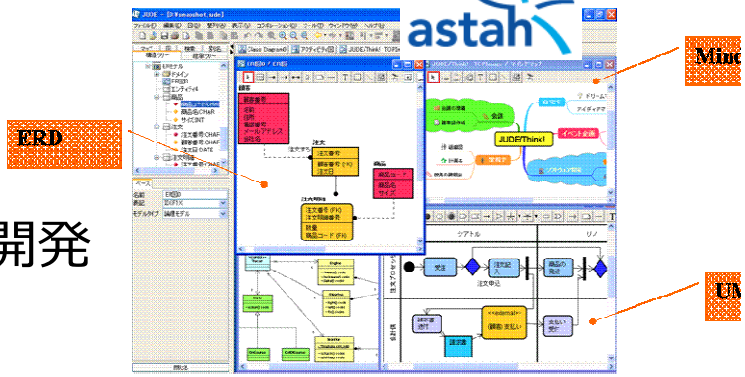
アジャイル時代のモデリング

平鍋健児
チェンジビジョン

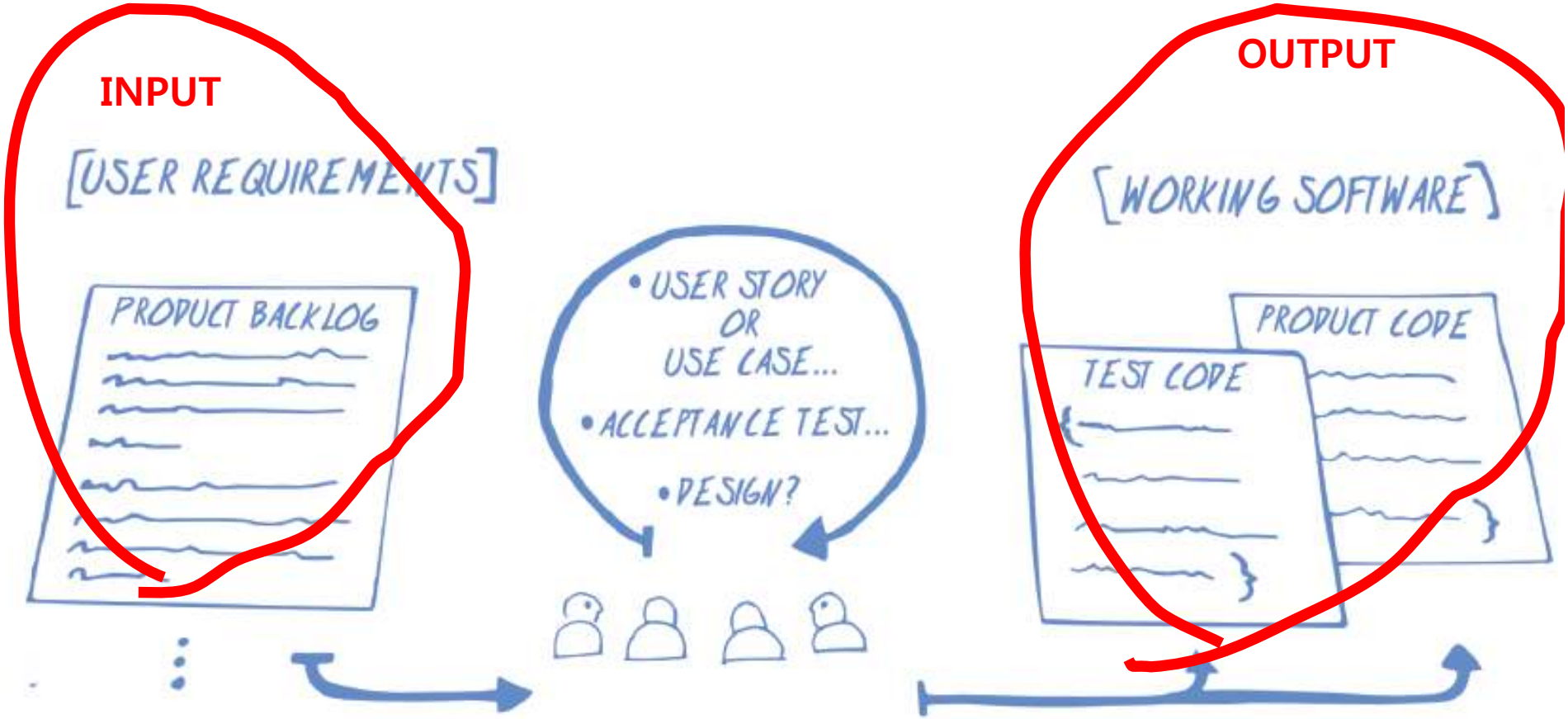
自己紹介

自己紹介

- ㈱永和システムマネジメント
  - 福井市（本社）、上野東京（支社）
  - Ruby と Agileを使ったシステム開発
- 株式会社チェンジビジョン
  - 福井市（開発部）、上野東京（本社）
  - astah* (旧：JUDE) の開発
- 平鍋健児
  - UML+マインドマップエディタ astah*の開発
  - 要求開発アライアンス、理事
  - 翻訳、XP関連書籍、『リーン開発の本質』『IMPACT MAPPING』等多数。
  - 著書『アジャイル開発とスクラム』、『要求開発』『ソフトウェア開発に役立つマインドマップ』



Agile と Design



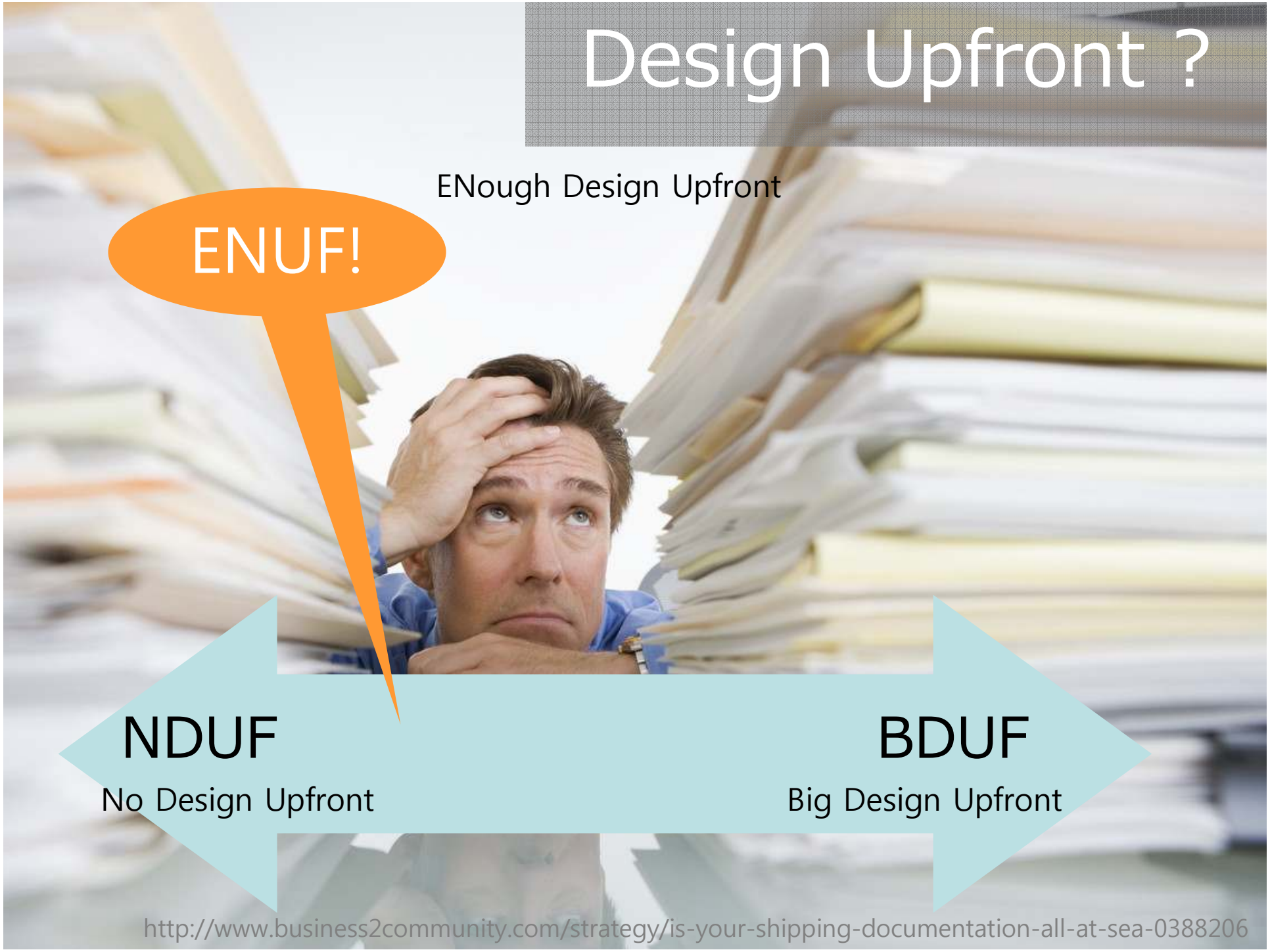
Where is Design?

“While code is the truth,
it is not the whole truth.”

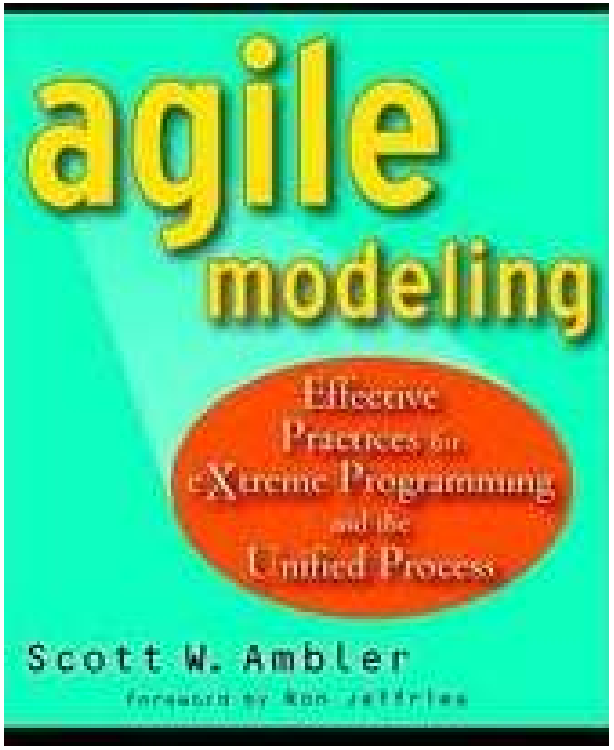
-Grady Booch


成長しつづけるが、
一貫した美がある。

Design Upfront ?



