

ソフトウェアテストシンポジウム2008 札幌

ソフトウェア要求の「型」 ～最初は「守」から～

2008年 10月 24日

クロススペース代表

渡部 洋子

yokowtnb@zd6.so-net.ne.jp

(講師紹介)

渡部 洋子 Yoko Watanabe

(クロススペース代表、戦略経営ネットワーク協同組合理事)

E-mail yokowtnb@zd6.so-net.ne.jp

ITコーディネータ、

ISMS主任審査員、

日本経営品質協議会認定セルフアセッサー、

日本システム監査人協会北海道副支部長、

公認システム監査人補、

情報処理技術者:システム監査、プロジェクトマネージャ、

(旧)特種、情報セキュリティアドミニストレータ、

監訳書「ソフトウェア要求」(2003)、

「RFP入門」(2004)、

「脅威モデル」(2005)、

「次世代CIO」(2006) (すべて日経BPソフトプレス社)

目次

1. 概要

- (1)要求とは？
- (2)要求定義の問題点
- (3)ソフトウェア工学と
要求工学

2. 要求の体系

- (1)全体図
- (2)要求の種類
- (3)開発プロセスと要求

3. 良い要求とは

- (1)良い要求とは？
- (2)優れた要求の特性
- (3)避けるべき表現
- (4)優れた要求を作成するには

4. 有益なプラクティス

- (1)要求開発のプラクティス
- (2)要求管理とその他の
プラクティス

参考文献

1. 概要

(1)要求とは？

ソフトウェア要求(Software Requirements)は、

{ システムの外側か
ら見た振る舞い }

ユーザーから
見た定義

1. 問題を解決したり、目的を達成するために、利用者にとって
必要な条件や能力

2. 契約書や標準規約書や仕様書やその他の正式な文書を満
たすために、システムやシステムの構成要素が満たすべき、
あるいはもつべき条件や能力

3. ソフトウェアやソフトウェアの構成要素を開発して
いくための基礎となるすべての要求の集合

4. ソフトウェア要求仕様書という用語の短縮形

開発者から見
た定義

{ システム内部
の特性 }

である。

出典:大西淳・郷健太郎 著「要求工学」

【要求と要件】

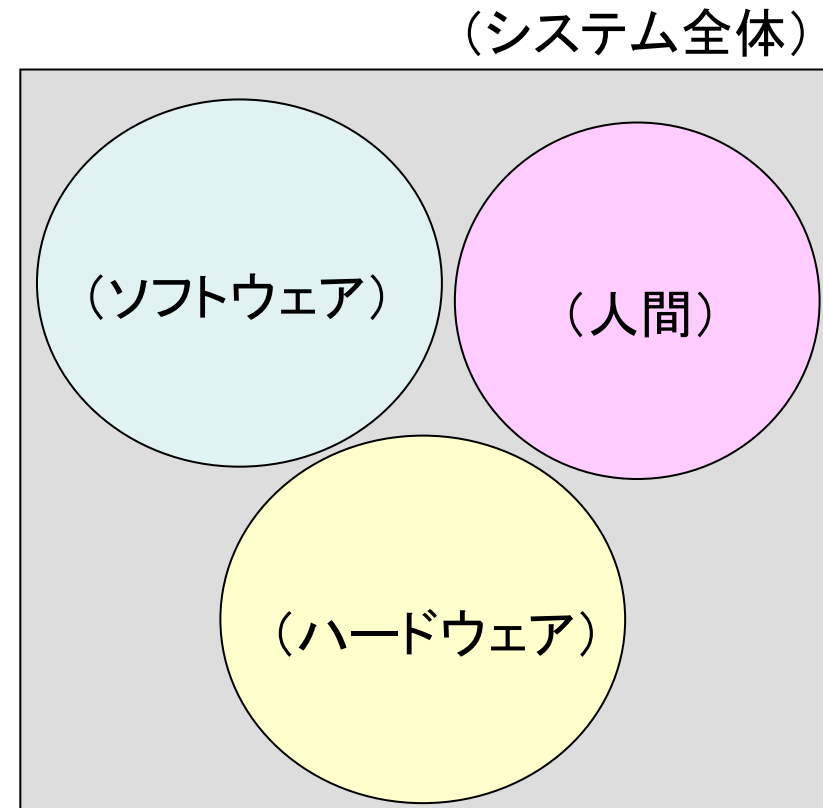
Requirements: 要求、要件

要求－顧客の視点から
システム全体に望むもの

要件－開発者の視点からシ
ステムとして実現しなければ
ならないこと

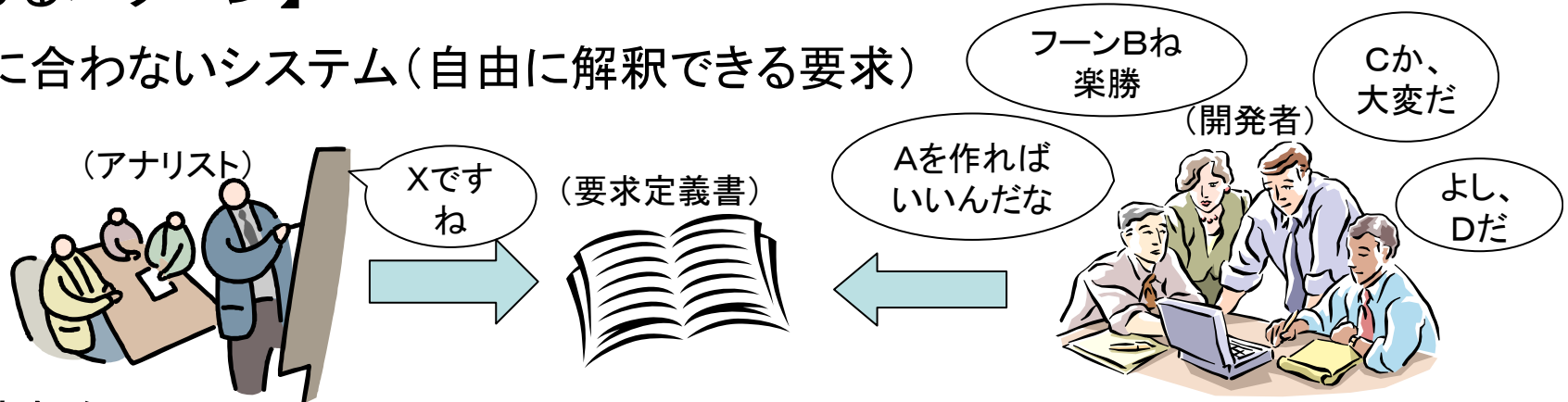
ソフトウェア開発者にとって
ソフトウェア要求＝「要件」

「要件定義」において必要な
のは顧客が望むものを明確
にすること→「要求」

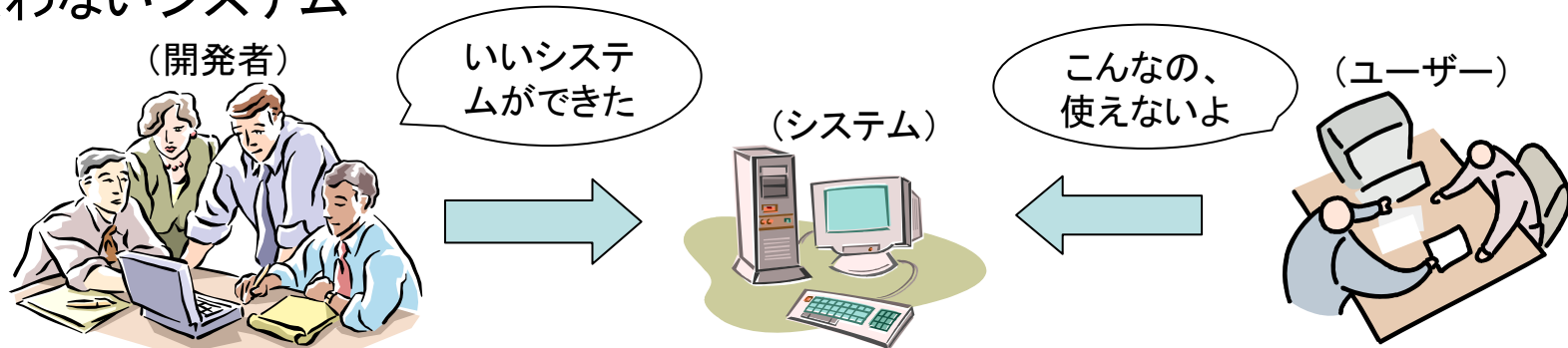


(2)要求定義の問題点 【よくあるパターン】

①要求に合わないシステム(自由に解釈できる要求)



②誰も使わないシステム



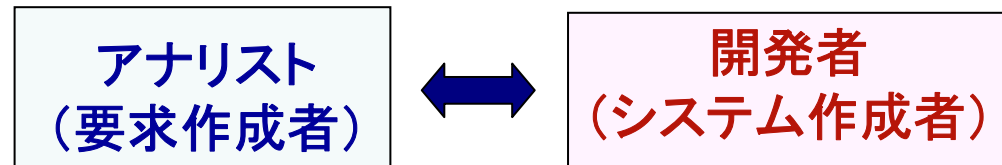
③目的に合わないシステム(現場の自己満足)



