ソフトウェア工学入門

FU14 ソフトウェア工学概論

第1回

吉岡 廉太郎

今日の内容

- ソフトウェア工学とは何か
  - 主要な要素と対象範囲
- 要求定義工程
  - ユースケース分析

対象となるソフトウェア

https://news.microsoft.com/ja-jp/windows_10_anniversary_update_desktop/

https://www.eki-net.com/top/jrticket/about/

http://www.volvocars.com/jp/cars/concept-cars/concept26

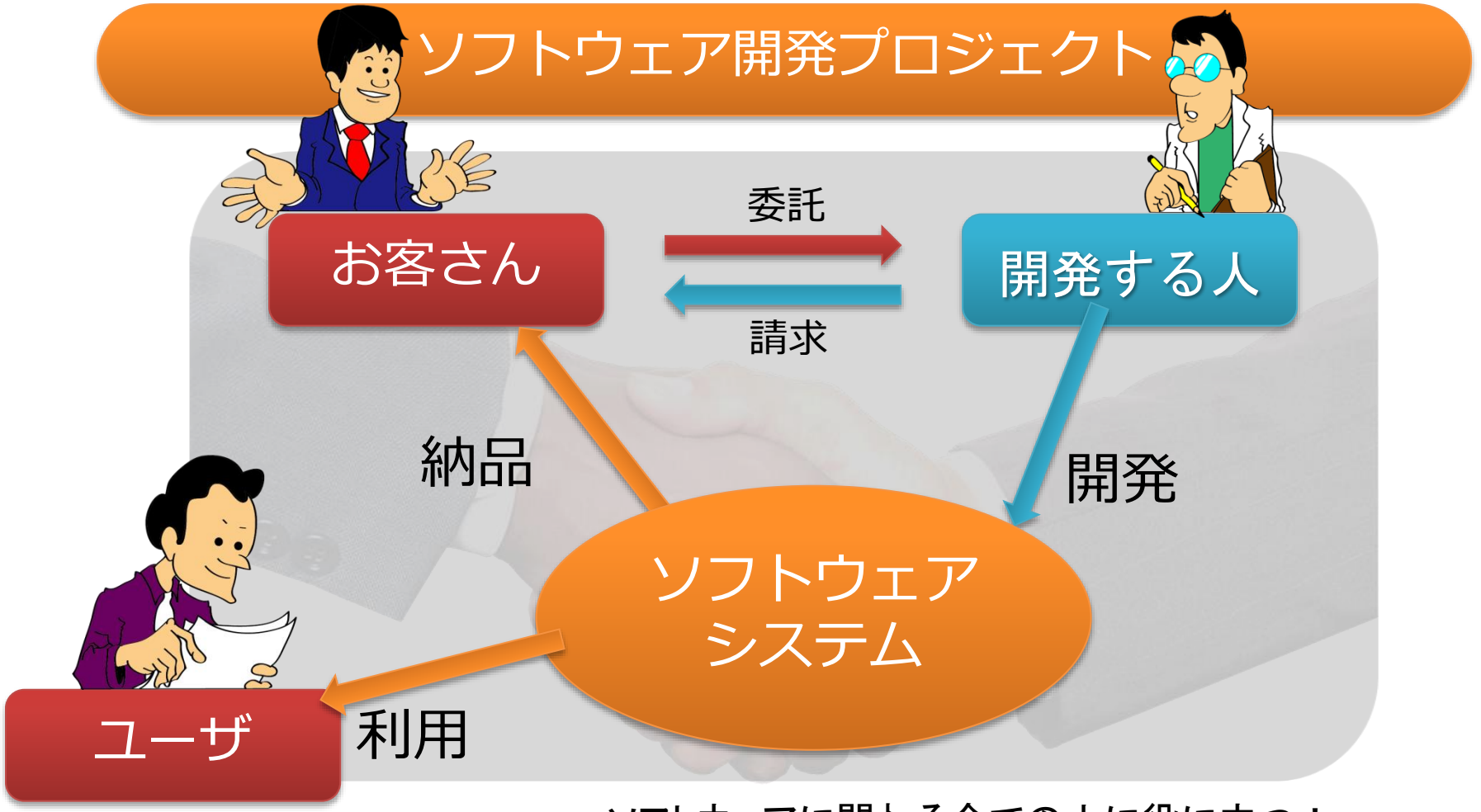
http://www.jre-water.com/pdf/100810jisedai-jihanki.pdf

ソフトウェア開発の将来は挑戦であふれている

ビジネスのスピードが加速	短時間で高い品質を実現する
価値の変化	モノは安くなり開発・管理コストが増加
コンピュータの処理能力向上	ソフトウェアに対する要求が複雑化
ネットワークの高速化	新しい技術や方法に柔軟に対応
ユーザインタフェースの進化	様々な機器、利用の場面、ユーザに対応

