

要求工学の現状と 要求工学知識体系(REBOK)第1版の紹介



青山 幹雄

南山大学 情報理工学部 ソフトウェア工学科

miko.aoyama@nifty.com

www.nise.org

2011年 9月13日

- 👉 要求工学の現状
- 👉 要求知識体系(REBOK)
- 👉 要求工学の新潮流
- 👉 今後の方向



👉 要求工学(RE: Requirements Engineering)とは

要求工学(RE: Requirements Engineering)とは

A coordinated set of activities for *exploring, evaluating, documenting, consolidating, revising and adapting the objectives, capabilities, qualities, constraints and assumptions* that **the system-to-be** should meet based on **problems** raised by the system-as-is and **opportunities** provided by new technologies.

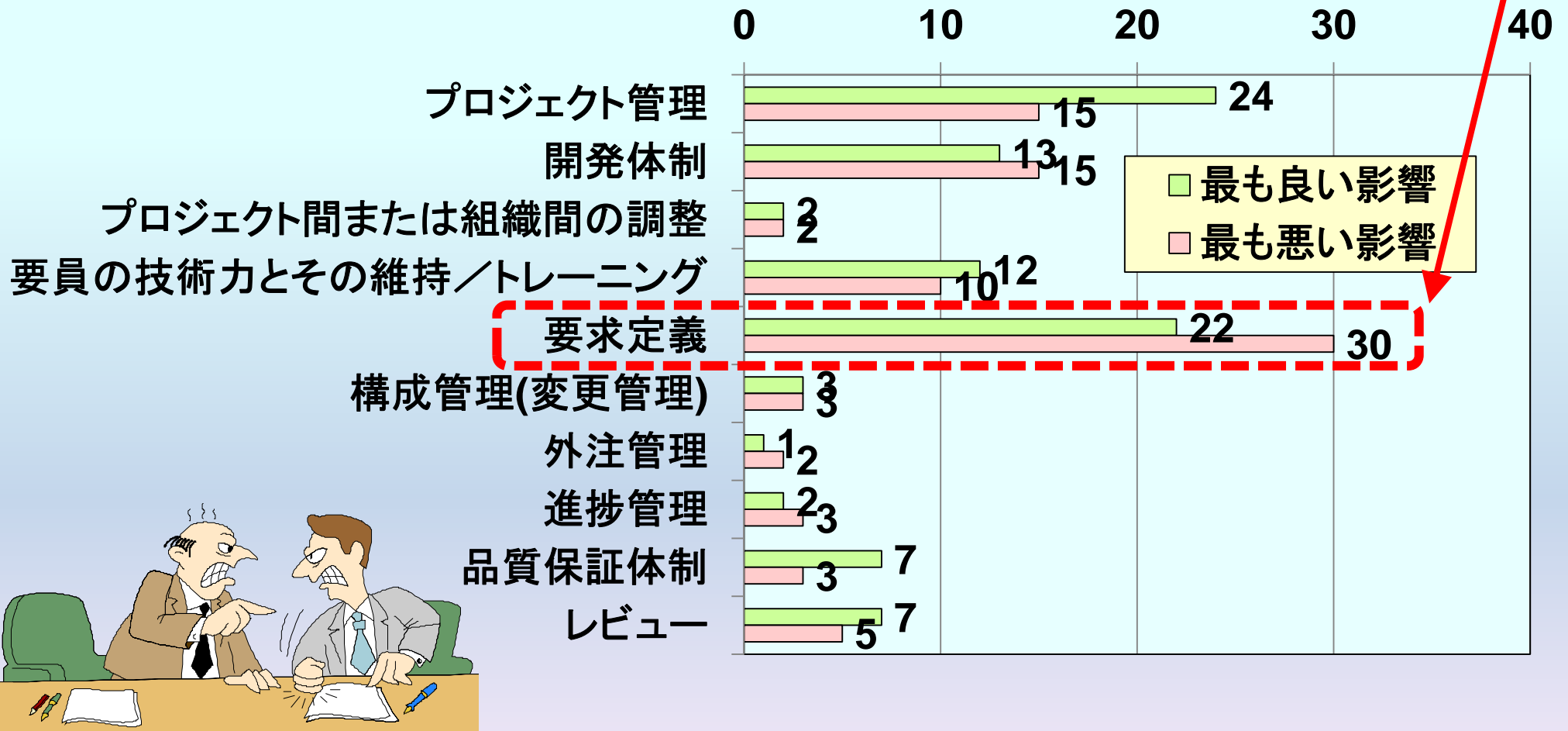
出典: A. von Lamsweerde, Requirements Engineering. John Wiley & Sons, 2009.

👉 要求工学国際会議(IEE International Requirements Engineering Conference)

- 👉 1993年: 最初の要求工学国際会議の開催, 2010年は18回目
- 👉 2004年: 日本(京都)で第12回を開催(<http://www.re04.org/>)

要求工学の現状 要求定義が成功の鍵!

👉 要求定義が品質へ最大の影響: 品質向上と低下の両面



出典: 平成17年度情報サービス産業におけるソフトウェア開発の実態アンケート調査結果(概要)

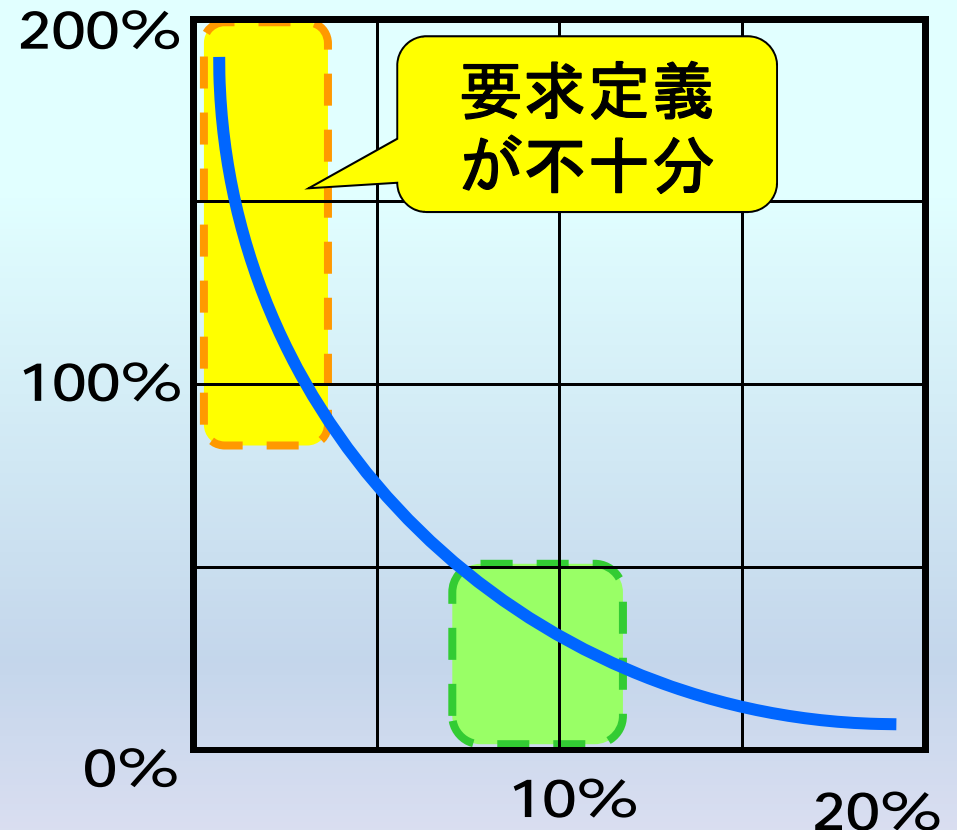
👉 要求定義の投資効果: NASAの統計データ*

👉 全開発コストの中で
要求定義に割くコストの効果

👉 2-3%のプロジェクト ⇒
開発コスト超過=80-
200%

👉 8-12%のプロジェクト ⇒
開発コスト超過=0-50%

コスト超過比率



要求定義のコスト配分比率

*参考文献: B. B. Roberts, et al., The Benefit of Integrated, Quantitative Risk Management, INCOSE, 2001.

要求工学の現状 要求工学の「成熟化」

- 👉 要求工学研究者による大著(700~800ページ超)が続々刊行
- 👉 国内でも40冊以上刊行



要求工学の現状

要求工学に関連する知識体系の出現と課題

👉 要求定義に関する様々な知識体系(BOK)/標準と認定制度

👉 海外: SWEBOK, BABOK(Business Analysis BOK), IREBのシラバス

👉 国内: ITSS[ITスキル標準]ほか

👉 要求工学を基礎から専門までカバーする知識体系がない

BOK	SWEBOK(Software Engineering BOK)	BABOK(Business Analysis BOK)	IREB(Int'l RE Board)	ITSS(IT Skill Standard)
最新版	2004	V2(2009)	V2(2009)	V3(2008)
組織	IEEE CS(米国)	IIBA(カナダ)	IREB(ドイツ)	経済産業省/IPA
対象者(人材)	ソフトウェア開発専門家	ビジネスアナリスト(BA)	要求エンジニア(RE)	コンサルタント, ITアーキテクト
内容	ソフトウェア工学の中のソフトウェア要求工学:基礎知識	ビジネス分析:専門的手順	要求工学:基礎知識	情報技術とスキル
認定	CSDP*, CSDA	CBAP*, CCBA	CPRE	情報処理技術者
認定レベル	*学卒で4年(7,000時間)以上のソフトウェア開発実務経験[大学で専攻の場合2年]	*高度専門家: 7,500時間以上のビジネス分析実務経験	初心者(Foundation Level)	中~高度技術者(スキルレベルを設定)

要求工学の現状 わが国のソフトウェア開発競争力の源泉

- 👉 オフショア開発, サービス化(クラウドコンピューティング)
 - 👆 国内は上流工程中心へ
- 👉 Capacity(量)からCapability(能力=競争力)へ
 - 👆 要求と技術の高度化に即した人材とその育成の必要性

👋 要求定義, アーキテクチャ設計

能力・競争力(Capability)



「要求」はソフトウェア工学の最後のフロンティア

👉 「要求」の難しさ

- 👉 要求のもつ本質的な課題とその獲得, 仕様化の難しさ

👉 「要求」の本質的な課題: 要求が内在する不安定性

- 👉 空間的不安定性: システム(要求)の境界の曖昧性
- 👉 時間的不安定性: ユーザ要求や外部環境の時間的変化
- 👉 社会的不安定性: 政治的, 人的, 組織的不安定性の反映, ユーザ/ベンダ間の力学, ベンダ間の過当競争

👉 現場における「要求工学」への取組みの遅れ

- 👉 個人のスキルやノウハウに依存
- 👉 「要求工学」への理解の遅れ
- 👉 適切なガイドラインがない

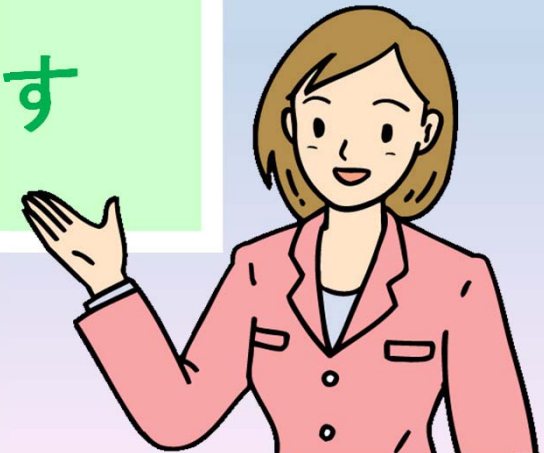


Twin Angels
REBOKカバーイラスト
REBOKがユーザとベンダの
協調の輪となる願いを込めて

2.

要求工学知識体系(REBOK)

REBOK編成に至る経緯と
REBOK全体像をご紹介します



要求工学知識体系(REBOK) 要求工学を学び, 適用するガイドがない

- 👉 要求工学とは何か?
- 👉 要求とは何か?
- 👉 要求定義ができるためにはどの技術を学ぶ必要があるか?
- 👉 要求定義とは何をすべきか?
- 👉 要求工学のどの技術はどんな課題に役立つか, あるいは, 役立たないか?
- 👉 要求工学をどのように適用すべきか?
- 👉 要求定義の人材育成はどうすべきか?
- 👉 関連する知識体系の位置づけは?
- 👉 どのような書籍を読むべき?



要求工学知識体系(REBOK) REBOKに至るJISA要求工学WGの活動

👉 JISA 要求工学調査検討WGの活動

[2006～2008年度, のべ100名以上が参画]

- 👉 2006年度: 要求開発の組織的な取組み: 要求開発コーディネータ組織の提案と事例収集
- 👉 2007年度: 要求工学ベストプラクティスの収集と整理
- 👉 2008年度: REBOKの検討とユーザにおける事例収集
 - 👉 要求工学の実践的な知識の習得と人材育成の仕組み作り

👉 JISA REBOK 企画WGの設立[2009年度~]

- 👉 現場の視点から要求工学のグローバルな知識ベースの活用
- 👉 JISA要求工学シンポジウムからのフィードバック
 - 👉 2007~2009年開催: 100名以上の参加者
 - 👉 2009年10月2日 第3回シンポジウム[REBOKの検討結果の議論]
- 👉 学会・国際会議での発表と討議
 - 👉 グローバルな要求工学コミュニティとの連携

- 👉 (1) ベンダ, ユーザが共通に利用できる知識体系
- 👉 (2) 要求アナリストに加えて, エンドユーザ, 経営者など, 要求開発に関与するアクタ/ステークホルダが, 必要に応じて習得すべき範囲と水準を整理した知識体系
- 👉 (3) ビジネス要求, システム要求, ソフトウェア要求の3つのスコープに応じた知識体系
- 👉 (4) エンタープライズシステム, 組込みシステムの要求開発に共通する技術の知識体系. ただし, ドメイン固有の知識は個別に定義されるものとする
- 👉 (5) グローバルに広く利用できる形態とする

👉 要求工学に関与する全てのアクタが一定の理解を持つこと

👆 要求アナリスト: ユーザの要求開発者, ベンダの情報システム部門

👆 要求の利用者: エンドユーザ, PM

👆 要求工学の理解者: 経営者, CIOなど

- ・ステークホルダ: 要求の関与者
- ・アクタ: 要求工学の関与者



要求工学知識体系(REBOK) 要求のステークホルダと要求工学のアクタ

👉 要求と要求工学プロセスに参画する人材の役割

	ユーザ			
	経営者	システム責任者	PM	エンドユーザ
要求の計画と管理	○	○	○	
要求獲得		○		○
要求分析		○		○
要求仕様記述		○		
要求のV&V・評価	○	○		○

	ベンダ			
	経営者	上級SE	PM	ソフトウェア開発者
要求の計画と管理	○	○	○	
要求獲得		○		
要求分析		○		
要求仕様記述		○		○
要求のV&V・評価		○		○

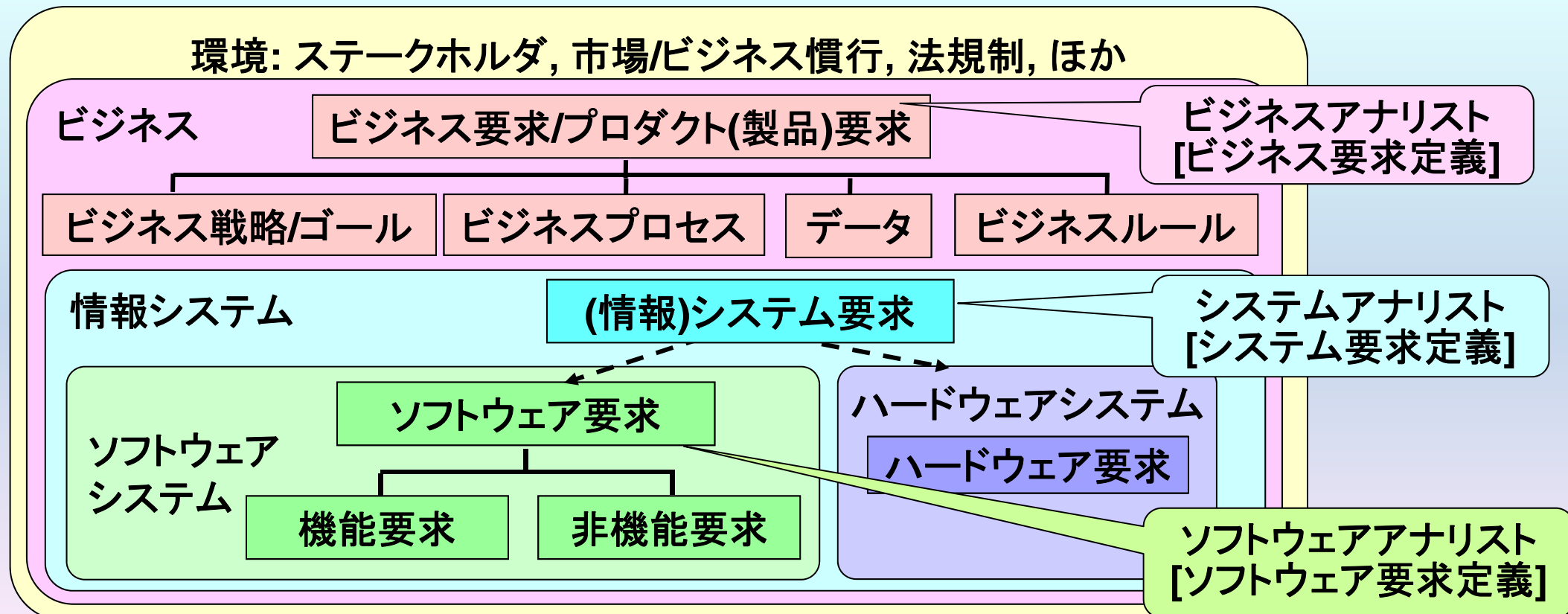
要求工学知識体系(REBOK) 要求の範囲と要求アナリストの役割

👉 要求の3つの範囲

👆 ビジネス/製品, (情報)システム, ソフトウェア

👉 要求アナリストの役割

👆 ビジネスアナリスト(BA), システムアナリスト, ソフトウェアアナリスト



要求工学知識体系(REBOK) 関連知識体系との相互運用性の確保

👉 国内とグローバル標準との対応付けを可能とする

👉 国内: 共通フレーム2007

👉 グローバル標準: BABOK, SWEBOK

👉 ソリューションへの橋渡しを可能とする

👉 ソリューション ⇒ システム要求, ソフトウェア要求

要求のスコープ		REBOK	BABOK	共通フレーム2007
ビジネス		ビジネス要求 プロダクト要求 [組込み]	ビジネス要求	(システム化構想, システム化計画)
ステークホルダ		(ステークホルダ要求)	ステークホルダ要求	利害関係者要件
ソリューション	システム	システム要求	ソリューション要求	システム要件
	ソフトウェア	ソフトウェア要求		ソフトウェア要件
移行		移行要求	移行要求	(移行計画)
運用		運用要求	—	(システム運用)