Agile Modeling



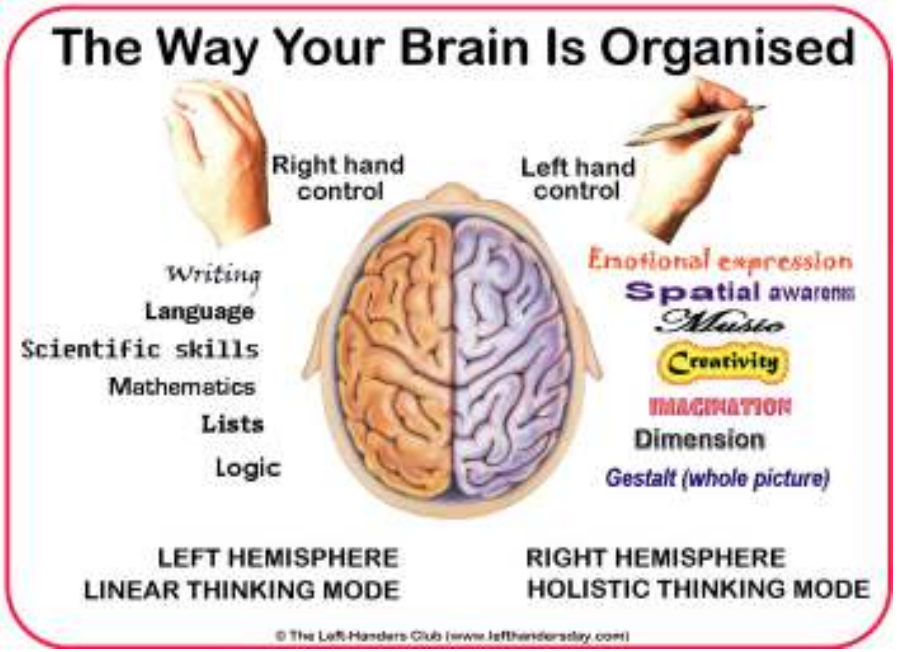
- By Scott Ambler

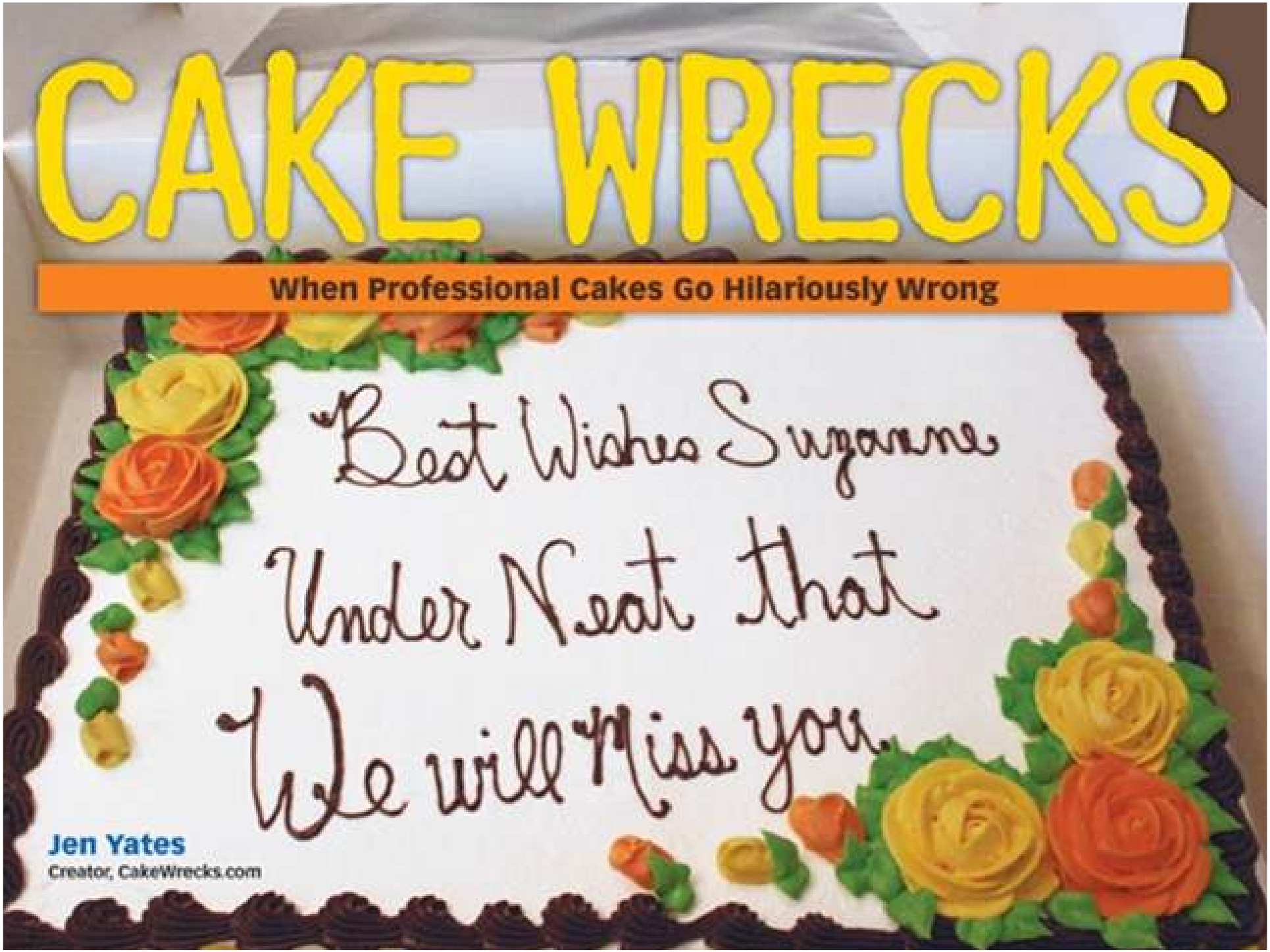
"Let's keep the modeling baby but throw out the bureaucracy bathwater"

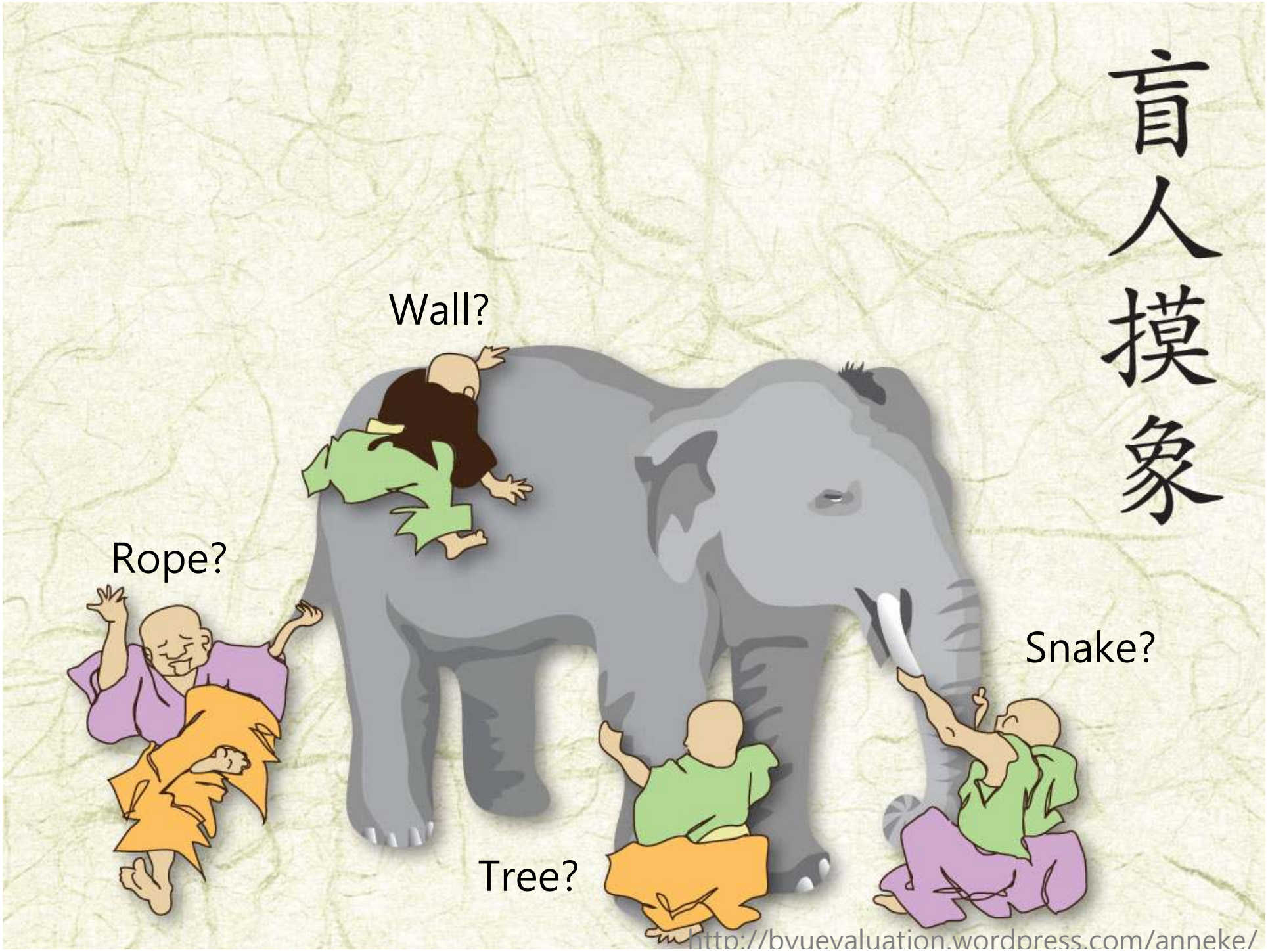
– Scott Ambler

なぜモデルを使うのか？

- 人間はイメージの方が概要をうまく掴むことができる。
  - A picture is worth 1,000 words.
  - 右脳と左脳



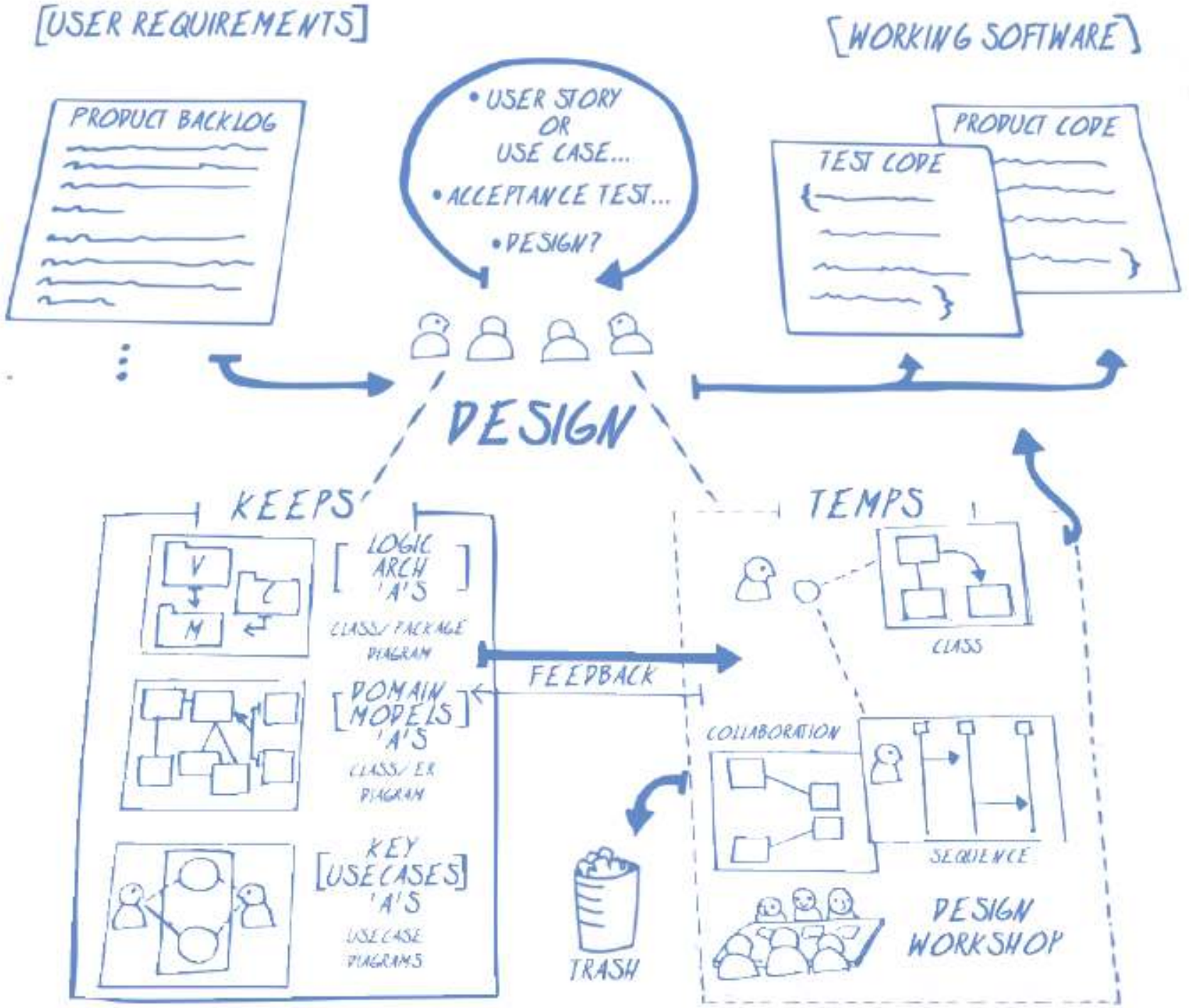

CAKE WRECKS
When Professional Cakes Go Hilariously Wrong
Best Wishes Suzanne
Under Neat that
We will Miss you.
Jen Yates
Creator, CakeWrecks.com


