

さまざまな視点に 合わせた仕様書の 作成・維持の支援手法

ソフトウェア品質シンポジウム2017

2017/9/16

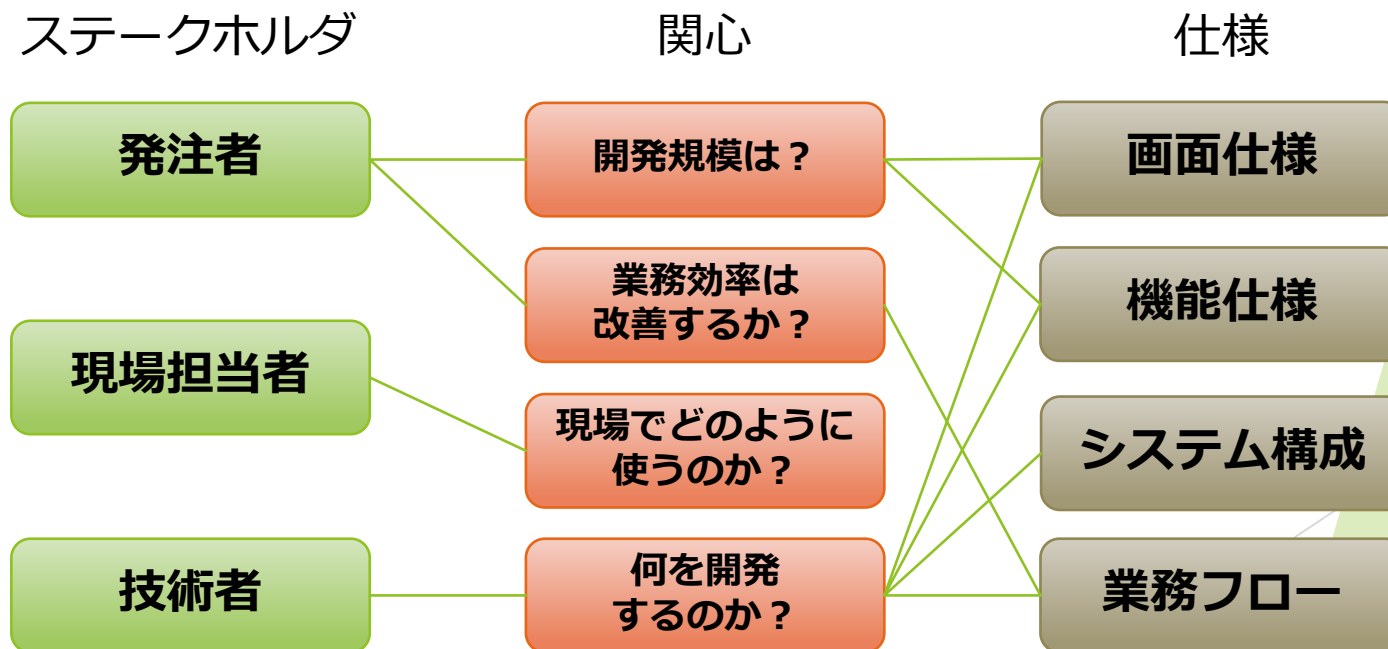
フリーランスエンジニア 酒井 雄太

概要

- ▶ 仕様書の課題
 - ▶ 仕様書の記述があいまいである
 - ▶ ステークホルダ毎に関心事は異なる
 - ▶ 多くの説明資料を用意すると不整合が生じる
- ▶ 課題解決のアイデア
 - ▶ 一つの厳密な仕様書から、さまざまな視点の仕様書を自動的に生成する
- ▶ アイディアの検証
 - ▶ 一つの仕様書・さまざまな仕様書の自動生成
 - ▶ 一定の効果と拡張性の感触を得た