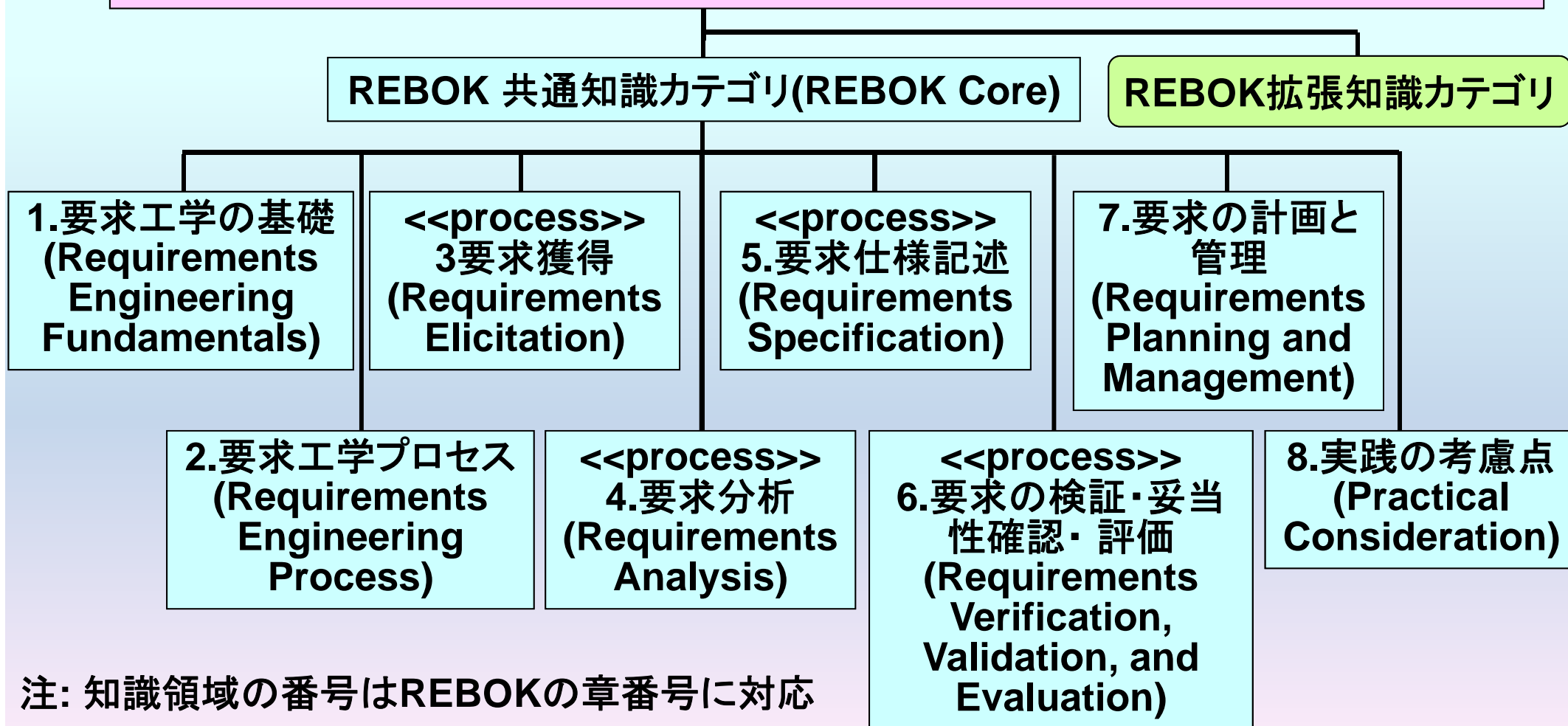
参考文献: IPA/SEC, 共通フレーム2007, 第2版, オーム社, 2009.

要求工学知識体系(REBOK) REBOK 共通知識カテゴリ: 主要8知識領域

👉 SWEBOKを基礎に, 要求工学の内容を強化した知識体系化

👆 知識カテゴリ: 共通のコア知識とドメインとの接点となる拡張とに分離

要求工学知識体系 REBOK(Requirements Engineering Body of Knowledge)



注: 知識領域の番号はREBOKの章番号に対応

👉 技術知識

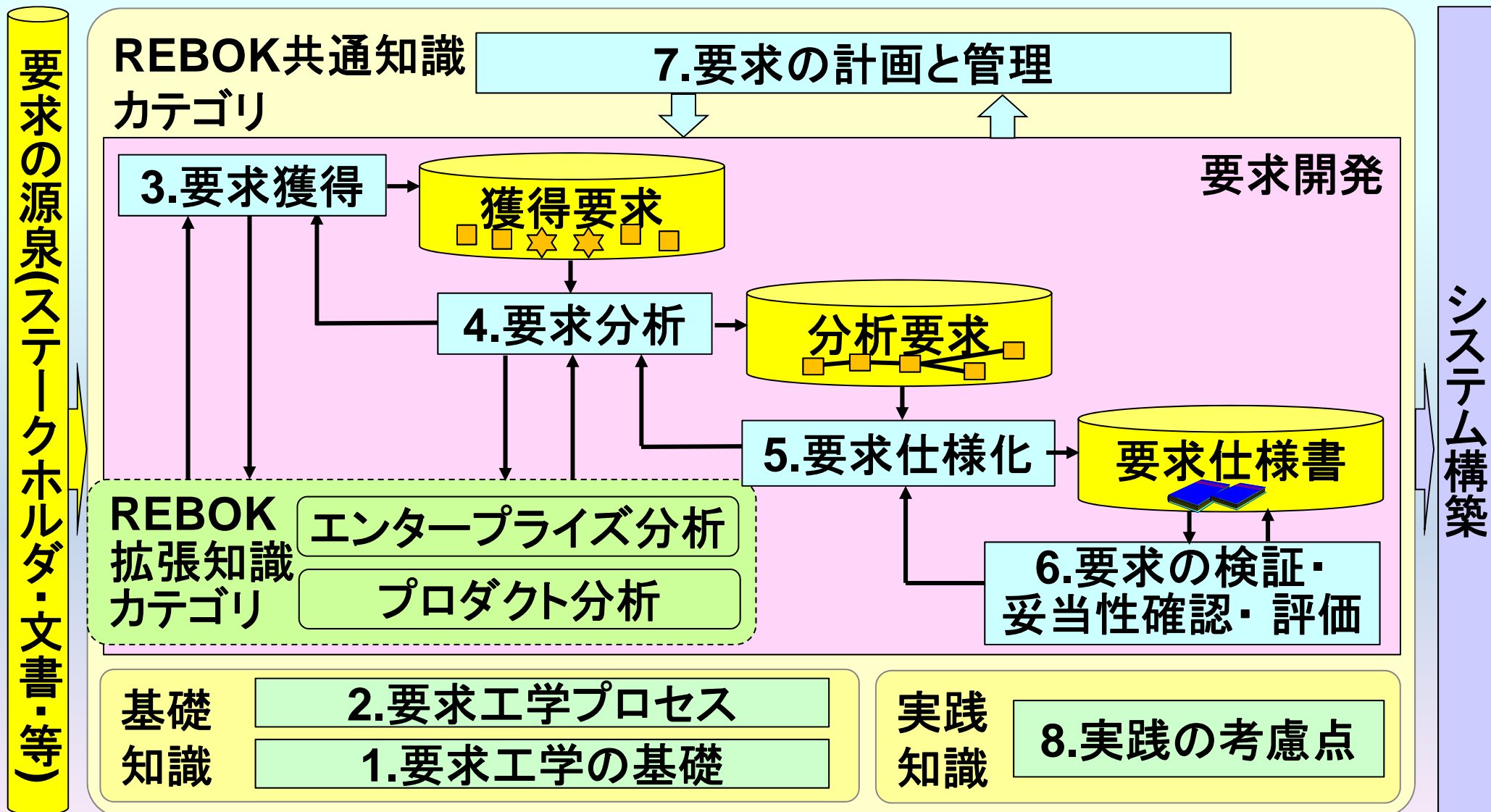
👉 要求工学の基礎, 要求工学プロセス, 要求の計画と管理, 実践の考慮点

👉 プロセス知識

👉 要求獲得, 要求分析, 要求仕様化, 要求の検証と妥当性の確認, 評価

知識領域	内容
要求工学の基礎	要求とそのスコープや性質などの基礎的事項. 機能/非機能要求も含む
要求工学プロセス	要求定義と管理のプロセスと主要なアクティビティなどに関する知識
要求獲得	顧客を含むステークホルダを明らかにし, 会議やインタビューなどを通して要求を引出す技術に関する知識
要求分析	要求項目を整理し, その間の関係づけ, 優先順位づけなどを行い, 実現すべき要求を明らかにして絞り込みに関する知識
要求仕様化	分析された要求を規定の書式や表記法で記述する技術に関する知識
要求の検証・妥当性の確認・評価	要求間の矛盾がないことや, 必要な顧客の要求項目を満たしていることの確認, あるいは, その達成の度合いを評価する技術などに関する知識
要求の計画と管理	要求管理を計画し, 遂行や成果物を管理する技術に関する知識
実践の考慮点	要求工学を実践する上で知っておくべき知識やベストプラクティス

👉 要求工学の反復的プロセス



要求工学知識体系(REBOK)

