

# エンタープライズ・アジャイル研究会 2020年度活動の振り返り

---

2021年5月10日



# 研究会概要

---

## 研究会の目的：

ユーザ企業の視点から、エンタープライズ・システム分野におけるアジャイル開発手法の適用状況の把握と適用に対する課題の議論・検討を行う。

## 開催日程：

原則として奇数月の第4水曜日、15時～18時。

第1回 2020/07/22      第2回 2020/09/16

第3回 2020/11/25      第4回 2021/01/27

第5回 2021/03/17

## 進め方：

各回テーマをあらかじめ決め、そのテーマに沿った事例を1～2社からご紹介頂く。また事前に参加メンバーへアンケートを配布、その結果をもとに議論を行う。

# 参加メンバー

---

アイエックス・ナレッジ(株)

(株)IHIESキューブ

(株)エヌ・ティ・ティ・データ

(株)シーエーシー

J F E システムズ(株)

(株)ジャステック

住友電工情報システム(株)

SOMPOシステムズ(株)

(株)テクノロジックアート

東京ガスiネット(株)

トヨタファイナンス(株)

ニッセイ情報テクノロジー(株)

日本デェイブレイク(株)

(五十音順)

計 13社、20名 (ABC協会含む)

# 検討テーマ・事例紹介

---

## **第1回（7/22）：各社アジャイル取組状況など**

- 各社のアジャイル開発への取り組み状況のご紹介
- 「エンタープライズ・アジャイル開発の品質管理」（株）テクノロジックアート 長瀬様
- 今年度の進め方の議論

## **第2回（9/16）：「アジャイル開発における品質管理（第1回）」**

- 「エヌ・ティ・ティ・データにおけるアジャイル開発での品質管理について」  
(株)エヌ・ティ・ティ・データ 町田様

## **第3回（11/25）：「アジャイル開発における品質管理（第2回）」**

- 「保守開発へのアジャイル開発の適用  
(アジャイルとオフショアのコラボによる保守開発の実践)」  
(株)IHIエスキューブ 千脇様
- 「GQM (Goal Question Metric) のご紹介」（住友電工情報システム(株) 中村様

# 検討テーマ・事例紹介

## 第4回（1/27）：「アジャイル開発における品質管理（第3回）」、 「アジャイル開発で役立つツール」

- 「トヨタファイナンスにおけるアジャイル開発事例（仮）」（トヨタファイナンス(株) 野澤様）

## 第5回（3/17）：「アジャイル開発における品質管理」、 「アジャイル開発のメリットと指標」

- 「SOMP Oシステムズにおけるアジャイル開発への取り組み」  
(SOMP Oシステムズ(株) 別所様)

# エンタープライズ・アジャイルについて

---

## 当研究会で議論の対象としている「エンタープライズ」とは？

- 主に企業内で利用する業務システムを想定
- 基幹系、営業支援、庶務等分野は問わず
- 規模は問わず
- 開発形態は問わず（SIerとして顧客へ提供する場合、社内で内製する場合、など）

## エンタープライズアジャイルに期待する効果

- 本質的には「動くものを早くリリース、フィードバックを受けて改善していき、ニーズに合った仕組みを提供する」したい
- 世間的には「早く、安く」？「品質は犠牲」？

# 各社アジャイル取組状況

---

## 取組状況

- ウォーターフォール開発が中心であるものの、アジャイル開発経験を持つ企業が大半（経験量にはバラツキがある）
- デジタル化の波の中、アジャイル開発の推進教育や社内ルールの整備も進行中

## 課題

- ベンダーとユーザにおけるアジャイル開発に対する温度差、誤認識  
-経営層や従来型にこだわる人・顧客にどうすれば理解してもらえる（意識を変えられる）か
- アジャイル開発人材の育成
- 品質マネジメント（計画、管理、改善）のあり方

