

# アジャイル開発における品質保証

## ～セルフチェックの活用～

### Quality Assurance in Agile Development – Utilization of self-check –

主査 : 阪本 太志 東芝デジタルメディアエンジニアリング (株)  
副主査 : 三浦 邦彦 矢崎総業 (株)  
リーダー : 柴崎 勝文 パナソニック (株)  
研究員 : 山内 誠 ビジネスキューブ・アンド・パートナーズ (株)  
後藤 健 T I S (株)  
松本 恭嘉 富士通セミコンダクター (株)

#### 研究概要

近年、ソフトウェア開発においてアジャイル開発手法が広まりつつある。従来からのウォーターフォール型開発に適したソフトウェア品質保証の体制や手順を、そのまま続けてアジャイル開発での品質保証は充分確保できるのだろうか。品質の見える化をする必要はないか、具体的にどうすれば良いのか、日ごろ研究員が現場で直面する課題について取り組むことにした。また、アジャイル開発の一番の効果は品質を良くすることではないか、という仮説をたてて研究に取り組んだ。

アジャイル開発にむけた品質保証の方法を検討し、提案にまとめることにした。われわれの提案は、レファレンスとしてソフトウェア CMMI v1.3 に謳われたアジャイル開発に関するプロセスエリアを選んで、個々に対比することで検証を実施することにした。その結果もあわせて報告する。

**Abstract:** In recent years, many agile development techniques are adopted in software development. Are the organization and the procedure of software quality assurance which are carried out in a waterfall type development process applicable by agile development? Are you assurance the quality of the software? Isn't visualization of software quality necessity?

A software-quality-assurance person in charge considers what you should do concretely.

We decided for a researcher to take up and study the subject which it has at the everyday.

We built up the hypothesis.

Is the effect which was excellent in agile development improving quality of software more?

Our team began research.

And we examined the method of the guarantee of quality suitable for agile type development.

We propose collectively the knowledge acquired from examination.

Our proposal chose software CMMI v1.3 as a reference. We chose some process area related to the agile development indicated to CMMI. We verified the method of inspecting in each process area.

This paper also reports the result.

#### 1. はじめに

##### 1.1 研究テーマ選定の背景と目標

本年度の第1回例会において第1分科会のなかで、「振り返りプロセス」に取り組みたいメンバーと、アジャイル開発<sup>[1]</sup>に適用するソフトウェア品質保証の活動について悩むメンバーが混在した。

分科会の中でわれわれAチームは、すでに開発現場がアジャイルな開発手法をいろいろ導入している実態に「ソフトウェア開発現場の変化」を感じて、不安とあせりで当惑している自分たちを改めて認識した。そこで、これからソフトウェア品質保証（以降、SQA）は何をすべきか、アジャイル開発に対して役立つ品質保証の考え方や方法論を研究することをテーマに据えた。

従来のウォーターフォール型開発における品質保証と、どこが違うのか、アジャイル開発では、どのタイミングで何を抑えれば品質保証ができるのか、という理論と実例を他者に学ぶことを目指した。

## 2. 現状の課題、問題点

### 2.1 問題点

携帯電話などのアンドロイド OS を使用する製品をアジャイル開発手法で開発している開発現場から品質保証活動に対する問題があがっている。この開発現場とは、顧客（メーカー）から製品開発を委託されたプロジェクトを指している。このプロジェクトの品質保証活動は、プロジェクト側が顧客へ納品する製品の品質保証を目的として行っており、開発初期から詳細にチェック項目が記載されたチェックリストを使用している。プロジェクトの詳細を以下に示す。

#### 開発プロジェクト

携帯電話などのアンドロイド OS を使用する製品開発

#### 開発手法

アジャイル開発

#### 品質保証活動の目的

顧客に対する製品の品質保証（ウォーターフォール型と同じ品質保証活動）

#### 品質保証活動の実施内容

開発の各工程完了時に、SQA メンバーによる品質保証の監査チェックリストを使用して開発メンバーへ対ヒアリングを実施

#### 品質保証活動に対する問題

- (1) チェック項目の内容が詳細のため 1 回の品質保証チェックにかかる時間がかかり過ぎて、開発メンバーの負荷が多い
- (2) 1 つのプロジェクトに対する品質保証活動の回数が多すぎるため、SQA メンバーの負荷が多い
- (3) 製品の品質保証活動を開発初期から行いたいが開発後期に一斉に行われる。

