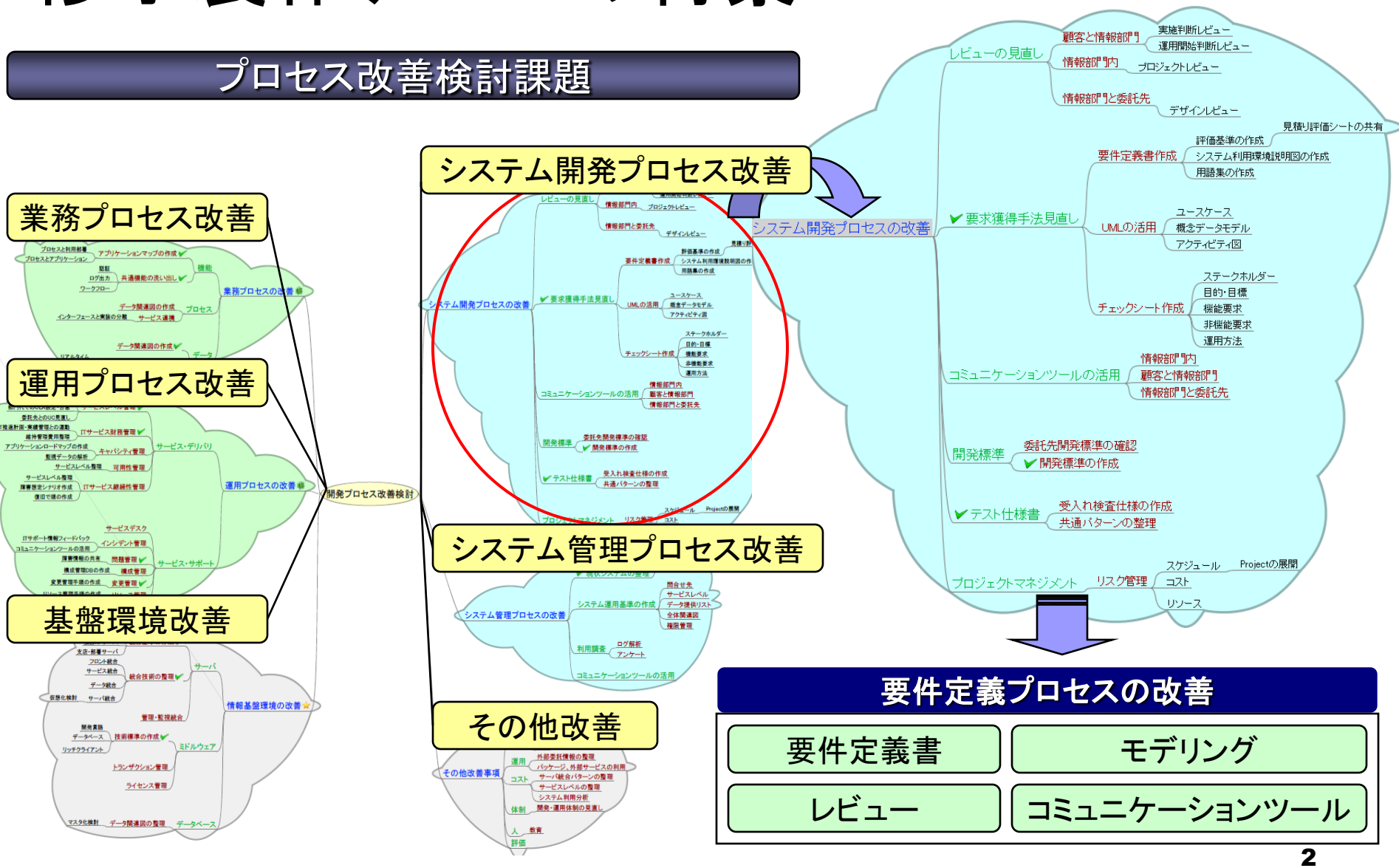
トップエスイー修了製作

ユーザ企業における、要件定義プロセスの標準化提案

鹿島建設株式会社
角川　友隆
t-sumikawa@kajima.com

修了製作テーマの背景

プロセス改善検討課題



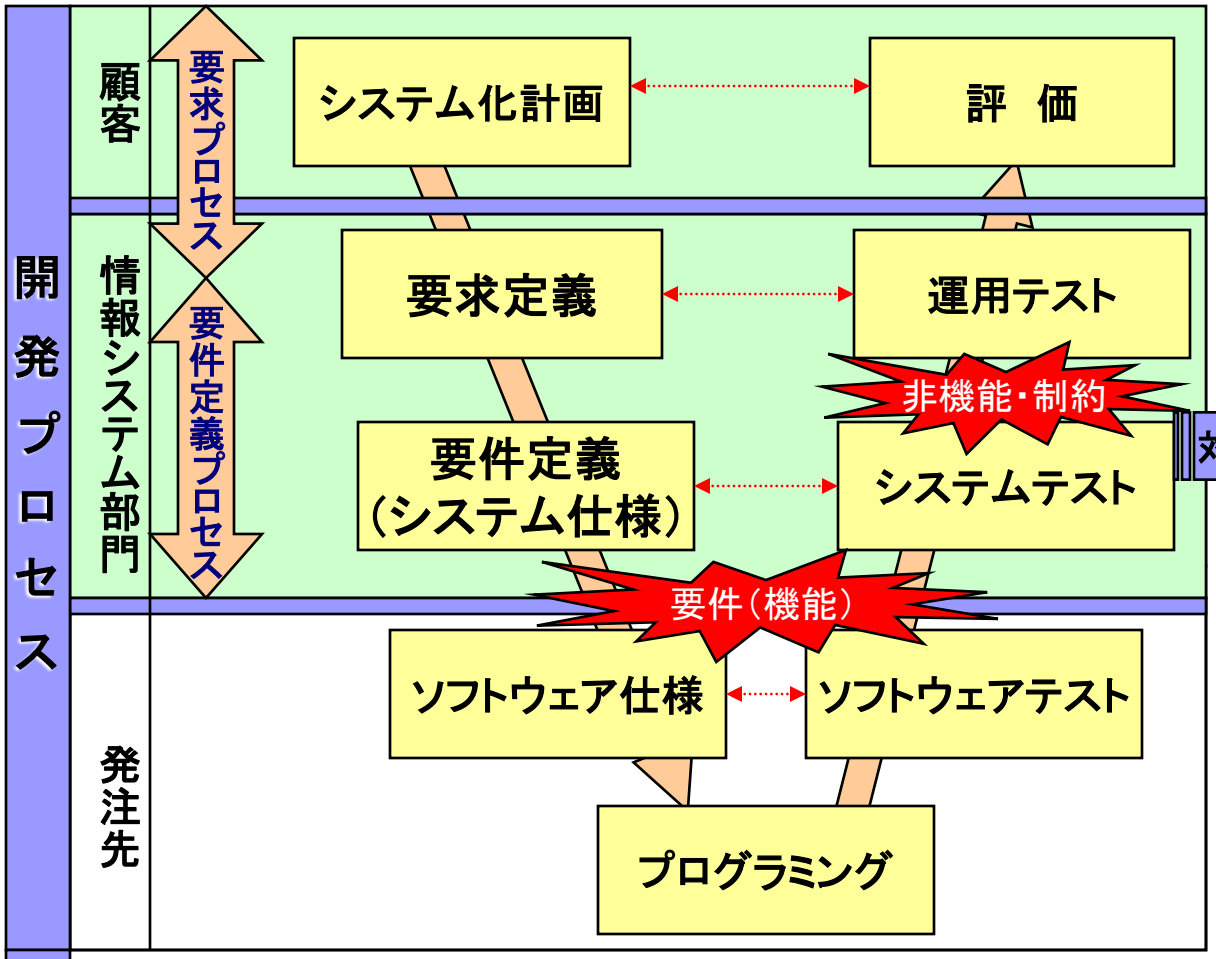
修了製作テーマの背景

- 要件定義プロセスの重要性
  - 利害関係者が複数存在する場合において、不明確な目的や、あいまいな業務仕様などによって、要求仕様変更が発生すると、開発プロジェクトの品質・納期・コストに大きな影響を及ぼす。これらのリスクを最小化するためにも、ユーザ企業における要件定義力の向上が求められる

