

---

# 現場を助けるモデリング

---

2007年9月7日(金)

Modeled by 

---



---

# 自己紹介

---



## (株)オージス総研『オブジェクトの広場』

弊社 Web サイトにて公開中の月刊オンラインマガジン

<http://www.ogis-ri.co.jp/otc/hiroba/index.html>

## スピーカ紹介:

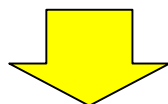
- 山内亨和 (AMS部モデリング技術チーム)  
開発プロセス導入・定着化・改善に従事  
オブジェクトの広場にて数本の記事を連載中
- 田中恒 (組み込みソリューション部第一チーム)  
某自動車部品メーカーにて組み込み系システムのコンサルティングに従事  
オブジェクトの広場にて『モデリングカフェ「Square」』を連載中

# はじめに

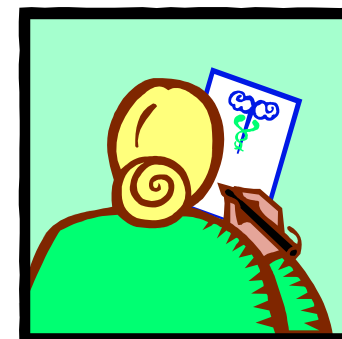
モデリングしなくても  
実装できるよね

役に立ったのか  
よく分からなかった

数年前、モデリングって流行ったなあ



だまされてた！？





---

# 私達のメッセージ

---



**モデリングは手段！**

必要に応じて 目的にあったモデリングしよう！

# モデリングが役立つ状況

## 代表例

開発の状況	
他人と仕事する	打ち合わせをする
	成果物を作成する
問題解決 ・業務分析 ・要求分析 ・設計など	問題を理解する
	問題の解決策をテストする

# この発表では

モデリングの成功体験を共有したい！

開発の現場をロールプレイで再現



モデリングが  
役立つ状況は？

どう使う？



---

# とある開発現場のお話

---



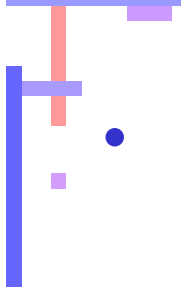
---



---

# とある開発現場

---

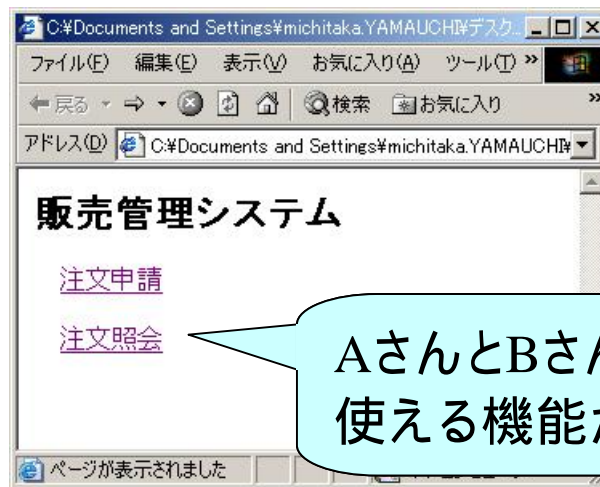
- 
- 登場人物
    - 先輩SE
      - 豊富な経験がある
    - 新人SE
      - 学生時代にプログラミング経験あり(やや老け顔)
  - シチュエーション
    - 販売管理システムのアーキテクチャを開発している
    - アーキテクチャの一部を新人SEが担当した
      - 新人SEは認可機能を実装した
    - 設計のレビューをしているところ...



