

# 派生開発におけるアジャイルテストの改善

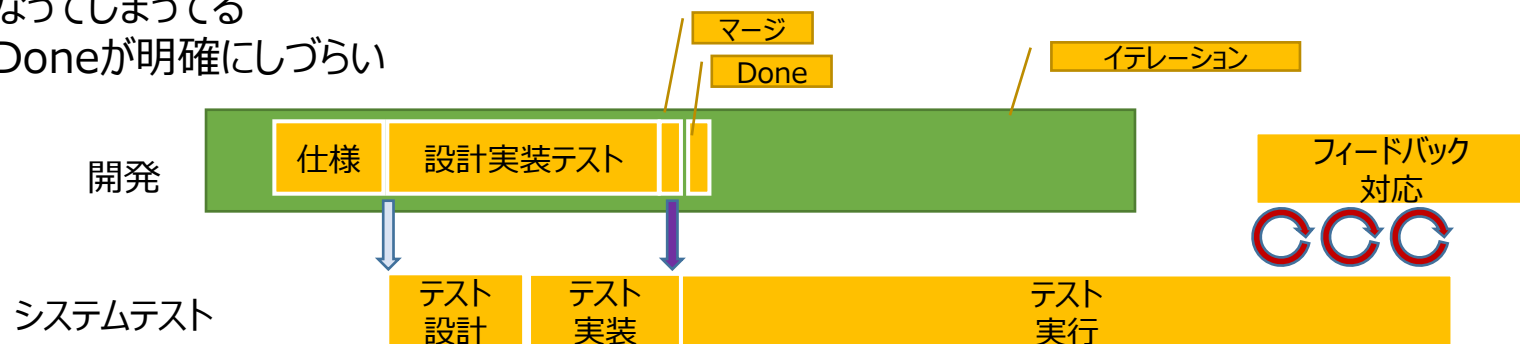
## これまでの研究の経緯

- XDDPとAgileの融合。2014年に派生開発推進協議会の中でAgileに興味を持つメンバーが集まり立ち上げ。以下精力的に活動してきた。
  - 2015年：2つのAgileXDDPの提案
  - 2016年：XDDPを活用してアジャイル派生開発に挑む(ET2016)
  - 2017年：スクラムX (Ver 1.0)の提案
  - 2018年：スクラムX (Ver 2.0)の発表
- これまでの活動概要
  - XDDPにAgileプラクティスを取り入れることでXDDPを改善
  - AgileにXDDPの手法を取り入れることでAgileを改善
  - モデルベース開発 + XDDP + Agile

**今年はアジャイル開発のテストに注目！**

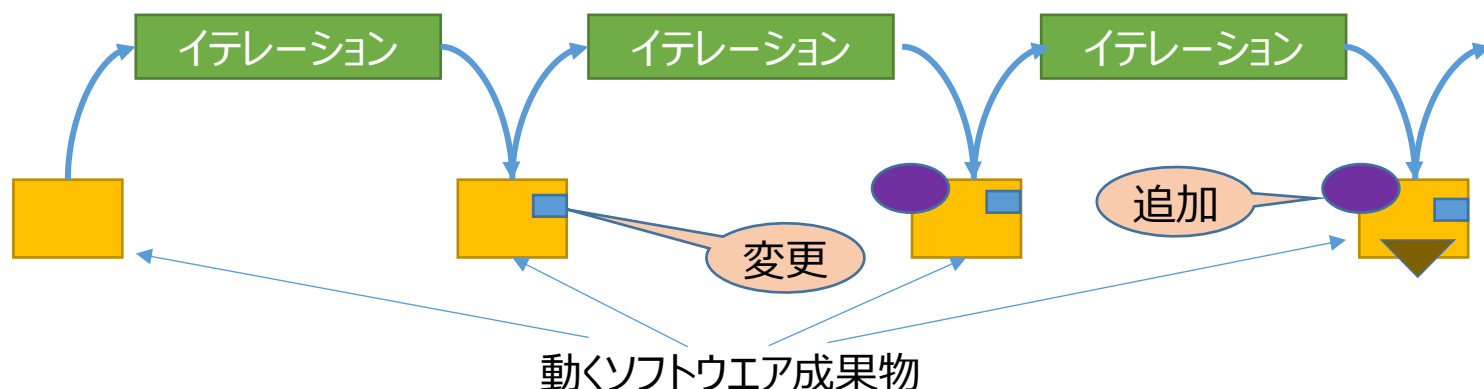
## アジャイル開発におけるテストの問題

- ① テストのフィードバックが遅れる
- ② テスト設計の材料がないため、デグレのリスクがわかりにくいので、決められた仕様を確認するテストになってしまってる
- ③ Doneが明確にしづらい