【よくあるパターン】を良く見ると

①要求に合わないシステム

開発者により解釈が異なる要求



②誰も使わないシステム

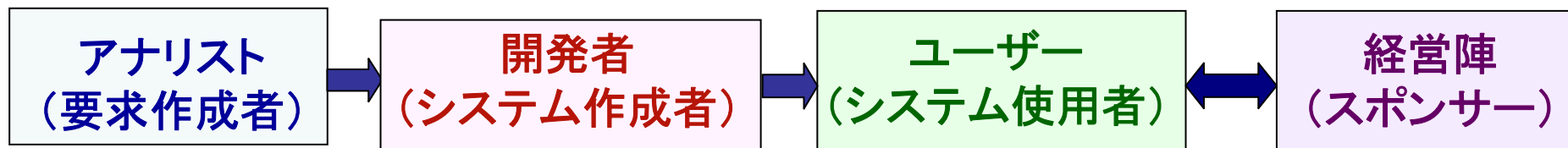
開発者には最高のシステムでもユーザーには使え(わ)ない



③目的に合わないシステム

現場の自己満足

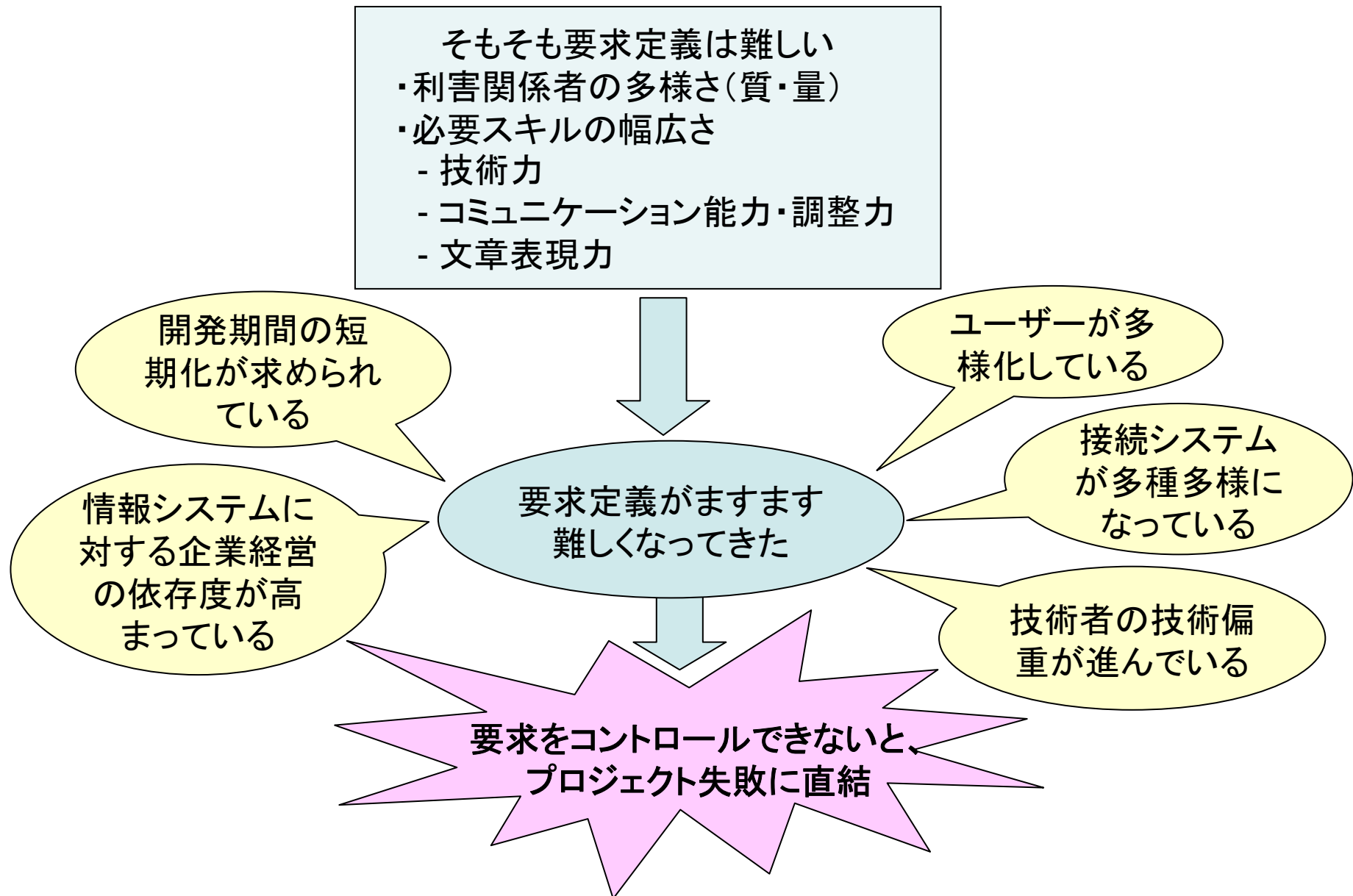
現場のユーザーの役には立っても経営課題を解決しない



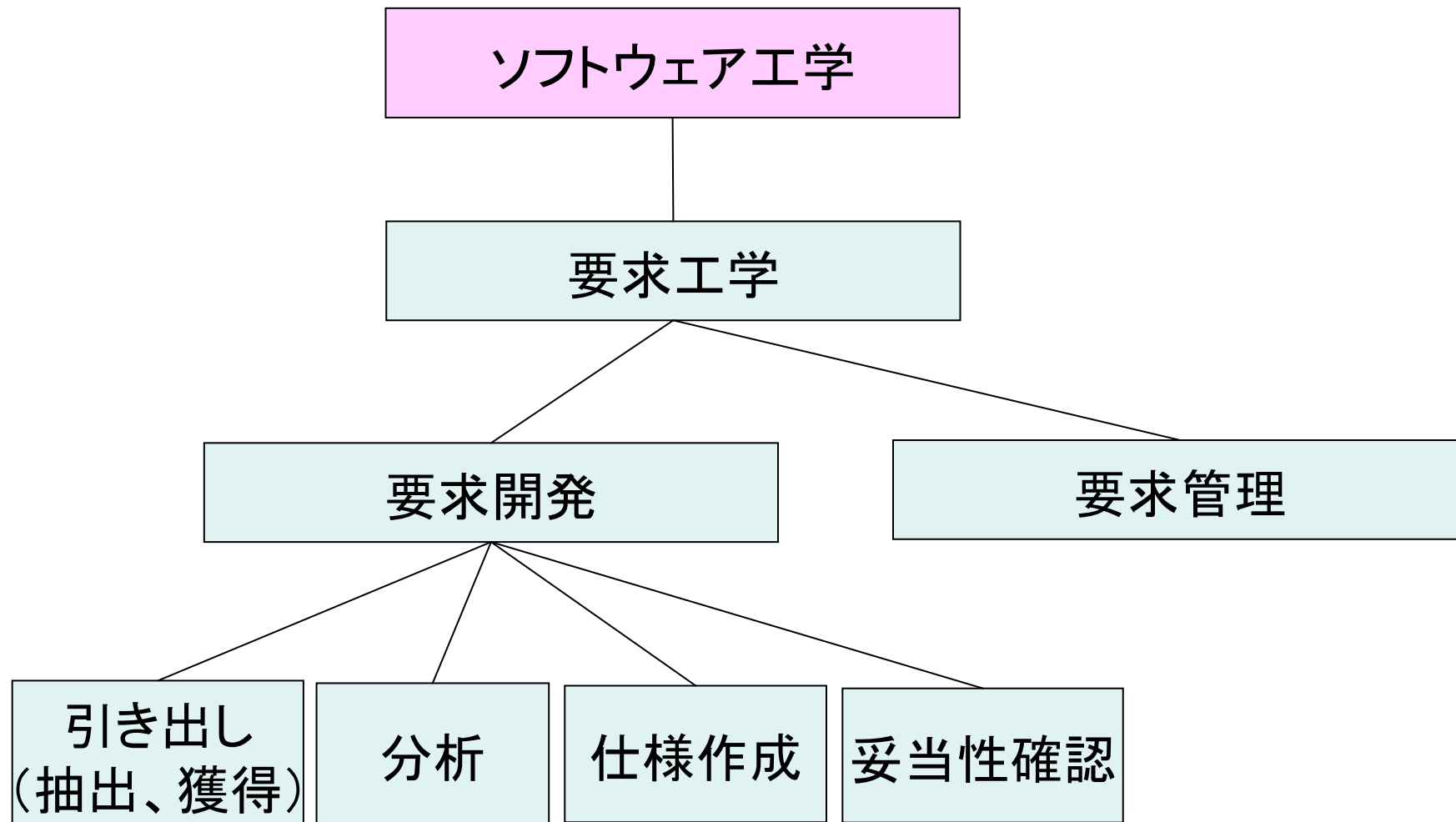
【プロジェクトの失敗につながる要因とその分類】

	要因	分類			備考
		PM	文書化	要求	
②	ユーザーを巻き込めない	○		○	体制作り、 コミュニケーション、 政治問題
②③	ステークホルダーの見落とし	○		○	
全体	甘いスケジュール	○		○	見積り、状況把握
③	要求の肥大化	○		○	変更管理、 スコープ管理
③	多すぎる要求変更	○		○	
①	あいまいな要求		○	○	内容の問題(②)、 表現の問題(①)
①	定義された要求の不足		○	○	
①	装飾過多の要求			○	本質の見落とし、 目的が不明？
③	目的に合わない要求			○	

