

品質重視のアジャイル開発 ～生き残りの鍵は、技術力と定量化～

2021年 6月26日
株式会社イデソン
誉田 直美

自己紹介

氏 名：簗田 直美（ほんだ なおみ）
現 職：株式会社イデソン 代表取締役
公立はこだて未来大学 客員教授 博士（工学）

略 歴：

ソフトウェア品質の専門家として、30年以上に渡って大手電機会社で活躍。ウォーターフォールモデル開発およびアジャイル開発の両方に精通し、現場での豊富な経験を保有している。近年では、AIシステムの品質保証にも関わる。単なる知識にとどまらず、現場の事情を考慮した実践的な品質保証が特徴である。
2020年、(株)イデソン設立。ソフトウェア品質に関するコンサルティングを中心に活動している。執筆や講演による啓蒙活動にも力を注いでいる。<https://ideson-worx.com/>

主な著書・執筆活動：

品質重視のアジャイル開発 ～成功率を高めるプラクティス・Doneの定義・開発チーム編成～ （日科技連出版）2020年9月発行
ソフトウェア品質判定メソッド ～計画・各工程・出荷時の審査と分析評価技法～ （日科技連出版、編著）2019年8月発行
ソフトウェア品質会計（日科技連出版）2010年発行 <2010年度 日経品質管理文献賞受賞>
ソフトウェア品質知識体系ガイド 第3版-SQuBOK Guide V3-（オーム社、監修）2020年11月発行。なおV2（2014年11月発行）は執筆リーダー、V1（2007年11月発行<2008年度 日経品質管理文献賞受賞>）は著者として執筆に継続的に関与
ソフトウェア開発 オフショアリング完全ガイド（日経BP社 共著）2004年10月発行
見積りの方法（日科技連出版、共著）1993年

受賞：

品質管理学会 品質管理推進貢献賞（2020/11）
プロジェクトマネジメント学会 文献賞（2016/9）
第5回世界ソフトウェア品質国際会議（5WCSQ）最優秀論文賞および最優秀発表賞（2011/11）
第4回世界ソフトウェア品質国際会議（4WCSQ）最優秀論文賞（2008/9）

学会：情報処理学会、品質管理学会、プロジェクトマネジメント学会

委員活動：

日科技連SQiPソフトウェア品質委員会 副委員長
プロジェクトマネジメント学会 上席研究員
筑波大学大学院 非常勤講師（2012年～2016年） 鳥取大学 非常勤講師（2017年）