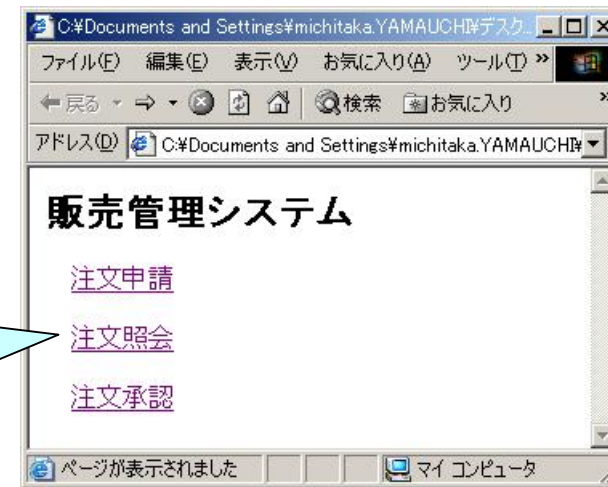
# 認可機能への要求

- 概要
  - ユーザーに機能を使用する権限があるか判定すること
- 例

Aさんのメニュー



Bさんのメニュー



AさんとBさんで  
使える機能が違う

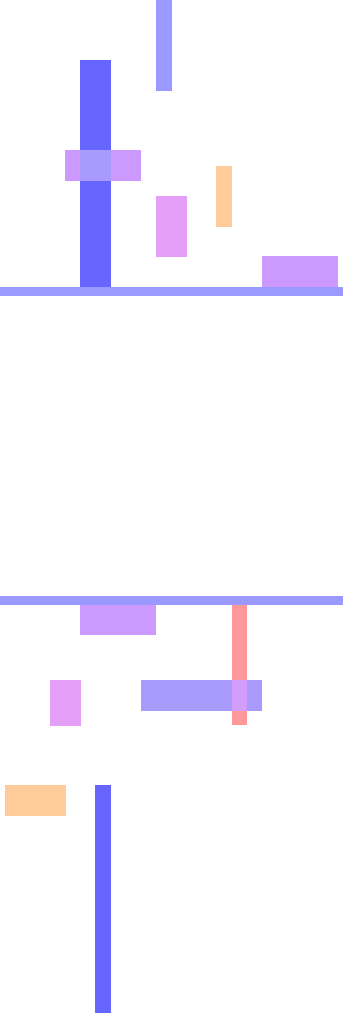
# 認可機能への要求 ~ユーザーとロール~

- ユーザーとロールとは
  - ユーザーは様々なロールを持てる
- ロールと機能
  - ロールとはユーザーが使える機能の集合

ロール: 一般社員、上長

Aさんは一般社員  
Bさんは一般社員でもあり、上長でもある

		機能		
		注文申請	注文照会	注文承認
ロール	一般社員			-
	上長			



---

# シーン1 設計をレビュー中

---

ここで伝えたいのは

開発の状況	モデリングの目的
打ち合わせをする	自分の考えを相手に伝える
問題を理解する	情報を整理して問題を理解する

---

# 新人がソースで説明するも...

```
// AuthUserクラス
package jp.co.xxx.auth;
import java.util.HashSet;
import java.util.Iterator;
public class AuthUser {
    private String userId;
    private HashSet roles;
    public AuthUser(){
    }
    public AuthUser(String userId){
        this.userId = userId;
    }
    public void addRole(AuthRole role){
        roles.add(role);
    }
    public String getUserId() {
        return userId;
    }
    public void setUserId(String string) {
        userId = string;
    }
    public boolean isPermitted(String functionCd){
        for(Iterator ite = roles.iterator(); ite.hasNext());{
            AuthRole role = (AuthRole)ite.next();
            if(role.isPermitted(functionCd)){
                return true;
            }
        }
        return false;
    }
}
```

- こんな経験ありませんか？
  - 新人SE(レビューイ)の側では
    - がんばって話しているのに伝わらない
  - 先輩SE(レビューア)の側では
    - 要点が分からなければ、問題も指摘できない



