アジャイル開発での UML適用ガイドライン（仮称）

UMLモデリング推進協議会(UMTP)
アジャイル開発部会

平成26年 2月 24日

中菱エンジニアリング㈱　古川　剛啓
ネットイヤーグループ　小倉　英一郎

目次

- ガイドラインの構成
- １章　まえがき
- ２章　なぜモデリングなのか
- ３章　モデルとアジャイルプラクティスを有効に活用する
- ４章　アジャイルソフトウェア開発のための Tips

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㈱

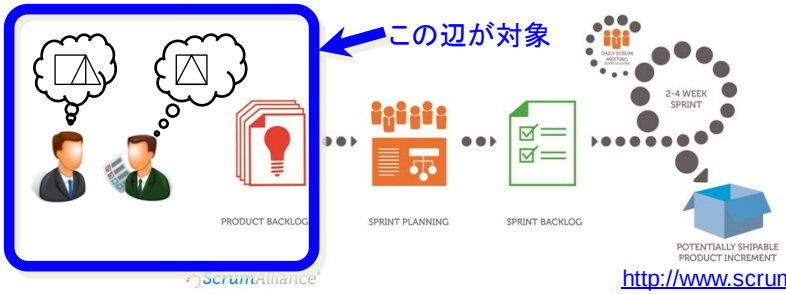
UMTP アジャイル開発部会メンバ
UMTP L3モデラー
OMG Advanced
認定スクラムマスター
名古屋アジャイル勉強会スタッフ

ガイドラインの構成

- 以下の章で構成
  - ✓1章　まえがき
  - ✓2章　なぜモデリングなのか
  - ✓3章　モデルとアジャイルプラクティスを有効に活用する
  - ✓4章　アジャイルソフトウェア開発のためのTips

1章　まえがき

- ガイドラインの対象者
  - ✓ UMLやアジャイルソフトウェア開発に興味がある人
- ガイドラインの読み方
  - ✓ 各章は独立しているので、どの章からでも読める様になっています
- ガイドラインの背景
  - ✓ UMLとアジャイルプラクティスを組み合わせるとより分かり易くできます
  - ✓ 特に組み込みソフトウェアはUMLとの親和性が高く、効果が高いと考えています
  - ✓ 組み込みソフトウェアだけに限った内容では無く、ソフトウェア開発全般に適用できます



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

2章　なぜモデリングなのか

この章では、モデリングに関する以下の考え方を示しています。

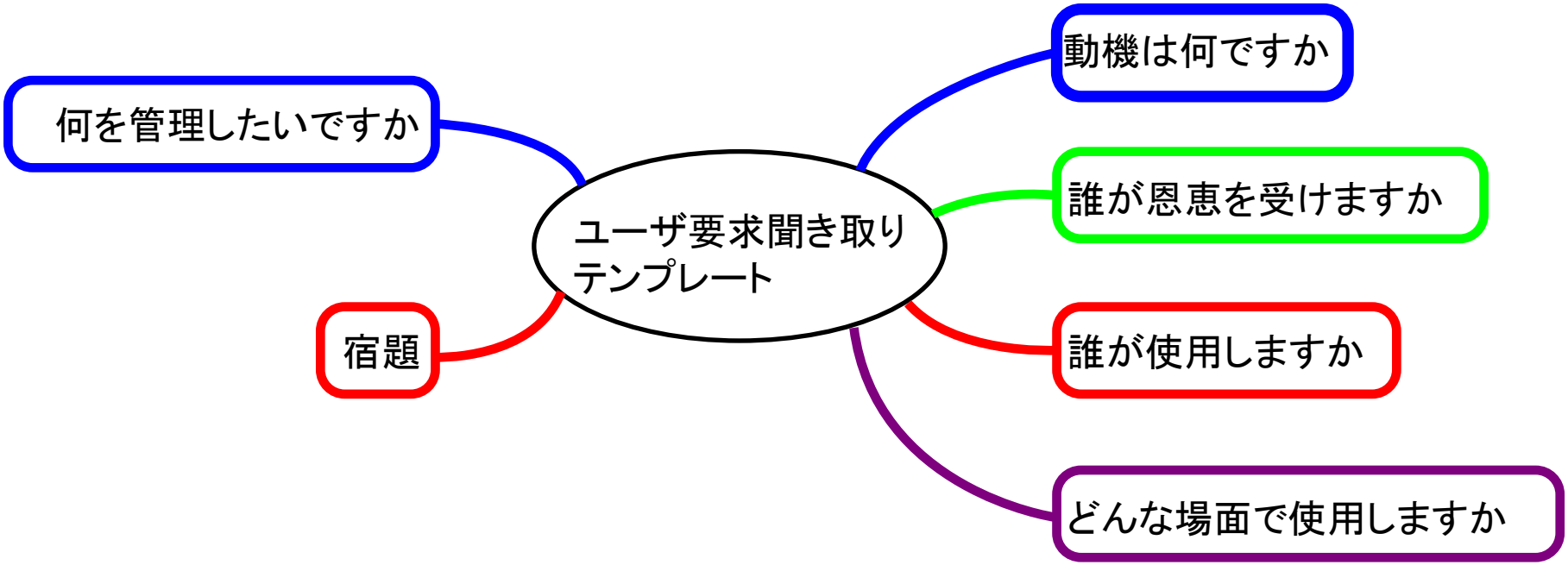
- モデルを使うことのメリット
- モデリングツール使用の要否
- モデリングツールの選定で考慮すること