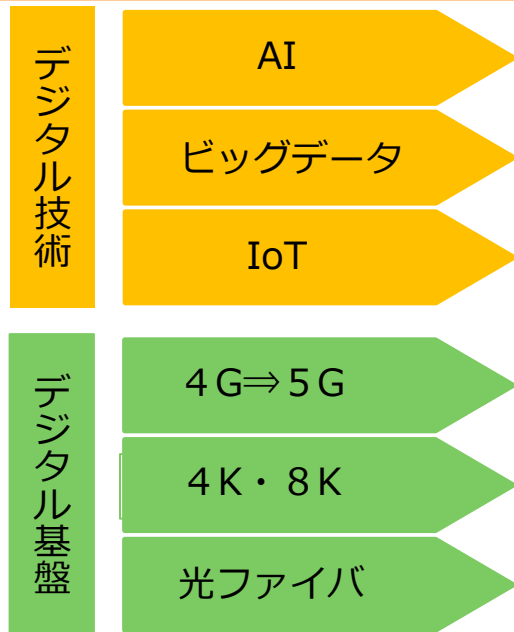


1. はじめに

IoT, AI, 5Gを軸とした新たな価値創造へ

Before Corona

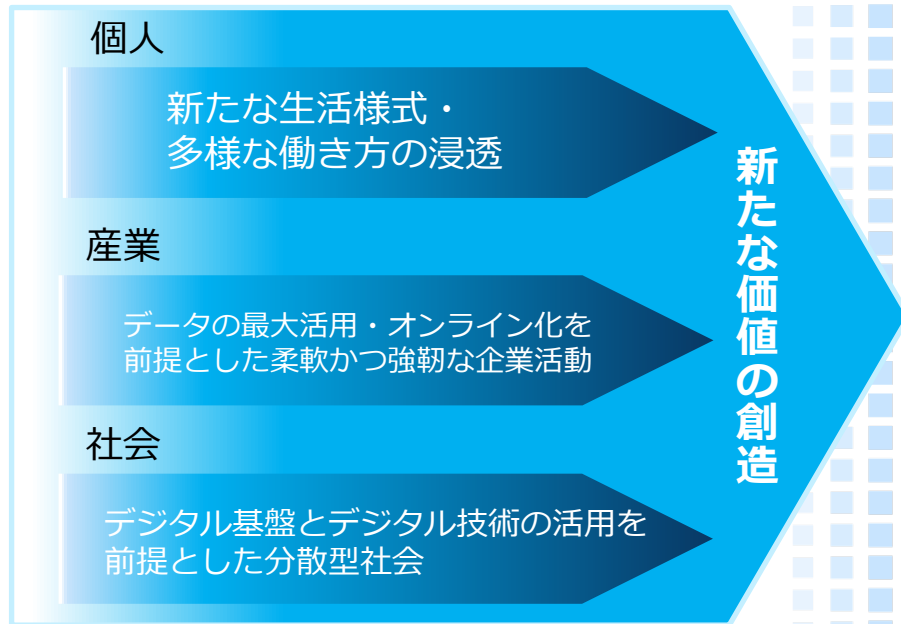
デジタル基盤整備及びデジタル技術活用により
デジタルトランスフォーメーションを推し進め
産業の効率化や付加価値化を目指してきた



新型コロナウイルス感染症の
世界的流行

新常態

人の生命保護を前提にサイバー空間とリアル空間が
完全に同期する社会へと向かう不可逆的な進化が
新たな価値を創出



出典：令和2年版情報通信白書（総務省）

<https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r02/summary/summary01.pdf>

を著者が一部変更

不確実な時代だからこそアジャイル開発が求められる

その期待に、アジャイル開発は応えていますか？

- アジャイル開発でバグ多発！ どうすればよい？
- 顧客から、品質確保の説明を求められて困っている
- アジャイル開発を始めたいが、品質が大丈夫か不安
- アジャイル開発に挑戦したら、大失敗
同じ失敗は繰り返したくない