> 要件定義書をユーザ企業だけで作成しているのは22%
> ※ JUAS 平成15年12月 ユーザ企業IT動向調査　調査概要報告書より

- 手法・ツールの適用による解決
  - ゴール指向分析手法であるi*フレームワークを用いて、利害関係者要件の定義を行い、IEEE Std 830の形式に基づいた要件定義に変換するプロセスを標準化する
    - 機能　（i*法⇔UML、Matrix分析、IFPUG法）
    - 非機能（FURPS+分析、シナリオ分析、ATAM、チェックシート）

現状の問題(開発プロセス・手法)



共通フレーム2007
(要件定義プロセス)

利害関係者要件の定義

- ニーズ識別と制約
- 業務要件
- 業務環境要件
- 機能要件
- 非機能要件
- スケジュール要件

i＊法 活用

利害関係者要件の確認

- 要件の合意と承認
- 要件変更ルール決定

手法

UML活用課題

- ユースケース図
- アクティビティ図
- 概念クラス図

- システム全体像が把握し難い
- ゴールとの関係や入出力資源情報が明確にならない
- 非機能要件に関しては後回しになるケースが多い
- 見積り妥当性の検証が難しい

要件定義プロセスでのi*法利用

i*法を選択した理由

- システムゴール（KAOS）＜ ステークホルダー間の意図的ゴール（i*法）
  - 利害関係者が複数存在する場合のゴールを明確化
- サービス全体像の早期認識共有
  - アクタの本質的な活動とビジネスプロセスのモデル化で全体像を把握
- モデル情報の有効活用
  - 外部ファイルの出力機能（XML、Formal Tropos）・・・St-tool

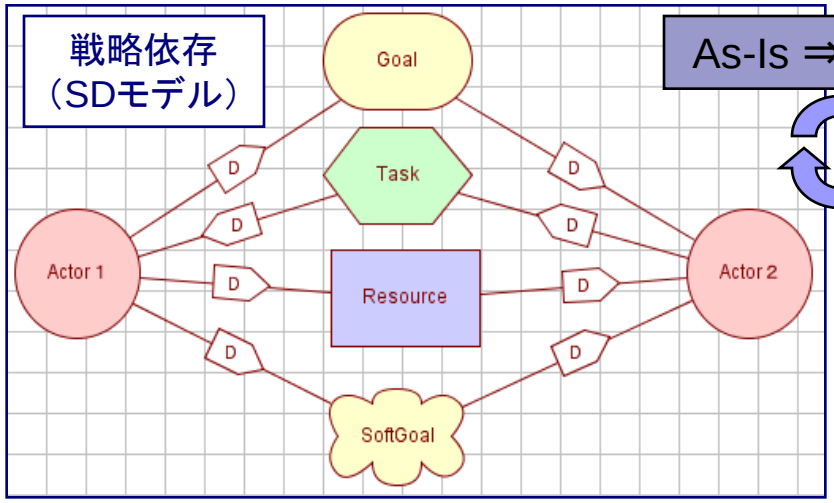
i*法に期待できること

- 要求プロセスでの活用 （トップエスイーでの講義内容）
  - 現状のビジネスを理解（As-Is）
  - システム導入による効果を分析（To-Be）
- 要件定義プロセスでの活用 （修了製作内容）
  - 機能要求候補の抽出
    - タスク分解による機能候補の抽出
    - ビジネスプロセスとリソースの関係を明確化（ゴール⇔タスク⇔リソース）
  - 非機能要求分析から機能やアーキテクチャ要件を抽出

