

32SQiP 研究会 特別講義 レポート

作成日： 2016年12月19日

書記氏名： 大島 修

日時	2016年12月16日(金) 10:00 ~ 12:00
会場	(一財)日本科学技術連盟・東高円寺ビル 2階講堂
テーマ	アジャイル開発の成功を握る、テスト・品質保証担当と開発チームの考え方
講師名・所属	永田 敦 氏 (ソニー株式会社)
司会者	中谷 一樹 氏 (TIS 株式会社)
アジェンダ	<ol style="list-style-type: none">1. アジャイル開発の成功を握る、テスト・品質保証担当と開発チームの考え方2. アジャイル開発の実態3. DevQA4. DevQA の改善事例5. アジャイル開発のメトリクス6. ポイント7. 課題
アブストラクト	<p>『アジャイル開発の成功を握る、テスト・品質保証担当と開発チームの考え方』と題して、アジャイル開発の現場で起きている問題点、QA が現場に入り込む際の障壁などについて解説を頂いた上で、Developer (開発) と QA の協働の有り方とその効果について、実践から得られたノウハウや、成功事例を紹介頂きながら、御教示頂きました。</p> <p>IOT 時代に突入し、要件が決めきれない開発が増えていく中、アジャイル開発への取り組みは、今後ますます注目されていくと思います。永田さんは、いち早くからアジャイル開発の現場に QA の立場として寄り添い、品質改善を推進して来られました。開発と QA の間の見えない壁から生じる様々な課題に向き合い、苦労や工夫を重ねて来られた経験やノウハウ、そして DevQA (開発と QA の協働) を広めたいという熱い想いが詰まった実践的で有意義なご講演でした。また、アジャイルの本質を改めて認識させられることもあり、多くの気付きを与えて頂いたご講演でした。</p> <p>ありがとうございました。</p>

第7回の特別講演では、『アジャイル開発の成功を握る、テスト・品質保証担当と開発チームの考え方』と題して、永田さんから御講演を頂きました。

永田さんは、品質保証部にご所属で、主にアジャイル・コーチングを実施し、アジャイル開発の品質改善や、アジャイル RCA によるソフトウェア障害の要因分析と対策などを実践されておられます。また、国内・国外の各種ソフトウェア品質会議等で、アジャイル開発と品質保証の取り組みについてご発表されたり、世界で唯一の“アジャイル・インスペクション・マエストロ”の称号をお持ちであることから、ソフトウェア品質業界では、「アジャイルおじさん」、「世界のナガアツ」とも呼ばれています。

今回は、「DevQA (開発と QA の協働)」について御講演頂きました。開発と QA の信頼関係の構築、そして、その信頼関係の中で互いにフィードバックを得ることで、素晴らしい効果を生み出すことを御教示頂きました。ありがとうございました。

1. アジャイル開発の成功を握る、テスト・品質保証担当と開発チームの考え方

- ◆2011年から取り組んでいるアジャイルテストングやQAの取り組み、考え方の紹介。
- ◆DevQA(でぶきゅーえー)を流行らせて、開発とQAがコラボして品質が高いものを提供したい。
- ◆「品質を良くすることができなければ、アジャイル開発は成功しない」
→テストング、QAが支えていかなければいけない。

2. アジャイル開発の実態

- ◆スクラムの手法についての詳細な説明は、本日は割愛する。(聴講者の中で「スクラムを聞いたことがない」という方はいないため)
スクラムを中心とした話にはなるが、スクラム=アジャイルではないということは理解して頂きたい。
- ◆用語の定義に書かれていることは、本当に実現可能なのか?
プロダクトバックログ：プロダクトに必要なものがすべて上げてあるリスト
→最初に必要なものをすべて上げることができるか？ →できない！
スプリントバックログ：プロダクトバックログの項目を完成させるために必要な作業の計画
→“完璧な”必要な作業の計画を立てられるか？ →立てられない！
※初めから、プロダクトに必要なものをすべて上げること、完璧な作業計画を立てることは、不可能である。
スプリント計画を行う理由、デイリーミーティングを行う理由は、そこにある。
- ◆デイリーミーティングは、計画通りに行かないから実施する。
「昨日実施したこと、今日実施することの報告」そんなことをするためだけにやるのではない。
リスクや問題を明らかにして共有し改善していく。短い一定周期で改善活動を回すことが重要。
→自ら現場が改善活動を回していく。これがアジャイルである。
- ◆要求のクリープは、PO (プロダクトオーナー) と SM (スクラムマスター) が同一人物の時に起きやすい。それは、チームを健康に保とうとする SM の意識よりも、売上向上のために要求を受け入れようとする PO の意識の方が勝ってしまうためである。
→PO と SM は別々にしなくてはいいけないが、実際は同一人物のケースが多い。
- ◆ストーリーポイントとは、スプリントバックログを見積る際の単位。プランニングポーカーと呼ばれているが、必要な作業の大きさを相対値で表す。例えば、ある入力画面の構築を5としたら、編集画面の構築は3くらい、というように相対的な大きさを感覚で出す。
→工数(時間)で見積る方が正確のように思うかもしれないが、個人差が出るし、プレッシャーにも弱い。
- ◆Velocity とは、イテレーションあたり、開発チームが Done したスプリントバックログのストーリーポイントの合計、すなわち、チームのイテレーション当たりのパフォーマンスである。
→Velocity は増えていくものだと考える人がいるが、容易に増えるものではない。改善活動以外では増えない。
- ◆QA が、早期からのシステムテストの実施に介入する。
→システムテストをイテレーションでちゃんとやる。パフォーマンス問題が出荷前に分かったのでは遅すぎる。
- ◆QA が、設計フェーズに飛び込み、早いタイミングで評価しよう。
→しかし、本来の設計をやりたいのに、QA が設計に入ってくると、あれを出せ、これを測れ、あれを直せと言われて、余計な負荷がかかる。そのため、設計リーダーのガードは固い。