われわれは、このプロジェクトの問題として、ソフトウェア品質保証活動がウォーターフォール型開発とアジャイル開発といった異なる開発手法に対して、同じ品質保証活動を適用していることが原因ではないかと考えた。2.2 節では本研究のアジャイル開発について述べ、2.3 節では、両者の開発手法の違いを確認する。

### 2.2 本研究のアジャイル開発について

本研究のアジャイル開発は、開発初期から開発末期まで短期間（1 か月未満）で製品開発を行い、顧客への納品を繰り返している反復開発を意味している。

### 2.3 ウォーターフォール型開発とアジャイル開発の製品開発ライフサイクルの違い

ウォーターフォール型のソフトウェア開発では、高品質に、且つ生産性高く開発するために開発初期に要求の確立を図る。仕様は形式化してドキュメントにまとめる。これらの行為を開発モデルとして定義し、それに沿って開発を進めることが求められている。

しかし、そもそも要求が刻々と変化している場面では、早期に要求を固定することは製品やサービスの販売リスクを拡大することもある。また、開発中に技術リスクが大きく、実際に作って見ないとそのリスクを回避できない場合もある。

ウォーターフォール型開発を採用する機器の組込みソフトウェア開発などでは、試作から量産まで大きなV字モデルの製品開発ライフサイクル（図 1）が 2, 3 回繰り返され、1 つのV字モデルが 3 か月から半年などの長期間で開発が行われることが多い。一方で、従来のウォーターフォール型ではない、つまり非ウォーターフォール型開発として、代表的にはアジャイル開発が必要とされてきている。

本論文では、アジャイル開発として、いろいろなプラクティスが世の中に提唱され、実施されているが、それぞれの手法や手順にとらわれることなく、すべてを包含してアジャイル開発と呼ぶことにする。

アジャイル開発の製品開発ライフサイクル（図 2）は試作から量産までイテレーション（反復開発）が繰り返され、小さなV字モデルが複数存在する。さらに、1 つのV字モデルは 1 週間～3 週間などといった短期間で行われることが多い。このことから両者には 1 つのV字モデルの期間に大きな違いがあることがわかる。