ソフトウェア工学は誰の役に立つのか？

プロジェクトの利害関係者

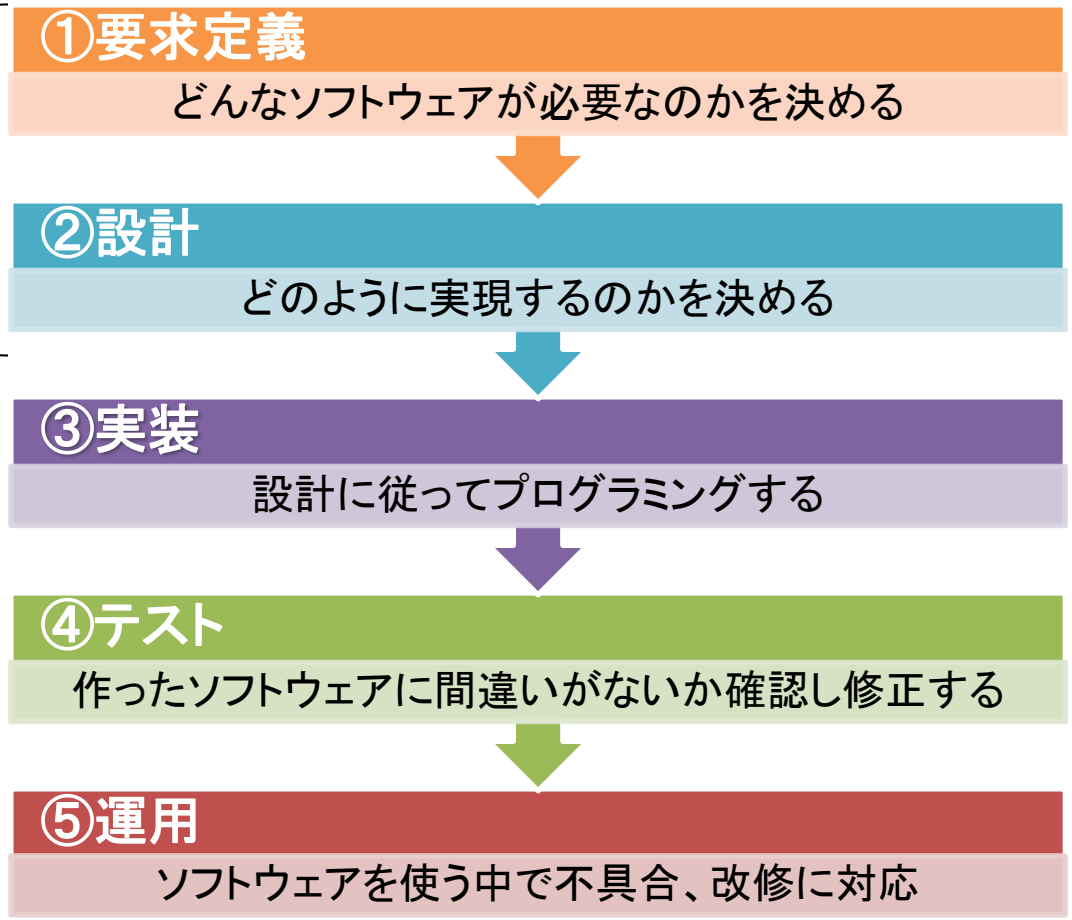


…ソフトウェアに関わる全ての人に役に立つ！

ソフトウェア開発の基本プロセス

上流工程

- どのような目的でどのようなソフトウェアが必要か明確にする
