UMTP
UML Based Modeling Technologies Promotion

アジャイル開発における UML適用ガイドラインと アジャイルな現場でのモデリング実践

UMLモデリング推進協議会(UMTP)
アジャイル開発部会

中菱エンジニアリング㈱　古川　剛啓
㈱オージス総研　原田　巌

目次

- アジャイル開発部会紹介
- アジャイルソフトウェア開発向けUML適用ガイドライン（Ver. 1.1）紹介
- ガイドラインの適用事例紹介

アジャイル開発部会活動内容

主査：中原　俊政　氏（東京情報大学　非常勤講師）
副主査：竹政　昭利　氏（株式会社オージス総研）

◆目標
　アジャイル開発でのUMLの適用事例の収集とUMLの効果的な適用方法を研究する。

◆活動内容
1. アジャイル開発＋UMLの事例を収集する。
2. アジャイル開発におけるUML適用ガイドラインを準備する。
3. アジャイル開発＋UMLの事例セミナーを実施する。
等

◆活動頻度
1. 月1回程度の打合せ
2. セミナー（年1～2回程度）

他団体（IPA）との交流（セミナー）の様子

2013年7月23日
『現場に学ぶ実践アジャイルモデリング：アジャイルにモデリングは必要か』

株式会社ゼンアーキテクツ　岡 氏

アジャイル開発＋UMLの事例セミナーの様子

2016年2月18日
『DtoDに基づくアジャイル要求入門』

㈱オージス総研　藤井氏（写真 左）
張氏（写真　右）

自己紹介

ふるかわ　よしひろ

古川　剛啓

中菱エンジニアリング㈱
航空ソフトウェア設計室
グループマネージャ

UMTP L3モデラー
UMTP アジャイル開発部会メンバ
OMG Advanced
認定スクラムマスター

UMTP
UML Based Modeling Technologies Promotion

アジャイルソフトウェア開発向け UML適用ガイドライン（Ver. 1.1）紹介

中菱エンジニアリング㈱　古川　剛啓

ガイドライン目次

まえがき（1章）

- ✓ 昨今、ソフトウェア開発に携わる人々の間で「アジャイル」というキーワードがよく聞かれるようになっている。
- ✓ 特にアイデアを素早く形にし、市場に提供することがより利益を生むようなサービス業ではこの傾向がより顕著に表れている。
- ✓ 組み込みソフトウェア業界では、アジャイルソフトウェア開発の需要が低い状況にある。

まえがき（1章）

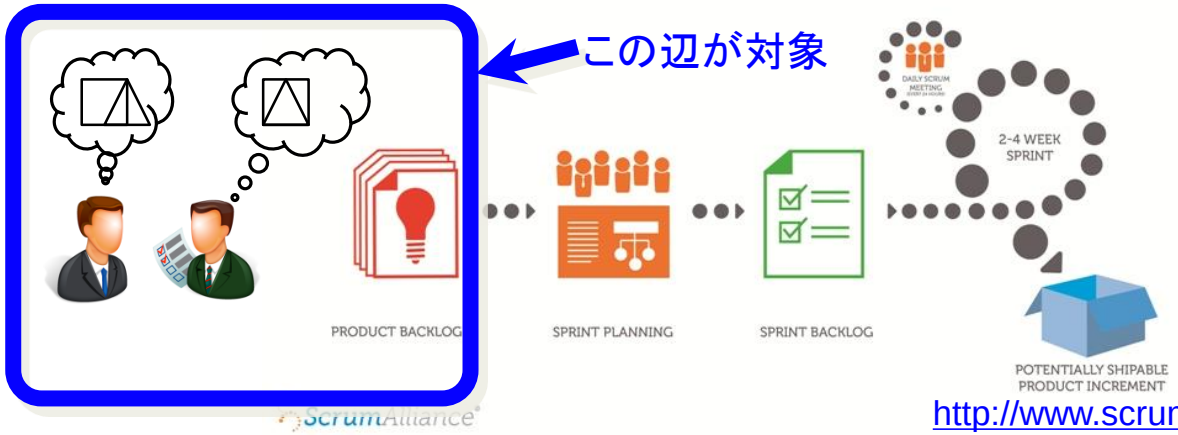
UMTP
UML Based Modeling Technologies Promotion

- ✓ 組み込みソフトウェアはUMLと親和性が高い。
  - ➢ クラス図は、ハードウェアとソフトウェアの境界を明確にし、ハードウェアとソフトウェアの結合を疎にすることが容易
  - ➢ ステートマシン図 は、イベント駆動型プログラムの動作を記述するのに最適　等
- ✓ UML等を活用したモデル駆動開発は、以下の特徴がある。
  - ➢ ユースケース単位で開発することができる
  - ➢ モデルを繰り返し検討することでより洗練された高品質なソフトウェアが製作可能
- ✓ この様な背景から、ガイドラインはソフトウェアの技術者を対象とした。

UMLやアジャイル関連書籍とどこが違うのか（1.3項）

UMTP
UML Based Modeling Technologies Promotion

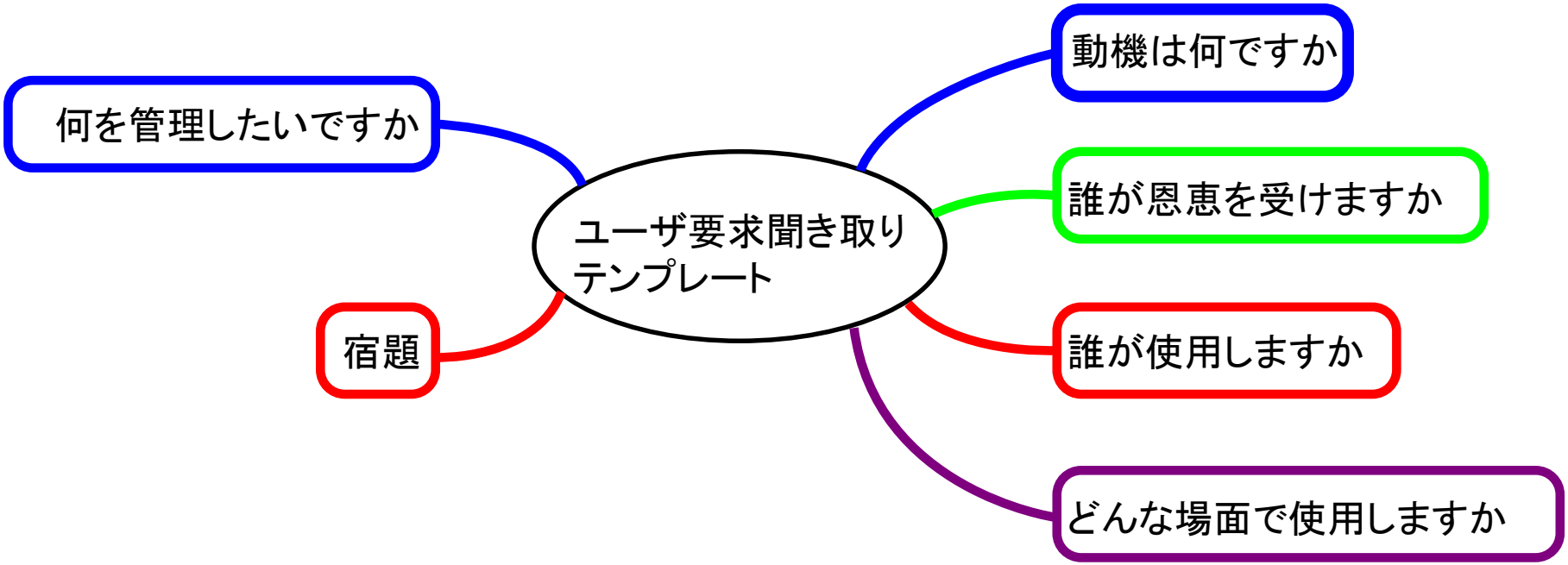
- ✓ 抽象度の高いモデルと具象度の高いアジャイルプラクティスを組み合わせることにより、高いレベルで開発者間の意識共有がされると共に、スプリント毎に妥当性確認がされるスピーディーで確実な開発を行うことを目指す。
  - Scrumでは規定されていないプロダクトバックログアイテムの発見手法を説明
  - プロダクトバックログにユースケース図を使用することを提案
  - ユースケース図の説明にアジャイルプラクティスを使用



http://www.scrumalliance.org/why-scrum

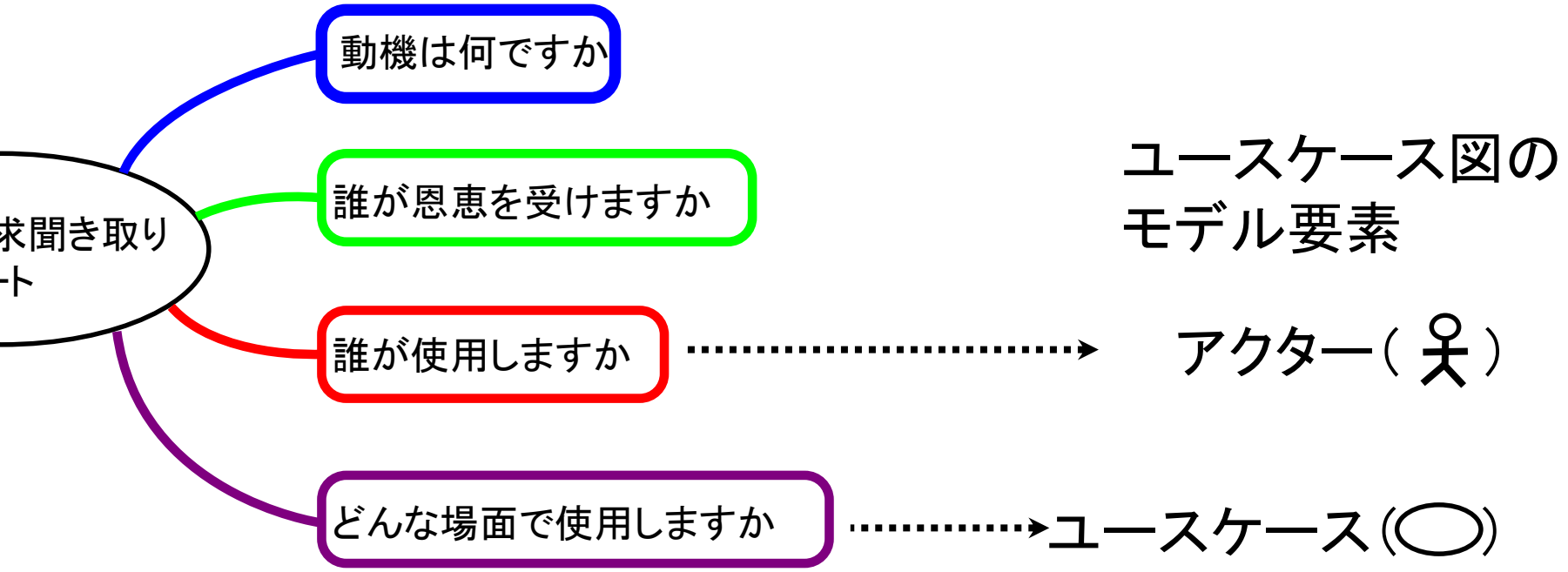
インタビューによる機能・性能の洗い出し（3.3.1項）

2012年3月28日セミナー
㈱チェンジビジョン　平鍋氏　講演資料より

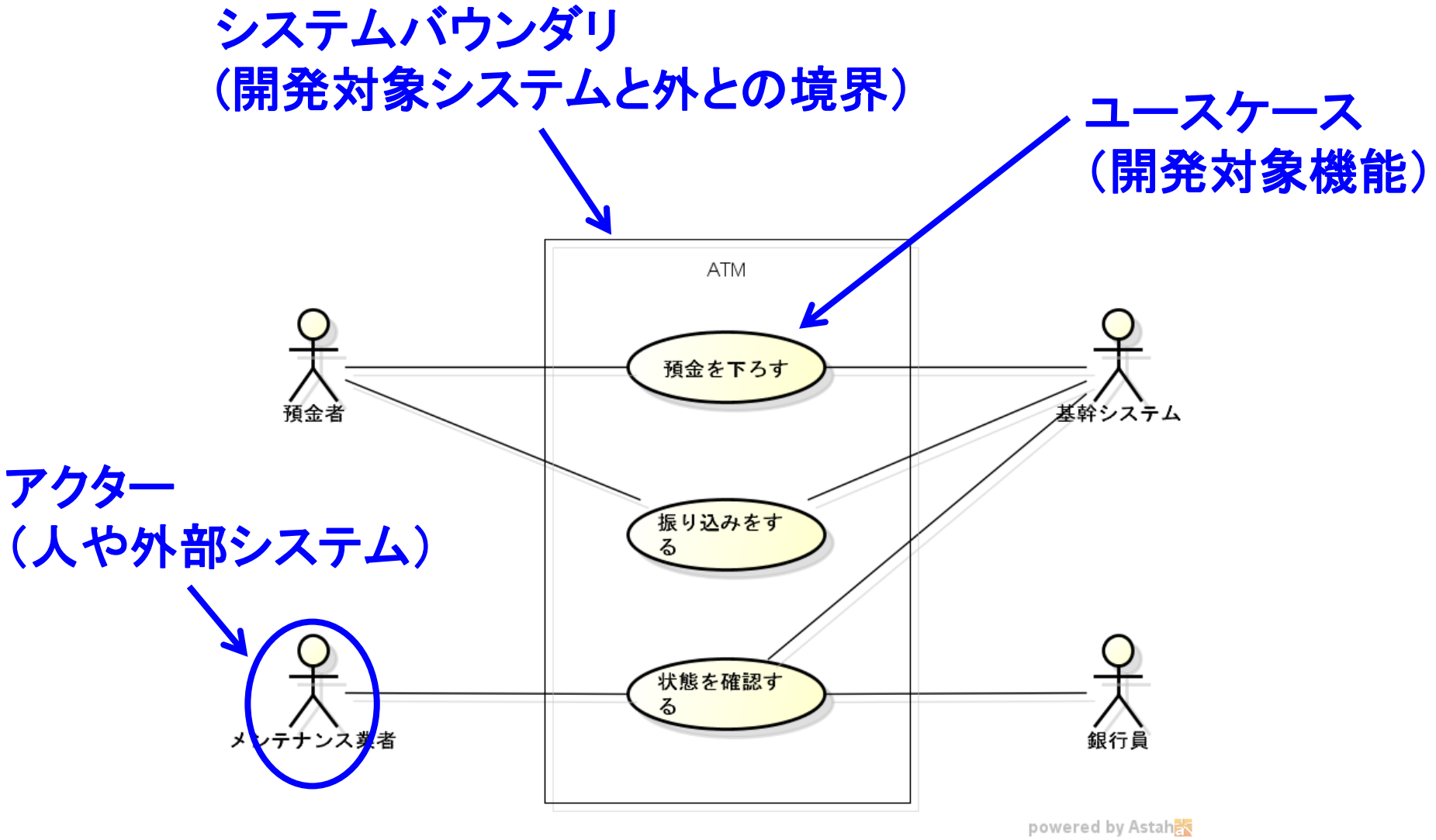


テンプレート化されたマインドマップを足掛かりにユーザの要求を見える化していく

ユースケース図による要求項目の見える化（3.4.1項）



ユースケース図(例)