これじゃ、  
分らん！

# 新人の失敗

## ✓ がんばってしゃべっているのに伝わらない

- しゃべることだけで伝えようとした
- 相手に伝わっているか確認できない

## □ 打ち合わせでは

- 会話 + モデリングでコミュニケーション!
  - 話すだけでなく、書く
  - 間違いが見つければ、その場で書き直す

## ✓ 要点が分からない

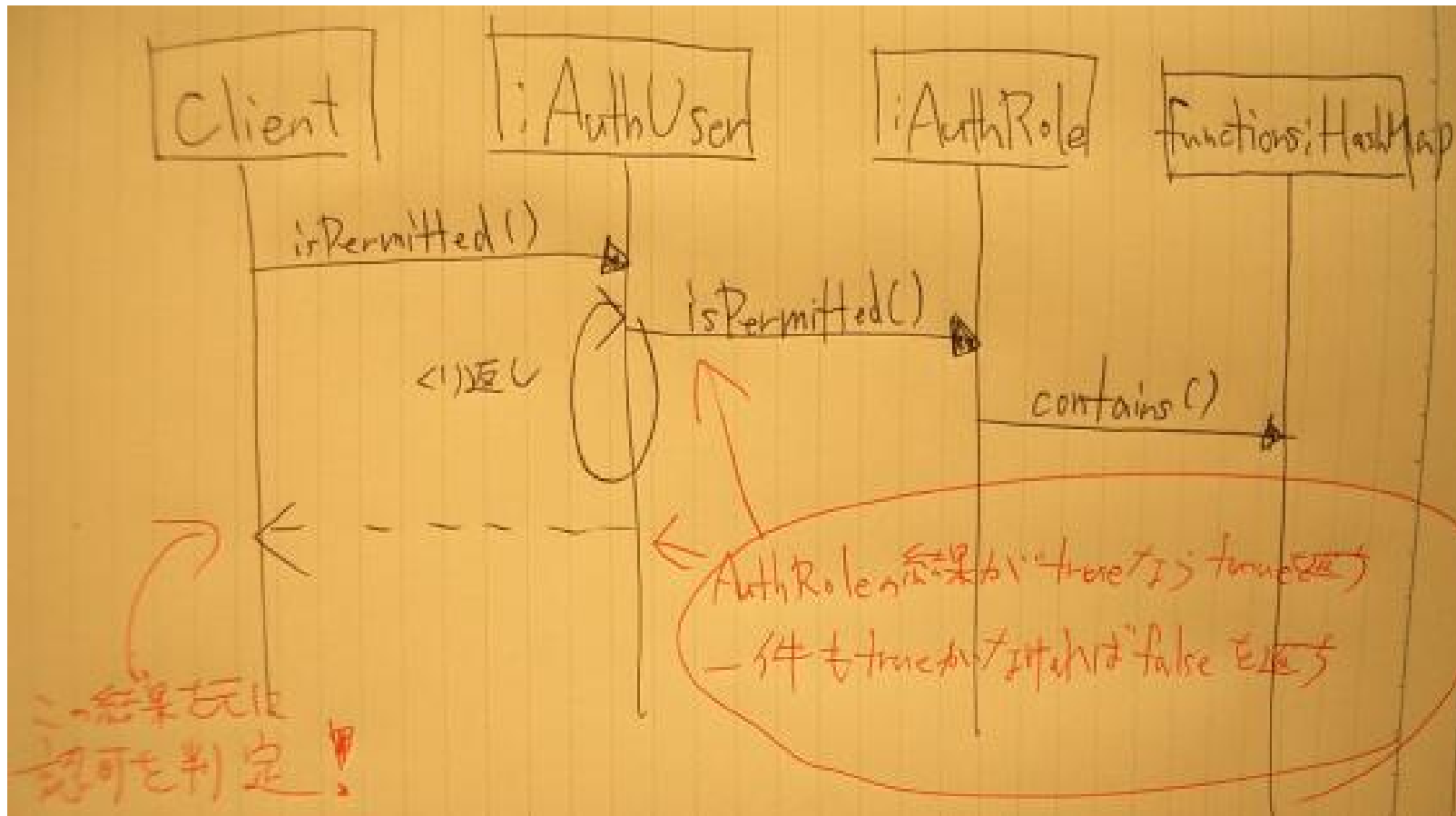
- ソースコードで伝えた
- 機能と各クラス・メソッドの関係を知りたい
- ソースコードと機能では情報の粒度が違う

## □ 問題を理解するには

- 情報を整理して
- 図を書こう!