そのとき、テストエンジニアに求められるのは何でしょうか？

2.ソフトウェア品質保証の原則

ソフトウェア品質保証の原則

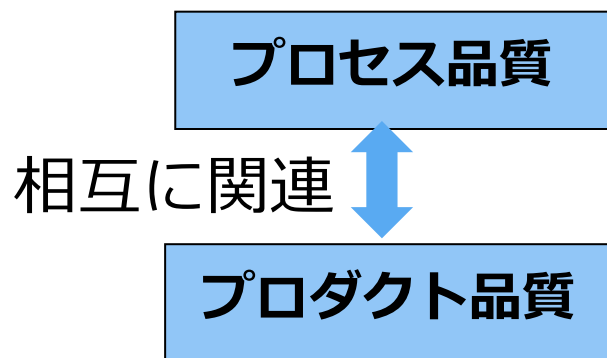
■ プロセス品質とプロダクト品質の両面から保証する

- プロセス品質

- 品質の良いものを作り出すプロセスを構築し、そのプロセスを適用する

- プロダクト品質

- 作られたプロダクトの品質が、実際に良いことを確認する



※片方だけ良くするのは通用しない

- 「モノさえよければ、作り方はどうでもよい」は、現実には成立しない
- 「開発標準が守られない場合は、ほぼ確実に品質の欠如が発生する」

R. S. Pressman

アジャイル開発でも、基本的な考え方は同じ

「品質保証」は、顧客の納得が鍵 <確証、客観性が重要>

■ 品質保証

品質要求事項が満たされるという**確信を与えることに焦点を合わせた**品質マネジメントの一部

■ 品質

本来備わっている特性の集まりが、要求事項（※）を満たす程度

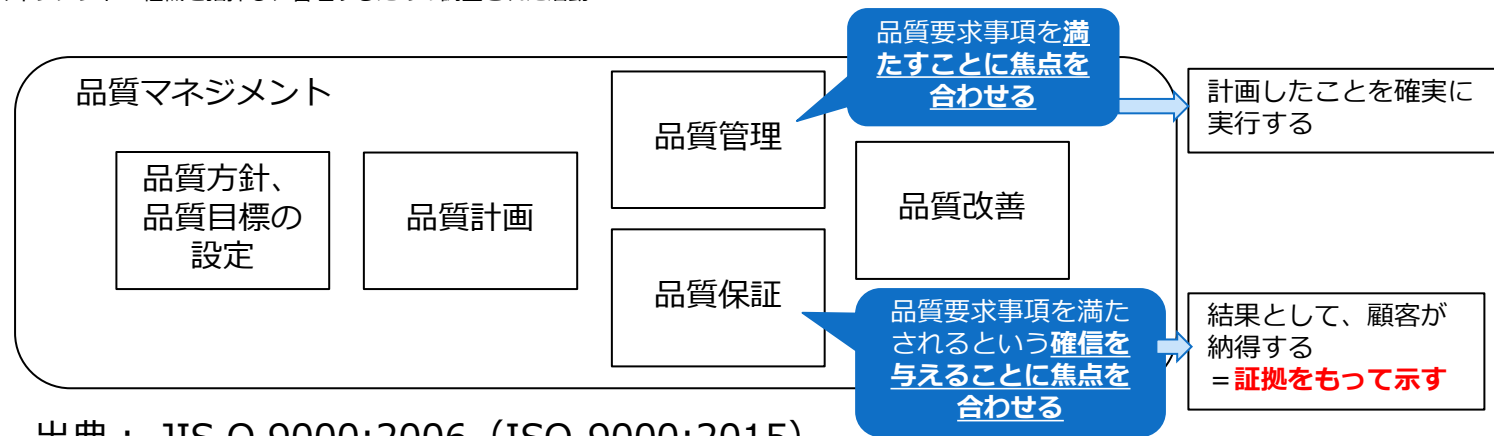
※要求事項 … 明示 / 暗黙的に了解 / 義務として要求されているニーズまたは期待

■ 品質マネジメント

品質に関するマネジメント

※品質マネジメントには、品質方針及び品質目標の設定、並びに品質計画、品質保証、品質管理
及び品質改善を通じてこれらの品質目標を達成するためのプロセスが含まれ得る

※マネジメント…組織を指揮し、管理するための調整された活動



出典： JIS Q 9000:2006 (ISO 9000:2015)