- ソフトウェアを構成するコンポーネントやそのやりとりの方法などを決める

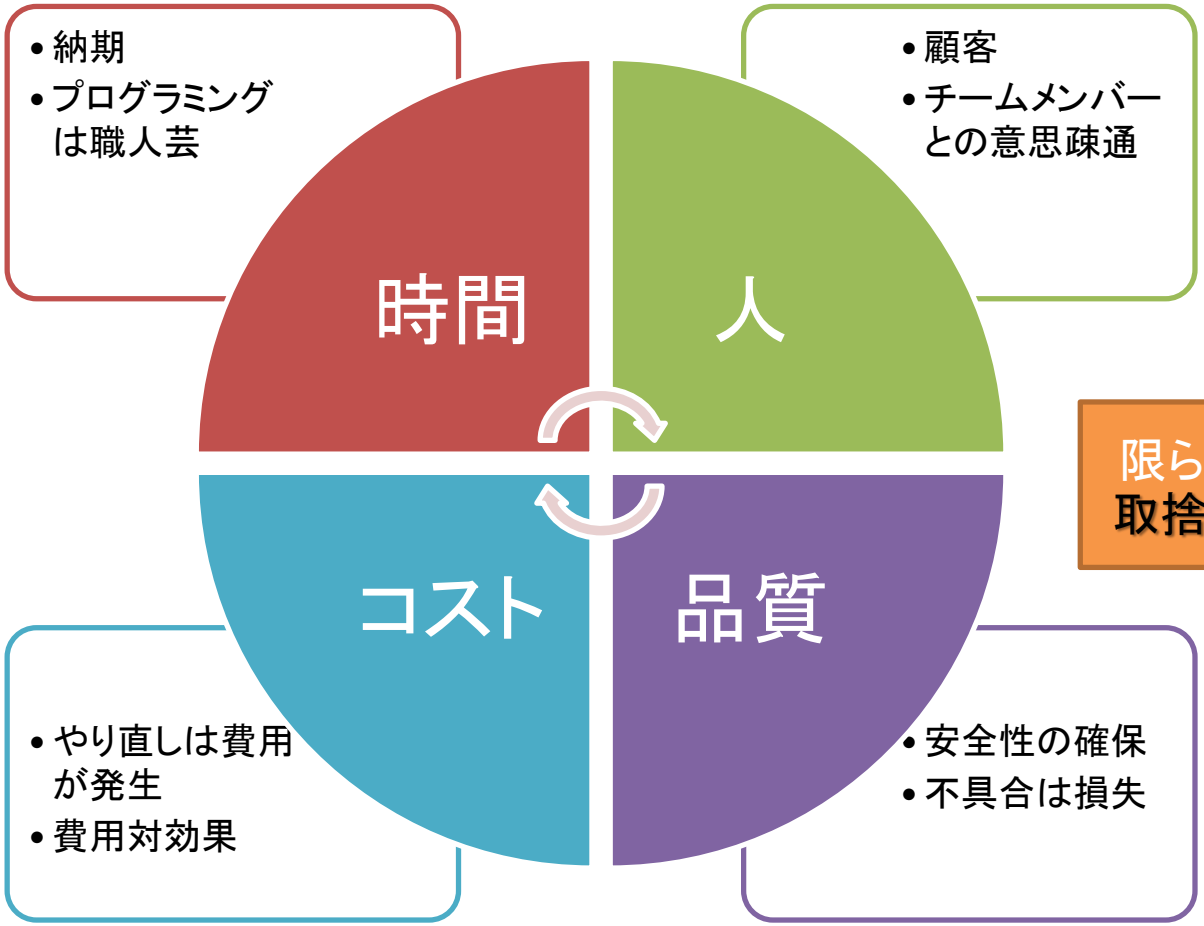


開発チームの構成

ほとんどのソフトウェアはチームで開発する

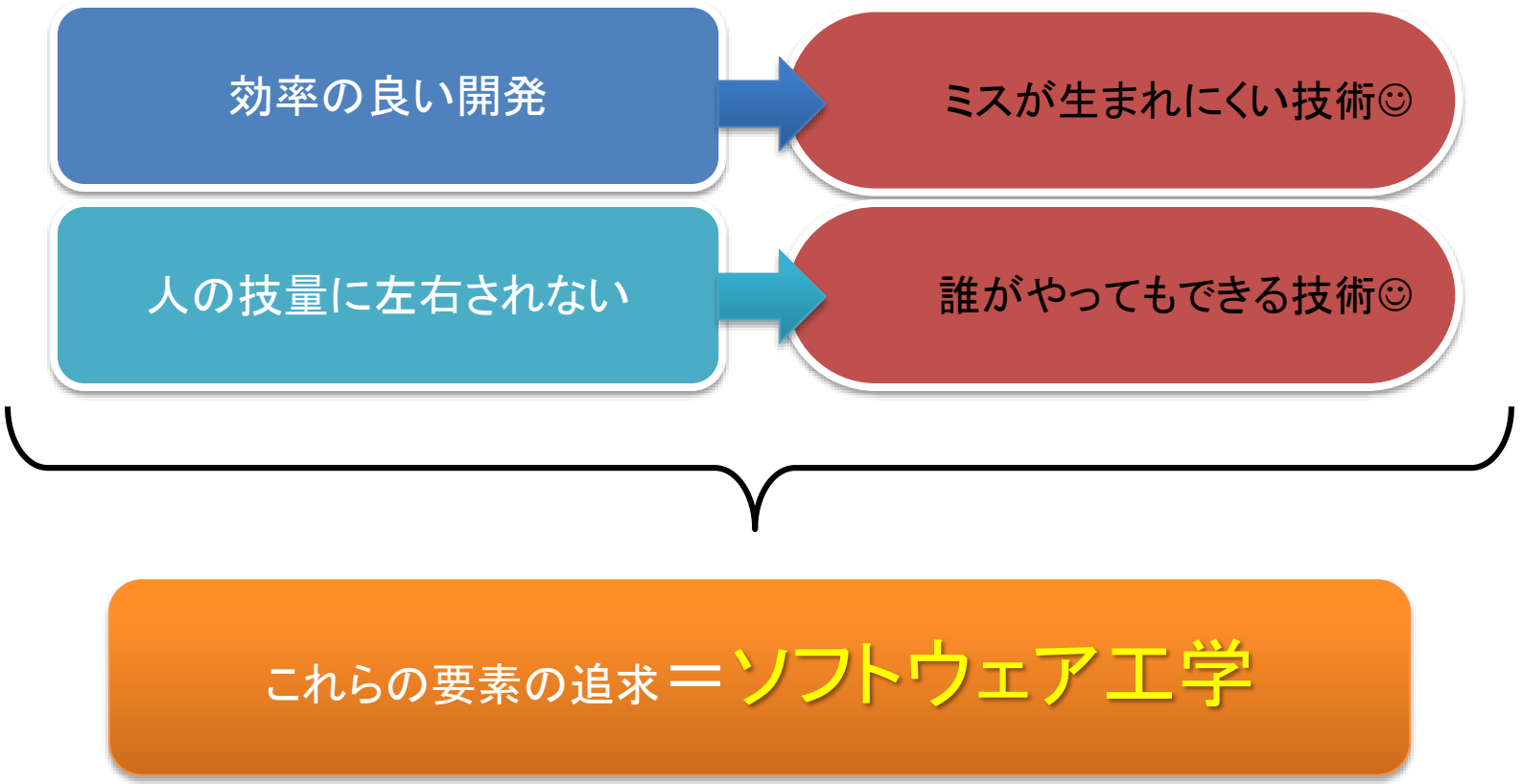