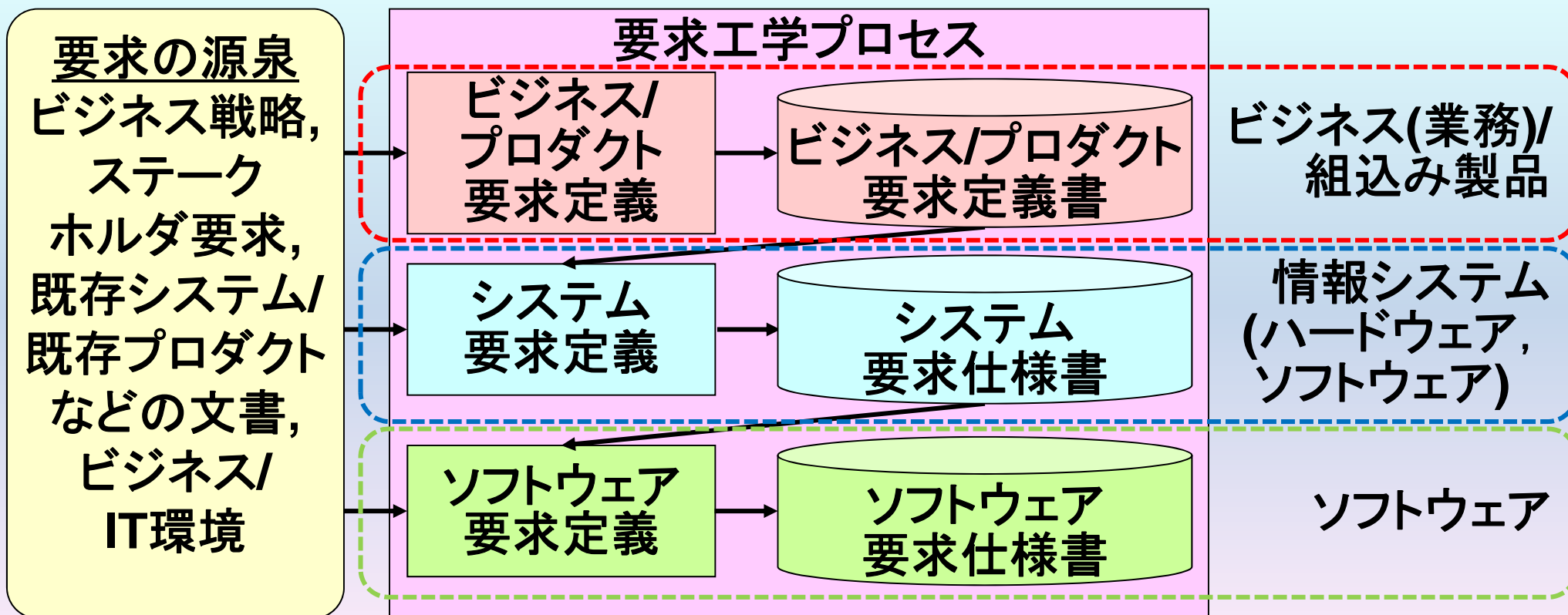
[2] 要求工学プロセス: 要求スコープと要求定義

👉 段階的要求定義: 要求の3段階スコープに応じた要求定義

👉 ビジネス/プロダクト, システム, ソフトウェア

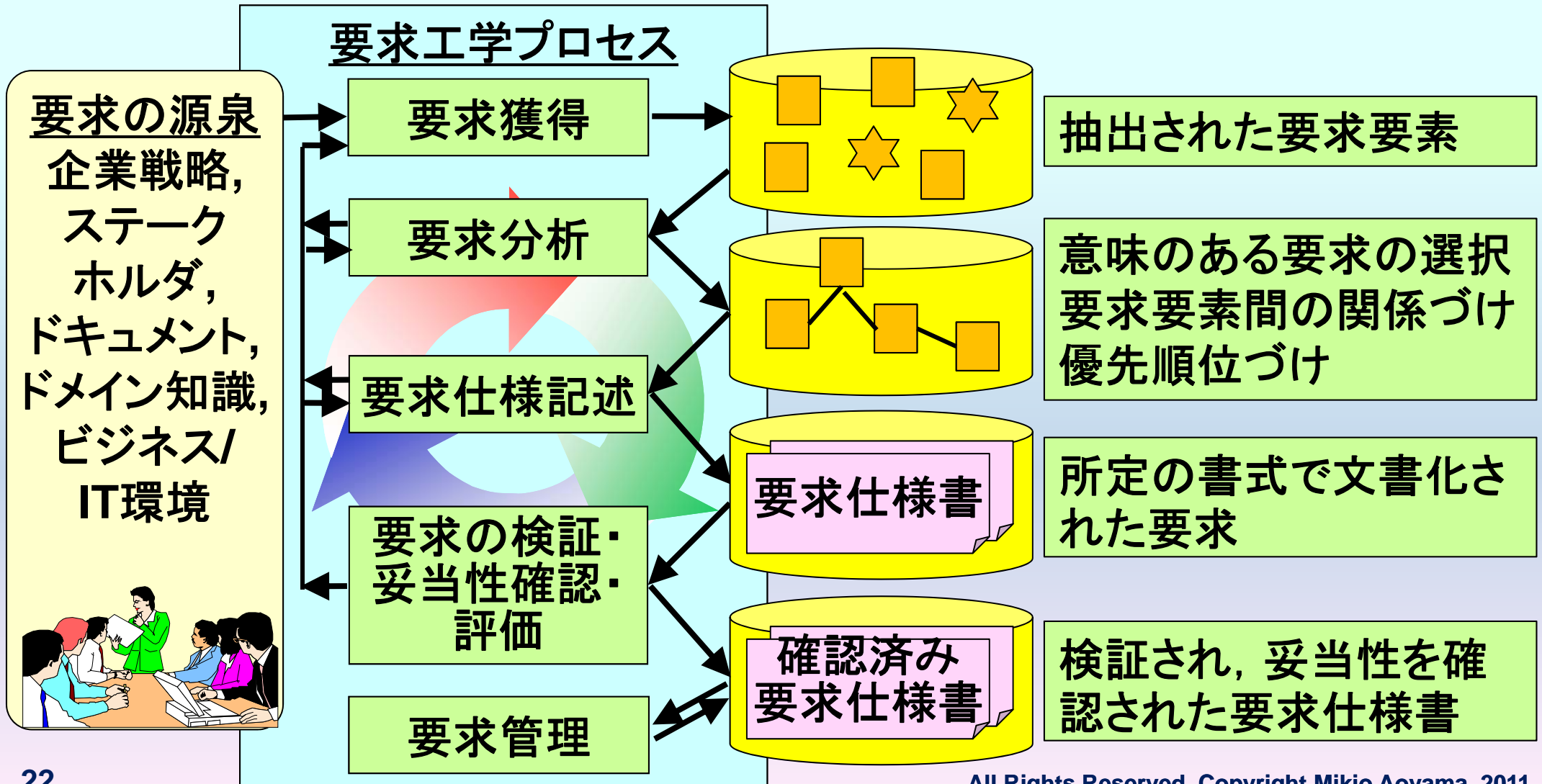
👉 反復: 各スコープ毎に要求定義の反復

👉 要求獲得, 要求分析, 要求仕様化, 要求の検証・妥当性確認・評価



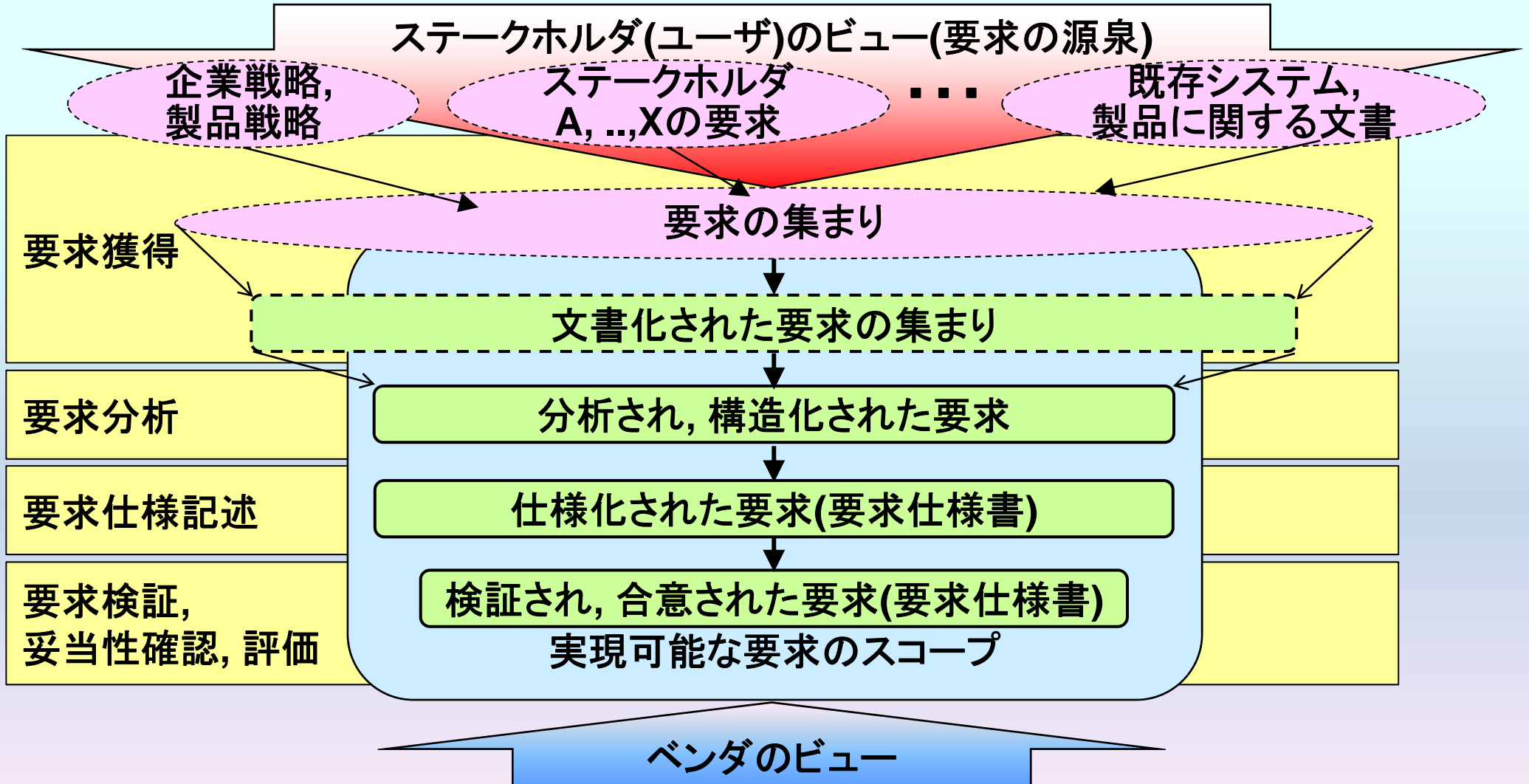
要求工学知識体系(REBOK) [2] 要求工学プロセス: 要求工学プロセス

👉 スコープ毎の要求定義の段階的な4つのプロセスと成果物



要求工学知識体系(REBOK) [2] 要求工学プロセス: 要求の形成

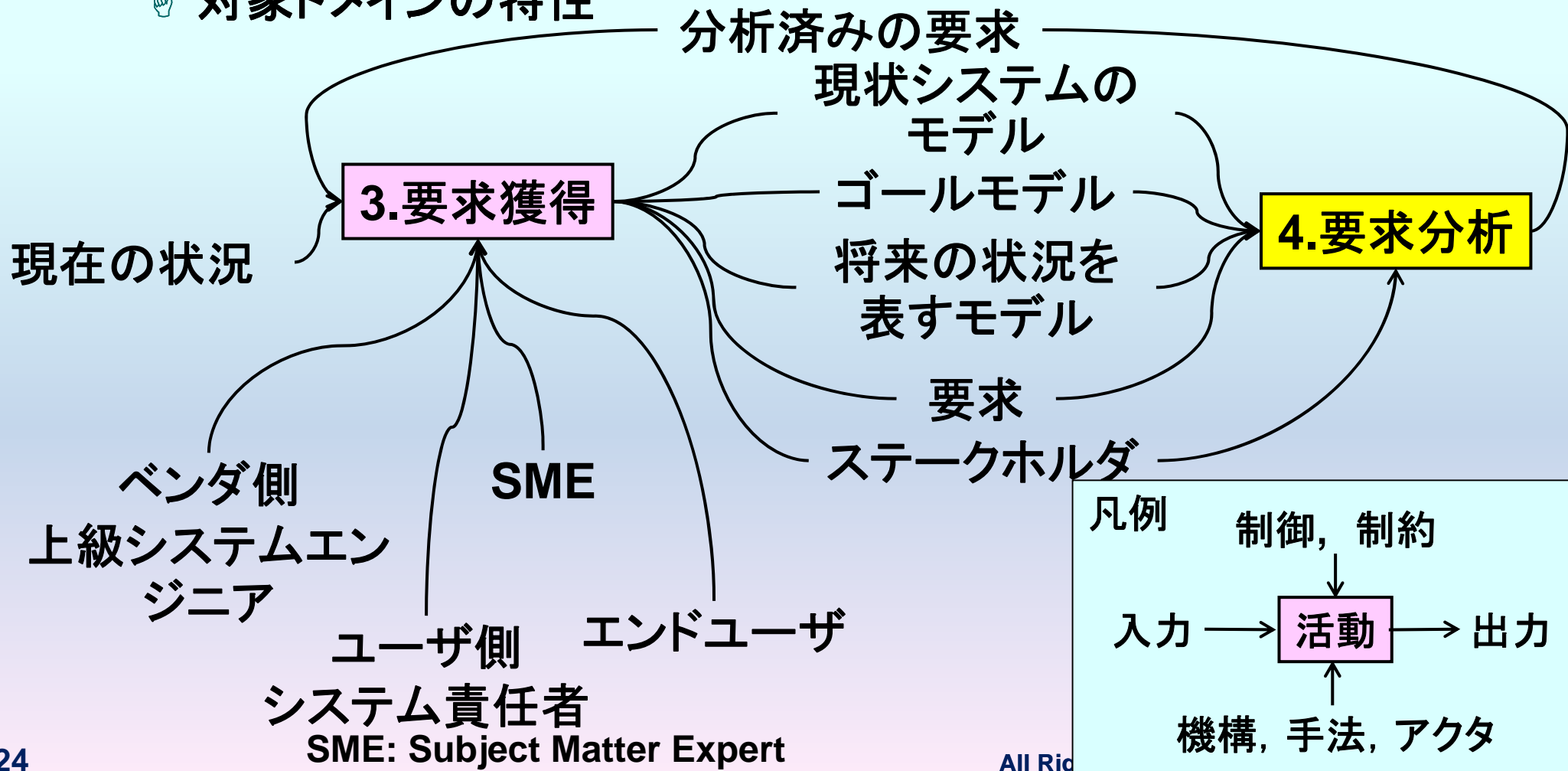
👉 要求定義: 要求の獲得からの段階的絞り込みと整合



👉 要求獲得: 要求工学の入口として技術の進化と深化

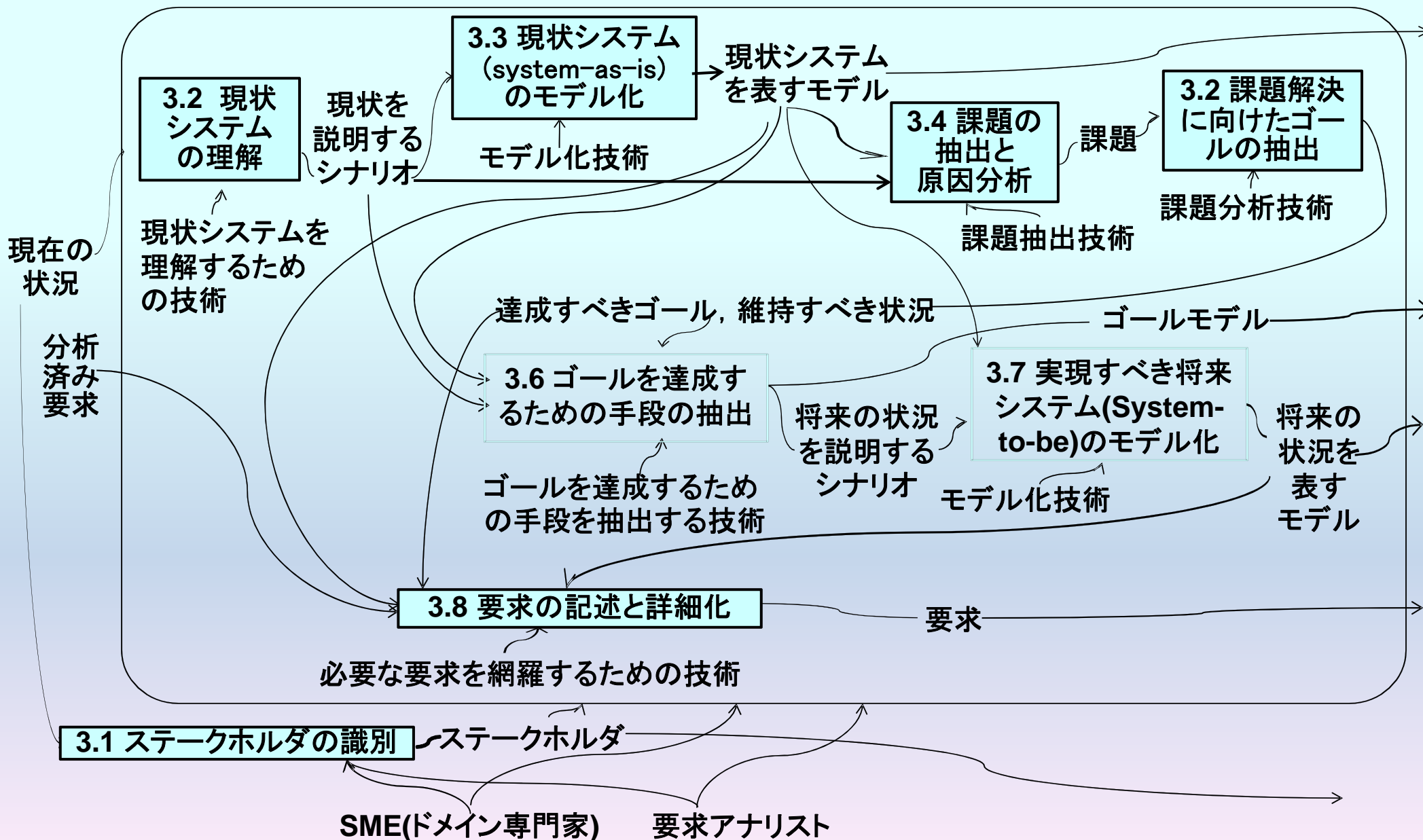
👉 ステークホルダ, ゴールの重要性

👉 対象ドメインの特性



要求工学知識体系(REBOK)

[3] 要求獲得: [3.0] 概説: 要求獲得プロセス(詳細)



👉 ステークホルダ(Stakeholder(s))[利害関係者]

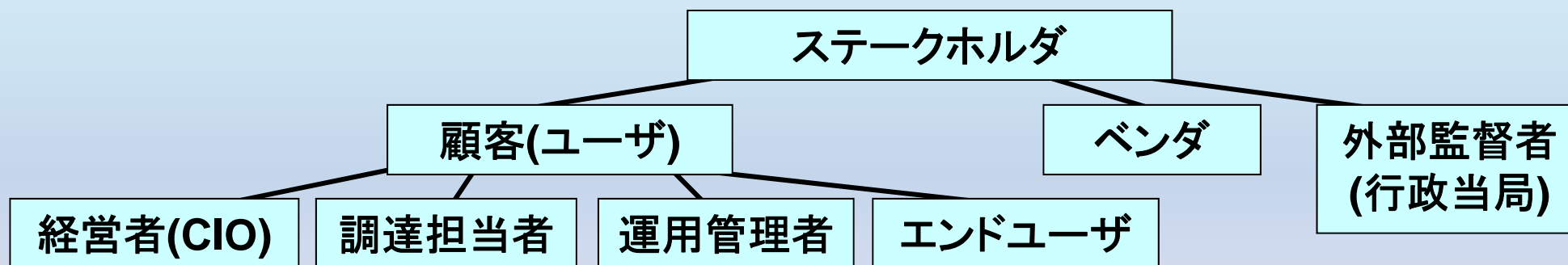
- 👉 システムに関与する個人, グループ, 組織
- 👉 システムに対する要求の源泉

ステークホルダの概念: 1930年代に企業経営において提起*

👉 ステークホルダ分析(Stakeholder(s) Analysis)

- 👉 ステークホルダとその間の関係を発見し, 重要性和リスクを評価

👉 ステークホルダのシステムへの関与とその影響を明確化



参考文献: A. Rotem-Gal-Oz, From Stakeholders to Models: It Is All a Matter of Viewpoints, Apr. 2007, http://msdn2.microsoft.com/en-us/library/bb447667.aspx#04_03_views_topic1.

*M. B. E. Clarkson, Stakeholder Framework for Analyzing and Evaluating Corporate Social Performance, Academy of Management Review, Vol. 20, No. 1, 1995, pp. 92-117.

👉 ステークホルダマトリクス: 影響度と重要度をマトリクスで表現

👉 影響度(Influence): 意思決定に及ぼす相対的力

👉 影響度の分類例

👉 主要顧客(Primary Customers): 重視すべき顧客

👉 一般顧客(Secondary Customer): その他の顧客

👉 法規などによる分類例

👉 順法者(Complier): その行為が法規に準拠する必要がある顧客

&例: 病院システムにおける医者などの医療従事者

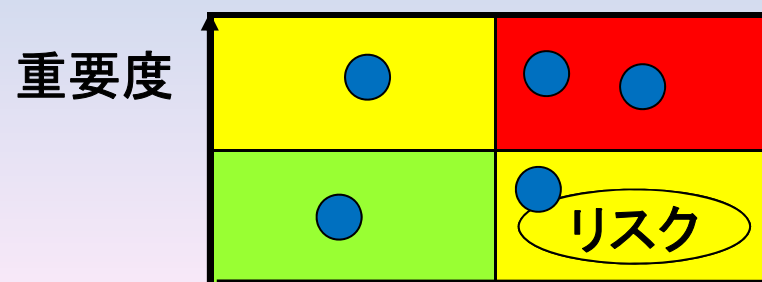
👉 重要度(Importance): 開発が成功するための必要性

👉 必須(Mandatory)


👉 望ましい(Desirable)


👉 あれば良い(Nice to Have)

ステークホルダマトリクス



ユーザ理解の方法

 ユーザのモデル化(User Modeling)

 ペルソナ(Persona)

 ユーザ情報の獲得方法(Collecting User Information)

 アンケート: ユーザプロファイリング(Profiling)

 観察(Observation)

 ライフログ(Life Log)

 ユーザ情報の分析方法(Analysis Methods of User Information)




 コンジョイント分析(Conjoint Analysis)

 データマイニング(Data Mining)

 機械学習(Machine Learning)

 協調フィルタリング(Collaborative Filtering)

 ベイジアンネットワーク(Baysian Network)

 現状システム理解の方法 具体例からのアプローチ シナリオ(Scenario), ユースケース(Use Case), ユーザストーリー(User Story) エスノメソドロジー(Ethnomethodology)/エスノグラフィー(Ethnography) 全体の枠組み(モデル化)からのアプローチ 概念モデリング(Conceptual Modeling)[ドメインモデリング] Zachmanフレームワーク 特定ドメインにおけるアプローチ エルゴノミクス(Ergonomics)[人間工学]

👉 シナリオ(Scenario)

👉 シナリオは使用に関する具体的なストーリー (A scenario is a concrete story about use)[Carroll 2000]

👉 注:ストーリーは使用に限定されない

👉 ストーリーは図, ビデオなどを含む

👉 シナリオの目的

👉 ユーザによるシステムの使用を具体的に理解する

👉 シナリオの表現

👉 (工学的な)目的をもって一定の規則に従って記述されたストーリー

👉 ユーザの言葉(自然言語)で, ユーザの操作を記述

👉 ユースケースのシナリオなど

参考文献:

J. M. Carroll (ed.), *Scenario-Based Design*, John Wiley & Sons, 1995.

J. M. Carroll, *Making Use: Scenario-Based Design of Human-Computer*

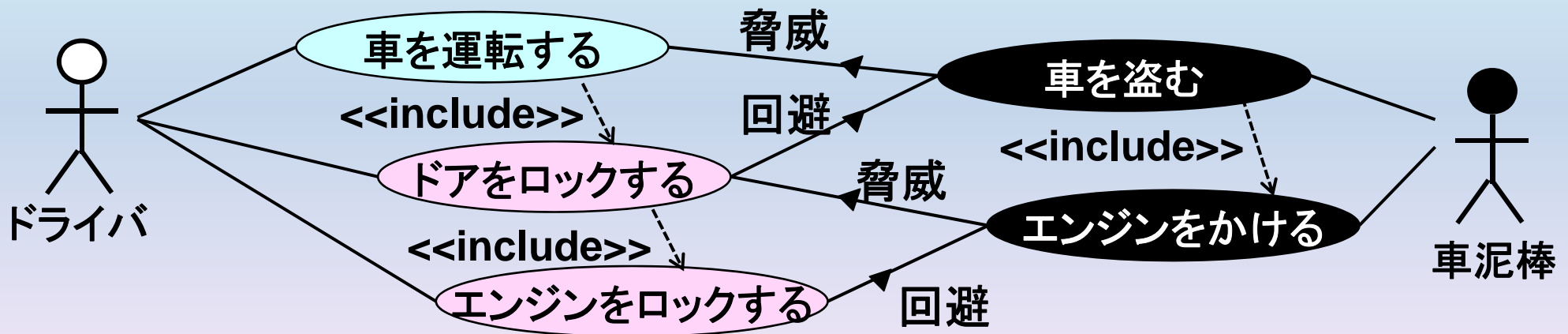
Interactions, MIT Press, 2000 [郷 健太郎 (訳), シナリオに基づく設計, 共立出版, 2003].