3章　モデルとアジャイルプラクティスを有効に活用する

http://www.flickr.com/photos/jcraveiro/114311213

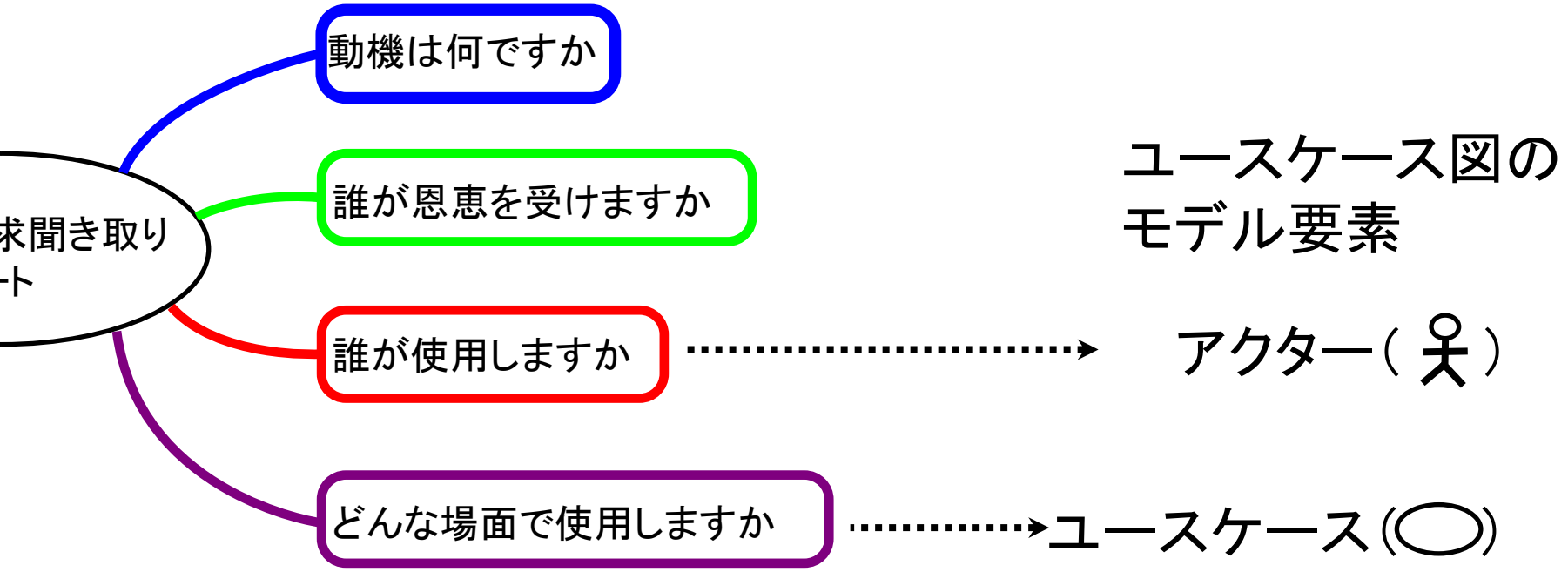
マインドマップを使って要求を聞き取る

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

マインドマップからユースケース図を作成する

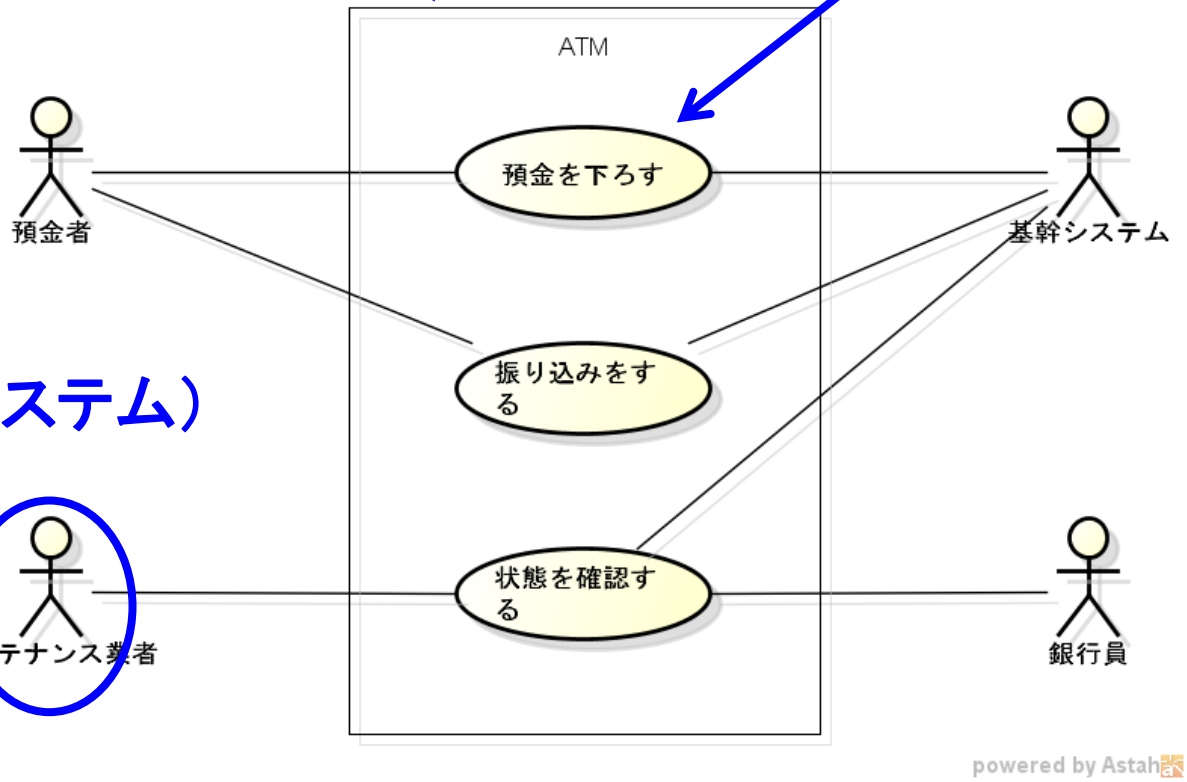


ユースケース図(例)

システムバウンダリ
（開発対象システムと外との境界）

ユースケース
（開発対象機能）

アクター
（人や外部システム）



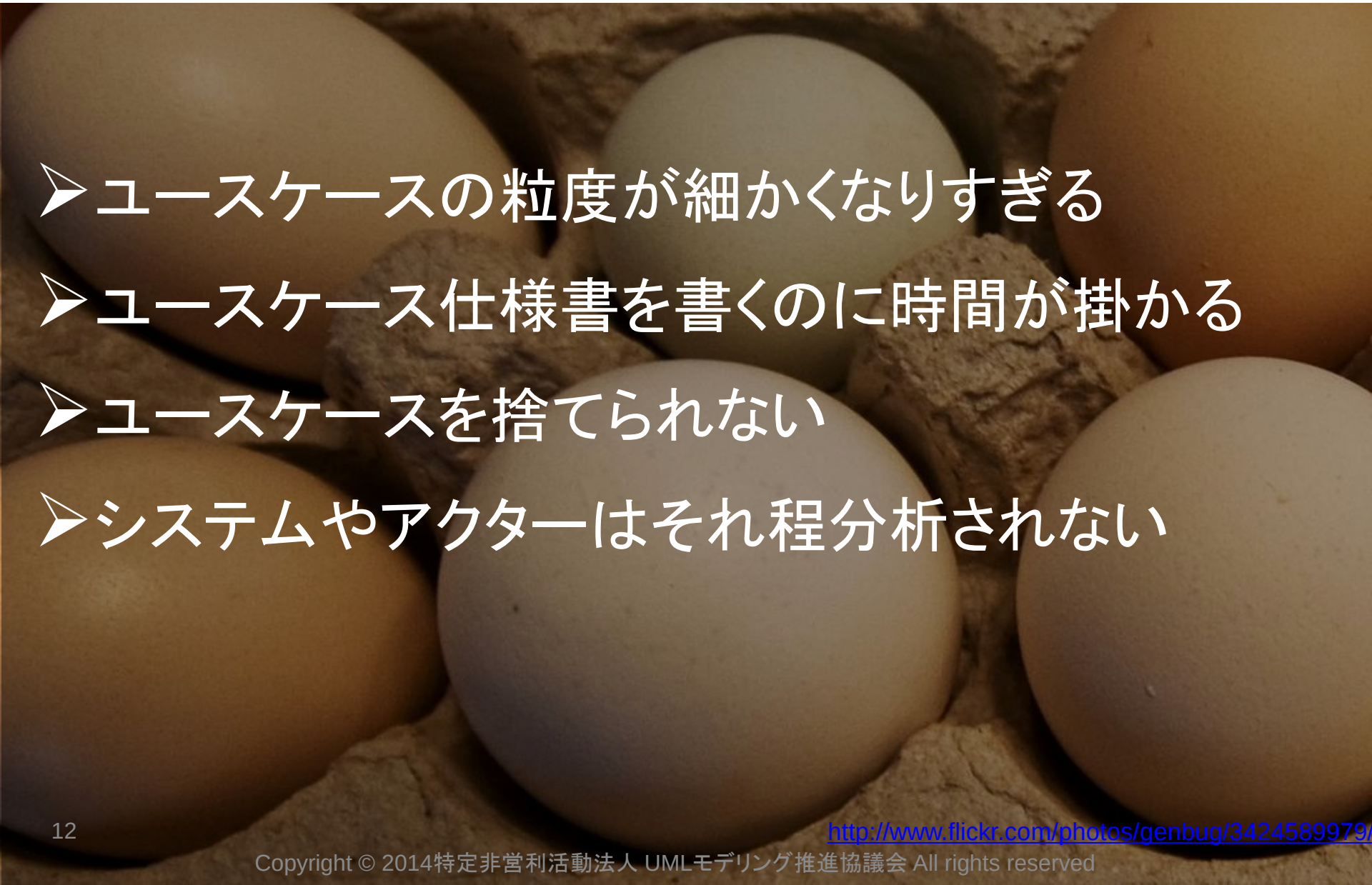
ユースケース図を作成する際のポイント（教科書的に）

- ユースケースの粒度を揃える
  ⇒ 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仕様書を書く
  ⇒ ユースケースの仕様を明確にする