など、様々

---

# アジャイル開発における品質管理

# なぜ私たちは、アジャイル開発の品質管理を議論するのか？

---

➤ アジャイル開発とウォーターフォール開発では品質管理の仕方が違うのではないか？

## □ ウォーターフォール開発

開発しているソフトウェアが、本来の開発目的に合致したものであるかの確認は、開発終盤まで確認できない。

⇒各工程の中間成果物（設計ドキュメント 等）で、「できるであろうソフトウェア」の正しさ（品質）を確認にする。

## □ アジャイル開発

短い期間（数週間～1か月程度）で、「動くソフトウェア」を作ることの反復し、最終成果物を作り上げる。

⇒短い期間で作られた、「動くソフトウェア」を直接確認することで、そのソフトウェアの正しさ（品質）を確認する。



**ウォーターフォール開発を主軸としていた組織での、アジャイル開発の品質管理（の仕方）を考える。**

# アジャイル開発推進者・開発者の「品質管理」に関する悩み

---

- 管理者層や品質管理部門（第3者）への説明
  - ✓ ウォータフォール開発と同様の品質管理手法の適用や評価を求められる。
    - 設計ドキュメントなどの中間成果物の品質報告
    - リリース直前には十全なテストの実施
    - 信頼度成長曲線を用いた品質評価
  
- 品質評価のタイミング
  - ✓ いつ、どんな指標を使って品質を評価すればよいかわからない。
  - ✓ バグを見つけたらすぐに修正するため、バグの件数や混入原因などに基づく分析がし難い。
  
- テストの抜け・漏れ防止や負荷の軽減
  - ✓ 開発者の思い込みより、テストケースがテストシナリオごと抜けてしまい、品質課題を開発を発生させ、大きな問題となった。
  - ✓ 短い期間での反復開発であるため、効率的なテストの実行が求められる。

# 当研究会での事例共有と議論 1

---

## ➤ 品質管理の手法

- ✓ レビュー、エラー、テストの分析・評価
- ✓ プログラムのソースコードとしての品質評価（行数や複雑度など）

## ➤ 品質管理指標

- ✓ ウォーターフォール開発と同じ指標を用いている  
（レビュー時間、エラー摘出密度、テスト密度 など）



- ◆ **品質管理の手法や管理指標は、ウォーターフォール開発と変わらない。  
相違点といえば、中間成果物（設計ドキュメント）の品質評価がないこと。**
- ◆ **ソースコードの品質評価は、ウォーターフォール開発でも求められる。**

## 当研究会での事例共有と議論 2

---

- 各社事例での品質を確保する取組み（工夫）
  - ✓ リリース前の「テストスプリント」の設置（ウォーターフォール開発のシステムテストの位置づけ）
  - ✓ 第3者によるテスト実行
  - ✓ ツールを活用した、テスト自動化
  - ✓ 品質保証のパターン集（QA2AQ（Quality Assurance to Agile Quality））の適用
  - ✓ 品質保証部門では品質を評価できない。アジャイル開発の有識者の確認が有効
  - ✓ アジャイル開発の有識者がアジャイル開発プロジェクトの怪しい香りを探り出す
  - ✓ 標準的な品質管理指標の提供、開発チームは標準から必要な指標を選択  
（開発チームごとに適した品質管理指標があり、チームの自立性を重視することが重要）
  - ✓ スプリントバックログに、「品質評価」や「品質向上取組み」を設定する
  - ✓ ユーザーストーリーの受け入れ基準やDoneの定義に要求される品質を明記する
  - ✓ 開発チームの進捗や生産性に変化がないかモニタリングする

## ここまでのまとめ

---

- アジャイル開発とウォーターフォール開発では品質管理の仕方が違う？



- ◆ ウォーターフォール開発との「品質管理」面での差はない？
- ◆ アジャイル開発を特別な開発手法と考えないほうが良いのではないか？



- アジャイル開発とウォーターフォール開発では、開発プロセスが異なる。よって、大局的には差はないとしても、アジャイル開発の特徴を損なわない品質の管理が必要ではないか？。また、重要視するものが異なるのではないか？（これまでの議論も振り返りながら・・・。）

