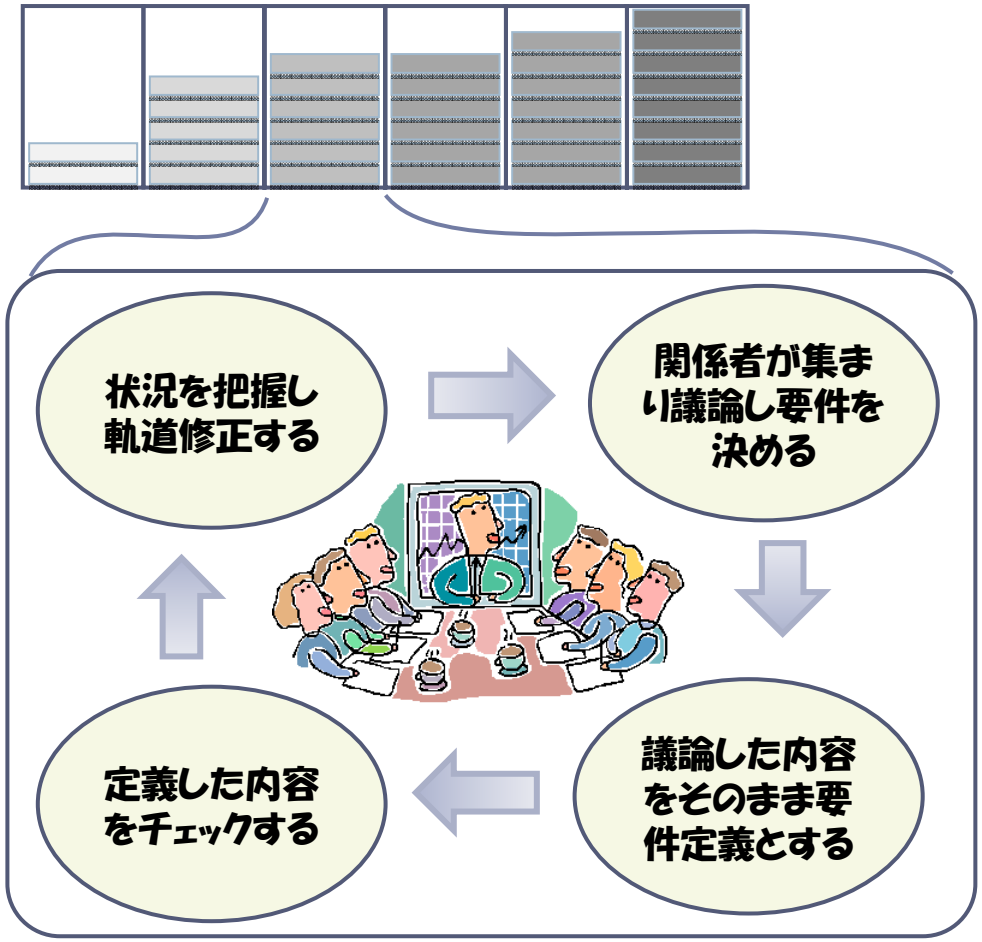


- 軌道修正しながら作業を進める
  - タイムボックス内で意志決定できる状態にするには？
    - 様々な角度の情報をつなげて少量作成する



進め方のポイント

有効な要件を定義するためには



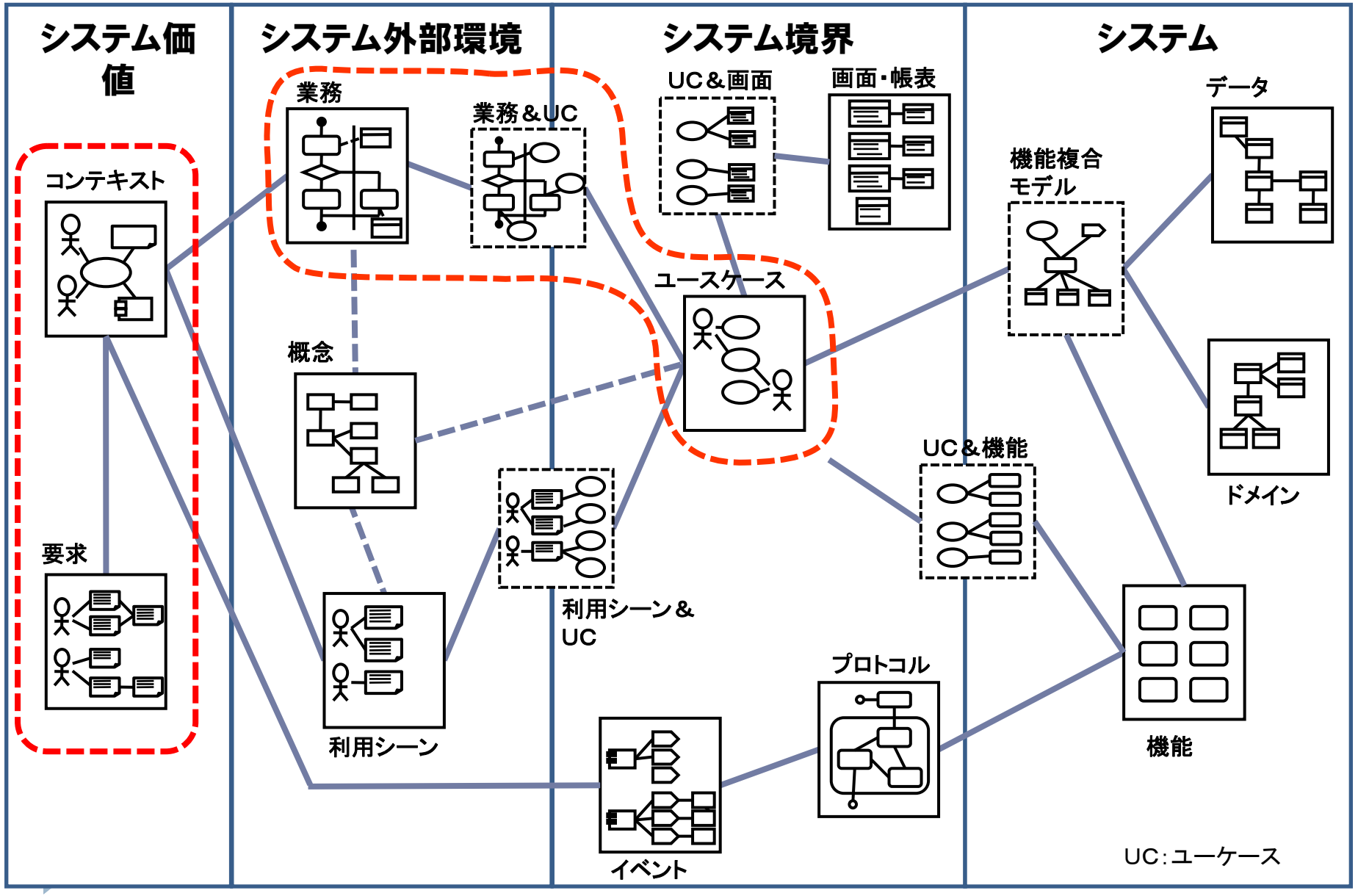
- 軌道修正できるタイミングをもつ
- 高速に定義できるようにする
  - 議論しながら要件を決める
  - 議論するための方法をもつ
  - 議論した内容をそのまま要件定義書とする
- 状況判断ができるように定義した内容を分析する
  - 定義した内容をチェックする
  - 状況を把握し軌道修正する
- 全体を見通し個別の定義にドリルダウンできるようにする