- アクターは「役割」を表す
  ⇒ 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・・・



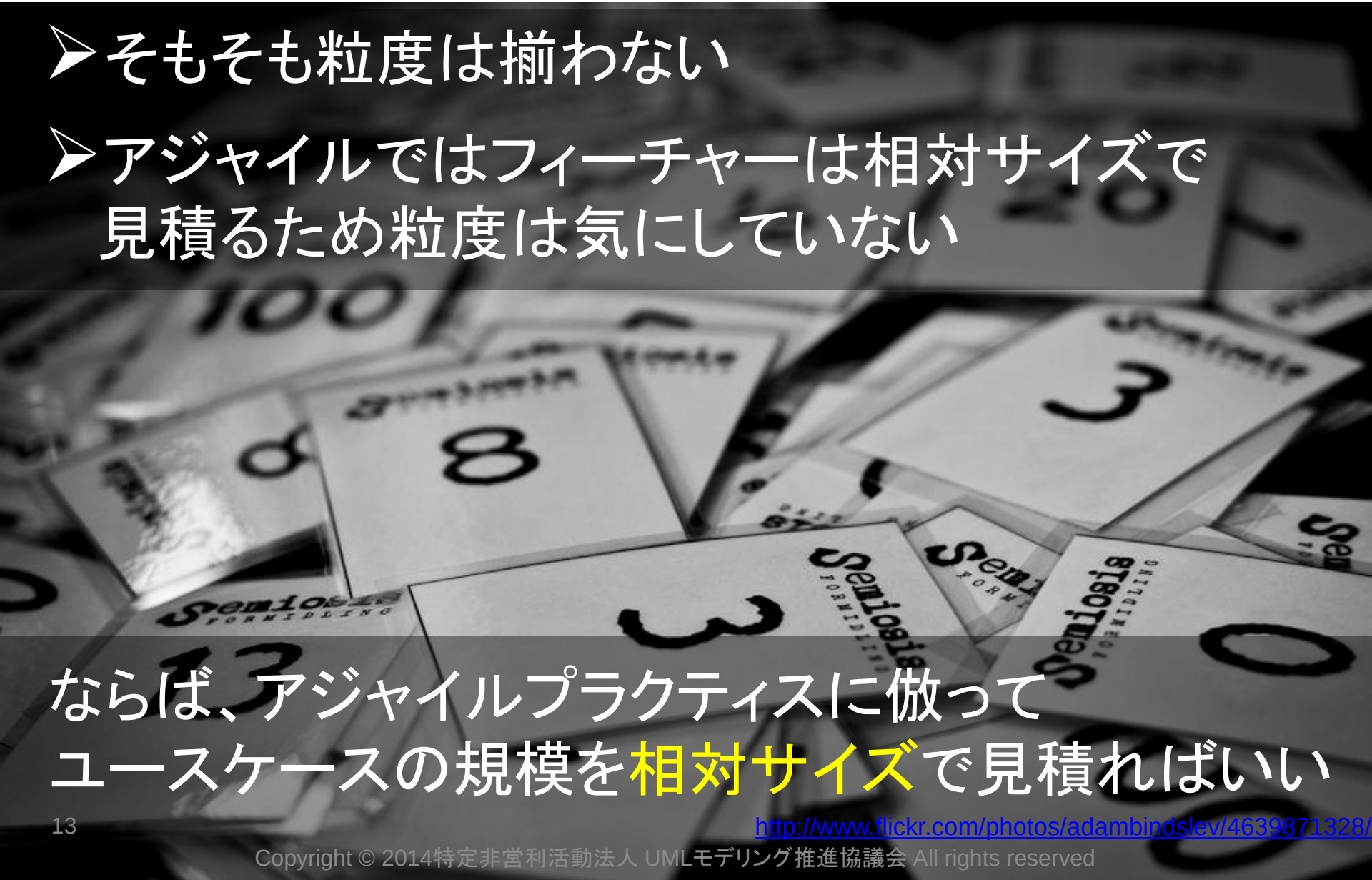
- ユースケースの粒度が細かくなりすぎる
- ユースケース仕様書を書くのに時間が掛かる
- ユースケースを捨てられない
- システムやアクターはそれ程分析されない

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

ユースケースの規模を相対サイズで見積る

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短い文章で簡潔に示したもの

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
- フィーチャーの粒度によらない

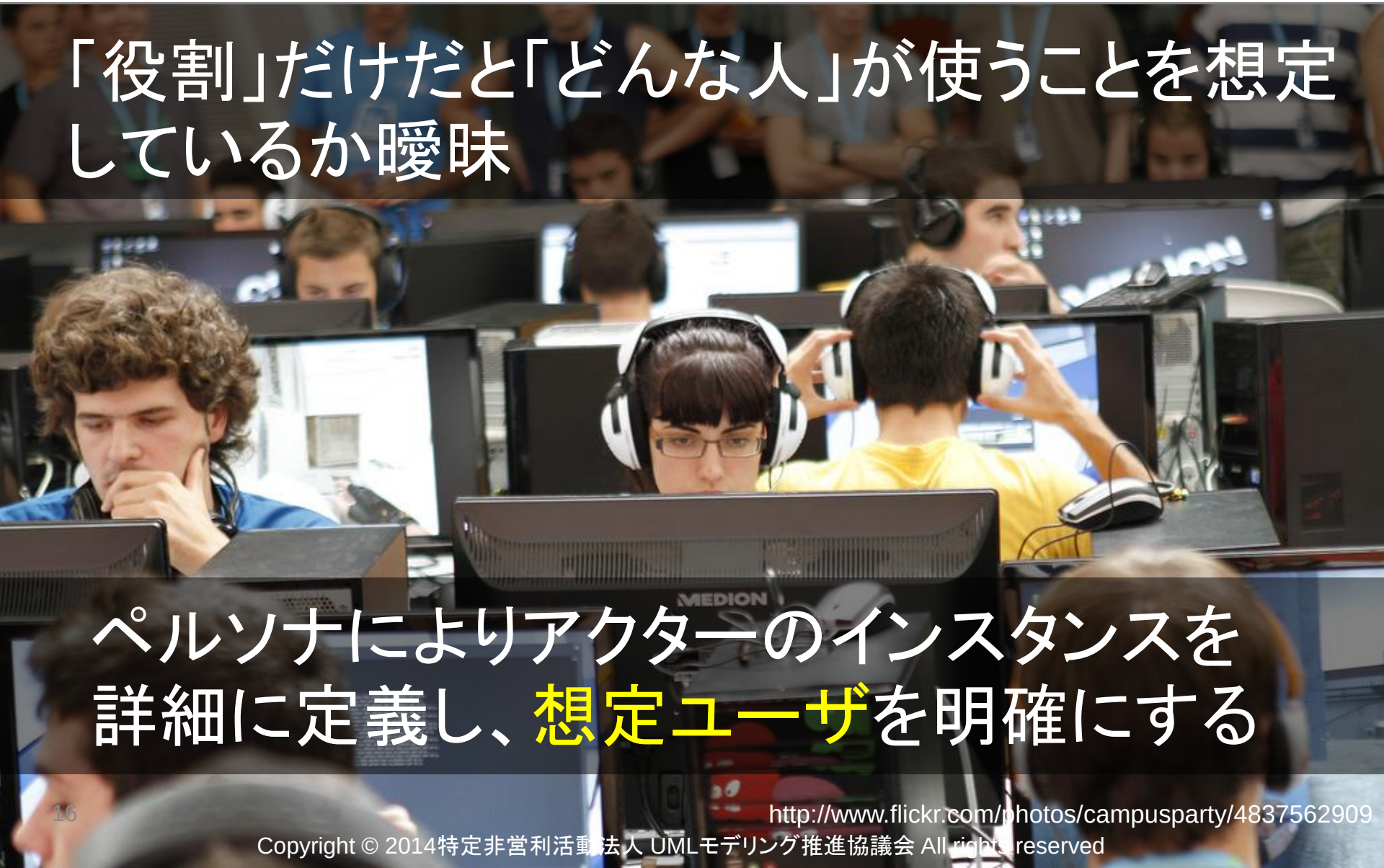
ユースケース仕様書は適切なタイミングで必要な量だけ書く

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

ユースケース仕様書とユーザーストーリーの記述例

ユースケース仕様書	ユーザーストーリー
目的 　システムにログインする 事前条件 　・認証システムと正常に通信できること 　・認証システムにアカウントが作成され、パスワードが設定されていること 事後条件 　・システムにログインできること ユースケースシナリオ（正常処理） 　1. ユーザは、アカウント名とパスワードを入力する 　2. システムは、アカウント名とパスワードが正しいことを認証システムに問い合わせる 　3. 認証システムは、アカウント名及びパスワードが正しいことを確認する 　4. 認証システムは、アカウント名とパスワードが正しいことを応答する 　5. システムは、初期画面を表示する 　　：	ユーザとしてシステムにログインするための機能が欲しい、それは個人毎にパーソナライズされた機能を使用するためだ。

