

# テストの哲学的解釈

株式会社日本システムディベロップメント  
アジャイルプロセス協議会 知働化研究会  
時本 永吉

# 結論

アジャイルのテストの考え方って素晴らしい

# ふと思っってしまった

テストは何故するの？

テストは必要なの？

テストをしないと何が困るの？

# 良いコードをかきたい

何がしたいか明確なコードを記述する



言葉（メソッド）を作り、  
文章として読めるものを記述する



オブジェクト指向

# 良いコードをかきたい

↓

抽象的な言葉（クラス／メソッド）と  
具体的な文脈（シナリオ）の記述する

↓

言葉が文脈に織り込まれたとき、  
正しい意味（意図）で解釈できるかを  
確認する

# テストって

文脈（シナリオ）に適合するよう  
テストケースを考える

だったら、

シナリオとテストケースを考えるのは  
同じタイミングじゃないか？

別々に行ったテストに何の意味があるのか？

# テストって

設計者とテストケース設計者が別の場合、  
二人が意識違いをしていないか、  
どうやって確認してるんだろう

# テストって

別々にすると無駄じゃない？  
別々にすることで、  
設計者、実装者、テストケース設計者  
の三人の考えが発生するし、  
それだけ記述も重複してしまう

# テストをしないって意味じゃない

主張には根拠が必要  
コードにテストが必要なのは当たり前

気になったのは、  
テストが根拠を主張するためものではなく、  
契約（承認が必要なドキュメント）  
でしかないこと

# 契約に縛られてる

設計書書いて、承認もらって、  
コード書いて、承認もらって、  
テストケース書いて、承認もらって、  
テストを実行して、承認もらって・・・

# 試してみた

デスマ気味で承認プロセスが  
アバウトになったプロジェクトで  
アジャイルに振る舞ってみた

# 試したプロセス

設計書を受け取る



設計書に対するテストケースを書いて、  
漏れているケースを確認する  
(テストデータと検証環境も整えておく)



実装する



テストを実行して、  
検証環境で期待通り動いたことを確認する

# 得られた効果

実装前に  
意味・意図を確認するため、  
名前が付けやすい  
(メソッドを作る単位が明確になる)