具体的にどうするの？

UMLを使って行う場合のイメージ

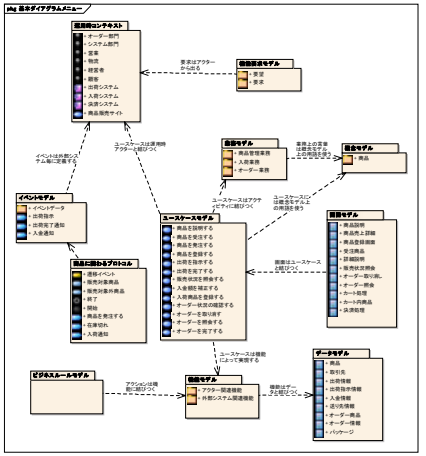
全ての情報をつなげる　UML

関係ダイアグラム

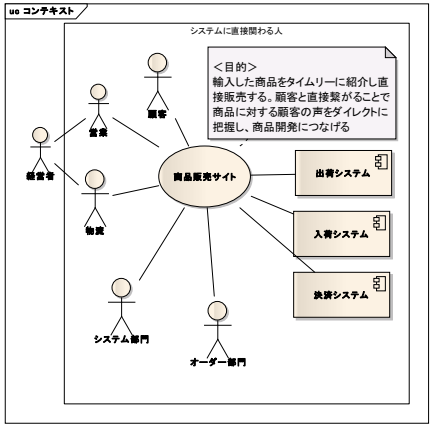


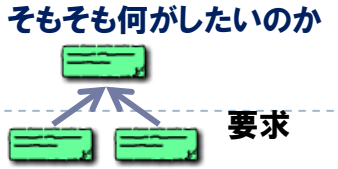
システムの全体像をつかみ方向性を示す

- 全体像をつかむためには？
  - システムを取り巻く環境からシステムを説明できるようにする
    - 全情報を一元管理し素早くその情報にアクセスできる
    - 全体を見ながら説明する
    - 簡単に情報を分類できる



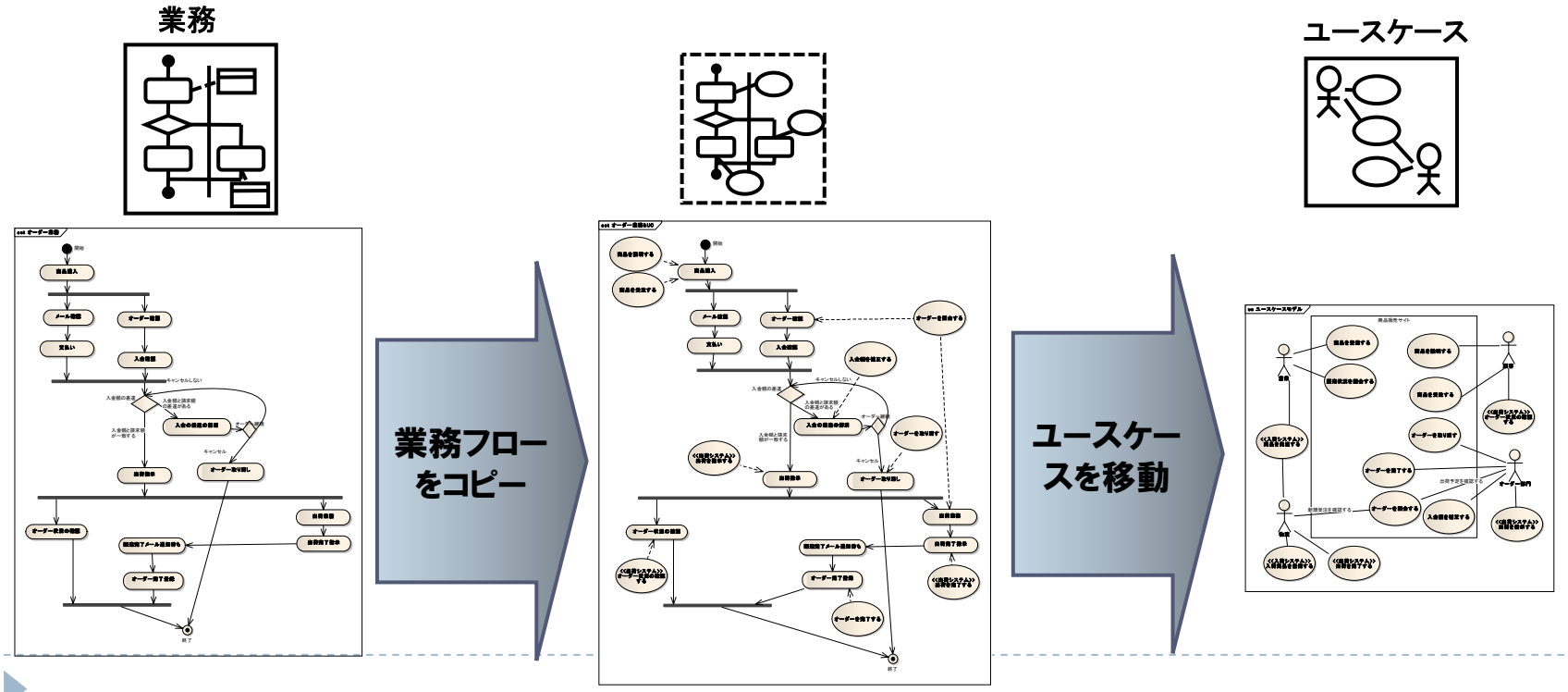
- 方向性を示すためには？
  - 要求を整理してシステムの目的･価値につなぐ
    - 要求を構造化して整理する
    - システムの目的･価値にとしてまとめる





