\Orchestrating a brighter world NEC

アジャイル開発における
ITベンダー・メーカーの品質保証／検査の取り組み
日本電気株式会社編

2019年7月18日

日本電気株式会社

Orchestrating a brighter world

未来に向かい、人が生きる、豊かに生きるために欠かせないもの。
それは「安全」「安心」「効率」「公平」という価値が実現された社会です。

NECは、ネットワーク技術とコンピューティング技術をあわせ持つ
類のないインテグレーターとしてリーダーシップを発揮し、
卓越した技術とさまざまな知見やアイデアを融合することで、
世界の国々や地域の人々と協奏しながら、
明るく希望に満ちた暮らしと社会を実現し、未来につなげていきます。

NECからの取り組み紹介

前半

NECでの組織的なアジャイル開発の品質保証の取り組み状況を紹介

ソフトウェアエンジニアリング本部
水野　浩三

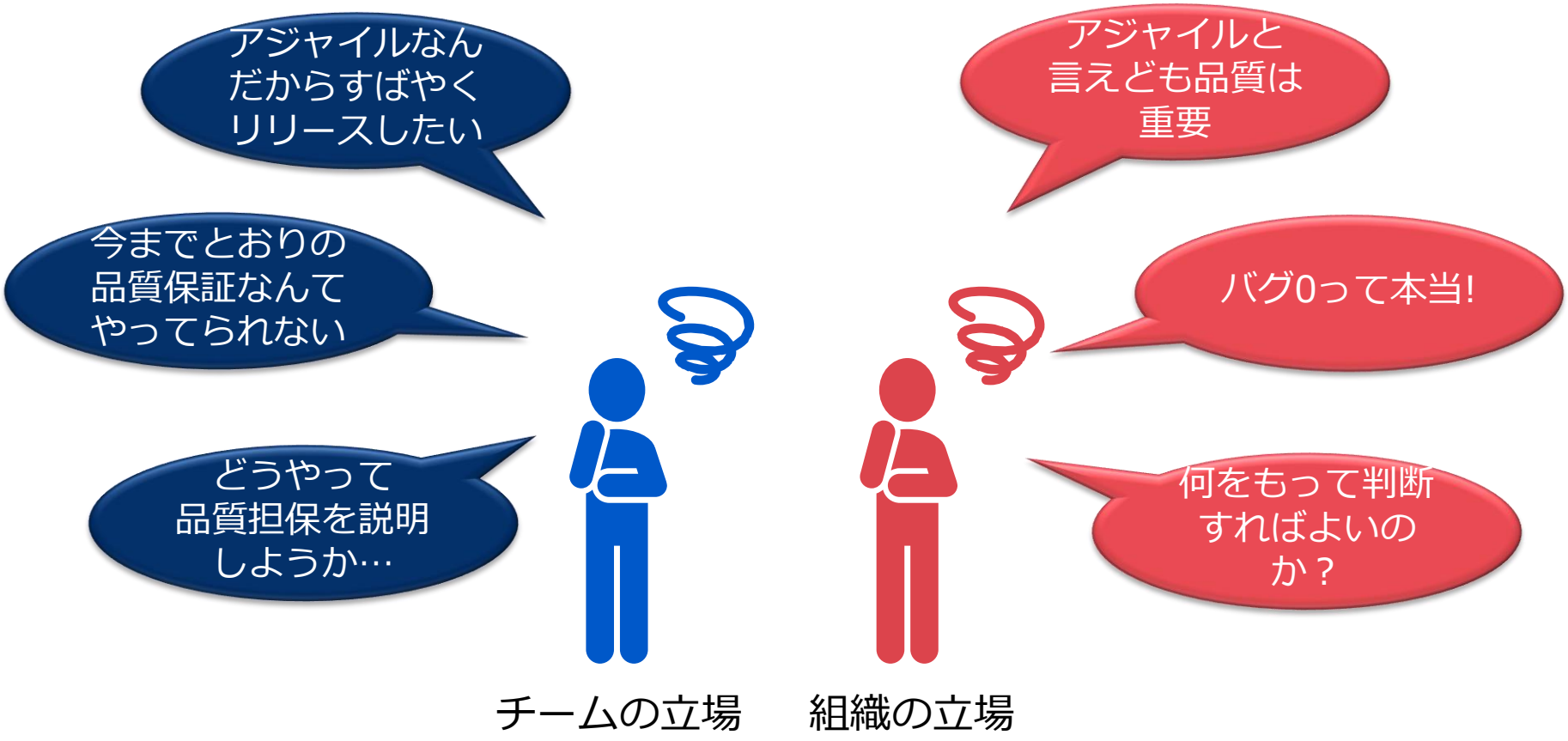
後半

循環的複雑度を利用した品質定量評価の事例を紹介

スマートエネルギー事業部
近藤　靖彰

アジャイル開発の品質に関するジレンマ

チーム、組織それぞれの立場で悩み、葛藤、時には対立も。
過去の実績、従来のやり方に固執しがち。



組織的なアジャイル開発品質保証の取り組み

アジャイル開発に対する取り組み変遷

2000年代中～下旬

世界的にアジャイル開発が普及している状況に危機感

2010年～2012年

アジャイル開発の推進を行うも、ビジネス的盛り上がりはなく活動は限定的

2012年～2016年

「NECアジャイル」としてツールや方法論の整備・人材の育成を行い、アジャイル開発の備えを整える

- 従来の延長での開発
- 品質問題が顕在化

2016年～現在

製品開発を中心に、アジャイル開発としての品質保証の枠組みを明確化

- 本来のアジャイル増
- 俊敏さが求められる

製品開発を中心にプラクティスを積み、ノウハウ蓄積