## アジャイル開発のテスト問題を解決する

気づき：そもそもアジャイル開発は派生開発になっている

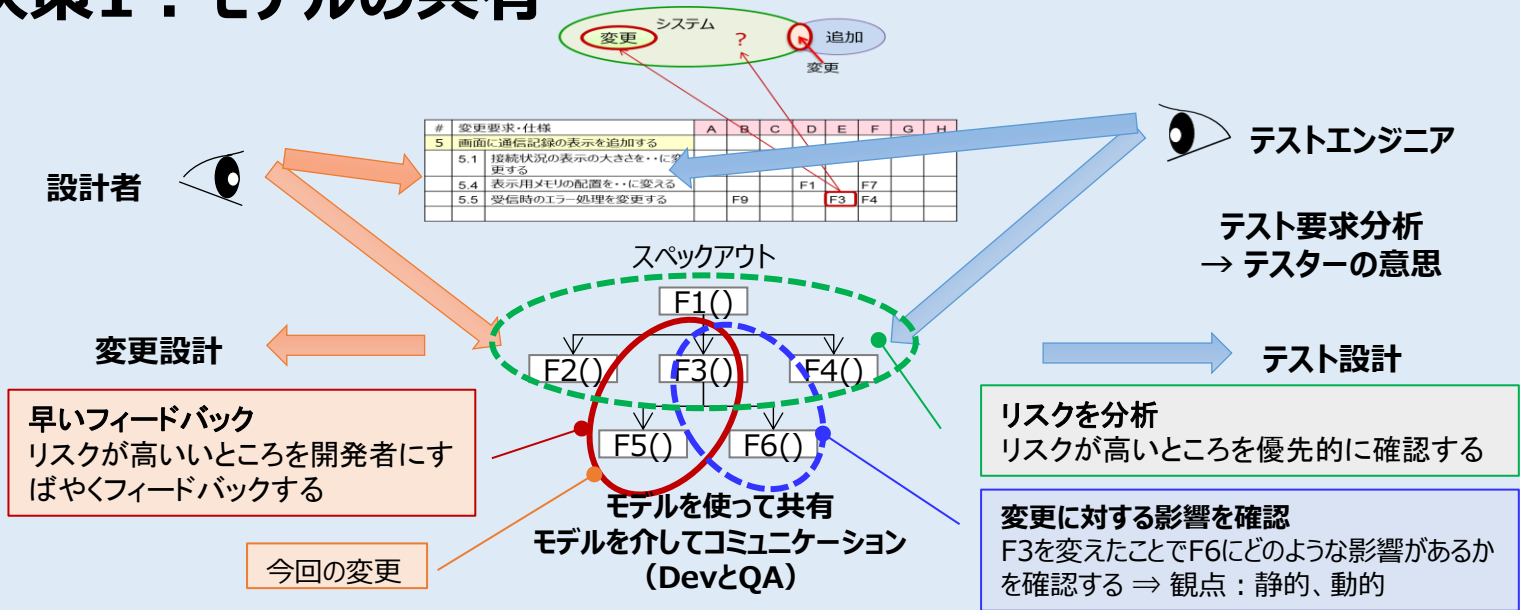


- アジャイル開発においても派生開発の問題が起こっている
- これまでの活動から、次の解決策を提案する

## 問題に対する解決策

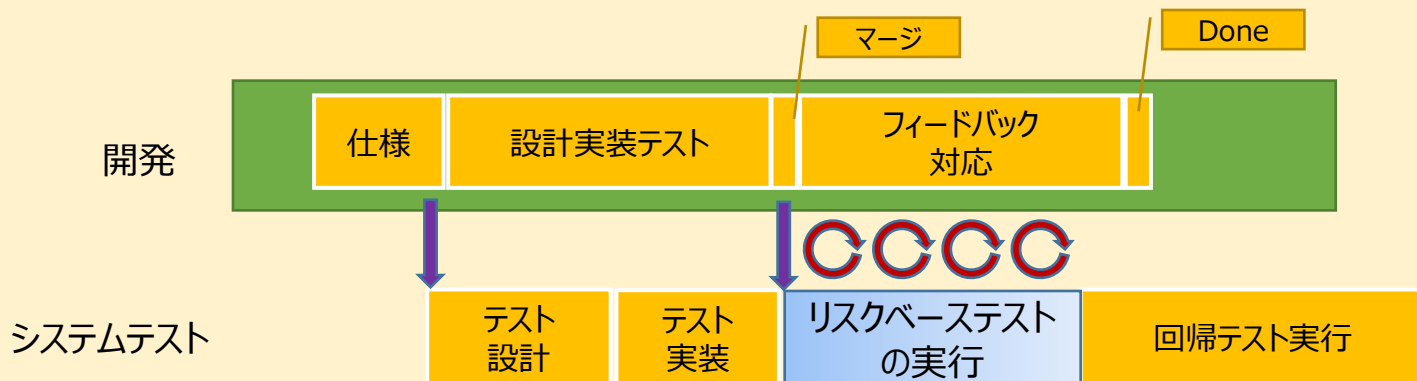
1. 設計で作成したモデルをテストエンジニアが、参照して、変化点を共有しリスクを分析する **(モデル共有)**
2. テストエンジニアは、共有情報によりリスクベーステストを用いて、テストの早いフィードバックを設計に返す **(リスクベーステスト)**
3. アジャイルモデルのトレーサビリティマトリクスを作り、回帰テストの戦略を立てる **(アジャイルモデルTM)**

### 解決策1：モデルの共有



### 解決策2：リスクベーステスト

高いリスクのアイテムからテストを実行することで、設計者が一番気になっているフィードバックを素早く返す



### 解決策3：アジャイルモデルTM

	Model1	Model2	Model3	Model4	...	Modeln
PBI 1	← ○	○				
PBI 2		↑	○			
PBI 3	←			○		
...				↑		
PBI n	←			○		○

- PBI nを開発する際、Model 2とModel4を変更し、Model nも新たに変更した。すると、Model 2にかかわるPBI 1のユースケースにおける振る舞いに影響を与えているかもしれない
- 同様にModel4の変更でPBI 3のユースケースにもインパクトがあるかもしれない
- これにより、PBI 1およびPBI 3のユースケースを回帰テストにくわえるモチベーションになる