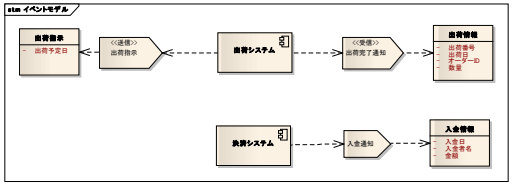
個人作業からチーム作業に移行する

- 個人作業からチーム作業に移行する
  - チームとして効率的に作業するためには？
    - 要件を導き出す仕組みを用意する
      - ダイアグラム作成の手順が決まっている
      - 実体が一つなので何処で修正しても全てが変わる

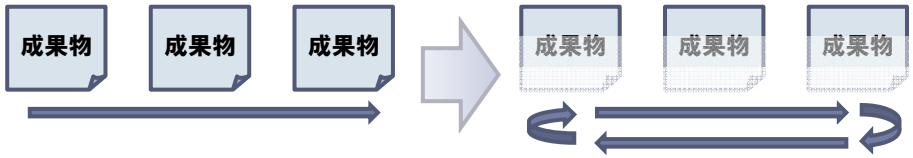


軌道修正しながら作業を進める

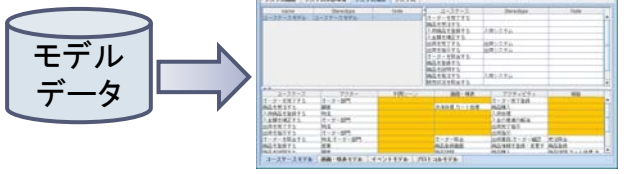
- タイムボックス内で意志決定できる状態にするには？
  - 様々な角度の情報をつなげて少量作成する
    - 一つ一つの重たいドキュメントを作成しない



  - 徐々に洗練化する
    - 一度に完成させない　　定義は相互に依存している



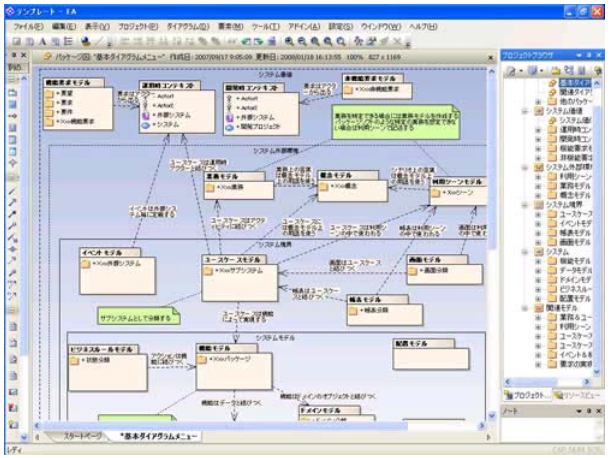
  - 定義情報を再利用してチェックする
    - 関係性を利用することでチェックできる