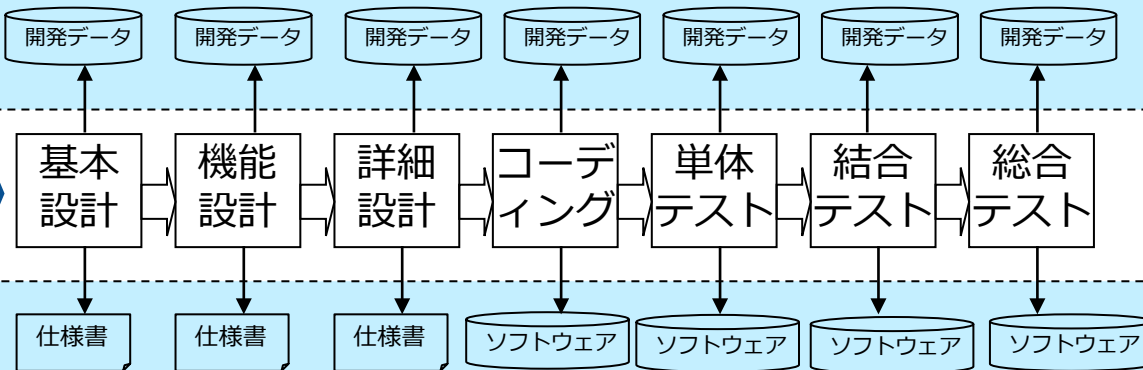
代表例：SWのウォーターフォールモデル開発①全体像

適用結果を
フィードバック

定量データ分析、バグのなぜなぜ分析

適用

＜プロセス品質＞ 各工程の実行状況に対する分析評価



＜プロダクト品質＞ 各工程の成果物に対する分析評価

適用

設計やテスト仕様書の品質評価、ソフトウェアの品質評価

開発標準

- 各工程で実施すべき作業項目
- 工程移行基準

- 設計標準
- 仕様書標準
- テスト標準
- コーディング規約

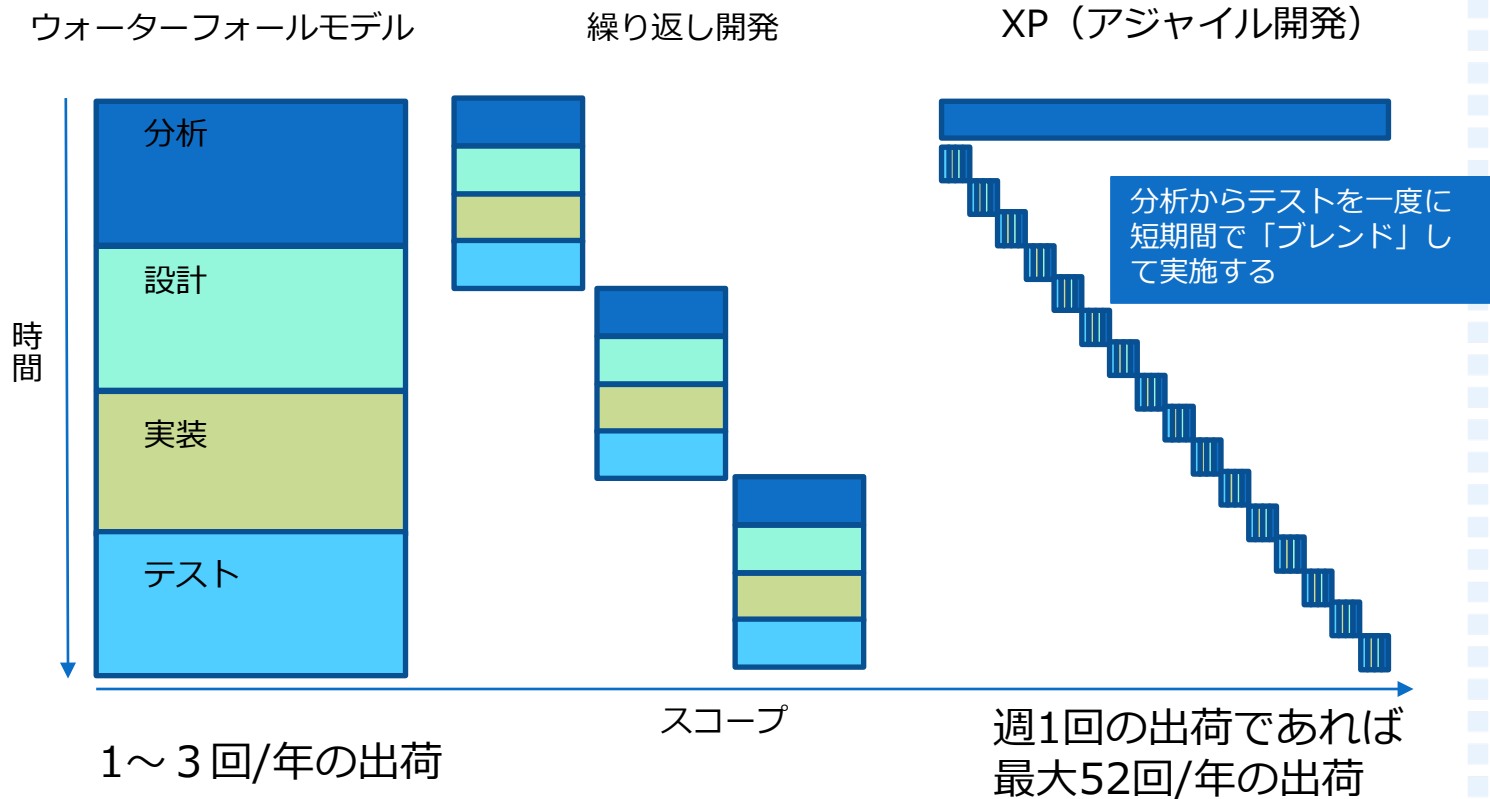
出荷判定

出荷

3.アジャイル開発の品質保証の難しさは どこにあるか

アジャイル開発の品質確保の難しさは「短期間」

(短いウォーターフォールモデル開発ではない)



出典：K. Beck, Embracing Change with Extreme Programming. Computer, 32, 70-77, 1999
に基づき、筆者が追記

主に欧米でよく適用されるプラクティス

No	プラクティス	適用率(%)
1	朝会	86
2	スプリント計画	80
3	ふりかえり	80
4	スプリントレビュー	80
5	短いイテレーション	67
6	プランニングポーカー（チームでの見積もり）	61
7	カンバン	61
8	リリース計画	57
9	献身的な個客・プロダクトオーナー	57
10	単一チーム（開発とテストの統合）	54
11	頻繁なリリース	50
12	共通の仕事場	45
13	プロダクトロードマッピング	45
14	ストーリーマッピング	38
15	アジャイルポートフォリオ計画	33
16	アジャイル・リーンUX	28

No	プラクティス	適用率(%)
1	単体テスト	69
2	コーディング規約	58
3	継続的インテグレーション（CI）	53
4	リファクタリング	41
5	継続的デリバリー（CD）	40
6	継続的開発	35
7	ペアプログラミング	34
8	テスト駆動開発（TDD）	33
9	自動受入テスト	33
10	コードの共同所有	31
11	持続的ペース	25
12	振る舞い駆動開発（BDD）	22
13	エマージェント・デザイン	14