☞ ミスユースケース(Misuse Case)

- ☞ システムに危害を加えるユースケース(use case from the viewpoint of an actor hostile to the system)
- ☞ 悪意のアクタ, 誤った操作を行うアクタ

☞ ミスユースケースの利用

- ☞ セキュリティ要求, 安全性要求の記述
- ☞ ユースケースに対する脅威(threaten)とその回避(mitigate)の表示

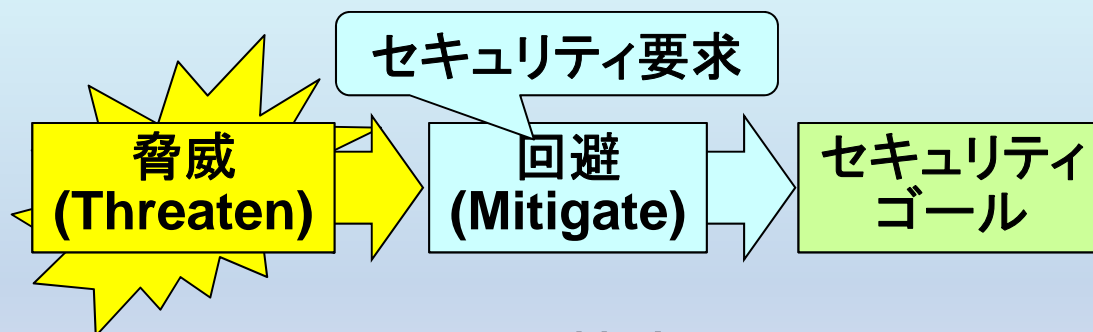


👉 セキュリティ/安全性要求の定義

- 👉 一般的な定義: 悪いことが起こらないこと (Not happening something wrong/harmful) ⇒ 要求としての定義と検証 (満足することの立証)が困難
- 👉 転換した要求定義: 脅威からシステムを守ること(Protect assets from harm)

👉 セキュリティ要求の獲得

- 👉 機能要求の特定
- 👉 セキュリティゴールの特定
- 👉 セキュリティゴールを満たすセキュリティ要求の特定
- 👉 セキュリティ要求がゴールを満たすことの確認



[3] 要求獲得: [3.5] 課題解決に向けたゴールの抽出: ゴールとは

👉 ゴール(Goal)[目標(Objective), 目的(Purpose)]

👉 システムのあるべき姿: ゴールは状態や振舞いとして定義

👉 システム: ビジネスシステム, 情報システム, ソフトウェアシステム

👉 ゴールの例: 常時顧客の欲しい品を提供できる

👉 理由(Rationale)[なぜ(Why)]: なぜ, その要求が必要か?

👉 理由の例: 常時顧客が欲しいから, 顧客が欲しい商品をタイムリーに提供したいから

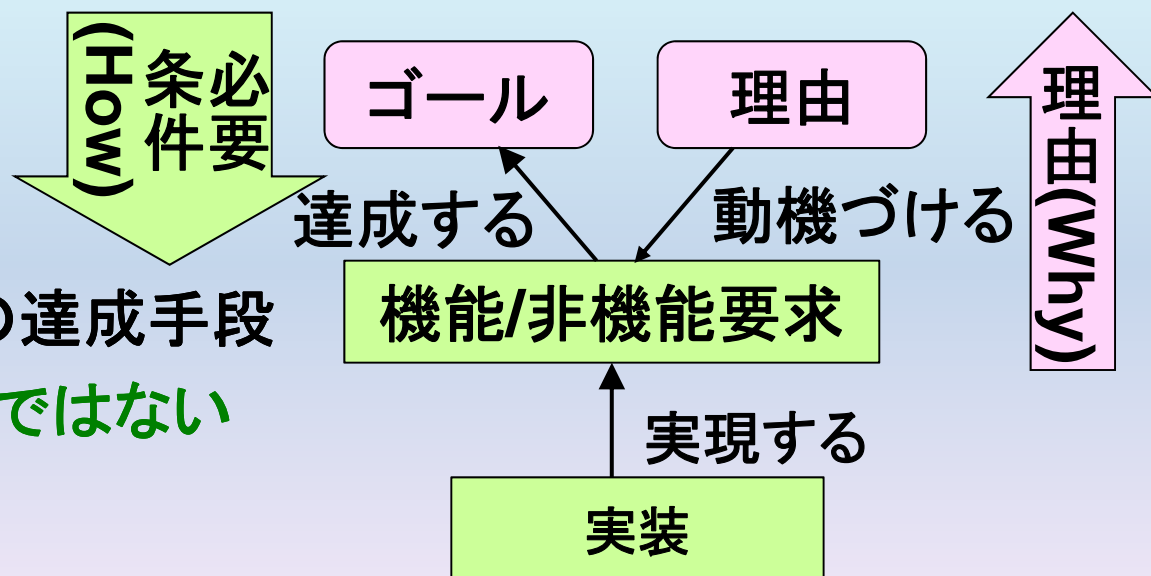
👉 ゴールの意義

👉 課題解決とはゴールの達成

👉 機能/非機能要求はゴールの達成手段

👉 機能/非機能要求は始点ではない



👉 まず, ゴールを定義する



参考文献:

A. van Lamsweerde, Goal-Oriented Requirements Engineering, Proc. RE '01, pp. 249-262.

A. van Lamsweerde, Requirements Engineering, John Wiley & Sons, 2009.

 **ゴール指向要求工学(GORE: Goal-Oriented RE)** **KAOS(Knowledge Acquisition in autOmated Specification)
[A. van Lamsweerde, et al., 1991]** **木構造によるゴールの形式的モデルの提案** **NFR(Non-Functional Requirements) Framework [L. Chung,
et al., 2000]** **非機能要求(NFR)の概念とソフトゴールの提案** **i*(eye star)/URN(User Requirements Notation) [E. Yu, 1995]** **ゴールのネットワーク構造(SDモデル, SRモデル)の提案****参考文献:  活発な研究対象**

A. Dardenne, S. Fickas, A. van Lamsweerde, Goal-Directed Concept Acquisition in Requirements Elicitation, Proc. IEEE IWSSD 91, Oct. 1991, pp. 14-21.

L. Chung, B. A. Nixon, E. Yu, J. Mylopoulos, Non-Functional Requirements in Software Engineering, Kluwar Academic, 2000.

E. Yu, Modelling Strategic Relationships for Process Reengineering, PhD Thesis, U. Toronto, 1995.

E. Yu, Towards Modelling and Reasoning Support for Early-Phase Requirements





 i*モデル

-  ゴールを理由(Rationale)によりモデル化
-  現状システム(As-Is)と将来システム(To-Be)を分析

 SD(Strategic Dependency)[依存関係]モデル

-  プロセス間の構造表現
-  アクタ間の意図(Intention)の依存関係をグラフでモデル化

 SR(Strategic Rationale)[理由]モデル=ゴール指向プロセスモデル

-  プロセス内の構造とその理由(必要条件)を表現
-  各アクタのゴール, ソフトゴールとそれを実現するタスク, リソースとのリンク関係をグラフでモデル化
 -  達成する: Goal, Sub-Goal, 実行する: Task
 -  消費する: Resource

👉 i*の要求獲得プロセス

👉 現状システム(As-Is)のモデル化

- 👉 SDモデル: 現状システム全体のゴール指向プロセスモデル
- 👉 SRモデル: あるアクタのゴール指向プロセスモデル

👉 現状システムモデルの分析

- 👉 ゴールの達成度合い
- 👉 競合などの問題点

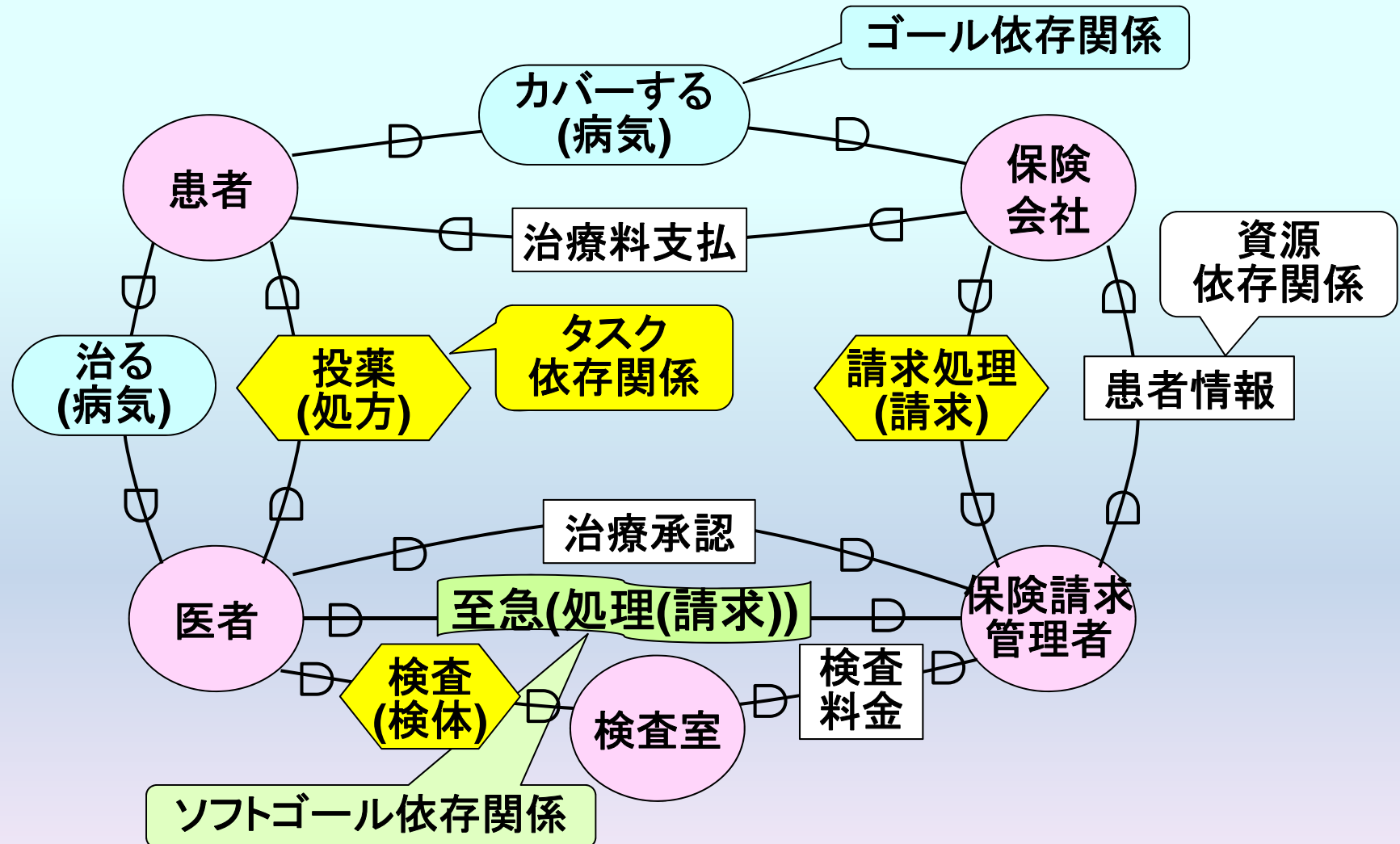
👉 将来システム(To-Be)のモデル化

- 👉 SDモデル, SRモデル

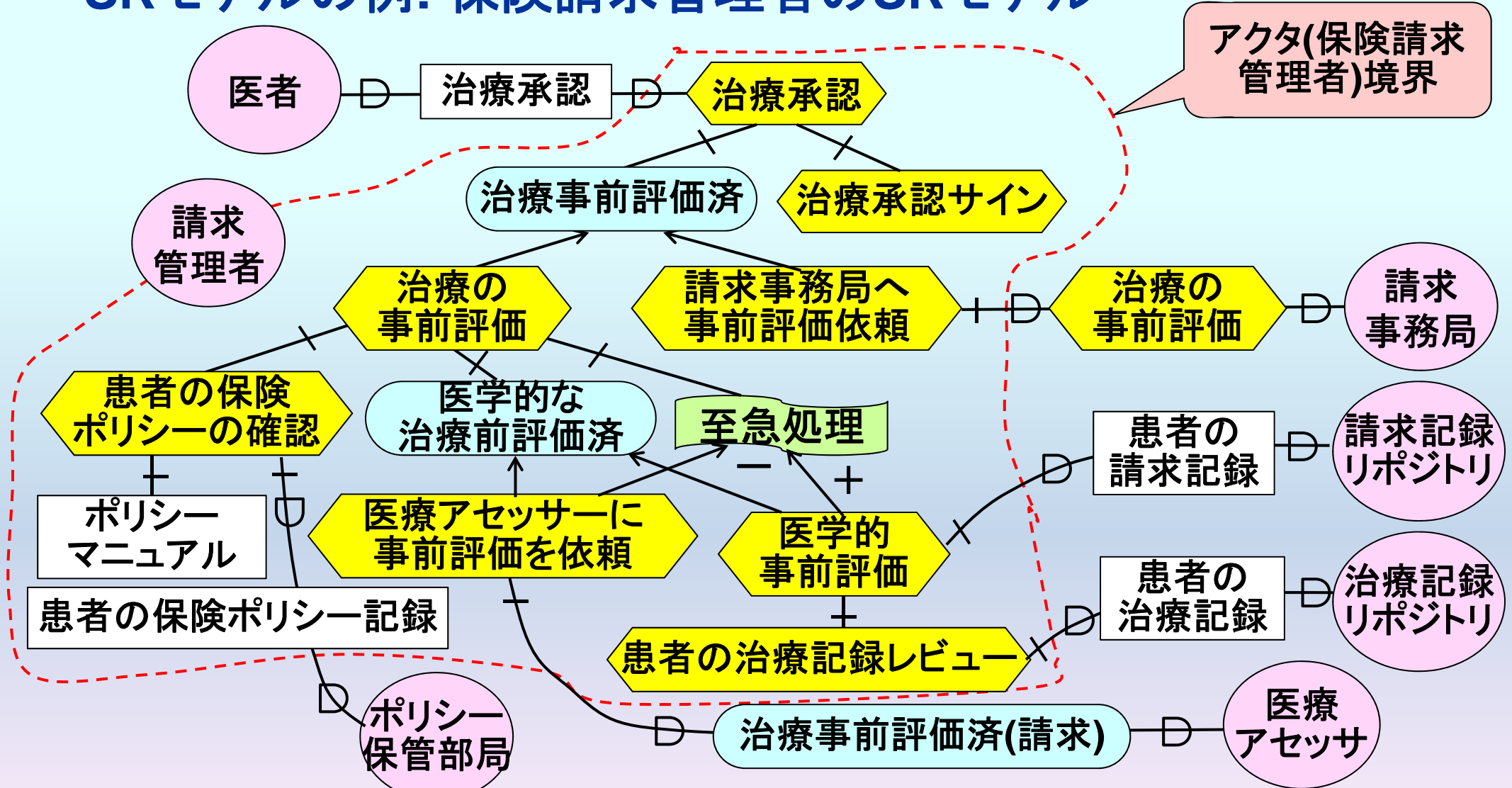
👉 将来システムの評価

- 👉 ゴールの達成度合いの向上
- 👉 現状システムの問題の軽減

👉 SDモデルの例: 医療情報処理



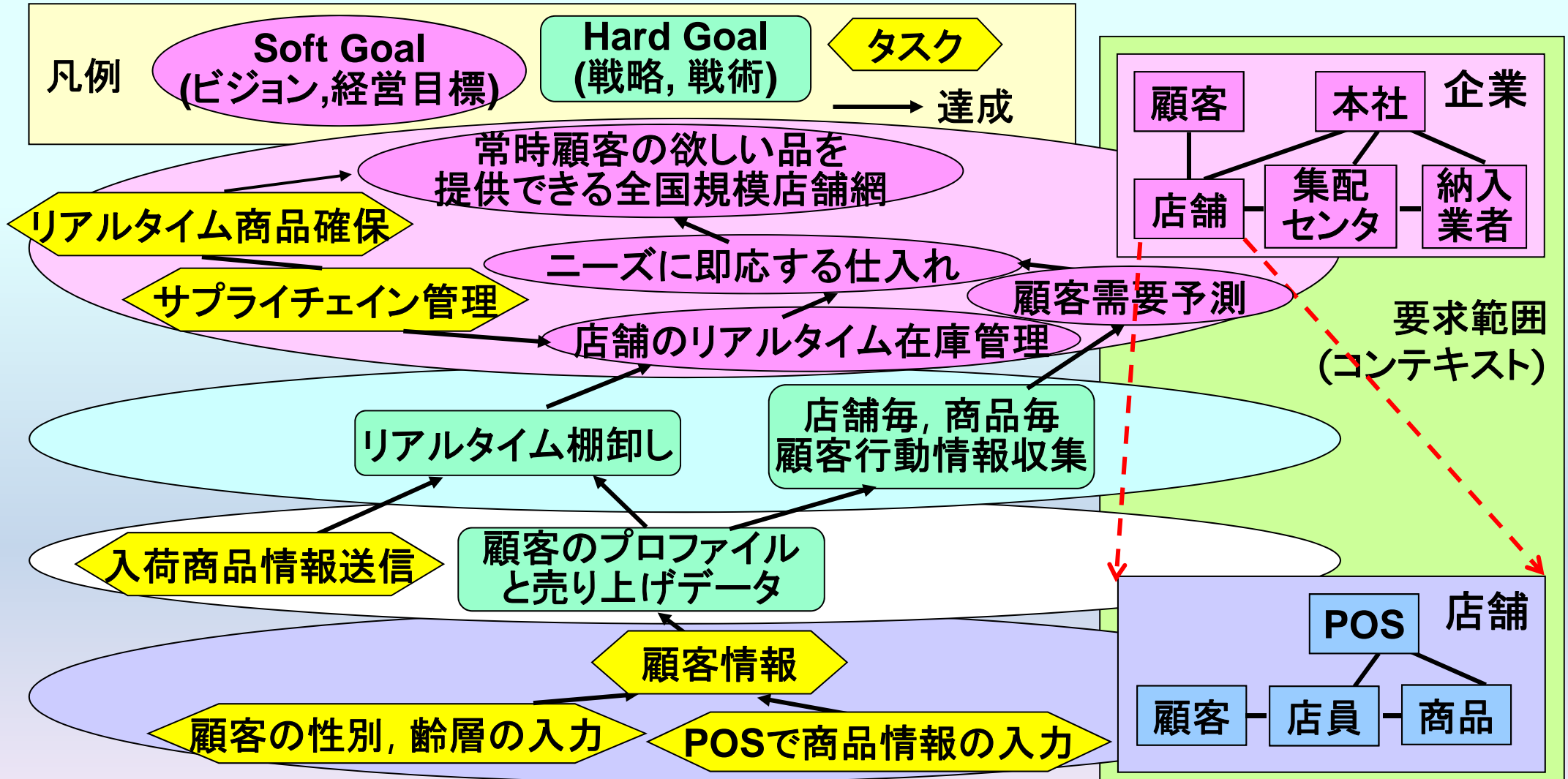
SRモデルの例: 保険請求管理者のSRモデル



要求工学知識体系(REBOK)