ペルソナを使ってアクターのインスタンスを定義する

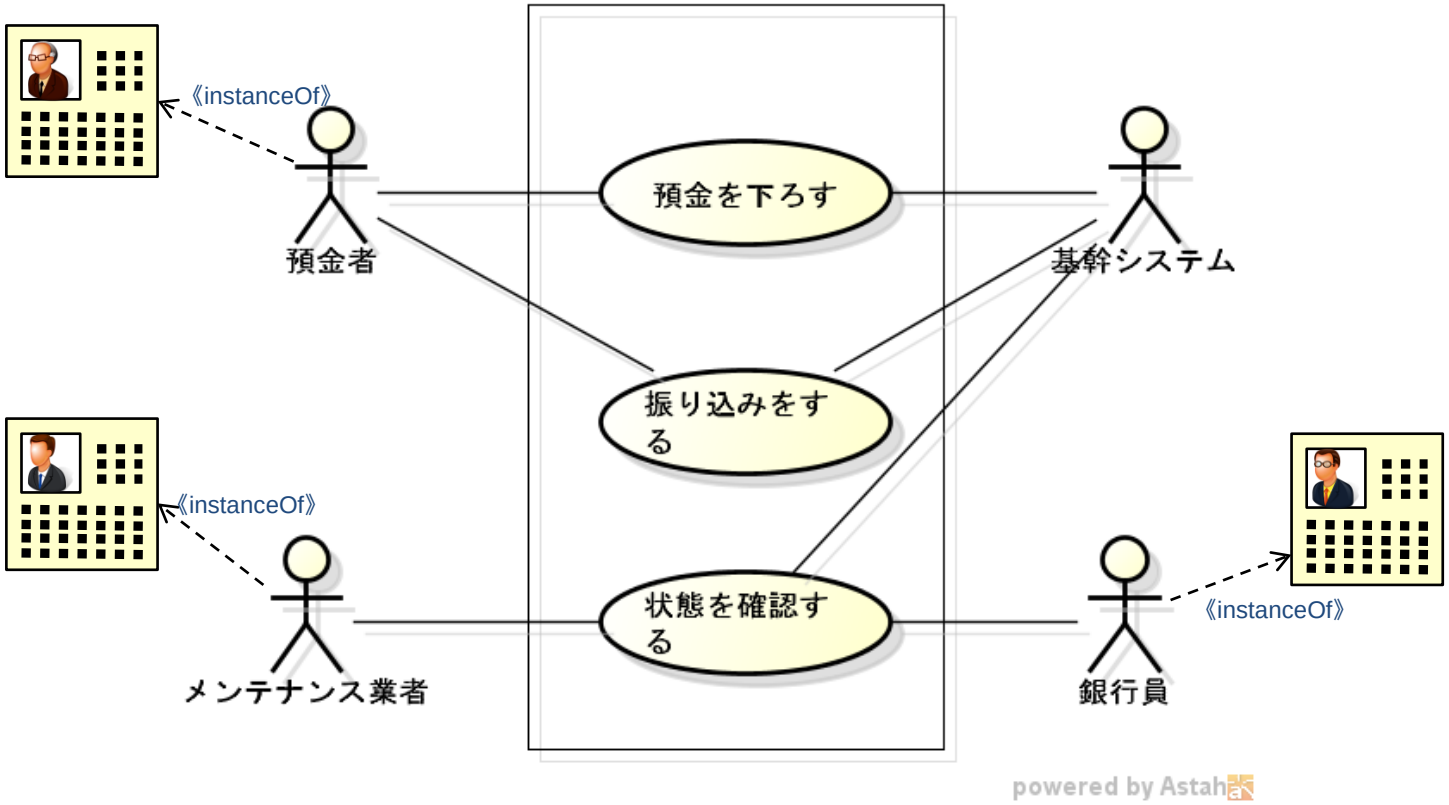


「役割」だけだと「どんな人」が使うことを想定しているか曖昧

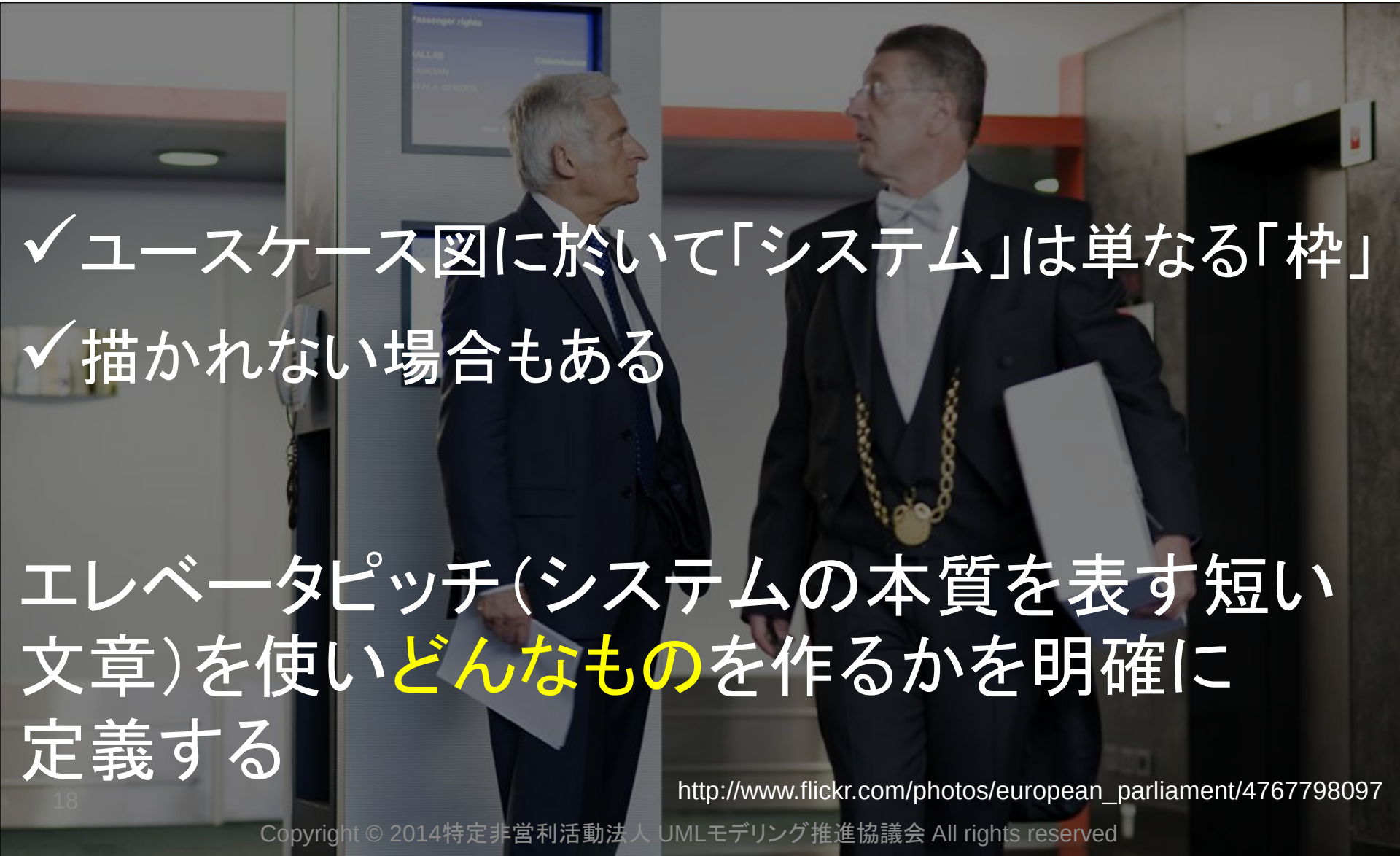
ペルソナによりアクターのインスタンスを詳細に定義し、想定ユーザを明確にする

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

ペルソナを追加したユースケース図（例）



エレベータピッチを使ってシステムを説明する



- ✓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

エレベータピッチの例

UMTP
UML Based Modeling Technologies Promotion

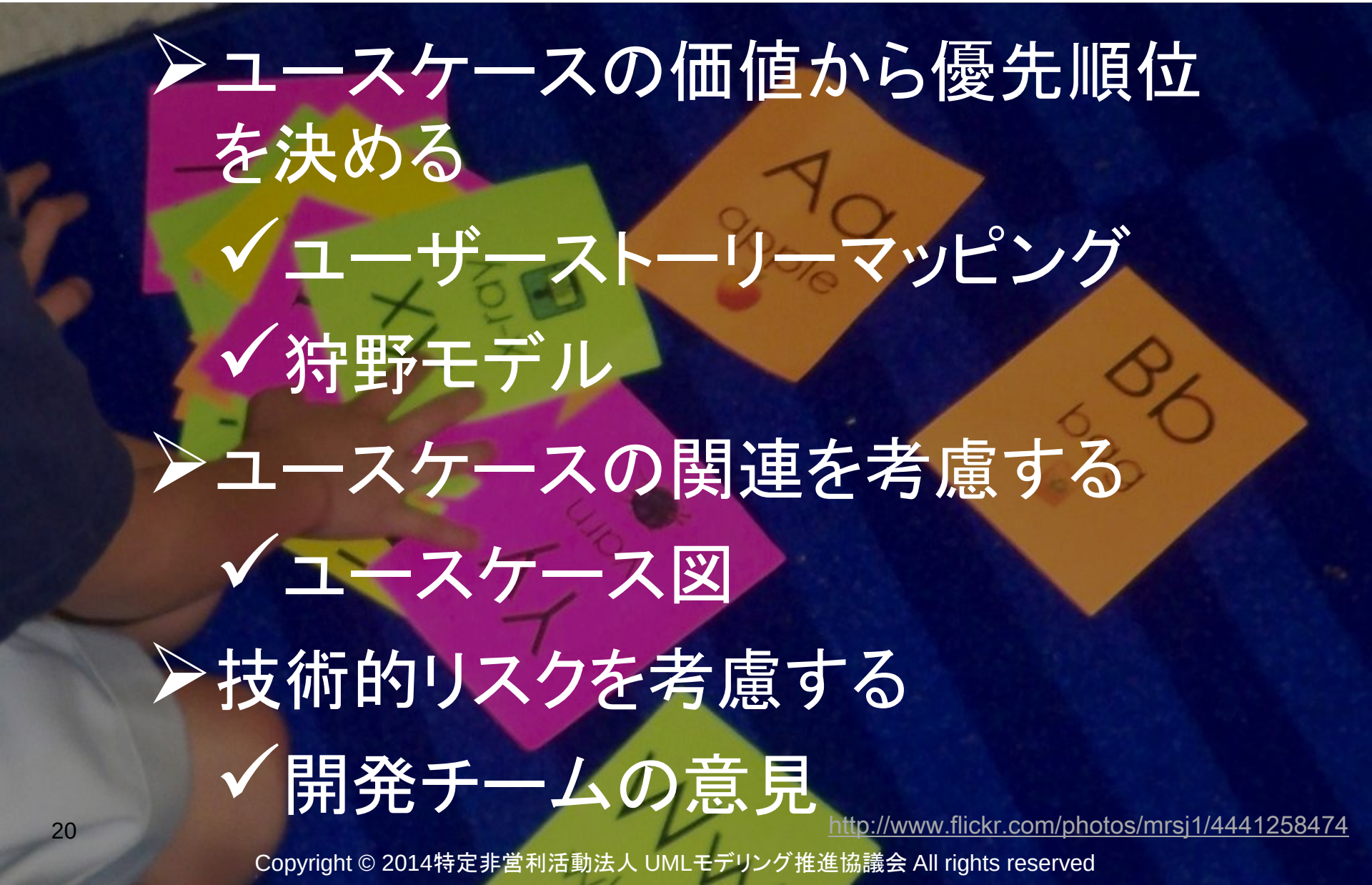
エレベータピッチのテンプレート