# モデルの例

例: 打ち合わせ中に即興で書いたシーケンス図



開発の状況

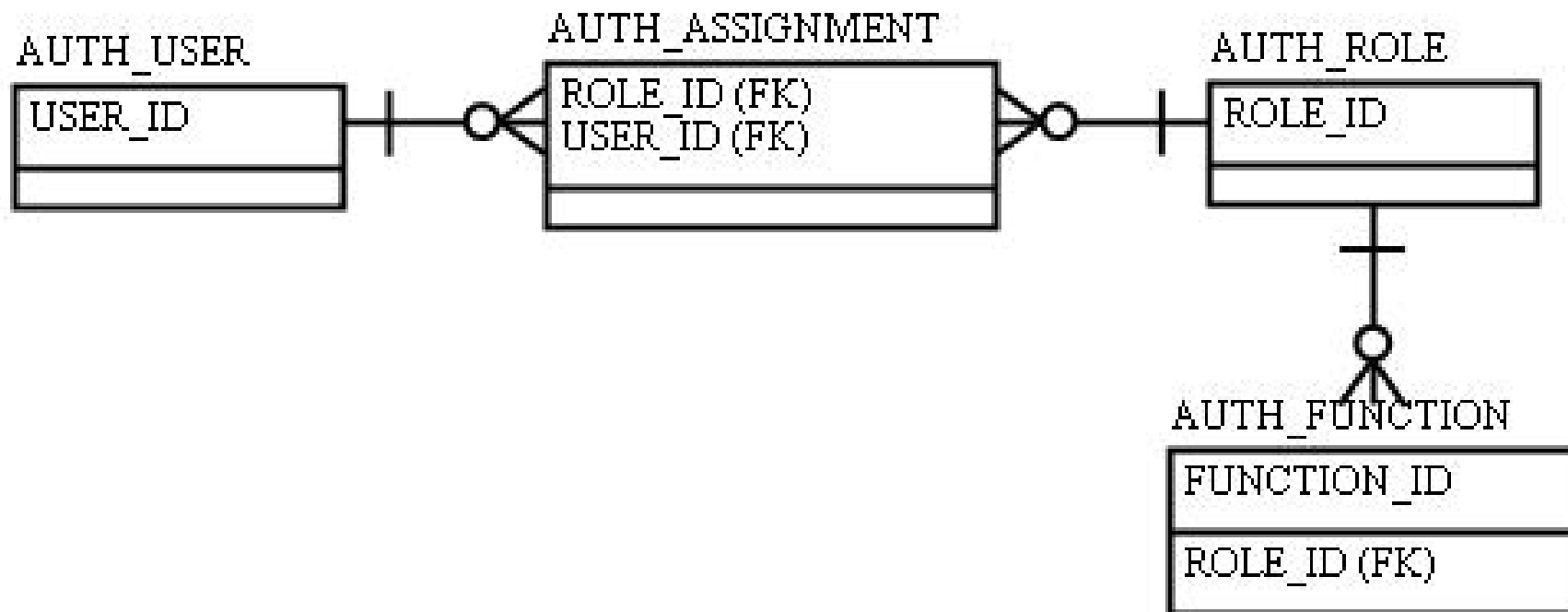
打ち合わせをする

問題を理解する

# 注意！問題にあった図を

- 例えば。。。
  - テーブル設計をしたい場合
  - クラス図 or ER図

例：ER図 (IE表記)