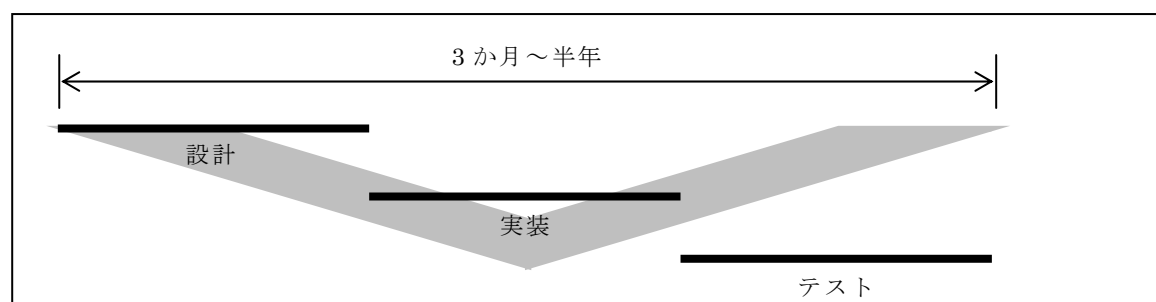


図 1. ウォーターフォール型開発の製品開発ライフサイクル

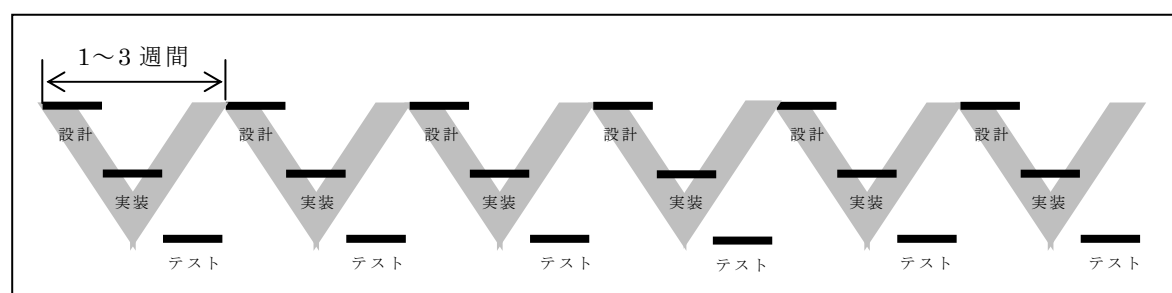


図 2. アジャイル開発の製品開発ライフサイクル

### 2.4 ウォーターフォール型開発の品質保証活動

2.2 節で述べたようにウォーターフォール型開発の製品開発ライフサイクルが長く、一般的には図 3 のように各工程の移行時に品質保証活動が行われており、開発メンバーと SQA メンバーの負荷は全体の工数に対して大きくならない傾向がある。

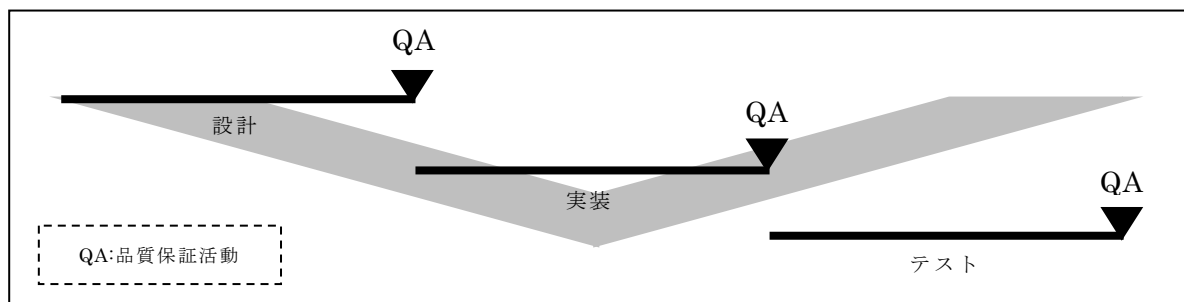


図 3. ウォーターフォール型開発の品質保証活動

## 2.5 アジャイル開発へのウォーターフォール型開発の品質保証活動の適用

2.2 節で述べた違いがあるウォーターフォール型開発の品質保証活動をアジャイル開発に当てはめるとどのようなことが起こるか、実際に品質保証活動を行っている複数の組織からヒアリングを行い、2つの型が存在することがわかった。

### ①品質保証活動増加型

アジャイル開発の小さなV字に対して、各工程移行時に品質保証を行う。各工程は3日～1週間などと短期のために常に品質保証活動を実施している状態が続く。

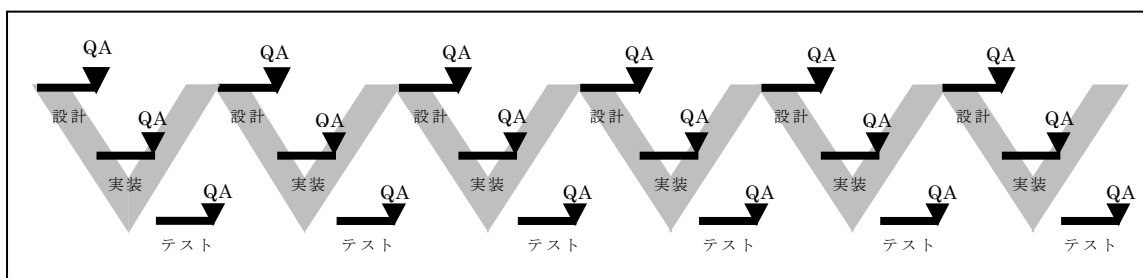


図 4. 品質保証活動増加型の開発ライフサイクル

### ②品質保証活動の最終実施型

アジャイル開発の場合、イテレーション（設計～テスト）が行われているため、製品レベルでの各工程の終了は製品開発最終段階になることが多く、工程移行時に品質保証を行うと開発後期のみ品質保証活動が行われている。