「潜在的なニーズを満たしたり、抱えている課題を解決したり」**したい**
「対象顧客」**向けの、**「製品名」**という製品は、**「製品のカテゴリー」**である。**
これは「重要な利点、対価に見合う説得力のある理由」**ができ、**
「代替手段の最右翼」**とは違って、**「差別化の決定的な特徴」**が備わっている。**

例：
スーパーの来店客のレジ待ち時間のストレスを楽しみやメリットに変えたいスーパーマーケット向けの「待ち時間気にならない君」という製品は、POSシステムの拡張です。
これは、顧客満足の向上と機会損失の低減ができ、
一般的なポイント付与とは違って、顧客のストレスをメリットに変えることができます。

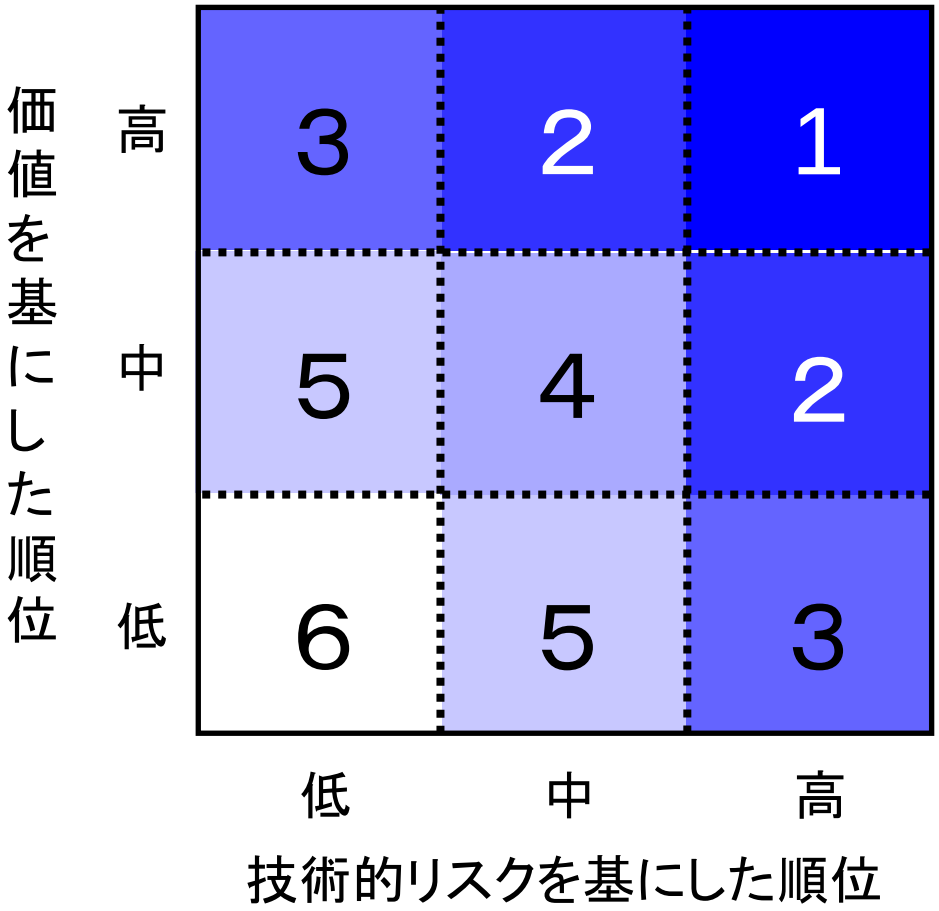
ユースケースの優先順位を決める

- ユースケースの価値から優先順位を決める
  - ✓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
- 技術的リスクを考慮する
  - ✓開発チームの意見