盲人摸象
Wall?
Rope?
Snake?
Tree?
http://byuevaluation.wordpress.com/anneke/

どんなモデルが
コードの次に必要か

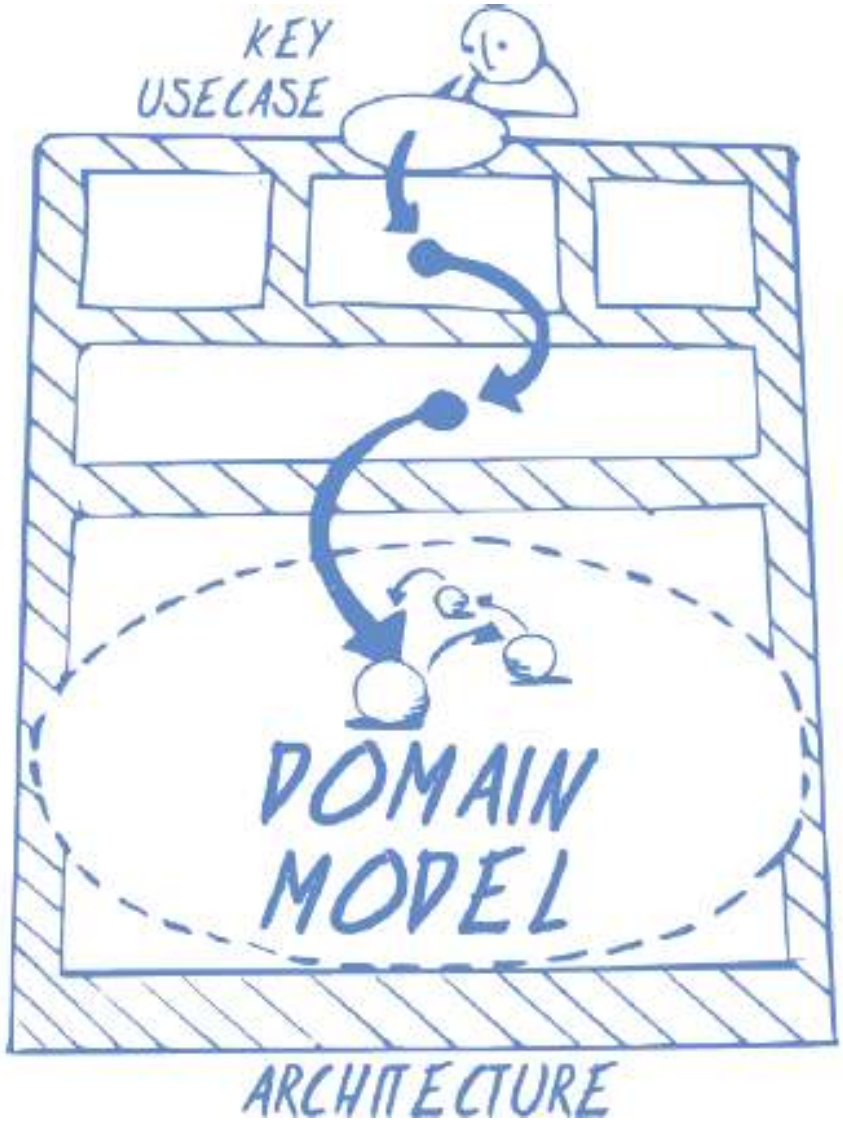
問題

- 「ビッグピクチャ」の共通理解をつくるために、一番シンプルな、モデルのセットとは何か?


[USER REQUIREMENTS]
[WORKING SOFTWARE]
PRODUCT BACKLOG
• USER STORY OR USE CASE...
• ACCEPTANCE TEST...
• DESIGN?
TEST CODE
PRODUCT CODE
DESIGN
KEEPS
TEMPS
V
C
M
[LOGIC ARCH 'A'S]
CLASS/ PACKAGE DIAGRAM
CLASS
FEEDBACK
[DOMAIN MODELS] 'A'S
CLASS/ ER DIAGRAM
COLLABORATION
SEQUENCE
KEY [USECASES] 'A'S
USECASE DIAGRAMS
TRASH
DESIGN WORKSHOP

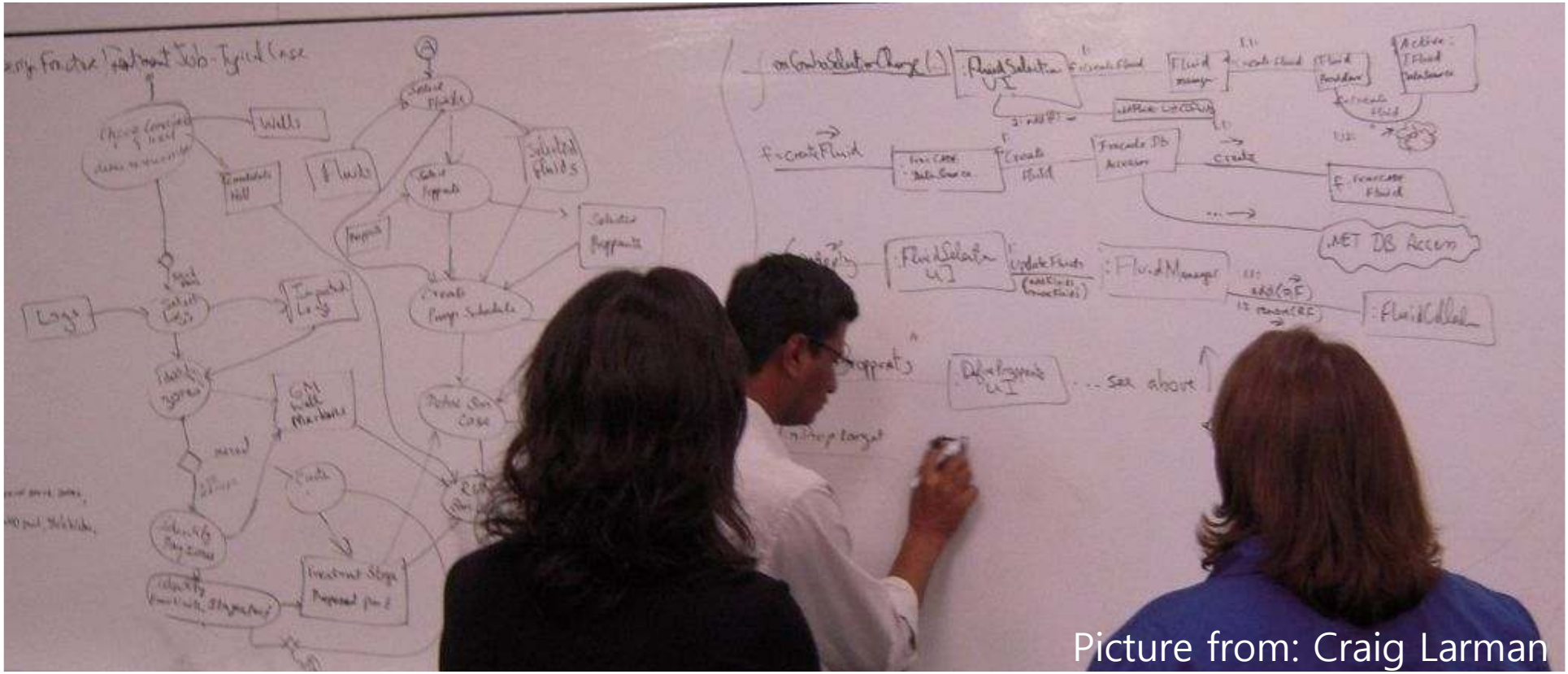
“Keeps”

- **Architecture**
  - As Class/Package Diagrams
- **Domain Model**
  - As Class Diagram/ER Diagrams
- **Key Use Cases**
  - 1. As Use Case Diagrams
  - 2. As Sequence/Communication Diagrams



“Temps”

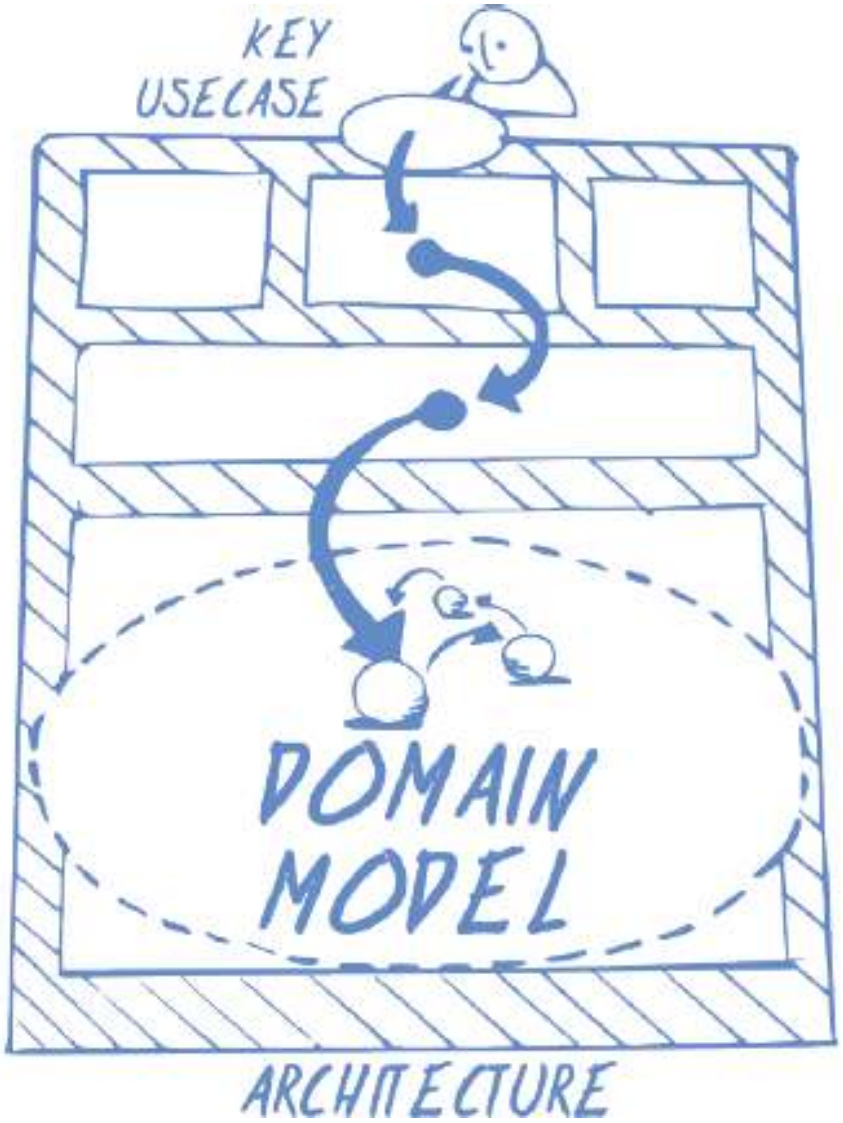
- **Casual Modeling**
  - Ex. As Class+Sequence Diagrams, and others
  - “Keeps”の上に乗せる (ツールを使ったり、印刷の上に書いたり)
  - ホワイトボード!



Picture from: Craig Larman

“Keeps” (again)

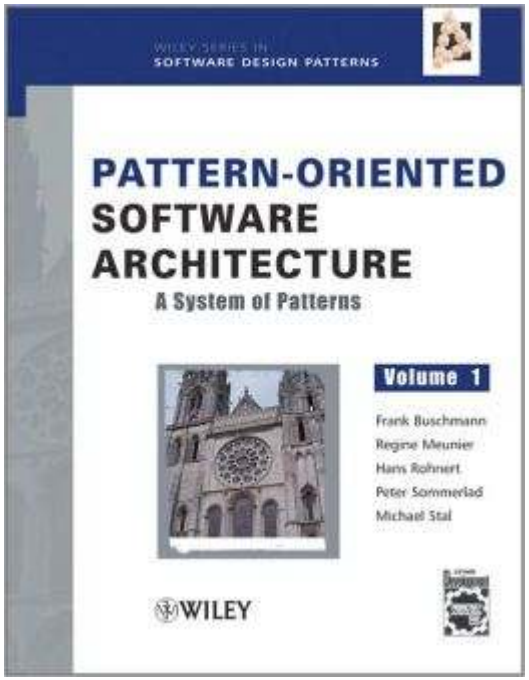
- **Architecture**
  - As Class/Package Diagrams
- **Domain Model**
  - As Class Diagram/ER Diagrams
- **Key Use Cases**
  - 1. As Use Case Diagrams
  - 2. As Sequence/Communication Diagrams