ステークホルダ毎に関心事は異なる

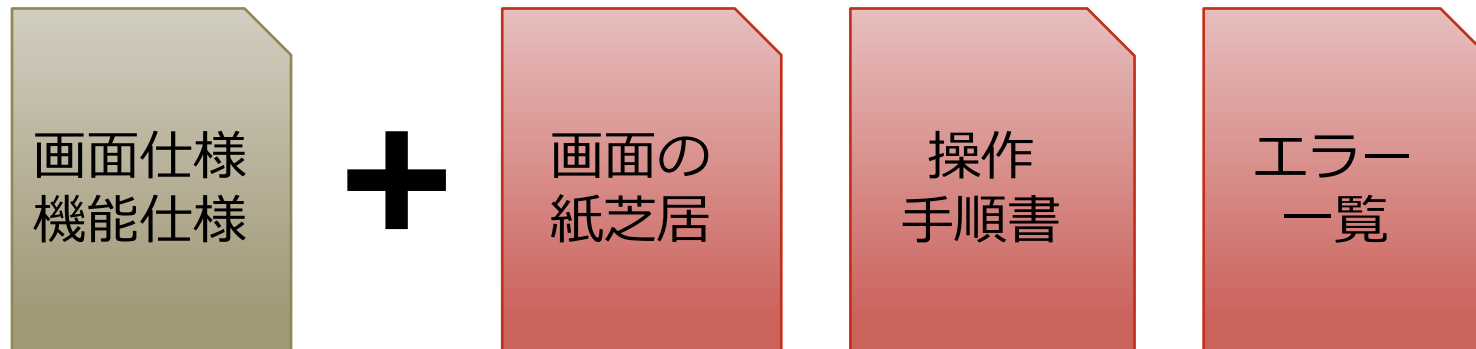
- ▶ ステークホルダ間のコミュニケーションのために仕様書が用いられるが・・・



技術者ではない読み手には不適切な形式

多くの説明資料を用意すると 不整合が生じる

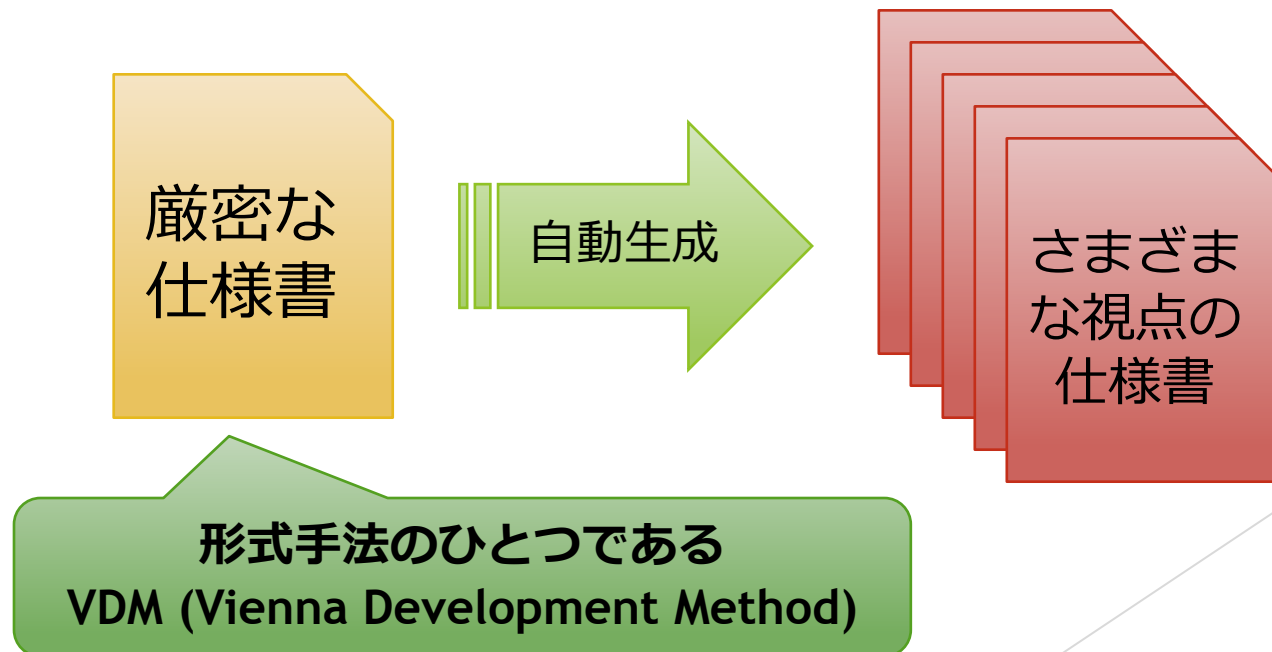
- ▶ 読み手に合わせた説明資料を用意することが多い



曖昧な仕様をもとに作成された資料は
不整合が生じる

課題解決のアイデア

- ▶ 一つの厳密な仕様書から，さまざまな視点の仕様書を自動的に生成する



形式手法 (Formal Methods) とは

数理論理学に基づく科学的な裏付けを持つ仕様記述言語を用いて設計対象の機能や性質を表現する・・・

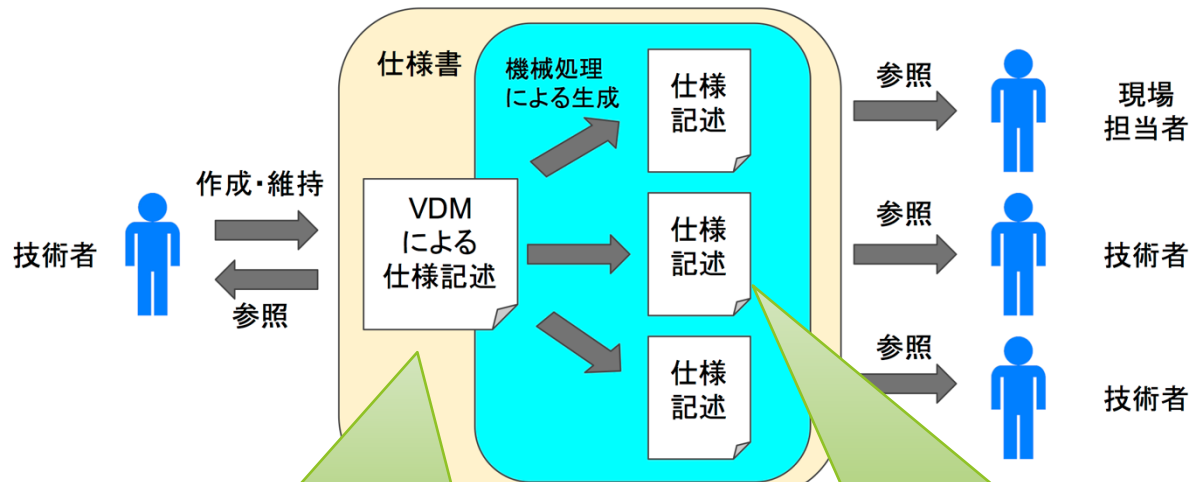
- ▶ 仕様や設計を「きちんと」書こうとする過程により
 - ▶ 曖昧さを解消することになる
 - ▶ 不正確さ, 不整合も表面化する
 - ▶ システムに対する理解が深まる
- ▶ 仕様や設計を「きちんと」書いた結果により
 - ▶ 誤解なく情報を共有できる
 - ▶ **科学的・系統的な分析・検証ができる**
 - ▶ 分析・検証をツールに支援させることができる

VDM (Vienna Development Method) とは

- ▶ 仕様記述に特化した言語をもちいることで、**仕様を厳密に記述**できる。自然言語の仕様書に含まれる曖昧さを取り除くことができる
- ▶ (用語の統一、構文チェック、型チェックなどを) ツールがサポートしてくれる
- ▶ 仕様を**動作させる**ことや、テストができるため、開発者や評価者になじみやすい
- ▶ 日本語のツール、書籍、Web情報が、他の形式手法に比べ多い。
- ▶ オープンソースのツールが公開されている
- ▶ 文書やプログラムの記述力や論理的な思考力の向上に役立つ