メンバーの連携が成功のカギ！

ソフトウェア開発はバランスが重要！

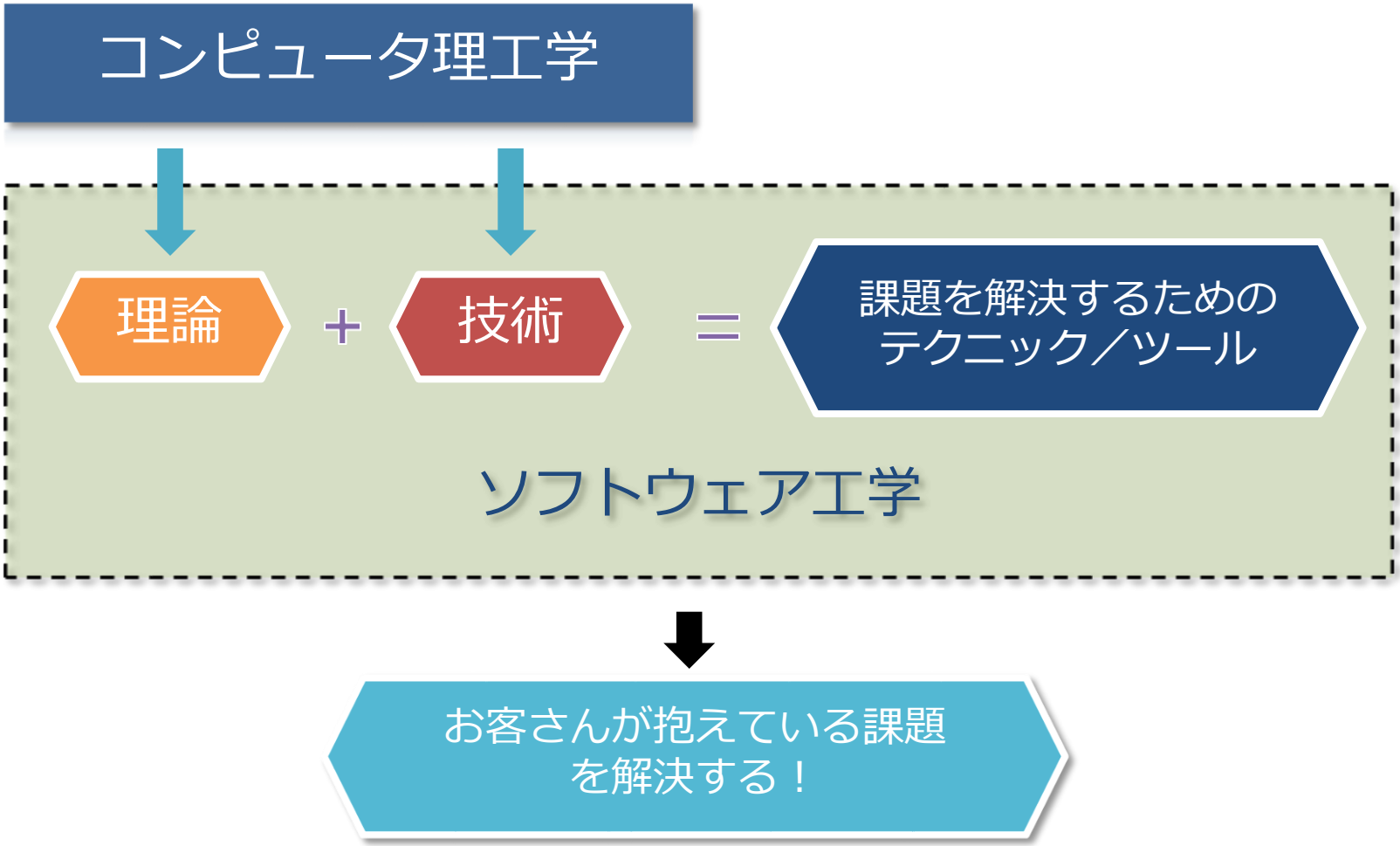


限られた資源＝**取捨選択**が大切

品質を確保するために必要なこと



コンピュータ理工学との関係