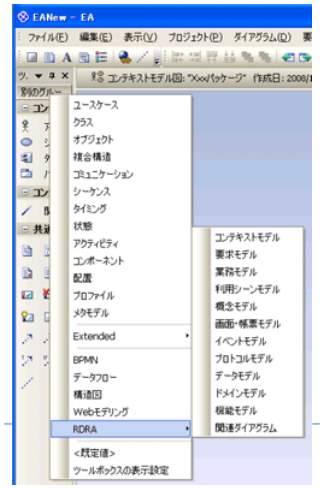
リレーションシップ駆動要件分析の情報

http://k-method.jp

RDRA用テンプレート



RDRA用プロファイル



書籍

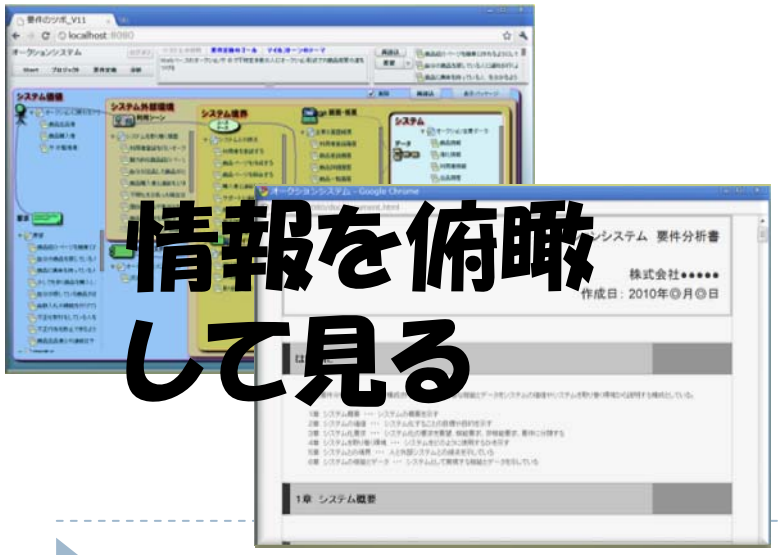
＜リレーションシップ駆動要件分析の情報サイト＞
UMLツールのEnterpriseArchtectのテンプレートやプロファイルがダウンロード出来ます

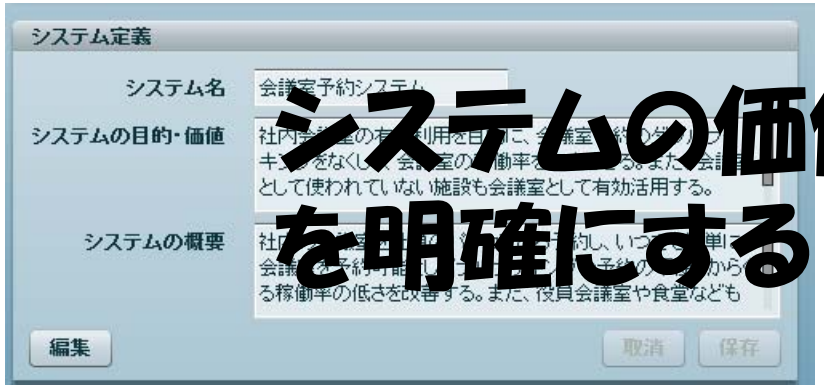
具体的にどうするの？

要件定義支援ツールを使った場合

システムの全体像をつかみ方向性を示す

- 全体像をつかむためには？
  - システムを取り巻く環境からシステムを説明できるようにする
    - 全情報を一元管理し素早くその情報にアクセスできる
    - 全体を見ながら説明する
- 方向性を示すためには？
  - 要求を整理してシステムの目的・価値につなぐ
    - 要求を構造化して整理する
    - システムの目的・価値にとしてまとめる