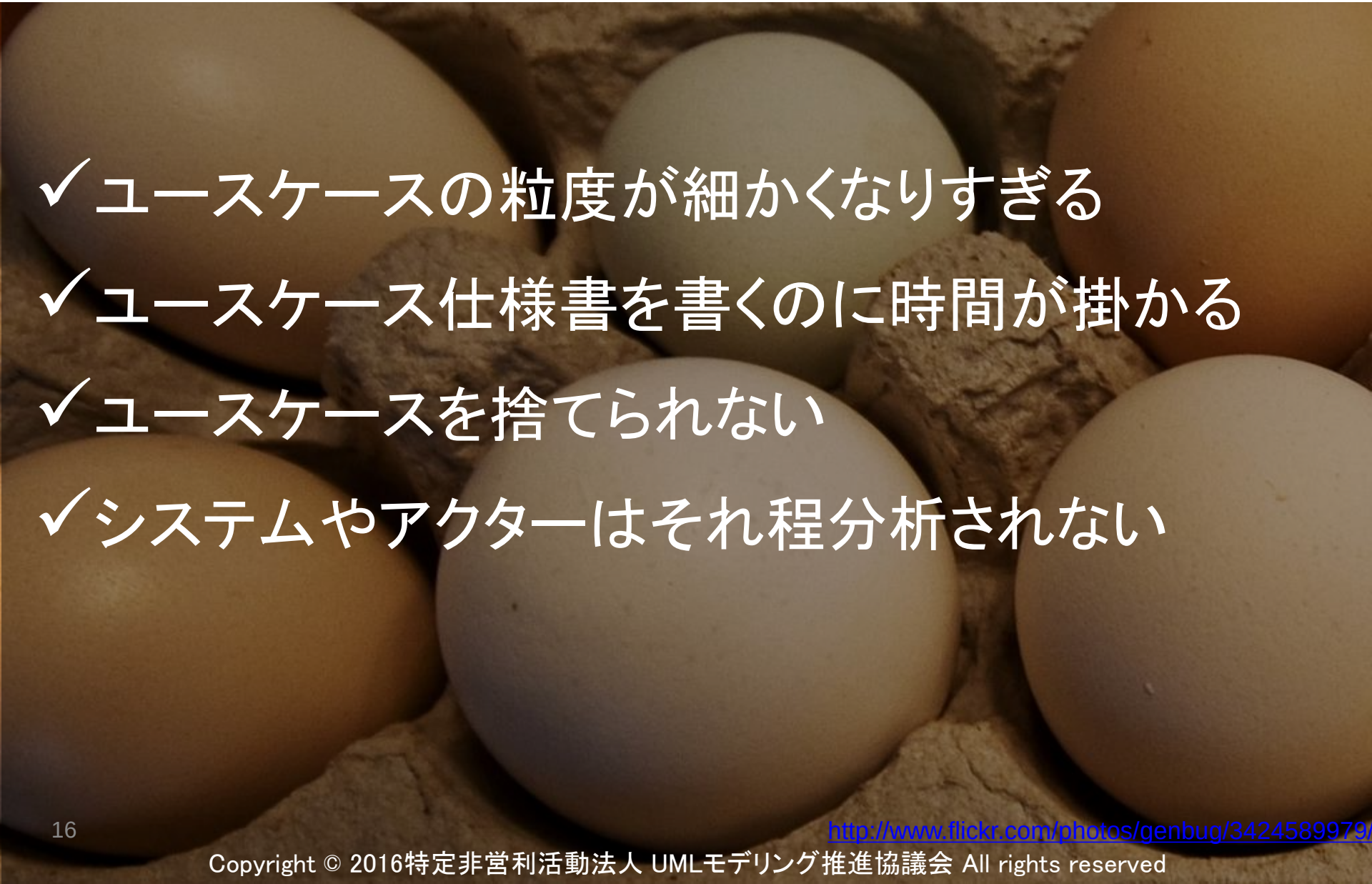
ユースケース図作成時の問題点（3.4.2項）

UMTP
UML Based Modeling Technologies Promotion

ユースケース図作成時の注意点

- ✓ユースケースを適切な粒度に保つ
  - ⇒ 見積りの精度向上のため
- ✓ユースケース毎にユースケース記述を書く
  - ⇒ ユースケースの仕様を明確にする
- ✓ユースケース記述の作成に時間を掛け過ぎない
- ✓アクター名は役割にする
  - ⇒ 同じ人でも役割によって使用する機能が異なる

実際にやってみると・・・



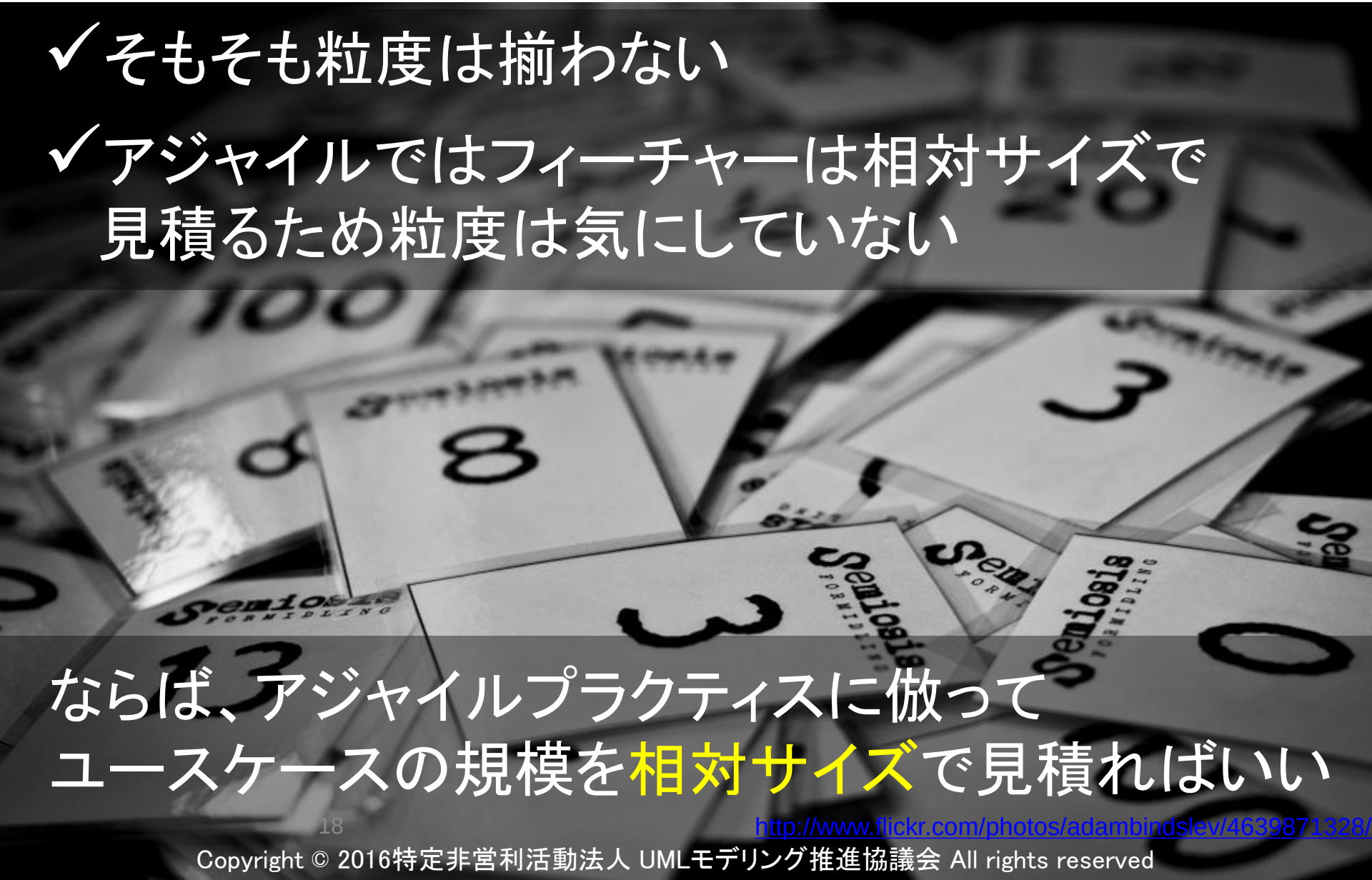
- ✓ユースケースの粒度が細かくなりすぎる
- ✓ユースケース仕様書を書くのに時間が掛かる
- ✓ユースケースを捨てられない
- ✓システムやアクターはそれ程分析されない

http://www.flickr.com/photos/genbug/3424589979/

ようこそユースケース地獄へ

ユースケース地獄に対応する(3.4.3項)