Architecture

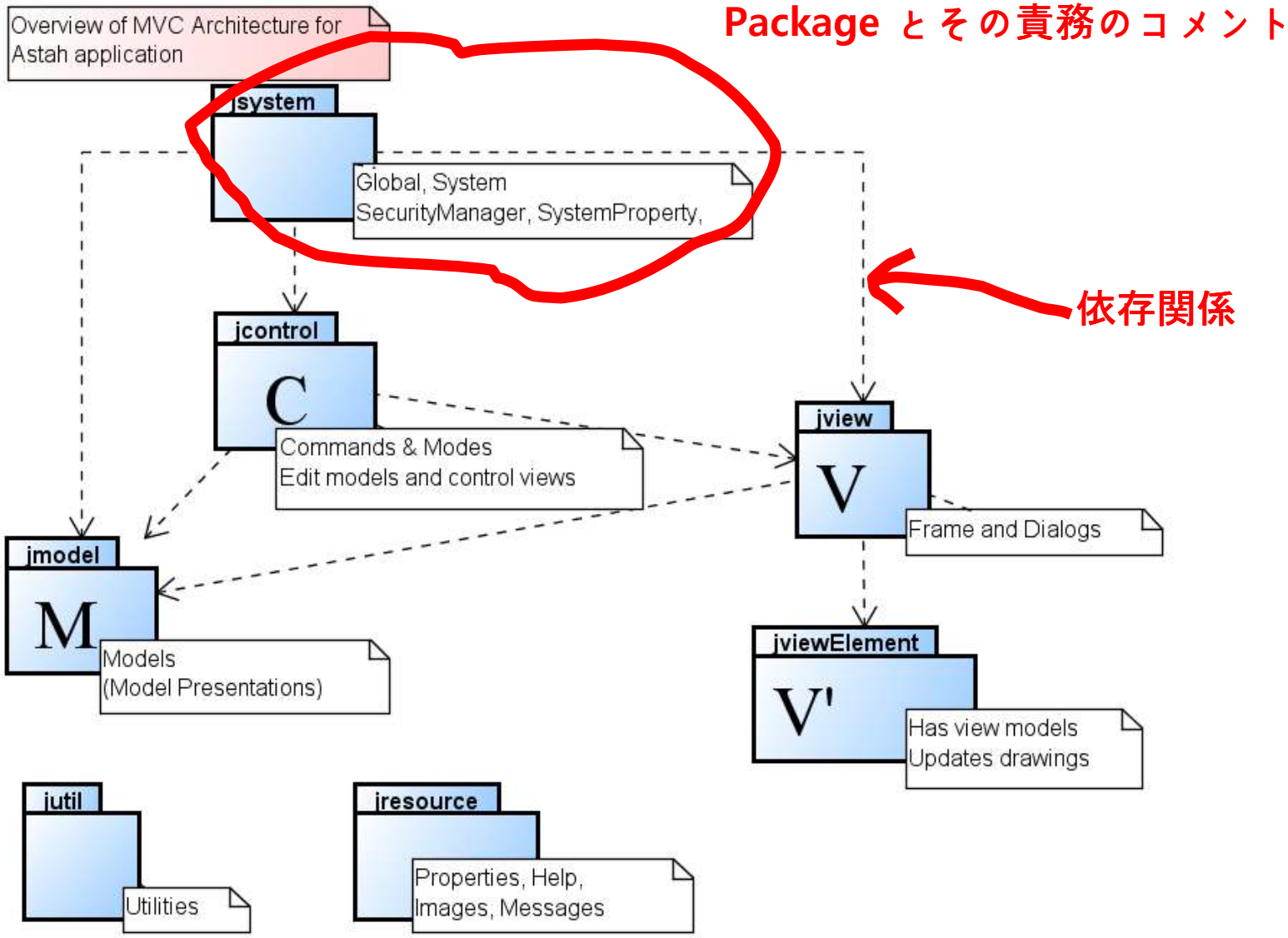
- システム全体の構造的な表現
- 大域的な層やTierが表現されることが多く、クラス図やパッケージ図がよく使われる。
- よく知られたパターンが存在する。
  - MVC, Blackboard, Pipes-Filters, Layers
- 「フレームワーク」を使うのもアーキテクチャ選択の１つ。

Pattern-Oriented Software Architecture (Vol.1-5)



- By Frank Buschmann et. al

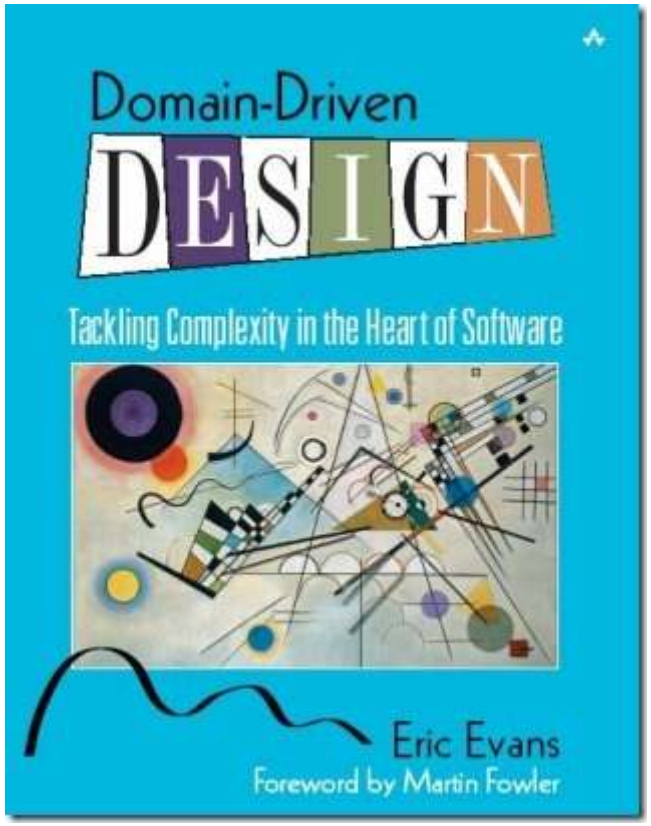
“Architecture”の例



ドメインモデル

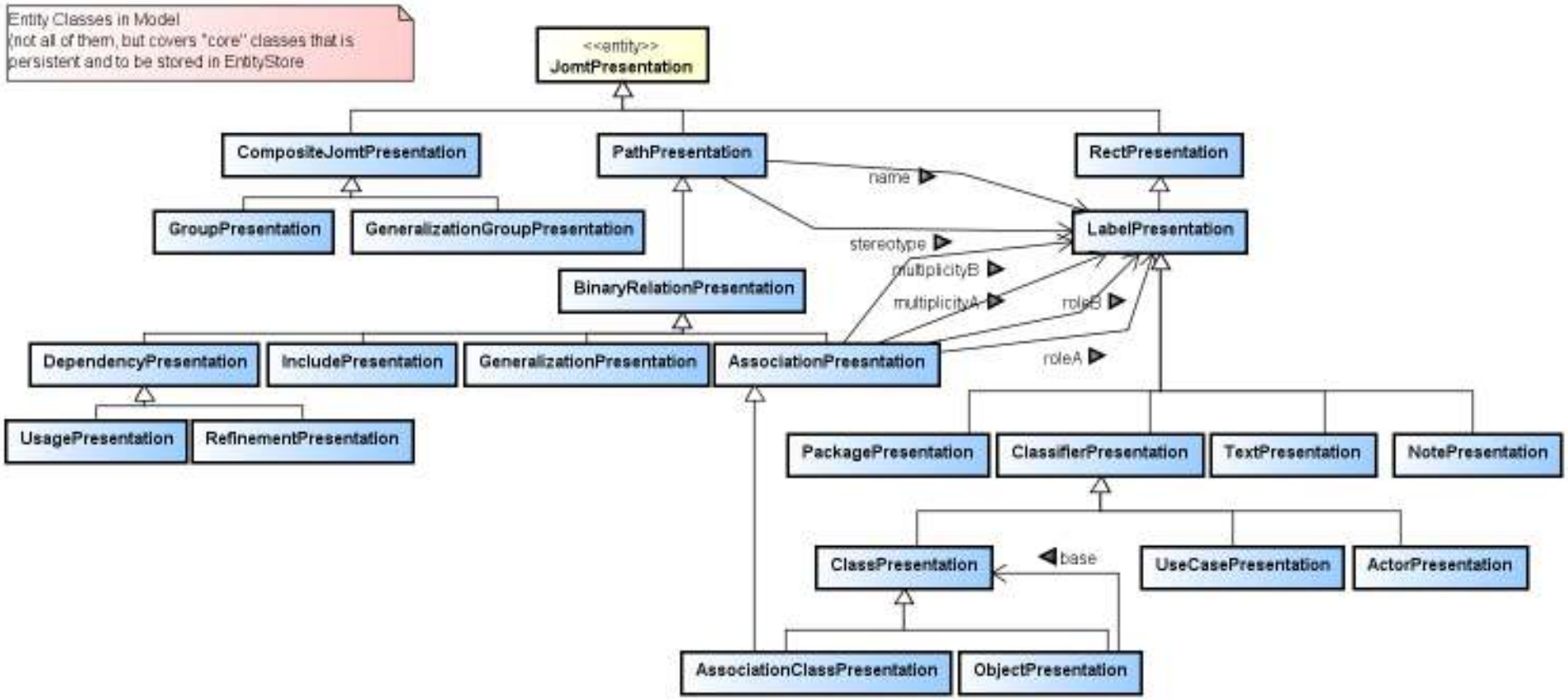
- 問題領域（ドメイン）の概念とそれらの繋がり
- 人間のコミュニケーション・レベル
  - 全ステークホルダーが話す語彙。
  - ユーザー、ドメインの専門家、ビジネス分析、開発者
- プログラミング・レベル
  - コードを構成する要素（クラス、データ、操作、ファイル、、、、、）の「名前づけ」
  - それらの多くが永続データ(entity)にマッピングされる。