# おさらい

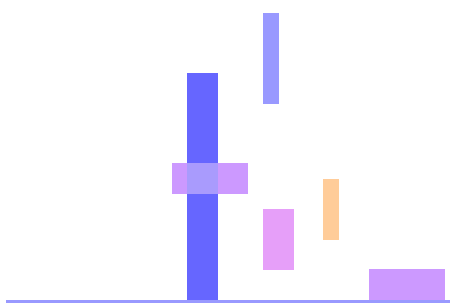
開発の状況

打ち合わせをする

問題を理解する

- 打ち合わせでは
  - 会話 + モデリングでコミュニケーション！
- 問題を理解するために
  - 情報を整理し、問題にあった図を使おう！





---

# シーン2 設計をテスト中

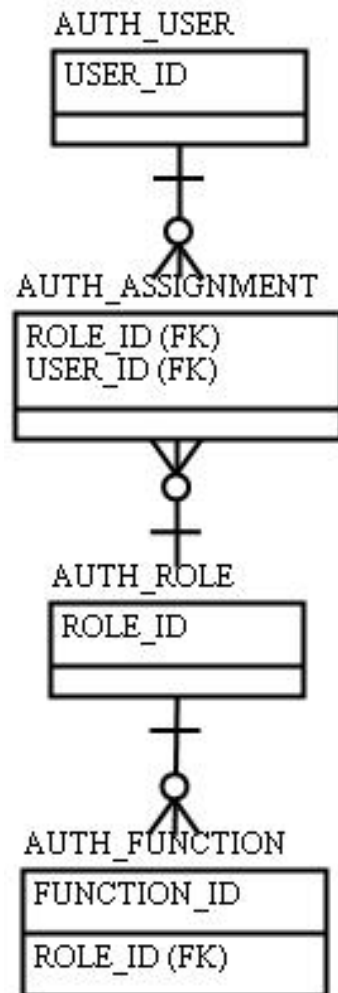
---

ここで伝えたいのは

開発の状況	モデリングの目的
問題の解決策をテストする	先に進む前にテストして、手戻りを減らす

---

# ER図は完成したものの...



- こんな経験ありませんか？
  - できあがったモデルが正しいのか不安...
  - テストで致命的なバグが見つかり、大変だった



これって  
正しいの？

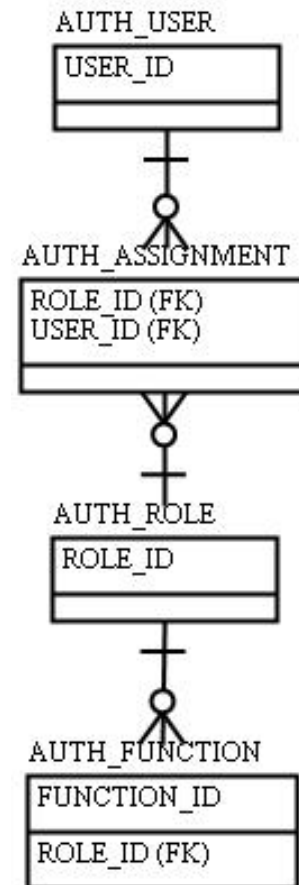
# 解決策は、作る前にテスト

- 解決策は正しいの？
  - 後で見つかった間違いの対応コストは高い！
  - レビューだけでなく、テストもしよう！
- モデルをテストって、どうすればいいの？
  - テスト対象を選ぶ
  - テストケースを用意する
  - テストツールを用意する
  - テストする
  - テスト結果を確認する

# 設計をテストしよう！

- テスト対象を選ぶ
  - 今回ならER図
- テストケースを用意しよう
  - 設計をテストするなら要求をテストケースに
- テストツールを用意しよう
  - テーブル設計をテストするならオブジェクト図