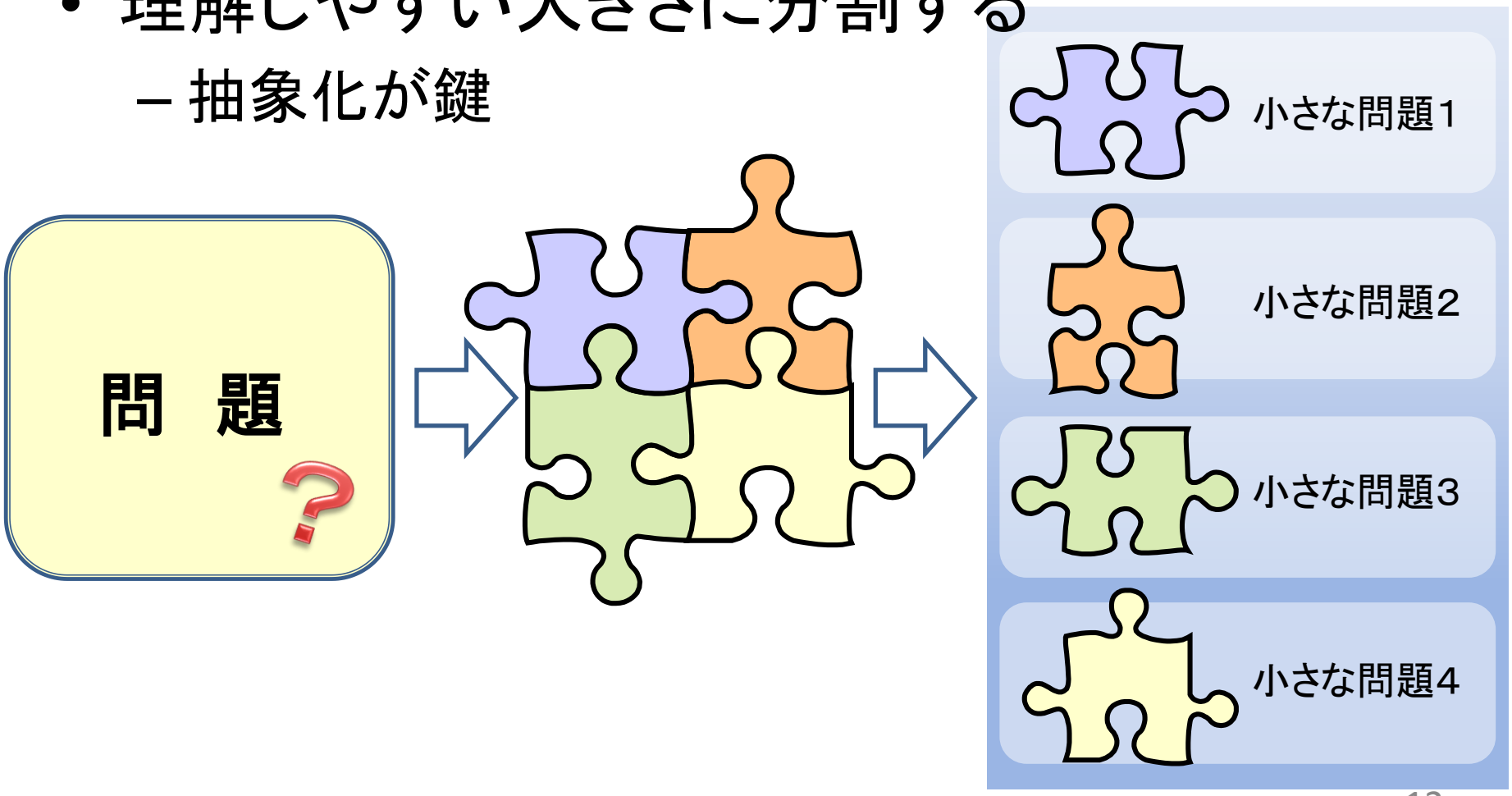
課題を解決すること

- ソフトウェアを通して人が抱える課題を解決する
- ソフトウェア開発＝課題の解決
- 与えられた課題を解決するとは
  - 課題を理解する
  - 解決策を構築する