では、どうすれば良いか？ 暗黙知で会話しているチームの輪に、どうやったら QA が入り込めるのか？
そこで、登場するのが DevQA の考え方である。

3. DevQA

◆DevQA の形成。

→QA は設計をサポートし、チームとして働く。テストや品質測定を行いフィードバックしてあげて品質の見える化を行えるようにする。設計の邪魔をされるのではない、ということに気付いてもらう。

◆アジャイルで、できるだけ短い時間間隔でリリースを行うのは、フィードバックを早く得たいからである。

◆本当の価値は顧客からのフィードバックからしか得られない。紙ベースでは要求が正しく取れないため、顧客からフィードバックをできるだけ早く得たい。本当に顧客がほしい価値をデリバリしたいため、モノを早く出してフィードバックをもらう。それがアジャイル開発の本質である。

◆これからは、ドキュメントに書けない時代である。IoT は、もやもやしている。そんな中でも、新しいアイデア=価値を速く市場に出したい。しかし、セキュリティ障害は起こせないし、品質は妥協できない。ところが、自分たちだけでなく、他のシステムとの協業、他社との協業が必要なため、仕様は変わるし決まらない。

◆DevQA ループの設計・成立のために重要なのは、何をフィードバックするのか、そのために必要な情報は何かを説明すること。相手が合意すれば、時間を割いて、その情報、成果物を用意してくれるはずである。

◆テストでは、何をいつどのようにテストするのかを説明し、そのテストをするためにどんな情報や成果物が必要かを説明する。すなわち、テスト計画である。

→テスト計画が立てられないと、DevQA ループは回らない。

◆QA がどうやってテストベースを手に入れるかという課題があるが、待っていてはダメ。

→QA が開発に入り込んでいく。ドメイン知識を学び、スクラムのイベントに必ず参加する。

4. DevQA の改善事例

◆SM (スクラムマスター) が QA の人と一緒に QA を実施した事例。SM が主導でユーザストーリー単位のテスト設計を実施し、そのテスト設計に QA が主導で評価観点を肉付けした。テスト設計で足りないインプットが見つかり QA から PO (プロダクトオーナー) にフィードバックするよう SM が促す。

→フィードバックループによって、結果的に PO は QA と開発からフィードバックを受けるようになった。

◆”仕様通り”という理由で開発から返されるバグ報告の量が 2 割から 1 割以下に減少した。

◆開発が、QA のテスト設計をレビューするようになった。

→テスト項目が多いが、無駄なテストはないだろうか？

◆開発と QA が同じ場所で作業していると、バグの修正内容をチケット更新だけでなく口頭で伝えられるので、バグ発生時の動作が把握し易い、というメリットがある。

◆設計と QA の信頼関係が重要。暗黙知だけでなく、形式知とセットで共有する。

5. アジャイル開発のメトリクス

◆ウォーターフォール開発でも進捗会議をしているが、Velocity や計画と実績の差異を細かい周期では見ていない。そのため、QA-in 判定間際で火を噴く。

◆Velocity は上げようとしてもなかなか上がらない。人が減ると下がるが、人を増やしても下がる。

Velocity を上げるためには、振り返りとカイゼンが必要である。

◆Velocity に対するアンチパターン。

→長時間残業、休日出勤はダメ。無知見開発は派生開発の特に悪いパターン。

◆大事なものは、チームが健康であること。健康は品質に関わるため、ストレスに敏感になること。

◆バックログの Done の定義には、必ず QA も関わること。

◆QA-in 以降のバグの分布を見ると、本事例では、基本機能のバグは、QA-in 前の評価で取られている、または、始めから入り込んでいない。

◆QA が作成するテストケースが、製品の詳細な振る舞いの仕様書となり、開発部隊はテストケースを開発のゴール (神様) と見なすようになった。すなわち、開発が QA の価値観を信頼するようになった。

6. ポイント

◆一晩でできたことではない、最初は不完全なもの。QA を巻き込み、フィードバック、振り返りを繰り返して少しずつ変えていった結果である。

7. 課題

- ・SM (スクラムマスター) への依存度が高く、自立できていない。
- ・Velocity がなかなか上がらない。人がやめるなどコントロールできない面がある。しかしマネージャーは無闇に上げたがる。
- ・人の育成が重要。自立的に動ける人材が必要。但し、時間が掛かる。
- ・このようなアジャイルチームになるまで2年かかった。アジャイルの普及活動も含め時間が掛かる。

<最後に>

- ・「品質とは誰かにとっての価値である。」ジェラルド・ワインバーグ
- ・ATDD(Acceptance test-driven development)
顧客の真に求める価値を理解し、表現、(測定)できなければ保証はできない。
→ビジネスモデル、要求開発の世界へ

<質疑応答>

- ①ATDDにおいて、詳細の振る舞いはQA が作るとのことだが、設計が作るのではないのか？
→設計では例外処理が弱い。設計ではリストや条件、タイミングの記載はあるが、QA でポイントを抑えた振る舞いのテストケースが出来上がるはず。
- ②スクラムの構成は？
→大規模開発のアジャイルの適用は難しい面がある。20人以上になってくると、コミュニケーション面で上手く回らないため、バックログを細分化してグループ化する。但し、バックログ間で齟齬が起きないように対策が必要。
- ③実践アジャイルテストでペアテストがあるが、開発者が開発したら隣ですぐにQA が機能テストをしているイメージで良いか？
→大切なのは、何をテストするか？どんな品質にするか？をプランニングした上でフィードバックしないと、意味がない。QA 観点でどういう観点で品質を見ていくか？を狙っている。

以上