※複数回答可

問題解決法として「自己組織化」を重視（しすぎ）

- 自己組織化した開発チームは、常に改善が回る



- しかし…

- 自己組織化するまでに時間がかかる
 - 多重下請構造に慣れた技術者が、自律した技術者になるまで、時間がかかる（1~2年か）
 - ✓ 経営陣は、それまで待てるだろうか？
- 自己組織化しているかどうか、客観的に判断できない
 - 自己組織化の定義や判断基準はない

日本の品質レベルを支えてきた技術の利用（が不十分）

<パフォーマンスデータの比較>

	インド	日本	米国	欧州他	全体
プロジェクト数	24	27	31	22	104
新規コード行数/人月*	209	469	270	436	374
出荷後12ヶ月バグ数/KL*	.263	.020	.400	.225	.150
参考：100KLのソフトウェア の出荷後12カ月のバグ数	26件	2件	40件	23件	15件

*：メジアン（中央値）を表示

日本は米国の1/20

出典：M. Cusumano, "Software development worldwide: the state of the practice", IEEE Software, 2003（最後の行は筆者が追記）

■ 日本の品質レベルを支えてきた技術とは

- バグにこだわる考え方と取り組み
 - レビュー
 - バグのなぜなぜ分析と水平展開
 - 品質ゲート など

主要なメトリクスが相対値（客観性を担保しにくい）

■ アジャイル開発の主要なメトリクスは、開発チーム内でしか通用しない相対値

- 開発チーム内では効果的に使える
- 開発チーム外との比較や基準値の横展開ができない

■ アジャイル開発でよく使用されるメトリクス

・ストーリーポイント

- その要件を実現するのに必要な作業量の相対値
 - ・工数（人時）の見積もりより、簡単で有効
 - ・工数（人時）の見積もりのように正確性を要求されない

・ベロシティ（速度）

- その要件を実現するのに必要な作業量の相対値
 - ・ベロシティを継続的に追跡することにより、その開発チームの開発できる量（開発できる速度）がわかる

3.品質重視のアジャイル開発とは

この状態をブレイクスルーするには？！

品質重視のアジャイル開発のポイントは「短期間」への対応

■ 「短期間」への対応

アジャイル開発とは、「所定の品質を確保したソフトウェアを
短期間で繰り返しリリースする手法」

■ 短期間への対応ポイント

1. 初めから品質を作りこむ＜プロセス品質＞

⇒ **プラクティスと開発チーム編成**

- ・ 後からバグ出しをする方法では間に合わない
- ・ 技術系プラクティスをリードできる技術リーダーが必須

2. 品質が作りこまれていることを、常に確認する＜プロダクト品質＞

⇒ **Doneの定義と定量化分析**

- ・ プロセス品質とプロダクト品質の基準でDoneの定義を構成
- ・ 定量化したメトリクスを収集し、過去や他チームと比較分析



推奨プラクティス ～3種類のプラクティスを意識する～

No	マネジメント系プラクティス
1	スプリント計画
2	デイリースクラム
3	要求ワークショップ
4	スプリントレビュー
5	振り返り (KPT)
6	タスクボード
7	バーンダウンチャート

No.1～5はスクラムによるものである

アジャイル開発宣言
から20年の知見より

No	技術系プラクティス
1	ペアワーク
2	テスト自動化と継続的インテグレーション
3	リファクタリング
4	ソースコードの共同所有
5	レビュー* ¹
6	ドキュメント化* ²
7	バグのなぜなぜ分析と水平展開* ³

アジャイル開発宣言
から20年の知見より

日本の品質レベルを
支えてきた技術より
(アジャイル開発で検証済)

No.1～4はXPによるものである (No.1はXPではペアプログラミングと表記)

*¹ソフトウェア工学で説明されるフェイスツーフェイスのレビューである

*²仕様書作成を明示するためのプラクティスである

*³バグはプロセス課題を示し、水平展開による残存バグ摘出が必須である

品質を決めるのは、技術系プラクティス

※テスト系プラクティスとバグにこだわる日本の技術は、テストエンジニアの領域



推奨する開発チーム編成 ～一定以上の技術力を確保～

立場	要件
技術リーダー	自ソフトウェア開発組織の上位20%に入る技術者。 コミュニケーション力があるほうが望ましい
開発チーム員	自ソフトウェア開発組織の上位80%に入る技術者