【難易度が高まる要求定義の状況】

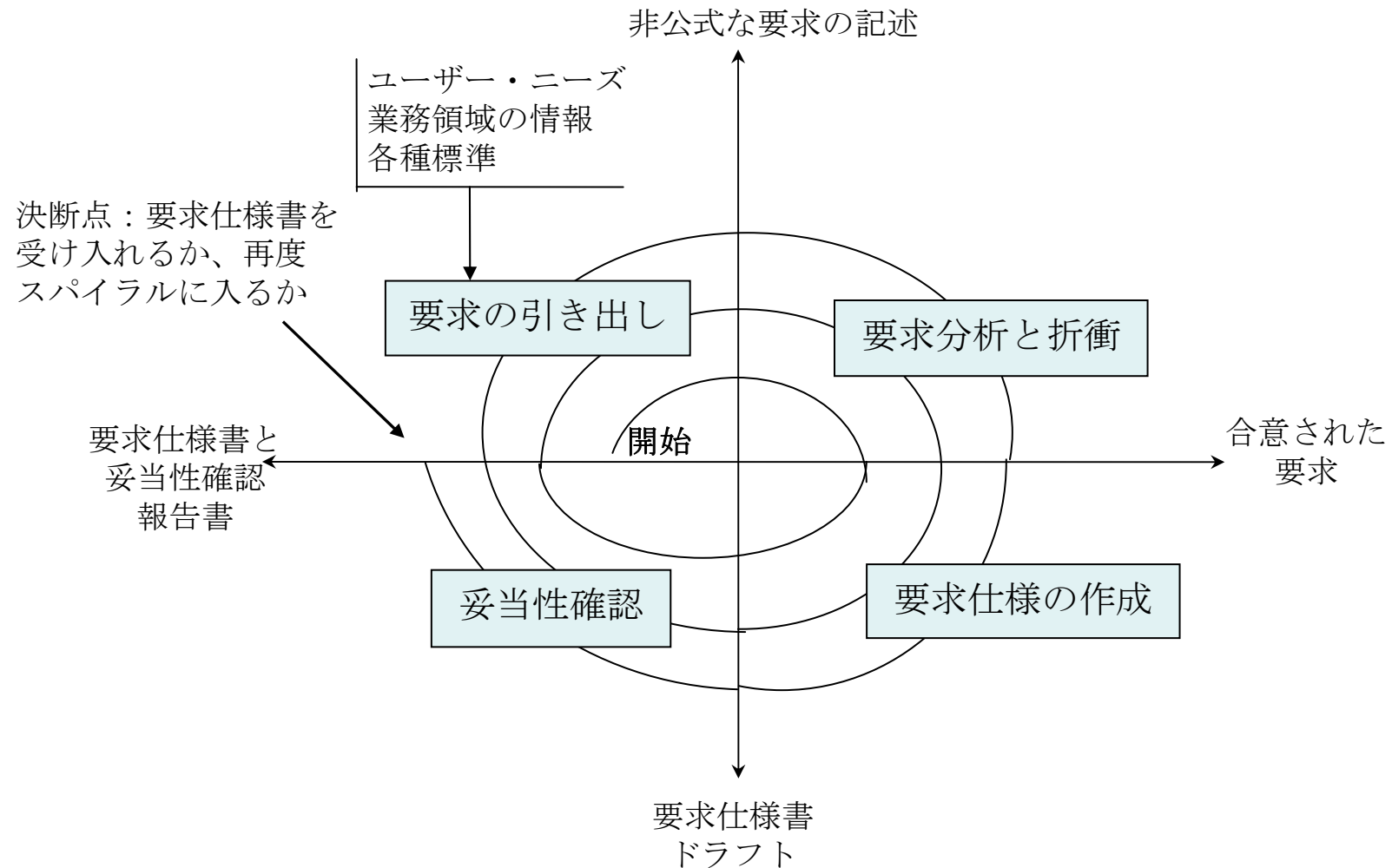


(3)ソフトウェア工学と要求工学



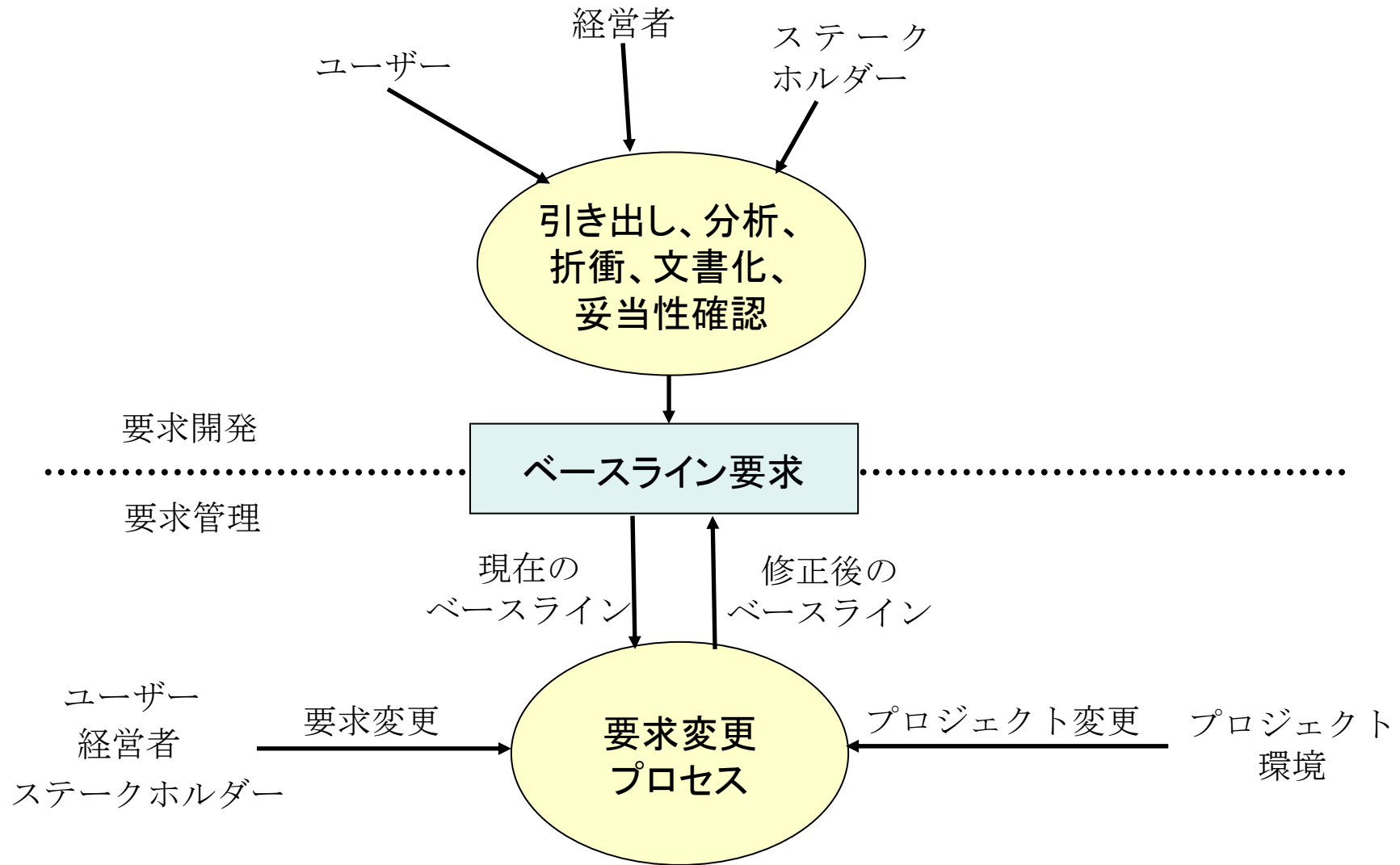
Karl E. Wiegers著「ソフトウェア要求」2003年、日経BPソフトプレス に一部加筆

【要求開発プロセスのスパイラル・モデル】



Guide to the Software Engineering Body of Knowledge, Trial Version. IEEE Computer Society Press
より渡部訳