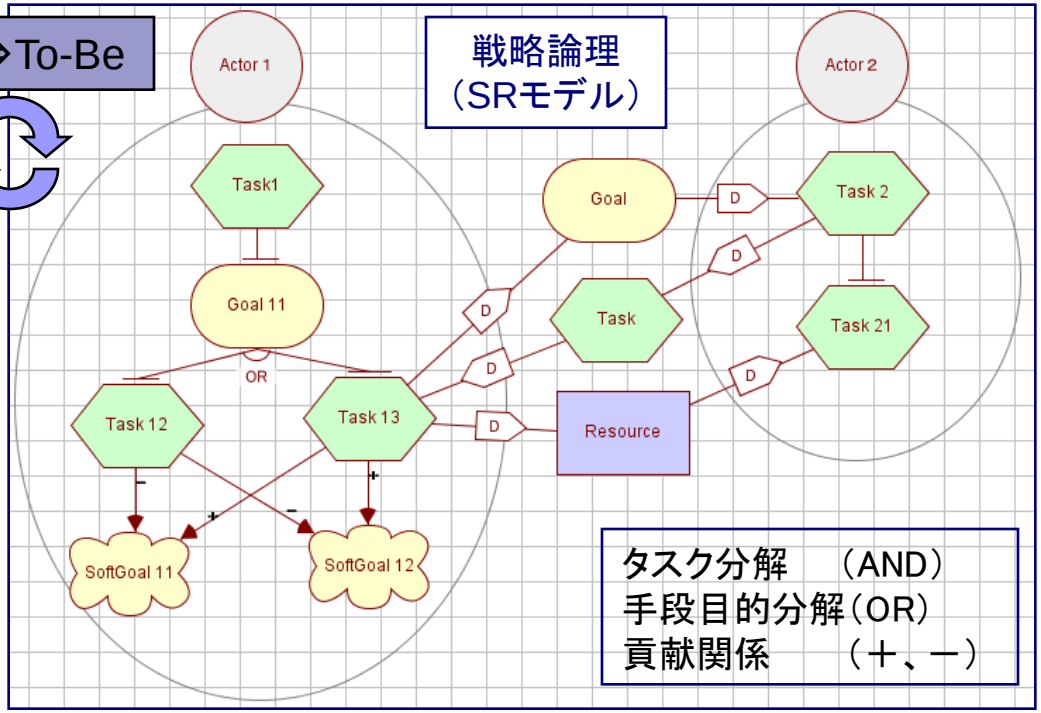
【参考】 Goal指向分析（i＊法）

ステークホルダ間の意図的ゴールを分析

Actor	ゴールの達成やタスクの実行に関心・責任を持つ実体（人、システム、役割）
Goal	達成されたか否かが明確なゴール（機能的要求）
SoftGoal	達成されたか否かが不明確なゴール（非機能的要求）
Task	ゴールを達成するための手順（ステークホルダの手続き、システムのプログラム）
Resource	ゴールの達成に利用される情報的、物理的実体（データ、メッセージ）