# 得られた効果

実装時に悩まない  
テストケースは設計書を補完するもの

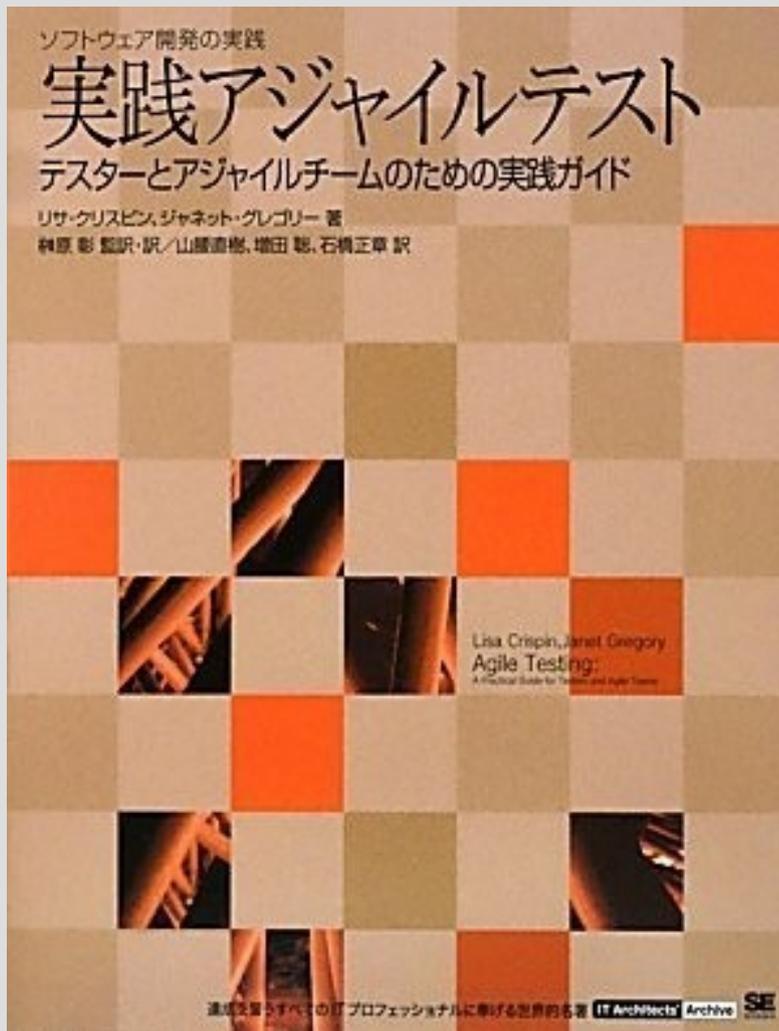
だから

早い  
読みやすい  
誤解釈・漏れがない

# 経験から至った結論

テストをプロセスではなく、  
確認する・主張する根拠とする  
という考え方で行くと  
早く、確かなものが作れる

# アジャイルテストの考え方



- アジャイルチームはアプリケーションの品質が既知の状態にあることを保証する
- テストと品質がアジャイル開発の中心
- アジャイルチームは、ソフトウェアが顧客の要求に合うように取り組んでいます