■ 優秀な技術リーダーの設定

- 技術系プラクティスの適切な実施は、技術リーダーの実力に大きく左右される

■ 開発チーム員は一定レベル以上をアサイン

- 技術レベル1 A以上を推奨
 - 1 A:適用する技法を使って、
 - 自由裁量部分を遂行できる

開発者の技術レベル

レベル	手法の理解と利用の技術レベル
レベル3	適用する条件に適合するように、手法そのものを改訂できる
レベル2	適用する条件に合わせて、手法をカスタマイズできる
レベル1	手法を適用できる
レベル1A	手法を使って、自由裁量の部分を遂行できる (経験を積みばレベル2になることができる)
レベル1B	手法を使って、手続き的な部分を遂行できる (経験を積みば、レベル1Aのスキルのいくつかをマスターできる)
レベル-1	手法を使えない、または使わない

出典：アリストター・コックバーン（著）、長瀬嘉秀/永田渉(監訳)、「アジャイルソフトウェア開発」、2002年、ピアソン・エデュケーション
およびB・バーム/R・ターナー（著）「アジャイルと規律 ～ソフトウェア開発を成功させる2つの鍵のバランス～」、2004年、日経BP社
から筆者作成

推奨するDoneの定義

～出荷OKの基準を予め明確化し、客観的に審査～

スプリント	バックログID	バックログ内容	審査日	審査者

対象	カテゴリ	No	審査観点	審査基準	確認する成果物	実績値	判定結果
プロセス品質	1. ツールによる確認	①	静的解析	ソースコード静的解析による高以上の指摘に対する修正 100%	ツール名:		
		②	実行行数	関数単位の実行行数200行以下 100%	ツール名:		
		③	ネスト	ネスト4以下 100%	ツール名:		
		④	テストカバレッジ	テストカバレッジ 100% ※何らかの理由により実行していないコードは、レビューにより妥当性を確認済である	ツール名:		
		⑤	セキュリティ	セキュリティ脆弱性検査による指摘に対する修正 100%	ツール名:		
		⑥	OSS	OSSライセンス違反検査による指摘に対する修正 100%	ツール名:		
	2. テスト結果	⑦	新規テスト	新規テスト実施率 100% ※非機能テストなど、機能以外のテストを含む	新規テスト仕様書		
		⑧	リグレッションテスト	リグレッションテスト実施率 100%	リグレッションテスト仕様書		
		⑨	テスト自動化	単体テスト自動化率 100% ※他のテストもできるだけ自動化する	テスト実施結果		
		⑩	実行結果の記録	テストの実行結果の記録 100%	テスト実施結果		
プロダクト品質	3. 成果物の評価	⑪	設計仕様書	設計仕様書の成果物評価による指摘に対する修正 100%	設計仕様書		
		⑫	テスト仕様書	テスト仕様書の成果物評価による指摘に対する修正 100%	テスト仕様書		
		⑬	ソフトウェア	ソフトウェア評価による指摘に対する修正 100% ※ソフトウェア評価対象には、マニュアル、インストーラーなど顧客に提供するものすべてが含まれる	評価結果報告書		
	4. 成果物の登録	⑭	設計仕様書	設計仕様書一式の構成管理ツールへの登録 100% ※技術調査メモなどの必要情報を含む	設計仕様書一式		
		⑮	ソースコード	ソースコード一式の構成管理ツールへの登録 100%	ソースコード		
		⑯	テスト仕様書	テスト仕様書一式の構成管理ツールへの登録 100%	テスト仕様書一式		
		⑰	テストコード	テストコード一式の構成管理ツールへの登録 100%	テストコード一式		
	5. バグ対応	⑱	未解決バグ	未解決バグ 0件	バグ記録		
		⑲	水平展開	バグ分析と水平展開実施率 100% ※対象: 当該スプリントで摘出された全バグ	バグ分析、水平展開実施結果		
	6. アクションアイテム対応	⑳	アクションアイテム対応	残存アクションアイテム 0件 ※対象: 当該スプリントで対応予定のアクションアイテム	アクションアイテムリスト		

プロセス品質

プロダクト品質

ツールで確認できるものは
自動計測・自動判定

実施すべきテストを確実に実施し、
テスト自動化していることを確認

審査者が記録

品質技術者が、実際に
仕様書の中身を見たり、
SWを動かす

残存バグがないことを
確実にする

定量化：比較を可能にするメトリクスの収集

＜手動計測を推奨するメトリクス＞

カテゴリ	単位	データ項目	説明
ストーリーポイント	ストーリーポイント (SP)	ベロシティ	今回のスプリントで完了したバックログ項目のストーリーポイントの合計値
		未完了ストーリーポイント	今回のスプリントで完了しなかったバックログ項目のストーリーポイントの合計値
Done判定後のバグ	件数	Done判定後のバグ	Done判定を受けた後に摘出したバグ
		うち新規バグ	Done判定を受けた機能から摘出されたバグ
		うちデグレードバグ	ある機能の開発の影響を受けて既存機能が動作しなくなったバグ
		うち既存バグ	既存機能から摘出したバグ
工数	人H	工数	開発チームが、アジャイル開発に費やした工数
		うちペアワーク	ペアワークに費やした工数
		うちレビュー	レビューに費やした工数。レビュー記録票の合計工数を基本とする
テスト項目数	項目	新規テスト項目	今回のスプリントで開発した機能に対するテスト項目
		リグレッションテスト項目	既存機能に対するテスト項目