Domain-Driven Design

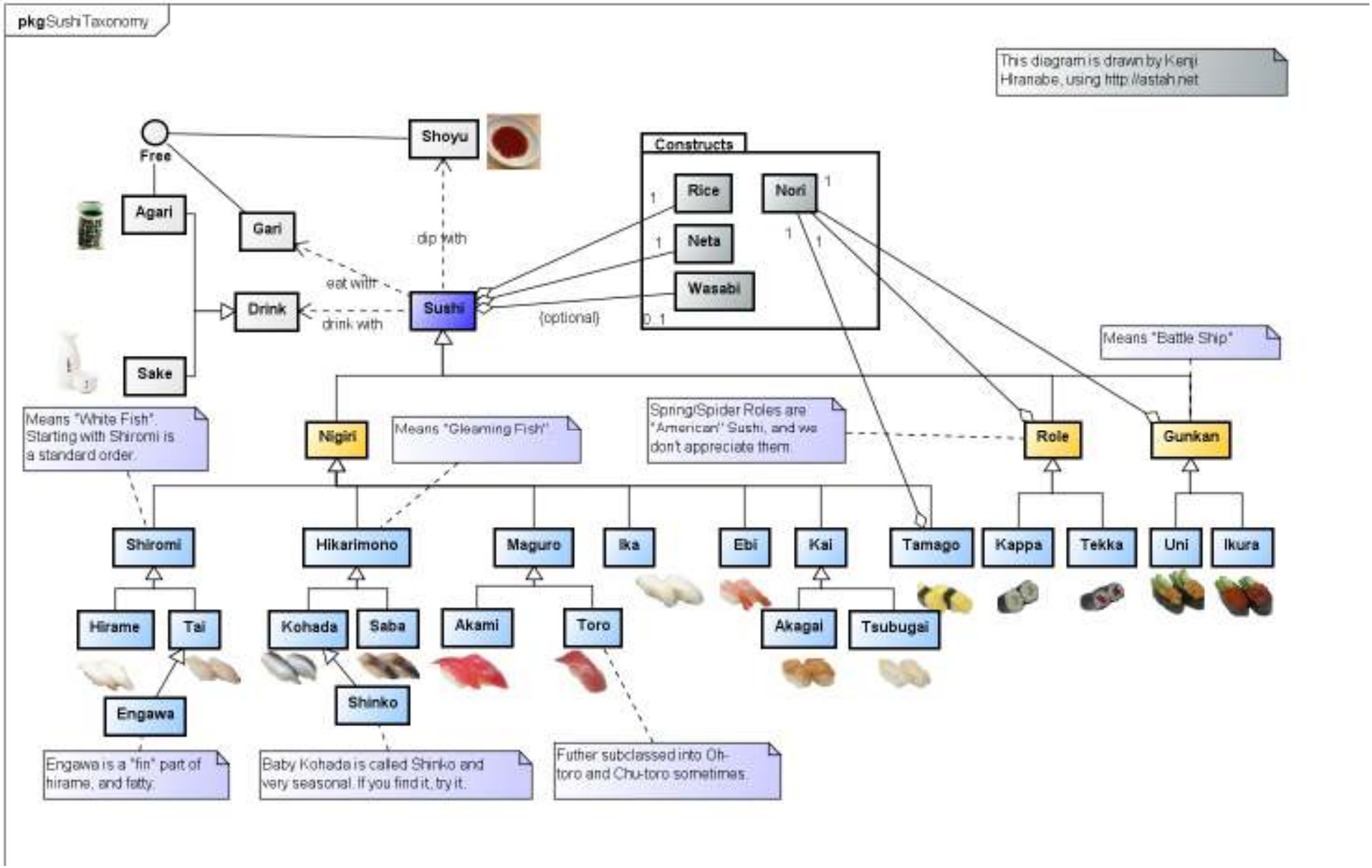


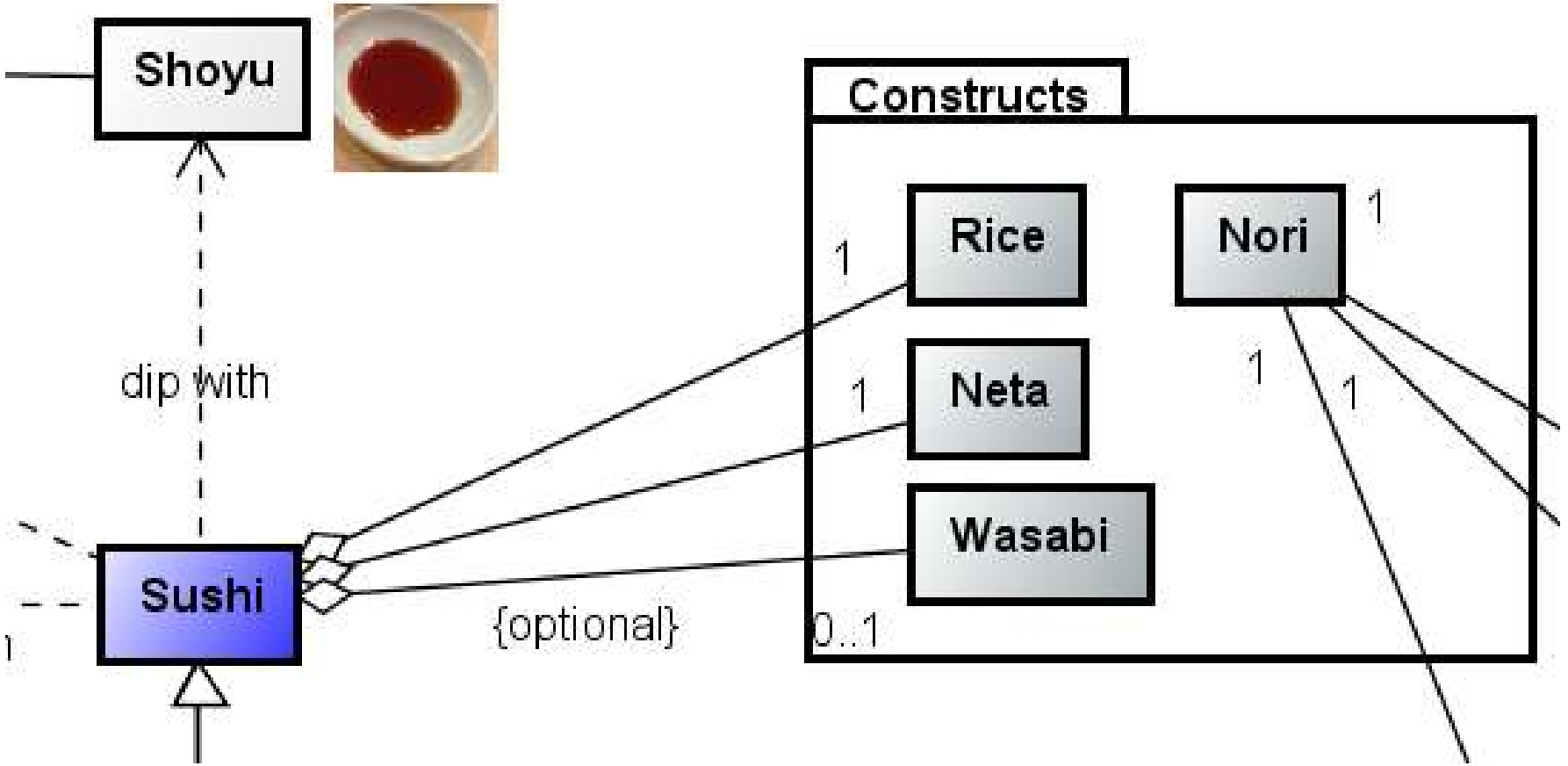
- By Eric Evans
- Ubiquitous Language

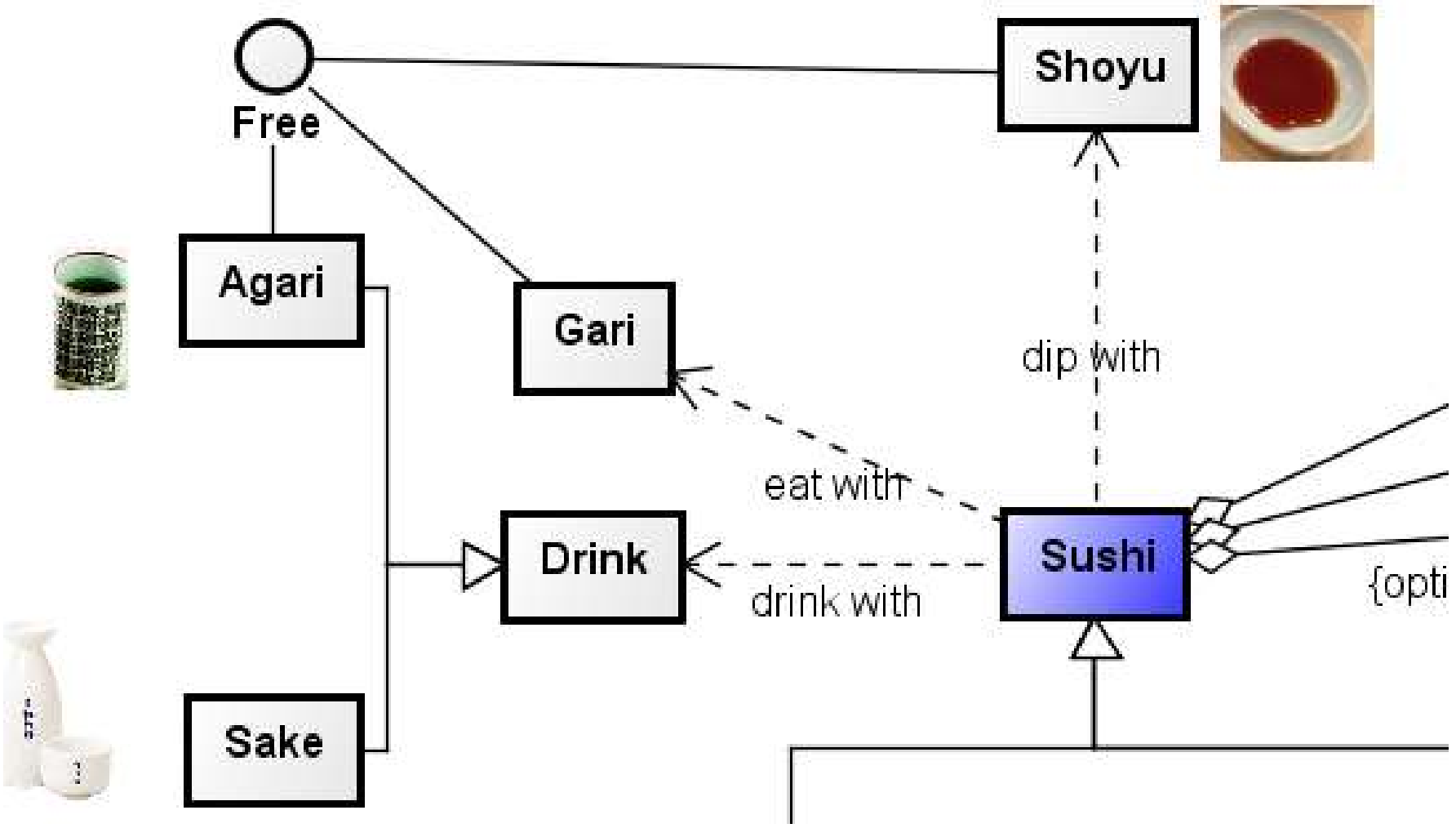
“Domain Model”の例

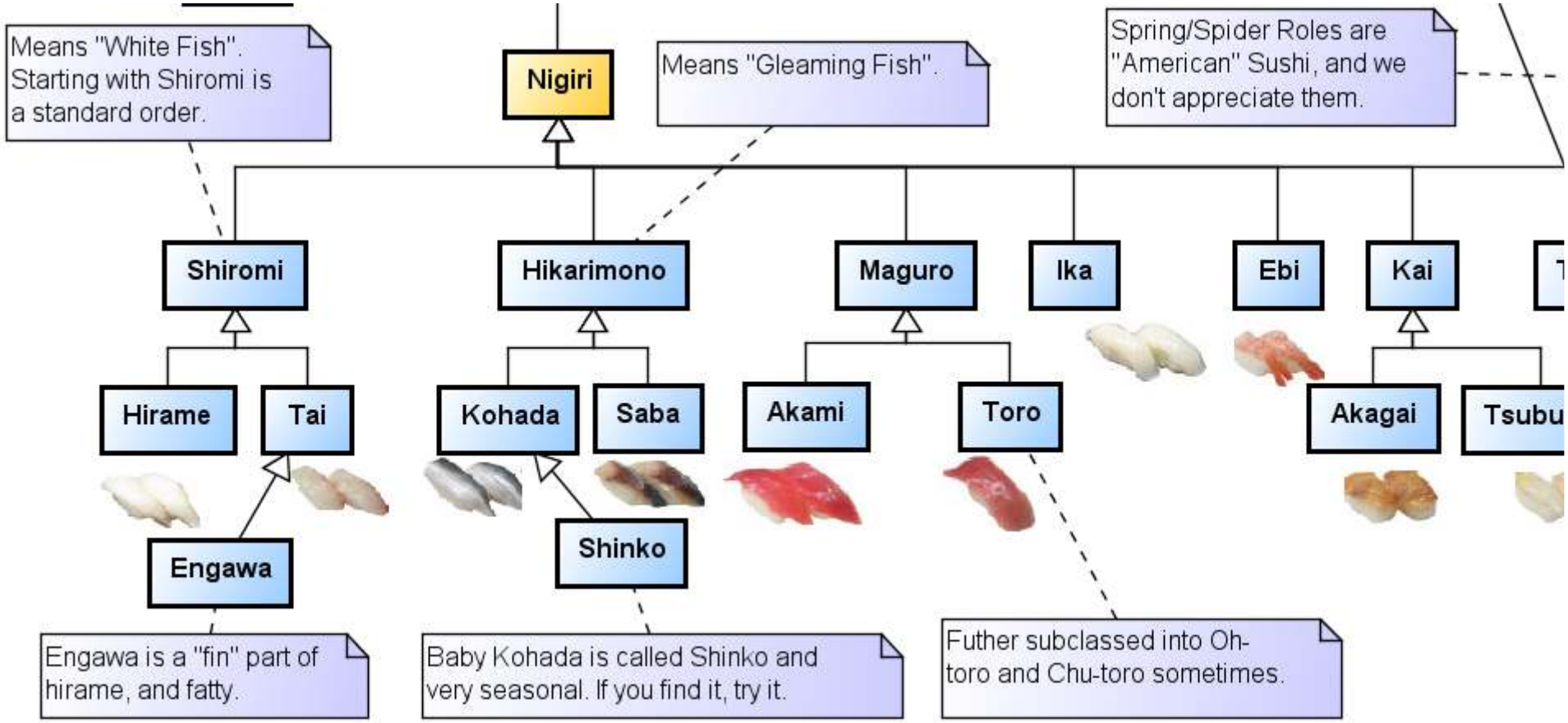


「寿司」のドメインモデル


pkg Sushi Taxonomy
This diagram is drawn by Kenji Hiranabe, using http://astah.net
Free
Shoyu
Agari
Gari
dip with
eat with
Drink
drink with
Sushi
Sake
Constructs
Rice
Nori
Neta
Wasabi
1
0..1
{optional}
Means "Battle Ship"
Means "White Fish". Starting with Shiromi is a standard order.
Nigiri
Means "Gleaming Fish"
Spring/Spider Roles are "American" Sushi, and we don't appreciate them
Role
Gunkan
Shiromi
Hikarimono
Maguro
Ika
Ebi
Kai
Tamago
Kappa
Tekka
Uni
Ikura
Hirame
Tai
Kohada
Saba
Akami
Toro
Akagai
Tsubugai
Engawa
Shinko
Engawa is a "fin" part of hirame, and fatty.
Baby Kohada is called Shinko and very seasonal. If you find it, try it.
Futher subclassed into Oh-toro and Chu-toro sometimes.