# もう一步先へ

テストは品質のために行う

品質とは、いかに顧客を満足させられたか

顧客を満足させるために  
顧客との意思疎通の確認に時間を費やしたい

# 考えてること

テストのこの考え方から  
開発プロセスって変えられないかな？

アジャイルマニフェストを  
哲学的（意味論的）に解釈してみる

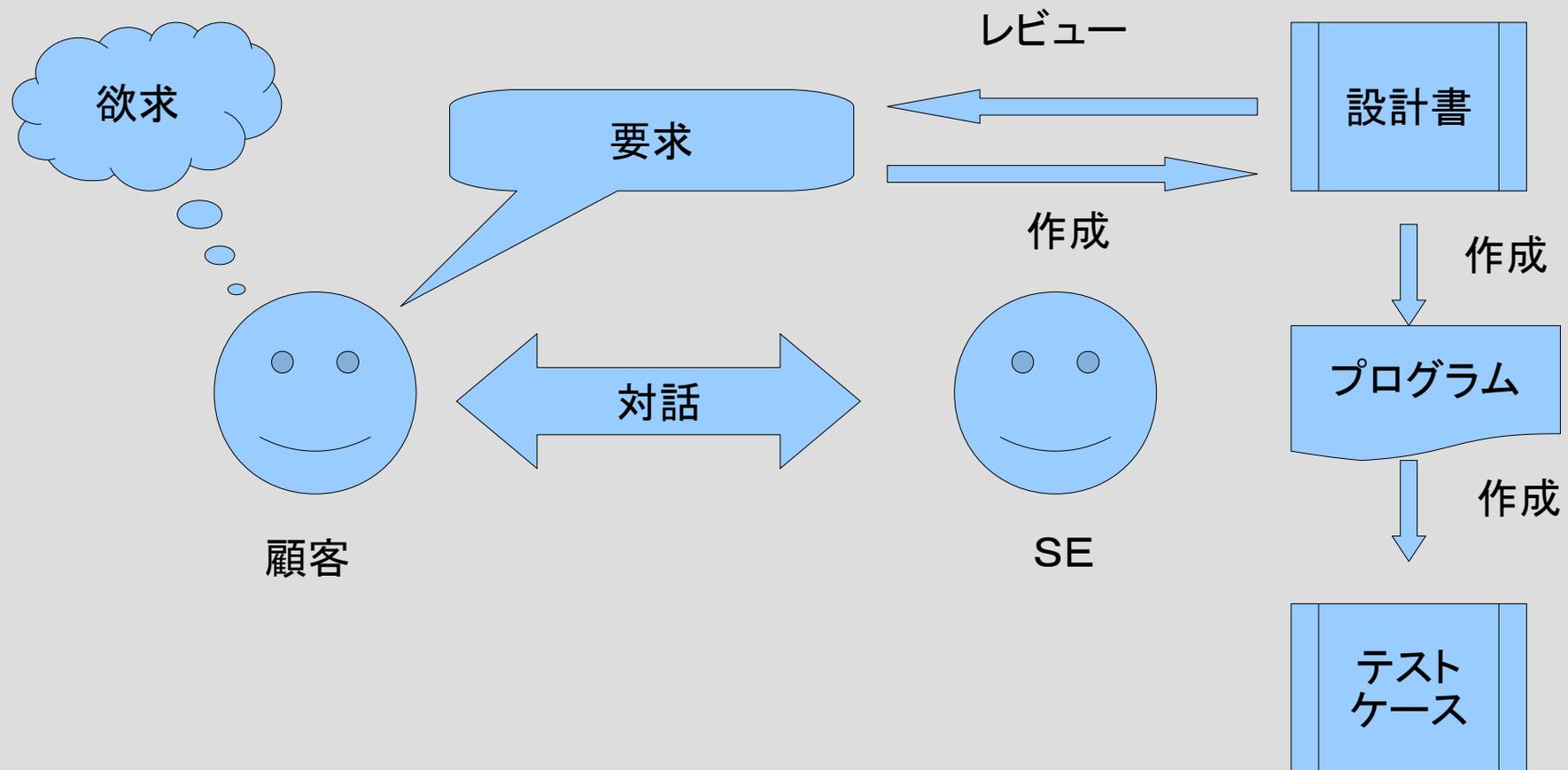
# 『包括的なドキュメントよりも 動くソフトウェアを』

設計書（シナリオ）の  
具体的な説明、  
完全性・無矛盾性の主張の根拠と  
レビューの議事録として、  
テストケースを併せて提供する

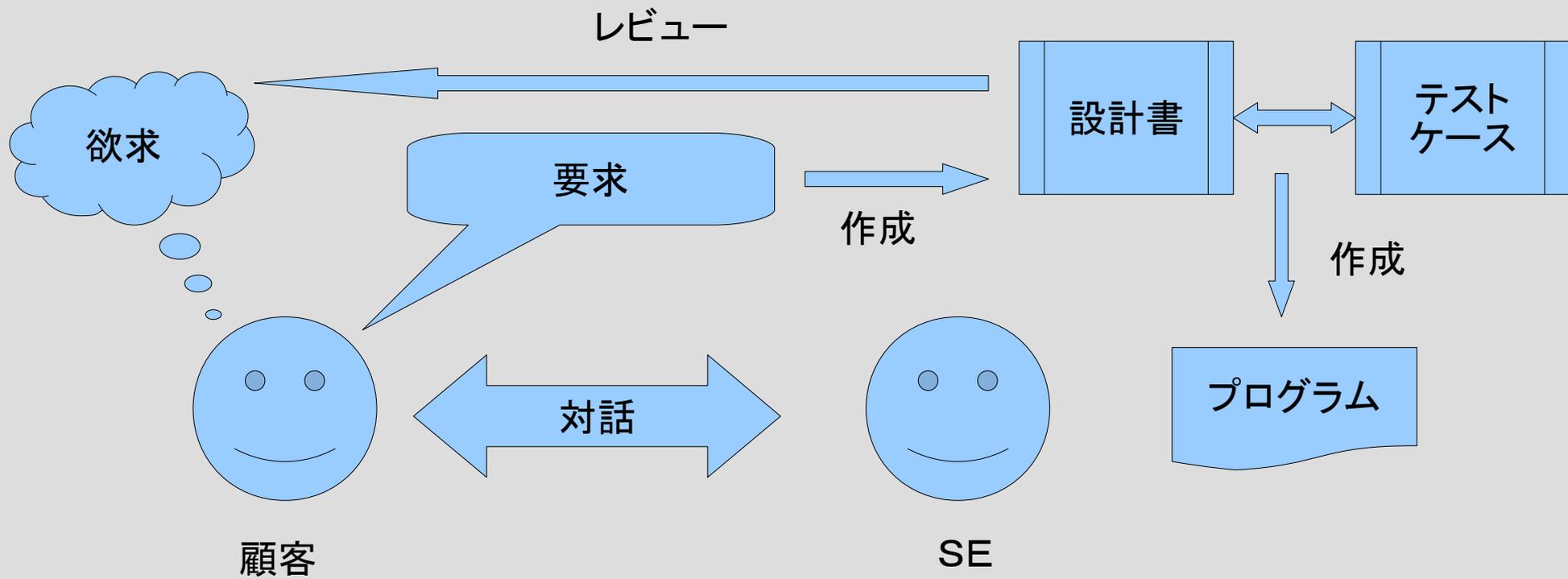
# 『契約よりも顧客との協調を』

要求を満たすのではなく、欲求を満たす  
欲求を満たすために語り合う

# これまで



# これから



# 『プロセスやツールよりも 個人との相互作用を』

意味論として徹底的に議論する

教養として哲学が必要だ

アジャイルに振る舞うだけではなく、  
アジャイルに振る舞うことによって  
得られるものについて、  
どうやれば得られるかを考え、議論する

# 知働化研究会

知働化  
研究  
会

- どんなプロジェクトでもアジャイルであるべきだが、アジャイルだとかアジャイルではないとか、そんな垣根を越えて、システム開発のあるべき姿について議論し合う
- テーマは各自持ち寄り

<http://www.exekt-lab.org/>

ご清聴ありがとうございました

批判お待ちしております  
TEF-ML へ