Aさんは一般社員  
Bさんは一般社員でもあり、上長でもある



		機能		
		注文申請	注文照会	注文承認
ロール	一般社員			-
	上長			

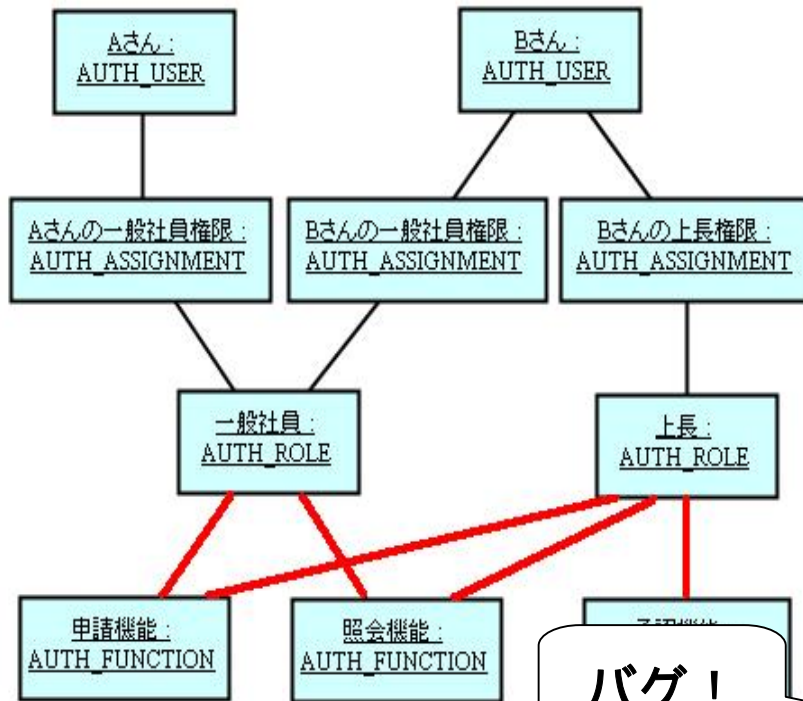
# 設計をテストしよう！

テストケース: 要求

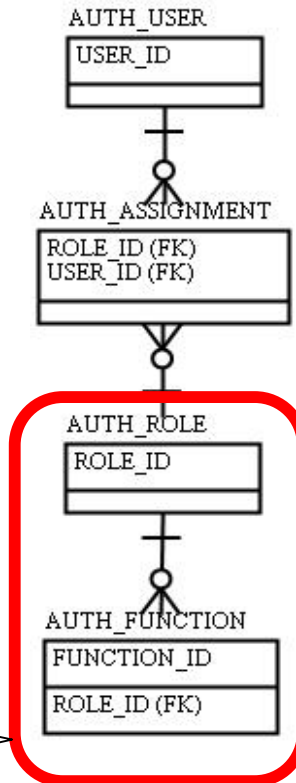
Aさんは一般社員  
Bさんは一般社員でもあり、上長でもある

	機能	注文申請	注文照会	注文承認
		一般社員		
上長				

テスト結果: ER図のオブジェクト図



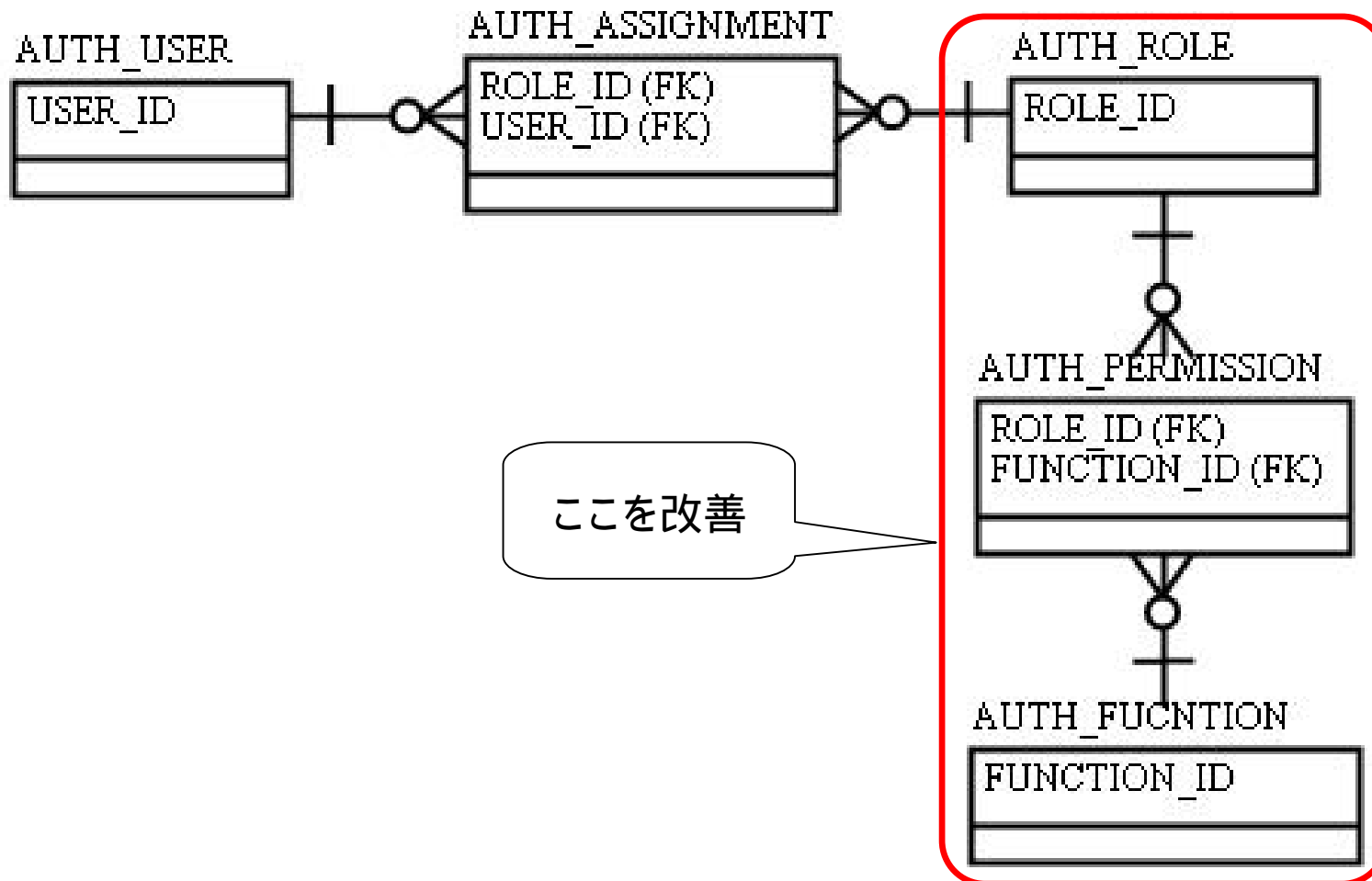
テスト対象: ER図



# 完成版ER図

開発の状況

問題の解決策を  
テストする



# おさらい

- 問題の解決策が見つかったら
  - レビューだけでなく、テストしよう！
    - 後で見つかった間違いの対応コストは高い



---

# シーン3 アーキテクチャの成果物を作る

---

ここで伝えたいのは

開発の状況	モデリングの目的
成果物を作る	読み手が知りたいことを書く

---



# 成果物を作れと言われたけれど

- こんな風に思っていないませんか？
  - 時間がもったいない
  - 作っても意味がない
  - 作れといわれたから仕方なく作ってる

作る必要ないんじゃないの！



プログラムだけで  
十分だろ！

# 成果物は誰のため？

- 開発現場の悲しい勘違い
  - 成果物は作ることが目的だと思っていないか？
  - 成果物が何の役にも立っていないと思っていないか？
- 成果物は読み手のために作る！
  - 逆説 読み手の役に立たない成果物はいらない

# 読み手のために成果物を作る

- 読み手は誰？
  - 後工程の担当者、運用担当、他
- 読み手はどんな仕事をしている？
  - 実装、マスタデータの登録、他
- 読み手が知りたいことは？
  - 実装の指針、テーブルの説明、他