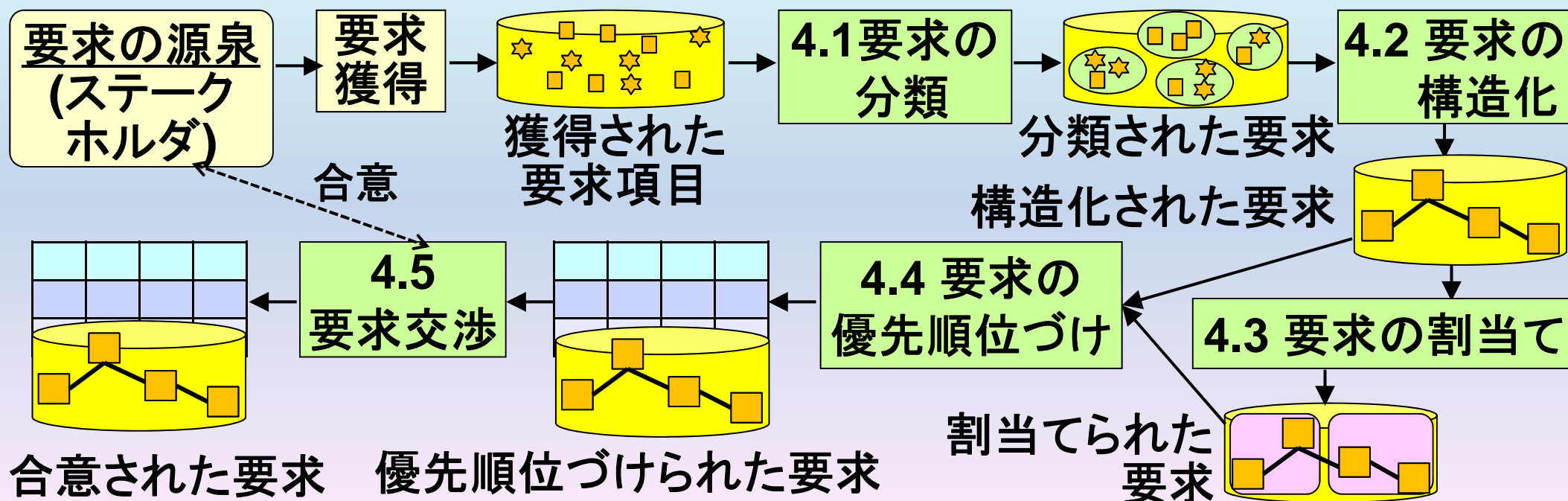
[3] 要求獲得: [3.6] ゴールを達成するための手段の抽出: 事例

経営目標から情報システムへゴールとシステムの階層的展開

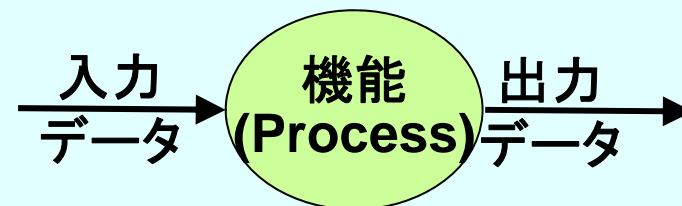


👉 要求分析プロセス: 分析＝分類と関係づけ

- 👉 4.1 要求の分類: 獲得した要求項目を一定の基準に基づき分類
- 👉 4.2 要求の構造化: 要求間を関係づけ, 図表などに整理して表現
- 👉 4.3 要求の割当て: 要求をアーキテクチャに割当て, 実現可能性を確認
- 👉 4.4 要求の優先順位づけ: 重要度, 緊急度などに基づく優先順位づけ
- 👉 4.5 要求交渉: 重要度などに基づき仕様に盛込む要求をユーザと合意



👉 機能の視点に基づく分析方法[詳細後述]



👉 視点: 機能=処理(Process)

👉 モデル: データフロー(入力データ・処理・出力データ)

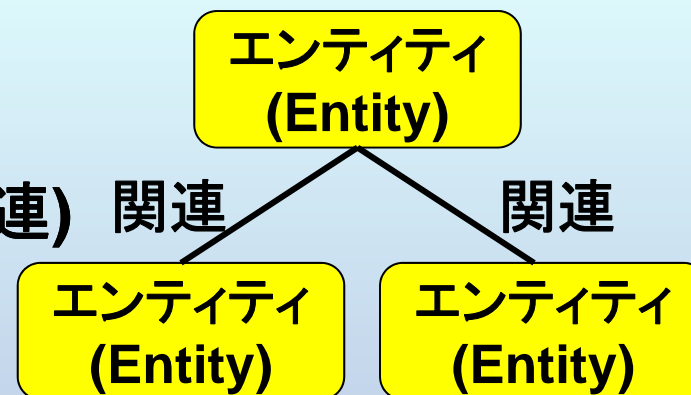
👉 分析方法: データフロー分析(Data Flow Analysis)

👉 構造の視点に基づく分析方法

👉 視点: エンティティ(Entity)=データ

👉 モデル: ER(Entity-Relationship: 実体・関連) 関連

👉 分析方法: 概念データモデリング

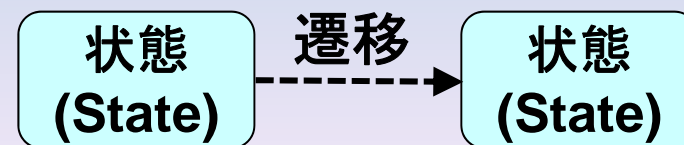


👉 挙動(振舞い)の視点に基づく分析方法

👉 視点: 状態(State)

👉 モデル: 状態遷移(State Transition)

👉 分析方法: 状態分析

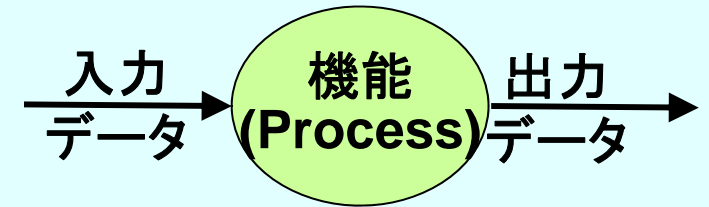


👉 データフローモデル(処理中心)

👉 モデル: IPO(入力データ・処理・出力データ)

👉 分析方法: データフロー分析(Data Flow Analysis)

👉 表現: データフロー図(DFD: Data Flow Diagram)

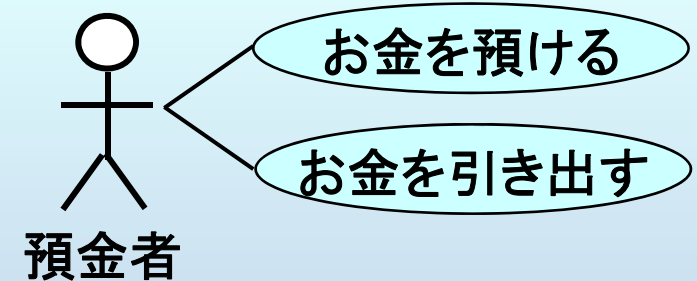


👉 インタラクションモデル(ユーザ中心)

👉 モデル: ユースケース(Use Case)

👉 分析方法: ユースケース分析

👉 表現: ユースケース図, ユースケースのシナリオ



👉 ビジネスプロセス/ワークフローモデル(作業中心)

👉 モデル: ビジネスプロセス, ワークフロー(作業(Work)とその入出力)

👉 分析方法: ビジネス分析(業務分析), ワークフロー分析

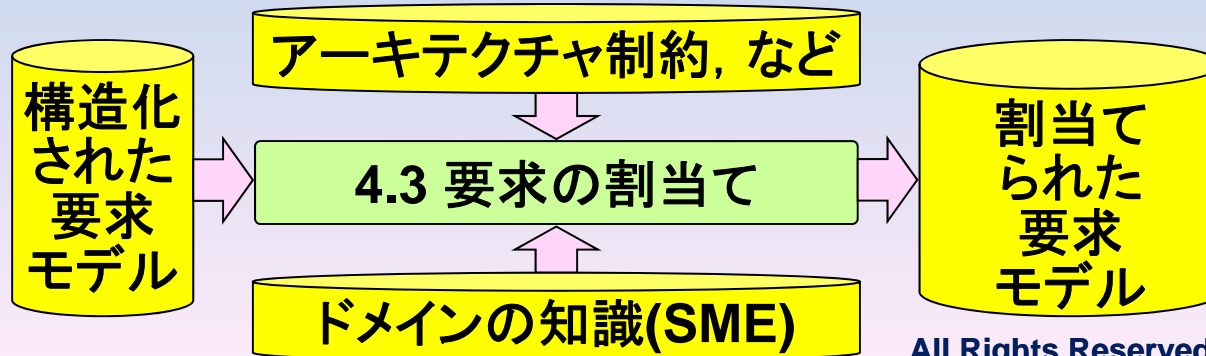
👉 表現: アクティビティ図, BPMN(Business Process Modeling and Notation)

👉 要求の割当て(Requirements Allocation)とは

- 👉 要求を, システムアーキテクチャやソフトウェアアーキテクチャの要素であるコンポーネントに割り当てること
- 👉 注: ある機能要求を実現するアーキテクチャは複数

👉 要求割当ての目的

- 👉 要求の実現可能性の見通しを得る
 - 👉 要求が想定しているアーキテクチャで実現可能か?
- 👉 要求へのアーキテクチャ制約を明確にする
 - 👉 アーキテクチャによる要求への制約
 - 👉 非機能要求への制約: 性能など



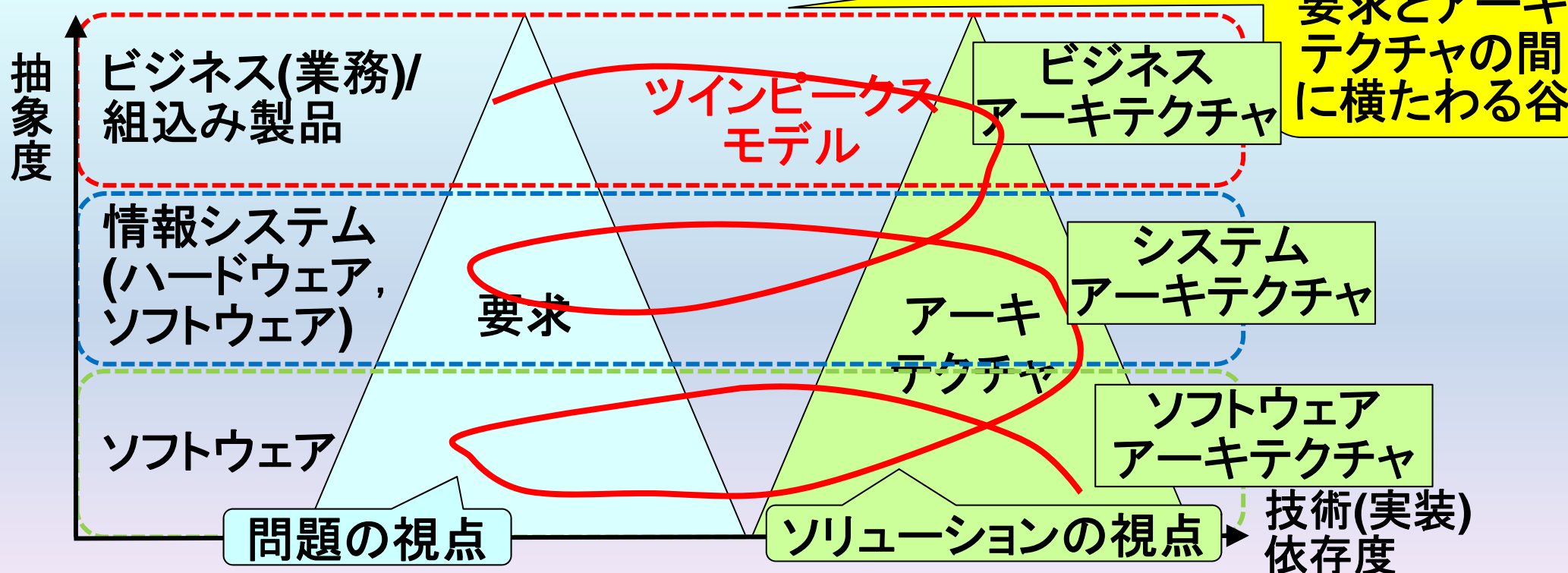
[4] 要求分析: [4.3] 要求の割当て: アーキテクチャ割当ての意義

👉 ツインピークスモデル (Twin Peaks Model)

👉 要求定義とアーキテクチャ設計とは分離不可: 並行プロセス

👉 要求とアーキテクチャは相互に依存し, 同時に重要

👉 並列する抽象構造と繰り返して段階的に詳細化

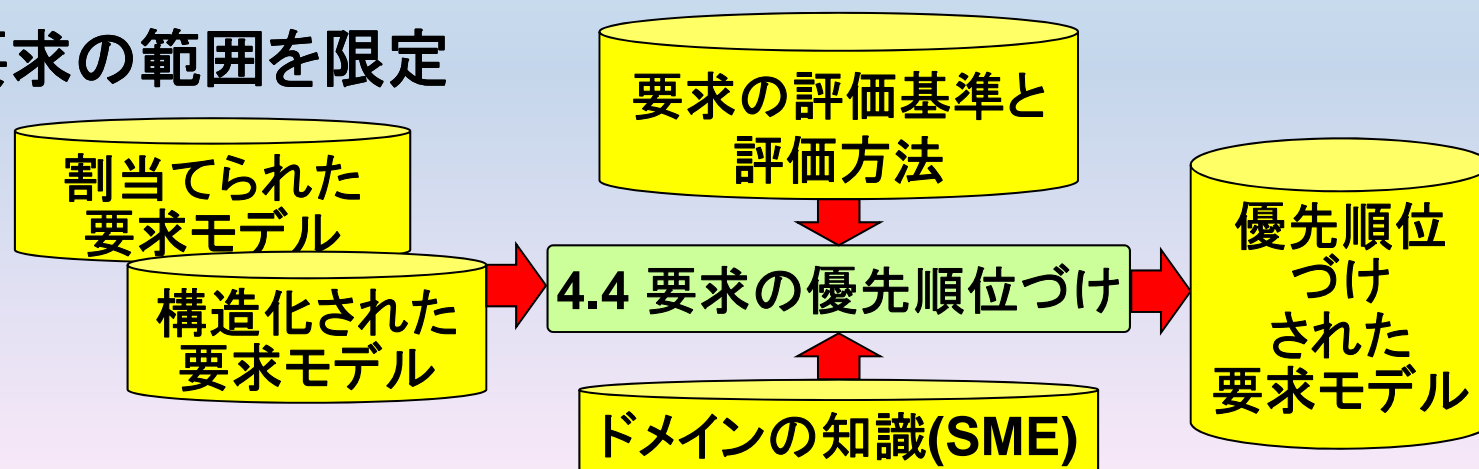


👉 要求の優先順位づけの目的

- 👉 予算, 納期などの制約条件を満たす最も価値の高い要求の決定
- 👉 多目的最適化問題

👉 要求の優先順位づけの内容

- 👉 要求間の矛盾, 対立の解決
- 👉 段階的な引き渡し計画の策定
- 👉 必要なトレードオフを決定
- 👉 ベースライン要求を決定
- 👉 不確定な要求の範囲を限定