UMTP
UML Based Modeling Technologies Promotion



- ✓そもそも粒度は揃わない
- ✓アジャイルではフィーチャーは相対サイズで見積るため粒度は気にしていない

ならば、アジャイルプラクティスに倣って
ユースケースの規模を相対サイズで見積ればいい

http://www.flickr.com/photos/adambindslev/4639871328/

先ずはユーザーストーリーを使う

ユーザーストーリーとは、実現したいと思っているフィーチャーを簡潔に示したもので、短い文章として表したもの

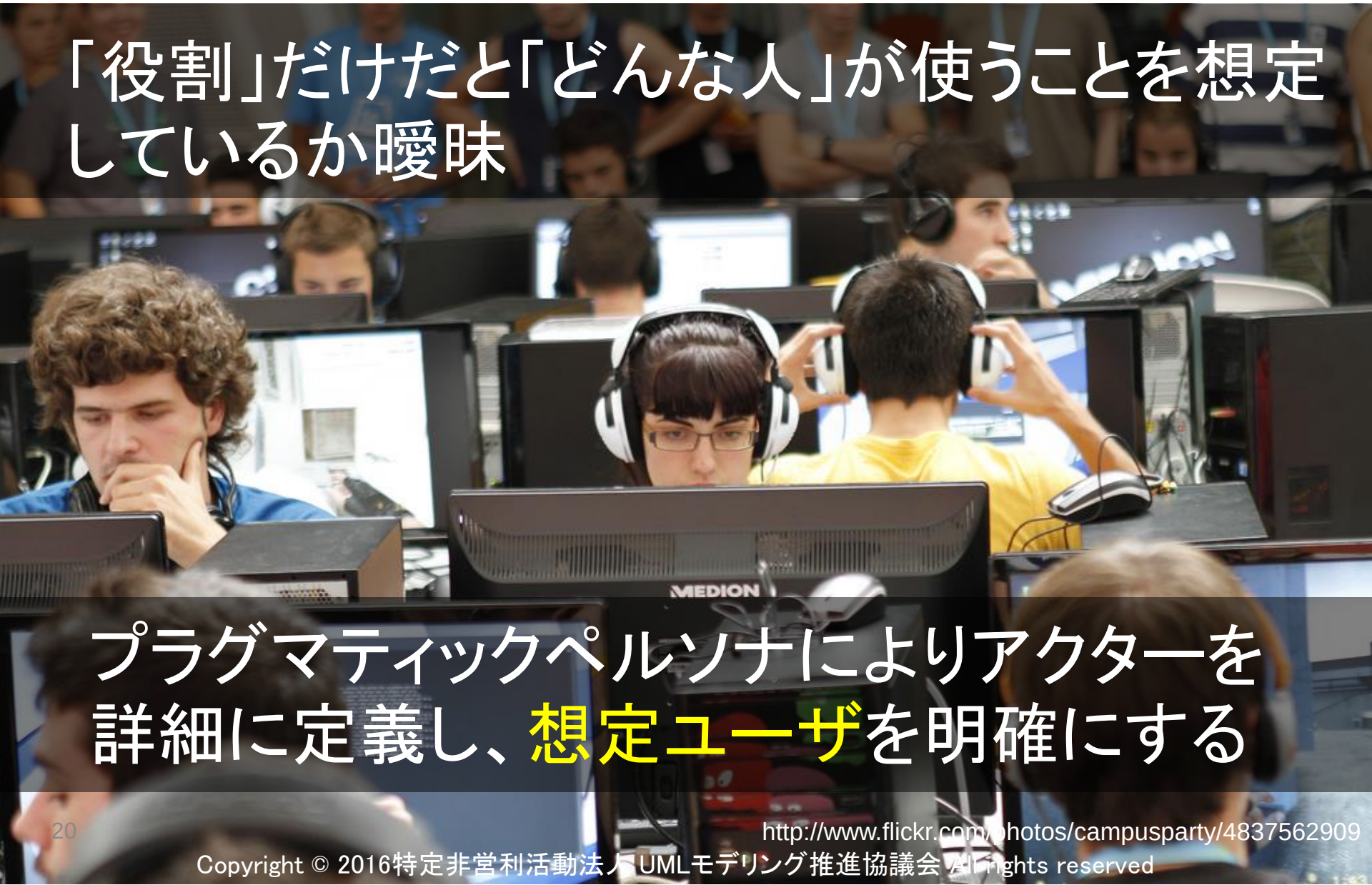
メリット

✓作成に時間が掛からない
✓フィーチャーの本質を捉える事ができる
✓ユースケースの粒度によらない

http://www.flickr.com/photos/psd/8591351239/

ユースケース仕様書は適切なタイミングで必要な量だけ書く

アクターのあいまいさに対応する（3.4.4項）

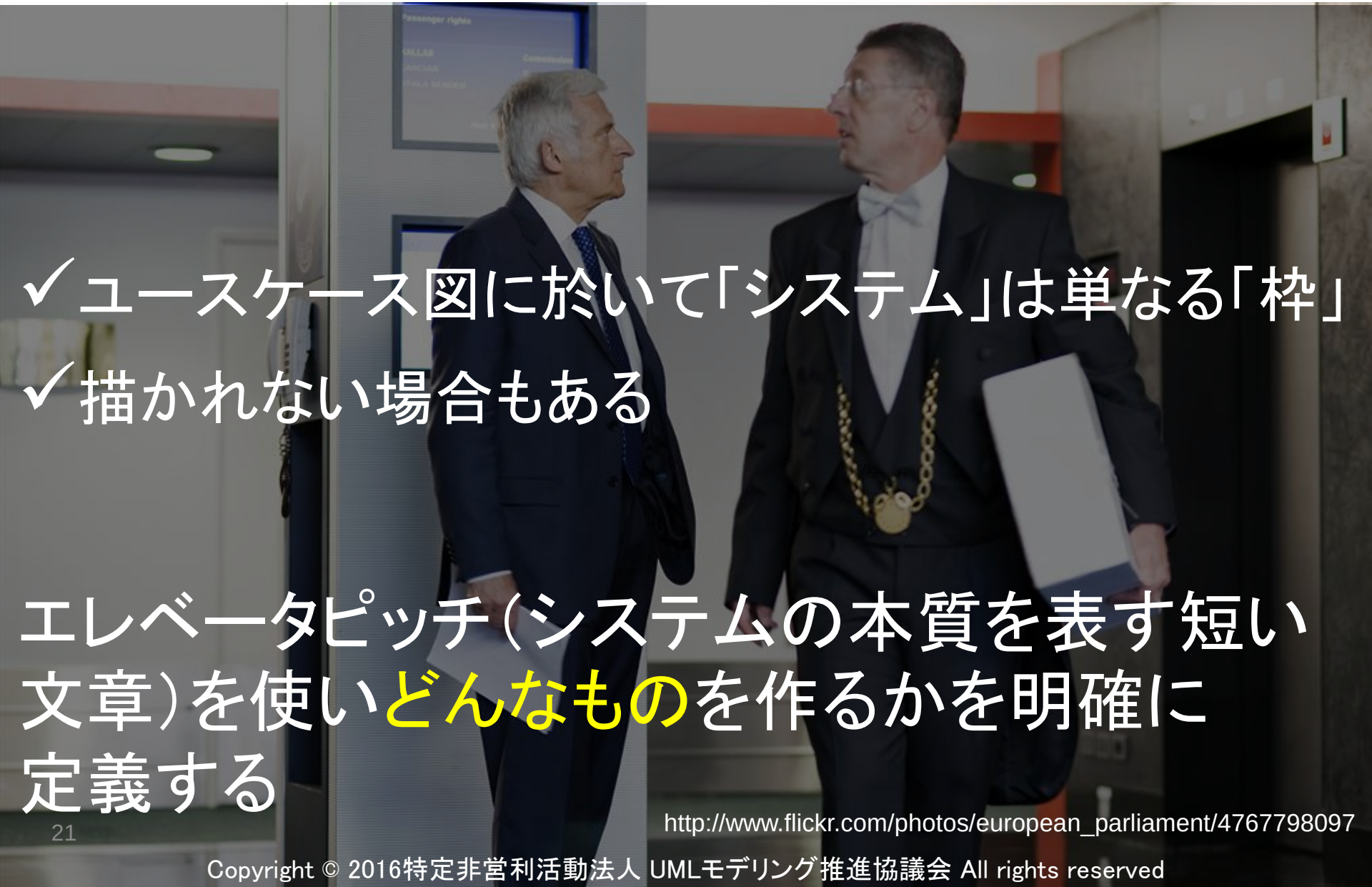


「役割」だけだと「どんな人」が使うことを想定しているか曖昧

プラグマティックペルソナによりアクターを詳細に定義し、想定ユーザを明確にする

http://www.flickr.com/photos/campusparty/4837562909

開発対象システムのあいまいさに対応する（3.4.5項）



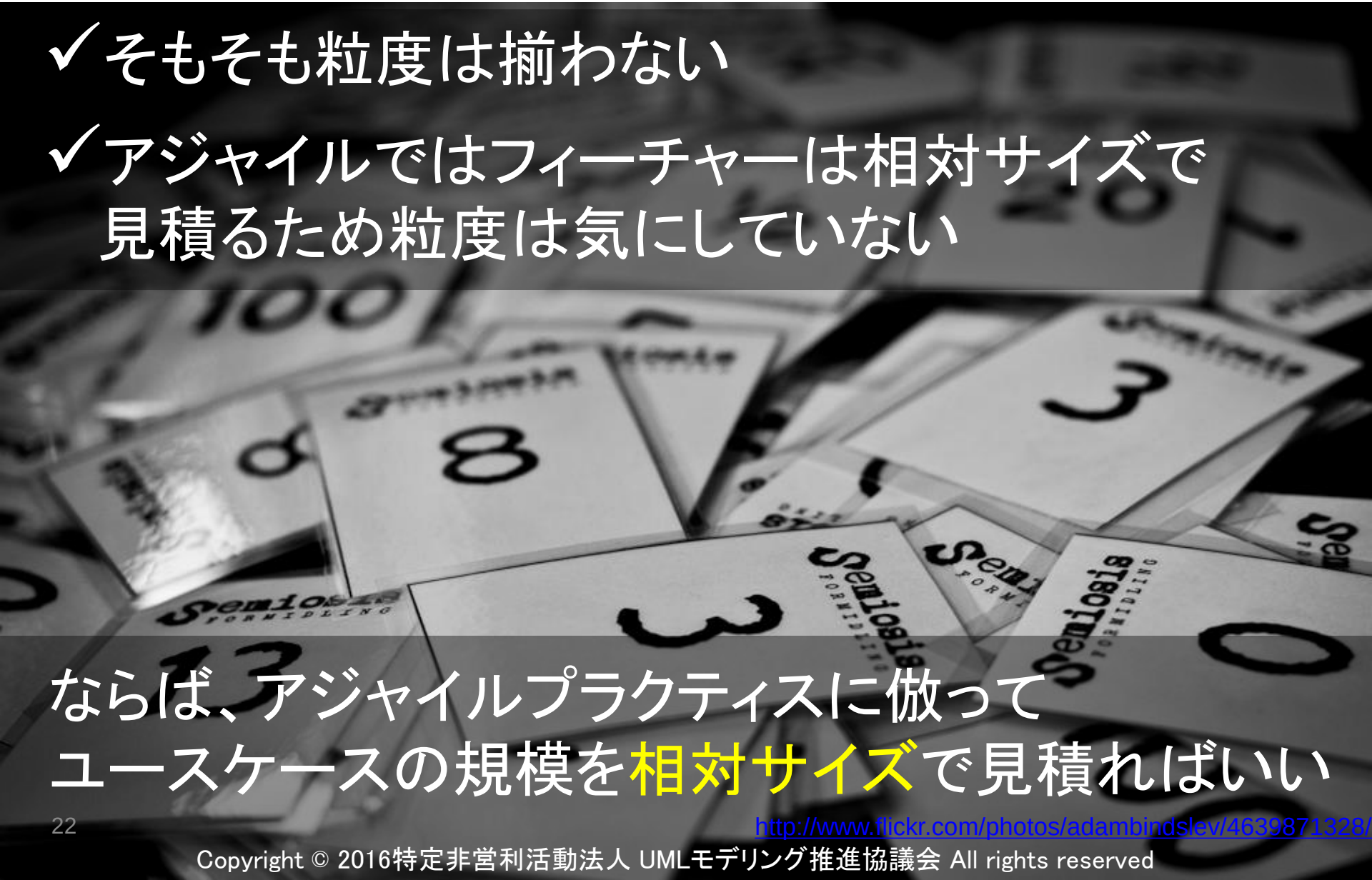
- ✓ユースケース図に於いて「システム」は単なる「枠」
- ✓描かれない場合もある

エレベータピッチ（システムの本質を表す短い文章）を使いどんなものを作るかを明確に定義する

http://www.flickr.com/photos/european_parliament/4767798097

ユースケースの規模を見積る（3.4.6項）

UMTP
UML Based Modeling Technologies Promotion



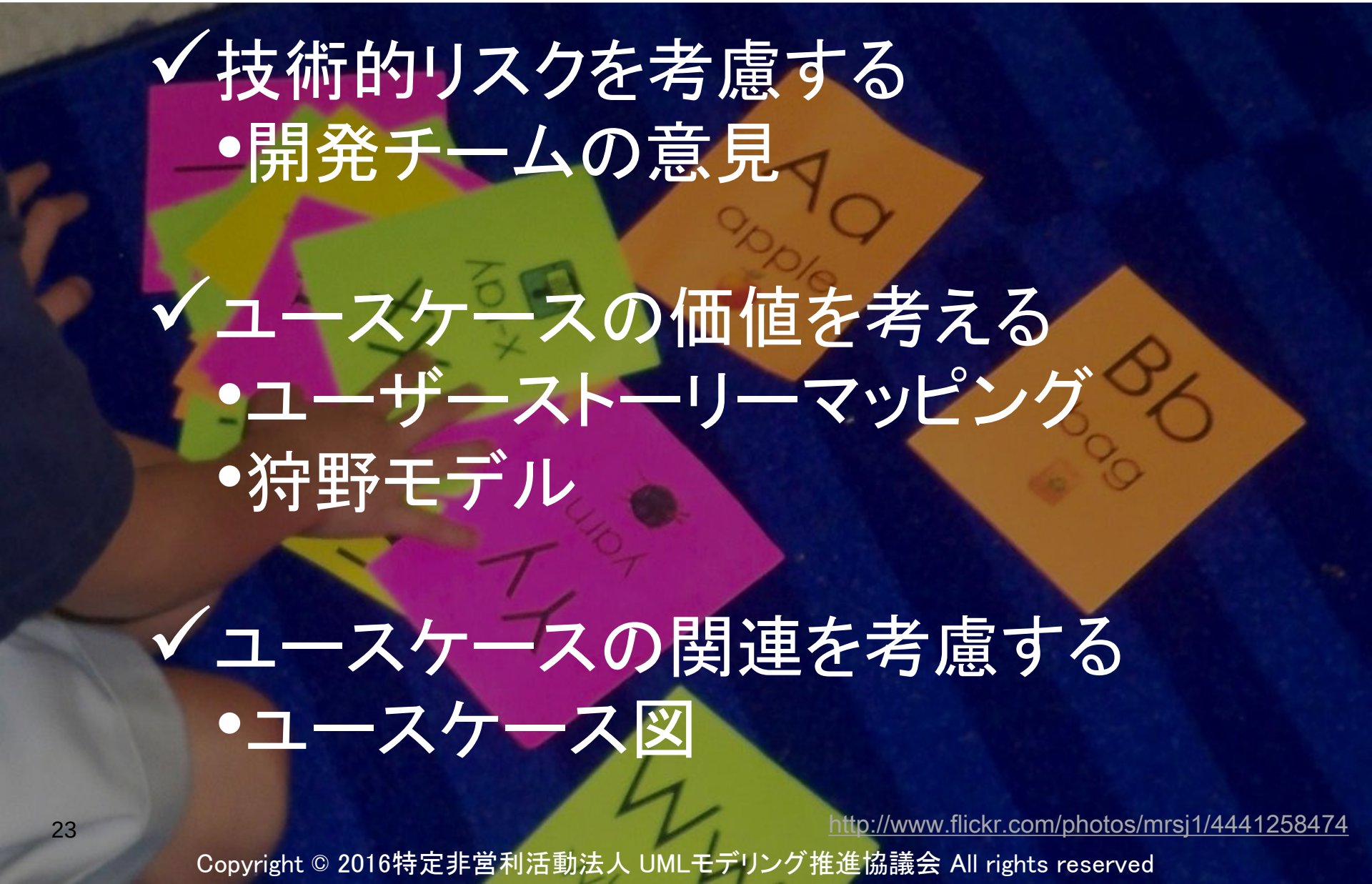
- ✓そもそも粒度は揃わない
- ✓アジャイルではフィーチャーは相対サイズで見積るため粒度は気にしていない

ならば、アジャイルプラクティスに倣って
ユースケースの規模を相対サイズで見積ればいい

http://www.flickr.com/photos/adambindslev/4639871328/

ユースケースに優先順位を付ける（3.4.7項）

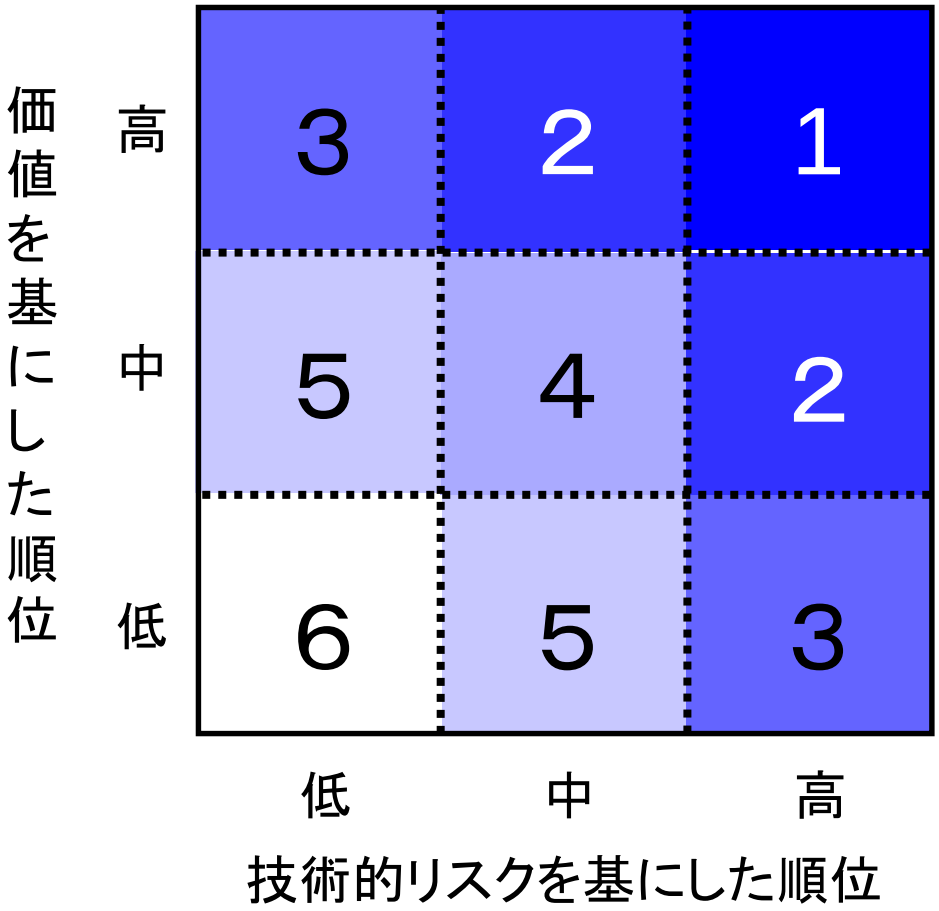
UMTP
UML Based Modeling Technologies Promotion