# アジャイル開発とウォーターフォール開発の違い

## 【開発プロセス面からの比較】

ウォーターフォール開発		アジャイル開発	
プロセス	品質確認・管理活動	プロセス	品質確認・管理活動
<ul style="list-style-type: none"> <li>開発の工程を「企画」、「設計」、「実装」、「テスト」に分割した上で開発を進める。</li> </ul>	<ul style="list-style-type: none"> <li>工程を跨いだ戻りが発生しないように、ゲートレビューなどで工程ごとの品質を確認</li> </ul>	<ul style="list-style-type: none"> <li>短期間（数週間～1か月程度）で、「動くソフトウェア」を作ることの反復し、機能を追加しながら開発を進める。</li> </ul>	<ul style="list-style-type: none"> <li>スプリントレビューで「動くソフトウェア」の機能を確認</li> </ul>
<ul style="list-style-type: none"> <li>それぞれの工程は決められた技術者が担当する。</li> <li>各工程での成果物（ドキュメント等）が次工程へ引継がれて、ソフトウェアが完成する。</li> </ul>	<ul style="list-style-type: none"> <li>工程の担当者が異なることによる品質影響を抑止するため、各工程の成果物を構造的に作成し、ドキュメントを精査する</li> </ul>	<ul style="list-style-type: none"> <li>開発は、小規模で固定された技術者が、「企画」～「設計」～「実装」～「テスト」を担当する。</li> <li>成果物はソフトウェアを作るための情報伝達に必要なものだけ作成する。</li> </ul>	<ul style="list-style-type: none"> <li>ペアプログラミングなどのモブワークにより、開発チーム一体でソースコードやテストの品質を高める</li> <li>ソースコード品質の向上により、保守容易性を向上する</li> </ul>
<ul style="list-style-type: none"> <li>「企画」の段階で、開発対象のソフトウェアの機能を全て決める。</li> </ul>	<ul style="list-style-type: none"> <li>「企画」段階で決定した機能で各工程の成果物を検証する</li> </ul>	<ul style="list-style-type: none"> <li>実装（追加）するソフトウェアの機能は、反復ごとにその時点で必要なものがバックログから抽出される。</li> </ul>	<ul style="list-style-type: none"> <li>スプリント単位で実装機能と要件（バックログ）の一致を検証する</li> </ul>
<ul style="list-style-type: none"> <li>動くソフトウェアの検証は、「テスト」工程で実施される。</li> </ul>	<ul style="list-style-type: none"> <li>プログラム(単体→機能)→システムの順に高網羅性→広範囲にテスト設計を行い検証</li> </ul>	<ul style="list-style-type: none"> <li>機能実装の都度、「テスト」行われる。</li> </ul>	<ul style="list-style-type: none"> <li>テストの自動化、逐次統合、CIなどにより、検証を効率化し、品質を高める</li> </ul>

# まとめ

➤ アジャイル開発とウォーターフォール開発では品質管理の仕方が違う？



◆ 大局的には、ウォーターフォール開発との「品質管理」面での差はない。

◆ しかし、開発プロセスが異なるので、ソフトウェアの品質を確保する上で、重視するポイントは異なる。

	ウォーターフォール開発	アジャイル開発	
	ポイント	ポイント	品質管理の留意事項
要件定義	ユーザにて業務要件定義	ユーザ（PO）にてプロダクトバックログの作成～メンテ	ストーリーやDoneに達成すべき品質を定義する
チーム開発	担当の開発力の差は、工程ごとの成果物の精査によって均質化	担当の開発力の差は、開発チームメンバで相互補完	開発力は、顧客の要求を含めた個々のチーム力に依存するため組織の基準値ではなく、チームのトレンド分析が重要
ソースコード	—	シンプルで理解し易くするリファクタリングを行う	不具合が混入しにくいだけでなく、要件の追加・変更に対応し易くする
テスト効率化	—	自動化を行う	設計・実装・テストを繰返し実施するうえでテストの自動化は重要

## アジャイル開発推進者・開発者の「品質管理」に関する悩み

- 管理者層や品質管理部門（第3者）への説明
  - ✓ ウォータフォール開発と同様の品質管理手法の適用や評価を求められる。
    - 設計ドキュメントなどの中間成果物の品質報告
    - リリース直前には十全なテストの実施
  
- 品質評価のタイミング
  - ✓ いつ、どんな指標を使って品質を評価すればよいかわからない。
  - ✓ バグを見つけたらすぐに修正するため、バグの件数や混入原因など基づく分析がし難い。
  
- テストの抜け・漏れ防止や負荷の軽減
  - ✓ 開発者の思い込みより、テストケースがテストシナリオごと抜けてしまい、品質課題を開発を発生させ、大きな問題となった。
  - ✓ 短い期間での反復開発であるため、効率的なテストの実行が求められる。