Shoyu
dip with
Sushi
1
{optional}
Constructs
1
Rice
1
Neta
Wasabi
0..1
Nori
1
1
1
1

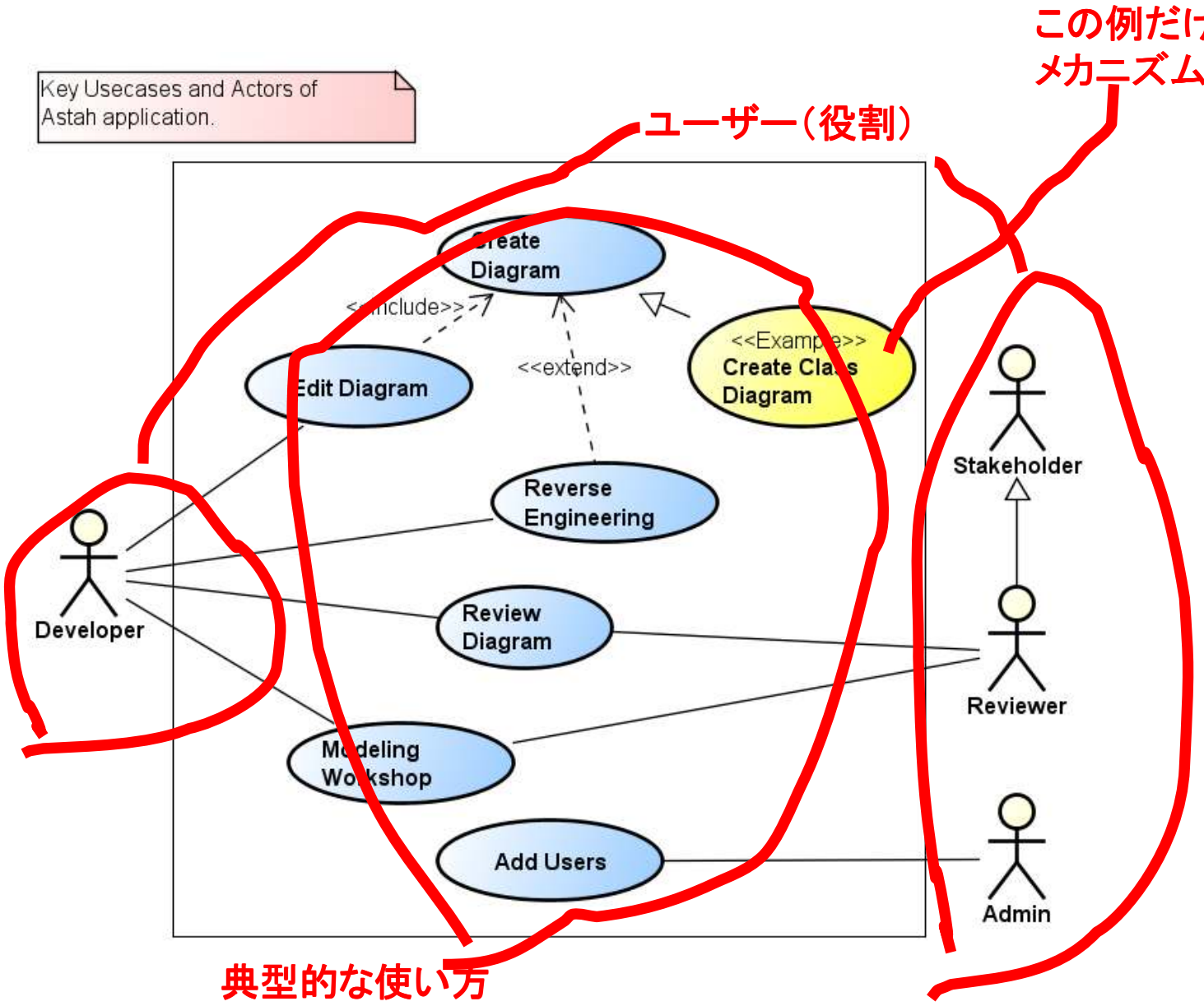

Free
Shoyu
Agari
Gari
dip with
eat with
Drink
Sushi
drink with
{opti
Sake


Means "White Fish". Starting with Shiromi is a standard order.
Nigiri
Means "Gleaming Fish".
Spring/Spider Roles are "American" Sushi, and we don't appreciate them.
Shiromi
Hikarimono
Maguro
Ika
Ebi
Kai
Hirame
Tai
Kohada
Saba
Akami
Toro
Akagai
Tsubu
Engawa
Shinko
Engawa is a "fin" part of hirame, and fatty.
Baby Kohada is called Shinko and very seasonal. If you find it, try it.
Futher subclassed into Oh-toro and Chu-toro sometimes.

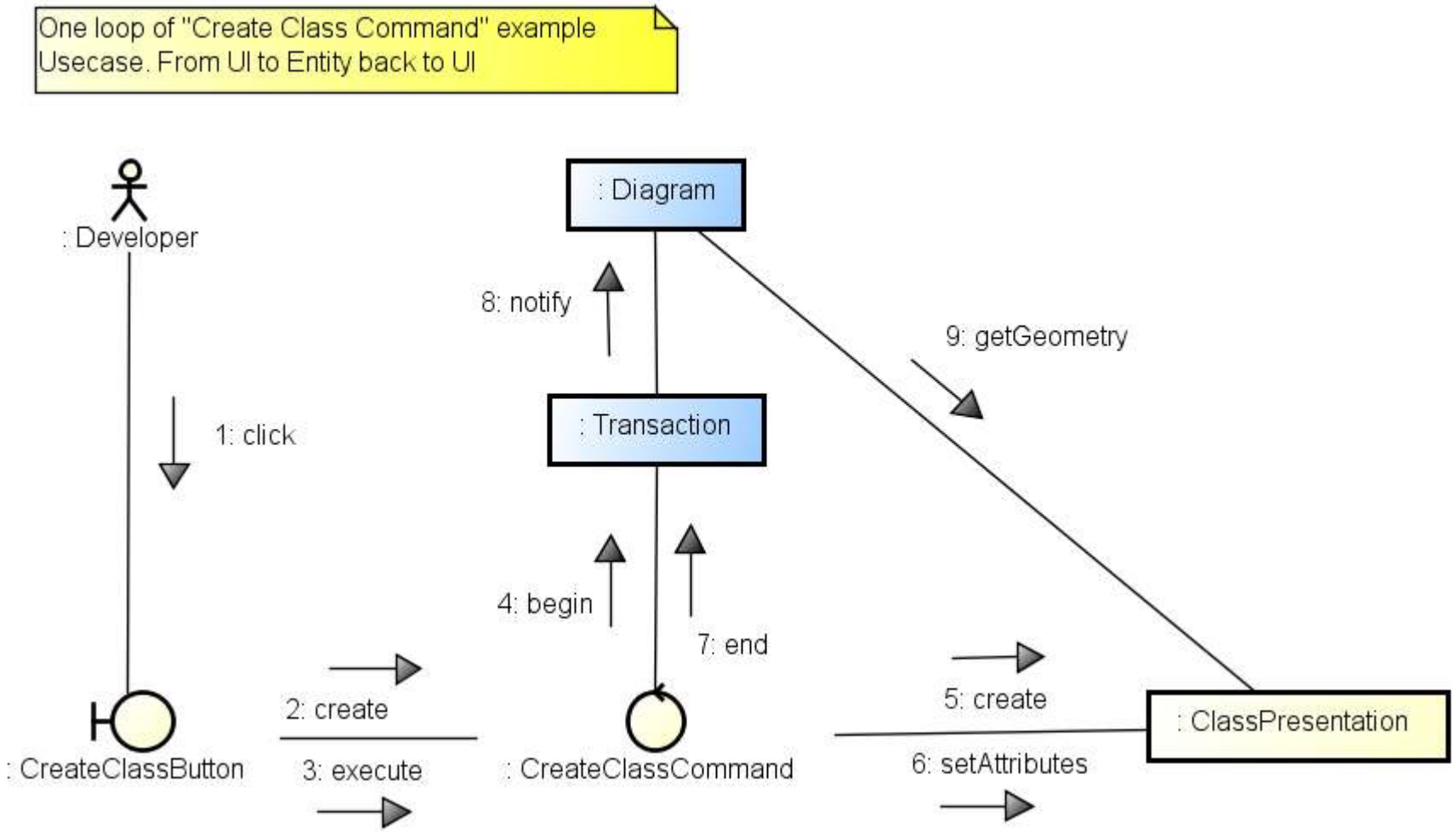
Key UseCases

1. ユーザーから見たシステムの代表的な使い方
   - 誰がユーザーで、何がうれしいか。
2. そのゴールを得るまでのアーキテクチャを貫通するメカニズム
   - シーケンス図やコミュニケーション図によって描かれる、制御の流れ。
   - 開発者には教育的な例示となる。

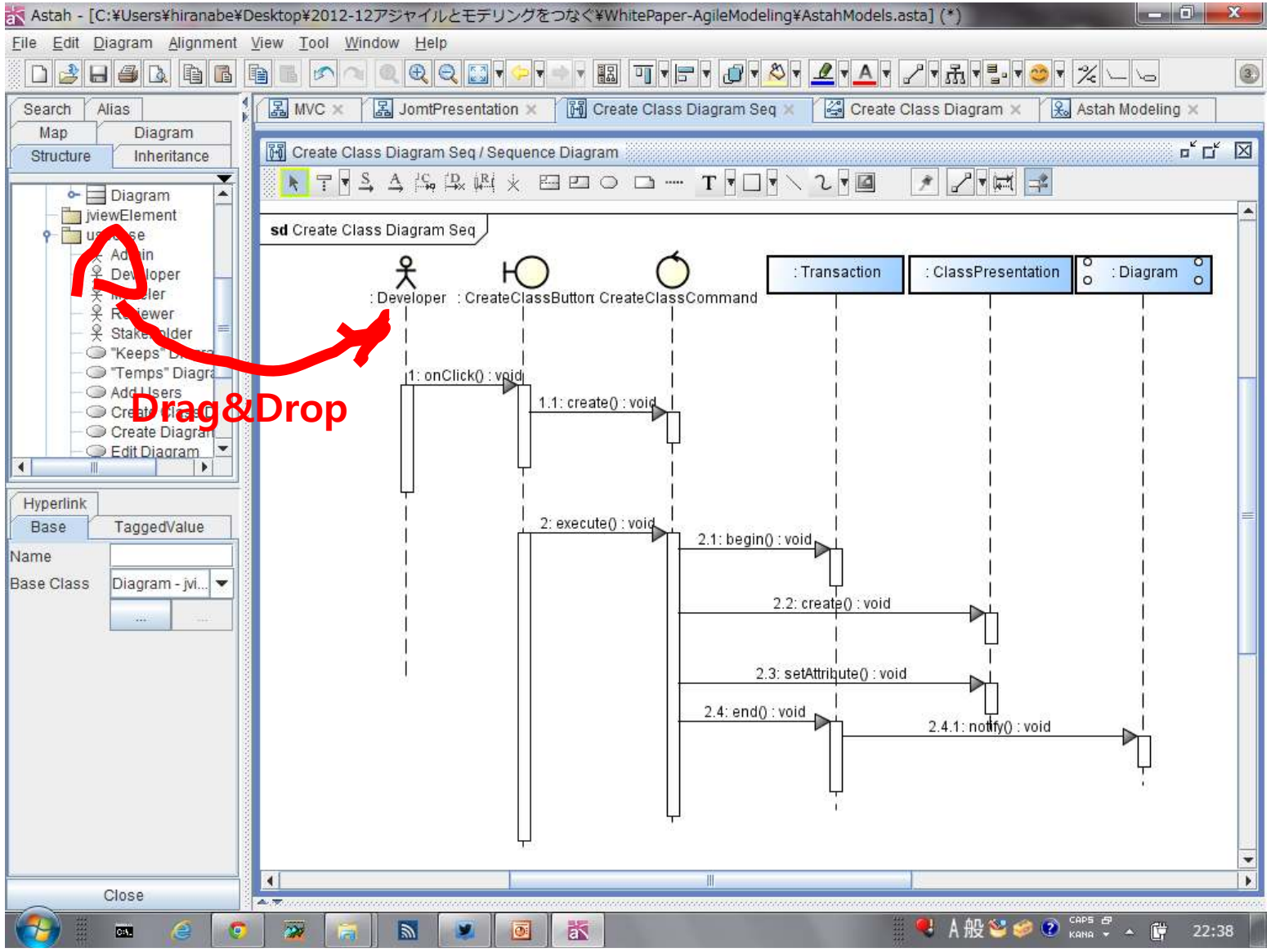
“Key UseCases”の例



“Key UseCases”の例(メカニズム)