大規模な課題、
複雑な課題、
…

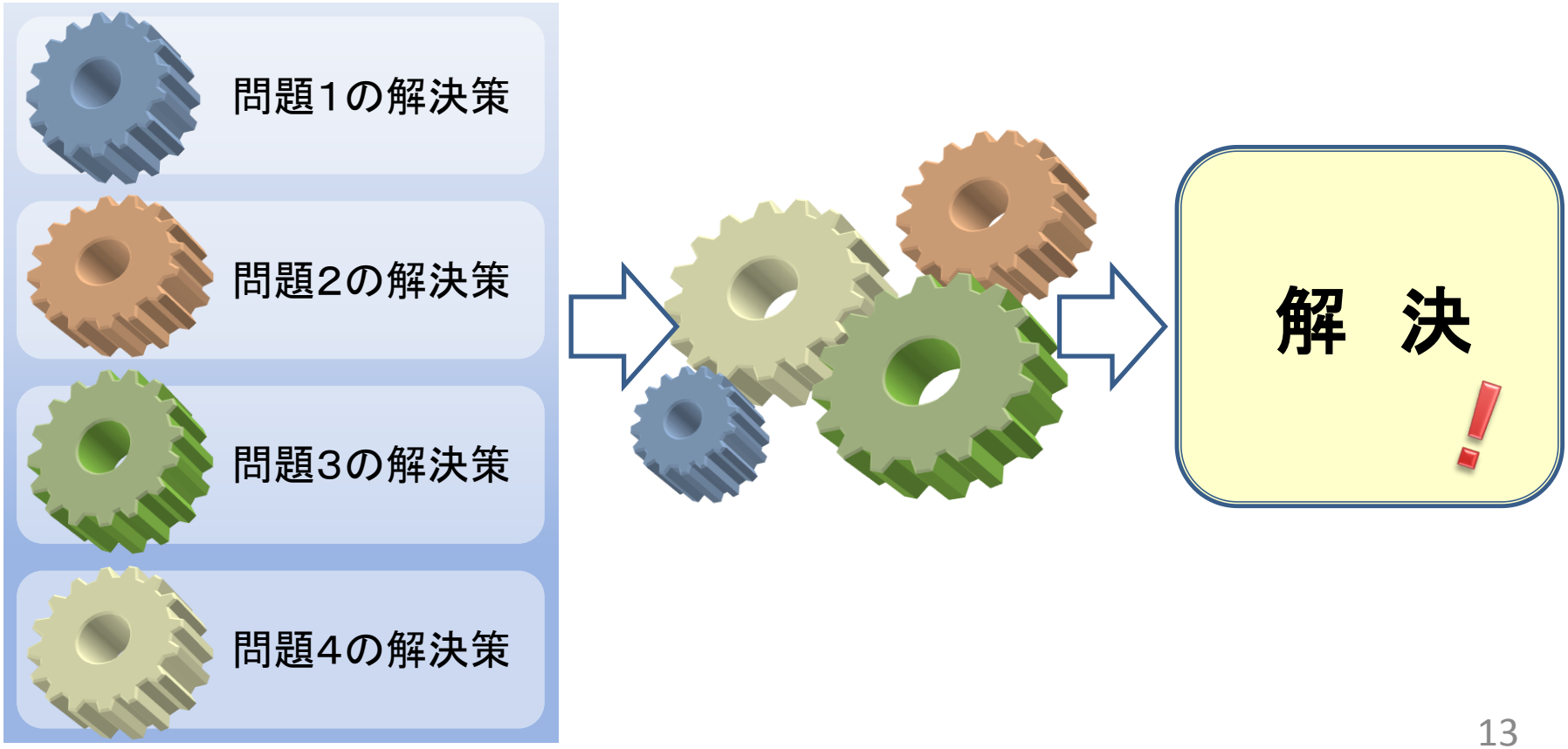
課題を解決する:問題の分割

- 理解しやすい大きさに分割する
  - 抽象化が鍵



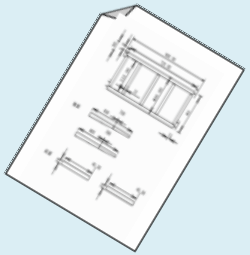
課題を解決する:統合

- 個別の解決策を組み合わせる
  - くっつけるときに問題が起きやすい



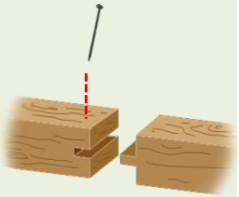
課題を解決する知恵

- テクニック
  - 形式的な手順のこと
  - ツールには依存しない、何かを達成するためのレシピ

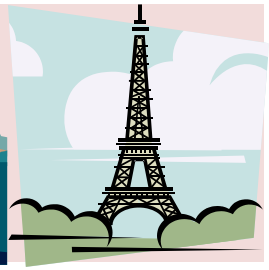


- ツール
  - 目的の達成を容易にする道具や自動化されたシステム

- プロージージャ
  - 成果物を得るためのツールとテクニックの組み合わせ方



- パラダイム
  - 成果物を構築するための哲学や取り組み方



要求定義工程

- ソフトウェア開発の最初の工程
  - 開発プロジェクトの最初に行う活動
- 顧客から要求を導き出す
  - 何が必要かを明らかにする
- 要求をモデリングする
  - 全ての利害関係者に分かるように表現する

要求定義とは

4.1.1

- 要求とは、必要としている動作を表現したもの
- 要求の構成
  - オブジェクトや現象
  - それらの状態
  - オブジェクトや状態の変化のきっかけとなる活動
- 顧客が必要としている動作・能力に着目する
  - どのように実現するかを言うことなく、どのような動作が必要かを表現する
  - システムとしての実現方法とは違う

要求の種類

4.1.2

機能要求	• 求める振る舞いを動作として表現
品質要求 （非機能要求）	• ソフトウェアが満たさなければならない品質に関わる性質
デザイン要件	• ソフトウェアが動作するプラットフォームやインタフェースの指定
プロセス要件	• システム開発の方法やリソースについての要件

利害関係者＝ステークホルダ

4.1.3

顧客	• ソフトウェア開発の費用を払う人
ユーザ	• システムを使う人
ドメインの専門家	• システムが自動化する業務を良く知る人
マーケティング担当者	• 将来の動向やユーザについて調査する人
弁護士など	• 政府、安全、法律・規則等に精通した人
専門家	• ソフトウェア開発者など技術に精通した人