さまざまな視点の仕様書を自動的に生成する

- ▶ 一つの仕様書を作成・維持するだけで、複数の仕様書を生成する



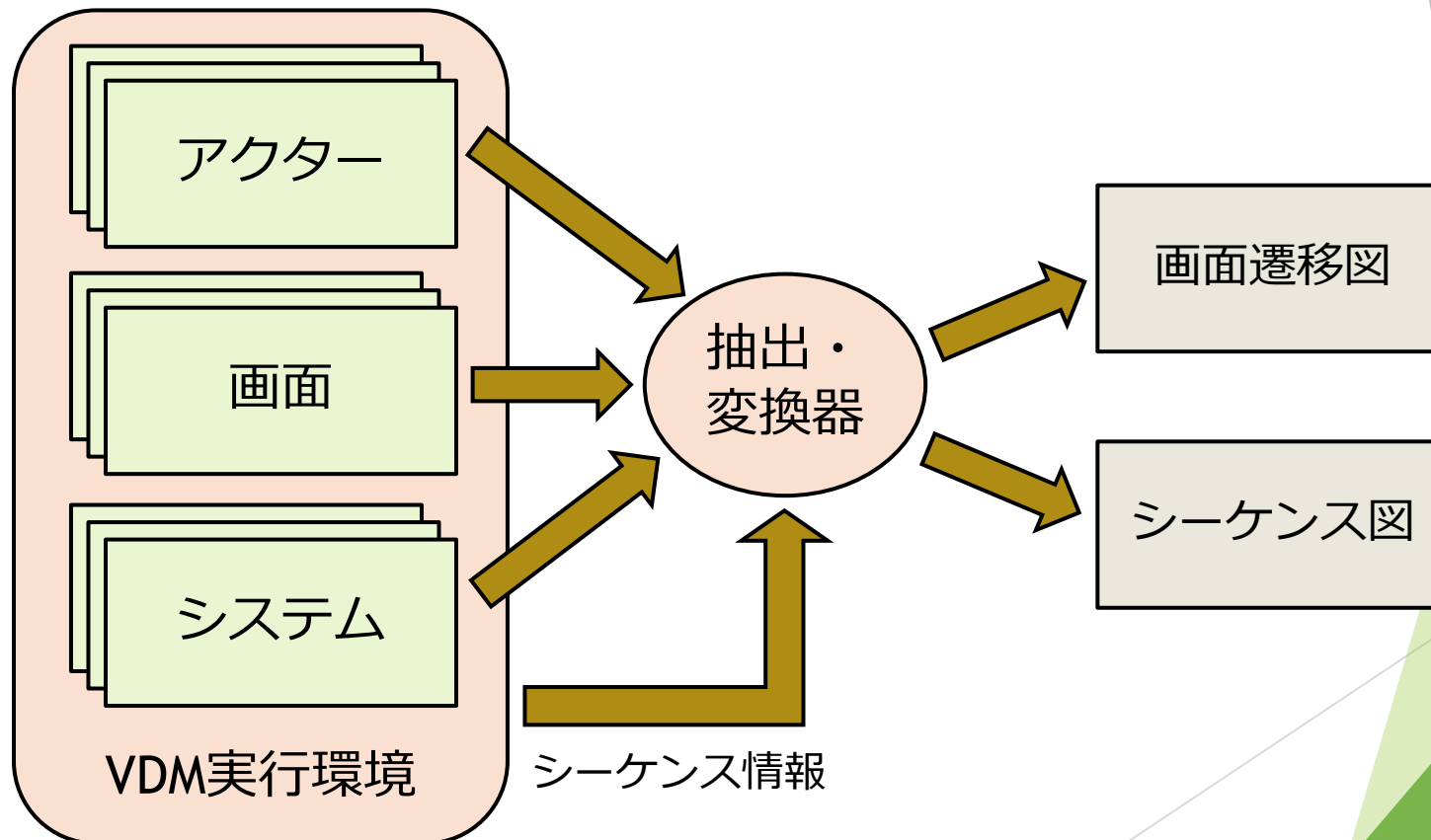
さまざまな視点で読むことができる

ある視点に特化しているので読みやすい

さまざまな視点に合わせた
仕様書の作成・維持の支援手法

自動的な仕様書の生成

- ▶ 本研究では、画面遷移図とシーケンス図を生成することをゴールとした



VDMによる仕様記述の方法

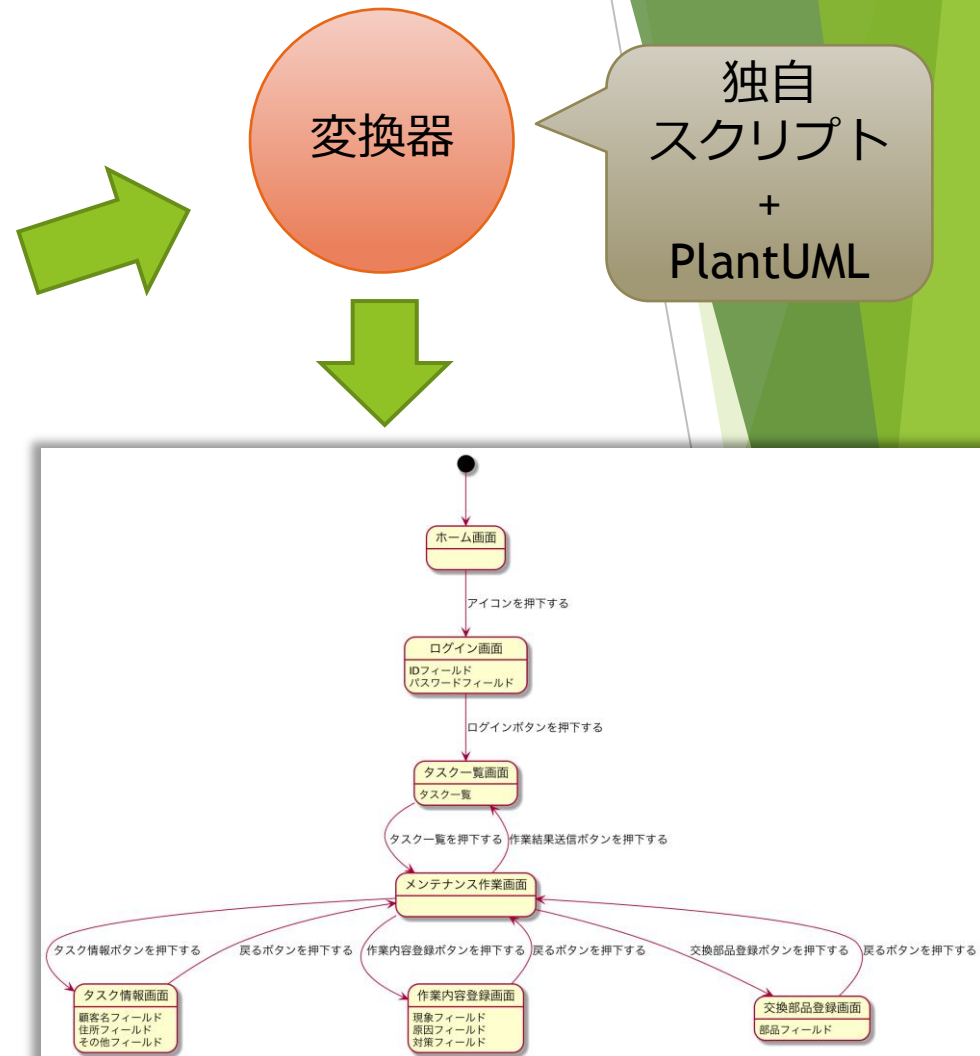
- ▶ VDMによる仕様書は、視点の異なる仕様書を生成することを狙った書き方をする
 - ▶ 各画面を1つのVDMクラスとして記述することで、画面毎の仕様をまとまった形で抽出しやすくする
 - ▶ 画面に対する一連の操作の具体例を記述することで、基本的な使い方の視点の仕様を生成できるようにする
 - ▶ 全ての操作に対して、ログ出力を記述することで、内部の振る舞いについての仕様を生成できるようにする

画面遷移図の生成

VDMによる仕様記述

```

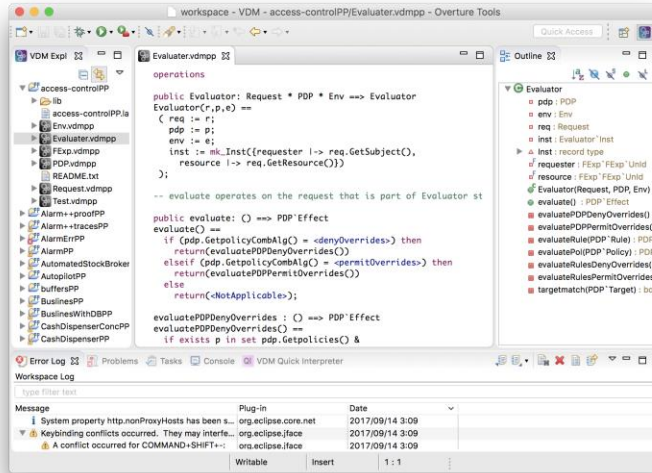
-- このクラスは ログイン画面 についての仕様記述である
class ログイン画面 is subclass of 画面
instance variables
-- UI要素をインスタンス変数として表現する
public IDフィールド : token := mk_token("NULL");
public パスワードフィールド : token := mk_token("NULL");
<省略>
--画面遷移を伴う操作の戻り値は 次の画面のインスタンスである
public ログインボタンを押下する : () ==> 画面
ログインボタンを押下する() ==
(
  --操作に対して、VDM++を実行した際に出力されるログが埋め込まれている
  io.print("ログイン画面.ログインボタンを押下する\n");
  let
    sys = システム`システムインスタンス,
    認証情報確認結果 =
    sys.認証情報を確認する(IDフィールド, パスワードフィールド)
  in
    if 認証情報確認結果 = true
    then
      (
        io.print("ログイン画面.タスク一覧画面に遷移する\n");
        return new タスク一覧画面();
      )
    else return self
  )
<省略>
end ログイン画面
  