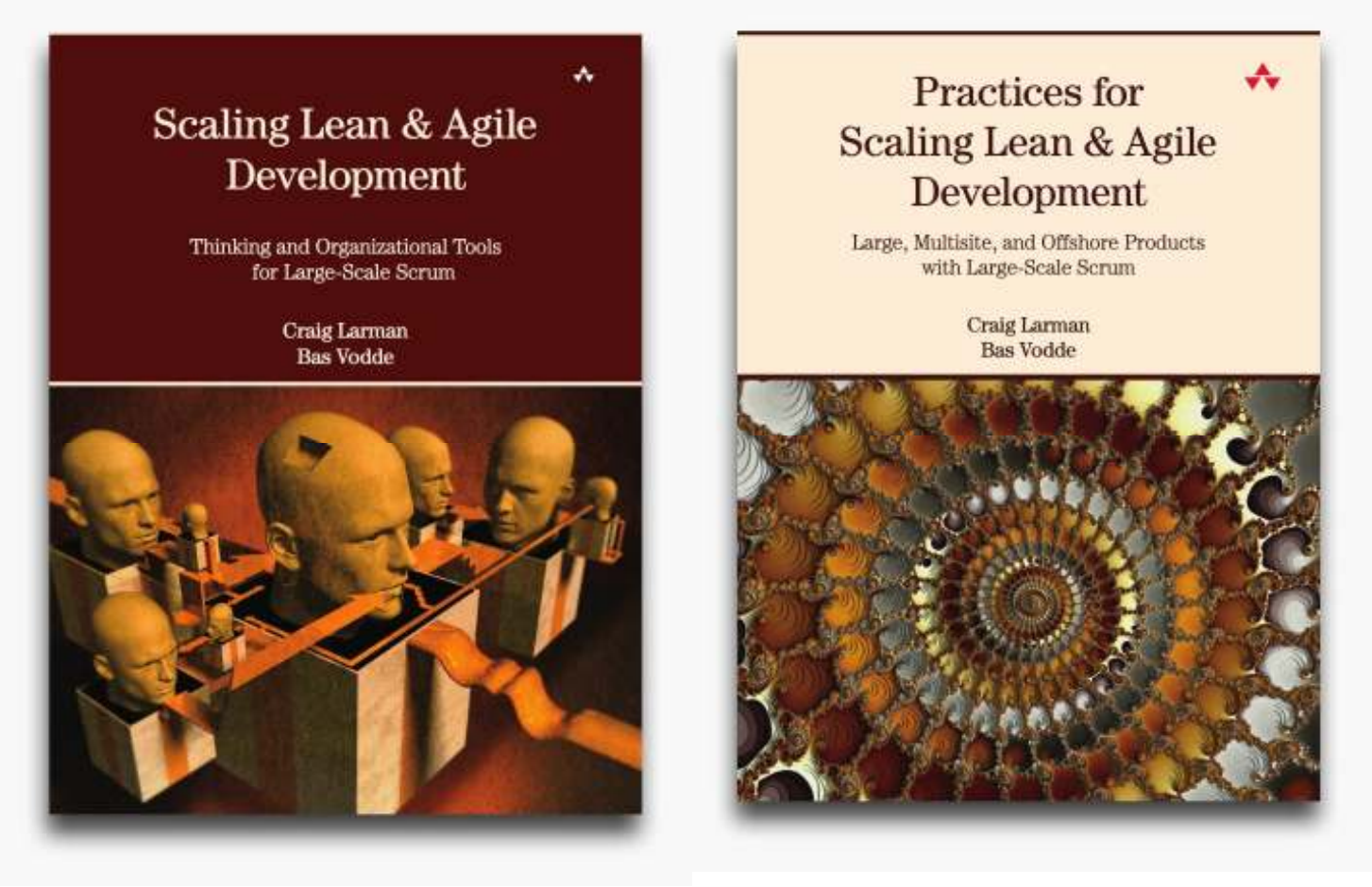


“Keeps”上に“Temps”を重ねる



Scaling

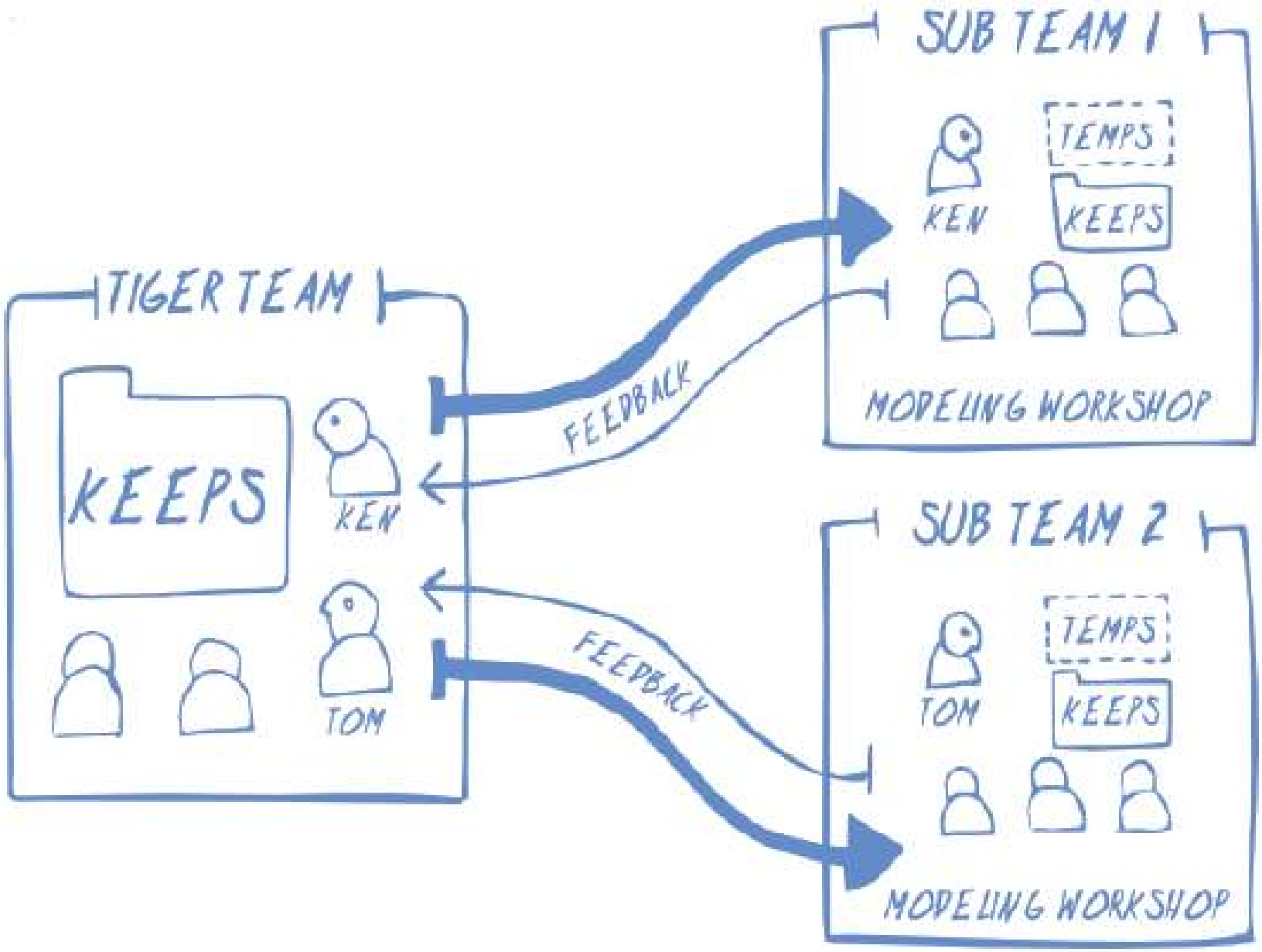
Scaling

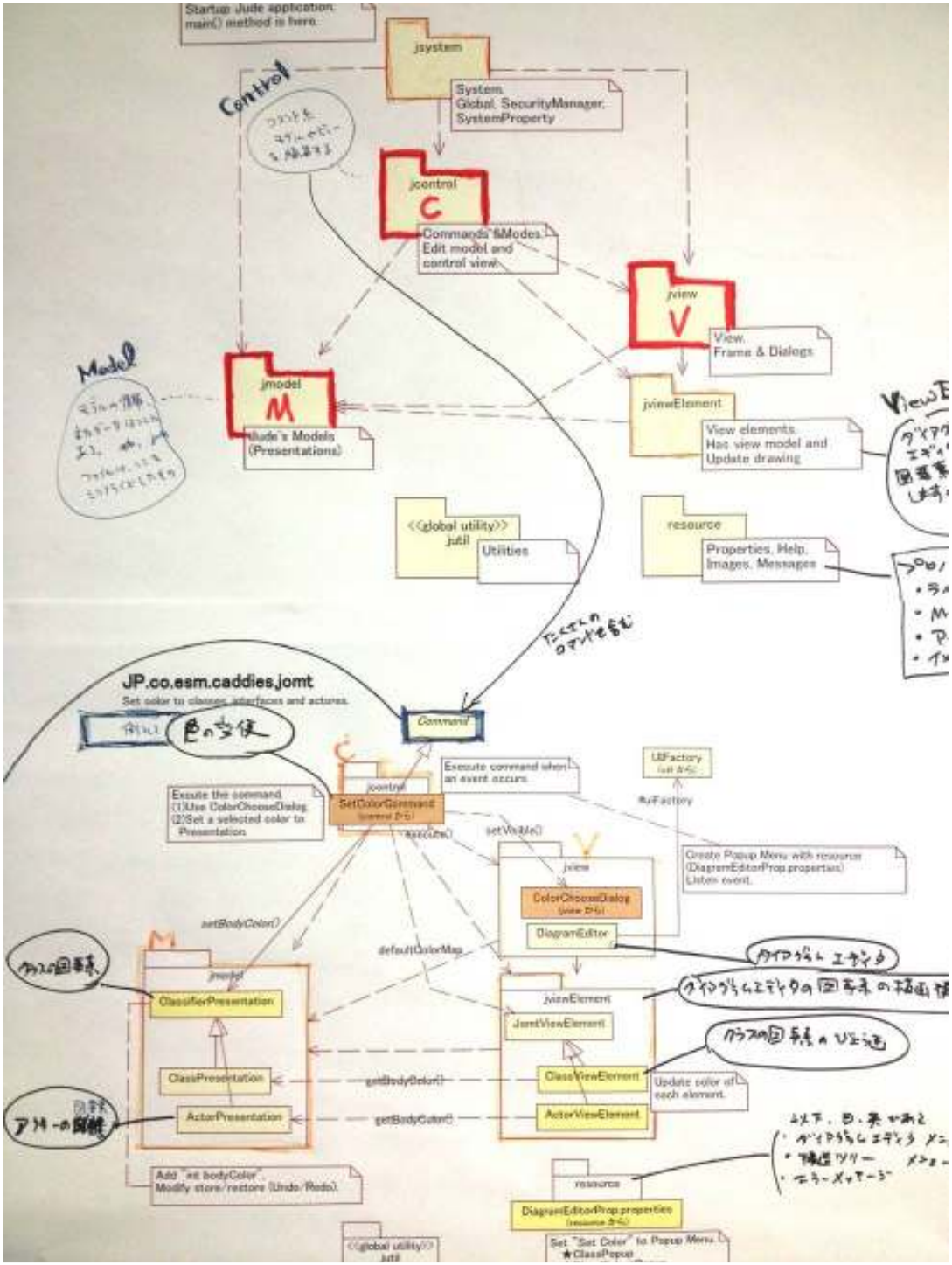


- By Craig Larman + Bas Vodde

"Rather than divide and conquer, an XP team conquers and divides"

–Kent Beck


SUB TEAM 1
TEMPS
KEEPS
KEN
MODELING WORKSHOP
TIGER TEAM
KEEPS
KEN
TOM
FEEDBACK
FEEDBACK
SUB TEAM 2
TEMPS
KEEPS
TOM
MODELING WORKSHOP


Startup Jude application.
main() method is here.
jsystem
System
Global, SecurityManager,
SystemProperty
Control
jcontrol
C
Commands &Modes.
Edit model and
control view.
jview
V
View.
Frame & Dialogs
Model
jmodel
M
Jude's Models
(Presentations)
jviewElement
View elements.
Has view model and
Update drawing
<<global utility>>
jutil
Utilities
resource
Properties, Help,
Images, Messages
JP.co.esm.caddies.jomt
Set color to classes, interfaces and actors.
Command
jcontrol
SetColorCommand
Execute command when
an event occurs.
UIFactory
execute()
setVisible()
jview
ColorChooseDialog
DiagramEditor
Execute the command.
(1)Use ColorChooseDialog.
(2)Set a selected color to
Presentation.
Create Popup Menu with resource
(DiagramEditorProp.properties)
Listen event.
setBodyColor()
defaultColorMap
jmodel
ClassifierPresentation
ClassPresentation
ActorPresentation
getBodyColor()
getBodyColor()
jviewElement
JomtViewElement
ClassViewElement
ActorViewElement
Update color of
each element.
Add "int bodyColor".
Modify store/restore (Undo/Redo).
resource
DiagramEditorProp.properties
Set "Set Color" to Popup Menu
★ClassPopup
<<global utility>>
jutil

“Model to have a conversation.”

–Craig Larman
and Bas Vodde


Jomt
UML
Jomt
Golf
astah
Jomt Model
Jomt View
Jomt Control
UML

他に Keepすべきもの

Others to Keep

- 設計の理由 (WHY's)
  - 意思決定の理由リスト
  - なぜこのフレームワークを選んだか。
  - なぜこのテクノロジを選んだか。
  - なぜこの設計、アーキテクチャなのか。
- テスト戦略
  - API as Acceptance Test Interface

- しかし、… 「人」“People” なのだ…

"A lot of design information lives in tribal memory."