- ✓技術的リスクを考慮する
  - •開発チームの意見
- ✓ユースケースの価値を考える
  - •ユーザーストーリーマッピング
  - •狩野モデル
- ✓ユースケースの関連を考慮する
  - •ユースケース図

http://www.flickr.com/photos/mrsj1/4441258474

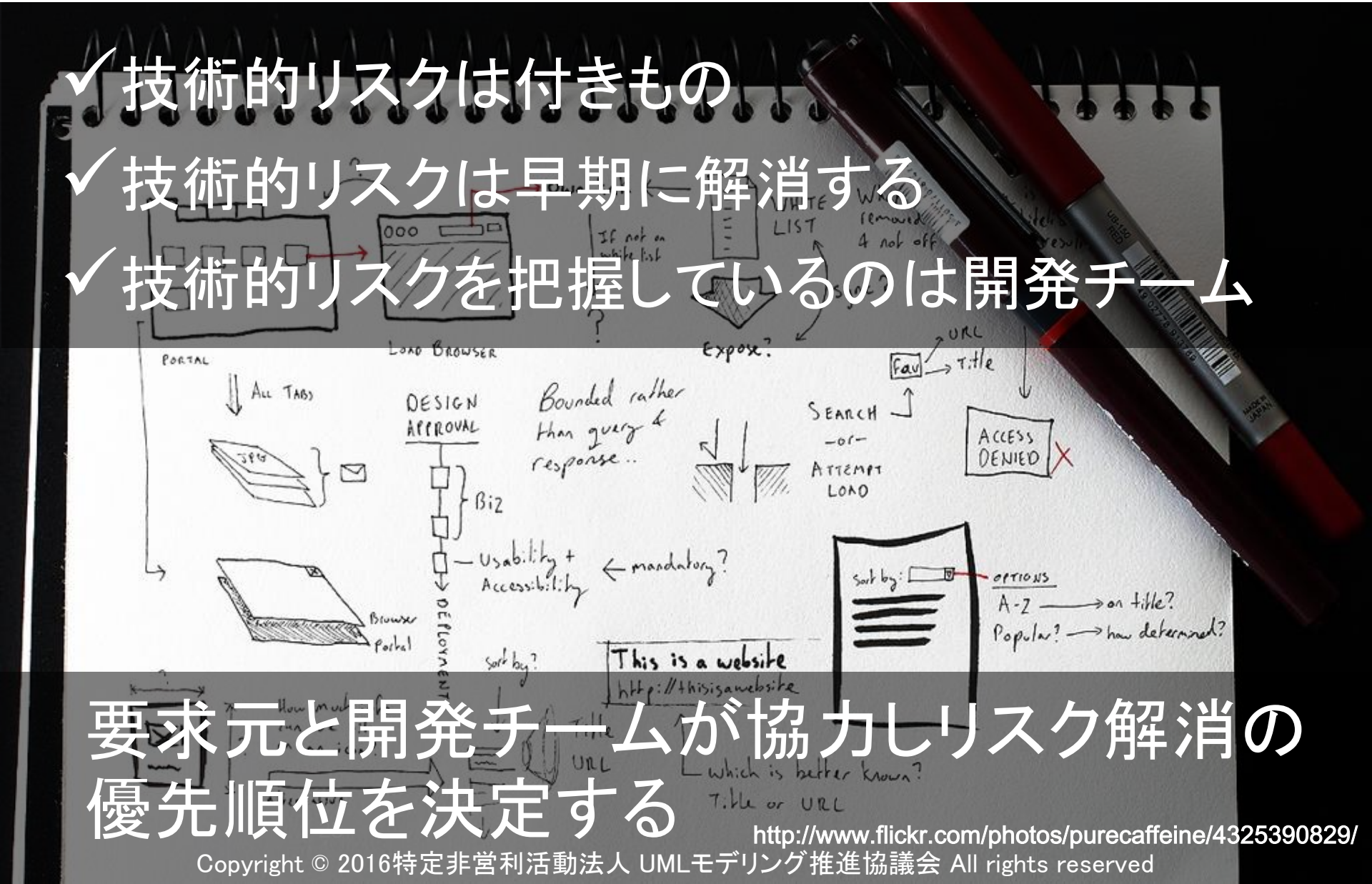
価値を基にした順位と技術的リスクをマッピングする



マスの中の数字は優先度を表す
数字が小さいほど優先度が高い

技術リスクを考慮する

UMTP
UML Based Modeling Technologies Promotion



- ✓技術的リスクは付きもの
- ✓技術的リスクは早期に解消する
- ✓技術的リスクを把握しているのは開発チーム

要求元と開発チームが協力しリスク解消の優先順位を決定する

http://www.flickr.com/photos/purecaffeine/4325390829/

ユーザーストーリーマッピングを使用する

- ✓ユーザーが体験するタスクの順番毎（ワークフロー）にユースケースをマッピング
- ✓最小の価値を有する製品（MVP：Minimum Viable Product）を構成するユースケースから優先順位を決定する

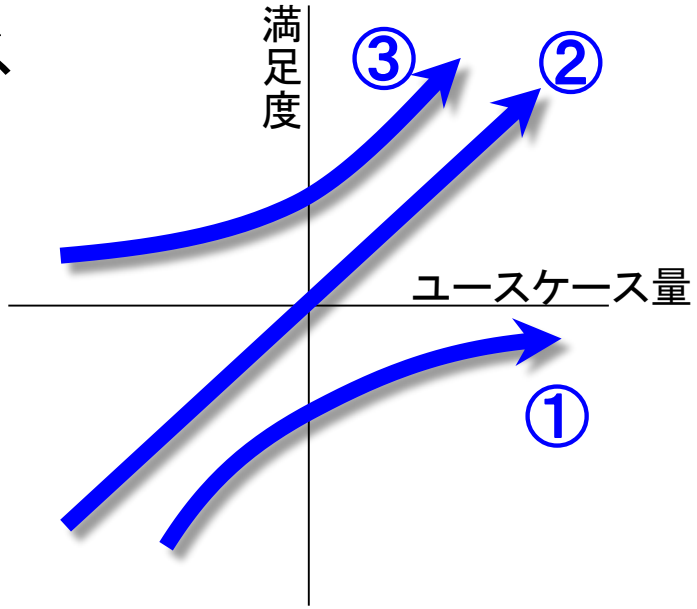
ユーザー（アクター）

http://www.flickr.com/photos/49942291@N06/6271934371/

狩野モデルを使用する

ユースケースをカテゴリー分けし、優先順位を決定する

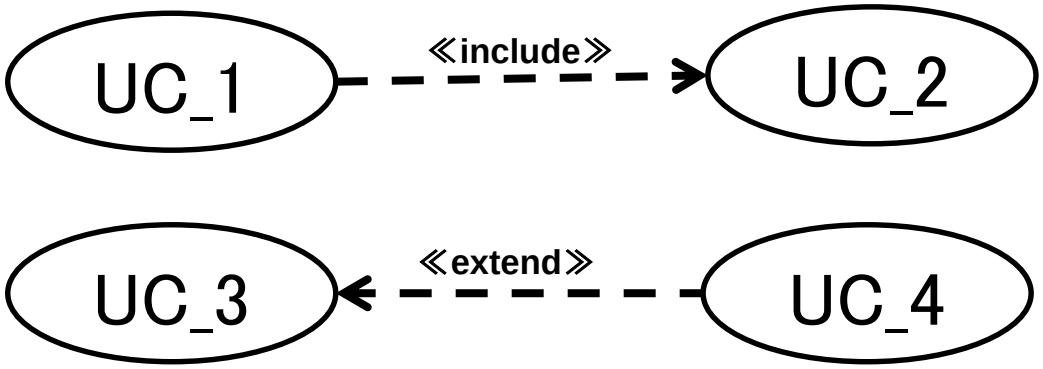
① 当たり前、または必須のユースケース
⇒ 製品に欠かせない
② 線形、一元的なユースケース
⇒ あればある程良い
③ 魅力的な、わくわくするユースケース
⇒ 大きな満足をもたらす



以上のカテゴリーを基に価値を考えると・・・

当たり前のユースケースは全て備える
線形のユースケースをできる限り多く備える

ユースケースを使用する

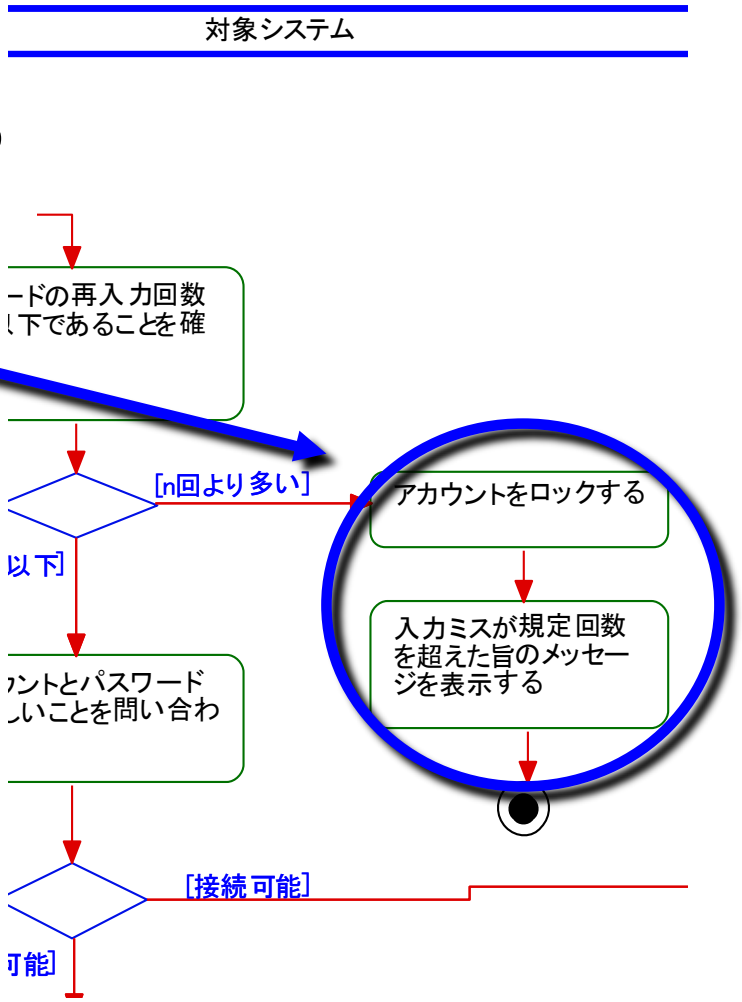


UC_1は、UC_2を包含する
UC_4は、UC_3を拡張する
従って、優先順位は
UC_2 ＞ UC_1
UC_3 ＞ UC_4　だが
UC_1とUC_4は別の観点での優先順位付けが必要

モデルを使用したPBIの詳細化（3.5項）

アクティビティ図を使用し、機能を分析する。

各アクションを詳細化されたPBIとする
例えば、これ



ユーザーストーリー

システム利用者として、規定回数以上パスワードを間違えたらアカウントをロックしたい。
なぜなら、アカウントの不正利用を防止したいからだ。

UMTP
UML Based Modeling Technologies Promotion

ガイドラインの適用事例紹介

㈱オージス総研　原田 巌

おことわり

- 今から話す内容はガイドライン策定中に私が現場で行った「Tips」のようなもので、完全にガイドラインに沿ったものではありません。
- 行った「Tips」はガイドラインに要素として組み込まれてはいます。

自己紹介

はらだ　　いわお

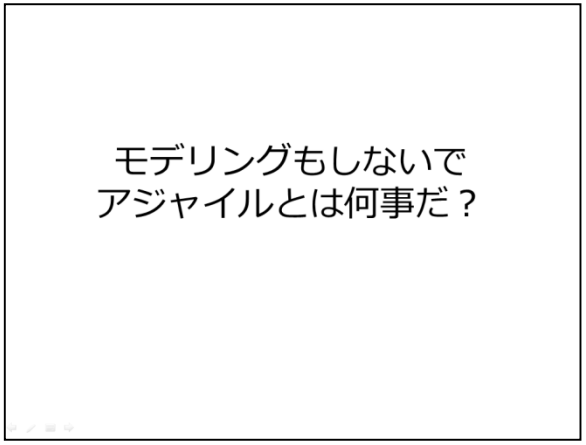
原田　巌

㈱ オージス総研
　技術部
　アドバンストテクノロジーセンター
　アジャイル開発センター（兼務）

UMTP アジャイル開発部会メンバ
認定スクラムマスター
認定スクラムプロダクトオーナー

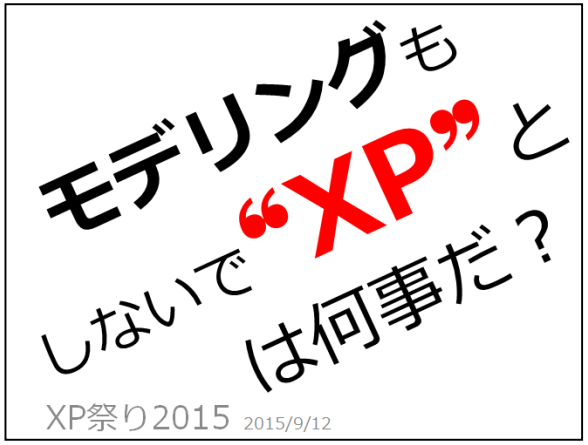
過去の発表

- 「アジャイル」界隈で「モデリング」絡みの発表をしています。
- 最初に発表した「モデリングもしないでアジャイルとは何事だ」は大きな反響を得ています。



2013年 DevLove甲子園
2014年 UMTPセミナー
2015年 要求開発アライアンス

2014年 DevLove甲子園

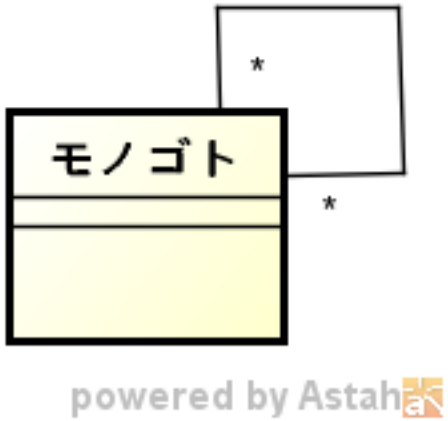


2014年 XP祭り（LT）
2015年 XP祭り（講演）

内容は
こんな感じ

＊「おお　**アジャイラー**よ！
　しんでしまうとは　なにごとだ！

モデル書いていますか？



発表者の背景

UMTP
UML Based Modeling Technologies Promotion

- ✓ SIerな中規模～大規模な開発
- ✓ DDDを始めとするモデルベース開発を実践してみた（＆勉強会で議論した）結果
- ✓ 自社だけでなくユーザー、協力会社も混ざった異文化コミュニケーション（死語？）な世界
- ✓ 既存のものをメンテナンスするよりは、新しい未知の領域を相手にしている
- ✓ アジャイル？な現場