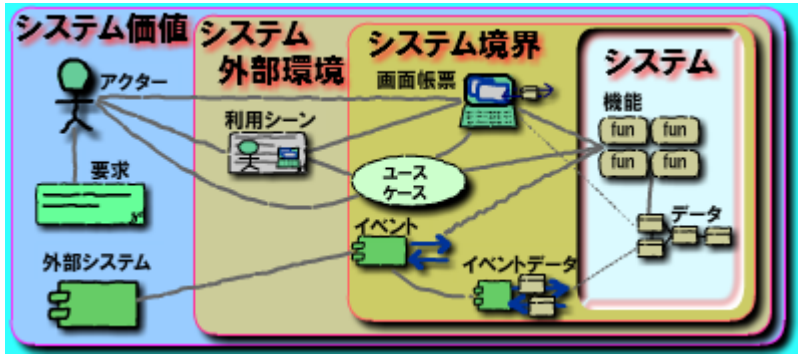


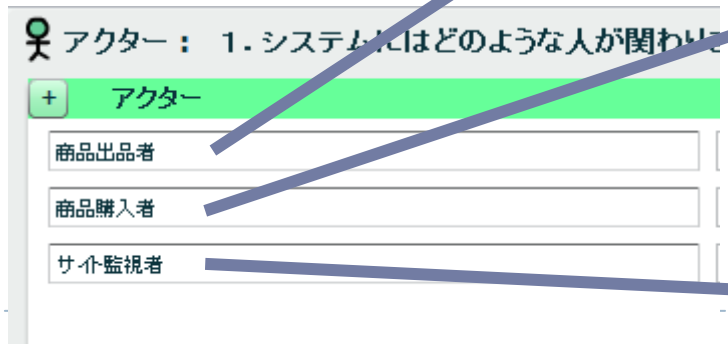


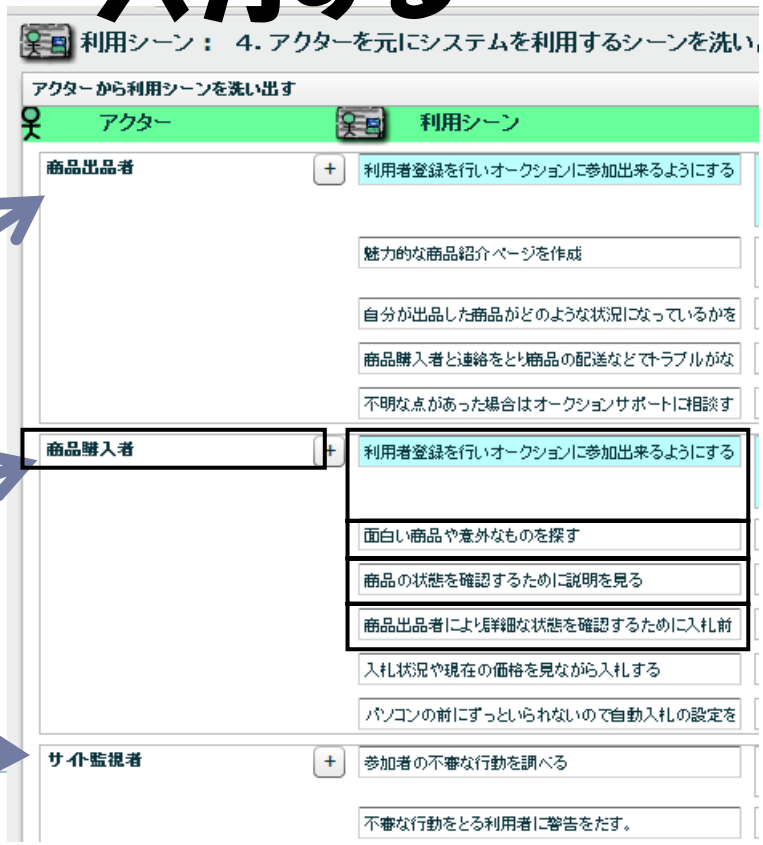
個人作業からチーム作業に移行する

- チームとして効率的に作業するためには？
  - 要件を導き出す仕組みを用意する
    - 情報をつなげて入力する

情報をつなげて入力する





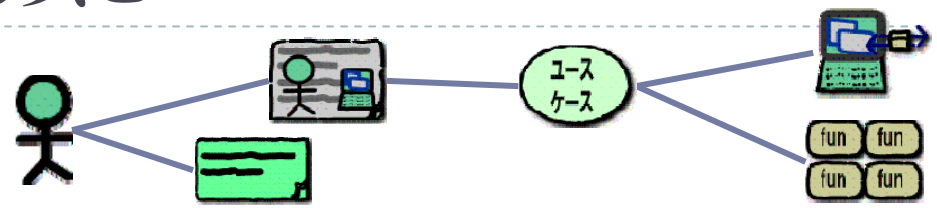


軌道修正しながら作業を進める

- タイムボックス内で意志決定できる状態にするには？
  - 様々な角度の情報をつなげて少量作成する
    - 名称と概要（責務、内容…）

  - テーマを持って作業する
    - ドキュメントの完成ではなく
    - 状態として管理する
    - メンバーの意識を合わせる

精度を向上させるために

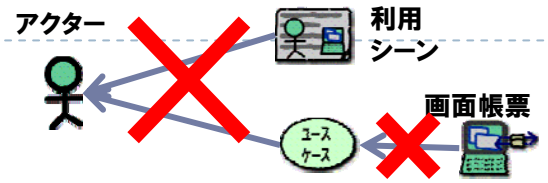


- 分析する
  - つなげて確認する
    - 誰が　どのような要求を持ち　どのようなシーンで
    - どのようなやりとりで　どの画面を使って　どんな機能を使うのか