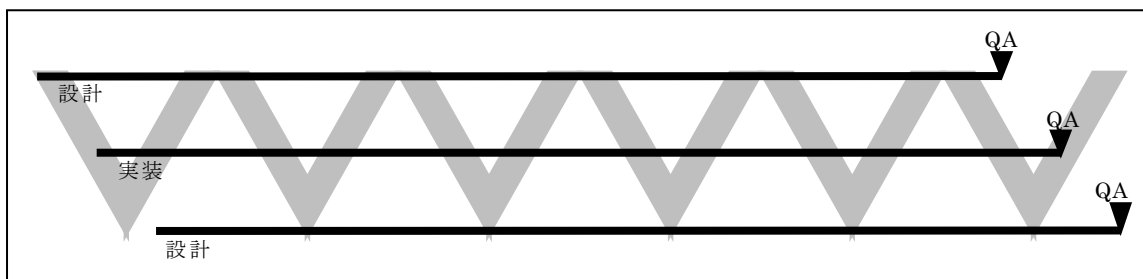


図 5. 品質保証活動最終実施型の開発ライフサイクル

これらの2つの型は、2.1 節の問題と一致していることがわかる。

- ・品質保証活動増加型 ⇒ 工数の増加[問題 (1)、(2)]
- ・品質保証活動の最終実施型 ⇒ 初期の品質保証の未実施[問題(3)]

## 2.6 アジャイル開発に対する品質保証活動の課題

2.4 節でのベタ型と問題を考慮した結果、アジャイル開発へのウォーターフォール型開発の品質保証活動を適用するには 2 つの課題があることがわかった。3 章では、以下の課題に対する改善アプローチを検討する。

### ①品質保証活動に対する過大な工数の抑制

短期の工程毎の SQA による、開発メンバーと SQA メンバーに対する工数を抑える。

### ②開発初期からの品質保証活動の実施

製品開発の品質保証活動を開発初期の時点から行う。

## 3. 新たなアプローチ

我々は 2 章であげられた 2 つの課題に対して、アジャイル開発手法で開発を行うプロジェクトが顧客の品質に対する要求を満足させるための品質保証活動には、セルフチェックと定期的な品質保証活動の組み合わせが有効ではないかと提言する。これはリーンソフトウェア開発で行われている製品リリースタイミングを参考にした<sup>[2]</sup>。アジャイル開発の多くはイテレーション毎に納品するが、リーンソフトウェア開発の製品ライフサイクル(図 6) では定期的な周期で納品を実施し、その間で実施されたイテレーションに対して納品を行う開発である。

新たなアプローチ(図 7)では、このリーンソフトウェア開発における納品を品質保証活動の定期的な置き換え、且つ、イテレーションにおける品質保証活動では開発メンバーによるセルフチェックを実施する。このことにより 2 つの課題に対して改善できると考えられる。