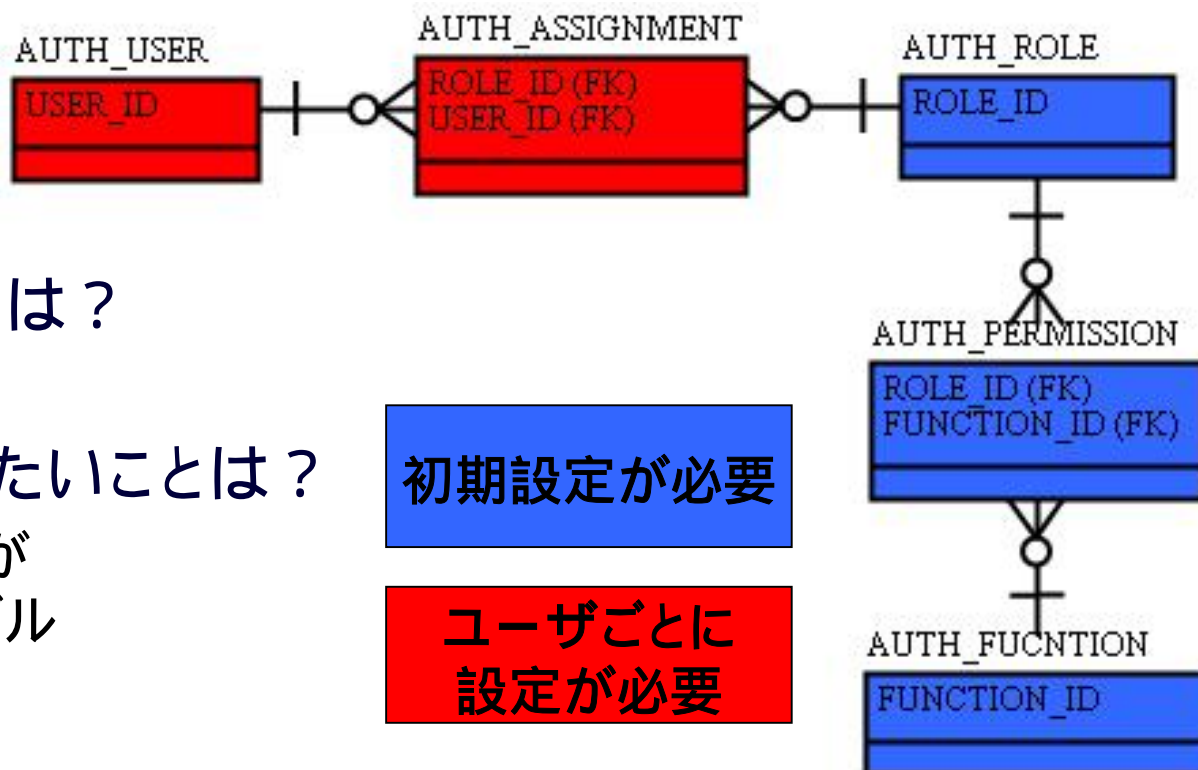
開発の状況

成果物を作る

読み手のためのモデルを作ってみよう！

例えば：

- 読み手は？
  - 開発者
- 読み手の仕事は？
  - テスト
- 読み手が知りたいことは？
  - テストデータが  
必要なテーブル



初期設定が必要

ユーザごとに  
設定が必要

# おさらい

- 成果物は誰のために作る？
  - 読み手のための成果物を作る



# まとめ



# モデリングの目的

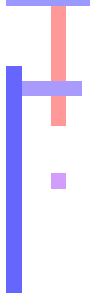
開発の状況		モデリングの目的
他人と 仕事する	打ち合わせをする	自分の考えを相手に伝える
	成果物を作成する	読み手が知りたいことを残す
問題解決 ・業務分析 ・要求分析 ・設計など	問題を理解する	情報を整理して問題を理解する
	問題の解決策を テストする	先に進む前にテストして、手戻りを減らす



---

# おしまい

---

- 
- モデリングは手段
    - 必要に応じて
    - 目的にあったモデリングしよう！







---



ご清聴、ありがとうございました。

今回の発表資料は  
『オブジェクトの広場10月号』  
にて掲載します。