👉 定量的/定性的な要求価値評価に基づく

- 👉 優先順位づけマトリクス
- 👉 4象限方式
- 👉 多目的最適化

👉 投票方式

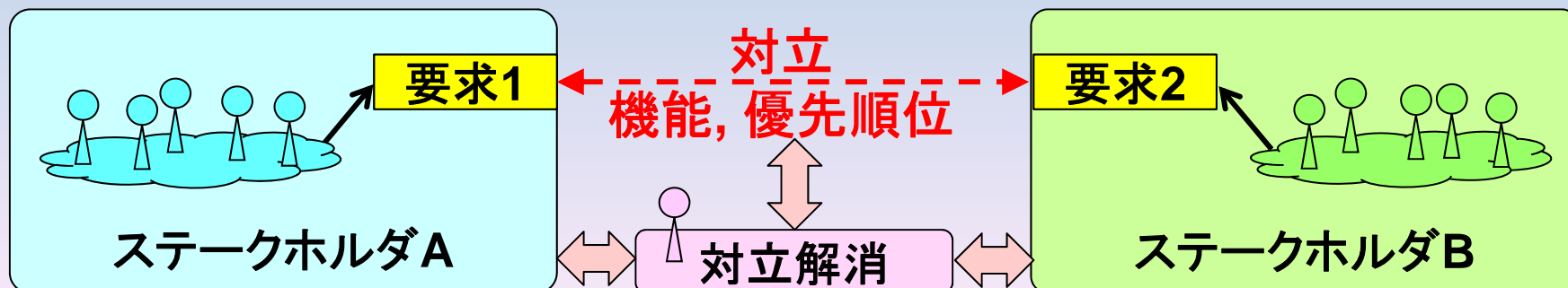
- 👉 100ドルテスト
- 👉 イエス/ノー投票
- 👉 プライオリティ方式
- 👉 MoSCoW

👉 要求交渉とは

- 👉 要求の範囲, 優先順位の妥当性などをステークホルダと合意形成
- 👉 要求の対立(Conflict)を適切なトレードオフにより解消(Resolution)

👉 要求交渉における課題: ステークホルダ間の要求の対立

- 👉 ステークホルダが必要とする要求の優先順位の違い
 - 👉 要求1: ステークホルダAは必要とするが, ステークホルダBは不要
- 👉 ステークホルダが必要とする要求間の機能/非機能の対立
 - 👉 要求AとBの機能的対立: Aの機能はBの機能と排他的
 - 👉 要求AとBの非機能的対立: AとBが実現する要求水準の違い



要求工学知識体系(REBOK) [5] 要求仕様化: [5.0] 要求仕様化プロセス

👉 要求を規定の書式や表記法で記述

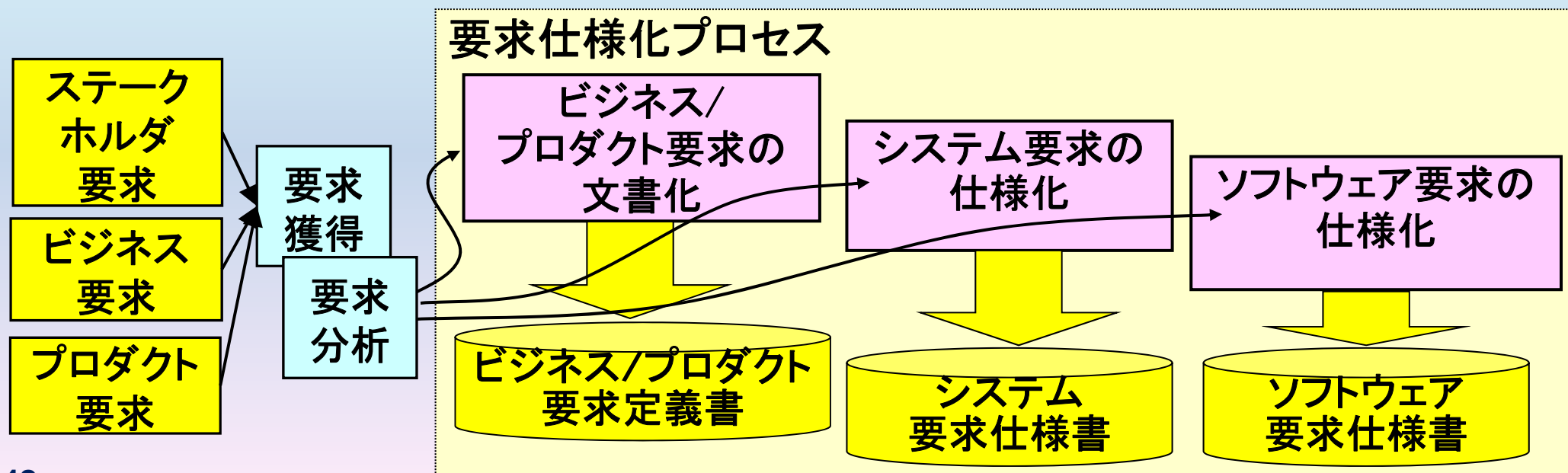
👉 規定の書式に従い, 分析で用いた表記法や自然言語などで記述

👉 3つのスコープに対応した文書テンプレート[国際標準に準拠]

👉 ビジネス要求の文書化: IEEE Std.1362-1998準拠

👉 システム要求の仕様化: IEEE Std.1233-1998準拠

👉 ソフトウェア要求の仕様化: IEEE Std. 830-1998準拠



- (1) 範囲(Scope)
- (2) 要求の背景, 目的(Background, Purpose/Goal, Scope of Requirements)
- (3) 関連ドキュメント (Referenced Documents)
- (4) ニーズと課題 (Needs and Issues)
- (5) 現状 (As-Is) の業務 (プロダクト) システム (Current System)
- (6) 変更とその正当性 (Justification and Nature of Changes)
- (7) 将来 (To-Be) の業務システム構想 (Concepts of Proposed System)
- (8) 組織に対する要求
- (9) 運用 (操作) シナリオ (Operational Scenarios)
- (10) 影響範囲 (Summary of Impacts)
- (11) 将来の業務システムの分析 (Analysis of the Proposed System)
- (12) 前提条件 (Assumptions)
- (13) 制約条件 (Constraints)

付録

- (1) 企業の外部環境、内部環境分析
- (2) 既存の情報システム分析

IEEE Std. 1362-1988, IEEE Guide for Information Technology – System Definition – Concept of Operations (ConOps) Document - Description, IEEE, 1998.

(1)はじめに(Introduction)

(2)システム概要(General System Description)

1)システムコンテキスト(System context)

2)システムの形態, 状況, 状態(System modes, states and conditions)

3)システムの主要機能(Major system capabilities)

4)システムの主要な制約条件(Major system constraints)

5)ユーザ特性(User characteristics)

6)前提条件と関連事項 (Assumptions and dependencies)

7)運用シナリオ(Operational scenarios)

(3)システムの性能, 状態および制約条件

(System Capabilities, Conditions, and Constraints)

1)機器の物理的制約(Physical)

2)システムパフォーマンス特性(System Performance Characteristics)

3)システムセキュリティ(System Security)

4)情報管理(Information Management)

5)システムの利用に関する制約(System Operations)

6)方針と規制(Policy and Regulation)

7)システムのライフサイクルの持続(System life cycle sustainment)

(4)システムインタフェース (System Interface)

(5)システム移行(System Transition)

IEEE Std. 1233-1998,
IEEE Guide for
Developing System
Requirements
Specifications, IEEE,
1998

- (1)はじめに(Introduction)
- (2)全体の概要(Overall Description)
 - 1)ソフトウェアの展望(Software Perspective)
 - 2)ソフトウェア機能概要(Software Functions)
 - 3)ユーザ特性(User Characteristics)
 - 4)制約(Constraints)
 - 5)前提条件と依存(Assumptions and Dependencies)
- (3)ソフトウェア機能要求(Specific Requirements)
 - 1)システム特性
 - 2)外部インタフェース要求(External Interface Requirements)
 - 3)ソフトウェア機能(Function)
 - 4)性能要求(Performance Requirements)
 - 5)論理データベース要求(Logical Database Requirements)
 - 6)設計制約(Design constraints)
 - 7)ソフトウェア品質特性(Software System Attributes)
 - a)信頼性(Reliability)
 - b)可用性(Availability)
 - c)セキュリティ(Security)
 - d)保守性(Maintainability)
 - e)移植性(Portability)
 - f)ユーザビリティ(Usability)

**IEEE Std. 830-1998
Recommended Practice for
Software Requirements
Specifications 準拠**

- 👉 従来の要求工学における検証と妥当性確認の位置づけ
 - 👉 ソフトウェア工学における検証・妥当性確認の定義と異なる
- 👉 SWEBOK: 「妥当性確認」のアクティビティのみ定義
 - 👉 妥当性確認の中に「検証(完全性などの確認)」と「妥当性確認」を含む
- 👉 現在の定義: ISO 29148 Requirements Engineering
 - 👉 Validation: confirmation that the requirements for a specific intended use or application have been fulfilled
 - 👉 Verification: confirmation that specified requirements have been fulfilled

文献 [発行年]	Thayer [1990]	Kotonya [1998]	SWEBOK [2004]	Cheng [2007]	ISO 29148 [2011]	REBOK [2011]
検証	○	○	○(妥当性確認に含む)	○	○	○
妥当性確認	×	×	○	○	○	○
評価	×	×	×	○	○	○

👉 検証(Verification)[正当性の確認]

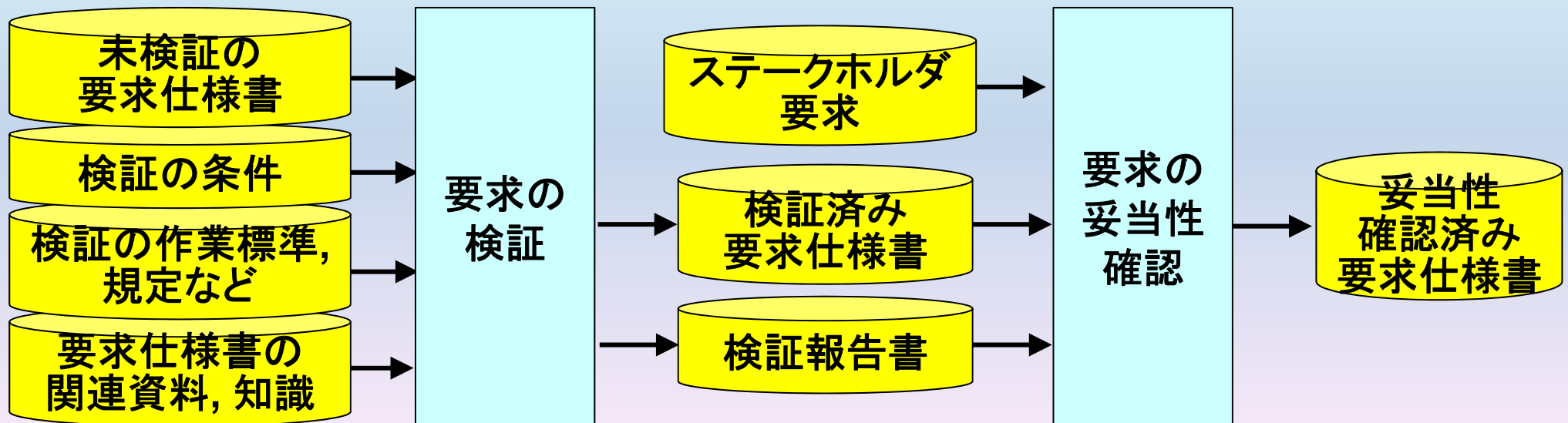
👉 要求仕様書を構造的, 意味的な特性に照らして正しいことを確認


👉 妥当性確認(Validation)

👉 要求仕様書がステークホルダ要求を満たしていることを確認

👉 評価(Evaluation)

👉 種々の要求評価基準に基づき, 要求の良さを評価



 検証方法 要求レビュー 例: 構造化ウォークスルー, 要求工学ワークショップ チェックリスト(What-If法) 検証の条件として考慮すべき要求仕様の特性の例

単一性(Cohesive)	要求の対象がひとつであること
完全性(Completeness)	要求に漏れや不完全な記述がないこと
一貫性(Consistency)	要求に矛盾がないこと
法令遵守(Compliance)	法律や規制などに準拠していること
独立性(Non-Conjugated)	要求がそれ自体で完結し, 他の要求に依存していないこと
追跡可能性(Traceable)	要求の源泉や設計など, 前後の工程の成果物と関連づけることが可能であること
最新性(Current)	要求が最新の条件に基づいていること
実現可能性(Feasible)	要求がプロジェクトや環境などの特別な制約なしに, 実現可能であること

[6.5] プロトタイピング: プロトタイピングの分類

👉 検証スコープによる分類

- 👉 水平プロトタイピング[静的]: 紙やプレゼンテーションツールを用いて, 画面のレイアウトなどを作成し, 確認
- 👉 垂直プロトタイピング[動的]: 画面とその入出力操作や遷移なども含めて確認を行い, アーキテクチャの実現可能性を含めて確認

👉 実現手段による分類

- 👉 ペーパ: ソフトウェアの実装を含まない(「ペーパ」は電子も含む)
- 👉 ソフトウェア: ソフトウェアの実装を含む

		プロトタイプ開発の方針	
検証スコープ	実現手段	進化型	使い捨て
水平 (モックアップ)	ペーパ	—	○
	ソフトウェア	○	○
垂直 (実行を含む)	ペーパ	—	—
	ソフトウェア	○	○

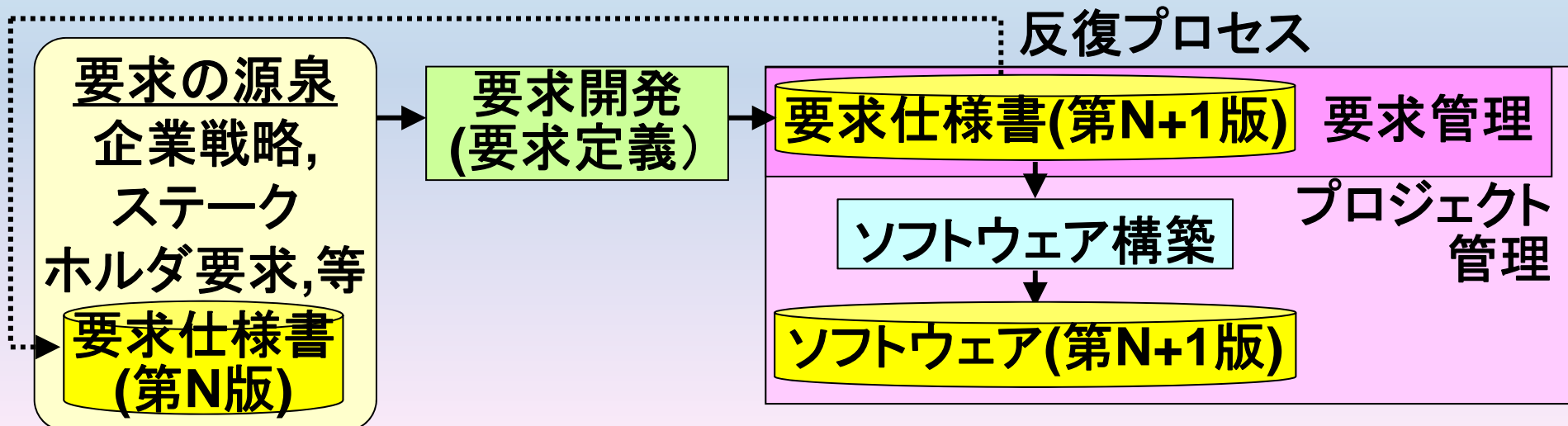
👉 要求管理の必要性

- 👉 要求の追加・変更の必要性
- 👉 プロジェクト管理の一環としての要求管理の必要性

👉 要求管理の対象

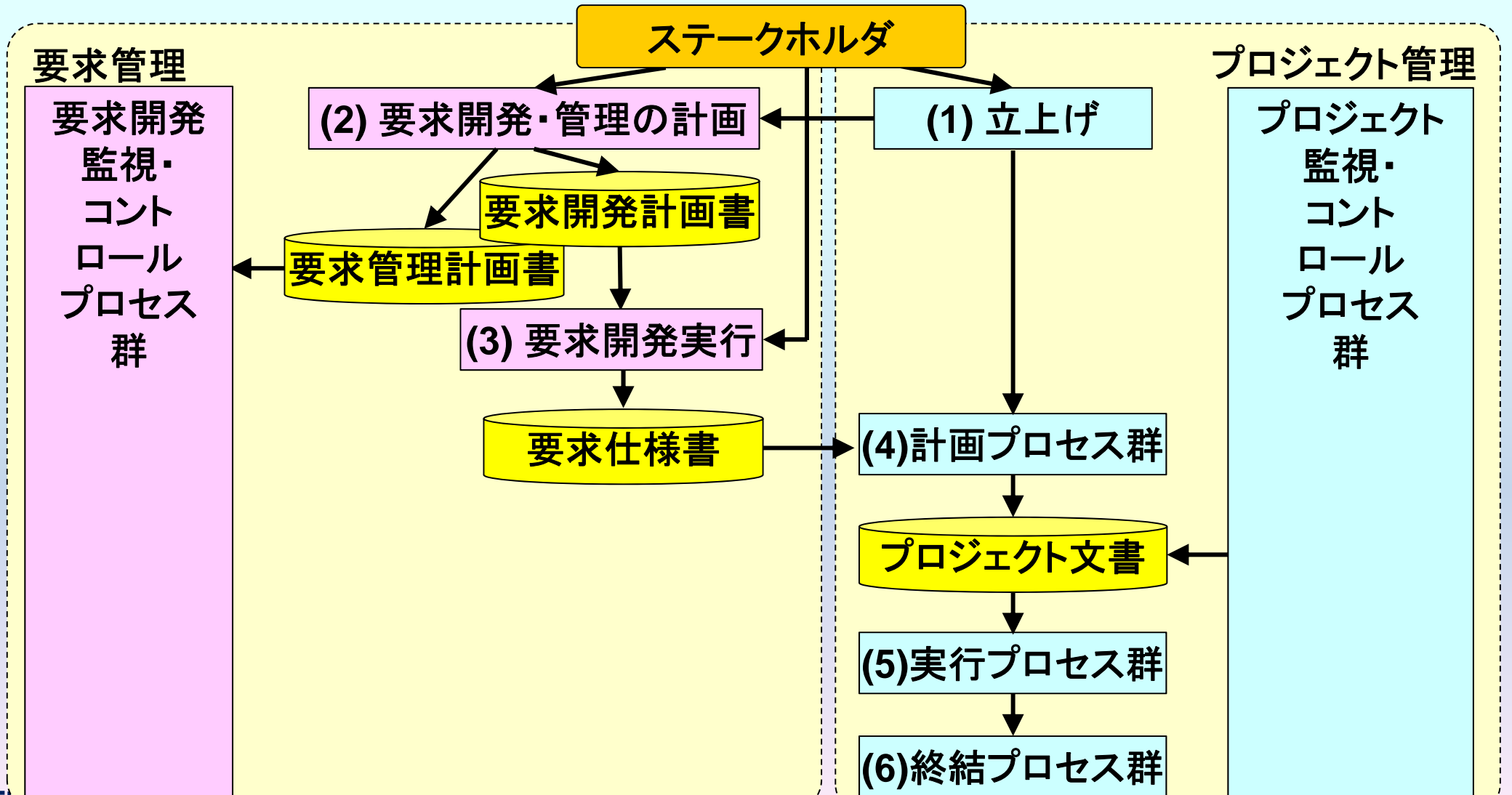
- 👉 要求仕様書: 最も価値のある文書の一つ
- 👉 要求仕様書の変更プロセス