# アジャイル開発推進者・開発者の「品質管理」に関する悩み

## ➤ 管理者層や品質管理部門（第3者）への説明

- ✓ ウォーターフォール開発と同様の品質管理手法の適用や評価を求められる。

- ・ウォーターフォールとは目的が異なるのでアジャイルを採用している、という理解は必要。
- ・品質管理の手法に大差はないが、プロセスの違いからウォーターフォールと同じ指標をすべて収集できるわけではない。開発チームで測っている測定値での報告を行うにあたり、開発チームに無理な測定作業を課さないことが重要。
- ・ユーザーストーリーの品質が満たされていることをユーザが承認している、という報告も重要。

## ➤ 品質評価のタイミング

- ✓ いつ、どんな指標を使って品質を評価すればよいかわからない。
- ✓ バグを見つけたらすぐに修正するため、バグの件数や混入原因などに基づく分析がし難い。

- ・テストスプリントや各スプリント完了時に評価を実施する。（2W~4Wごと、というのは妥当か？かなりの高頻度）
- ・スプリント内で発見されたバグの分析は強要せず、過去のスプリントをすり抜けたバグの分析・対策が製品の品質向上につながる。
- ・スプリント毎に品質目標を立て、品質が弱いところは次スプリントで強化させる。

## ➤ テストの抜け・漏れ防止や負荷の軽減

- ✓ 開発者の思い込みより、テストケースがテストシナリオごと抜けてしまい、品質課題を発生させ、大きな問題となった。
- ✓ 短い期間での反復開発であるため、効率的なテストの実行が求められる。

- ・テストの自動化など、継続的なインテグレーションやデプロイが行える環境を準備することで、品質を確保する。
- ➡ 自動化しても「テストケースがシナリオごと抜けてしまう」のは抑止できないが、アジャイル開発の課題ではなく、体制の課題。

## 新たな意見

- ユーザ企業やシステム部門の上位層がアジャイル開発への理解を深める指標を示せないか？
  - ✓ 日本で基幹系システム分野への適用がまだ多くない理由として、ウォーターフォール開発での品質を重要視したソフトウェア評価が定着し、スタンダードになっていることが考えられる。
  - ✓ アジャイル開発は、ユーザーの要求の変化に対応して、開発途中で仕様が見直されることを前提に、小規模な実装とテストをくり返しながらか開発する。開発プロセスが異なるものを、ウォーターフォール開発と同じ土俵で評価することに無理がある。

### アジャイルの特性(メリット)とそれを示す指標(案)を収集

- ◆ **アジャイル開発への理解を深める指標は2つに分類されるのではないか？**
  - ① 開発の結果（ゴール）を示す指標
  - ② 開発プロセスの有効性を示す指標

**次年度は、この辺りももう少し議論（深掘り）する予定**

# アジャイル開発への理解を深める指標

## ①開発の結果（ゴール）を示す指標（案）

#	アジャイル開発の適用メリット	メリットを示す指標（案）
1	使い勝手を見ながら、仕様変更・仕様追加によりソフトウェアの完成度を磨くことができる	プロジェクト開始後に追加変更されたバックログの数と実現された件数
2	効果の早期回収の把握により、企画妥当性を早期に評価できる	想定効果の回収率または累計効果実績値
3	開発スタート時点で業務要件が曖昧なものに対して、（限られた予算、期間の中で）ユーザー価値を最大化し、変化に柔軟に対応できる	顧客（ユーザー）満足度

# アジャイル開発への理解を深める指標

## ②開発プロセスの有効性を示す指標

#	アジャイル開発の適用メリット	メリットを示す指標（案）
1	（自律性）チームの開発プロセスの最適化	ベロシティの向上、安定
2	個人と対話	顧客および利用者との程度対話 （対話回数/週、対話時間/週）
3	最小モジュール単位に分割し開発するため、リリースを高頻度化することができる	ベロシティ
4	スプリントを繰り返す中で継続的に改善（成長）するチームを作ることができる	・スプリントのベロシティの向上率 ・レトロスペクティブでの改善案の件数
5	同一チームで一定期間活動することで、ある程度同一コストでの成長（生産性、品質）を享受できる	スプリント毎の生産性、品質（バグ発生低下）および見積もり精度の向上と計画乖離率で評価
6	スプリントゴールを明確にして、その作業に集中することで、目標を達成しやすい	スプリントバックログの達成率、割り込み作業の数
7	小さくリリースし、持続的に頻繁にカイゼンを行う（投資回収の早期開始）	最初のリリースまでのリードタイム、単位期間あたりのリリース回数
8	開発チームの成長を前提とした開発生産性の向上	ベロシティ（開発チームのスプリント毎の生産性）成長曲線