※テスト項目数は、自動実行と手動実行が混在する場合が多いため、本一覧へ入れている。

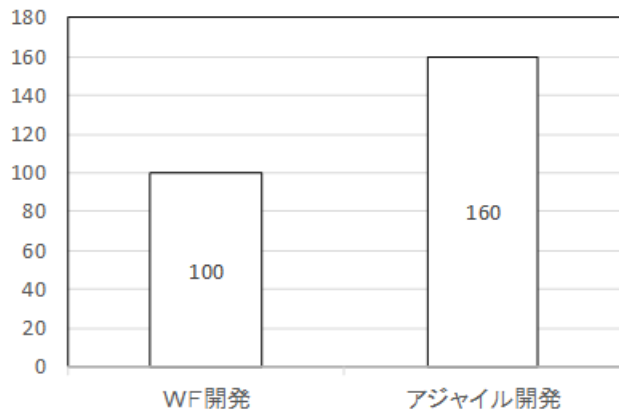
※アジャイル開発で収集するメトリクスは、上記以外にDoneの定義で自動収集する項目①～⑥がある



(参考) ウォーターフォールモデルとアジャイル開発の比較：生産性

新規開発：生産性

生産性 (Line/人H)
(相対値)

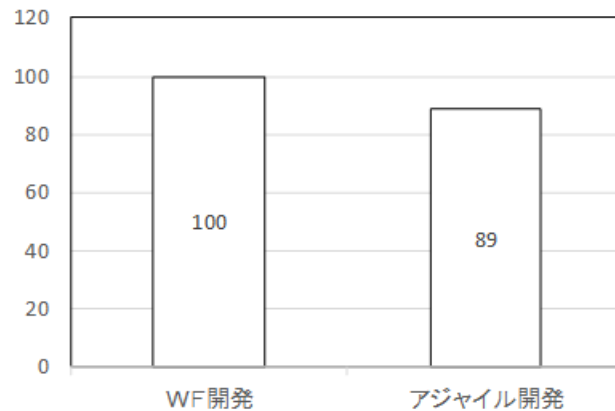


新規開発の生産性は、アジャイル開発のほうが良い

- ・アジャイル開発に向く開発対象や開発方法を選ぶためと推測

改修開発：生産性

生産性 (Line/人H)
(相対値)



改修開発の生産性は、ややアジャイル開発のほうが悪い

- ・ウォーターフォールモデルで開発したシステムは、テスト自動化などアジャイル開発に向く要件を満足していないためと推測

※新規開発：ベースとなるシステムがなく、新規に開発するプロジェクト

改修開発：ベースとなるシステムに対して機能追加を行うプロジェクト

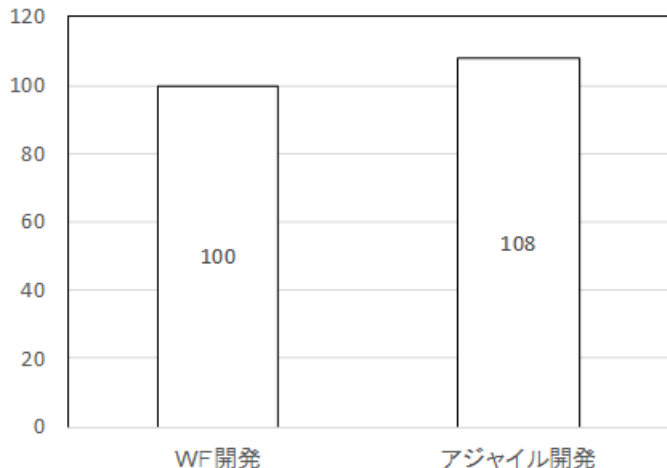
※データの出典：2015~2018年に大手電機メーカーで実施したソフトウェア開発プロジェクトデータ