👉 要求管理技術: 要求管理のための技術体系

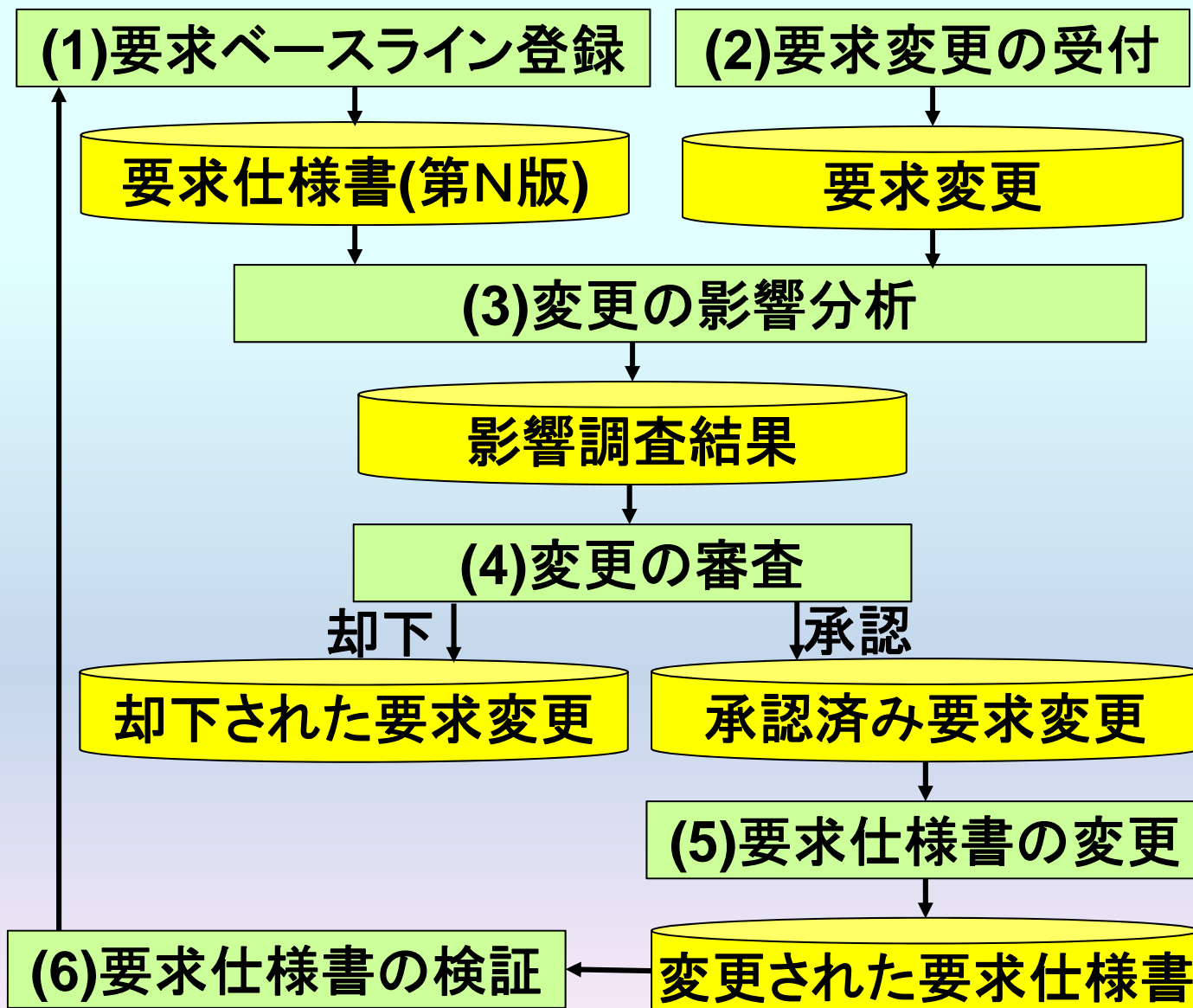


👉 PMBOKのプロジェクト管理プロセスと要求管理プロセス

👉 プロジェクト管理の計画: 要求仕様を計画の入力とする



- 👉 要求変更: 主として要求仕様書の変更
- 👉 ベースライン: ステークホルダと合意し, 承認された要求仕様書として, 開発, 見積りの基礎となる
- 👉 変更の審査: 影響分析に基づきステークホルダと変更内容を選択し, 合意



👉 要求トレース管理(Requirements Trace Management)

- 👉 要求仕様書をもとに, 個々の要求項目の発生から結果までを記録し, 検索するアクティビティ
- 👉 要求の依存関係を明らかにする
 - 👉 要求相互の依存性
 - 👉 要求とシステム設計, コンポーネントなど実装との依存性

👉 要求ト्रेसサビリティ(Requirements Traceability)

- 👉 トレース可能であること, または, その範囲
- 👉 要求トレースのための関係づけを要求属性として設定

👉 要求トレースの分類

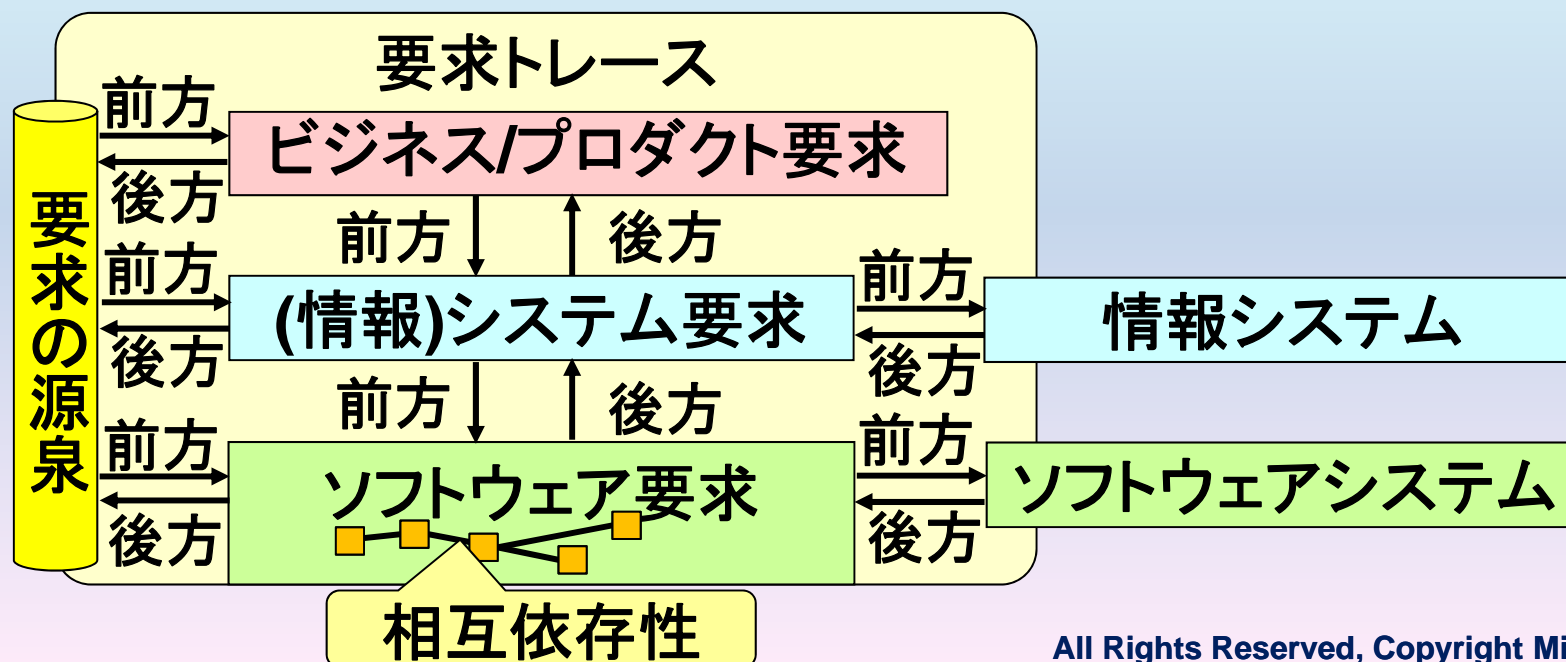
👉 (垂直)トレース(Extra-Requirements Traceabilityとも呼ぶ)

👉 前方(Forward)トレース

👉 後方(Backward)トレース

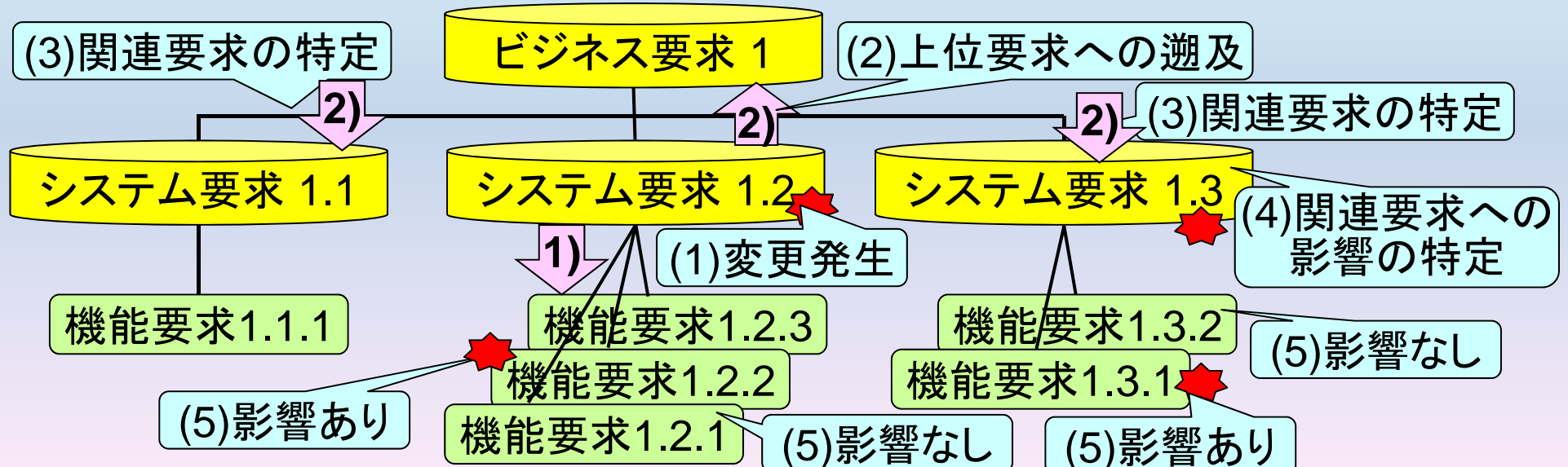
👉 (水平)トレース: 相互依存性(Inter-dependency) (Intra-Requirements Traceabilityとも呼ぶ)

👉 要求相互間の関係



👉 要求トレース情報を用いた影響分析の手順(例)

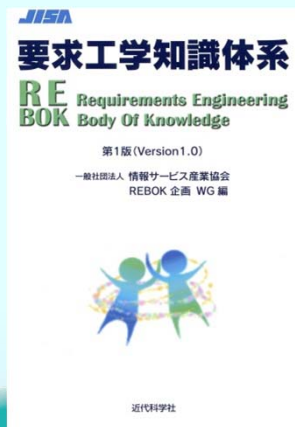
- 👉 1) 前方トレースによる影響の特定
 - 👉 (4)(5)前方トレースにより, 影響の有無を特定(4)(5)
- 👉 2) 後方/前方トレースによる影響の特定
 - 👉 (1)変更する要求の特定
 - 👉 (2)上位要求を追跡
 - 👉 (3)変更対象の要求と関係する要求を特定
 - 👉 (4)(5)変更対象の要求との関係から影響の有無を特定



要求工学知識体系(REBOK) REBOK活用のヒント: 活用のユースケース

要求工学とは
何か, 知りたい

要求工学の基礎を
お読みください
後の章は, 必要に
応じてお読みください



要求工学の導入
を推進したい

REBOKで要求工学の
全体像をつかんでください
要求定義に必要な役割と
行うべきアクティビティを
理解して下さい

要求工学を適用
して良い要求
定義をしたい

REBOKの中から
開発に応じて必要な
技術を選んでください
全て適用する必要は
ありません



要求工学を実践
できる人材を
育成したい

REBOKで要求工学の
人材育成コースを
作りましょう
1時間, 1日, 3日など
ステップアップしましょう

👉 REBOKの利用

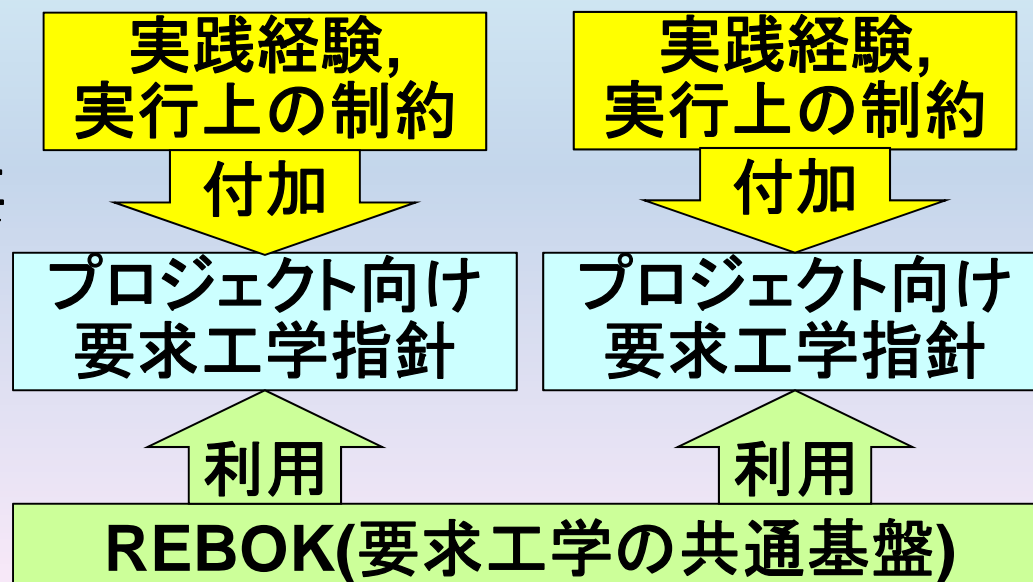
- 👉 REBOKは要求工学全体を網羅する基盤
- 👉 技術は必要に応じて選択して利用(全て利用する必要はない)

👉 現場の経験と制約の付加

- 👉 REBOKは要求工学の共通基盤として利用
- 👉 現場のプラクティス, 制約などを付加

👉 要求工学実践の秘訣

- 👉 REBOK第8章「実践の考慮点」
- 👉 JISA要求工学WGで収集した要求工学ベストプラクティス
- 👉 要求工学グッドプラクティスガイド: 3段階の実践ガイドライン



要求工学知識体系(REBOK) REBOK活用のヒント: 活用の期待効果

- 👉 個人の経験知を体系的, 組織的な知識化
 - 👉 REBOKに基づき経験知を共有できる組織の知識化
 - 👉 REBOKに基づき, 個人が経験知を体系的知識へ強化
- 👉 要求工学の組織的実践
 - 👉 共通基盤に基づくことによる, 組織全体の底上げ
 - 👉 個人能力の底上げ, 技術の漏れや誤りの防止
 - 👉 共通基盤上で競争力のある組織的取組の構築
- 👉 ユーザとベンダ間の共通理解の醸成
 - 👉 企業固有の用語やプロセスの共通化
 - 👉 経営者からエンドユーザまで

要求工学知識体系(REBOK) 今後のロードマップ

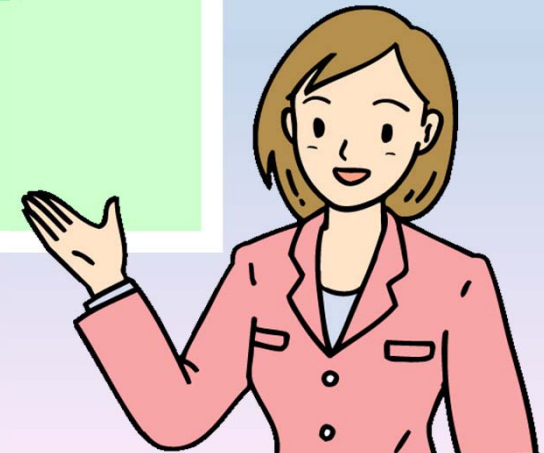
- 👉 要求工学知識体系(REBOK)ガイドの編成
 - 👆 REBOKの解説と実践の手引き
- 👉 REBOKに基づく人材育成ガイドラインの策定
 - 👆 人材育成カリキュラムのモデル策定
- 👉 REBOKベストプラクティスと現場からのフィードバック
 - 👆 ベストプラクティスの収集と公開
- 👉 要求工学のグローバルコミュニティとの関係
 - 👆 英語版の作成



3.