【要求開発と要求管理の境界】

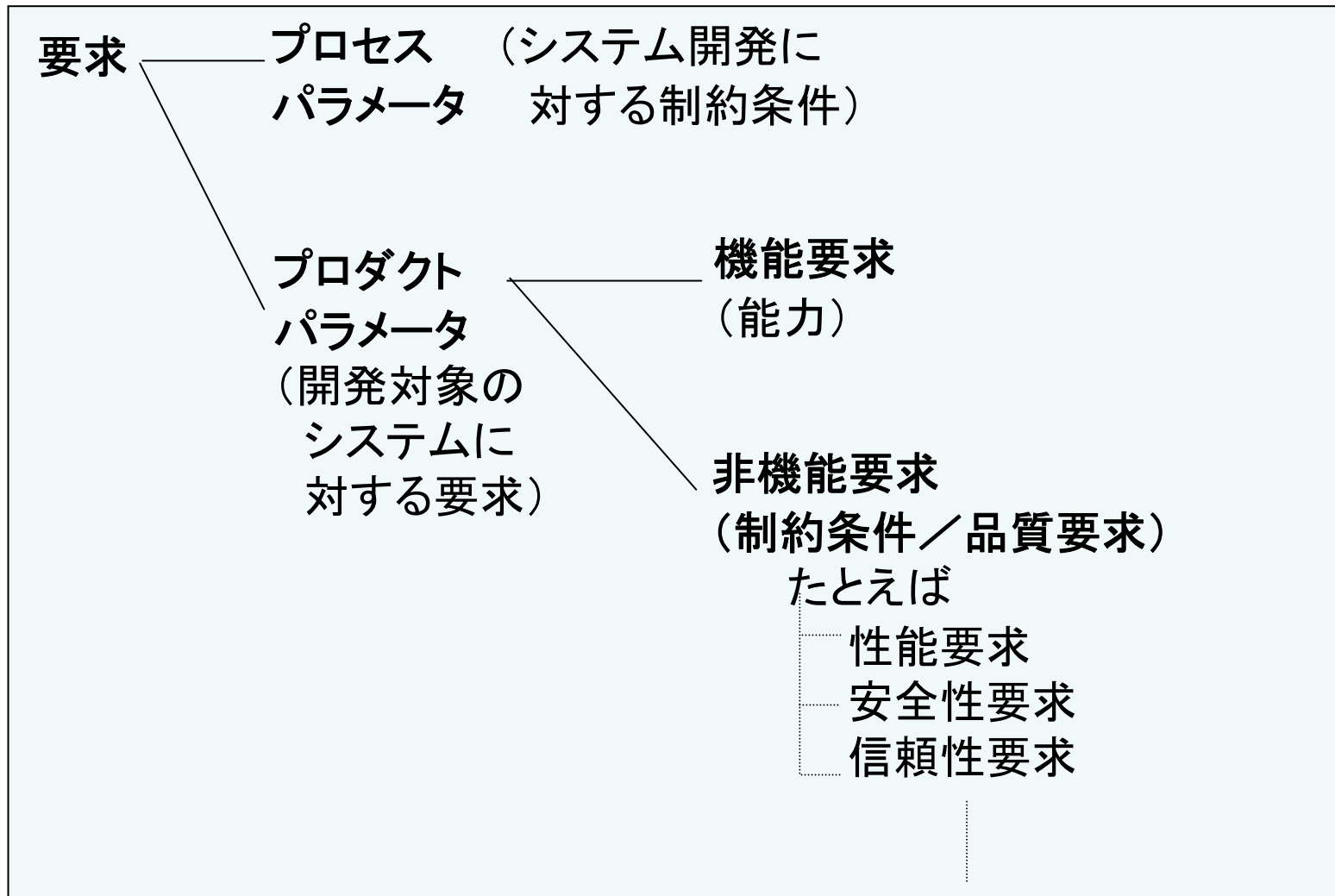


Karl E. Wiegers著「ソフトウェア要求」2003年、日経BPソフトプレス より一部修正加筆

2. 要求の体系

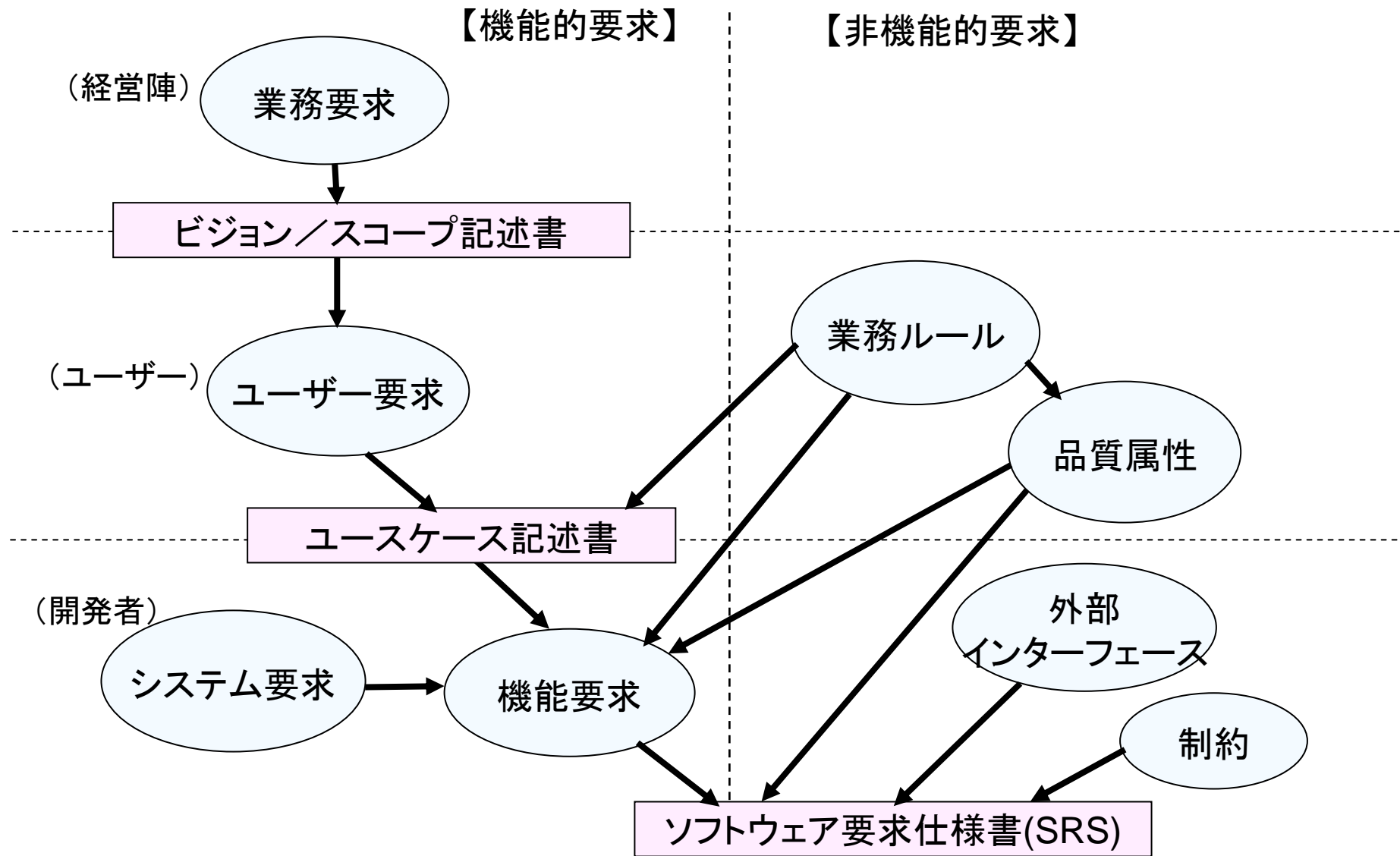
(1)全体図

① SWEBOKでの要求の体系



ソフトウェアエンジニアリング基礎知識体系 -SWEBOK- 松本吉弘監訳 オーム社 より

② Wiegersの要求の体系



出典: Karl E. Wiegers著「ソフトウェア要求」

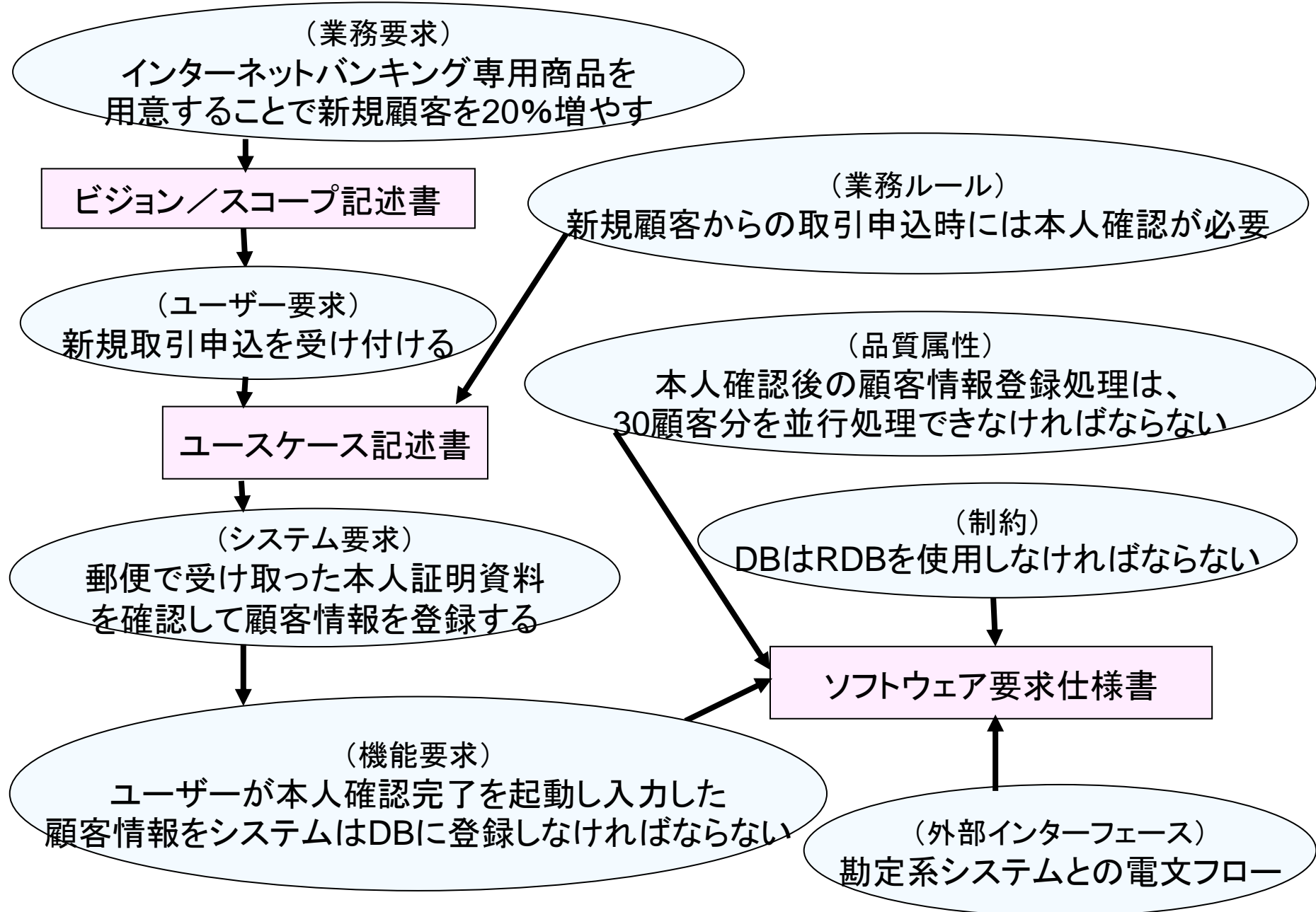
(2) 要求の種類(Wiegers)

要求情報	種類	レベル	まとめ先	内容
業務要求	機能	コンセプト	ビジョン／スコープ記述書	達成したい目的、経営課題
ユーザー要求		概要	ユースケース記述書	ユーザーがシステムを用いて行えること
システム要求		詳細	機能要求	サブシステムに分割する前のシステム全体の要求
機能要求		詳細	ソフトウェア要求仕様書	システムで実現すべきソフトウェア機能
業務ルール	非機能	概要	ユースケース記述書、機能要求、品質属性、制約	従わなければならないルール、システムの外側にある
品質属性		概要／詳細	機能要求、ソフトウェア要求仕様書	機能以外に必要とされる特性
外部インターフェース		詳細	ソフトウェア要求仕様書	システムと外部との間のインターフェース
制約		詳細	ソフトウェア要求仕様書	設計や実装の選択肢の制限となるもの