–Grady Booch

式年遷宮(Shikinen-Sengu)

http://savingjapan.net/2011/06/23/jft-press-release-shikinen-sengu-ceremonial-rebuilding-for-over-1300-years/

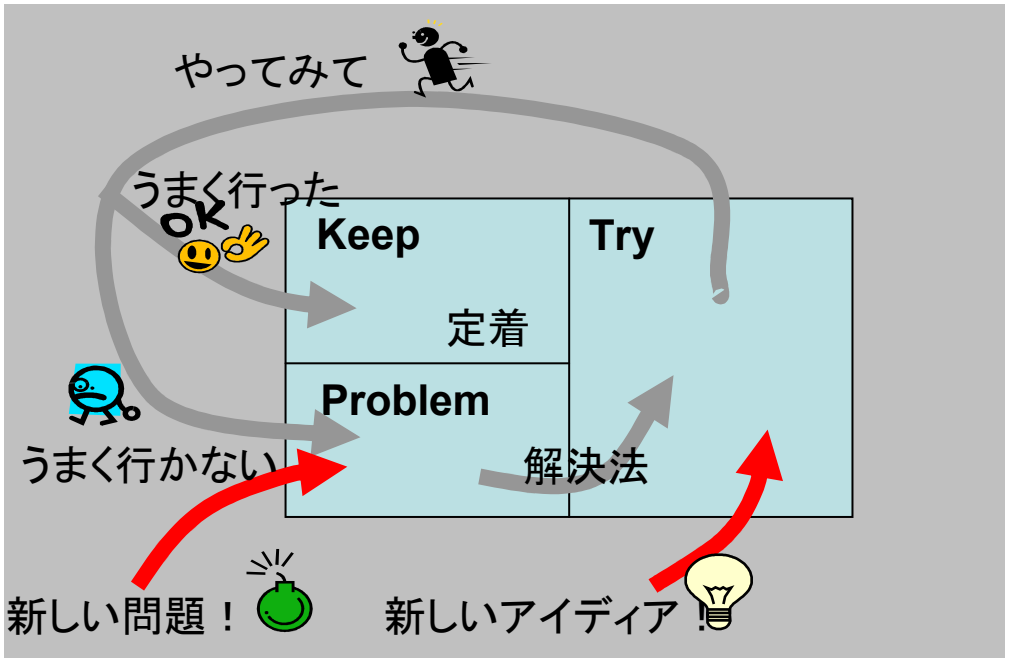
Tips

Tips

- プロジェクタ
  - ツールとホワイトボードを組み合わせて
- ふりかえり
  - 自分たちのベスト “Keeps” セットをさがせ。
- リバースエンジニアリング
  - 現在のコードベースを可視化。
- マインドマップ

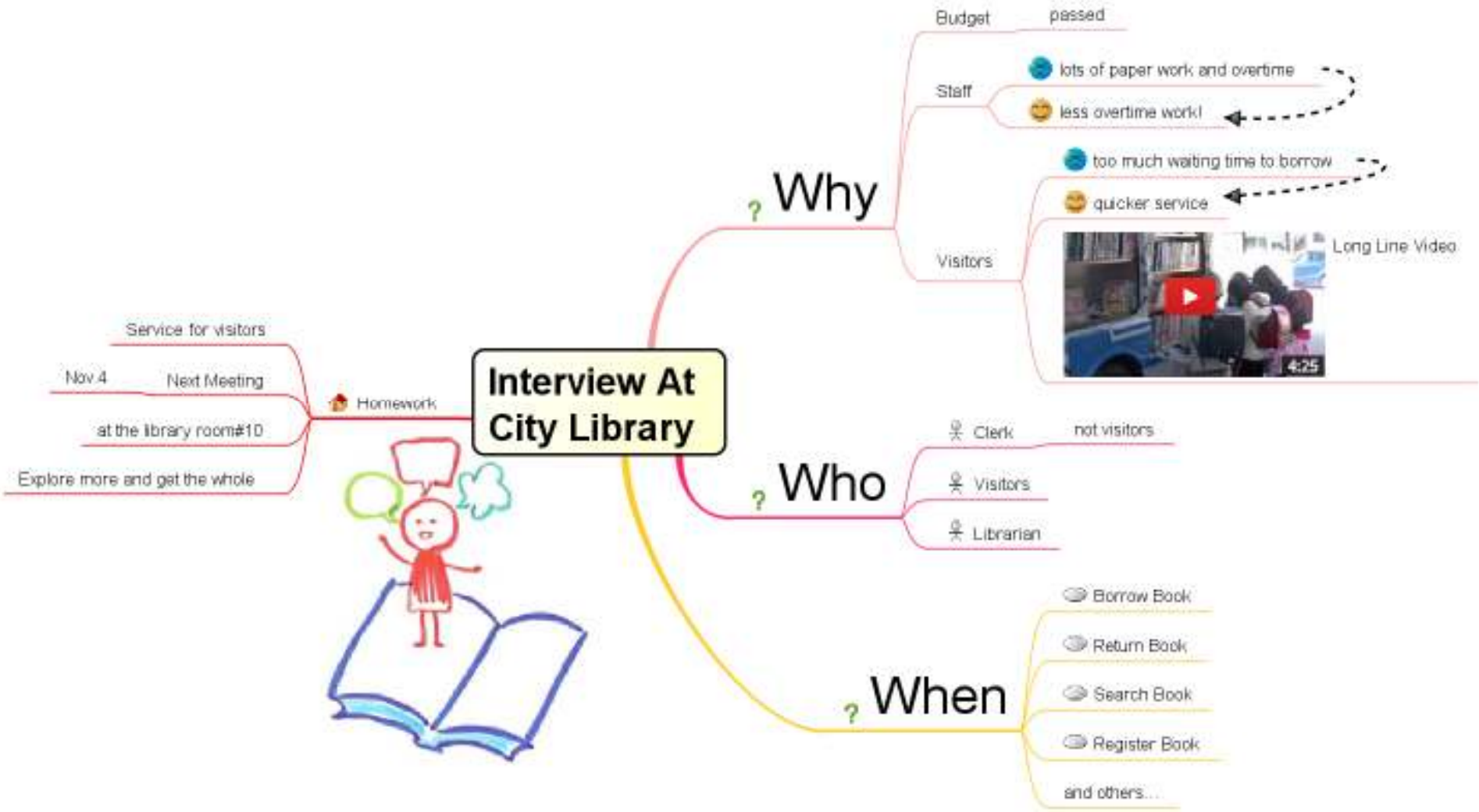
ふりかえり

- Keepするモデルは文脈依存する
- やってみてチームで合意。

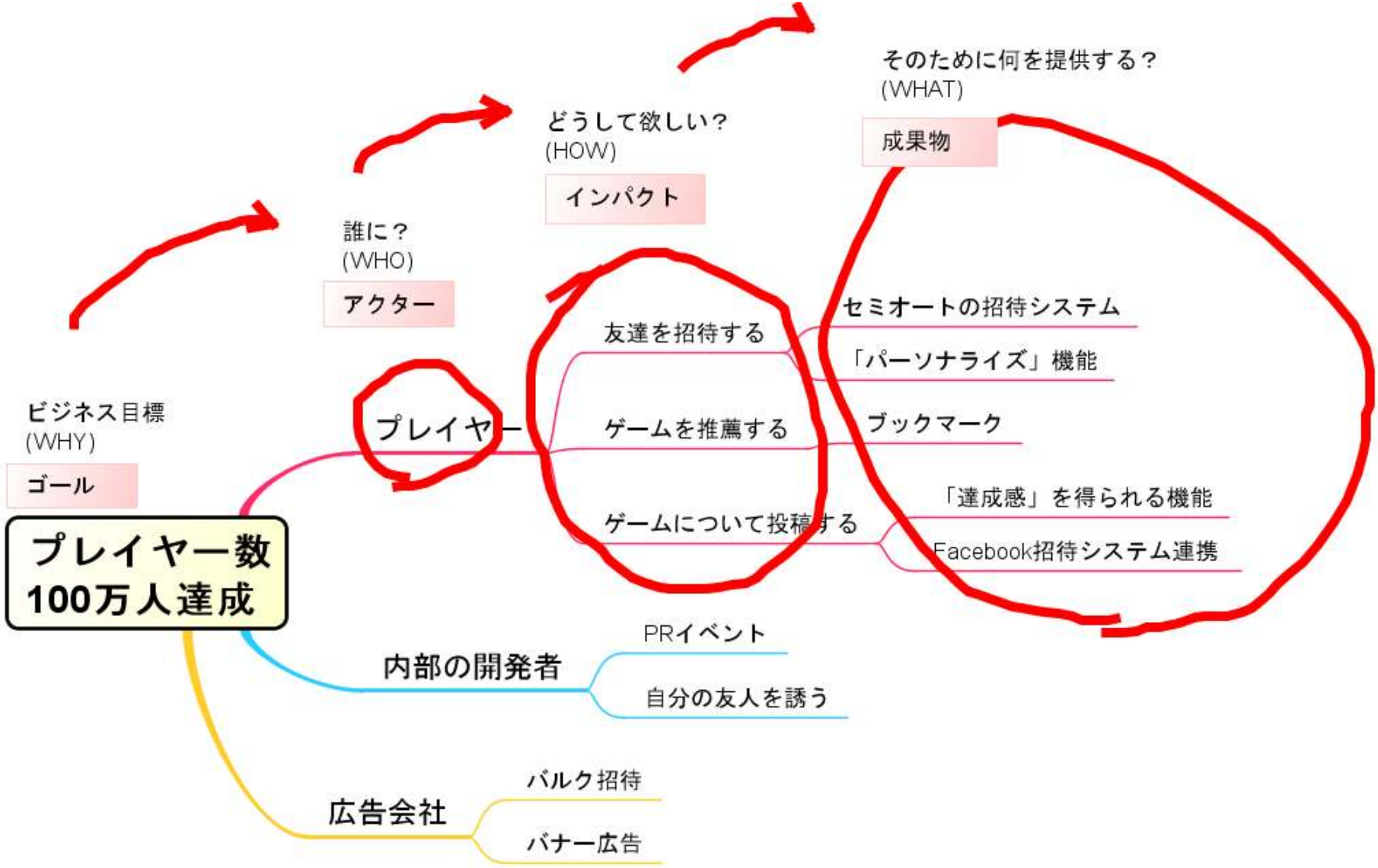


ふりかえりがカイゼンを導く

Mindmap Modeling



Impact Mapping



まとめ

- 設計はソフトウェア開発においてこれまでもっとも大事な部分であるし、アジャイル開発でも、もっとも大事な部分であり続けている。
- もっともシンプルなモデルセットを選び、それらをメンテナンスする。そして他のものはコードとテストにしてしまう。
- 自分たちのKeepをさがせ。
- 会話のためにモデルを利用
- “Divide and Conquer” でなく
  “Conquer THEN Divide”
- 形式化しにくいアイディアには、マインドマップを利用。

www.infoq.com/articles/kenji-modeling-agile

About Us | About You | Our Contributor Team | Purpose Index | Exclusive updates on:

Facilitating the spread of knowledge and innovation in enterprise software development

Search | Kenji

InfoQ

En | 中文 | 日本 | Fr | Br

858,214 Sep unique visitors

Development | Architecture & Design | Process & Practices | Operations & Infrastructure | Enterprise Architecture

QCon
San Francisco 2013
Nov 11 - Nov 15
London 2014
March 3 - Mar 7

Mobile | HTML5 | JavaScript | APM | PHP | Big Data | API Design | Cloud | IBM Code Rally | All topics

You are here: InfoQ Homepage ▸ Articles ▸ Modeling in the Agile Age: What to Keep Next to Code to Scale Agile Teams

Modeling in the Agile Age: What to Keep Next to Code to Scale Agile Teams

Posted by Kenji Hiranabe on Oct 07, 2013 | 6 Discuss

Share | My Reading List | Read later

Now that Agile methods have become mainstream in software development, working code (and automated tests) are being considered as the most important team artifacts.

Is modeling not needed any more? Is UML dead? I think that's not true.

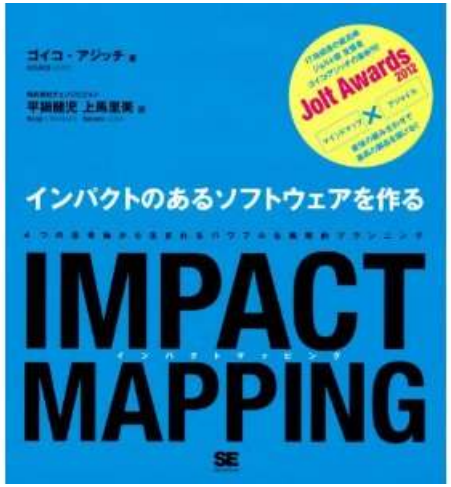
In this article, I'll explore the spaces where modeling fits and plays an important role in this Agile age, especially when development scales to multiple teams and a shared understanding of the system's "Big Picture" becomes essential

Popular | Week | Month | 6 months

The Functional Database

Visual Studio 2013 Release Date Announced, RC Available Now

Search: InfoQ Kenji



kenji.hiranabe@change-vision.com
kenji.hiranabe
@hiranabe
hiranabe