©2021 IDESON から、ウォーターフォールモデル開発とアジャイル開発の開発規模がほぼ同等のプロジェクト300件

(参考) ウォーターフォールモデルとアジャイル開発の比較：テスト項目

新規開発：テスト項目

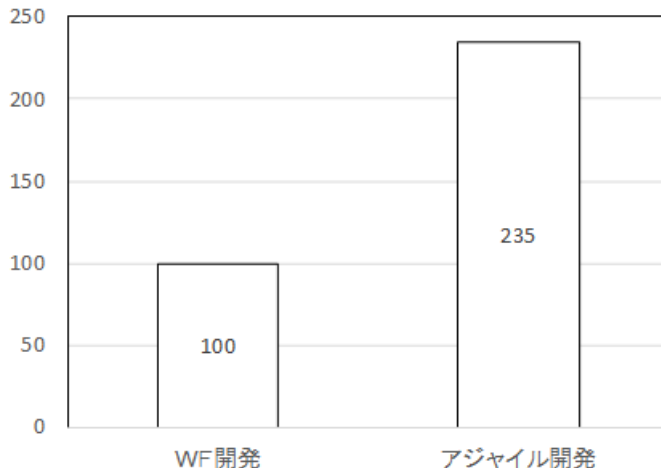
テスト項目数(項目/KL)
(相対値)



新規開発のテスト項目は、ややアジャイル開発が多い

改修開発：テスト項目

テスト項目数(項目/KL)
(相対値)



改修開発のテスト項目は、アジャイル開発のほうが多い

アジャイル開発のテスト項目は、多くなる傾向にある

- 論理的には同等になるものと思われるが、アジャイル開発の開発単位が小さい、ツールによる自動化を進めるなどの影響と推測

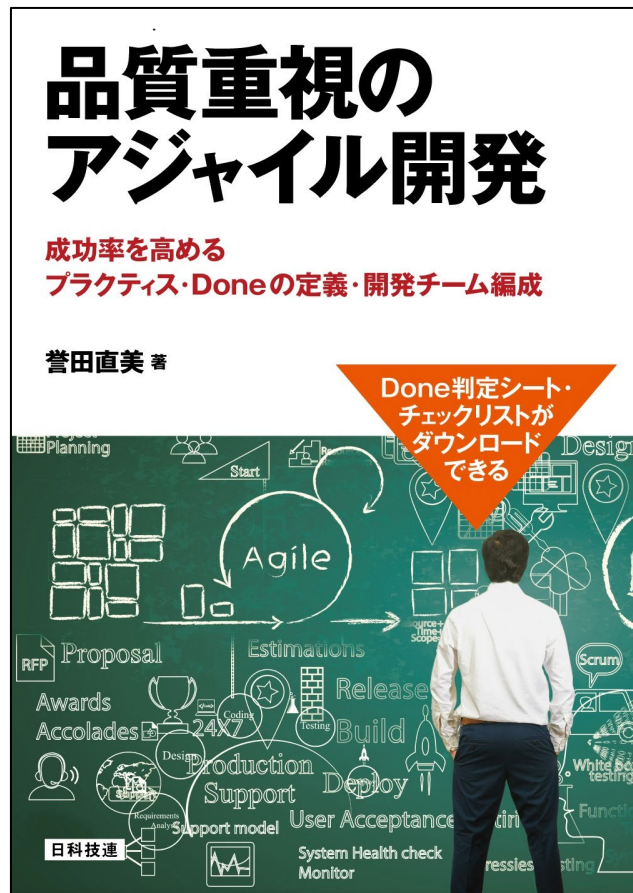
4.さいごに

さいごに

- アジャイル開発は、DX時代の主流になる
 - ・品質重視のアジャイル開発で基本を学び、そこから変化させる
- アジャイル開発は従来以上に技術力が求められる
 - ・テスト系プラクティスは、品質保証の要
 - ・バグにこだわる日本の技術は、バグ低減へつながる
- 定量化による客観的な分析は、ソフトウェア開発の基本
 - ・「比較」は顧客の納得性を高める

テストエンジニアの生き残りの鍵は
技術力と定量化





品質重視のアジャイル開発 ～成功率を高めるプラクティス・ Doneの定義・開発チーム編成～ 菅田 直美[著]

2020年9月26日発行

日科技連出版社 定価 2700円（税抜き）

ISBN

[目次]

1. アジャイル開発の概要
2. アジャイル開発を取り巻く課題
3. アジャイル開発の品質保証の実態
4. 品質保証のポイントを踏まえたアジャイル開発
5. アジャイル開発の準備とプラクティス
6. アジャイル開発のDone判定と品質技術者の役割
7. アジャイル開発におけるメトリクスの活用
8. リモートアジャイル開発の実際



<https://ideson-worx.com>