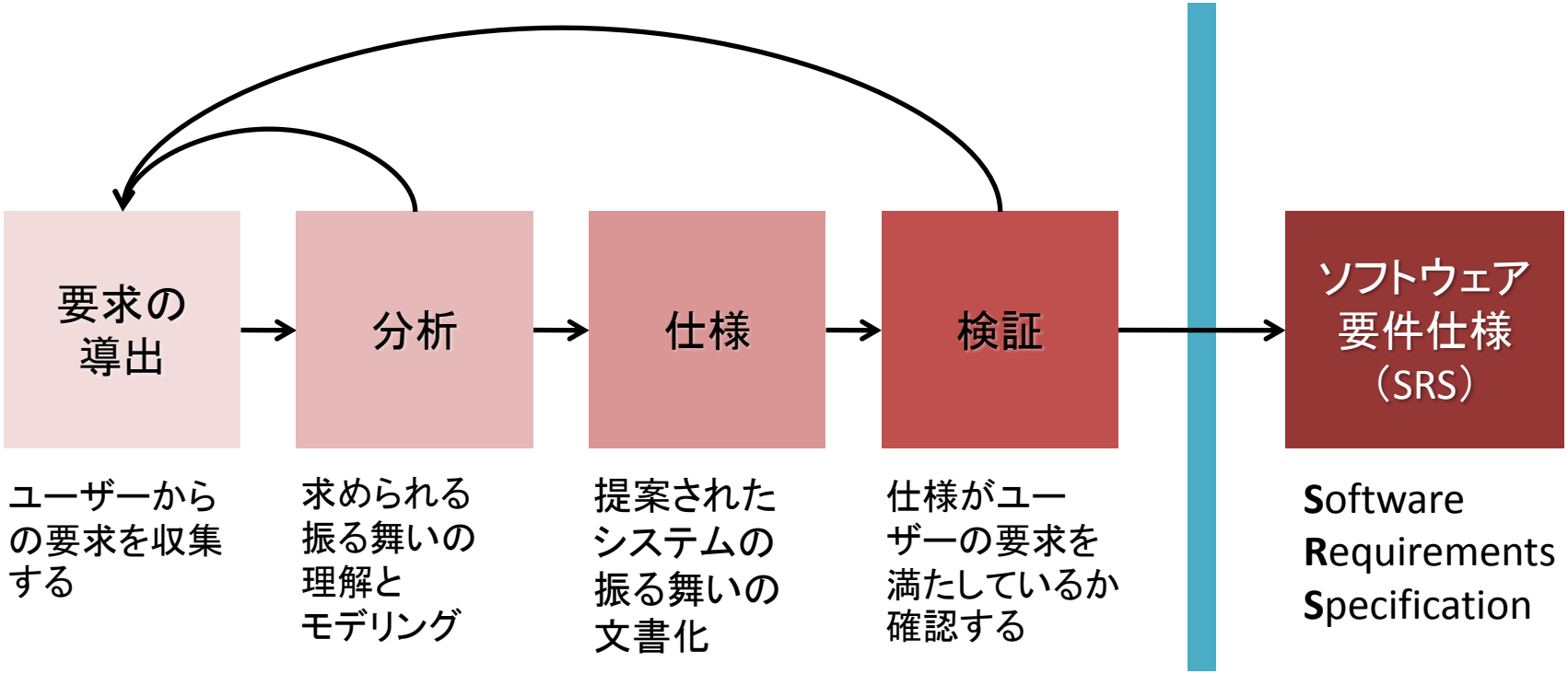
要求の導出と定義

- 何が必要なのか、顧客自身も分からないことがある
- 要求は、そのシステムに関与するすべての人と話合うことが重要
  - 要求を持つのはステークホルダである
  - ステークホルダごとに要求することや要求の重要性が異なる
- 要求の決定に当たっては必ず合意すること
  - 要求について合意できなければプロジェクトは必ず失敗する！

4.1.6

要求を導き出す手順

- 担当者：要求アナリスト、システムアナリスト
- 成果物：要求定義書



4.2

要求を導き出す手段

- 情報収集
  - 利害関係者へのインタビュー
  - 既存資料の調査
  - 現行システムを参考にする
  - ユーザの業務を習う
  - ユーザや利害関係者をグループでインタビュー
  - ドメイン（業務分野）特有の手法に着目する
- 発想支援
  - ユーザとのブレインストーミング
- 合意形成