UMTP
UML Based Modeling Technologies Promotion

問題頻発！

よく聞く失敗アジャイル開発の疑問

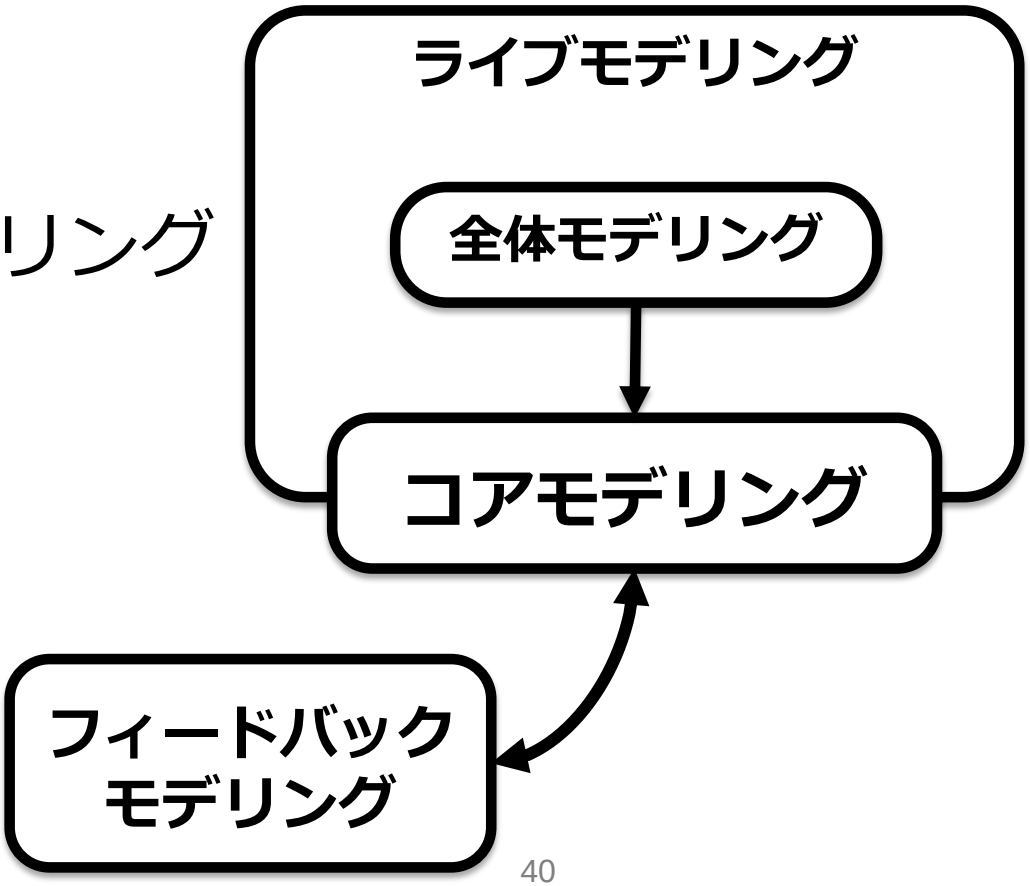
UMTP
UML Based Modeling Technologies Promotion

- 一向に決まらない仕様
  →仕様が決められないからアジャイル？
- 開発とユーザーの間の壁
  →組織の壁。縦割り組織。
  ドキュメントの投げ合い。
- 要件定義→設計→実装
  →小さいWaterfall。
  しかし、肝心のフィードバックを早く
  受けて次に向けての改善ができていない
  (組み込まれていない)

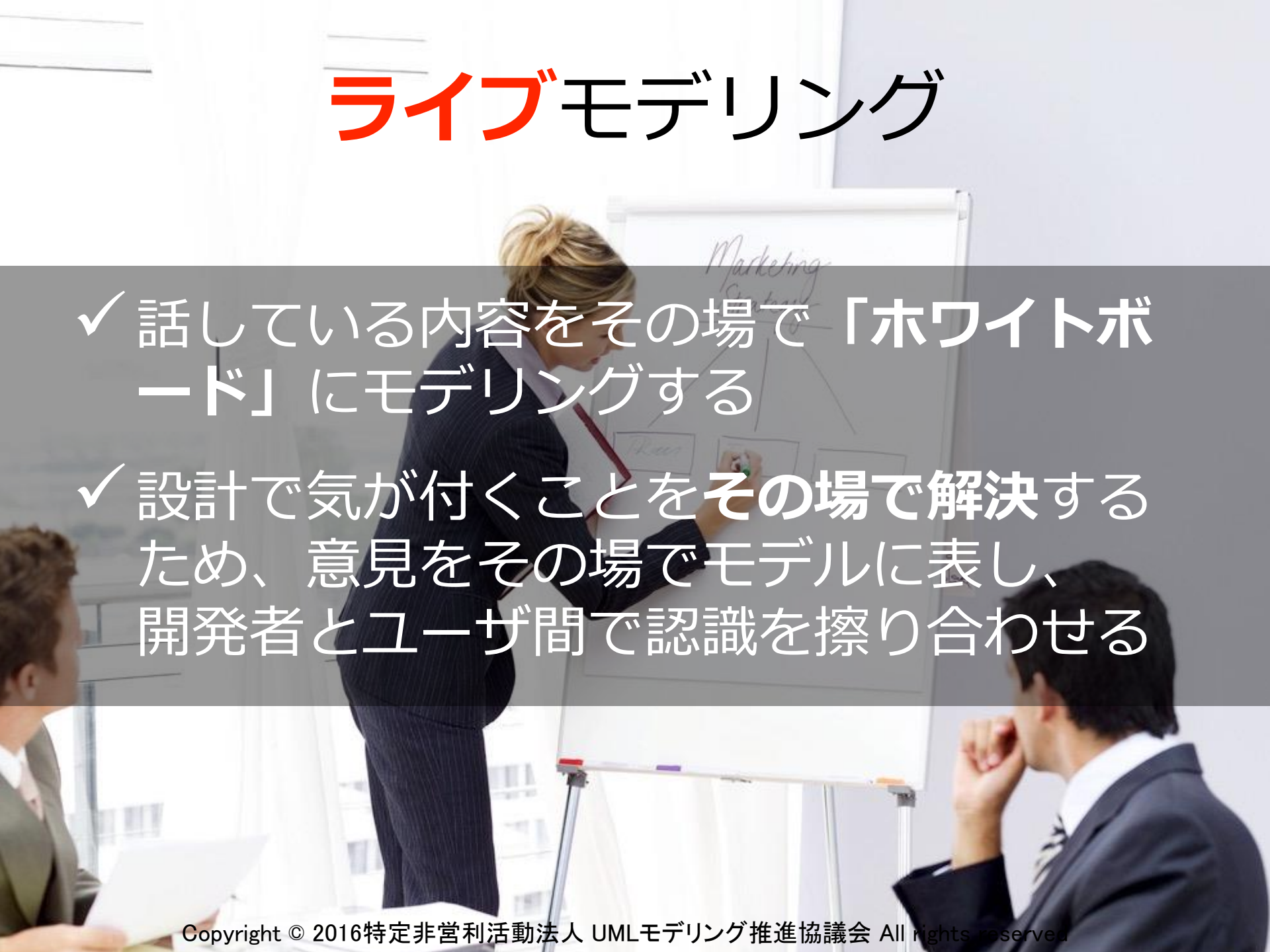
そこで**モデリング**してみる

今日話すこと

1. ライブモデリング
2. 全体モデリング
3. コアモデリング
4. フィードバックモデリング



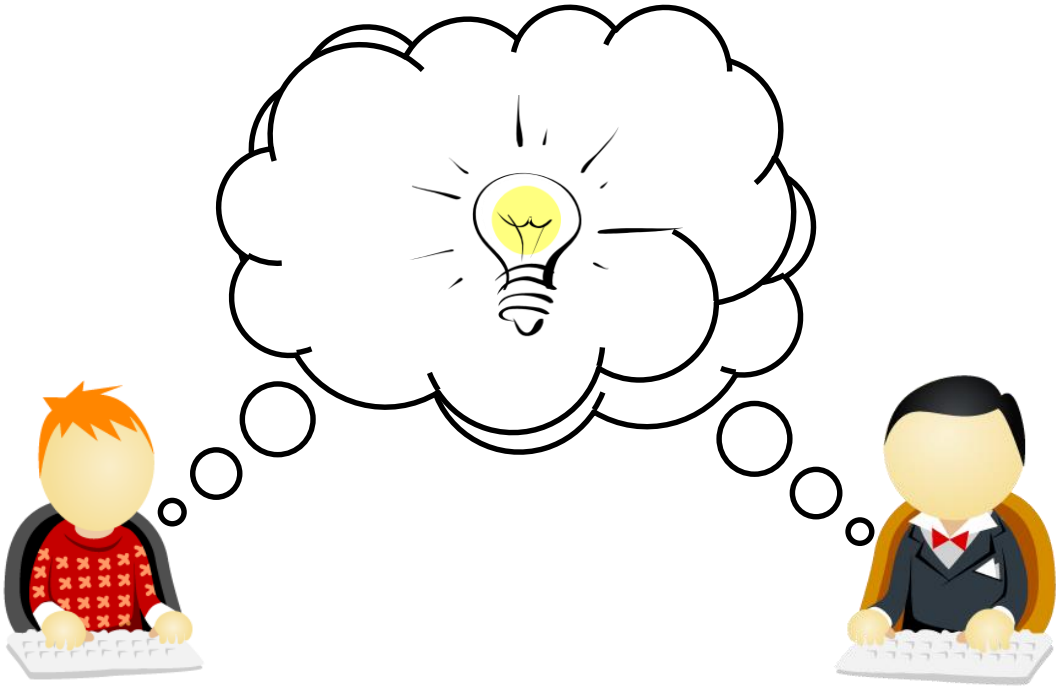
ライブモデリング



- ✓ 話している内容をその場で**「ホワイトボード」**にモデリングする
- ✓ 設計で気が付くことを**その場で解決**するため、意見をその場でモデルに表し、開発者とユーザ間で認識を擦り合わせる

モデルの意義

モデルで**得たいもの** → **共通認識**



アジャイル開発での問題点

UMTP
UML Based Modeling Technologies Promotion

- ミーティング後に開発側が考えて整理した結果（モデルや仕様書）を示されても、ユーザは自分の意見が反映されているのか分からないことが多い
  →結果として、動くソフトが見れるまで理解できず、誤りや変更があった時に影響の大きさを理解することが難しい
- クラスのように抽象化された概念を示された場合に認識のズレが発生する
  →例外ケースや特殊な業務で土台から崩される（ちゃぶ台返し？（と開発者は感じる））

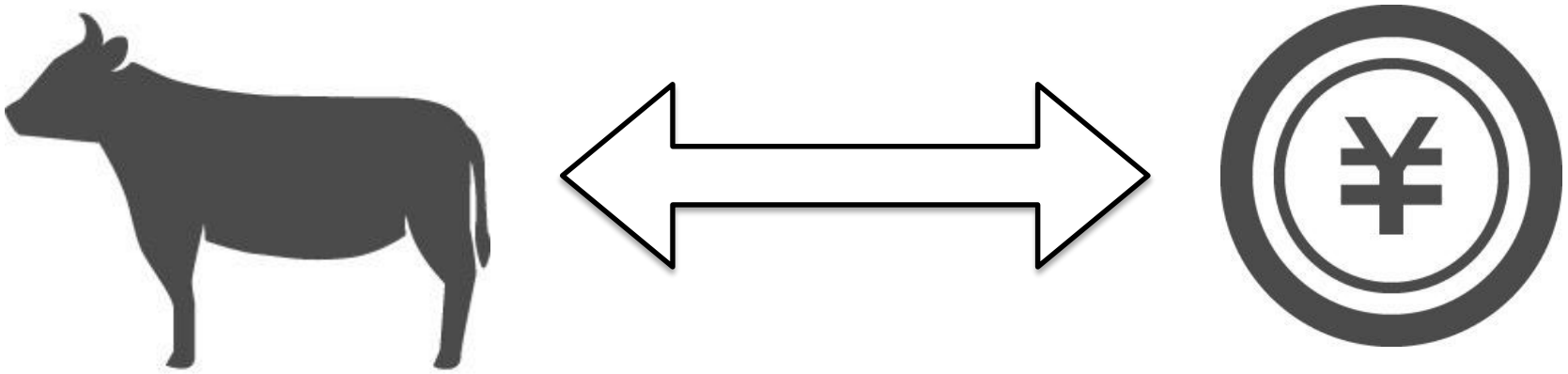
モデルで認識を合わせる

UMTP
UML Based Modeling Technologies Promotion

- 話している内容について認識を合わせる
  →繰り返し出てくる単語や文脈ごとの意味や対話する相手の表情などから重要な情報を抜き出す。
- 日本語の問題
  →自分の言葉（とモデル）で再度説明する。ストーリーとして話した時に気が付くモノとコト（関係）がある。

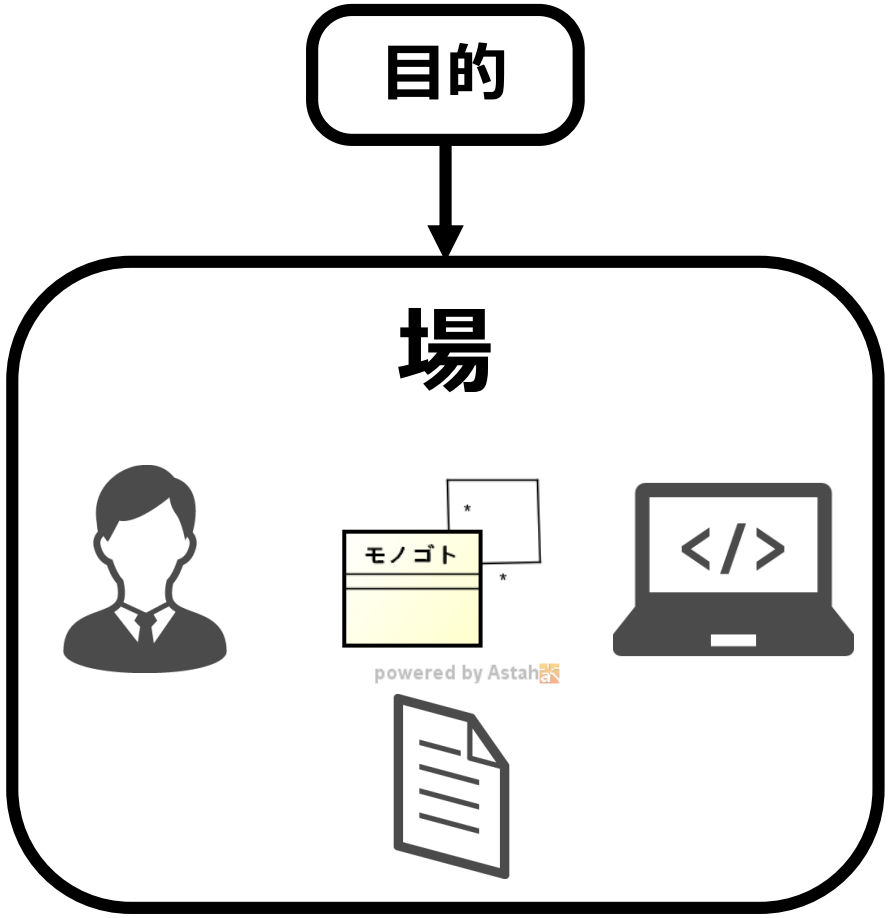
牝牛のベッシー

- 「抽象のはしご」による思考
- 高い抽象度で話をすると見えてくる**性質や共通点**
- 話す相手によって抽象度を行き来して認識を合わせることが重要
  （例：クラス図 ⇔ オブジェクト図）