```



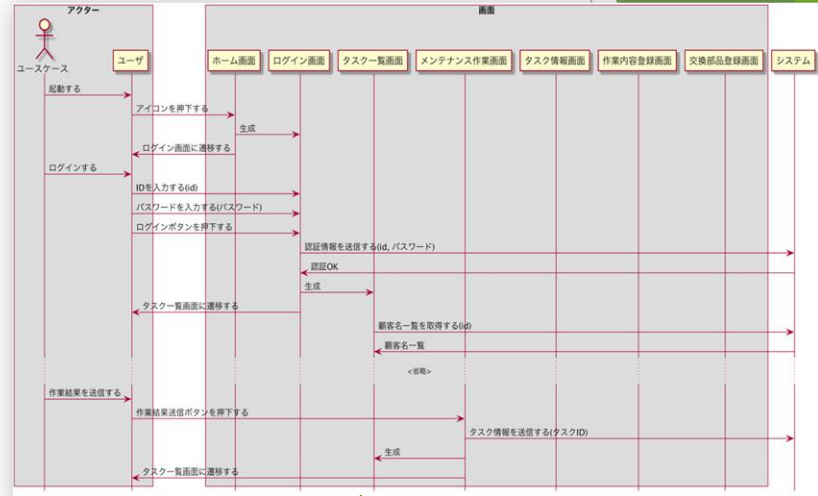
画面遷移図

シーケンス図の生成

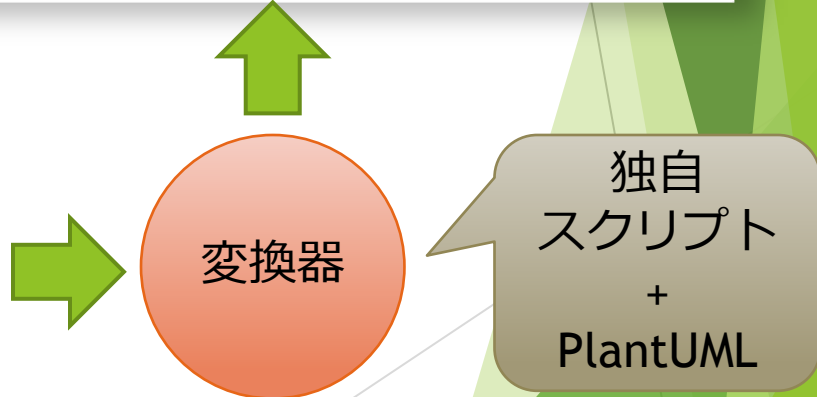
OvertureTool(VDM実行環境)



シーケンス図



ユースケース.基本シナリオを実行する
 ユーザ.起動する
 ホーム画面.アイコンを押下する
 ホーム画面.ログイン画面に遷移する
 ユーザ.ログインする
 ログイン画面.IDを入力する
 ログイン画面.パスワードを入力する
 ログイン画面.ログインボタンを押下する
 .
 .
 .



シーケンス情報(ログ出力)

結果と今後の展望

- ▶ 本提案手法の目指すこと
 - ▶ 読み手のさまざまな視点に合わせた仕様記述を作成し、複数の仕様記述間の不整合を防ぐ
- ▶ 期待する初期の効果を確認した
 - ▶ VDM仕様記述から、シーケンス図と画面遷移図を生成
 - ▶ 仕様記述間の不整合を防ぐ
 - ▶ 仕様書メンテナンスコストの削減
 - ▶ 提案手法の拡張の可能性
- ▶ さらなる発展が必要
 - ▶ より多くの形式の仕様書を生成する手法の考案
 - ▶ 読み手に応じた適切な形式や視点の追求

まとめ

- ▶ 関心事の異なるステークホルダーに合わせて多くの文書を作成・維持することは大変だが、形式手法はそのような課題に対しても有効である
- ▶ 今後は、技術者ではない読み手による評価などを行い、提案内容をより良いものにして、コミュニケーションの道具として使えるものに育てていきたい

ご静聴
ありがとうございました。