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

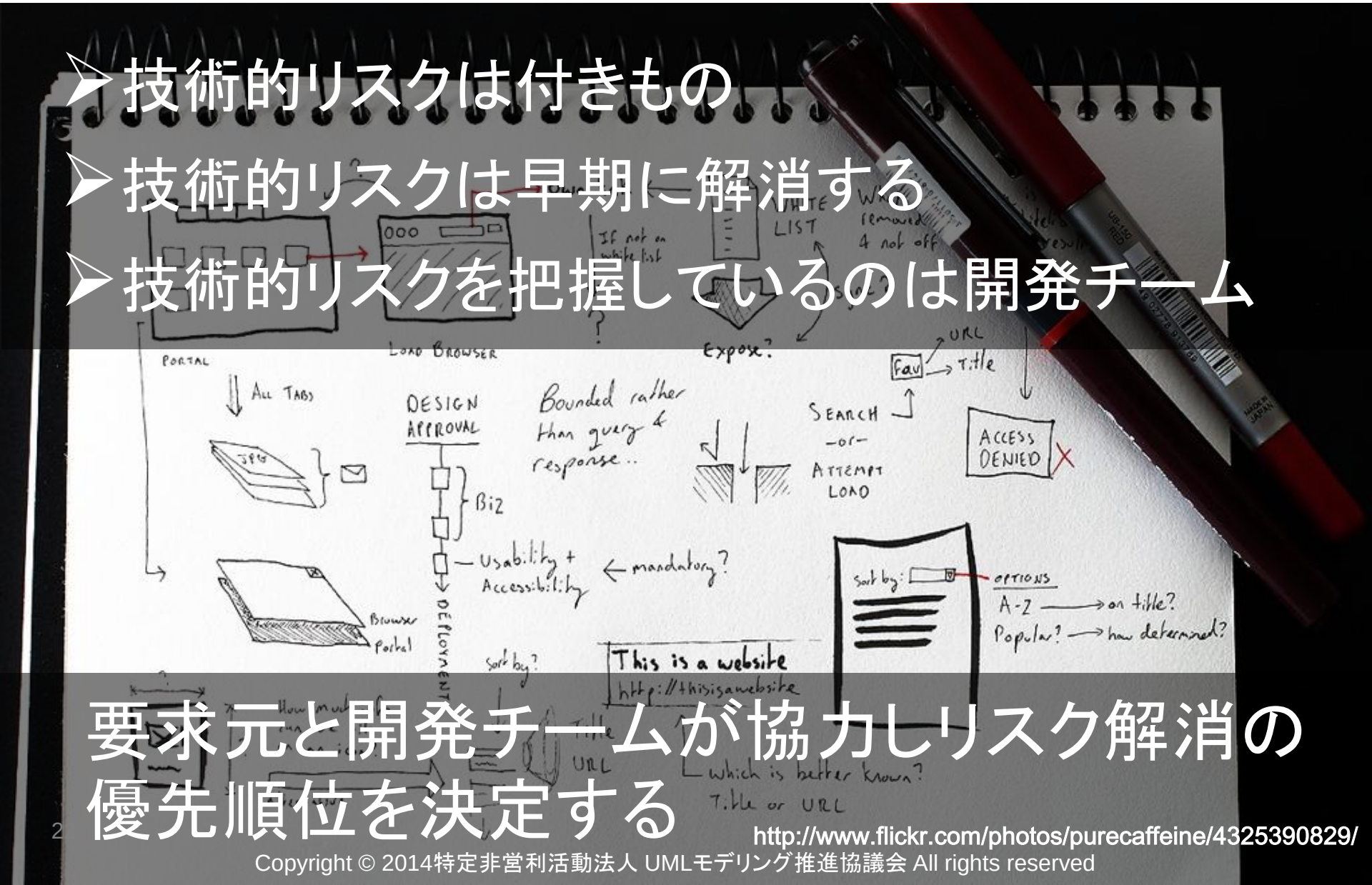
価値を基にした順位と技術的リスクをマッピングする



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

- 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

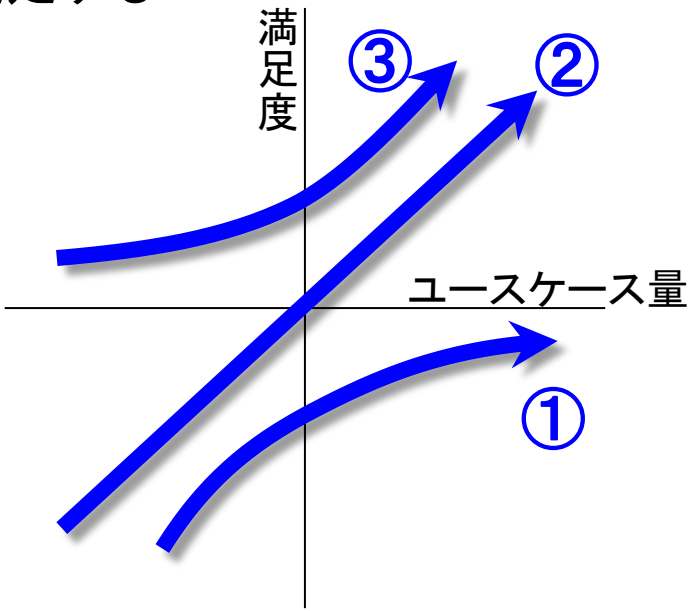
ユーザー（アクター）

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

狩野モデル

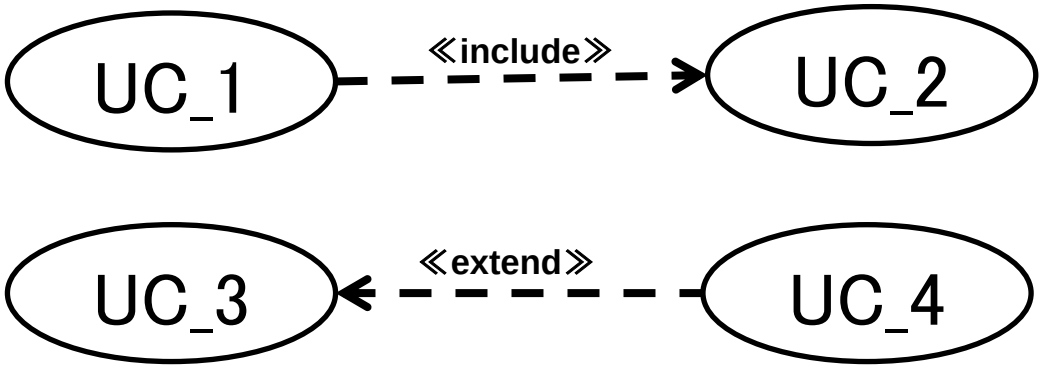
ユースケースをカテゴリー分けし、優先順位を決定する

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



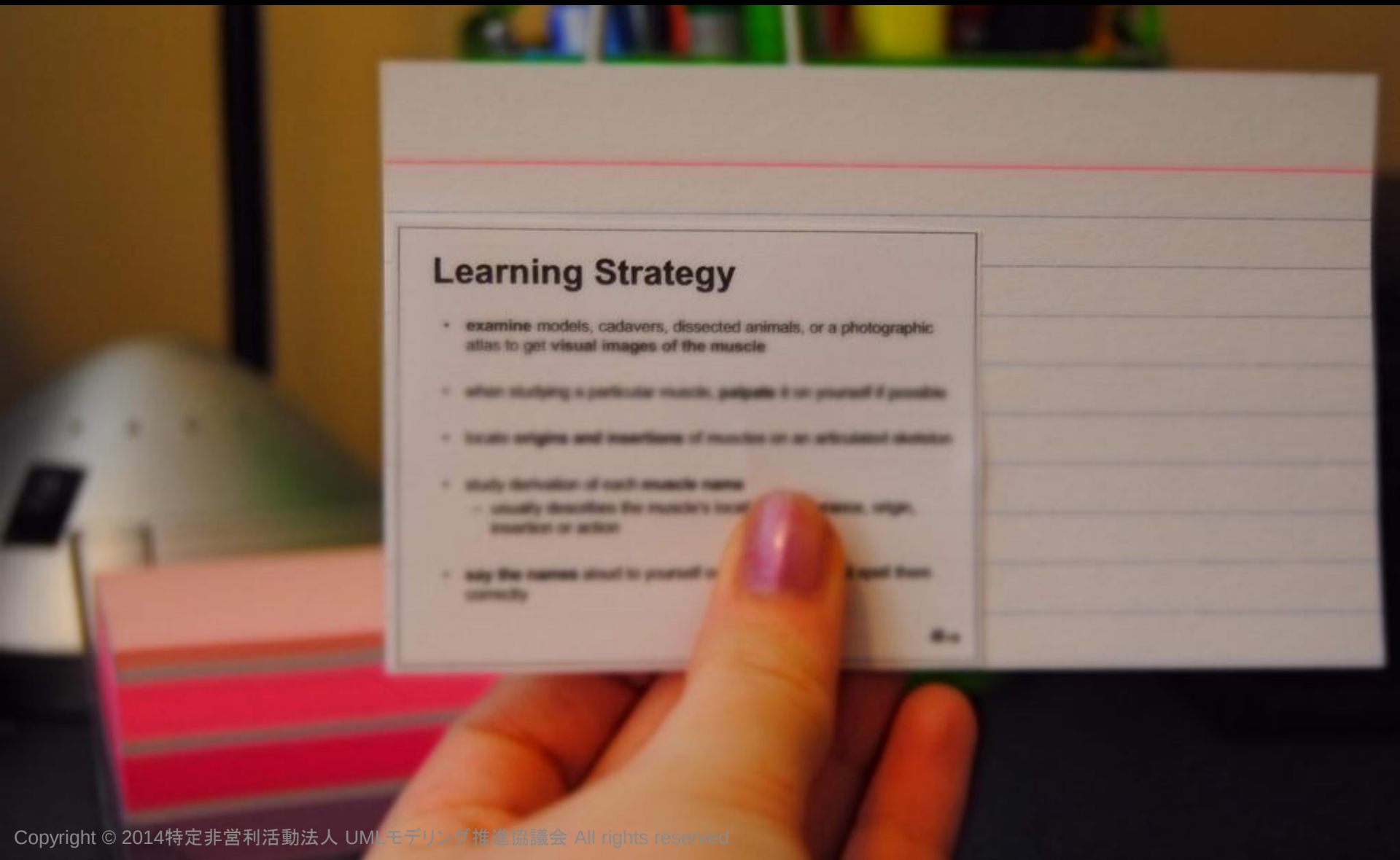
以上のカテゴリーを基に価値を考えると・・・
当たり前のユースケース(①)は全て備える
線形のユースケース(②)をできる限り多く備える
但し、当たり前のユースケースを初期のスプリントで開発しなければならないという事ではない。
魅力的なユースケース(③)は、上記のユースケースを実装した後に時間の許す範囲で対応する。　詳しくは、「アジャイルな見積りと計画づくり」で