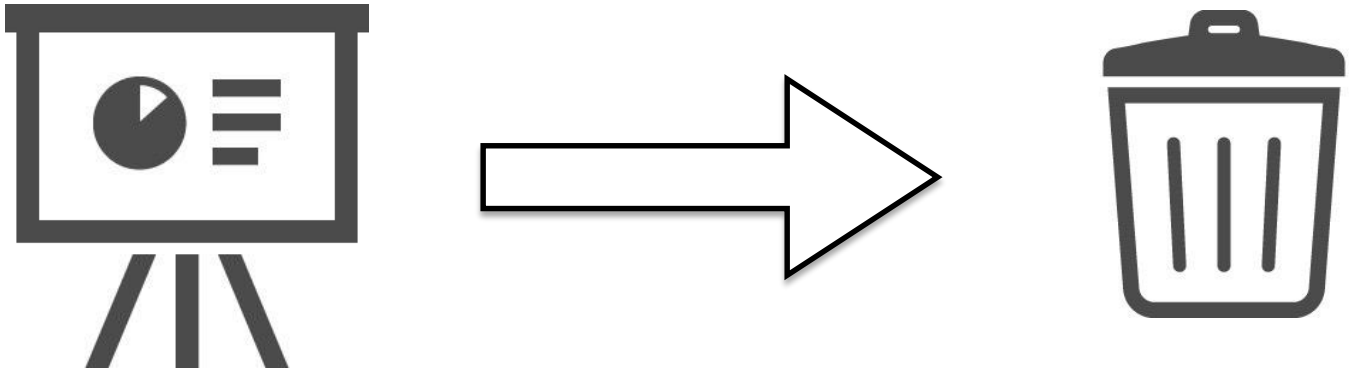
場の形成

- 『場の理論とマネジメント』伊丹敬之
  - アジェンダ
  - 解釈コード
  - 情報キャリアー
  - 連帯欲求

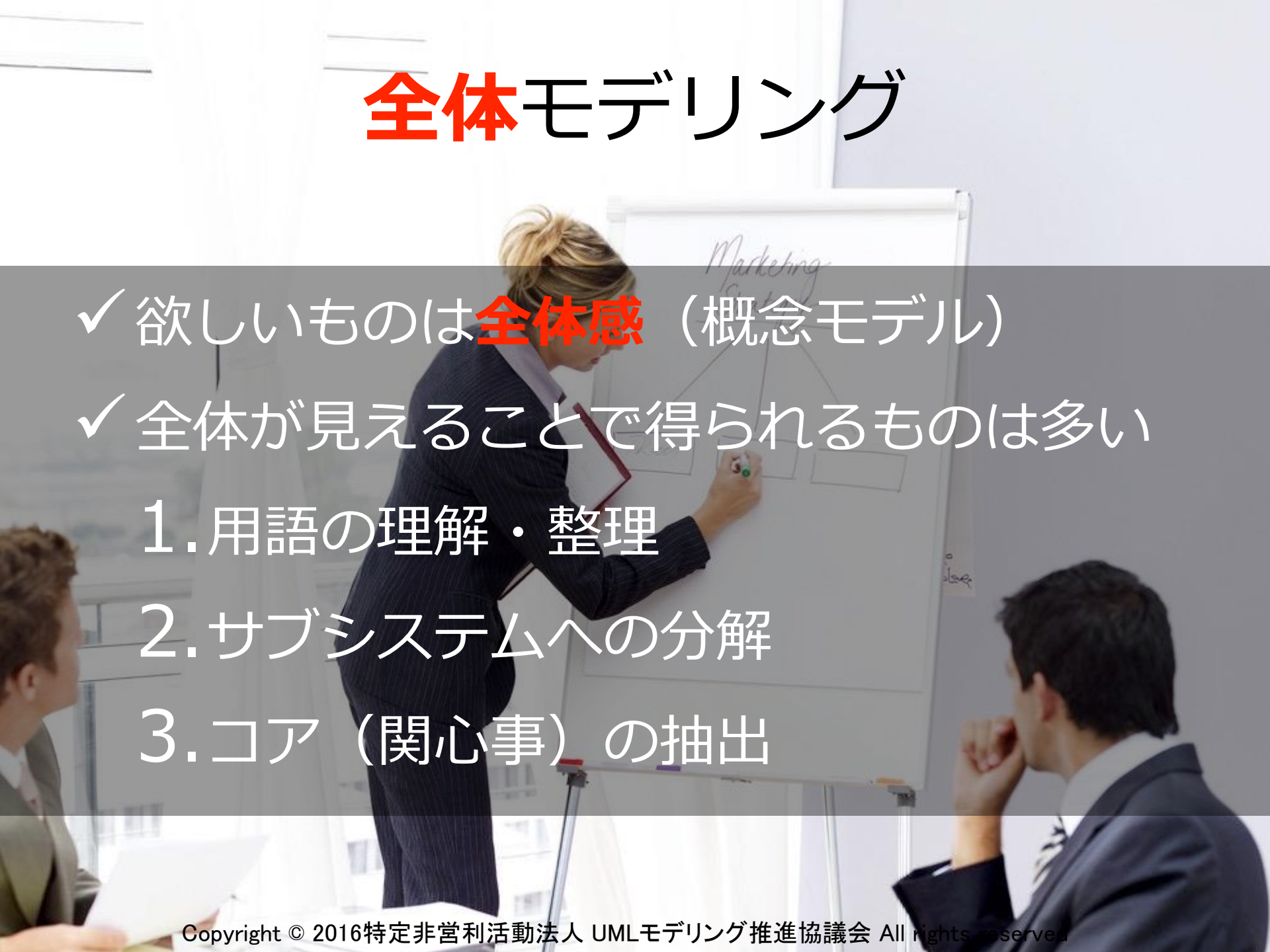


A3モデリング

- A3用紙でまとめられるくらいにしといた方が良い
  - 電子化すると（余計な）細部が気になる
  - 形になっていると覚えていなくていいと思えてしまう安心感へのアンチテーゼ
- ホワイトボードのような揮発性の高いものは無くなるべきだし、そもそも忘れてしまうようなモデルは要らない
  ＝ A3用紙分の情報なら書くのに時間不要



全体モデリング



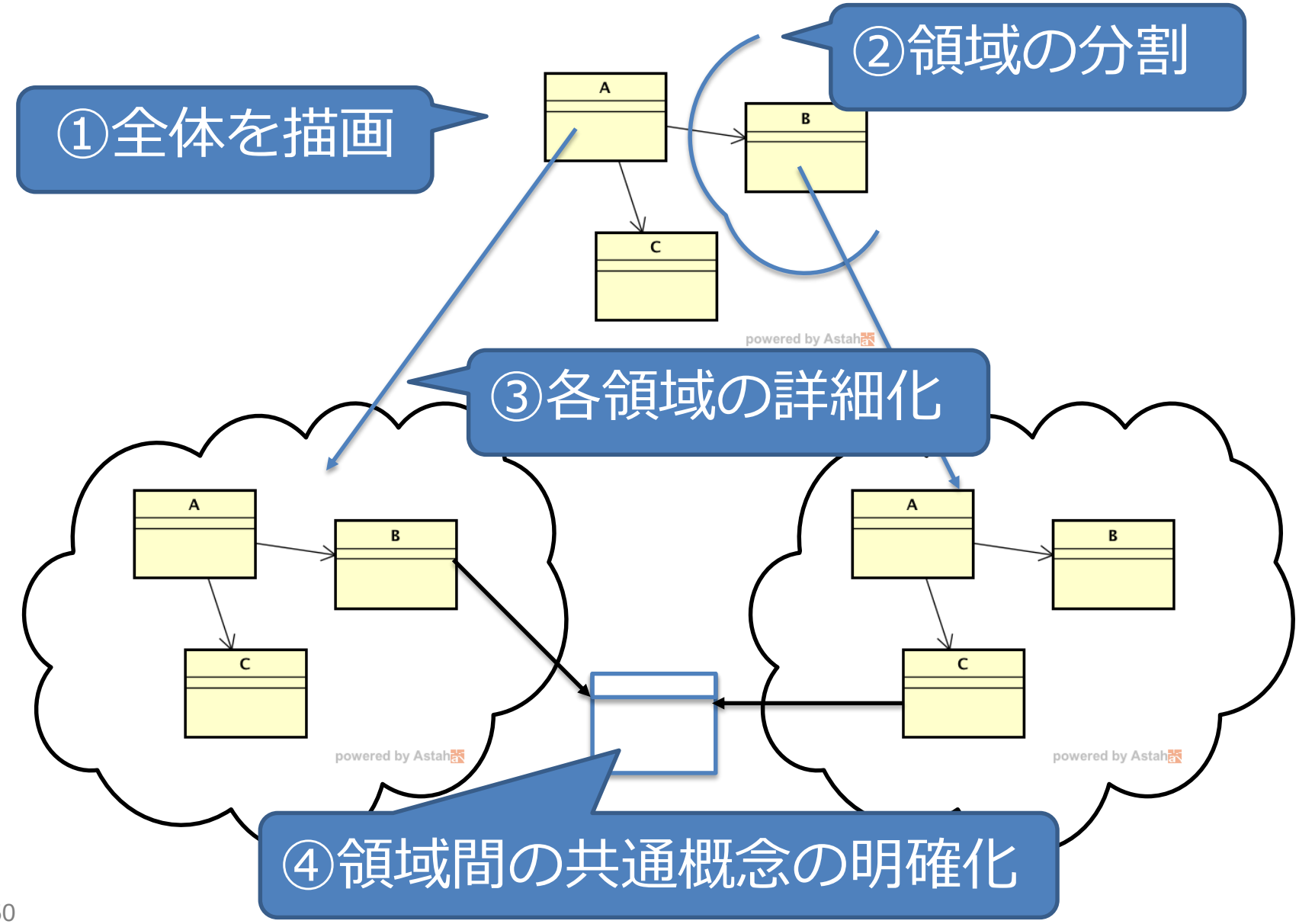
- ✓欲しいものは**全体感**（概念モデル）
- ✓全体が見えることで得られるものは多い
  1. 用語の理解・整理
  2. サブシステムへの分解
  3. コア（関心事）の抽出

アジャイル開発での問題点

UMTP
UML Based Modeling Technologies Promotion

- 「大きな泥団子」のまま進めてしまうことがある
  →分かっているつもりが大きな歪みとなって
  システムに打撃を与えてしまう
- 自分の領域は詳しいが他の領域は興味がない
  →情報が分断されて良いやり方への糸口がない
- 動かないと分からない？
  →動かす前に分かることもあるが、それを行わず
  動かせる具象レベルの世界で理解しようとして
  も時間が足らない。結果としてムダなものを作
  ってしまった後で、ビッグバン的にしか学べな
  い

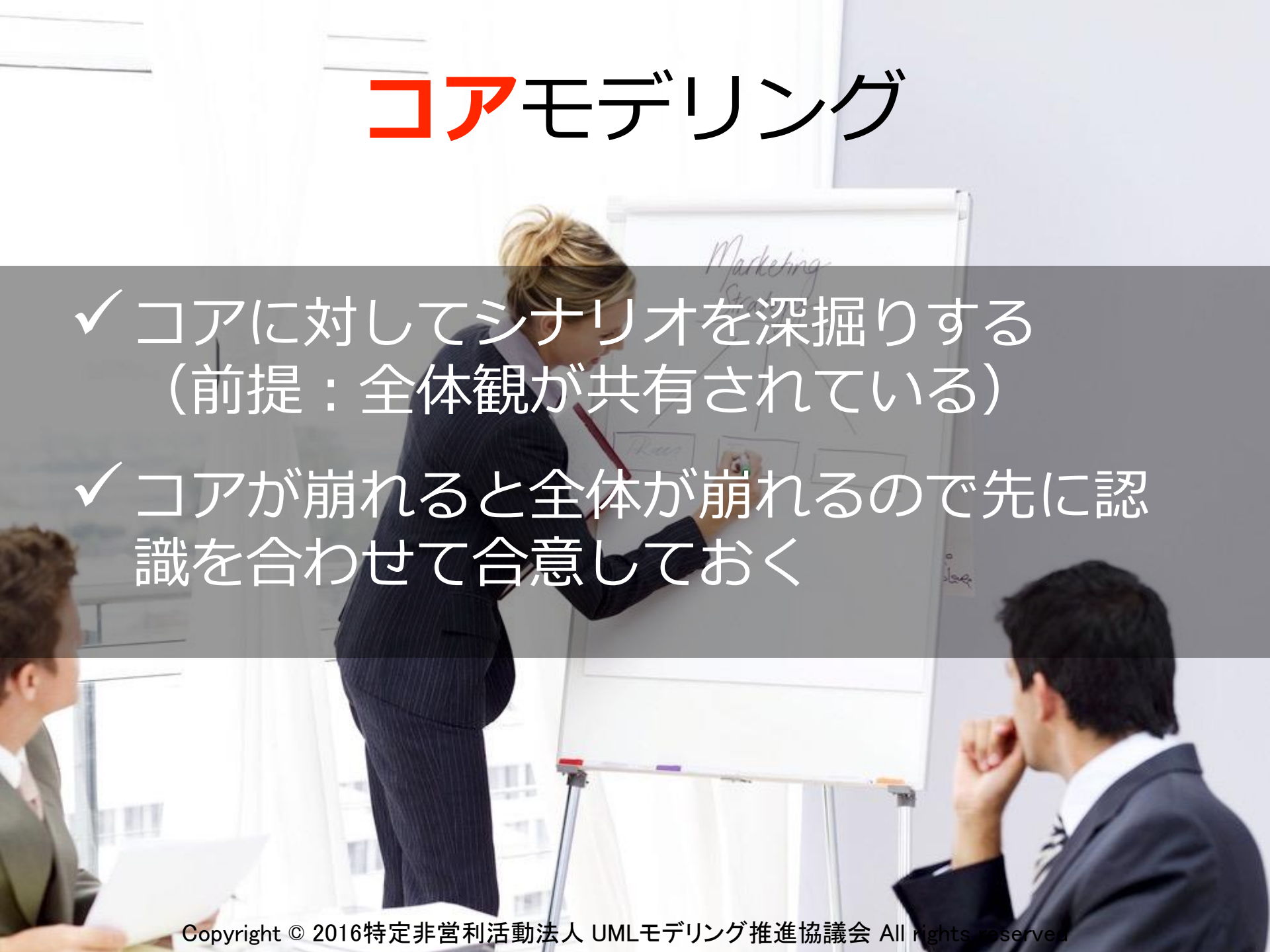
イメージ



全体感

- 関連を持った用語集
  →業務の言葉を明らかにしつつ、
  　言葉の関連に注目する。
  　（抽象度をあわせる。関連を見出す）
- 業務領域ごとの分割
  →用語を使われる業務毎に分割する。
- コンテクスト境界
  →業務領域によって同音異義語や同義語
  　があり、その存在を明確にする