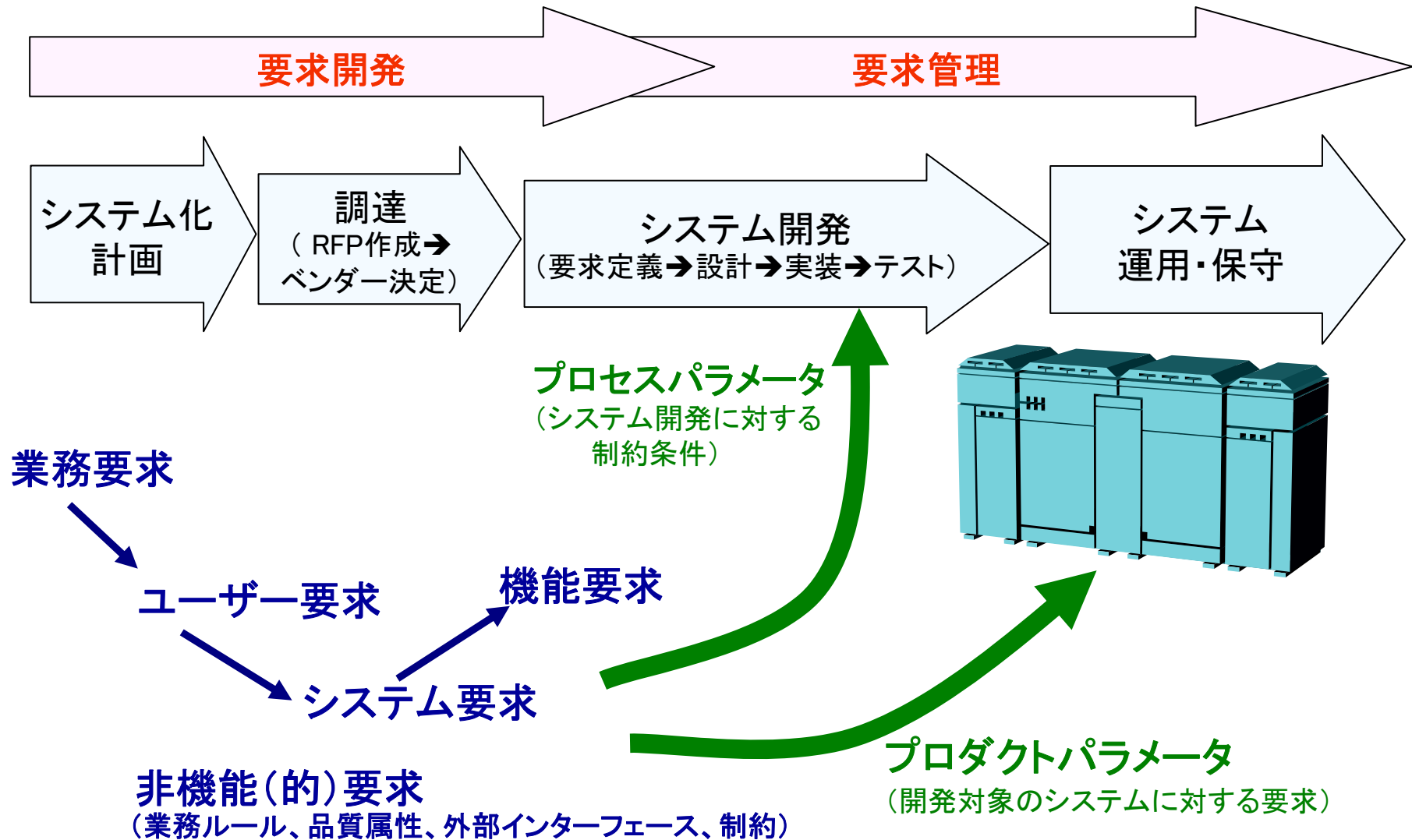
【要求文書の種類と内容】

要求文書	内容	備考
ビジョン／スコープ 記述書	業務要求を集めた文書、システムの戦略的なビジョンやスコープを記述する	「プロジェクト憲章」あるいは「マーケット要求記述書」となる場合もある
ユースケース 記述書	ユーザー要求をユースケースとして記述した文書	ユースケースに限らず、他の形式で表すこともある
ソフトウェア要求 仕様書	ソフトウェア要求を必要十分に記述した文書	IEEE標準830-1998にテンプレートあり

【各要求のサンプルイメージ】



(3)開発プロセスと要求

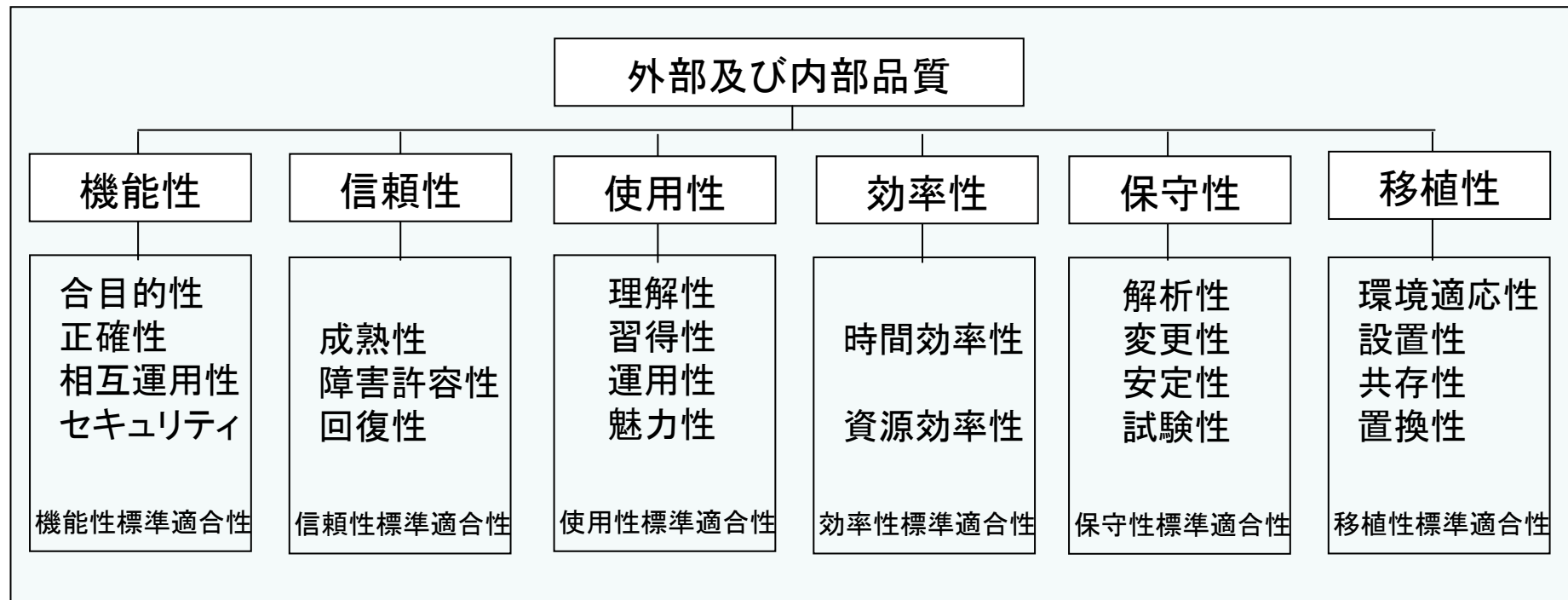


3. 良い要求とは

(1) 良い要求とは？

目的とするソフトウェア(システム)の品質を高くする要求？

JISの品質モデル(JIS X 0129-1:2003(ISO/IEC9129-1:2001より))



残りの要求

ほとんどが非機能要求の品質属性

【要求品質とは？】

要求品質 ≠ システム品質 (JISの品質モデル)

(紛らわしいので、以下では「要求品質」ということばは使用しない)