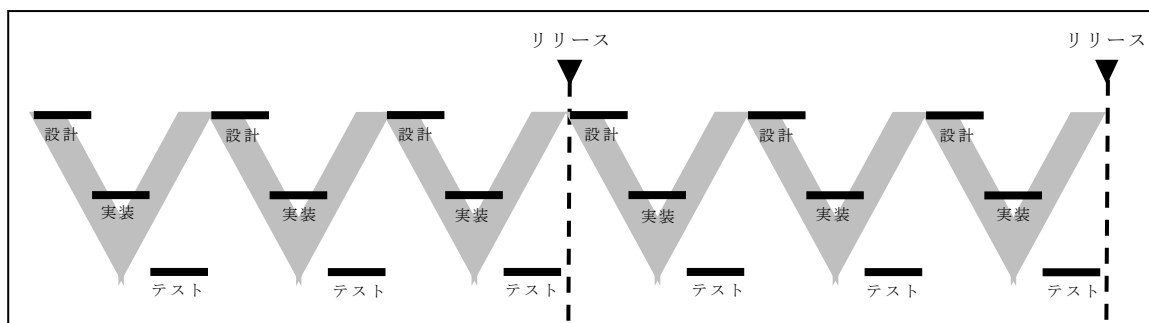


図 6. リーンソフトウェア開発の製品開発ライフサイクル

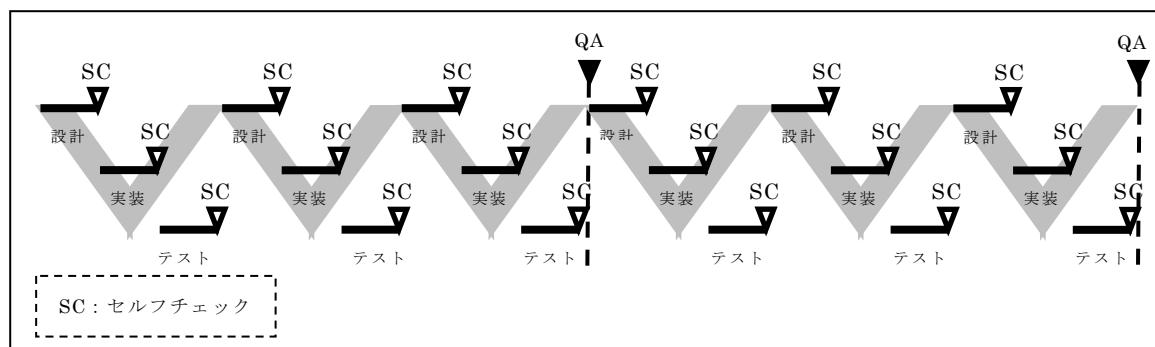


図 7. 新たなアプローチ

#### ① 品質保証活動に対する過大な工数の抑制への効果

定期的な品質保証活動の実施により、今までの品質保証活動と開発メンバーの短期間の品質保証活動を削減できる。また、セルフチェックにより必要最低限の工数でイテレーションに対する品質保証活動ができる。

#### ②開発初期からの品質保証活動の実施への効果

定期的の実施することにより開発初期から後期まで一定の品質保証を実施することができる。また、セルフチェックを行うことにより早期の問題発見を行うことができる。

本研究では新たなアプローチとして、アジャイル開発の品質保証における、セルフチェックを活用した品質保証活動を「QAAD42」(Quality Assurance in Agile Development by 42)と命名し検証を実施し、その効果を確認した。

### 4. 検証

#### 4.1 検証方法

本研究で考案した QAAD42 の有効性を検証するため、実際に適用している品質保証項目からセルフチェック項目を抽出し、開発者によるセルフチェックの試行を行った。試行に使用したセルフチェック項目は以下を満足するように作成した。

- ・セルフチェックの抽出元である品質保証項目は、CMMI v1.3 のプラクティスを網羅している
- ・アジャイル開発の品質保証項目は、CMMI v1.3 のアジャイル開発の留意事項を反映している

試行は以下の2点について確認を行った。

- ・SQA 監査工数の抑制
- ・問題点の早期検出

#### 4.2 検証結果

##### 4.2.1 SQA 監査工数の抑制

- ・開発者の工数抑制  
従来、SQA 監査のためにプロセス、生産物のエビデンスを開発リポジトリから探すなど SQA 監査のための工数をセルフチェックにしたことにより抑制できた。
- ・SQA の工数抑制  
従来、開発ライフサイクル全般にわたり全プロセスと全ての作業成果物を SQA が監査していたが、イテレーション内を開発者自身によるセルフチェックとしたため、セルフチェック部分の SQA 監査工数が抑制できた。