&　技術トレンド/事業環境変化の中で重要性アップ

開発プロセスと品質保証

案件によって達成すべき品質目標は異なるため、一律の対応はできない。
プロダクトの特徴に応じて対応するレベルを定義。

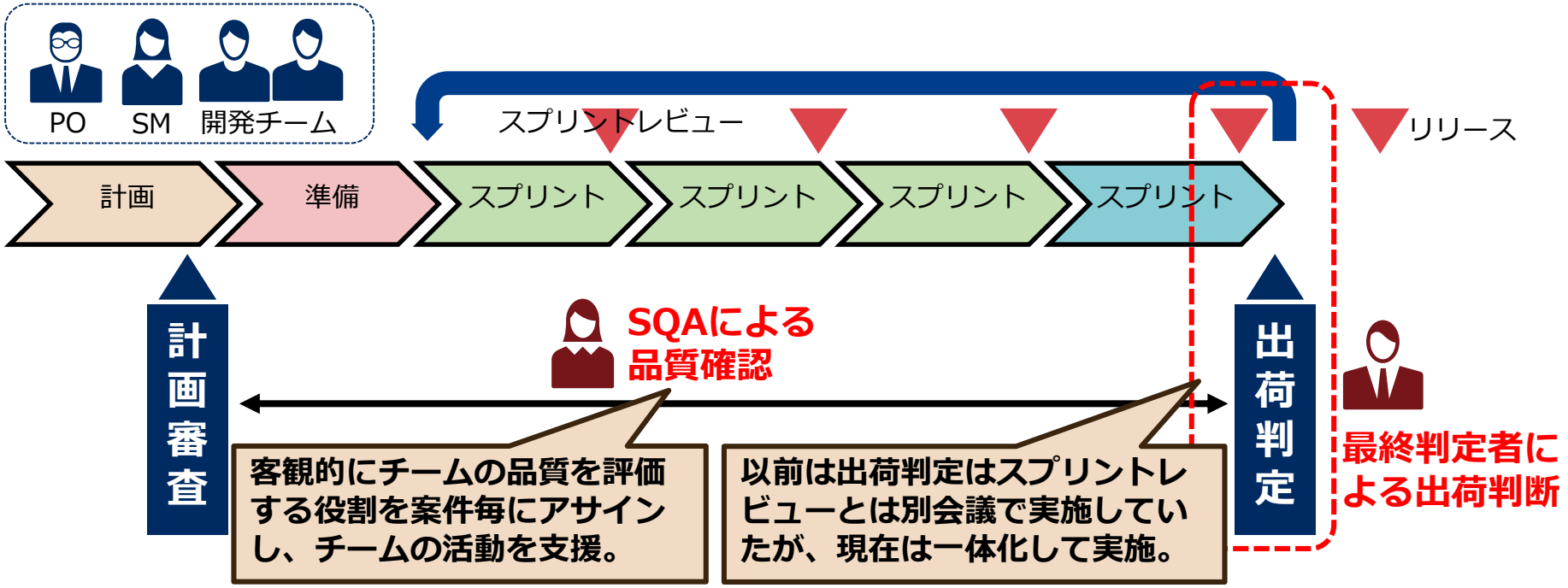
プロダクトの特徴による対応の定義

カテゴリ	開発プロセスの選択	組織による品質保証	審査形態	最終判定者
導入期	任意(アジャイル推奨)	対象外	不要	プロダクトオーナー
	プロダクト特性に応じた出荷基準を設定			
成長期	アジャイル	対象	会議による審査	組織の品質責任者
	要求される品質特性に応じた出荷目標基準の設定とフォロー			
普及展開期	アジャイル、またはウォータフォール	対象	会議または書類審査	組織長
	最高品質を達成する出荷目標基準とそのフォロー			

導入期・・・・機能の効果や妥当性の確認を目的とした開発。PoCなどの試作品やRPQ対応など。
成長期・・・・早期の機能開発による利用の拡大など、短いサイクルでリリースする場合を想定。
普及展開期・・既に普及しているプロダクトの派生開発など、安定稼働が要求されるもの。

審査プロセスと確認のポイント

複数回のスプリントでリリースするケース



審査	目的	確認内容
計画審査	計画の妥当性を確認し、開発着手の可否について会議にて審査	開発目的、内容、体制、日程、規模と品質の見積もり、リスク、課題
出荷判定	これまでのスプリントのレビュー結果を踏まえ、組織として製品のリリース合否を判定する	各スプリントレビューの結果、SQAによる品質確認の状況、組織の出荷判定基準の達成状況

アジャイル開発における品質保証の考え方

プロダクト品質中心の品質保証。動くソフトウェアによる確認が重要。

プロセス品質は、Doneの定義で達成状況を確認。必要条件だが、十分条件とはしない

- プロダクトの種別でDoneの定義の難易度は変わる
- できていて当たり前
- メトリクスを定義し、ツールなどによる計測、見える化を実現