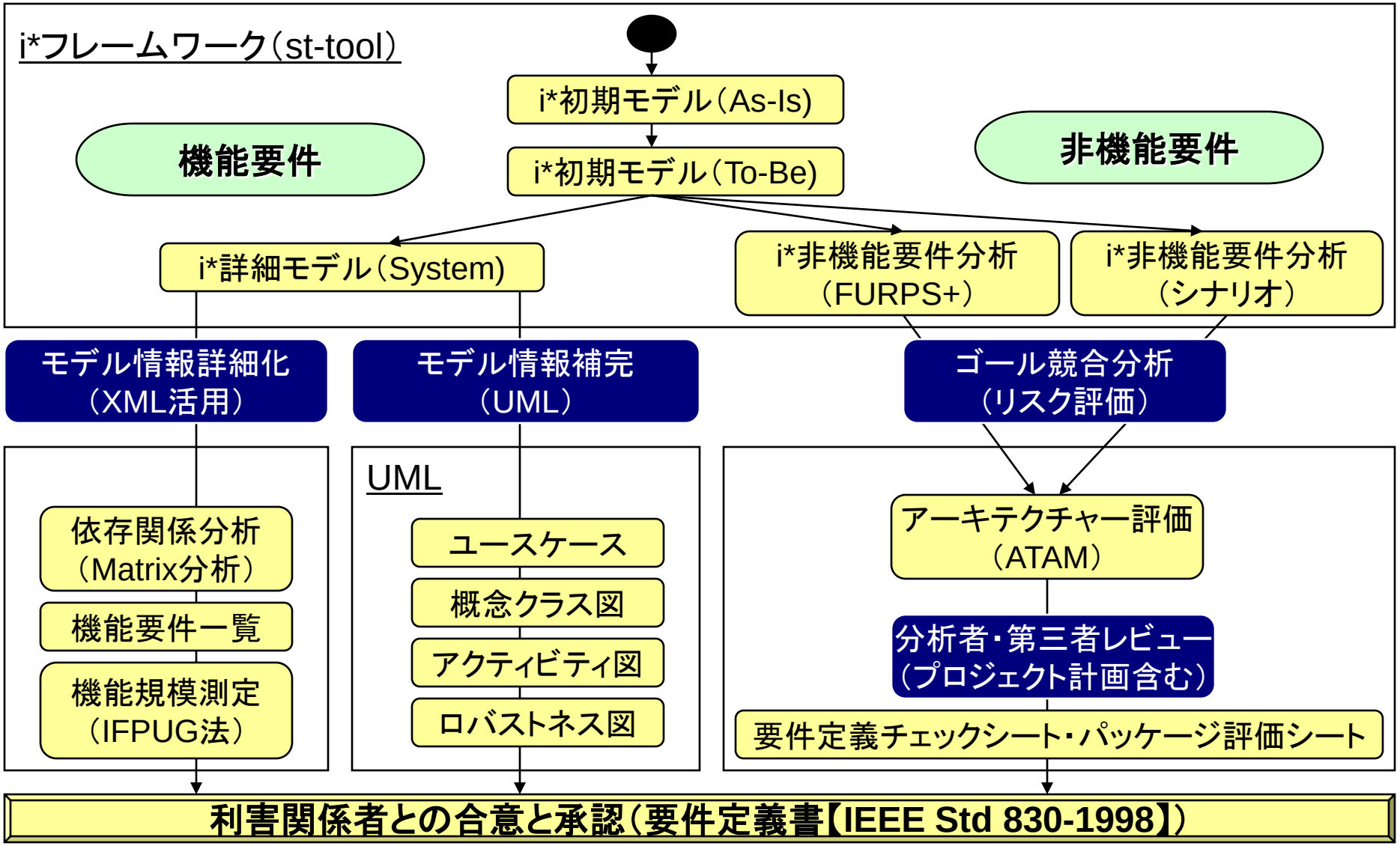


As-Is ⇒To-Be



SDモデル	アクタが何を期待しているか
SRモデル	アクタが何をどのように達成するか （サービス全体像の把握）

要件定義プロセス概略図

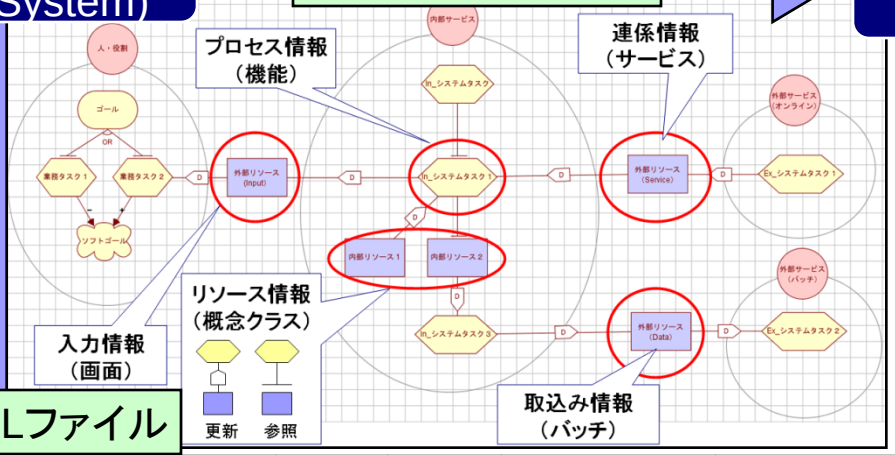


機能要件定義（モデル情報詳細化例）

i*詳細モデル（System）

関連付けルール

モデル情報補完（UML）



XMLファイル

i*法では表現できない部分

- ユースケース：代替系列
- アクティビティ図：実行順序（CRUD）
- 概念クラス図：相互関係、主要属性
- ロバストネス図：バウンダリ依存関係（画面遷移）

依存関係分析（Matrix分析）

DEPENDUM	DEPENDEE 人・役割	内部サービス	外部サービス（オンライン）	外部サービス（バッチ）	総計
外部リソース(Input)	1				
		1			
		1			
		1			
	1	3			
総計	1	3			

	内部サービス タスク	内部サービス リソース	
人・役割	業務タスク2	O	外部リソース(Input)
内部サービス	In_システムタスク1	I	外部リソース(Input)
		R	内部リソース2
		C	内部リソース1
		O	外部リソース(Service)
	In_システムタスク3	I	外部リソース(Data)
		C	内部リソース2
外部サービス(オンライン)	Ex_システムタスク1	I	外部リソース(Service)
外部サービス(バッチ)	Ex_システムタスク2	O	Ex_外部リソース(Data)