  - データのライフサイクルから機能の抜けを見つける
    - CRUD表を作ってデータ側から必要な機能を確認する

	Create	Read	Update	Delete
データA		機能C		
データB	機能A	機能B		
データC			機能D	

矛盾
がある

検証する

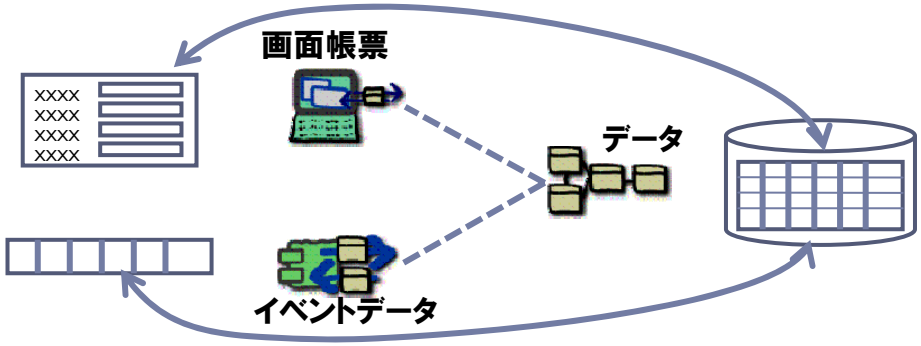


- **関連づけの漏れから精度を測る**
  - 関連チェック:
    - アクターが何にも関わっていない
    - ユースケースに画面・帳票が関わっていない
    - ～～

チェックポイント
アクターが担うべき利用シーンが全て出ているか
アクターが担うべきユースケースが全て出ているか
利用シーンには必ずユースケースが結びついているか
利用シーンには必ずアクターが結びついている

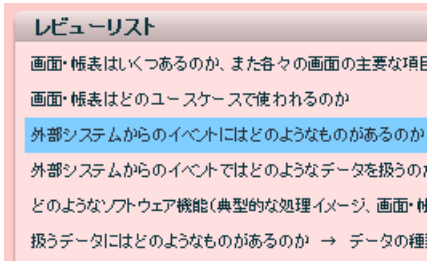
カテゴリー	アクター	関連する ユースケース
主要アクターカテゴリ	会議室登録者	
主要アクターカテゴリ	利用者	

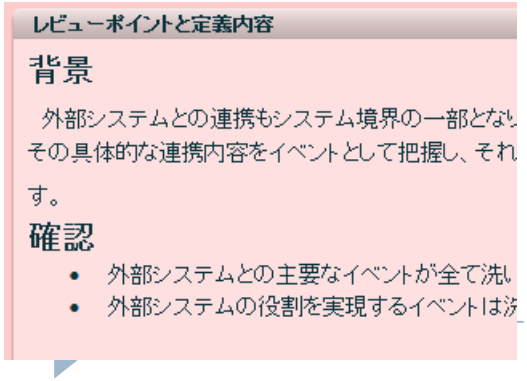
  - 項目チェック:
    - 画面上の項目がデータ上に定義されていない
    - イベントデータの項目がデータ上に定義されていない



