

アジャイル開発に対して、リスクベースドテストを実施するケーススタディ

所属 株式会社クレスコ

古谷恒平

k-furutani@cresco.co.jp

開発における問題点

アジャイル開発等の短納期開発・反復的な少人数チームでの開発の中で「定められた期間の中で」「効率的に品質を向上させる」テストを実施する必要性が高まっているが、どの様な方法で効率的に実施する事で開発の負担にならず効率的なテストが行えるかわかっていない。

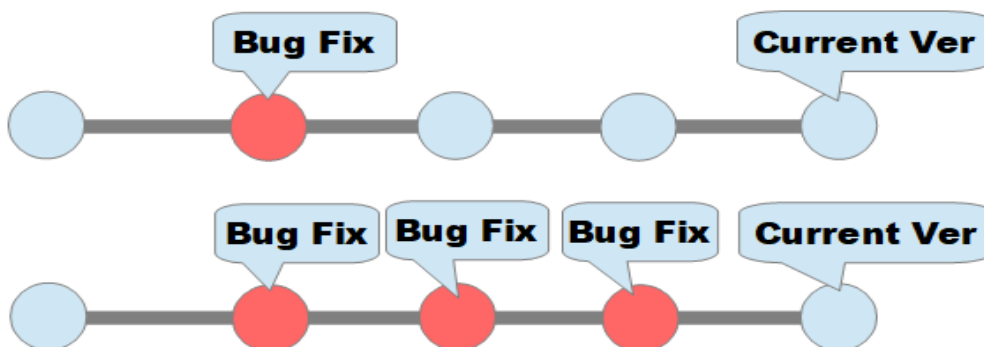
手法・ツールの適用による解決

少人数チームでの開発では従来のリスクベースドテストを運用すると、プロジェクト負担が大きかったため、軽量な3分類と、ソースコード履歴から追えるファイル単位での過去のバグ修正コミット回数の2要素だけを用いて、精度を維持した効率の良いテスト計画が行えることを確認した。

提案手法

以下のルールでテストの実施優先度・実施可否を判断する

1. 修正したファイルの過去のバグコミット数を確認し、バグ数大・中・小に分類する
2. 修正した要件の重要度を、大・中・小に分類する
3. 1をリスク発生確率、2をリスク影響度と判定し、右下のようなマトリクスにマッピングする
4. マトリクスの各マス毎に優先順位の数字を付け、各スプリントの工数が足る範囲でテストを実施する（例えば、下の図のように①～⑥の順番で数字を付け、工数との相談で①～③まで実施する、等にする）



		リスク発生確率		
		小	中	大
リスク影響度	小	—	—	④
	中	—	⑥	③
	大	⑤	②	①

評価

実際の小規模なプロジェクトデータを基に、リスクの判定結果が正しく出るかどうかを判定した結果

- ・10%のファイルのテストだけで全体の25%のバグ抽出を確認。
- ・通常のリスクベースドテストのリスク算出手法と比較しても精度が劣らない(今回の結果であれば上回る)事を確認。

	今回の手法で判断したリスク発生分類		
	リスク小	リスク中	リスク大
実際のバグ発生率	16%(208/1295)	36%(36/100)	58%(35/60)

今後

この手法が既存のリスクベースドテストのリスク算出手法と同程度の精度を出せることを確認した。今後は、

- ・ファイルの修正実績が溜まっていない開発初期においても通用する軽量な判断基準の作成
- ・プロジェクトの規模が拡大した場合にも活用しやすい判断基準の拡張
- ・今回の手法以外の軽量な判断基準の提案等を検討していく