何で良い要求を作ろうとする試みが難しいのか？

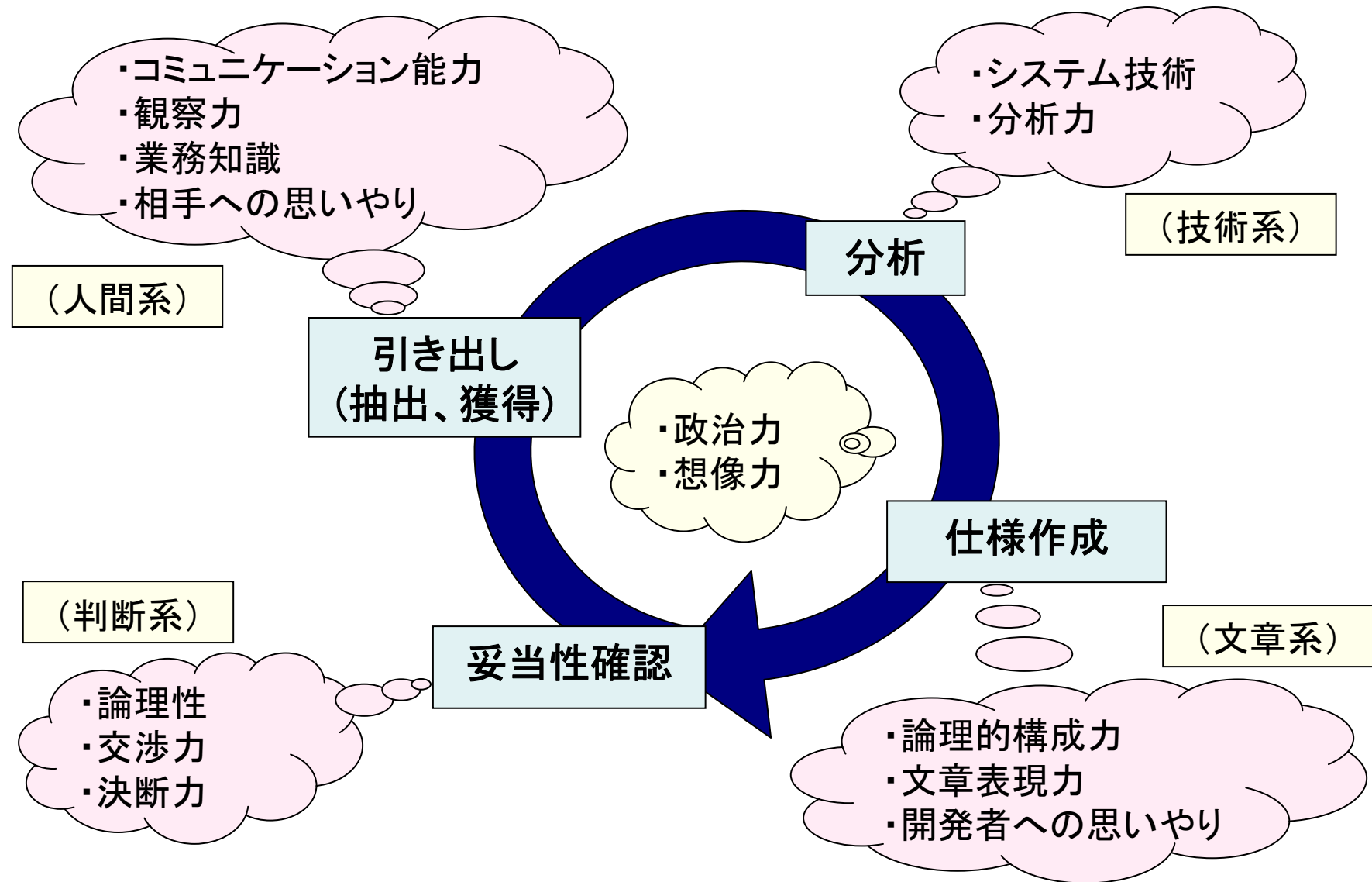
→ 要求開発プロセス内部で異なるスキルが要求されるから

→ → プロセスをブレークダウンして、必要スキルを整理する

→ レベルが違う要求が混在しているから

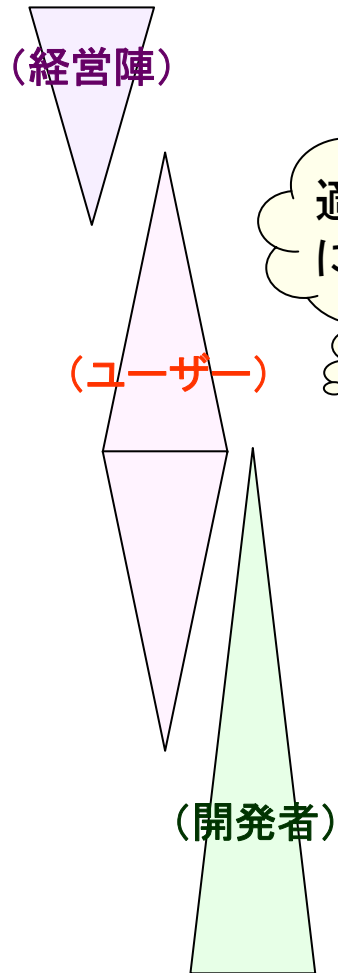
→ → 要求を構造化して、粒度をそろえる

【要求開発プロセスと必要スキル】

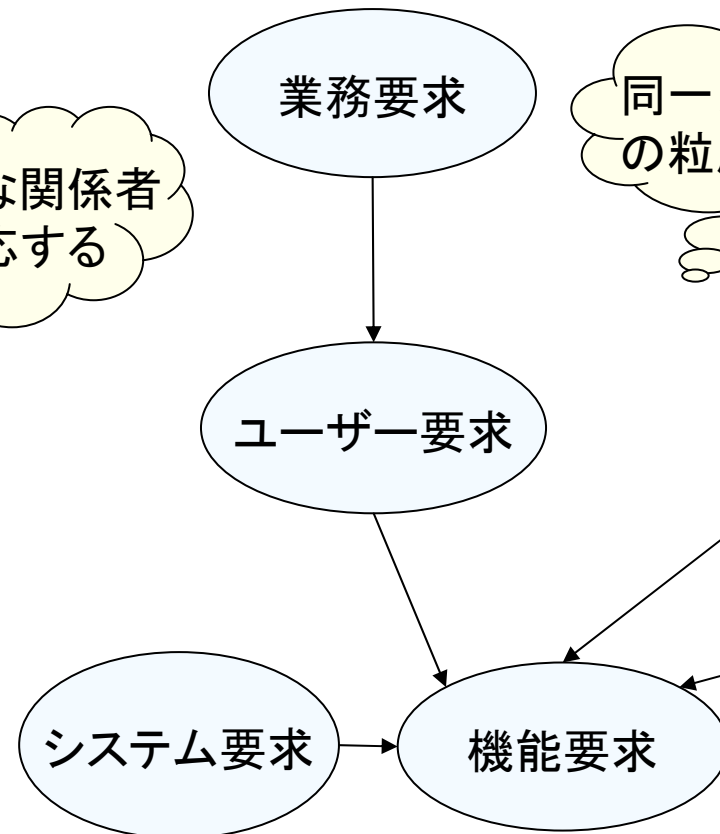


【要求の構造化】

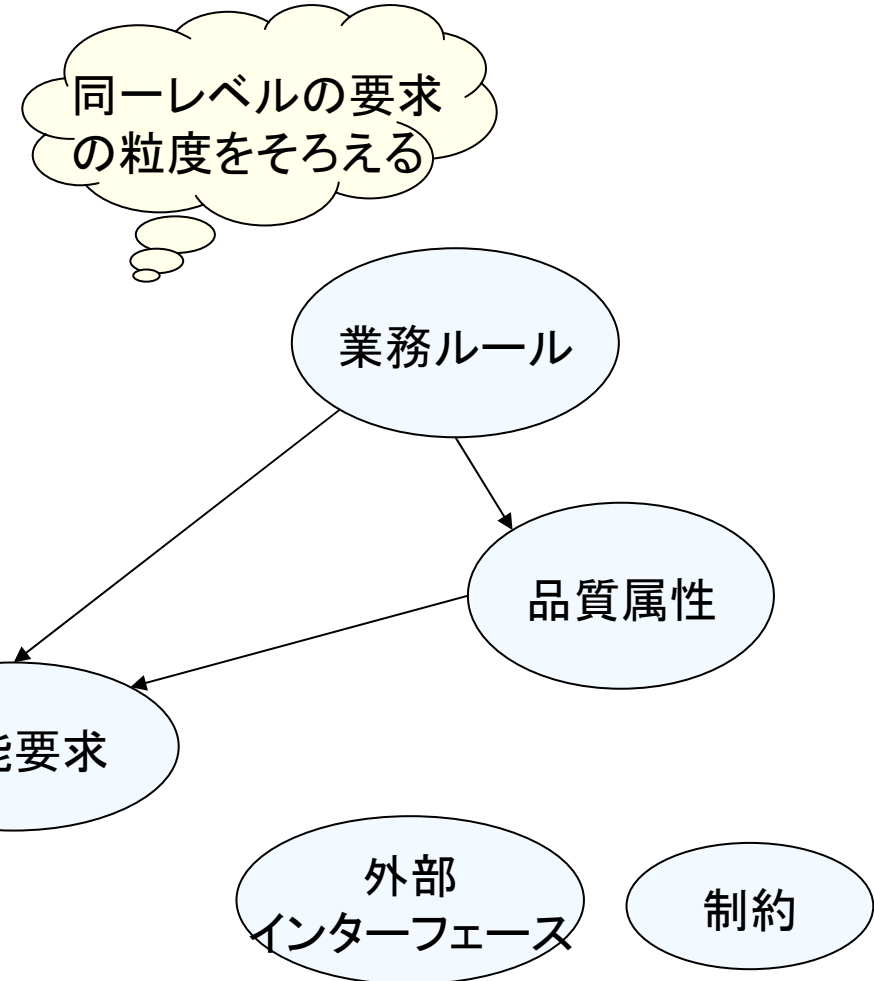
【関係者の関与】



【機能的要求】



【非機能的要求】



(2)優れた要求の特性

①優れた要求記述

- ・完全である
- ・正当である
- ・実現可能である
- ・必要である
- ・優先順位がついている
- ・あいまいでない
- ・検証可能である

②優れた要求仕様

- ・完全である
- ・整合性がある
- ・修正可能である
- ・追跡可能である

③指針

- ・用語は用語集に定義した通りに使う
- ・図表を用いて視覚的に表現する
- ・できる限り具体的なアクターを記述する
- ・できる限り定量的に表現する
- ・漠然とした要求は十分明確になるまで分解して詳細化する
- ・詳細度の整合性をとる。。。粒度
- ・要求記述を開始する前に記述ガイドを作り、書き手による違いを最小化する
- ・要求に設計を含めない、「どのように」解決するかではなく「何を」解決するかを記述する

(3)避けるべき表現(例)

- ・「適切な」「妥当な」→判断方法を説明する
- ・「少なくとも」「最大」→具体的な最大値と最小値を指定する
- ・「改善する」→特定の機能分野で何がどの程度変われば妥当な改善となるのか指定する
- ・「～してはならない」→要求は肯定的に記述し、システムがすべきことを表現する
- ・「十分な」→何がどれくらいあれば十分なのか明記する
- ・「その他」「など」→全項目を挙げる
- ・「より大きい」「より小さい」→両端の値が含まれるかどうか明確にする
- ・「ユーザーフレンドリーな」→使用ニーズと使用性の期待値を達成するシステム特性を記述する

【要求記述例】

- ・「システムはユーザーフレンドリーでなければならない」
→「訓練されたユーザーは、平均3分、最大5分で、注文受付処理を完了できなければならない」
- ・「システムは、毎日24時間100%使用可能でなければならない」
→「システムは、平日の午前8時から夜10時までの間は99.9%使用可能でなければならない」