コアモデリング



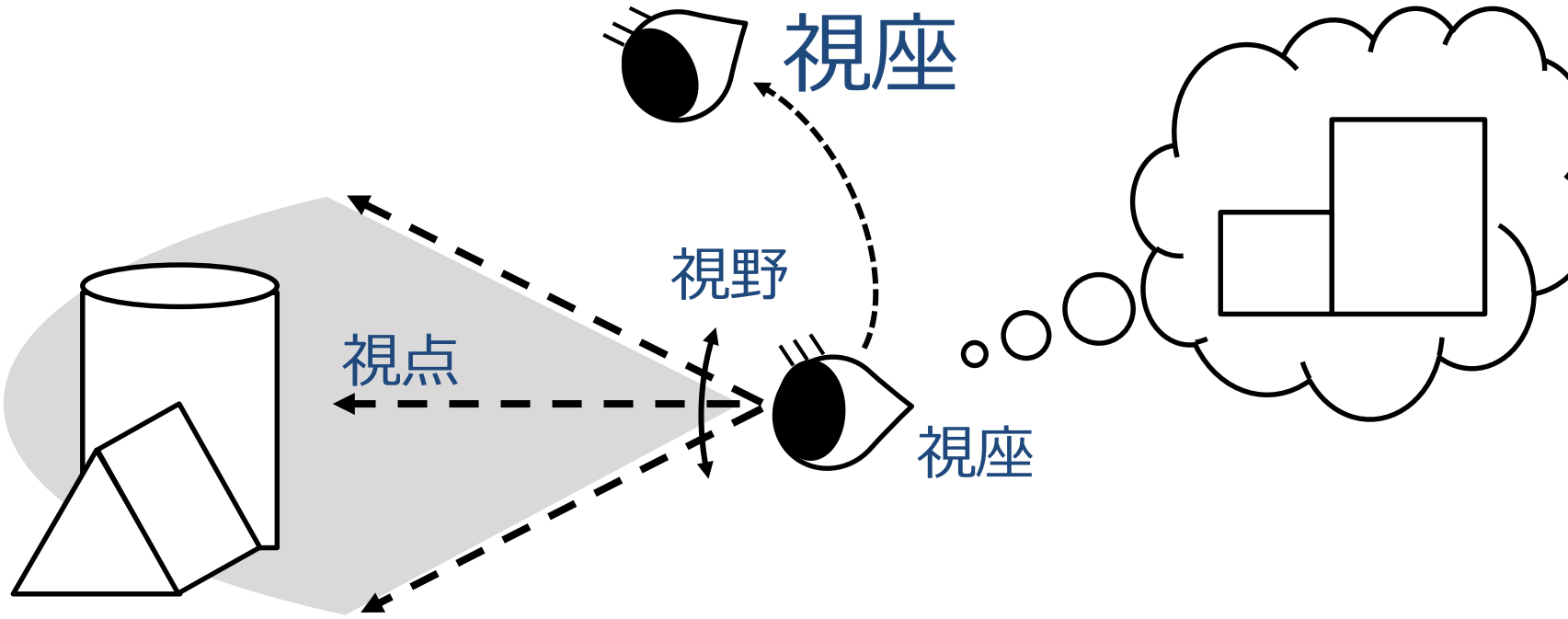
- ✓ コアに対してシナリオを深掘りする（前提：全体観が共有されている）
- ✓ コアが崩れると全体が崩れるので先に認識を合わせて合意しておく

目的と解決策

- 目的と解決策を分けて考える
  - **トップダウンアプローチ**
    - ○ ユーザのシステムへの要求から考える
    - ○ 目的がはっきりすると解決策も出しやすい
    - × ただ目的は最初から分かる訳ではない
  - **ボトムアップアプローチ**
    - ○ 要素間の性質から上位概念を探す
    - ○ 必要な要素は簡単に集まる
    - × 上位概念まで昇華することは難しい

視座・視点・視野

- 視座：モデルをどの立場から見るか
- 視点：モデルをどのように見るか
- 視野：モデルに表現する範囲はどこまでか

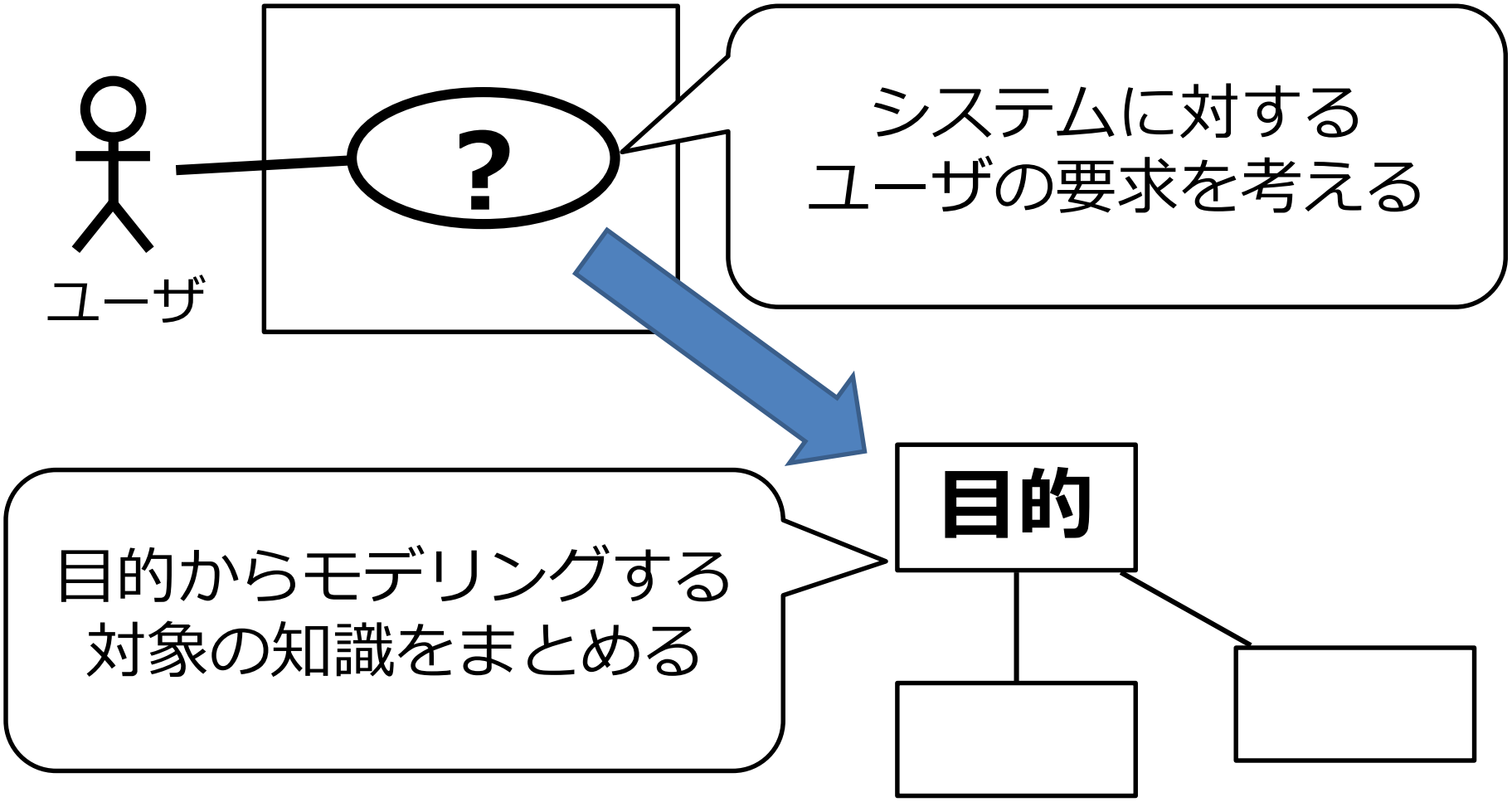


モデリングを**実践**して気付く点

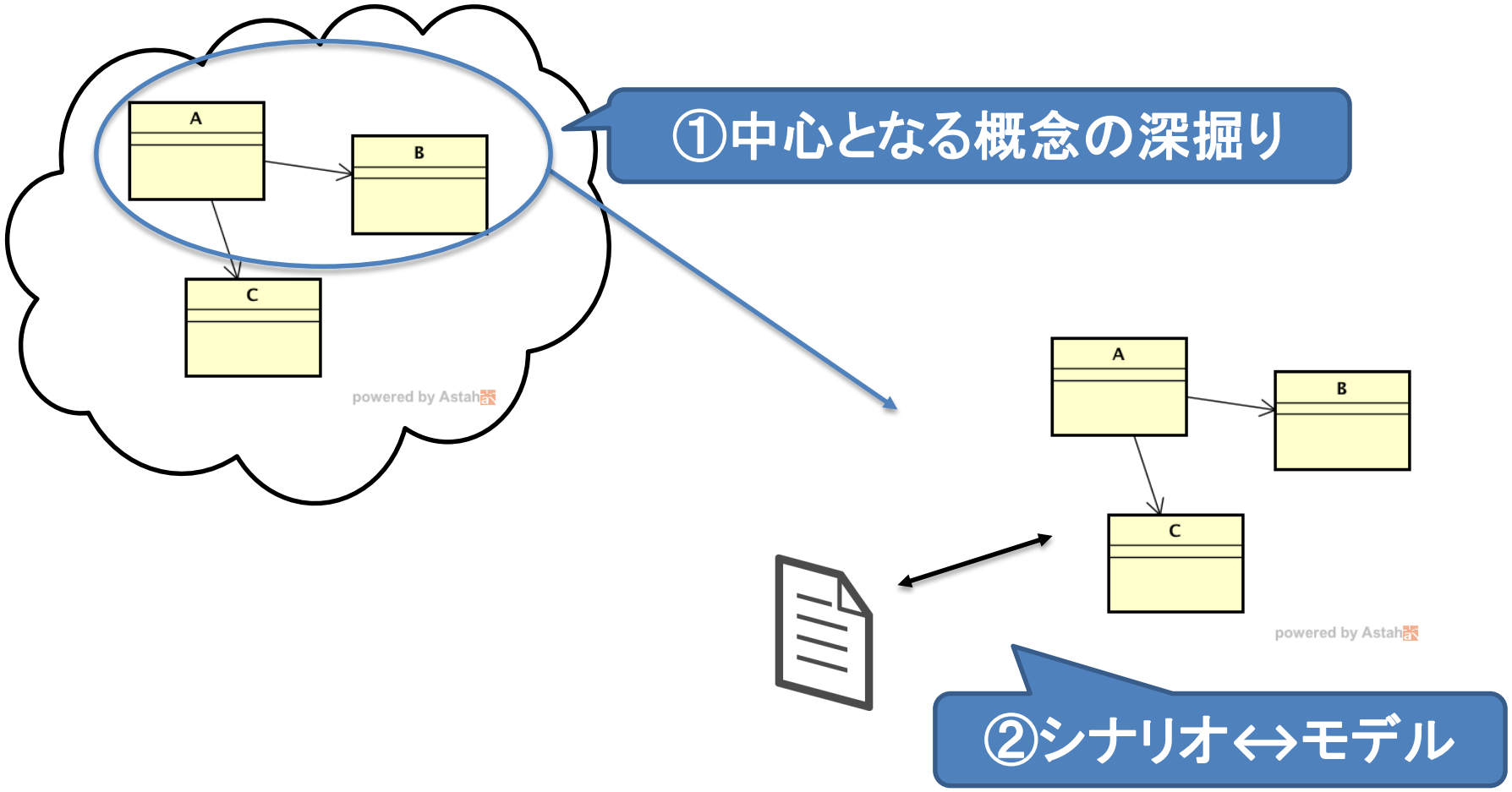
UMTP
UML Based Modeling Technologies Promotion

- 作る対象の目的と価値
  - 目的や前提を明らかにする
    - 特に視座の部分
  - モデリングする範囲を合意する
    - どの視点の話をしているのか？
    - 関心事は何で、どこまでコストを掛けるのか？
      - 説明責任は開発者
      - 設計のメリットとデメリットを示す
      - 見ることができる範囲は見ておく

進め方　例



イメージ



デザイン問答

すべてのモノの形や仕組みには**理由がある**

画像引用：デザイン問答 http://www.nhk.or.jp/design-ah/design-mondou/

伝えるのは「質」の話

- パタンを見出すこと

http://blog-imgs-47-origin.fc2.com/i/t/o/itokaya/IMG_5160-horz.jpg

モデルは捨像

- この世の森羅万象は表せない
- でも、「パタン」を見出すことで「木を見ながら森を見ることができる」
  - Ex) 門

[入口での転換(112)]より
<より大きなパタンとの結合>
…どんな建物や複合建物の場合でも、すでにその主入口の大まかな位置は分かっている－[大きな門口(53)]から敷地への門、[見分けやすい入口の集まり(102)]から各建物への玄関、そして玄関。いずれにしろ、入口は「外部」－公的世界－からさほど公的でない内部世界への転換をつくり出す。

http://fashionjp.net/creatorsblog/fujita/2010/08/post-89.html

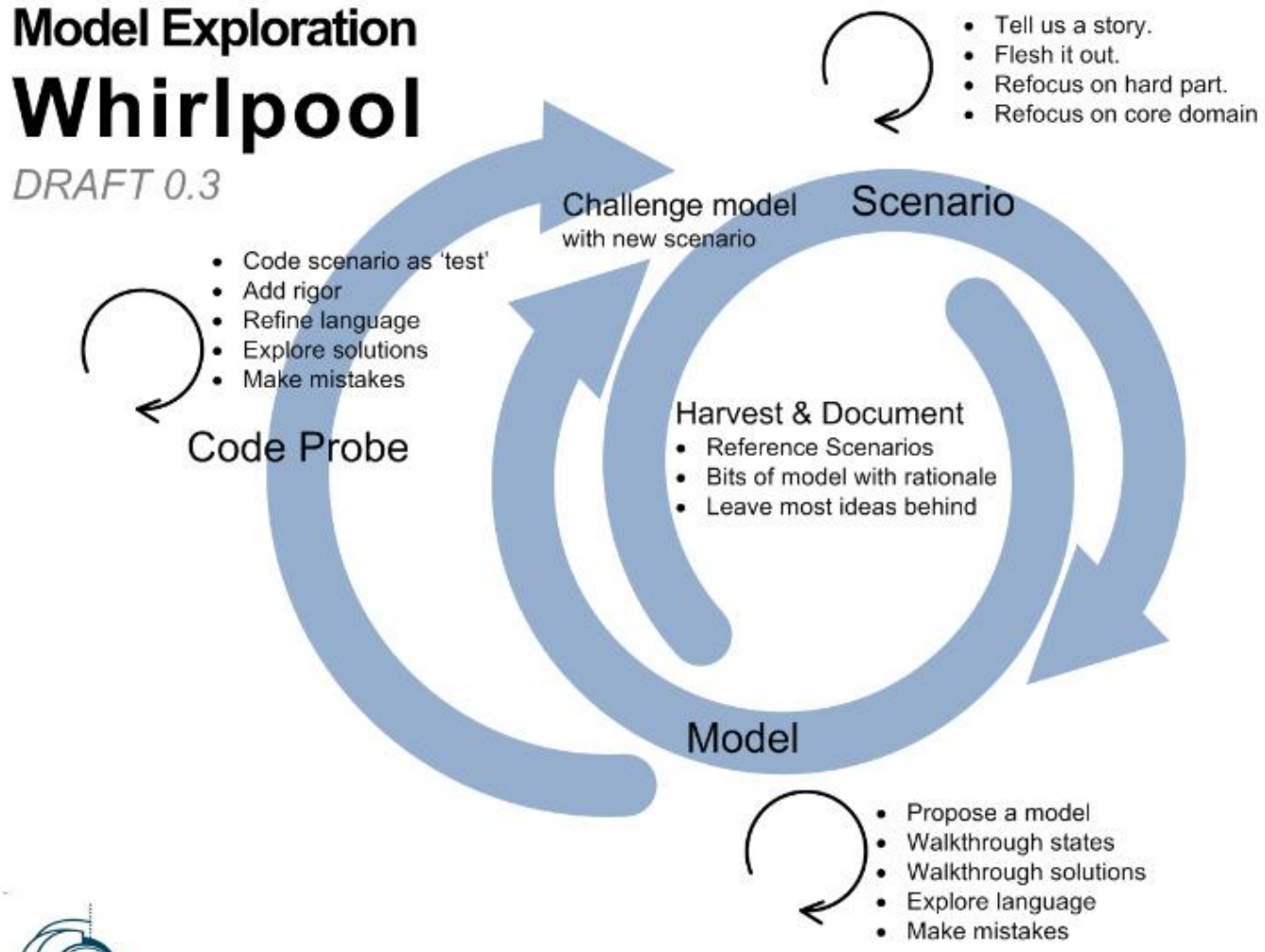
3分クッキング

- コアモデリングが進むとその場で考えるライブモデリングは厳しくなる場面が出てくる
  →ある程度は分析、想定を立ててモデルを作って持っていく方が良い
  - 予想できなかったことを特に深掘り
  - 「ない」ことを確認
  - fragileにしない