以上の結果から、QAAD42 は SQA 監査工数の抑制に有効であると考えられる。

##### 4.2.2 問題の早期検出

問題の早期検出について CMMI v1.3 のプロセスエリア PPQA のアジャイル開発における CMMI の解釈に照らし合わせて、検証結果を以下に示す。

##### 1) 客観的な評価はどのように実施するべきか

開発ライフサイクル全体を SQA による定期的な SQA 監査を実施することで、客観的な

評価が実施されていることを確認した。

2) どのプロセスおよび作業成果物を評価するか

- ・開発ライフサイクル全体  
開発ライフサイクル全体を上流と下流の2回に分けた定期的な SQA 監査を実施することで全プロセスとそのプロセスから作成される作業成果物を評価できることを確認した。
- ・イテレーション内  
イテレーション内のプロセスとそのプロセスから作成される作業成果物は、セルフチェックによる開発者自身の監査で評価できることを確認した。

3) チームのリズムにどのようにフィードバックするか

- ・開発ライフサイクル全体  
定期的な SQA 監査の結果は、開発サイクルの上流と下流の最後のイテレーションのふりかえりにフィードバックできることを確認した。
- ・イテレーション内  
セルフチェックの結果は、開発者自身がセルフチェックを実施するので、開発チームのリズムにあわせ、毎日の朝会やイテレーション内のふりかえりに随時、フィードバックできることを確認した。

以上の結果から、QAAD42 は開発ライフサイクル全体での SQA 監査とイテレーション内でのセルフチェックを組み合わせた客観的な評価として、問題の早期検出に有効であると考えられる。

#### 4.3 検証結果の考察

検証結果より QAAD42 は、品質保証体系の検査部門の役割であるプロセス監査活動において、特に反復的な開発を行うアジャイル開発において、有効であると考えられる。

副次効果として、開発者はセルフチェックを通じて、自ら判定を行ったことでプロセス品質に対する認識が深まった。

なお、本手法を適用する際の留意点として、検証結果から以下の項目が挙げられた。

- ・客観性を維持したセルフチェックを行うためには、チェック項目の過不足が発生しないように品質保証部門との連携が必要である
- ・開発者自身による判断誤りを防止するために、判断基準を定めたガイドラインの整備も必要である

#### 5. 今後の課題と取組み

今回検証した某メーカーによる通信機器の開発で1つのV字モデルが約1週間程度の短期間の開発であり、3章で述べたプロセスやプロジェクト品質保証、セルフチェックのタイミングについては1つの事例でしかない。

アジャイル開発の品質プロセスは発展途上であり、多岐にわたる検証を繰り返すことで、まだまだ改善余地があると思われる。

今後は会社規模、開発期間、開発規模、業種等の異なるさまざまな開発を早期段階から「開発チーム」、「会社、組織」などのリズムに合わせ「プロセス」、「成果物」、チェックタイミング、審査方法等について論議を繰り返し、検証を行い、検証結果について有効性の評価および改善を繰り返すことが必要である。

## 6. おわりに

本研究は、開発プロセスは従来から多く採用されているウォーターフォール型開発をベースに、品質保証部門のみによる監査ではなく、開発現場自身がセルフチェックを行うことで相乗効果を狙うアプローチの検証のひとつが実施できた。アジャイル手法は様々なプラクティスを擁しており、ソフトウェア品質保証の手法はこれからも変化していかざるを得ないと考えている。それにより、組み込み系やエンタープライズ系などにおいて実施されるアジャイル開発について、品質保証プロセスの定義を発展させていくことが期待できるものと推察する。

## 7. 謝辞

本研究活動においてご支援、ご協力をいただいた一般財団法人・日本科学技術連盟事務局の方々、第1分科会主査、副主査、研究員の活動をご承認いただいた各研究員所属企業の上司の方、すべての方々に深く御礼申し上げます。

## 8. 参考文献

- [1] 『アジャイルサムライー達人開発者への道』 Jonathan Rasmusson 著、西村直人・角谷信太郎 監訳近藤修平・角掛拓未 訳 オーム社、2011 年
- [2] 『リーン開発の本質 ソフトウェア開発に生かす 7 つの原則』 メアリー・ポッペンディーク、トム・ポッペンディーク 著、平鍋健児 監訳、高嶋優子、天野勝 訳 日経 BP 社、2008 年