受け入れ可能なリスクレベルで、チームの
設計・構築が進行するのに十分な要求を書くこと

参考:JUAS「非機能要求仕様定義ガイドライン」2008
(最後の参考文献参照)

(4)優れた要求を作成するには

- ・顧客と開発者のパートナーシップを確立する
- ・要求アナリストを育成・訓練する
 - － 特に「聞く」スキルが必要:アクティブ・リスニング
- ・要求工学を取り入れる:有益なプラクティスを利用する

(例)

- － 要求開発プロセスを明確化する
- － 機能だけではなく、ソフトウェア品質属性を意識する
(可用性、効率性、柔軟性(拡張性)、完全性、相互接続性、信頼性、堅牢性、使用性、保守性、移植性、再利用性、・・・)
- － ソフトウェア要求仕様書(SRS:Software Requirements Specification)の標準テンプレートを作成する

【顧客と開発者のパートナーシップ】

業務要求 ←—— 経営陣レベルの顧客、スポンサー

ユーザー要求 ←—— システムを使う顧客(ユーザー)

機能要求 ←—— 顧客と開発者

→ 優れた要求を作るためには顧客と開発者の協力と
コミュニケーションが必要

パートナーシップ

権利と責任を合意しておく必要がある

* 市販のパッケージソフト開発では

スポンサー顧客＝ユーザー

顧客代理人(マーケティング部門等)を考える

* アナリストはさまざまな立場の顧客が納得するように
調整する必要がある

【ソフトウェア要求に関する顧客の権利】

顧客には以下の権利がある

- ①アナリストに顧客が理解できることばで話してもらえる
- ②アナリストに業務とその目的について学んでもらえる
- ③アナリストにソフトウェア要求仕様書として記録してもらえる
- ④作成する全作業成果物について説明してもらえる
- ⑤アナリストや開発者が尊敬の念をもって接してくれる
- ⑥要求や開発のためのアイデアや代替案を提供してもらえる
- ⑦使いやすいシステムにしてもらえる
- ⑧再利用のために要求を調整する機会を与えてもらえる
- ⑨変更費用の正直な見積りを受け取ることができる
- ⑩機能と品質のニーズを満たすシステムを受け取れる

Karl E. Wiegers著「ソフトウェア要求」2003年、日経BPソフトプレス の表現を一部修正

【ソフトウェア要求に関する顧客の責任】

顧客には以下の責任がある

- ①業務についてアナリストや開発者を教育しなければならない
- ②要求を提供し明確化するために時間を費やさなければならない
- ③要求について明確かつ正確でなければならない
- ④タイムリーに決断を下さなければならない
- ⑤費用と実現可能性に関する開発者の評価を尊重しなければならない
- ⑥要求に優先順位をつけなければならない
- ⑦要求をレビューし、プロトタイプを評価しなければならない
- ⑧要求変更は即時に伝えなければならない
- ⑨プロジェクトの変更プロセスに従わなければならない
- ⑩アナリストが用いる要求工学プロセスを尊重しなければならない

Karl E. Wiegers著「ソフトウェア要求」2003年、日経BPソフトプレス の表現を一部修正

4. 有益なプラクティス

(1)要求開発のプラクティス

分類	プラクティス	
引き出し	<ul style="list-style-type: none"> ・要求開発プロセスを定義する ・ユーザー種類(クラス)を明確にする ・フォーカスグループを設置する ・システムイベントと応答を明確にする ・工作中的のユーザーを観察する ・要求を再利用する 	<ul style="list-style-type: none"> ・ビジョン/スコープ記述書を作成する ・それぞれのユーザークラスからキーパーソンを選ぶ ・ユースケースを明確にする ・要求引き出しワークショップを開く ・問題報告書を調べる
分析	<ul style="list-style-type: none"> ・コンテキスト図を描く ・実現可能性を分析する ・要求をモデル化する ・要求をサブシステムに割り当てる 	<ul style="list-style-type: none"> ・プロトタイプを作成する ・要求に優先順位をつける ・データディクショナリを作る ・品質機能展開を実行する
仕様作成	<ul style="list-style-type: none"> ・ソフトウェア要求仕様書のテンプレートを選ぶ ・各要求に固有なラベルをつける 	<ul style="list-style-type: none"> ・要求の発生源を明確にする ・業務ルールを記録する ・品質属性を明記する
妥当性確認	<ul style="list-style-type: none"> ・要求記述をインスペクションする ・受け入れ基準を定義する 	<ul style="list-style-type: none"> ・テストケースを作成し、モデルや要求をテストする

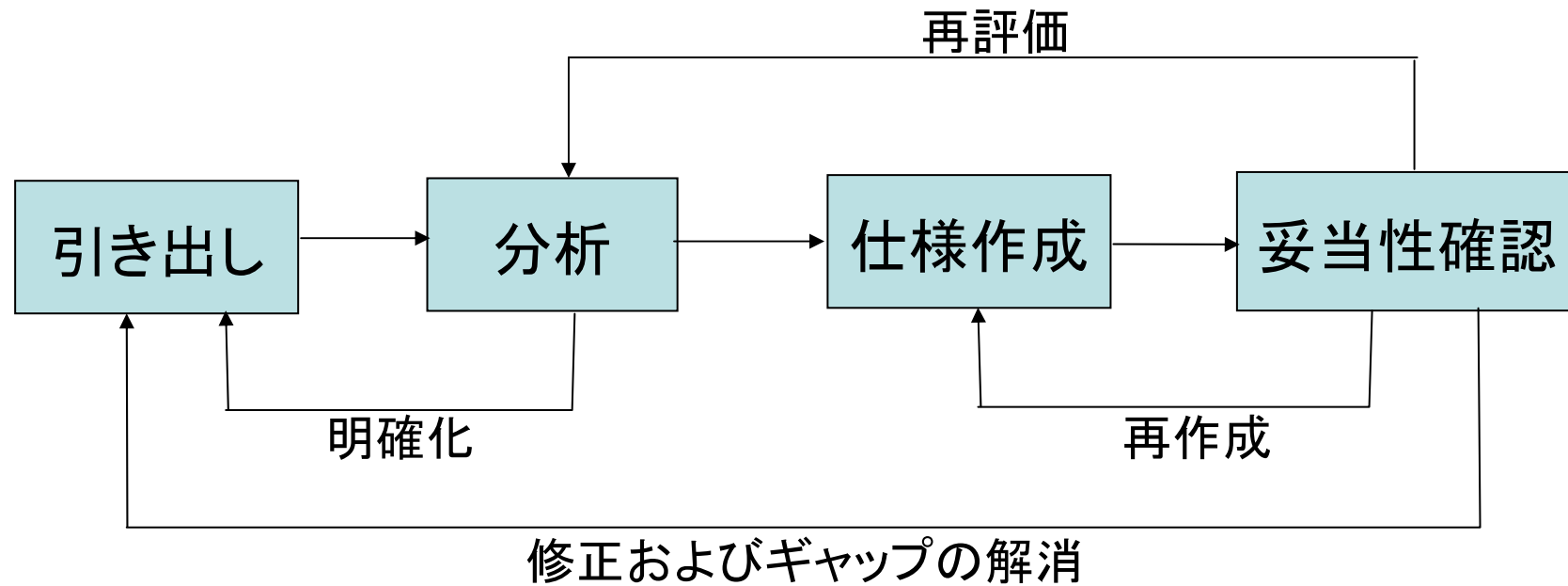
出典: Karl E. Wiegers著「ソフトウェア要求」より一部修正加筆

(2)要求管理とその他のプラクティス

分類	プラクティス	
要求管理	<ul style="list-style-type: none"> ・変更管理プロセスを定義する ・変更の影響分析を実行する ・変更履歴を保守する ・要求の不安定度を測る ・要求の追跡マトリックスを作成する 	<ul style="list-style-type: none"> ・変更管理委員会を設置する ・要求のベースラインを決め、バージョンを管理する ・要求ステータスを追跡記録する ・要求管理ツールを使用する
プロジェクト管理	<ul style="list-style-type: none"> ・適切な開発ライフサイクルを選ぶ ・要求変更時は約束を再折衝する ・要求にかかる作業工数を追跡記録する 	<ul style="list-style-type: none"> ・要求をベースにしてプロジェクト計画を立てる ・要求リスクを管理する ・過去の教訓を再検討する
知識	<ul style="list-style-type: none"> ・要求アナリストを訓練する ・開発者に業務の研修をする 	<ul style="list-style-type: none"> ・ユーザーの代表と管理者に要求に関する教育をする ・用語集を作る