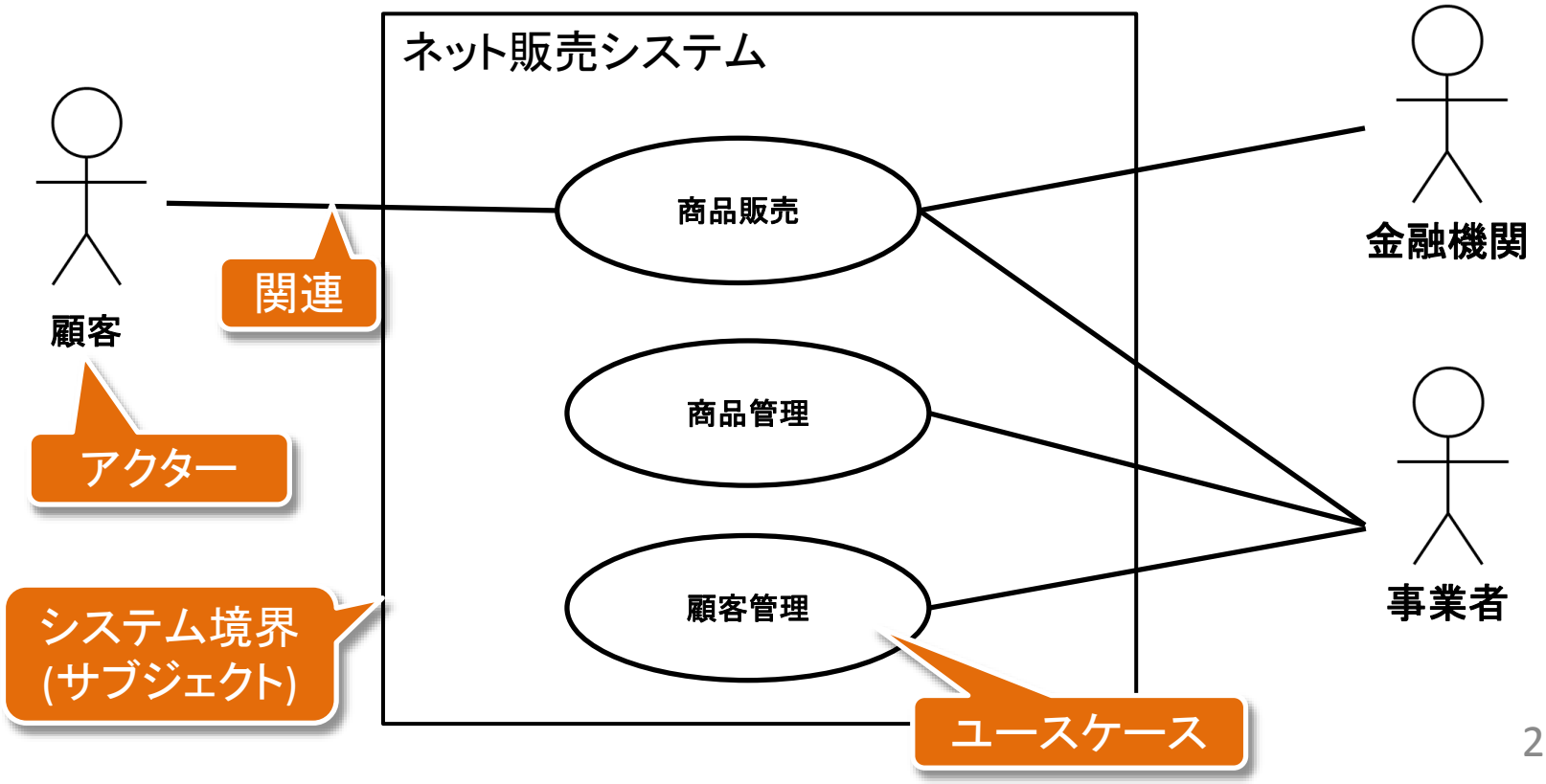
4.5
2.2.1

ユースケース分析

- ユースケース
  - 要求されるシステムの利用のされ方
  - 対象とするシステムの範囲（サブジェクト）とそれに関わる外部の役割（アクター）で表す
- ユースケースを考える目的
  - 動作や機能を明らかにする
  - 動作に関与する人や物を明らかにする
- ユースケース図
  - システムの使われ方を大雑把に捉える
  - 機能をすべて詳細に列挙しなくていい

ユースケース図

- ユースケースとの関係を記述した図
  - アクター　：ユースケースに関わる人や物
  - 関連　：アクターとユースケースの関係を示す線
  - ユースケース：対象となる活動



本日のまとめ

1.3.6

- 与えられた課題に対して
  - 分析する
  - 解決策を構築する
- 品質と効率を制御する
  - 異なる複数の観点・立場で捉える
- 要求を形作る（モデリング）
- システムの境界に留意する
  - ユーザの立場で考える（ユースケース）

次回講義の事前学習：
8.1, 8.2.1, 8.2.2, 8.2.5, 4.1.1, 4.1.2

授業の進め方

- 配点
  - クイズ　5%
  - 期末試験　60%
  - 演習　35%
- 講義担当のTA
  - 遠藤俊大 m5221201、新井雅裕 5201202

- ウェブページ
  - moodleサイト　http://sealpv0.u-aizu.ac.jp/moodle/
  - 授業ウェブサイト　http://borealis.u-aizu.ac.jp/classes/se1/
  - 講義資料は予習し、自分で用意（印刷等）してください

学内アクセス限定

- 教科書・参考文献

書名	著者	出版社
ソフトウェア工学	岸知二, 野田夏子	近代科学者
Software Engineering Theory and Practice I	Shari Lawrence Pfleeger, Joane M.Atlee	Prentice Hall
UML Distilled	Martin Fowler	Addison-Wesley

※その他、参考情報はウェブページに掲載します

授業の受け方

予習

- 教科書の指定範囲と講義資料を授業前に読む／分からないことは自分で調べてみる

講義（7限）

- 講義資料の解説を聞いて理解を深める／分からない事は、クイズの質問欄で聞く

講義内演習（8限）

- クイズや小課題を個人またはグループで解いて理解を深める

演習（9限）

- 演習課題で手を動かして理解する

復習

- 教科書を再度読んで理解を確認する