- アクタ間の依存関係明確化（ソフトゴール、ゴール、タスク、リソース）
- 内部サービス関係の詳細化（タスクーリソース関係）

ID	事前条件 外部リソース 登録データ	事前条件 外部リソース ・・・	事前条件 内部リソース 登録情報	事前条件 内部リソース マスタ情報	事前条件 内部リソース ・・・	システムタスク	チェック処理	トリガ	事後条件 外部リソース 登録データ	事後条件 外部リソース 検索データ	事後条件 外部リソース ・・・	事後条件 内部リソース 登録情報	事後条件 内部リソース マスタ情報	事後条件 内部リソース ・・・	制約	7 低 内部論理ファイル	10 中 内部論理ファイル	15 高 内部論理ファイル	5 低 外部インタフェースファイル	7 中 外部インタフェースファイル	10 高 外部インタフェースファイル	3 低 外部入力	4 中 外部入力	6 高 外部入力	4 低 外部出力	5 中 外部出力	7 高 外部出力	3 低 外部参照	4 中 外部参照	6 高 外部参照
1	I			N	C	登録処理	重複チェック					C	R			1						1						1		
2				E		検索処理	…					R																		
3		I		N	C	…	…			R		U																		

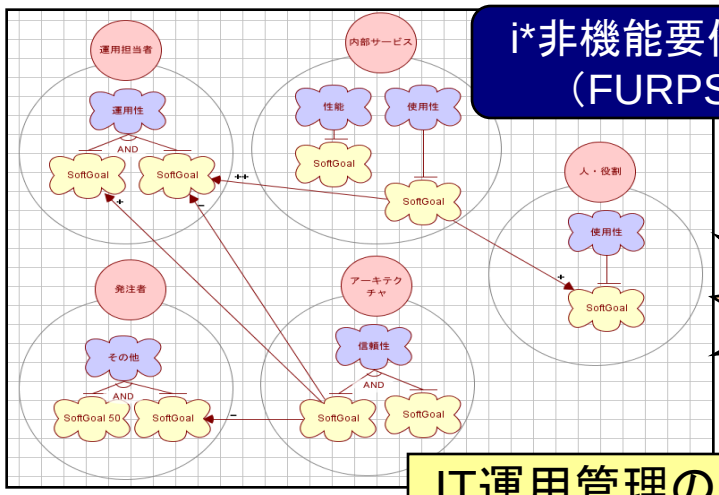
- システムタスク前後の状態明確化
  - 事前条件（内部・外部リソース）（入出力区分、存在条件）
  - 事後条件（内部・外部リソース）（入出力区分、CRUD表記）
  - UFP値の算出（IPFUG法への変換ルール）