ユースケース間の依存を考慮する



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

4章　アジャイルソフトウェア開発のためのTips



自己紹介

UMTP
UML Based Modeling Technologies Promotion

Eiichiro Ogura

Netyear Group
Technologist

Certfied

UMTP L3 Modeler
OCUP Advanced
Scrum Master
Scrum Product Owner

Communities

UMTP アジャイル開発部会
XP日本ユーザーグループ
すくすくスクラム

UMTP L3 MODELER

Professional Advanced

CSM

アジャイルなソフトウェア開発プロセス

様々なプロセスが提唱されている

・リーンソフトウェア開発
・スクラム
・フィーチャ駆動開発
・カンバン
・ RUP

使ってみると、すんなりとはいかない

http://www.flickr.com/photos/pravitnophotography/7548090692

http://www.flickr.com/photos/fatcatimages/8732866175

使ってみると、すんなりとはいかない

・現場で実際に使ってみると…

→想定とは違う行動をする
→意図したものとは違う結果になる
→説明されていない微妙な領域の問題がある

なぜそのようなことが起こるのか

http://www.flickr.com/photos/acornpomegranate/337465007

なぜそのようなことが起こるのか

プロセス・方法論は
すべてをフォローしてくれるわけではない

どうすればよいか

何か困ったことが生じたらTipsを見る

→現場でアジャイルソフトウェア開発を
行うときの参考にする

アジャイルソフトウェア開発のためのTips

UMTP
UML Based Modeling Technologies Promotion

http://www.flickr.com/photos/sidereal/341165583

Tipsの構成

名称：Tipsの名称
目的：どのようなときのためのTipsか
詳細：具体的な説明
関連するTips：あわせて利用できるTips
解説：Tipsの詳細を補足

Tipsの使い方

http://www.flickr.com/photos/tulanesally/5613095629

Tipsの使い方

1. 何かあったら見てみる

「うまくいってない…」

「とりあえず回っているけど、何か足りない」

Tipsの使い方

UMTP
UML Based Modeling Technologies Promotion

2. 全てを使う必要はない

問題解決に使えそうなものを使う

Tipsの使い方

UMTP
UML Based Modeling Technologies Promotion

3.「1つの問題 ＝ 1つのTips」とは限らない

Tipsを組み合わせて、解決策を作り出す

ケーススタディ

http://www.flickr.com/photos/tonyanthony/411805735

ケーススタディ

スクラムを始めたAさん

最初はチームメンバーも乗り気で張り切っていたが
だんだんと落ち着いてきて
今までと大差ないようになってしまった

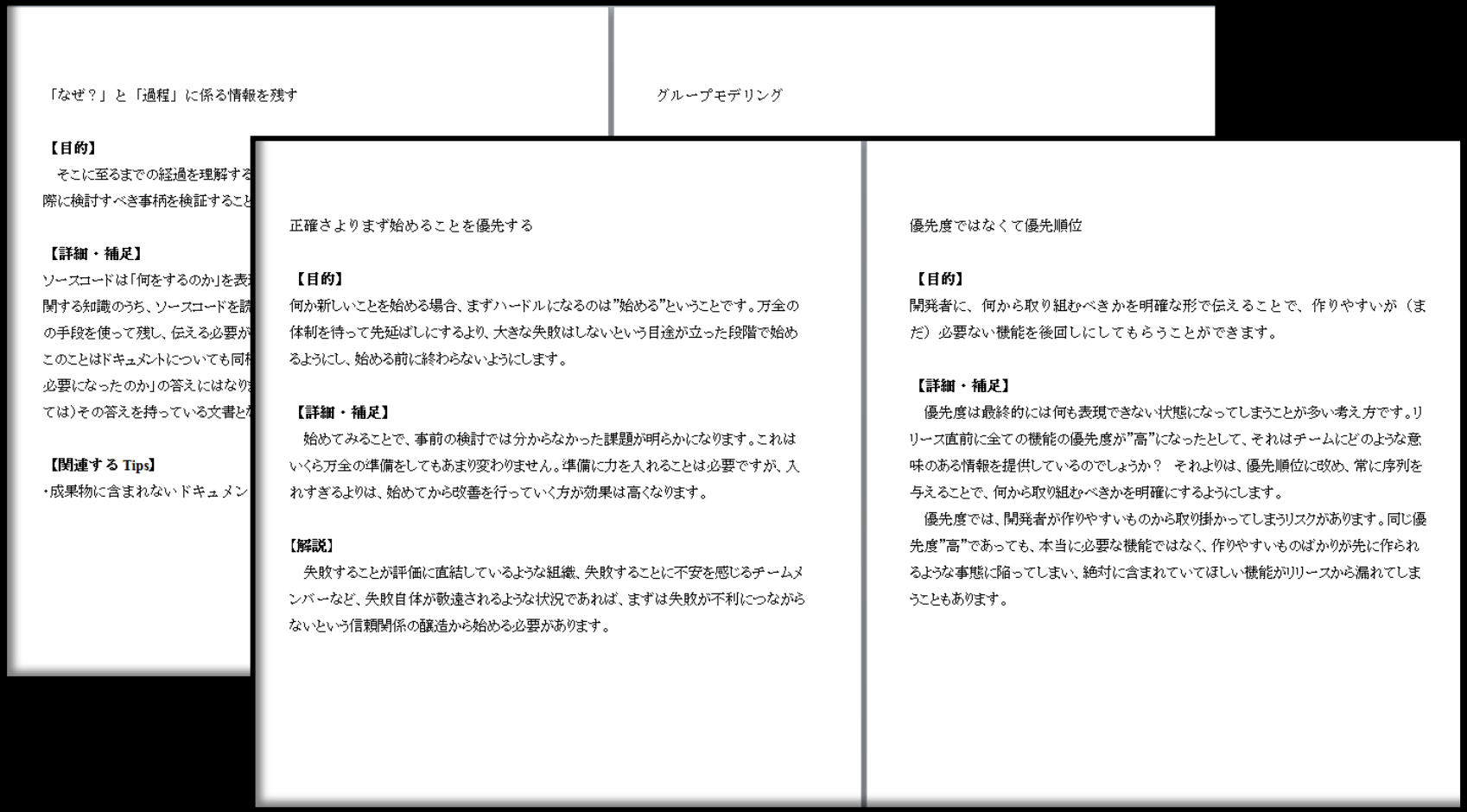
ケーススタディ

どのような状態？

・プロジェクトの最初から残業前提
・「作り方」から「詳細な設計」まで指示
・「開発速度UP –> タスクを詰め込む」の繰り返し

ケーススタディ

Tipsから使えそうなものを探す



「なぜ？」と「過程」に係る情報を残す