レビューする１

- **従来のレビューの問題点**
  - 成果物ベースのレビューは眠たくなる
    - ドキュメントの構造に沿った説明になる
    - 思考が限定的になる　広がりが得にくい
    - 説明を広げるのは個人の力量に負うところが大きい
- **レビューのポイントを示し定義内容を確認する**
  - レビューの観点を確かめレビューの組み立てを行う
    - レビューになれていないような場合に予め個々の情報に対する観点を確認しておく
  - 要件定義はどのような組み立てになっているか
    - 要件定義の組み立て方そのものを説明し、その上で個々の要件定義の観点を示す
  - ここでは何を確かめればいいのか
    - 例
      - 観点：利用するシーンとして主要なものが洗い出されているか



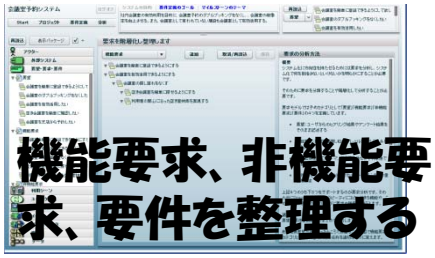


レビューする２

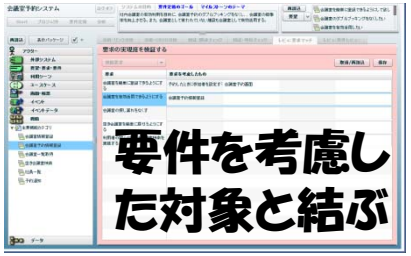
- **要求の考慮度を確認する**
  - 要求「～できること」を考慮したところを関連づける
    - 要求を分けて管理する（例：ステークホルダーの要求を確認する）
      - **要望、機能要求、非機能要求、要件**
  - 1．既にある**要求**を**要望**としてまとめる
    - 「要件登録」で**要望**を連続的に登録する
  - 2．機能要求、非機能要求として整理する
    - 「要求分析」で**要望**から**機能要求**、**非機能要求**にコピーし整理する
  - 3．ステークホルダーが関心をもつものを要件として洗い出す
    - 「要求分析」で**機能要求**、**非機能要求**から**要件**にコピーし整理する
  - 4．レビューの場で要件を突き合わせる
    - 「要求マッチ」で考慮したものを突き合わせる

要件登録

要求分析



要求マッチ



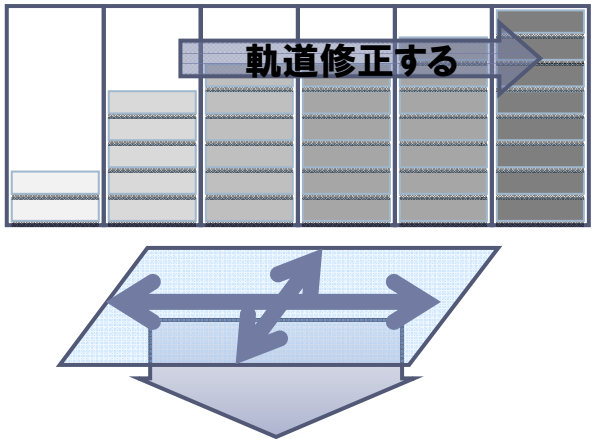
「要件のツボ」

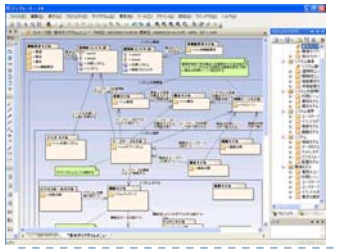
コミュニティ　エディション

http://www.vsa.co.jp/kaname

コミュニティ　エディション
は無料で使用できます

まとめ





- 要件定義は不明確なことを決めていく工程
- 状況に応じて軌道修正しながら進める
- そのためには広く網羅的に要件をまとめる
- 広く網羅的に進めるために要件定義のフレームワークを持つ
- タイムボックス毎に洗練化する
- 素早く作業するためにツールを使い一元管理する