出典: Karl E. Wiegers著「ソフトウェア要求」 より一部修正加筆

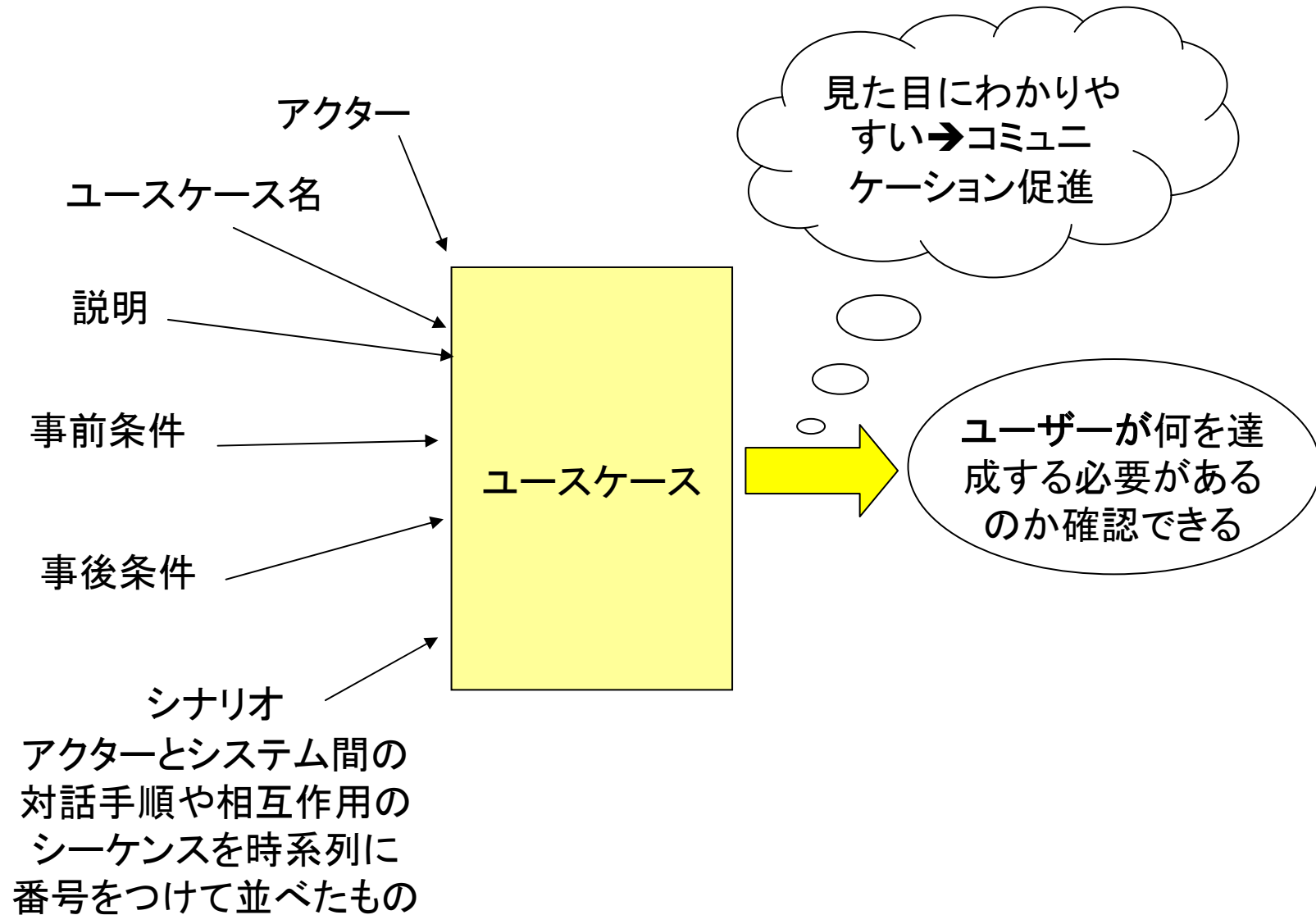
【要求開発プロセスの定義】



リニアな一本道ではなく、実際はインクリメンタルに繰り返す

Karl E. Wieggers著「ソフトウェア要求」2003年、日経BPソフトプレス より

【ユースケース作成(シナリオ分析)】



【ソフトウェア要求仕様書テンプレート例】

1 はじめに

- 1. 1 目的
- 1. 2 文書の規約
- 1. 3 対象読者と読み方の提言
- 1. 4 プロジェクト・スコープ
- 1. 5 参考文献

2. 概要

- 2. 1 背景
- 2. 2 業務機能
- 2. 3 ユーザークラスとその特性
- 2. 4 稼働環境
- 2. 5 設計と実装の制約
- 2. 6 ユーザー文書
- 2. 7 仮定と依存性

3 システム特性

- 3. X システム特性X
 - 3. X. 1 説明と優先順位
 - 3. X. 2 入力／応答シーケンス
 - 3. X. 3 機能要求

4. 外部インターフェース

- 4. 1 ユーザーインターフェース
- 4. 2 ハードウェアインターフェース
- 4. 3 ソフトウェアインターフェース
- 4. 4 通信インターフェース

5. その他非機能要求

- 5. 1 性能要求
- 5. 2 安全性要求
- 5. 3 セキュリティ要求
- 5. 4 ソフトウェア品質属性

6. その他の要求

付録A:用語集

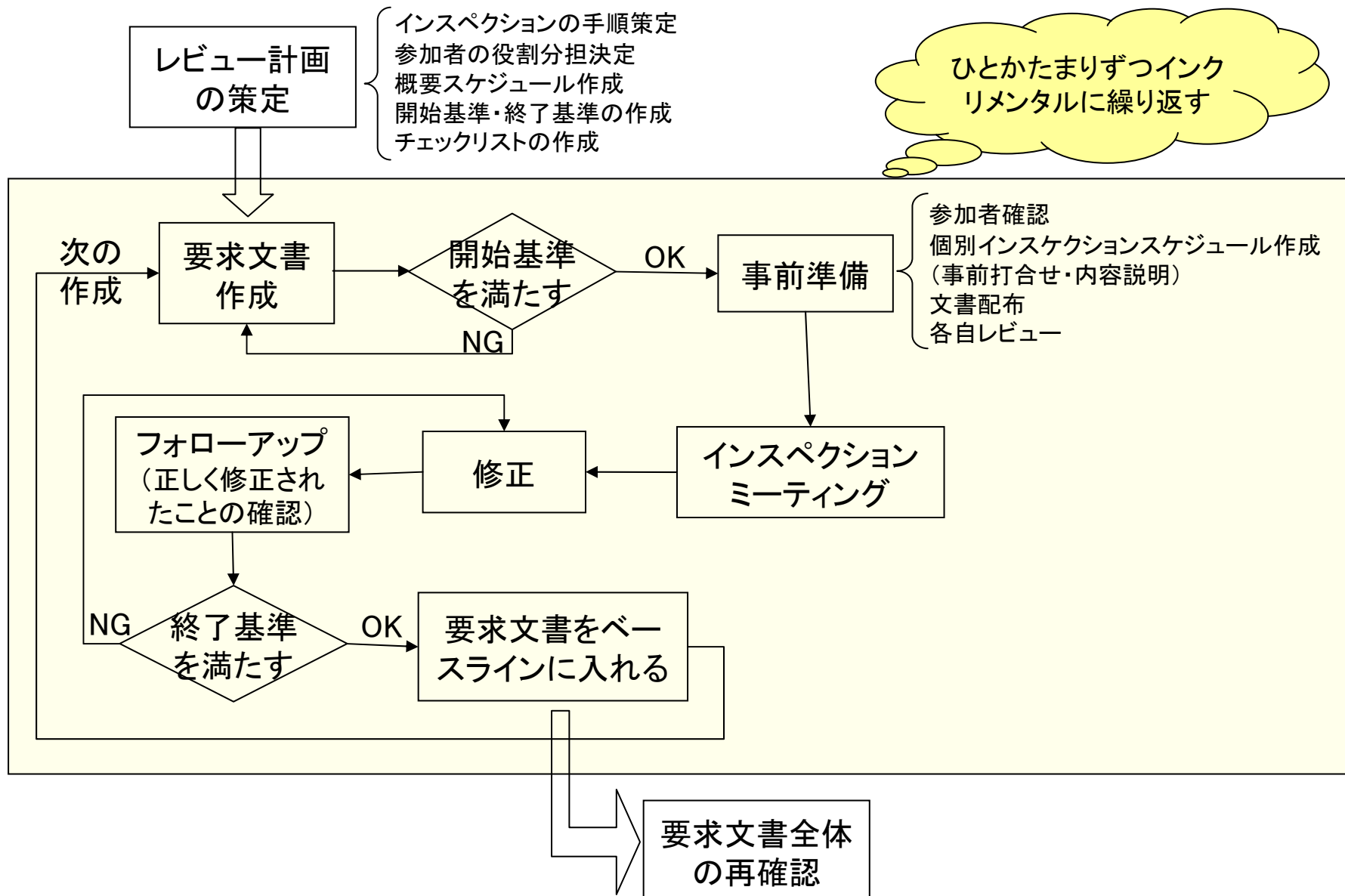
付録B:分析モデル

付録C:問題点一覧

付録D:保留項目一覧

出典:Karl E. Wiegers著「ソフトウェア要求」に一部加筆

【インスペクションプロセス】



参考文献

<http://www.swebok.org>

[日本語で読める要求の参考文献]

- ・「ソフトウェア要求」 Karl E. Wiegers 著 渡部洋子 監訳 日経BPソフトプレス 2003
- ・ソフトウェアテクノロジーシリーズ9「要求工学」 大西淳・郷健太郎 著 共立出版 2002
- ・「要求仕様の探検学－設計に先立つ品質の作り込み」 Donald C. Gause, Gerald M. Weinberg著 黒田純一郎 監訳 柳川志津子 訳 共立出版 1993
- ・「ソフトウェア要求管理－新世代の統一アプローチ」 Dean Leffingwell, Don Widrig著 石塚圭樹・荒川三枝子 監訳、日本ラショナルソフトウェア株式会社 訳 ピアソン・エデュケーション 2002
- ・「要求定義工学入門」 Pericles Loucopoulos, Vassilios Karakostas 著 富野壽 監訳 共立出版 1997
- ・「要求定義工学プラクティスガイド」 Ian Sommerville, Pete Sawyers 著 富野壽 監訳 共立出版 2000
- ・「要件プロセス完全修得法」 Suzanne Robertson, James Robertson 著 苅部英司 訳 三元社 2002
- ・「実践ソフトウェア工学」 Roger S. Pressman 著 飯塚悦功・西康晴 監訳 株式会社東芝S&S研究所訳 日科技連 2000

- ・「ソフトウェアエンジニアリング基礎知識体系 -SWEBOK- 」IEEE Computer Society
オーム社 2003
- ・「要求仕様定義ガイドライン」
～UVC (User Vender Collaboration) 研究プロジェクト報告書～
日本情報システム・ユーザー協会(JUAS) 2007
- ・検収フェーズのモデル取引・整備報告書「非機能要求仕様定義ガイドライン」
UVC (User Vender Collaboration) 研究プロジェクトⅡ 報告書
日本情報システム・ユーザー協会(JUAS) 2008