機能要件一覧

変換ルール

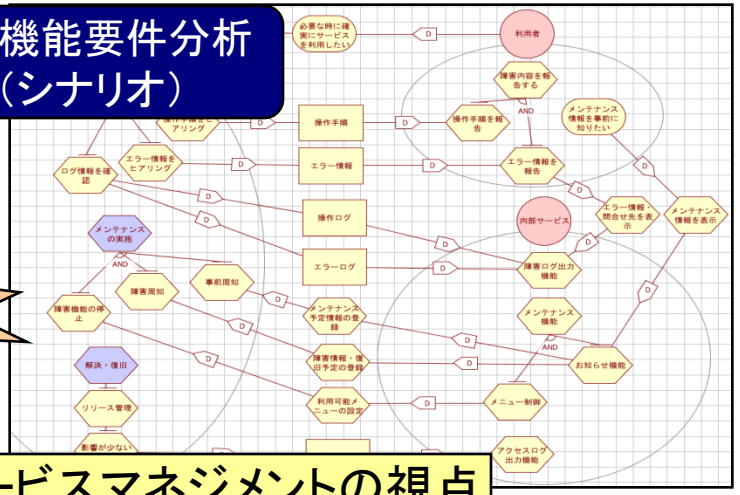
機能規模測定（IFPUG法）

非機能要件定義（ゴール競合分析）



i*非機能要件分析（FURPS+）

i*非機能要件分析（シナリオ）



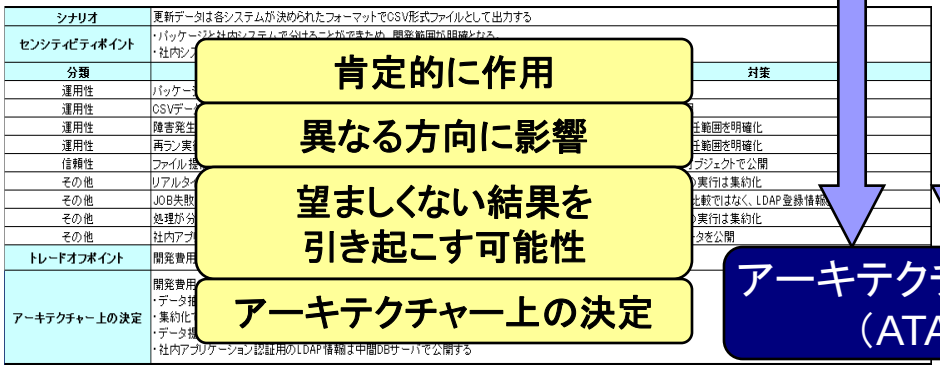
ゴール競合

IT運用管理の視点

- **FURPS+の分類で非機能要件をi*で分析**
  （使用性、信頼性、性能、運用性、その他）
- **非機能要件に関連の強いアクタを追加**
  （運用担当、発注者、AP基盤[Web、AP、DB]）

ITサービスマネジメントの視点

- **シナリオを顧客視点で作成しi*で分析**
  （サービス・サポートプロセスでの機能候補の抽出）
- **（例）サービスが利用できなくなる場合のシナリオ**
  （ログ制御、メニュー制御、お知らせなど）



肯定的に作用

異なる方向に影響

望ましくない結果を引き起こす可能性

アーキテクチャー上の決定

アーキテクチャー評価（ATAM）

- **ゴールが競合するような場合**
  ATAM（Architecture Trade-off Analysis Method）評価を用いてリスクを洗い出し、優先順位付けを行っていく