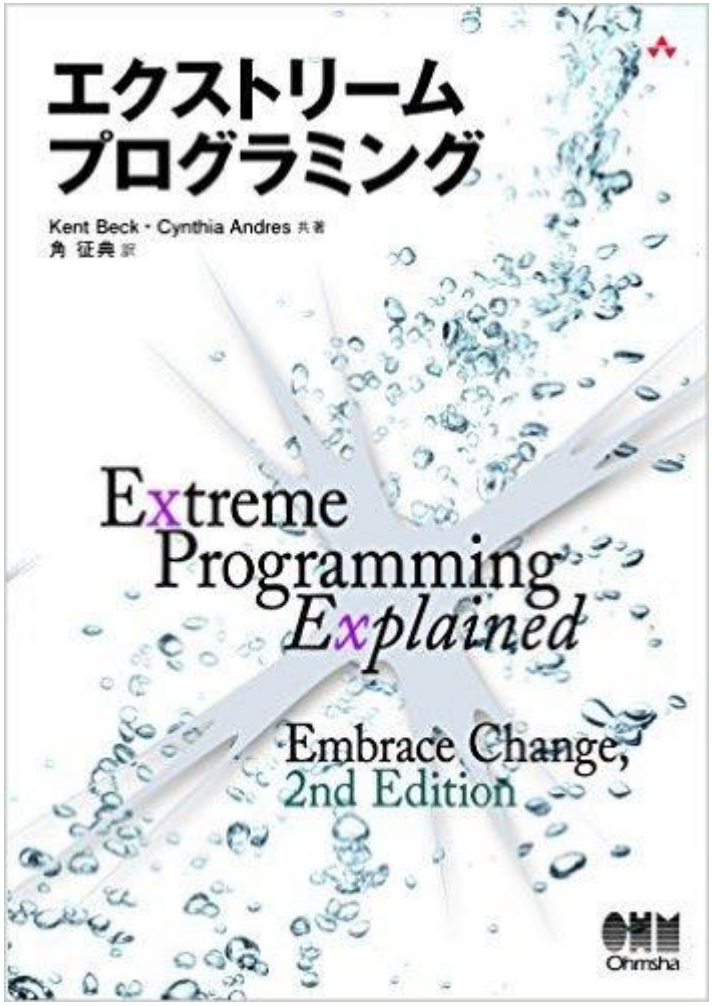
フィードバックモデリング

- ✓動くソフトウェアからモデル・シナリオへのフィードバックを回す

DDDを通して気付く学びのループ



XP実践していますか？



Amazon：http://www.amazon.co.jp/dp/4274217620/

XPとモデリング

アジャイルモデリング by Scott Ambler

“モデリングは**XPの1つの要素**である**”**

“リファクタリングと**テストファースト開発**という**XPのプラクティス**は、一般的に従来型の**モデリングプロセス**に結び付けられている**「きれいな設計を促す」「コードを書く前に設計をよく考える」**という２つの重要な目的を達成するために役に立ちます**”**

アジャイルモデリング XPと統一プロセスを補完するプラクティス　第16章、第17章
http://www.amazon.co.jp/dp/4798102636

UMTP
UML Based Modeling Technologies Promotion

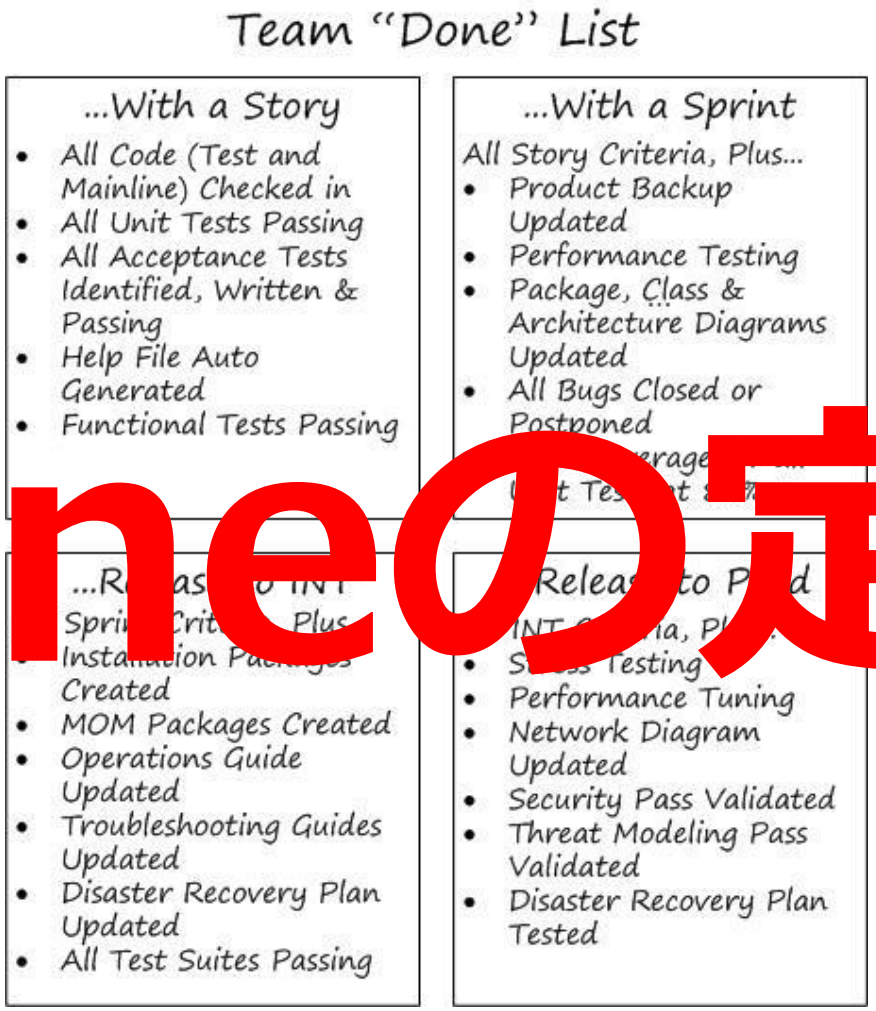
~~Big Design Up Front~~

“論点となるのは、設計を**するかどうか**ではなく、設計を**いつするか**である”

- **LDUF／ENUF**
  設計の戦略と設計のシンプリシティ
  - 対象者に適している
  - 情報が伝わりやすい
  - うまく分割されている
  - 最小限である

エクストリームプログラミング 第14章 設計：時間の重要性

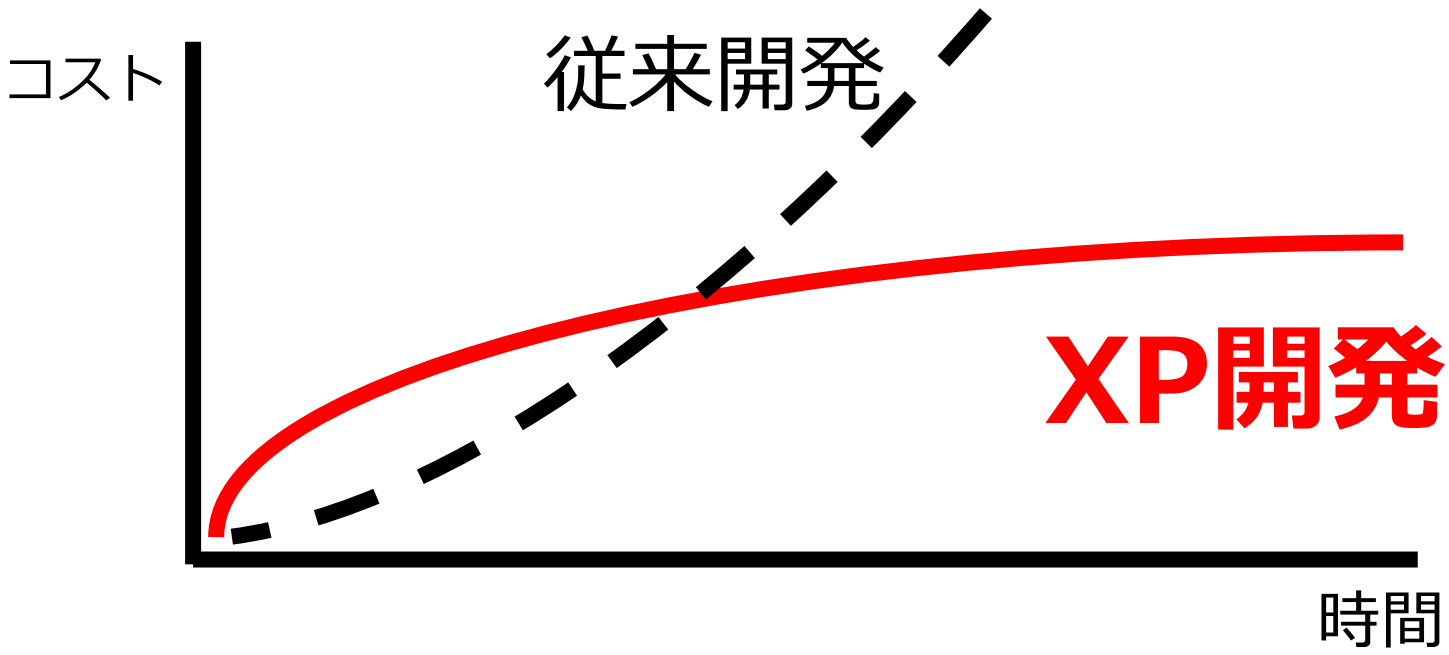
How Do We Know When We Are Done?



Doneの定義

How Do We Know When We Are Done?
https://www.scrumalliance.org/community/articles/2008/september/how-do-we-know-when-we-are-done

今日のキーワード



変更のコストを一定に保つ

書籍：エクストリーム・プログラミング ソフトウェア開発の究極の手法
P.23 図 3 変更のコストは時間とともに劇的に上がらないことがある

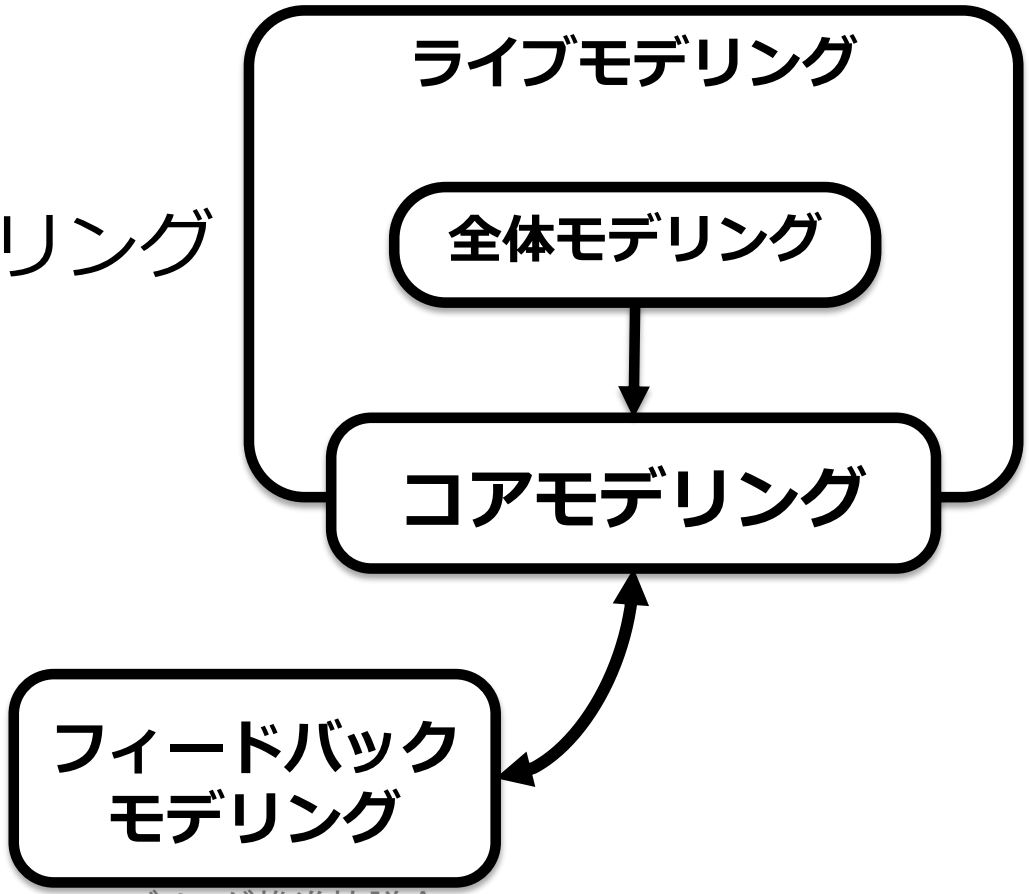
小さくそして速く

1. シナリオを探索する（理解）
2. モデルを小さく作る（発見）
3. モデルをコードにする（実現）
4. シナリオを実行する（検証）
5. 動くコードからフィードバックを得て、小さくイテレーティブに開発することで**学んで**いける

書籍：IMPACT MAPPING　P28-29 反復デリバリによる改良

今日話したこと

1. ライブモデリング
2. 全体モデリング
3. コアモデリング
4. フィードバックモデリング



UMTP
UML Based Modeling Technologies Promotion

ご清聴ありがとうございました

ガイドライン目次（1／2）

ガイドライン目次（2／2）