要求工学の新潮流

要求工学の最新トピックスを
紹介します



👉 要求工学の技術と対象の両面で急速な進化

👉 技術的進化

👉 複雑度と多様性への対応

👉 動的化

👉 人と社会への浸透(スコープの拡張)

👉 ビジネス, 社会的な問題への適用拡大

👉 ビジネス価値, ユーザ価値を極大化できるIT(要求)

複雑度と多様性(論理的進化):
プロダクトライン, サービス化,
モバイル, コンテキストウェア

動的化(時間的進化):
アジャイル, RE@Run.Time

人と社会への浸透(スコープ拡張):
ユーザ中心, エモーショナル,
法令, グローバル, セキュリティ,

要求工学の新潮流

要求の対象システムの進化と拡大

👉 情報システムの進化への対応

👉 サービス, クラウドの要求工学

👉 組込み, モバイル製品への拡張

👉 プロダクトライン要求工学, 多様性への対応

👉 モバイルとコンテキストウェア(Contextual Design)への対応

👉 ユーザ中心要求工学(User-Centered Requirements Engineering)

👉 ユーザインタラクションとユーザ経験(UX: User eXperience)

👉 エモーショナル要求工学

👉 グローバル化: グローバル要求工学

👉 社会化

👉 法令要求工学(Legal Requirements Engineering)

👉 政治と力関係(Power and Politics in RE)

👉 新たな要求工学モデル

👉 探索型(Inquiry-Based)

👉 発見サイクル(Discovery Cycle)[Alexander '09]

👉 Volere [Robertson & Robertson '99, '06]

👉 継続的要求工学(Continuous RE)

👉 継続的要求工学[Pohl '10]

👉 組込みシステムの要求工学 [Berenbach '09]

👉 アジャイル要求工学(Agile RE)

👉 ランタイム要求工学(RE@Run.Time)

👉 実行しながら要求変更するシステムのための要求工学

👉 自己適応システム(Self-Adaptive Systems)

👉 発見サイクル(Discovery Cycle)

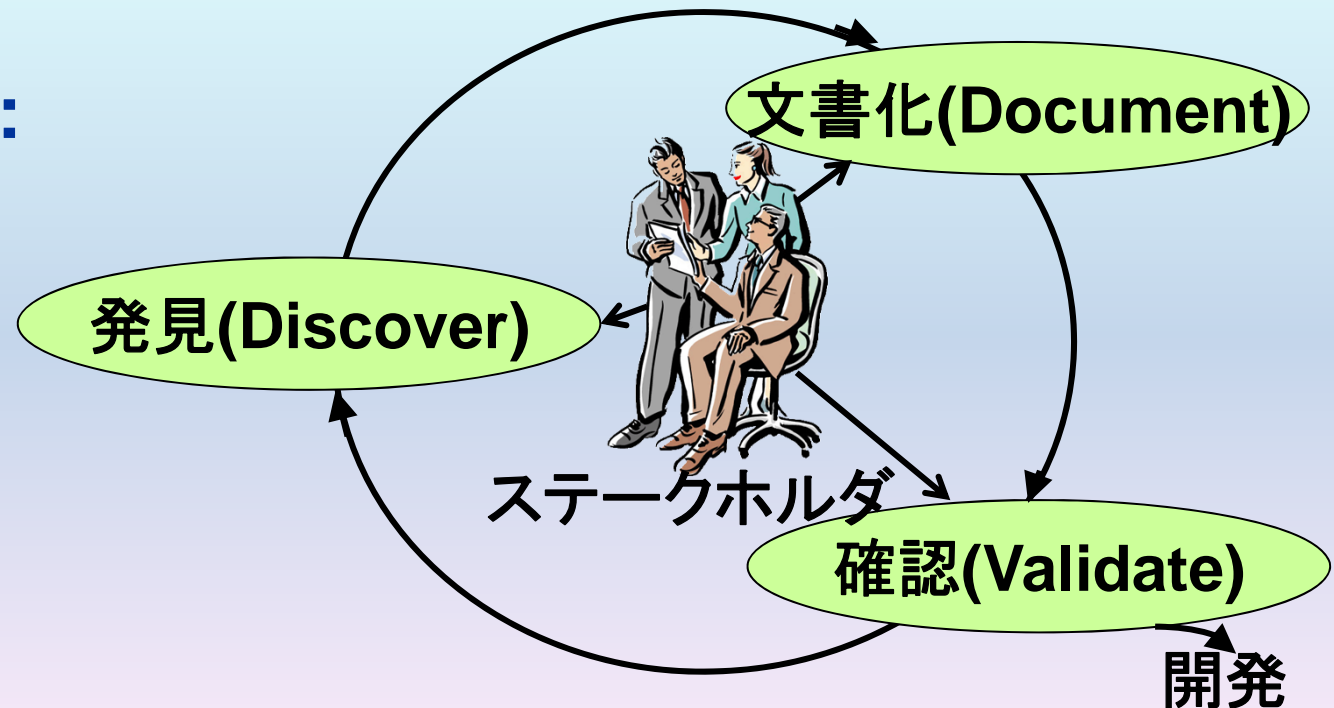
👉 基本戦略

👉 ステークホルダと「要求」の段階的, 協調的獲得

👉 構造モデルの特徴: 簡潔化

👉 発見=獲得+分析

👉 挙動モデルの特徴: 強い繰り返し構造



要求工学の新潮流 要求工学モデル: 継続的要求工学

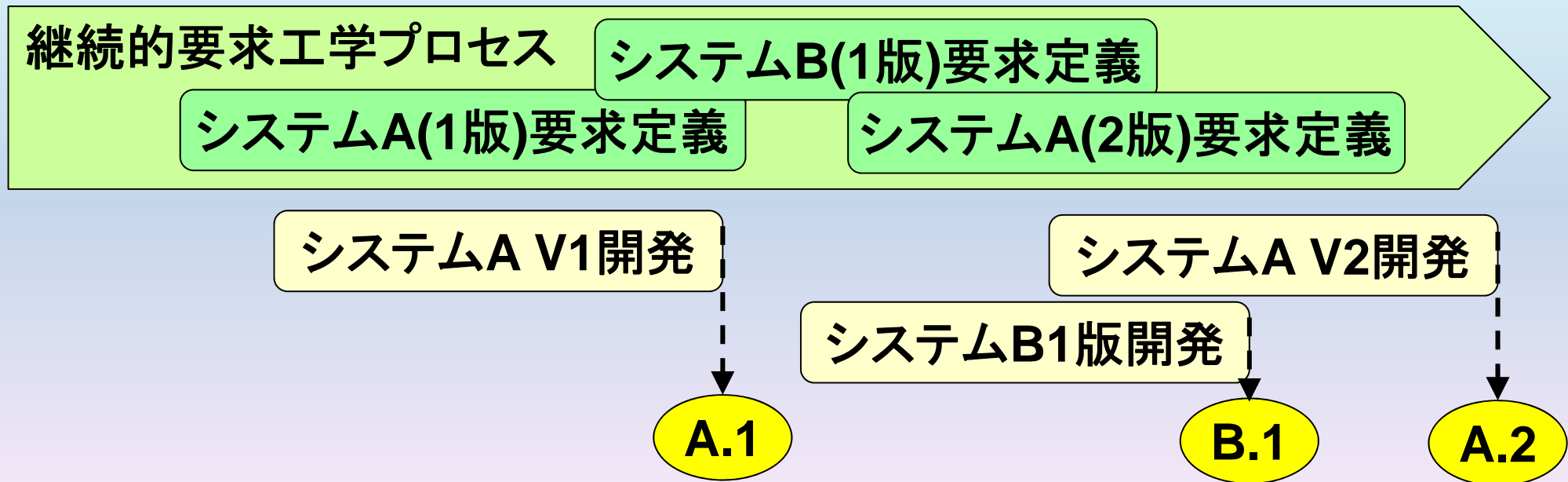
👉 プロジェクト・プロダクト境界を越えて続く要求工学プロセス

👉 例: プロダクトライン要求工学

👉 スコープの拡張

👉 クロスライフサイクル: 製品の複数ライフサイクル(版)

👉 クロスプロジェクト/クロスプロダクト: 複数製品

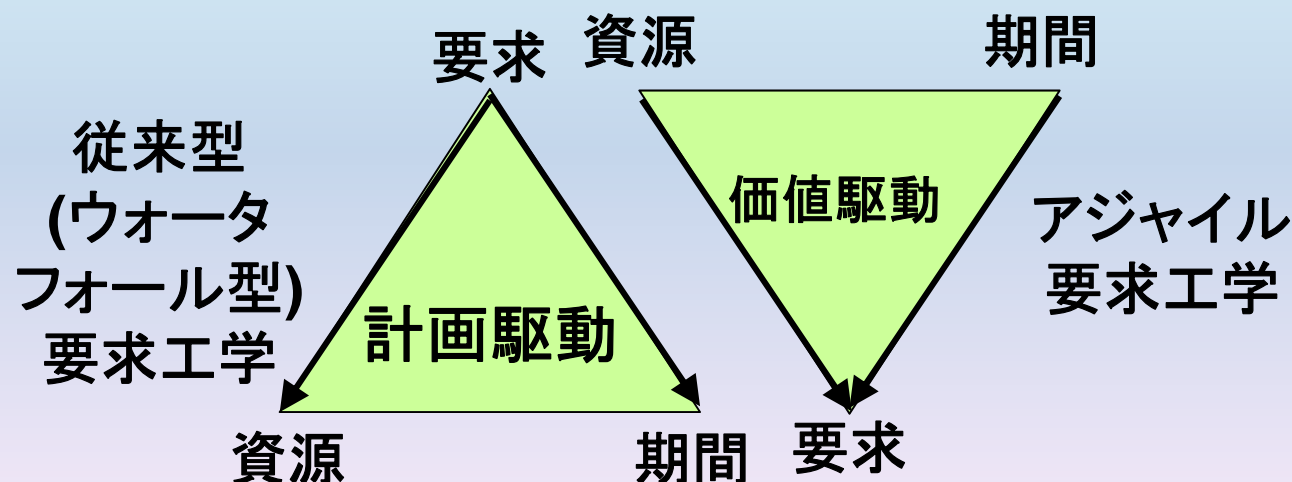


👉 アジャイル要求工学

- 👉 アジャイル開発のための要求工学(RE for Agile Development)
 - 👉 短期繰り返しサイクル⇒要求と成果物との関係が直接的
 - 👉 例: SCRUM
- 👉 要求工学モデルの見直し
 - 👉 計画駆動(Plan Driven) ⇒ 価値駆動(Value Driven)
 - 👉 インクリメンタルな価値提供

👉 研究の状況

- 👉 アジャイル開発における要求定義の構造化
- 👉 多くの研究課題



👉 開発(Design.Time)と利用(Run.Time)の融合

- 👉 自己適応型システム(Self-Adaptive)
- 👉 サービス指向, クラウドコンピューティング: 利用しながら変更




👉 技術的課題

- 👉 実行時に自己の要求の表現(Run-Time Representations of Reqs)
- 👉 要求モデルの進化とアーキテクチャとの同期(Evolution of the Reqs Model and its Synchronization with the Architecture)
- 👉 不確定性への対応(Dealing with Uncertainty)

👉 研究の現状

- 👉 2010年からワークショップ開催
- 👉 コミュニティは小さいが今後重要な研究課題となる可能性

 国内における方法論の提案と適用

-  企業内の要求定義技術の体系化
-  ビジネスからITソリューションへの連携(Strategic Alignment)
-  ドメインへの対応: エクスペリエンス指向

方法論名称	提供企業	適用対象
TERASOLUNA	NTT DATA	企業情報システム
Tri-Shaping	富士通	企業情報システム
エクスペリエンス指向アプローチ	日立	ユーザ経験関連

参考文献: TERASOLUNA, <http://www.terasoluna.jp/>.

若杉 賢治ほか, 新手法に見る要件定義の鉄則(前篇)(後編), 日経コンピュータ, 2011年5月12日号, pp. 80-85, 2011年6月9日号, pp/ 112-115.

坂野 裕ほか, お客様との協創を実現するエクスペリエンス指向アプローチによるシステム開発, 日立評論, Jul. 2009, pp. 60-62.

👉 要求の3つのスコープ毎の研究コミュニティ

- 👉 ビジネス/製品, (情報)システム, ソフトウェア
- 👉 REはソフトウェア工学と共通のコミュニティ
- 👉 この他ソフトウェア工学国際会議(ICSE)など

スコープ	会議名/主催組織[URL]	創設年
ビジネス	BPM(International Conference on Business Process Management)/会議体 [www.bpmYYYY.org]	2003
システム	Annual INCOSE International Symposium / INCOSE (International Council on Systems Engineering) [www.incose.org]	1991
ソフトウェア	RE(International Requirements Engineering Conference)/ IEEE Computer Society [www.requirements-engineering.org, www.reYY.org]	1993

要求工学の新潮流 要求工学国際会議(RE)

- 👉 日程: 毎年8月末~9月
- 👉 参加者数: 250~300人
- 👉 開催地: トレント@イタリア(2011), シカゴ(2012)
- 👉 公募論文カテゴリ: 研究(研究, 評価, ビジョン), 実践



👉 要求工学=ソフトウェア工学のフロンティア

- 👉 課題の広がりと深まりの増大
- 👉 ソフトウェア工学領域で最も新しい領域

👉 ソフトウェア工学を超えて

- 👉 要求工学の対象>ソフトウェア工学
- 👉 情報技術に加え, ビジネス(経営学), 人間(心理学), 社会(社会学)などの多様な側面

👉 研究チャレンジ

- 👉 多くの新しい研究課題
- 👉 実践の課題
 - 👉 ユーザ: 競争力の源泉
 - 👉 ベンダ: ソフトウェア開発成功の鍵

👉 要求工学知識体系(REBOK)

👉 要求工学の整理と体系化:要求工学の全体を示す地図

👉 ユーザとベンダによらない役割に基づく枠組み

👉 現場の視点から、ソリューションへの橋渡し

👉 要求: ビジネス/プロダクト, システム, ソフトウェア

👉 要求定義: 獲得, 分析, 仕様化, 検証・妥当性確認・評価

👉 要求管理

👉 要求工学の新潮流: 多くの研究課題



- [1] A. Abran, et al. (eds.), **Guide to the Software Engineering Body of Knowledge, 2004 Version**, IEEE Computer Society, <http://www.swebok.org> [ソフトウェアエンジニアリング基礎知識体系, オーム社, 2005].
- [2] 青山 幹雄ほか, 要求工学知識体系(REBOK)の開発と評価, ソフトウェアエンジニアリング最前線2010, 近代科学社, Aug. 2010, pp. 25-32.
- [3] M. Aoyama, T. Nakatani, and S. Saito, **REBOK Manifest: Towards a Requirements Engineering Body Of Knowledge**, Proc. IEEE RE '10, IEEE CS, Sep.-Oct. 2010, pp. 383-384.
- [4] 青山 幹雄ほか, 要求工学のモデル論, 電子情報通信学会 知能ソフトウェア工学, Mar. 2011, pp. 43-48.
- [5] B. H. C. Cheng and J. M. Atlee, **Research Directions in Requirements Engineering**, Proc. ICSE '07, Future of Software Engineering, IEEE CS, May 2007, pp. 285-303.
- [6] IIBA, **A Guide to the Business Analysis Body of Knowledge (BABOK Guide), Version 2.0**, IIBA, 2009 [ビジネスアナリシス知識体系ガイド Version 2.0, IIBA日本支部, 2009].
- [7] IREB, **Syllabus: IREB Certified Professional for Requirements Engineering Foundation Level, Ver. 2.0**, Oct. 2009, <http://certified-re.de/en/>.
- [8] ISO/IEC 29148:2011, **Systems and Software Engineering – Life Cycle Processes – Requirements Engineering**, 2011[発行予定].
- [9] JISA REBOK 企画WG, 要求工学知識体系, 第1版, 近代科学社, 2011.
- [10] 経済産業省/JUAS, 要求仕様策定ガイドライン, 2007.
- [11] B. Nuseibeh and S. Easterbrook, **Requirements Engineering: A Roadmap**, Proc ICSE '00, Future of Software Engineering, ACM, May 2000, pp. 37-46.
- [12] 大西 淳, 郷 健太郎, 要求工学, ソフトウェアテクノロジーシリーズ, Vol. 9, 共立出版, 2002.
- [13] K. Pohl, **Requirements Engineering**, Springer, 2010.
- [14] K. E. Wiegers, **Software Requirements**, 2nd ed., Microsoft Press, 2003 [ソフトウェア要求:顧客が望むシステムとは, 日経BPソフトプレス, 2003].
- [15] E. Yu, et al, **Social Modeling for Requirements Engineering**, MIT Press, 2011.



ご清聴ありがとうございます

REBOKで要求工学を使いこなそう！