要件定義（非機能・制約・PJ計画・体制）

要件定義チェックシート（127項目）

プロジェクトの成否に影響を及ぼす
プロジェクト計画や**開発体制**もこの段階で確認

大分類	中分類	チェック数
機能（F)	機能要件管理	15
使用性（U）	UI ユーザ対応	2 1
信頼性（R)	信頼性 セキュリティ	4 6
性能（P)	性能	13
運用性（S)	管理 監視 保守 障害 復旧	13 9 11 4 4
その他（＋）	移行 設計制約	3 13
プロジェクト計画	契約調整 契約	21 8

属人性を排除（分析者・第三者レビュー）

パッケージ評価シート（144項目）

機能中心で選定することが多いパッケージについて、**設計や運用時の制約**などを確認

大分類	チェック数
製品全般	4
ライセンス方針	18
製品ライフサイクル	5
製品サポート体制	17
製品システム要件	16
バージョンアップ方針	10
稼動要件	12
アドオン	11
カスタマイズ	11
再利用コンポーネント	6
設計・構築・テスト	7
性能	4
運用	23

IEEE Std 830-1998との対比

章	目次	適用	対象成果物
1.1	目的	×	SRS固有の項目
1.2	適用範囲	○	i*要求分析（AS-IS・TO-BE）
1.3	用語・略称の定義	○	用語集
1.4	参考文献	×	SRS固有の項目
1.5	概要	×	SRS固有の項目
2.1	製品の全体像	○	i*要件分析（System）、依存関係分析、タスク-リソース分析
2.2	製品の機能	○	ユースケース図
2.3	ユーザ特性	○	i*非機能要件分析（FURPS+）
2.4	制約条件	○	i*非機能要件分析（シナリオ）、制約一覧
2.5	前提と依存関係	○	i*非機能要件分析（FURPS+）
3.1	外部インタフェース	○	i*要件分析（詳細化）、機能要件一覧
3.2	機能	○	ユースケースシナリオ、アクティビティ図、ロバストネス図
3.3	パフォーマンス要求	○	i*非機能要件分析（FURPS+）、非機能要件一覧
3.4	論理データベース要求	○	概念クラス図
3.5	設計上の制約	○	制約一覧、アーキテクチャ評価
3.6	ソフトウェアシステムの属性	○	i*非機能要件分析（FURPS+）
3.7	個別の要求の構成	○	i*非機能要件分析（シナリオ）

事例評価

事例1 （Webシステム・スクラッチ開発）

- 概算見積り
  - ☑ 開発時の見積り金額と比較した結果、算出工数とほぼ同規模に
- 適用範囲
  - ☒ Webシステムの場合でUIが重視されるような場合には別途UI定義書が必要

事例2 （バッチシステム・パッケージ利用）

- 要件定義プロセスの有効性
  - ☑ ベンダー作成の要求仕様書をi*で詳細化し、今回の要件定義プロセスを適用
    （非機能要件、パッケージ制約など、要件定義段階で明確にすべき30の不明点を指摘）
- 概算見積りの妥当性
  - ☑ 算出工数と実際のSIベンダー見積り工数で大きな開きがあったが、
    別ベンダーでの再見積りでは、算出工数とほぼ同規模に
- プロジェクト関係者での早期認識共有
  - ☑ 担当者への特別な教育は不要
- 仕様変更プロセス管理
  - ☑ 機能要件一覧から、タスク⇔リソースの追跡から影響範囲を特定
  - ☑ ドキュメントのトレーサビリティを確保

今後の展開

展開・再利用

- 実プロジェクトでの活用
  - 継続的な見直しによるプロセス改善
  - 特に、大規模開発、再構築案件での適用
- 要件管理プロセスでの有効活用
  - 要件管理リポジトリなどによる再利用化（構成管理など）
  - 非機能要件、制約事項のテンプレート化

最適化

- アプリケーションのシンプル化
  - 既存のサービス整理
    - データのシンプル化（正規化）
    - 業務アプリケーションの統合
    - 付加価値を生まない業務プロセスを削減し付加価値を生むプロセスに集中（BPR）
- アーキテクチャ分析の全体最適化
  - ITインフラ全体のバランスを考慮した設計
  - 汎用的に活用可能なインフラ設計（運用段階においてAPと環境を分離）

以　上