【目的】

そこに至るまでの経過を理解する
際に検討すべき事柄を検証するこ

【詳細・補足】

ソースコードは「何をするのか」を表
関する知識のうち、ソースコードを読
の手段を使って残し、伝える必要が
このことはドキュメントについても同
必要になったのか」の答えにはなり
ては）その答えを持っている文書と

【関連する Tips】

・成果物に含まれないドキュメン

グループモデリング

正確さよりまず始めることを優先する

【目的】

何か新しいことを始める場合、まずハードルになるのは”始める”ということです。万全の体制を待って先延ばしにするより、大きな失敗はしないという目途が立った段階で始めるようにし、始める前に終わらないようにします。

【詳細・補足】

始めてみることで、事前の検討では分からなかった課題が明らかになります。これはいくら万全の準備をしてもあまり変わりません。準備に力を入れることは必要ですが、入れすぎるよりは、始めてから改善を行っていく方が効果は高くなります。

【解説】

失敗することが評価に直結しているような組織、失敗することに不安を感じるチームメンバーなど、失敗自体が敬遠されるような状況であれば、まずは失敗が不利につながらないという信頼関係の醸造から始める必要があります。

優先度ではなくて優先順位

【目的】

開発者に、何から取り組むべきかを明確な形で伝えることで、作りやすいが（まだ）必要ない機能を後回しにしてもらうことができます。

【詳細・補足】

優先度は最終的には何も表現できない状態になってしまうことが多い考え方です。リリース直前に全ての機能の優先度が”高”になったとして、それはチームにどのような意味のある情報を提供しているのでしょうか？ それよりは、優先順位に改め、常に序列を与えることで、何から取り組むべきかを明確にするようにします。

優先度では、開発者が作りやすいものから取り掛かってしまうリスクがあります。同じ優先度”高”であっても、本当に必要な機能ではなく、作りやすいものばかりが先に作られるような事態に陥ってしまい、絶対に含まれていてほしい機能がリリースから漏れてしまうこともあります。

ケーススタディ

・プロジェクトの最初から残業前提
→『アジャイルを行わない時間を決める』

・「作り方」から「詳細な設計」まで指示
→『どう作るかは開発者に任せる』

・「開発速度UP –> タスクを詰め込む」の繰り返し
→『たまには休む』

ケーススタディ

http://www.flickr.com/photos/dawilson/2559483661

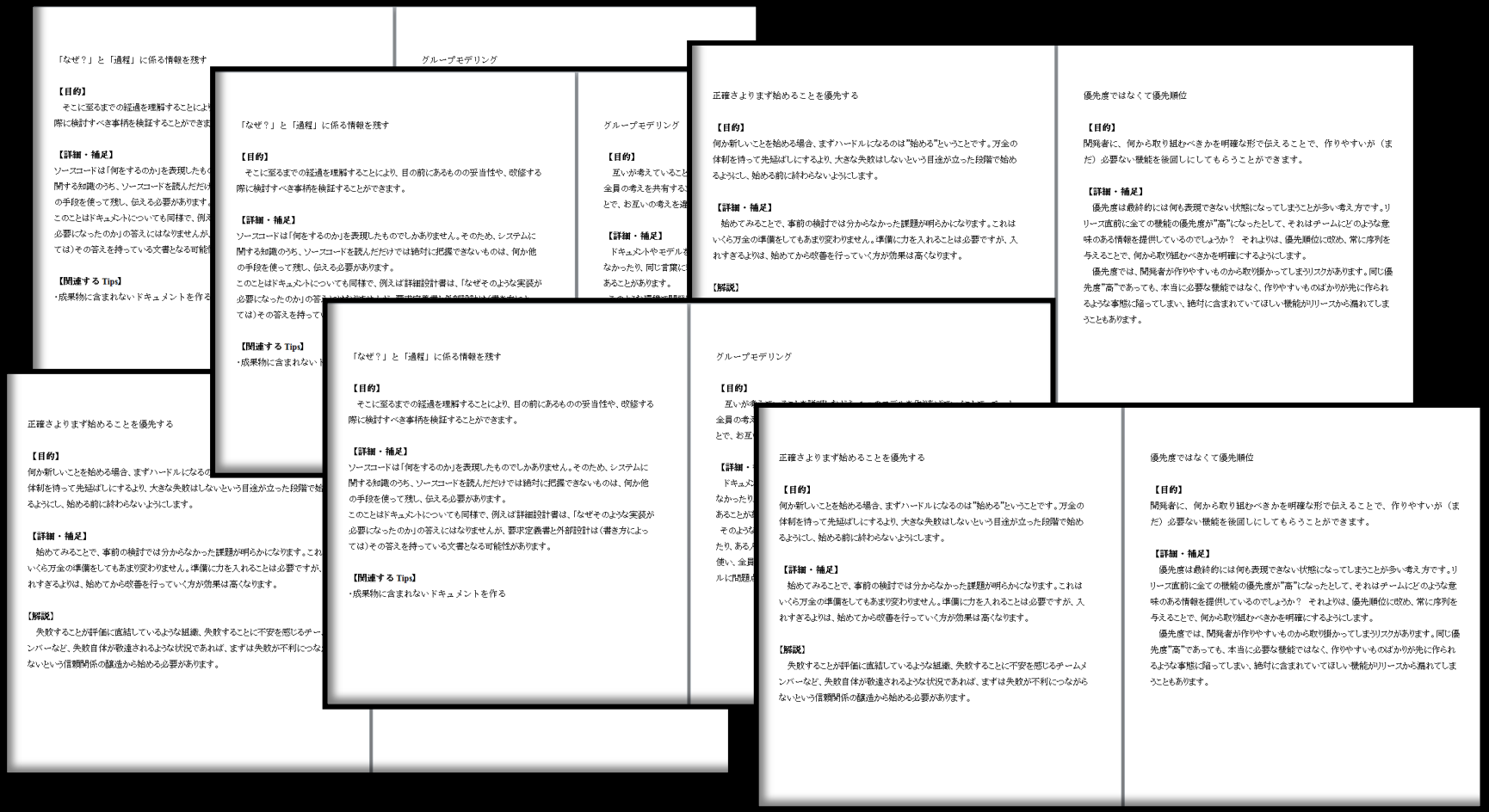
ケーススタディ

どのような状態？

・チームメンバーのスキル・リテラシがばらばら
・イテレーション終盤はいつもばたばた
・「動けばいい」を繰り返した結果、メンテ困難に

ケーススタディ

Tipsから使えそうなものを探す



ケーススタディ

・チームメンバーのスキル・リテラシがばらばら

→『成果物に含まれないドキュメントを作る』

→『グループモデリング』

・イテレーション終盤はいつもばたばた

→『タスク及びその進捗を壁に貼り出す』

・「動けばいい」を繰り返した結果、メンテ困難に

→『中長期的な品質に拘る』

→『「なぜ？」と「過程」に係る情報を残す』

Be agile with this guidelines!

UMTP
UML Based Modeling Technologies Promotion

http://www.flickr.com/photos/elenapardo/12394337334

最後に

本日説明したガイドラインは３月末に
UMTPアジャイル開発部会のWebサイトからダウンロード開始予定です。

UMTP
UML Based Modeling Technologies Promotion

ご清聴ありがとうございました。