必要に応じて中間成果物でも確認する

- 動くソフトウェアの確認は、外部特徴が中心
- プロダクト内部は、ソースコードの静的解析で確認するが、十分でないケースもある
- SQAが設計書などをサンプリングして確認

バグは基本0

- バグを発見した場合は、即修正
- 確実なバグ分析と1+n施策の実施
- デグレードは特に要注意

困っていること、今後の課題

他部門への品質保証の仕組みのスケールアウト

- 今回紹介したケースは、アジャイル開発をある程度理解している理想的な部門での事例。
- 依然ウォーターフォールでの品質保証が中心の部門もある。
- 開発チームの立ち上げ、育成と同様に、組織の品証部門の育成も必要。

統計的な分析の利用

- より多くの実績データの収集が必要。
- 数字じゃないと納得しない人もいる。
- 実績の母数が多くなれば、統計的な分析により、傾向や確からしさの基準みたいなものが見えてくるはず。

「価値」の評価

- 現状は当たり前品質を確認している。（どちらかというと従来の延長線上）
- アジャイル開発本来の「ビジネス価値」をどのように評価するか。

循環的複雑度を利用した品質定量評価の事例

私のアジャイル開発を行うまでの軌跡

- 2007～2011年

  電力会社向けSIを行うSEとしてミッションクリティカルなシステム開発

- 2011年

  事業部でサービスを立ち上げることになり、開発のまとめ役として参加

- 2014年

  スケールアウト可能なシステムへの再構築で、初のスクラム実施

- 2017年

  認定スクラムマスター（CSM）となり、スクラムのプラクティスを導入

- 2018年

  認定スクラムプロダクトオーナー（CSPO）となり、事業部に本格導入

今までのアジャイル体験

組織で初めてのアジャイル

- どこから始めるか？

アジャイル初心者が多いチームでのスクラム

- 必要スキルは？どうやって伝える？

スクラムがうまく行った事例、うまくいかなかった事例

- どんな特性が結果に影響する？

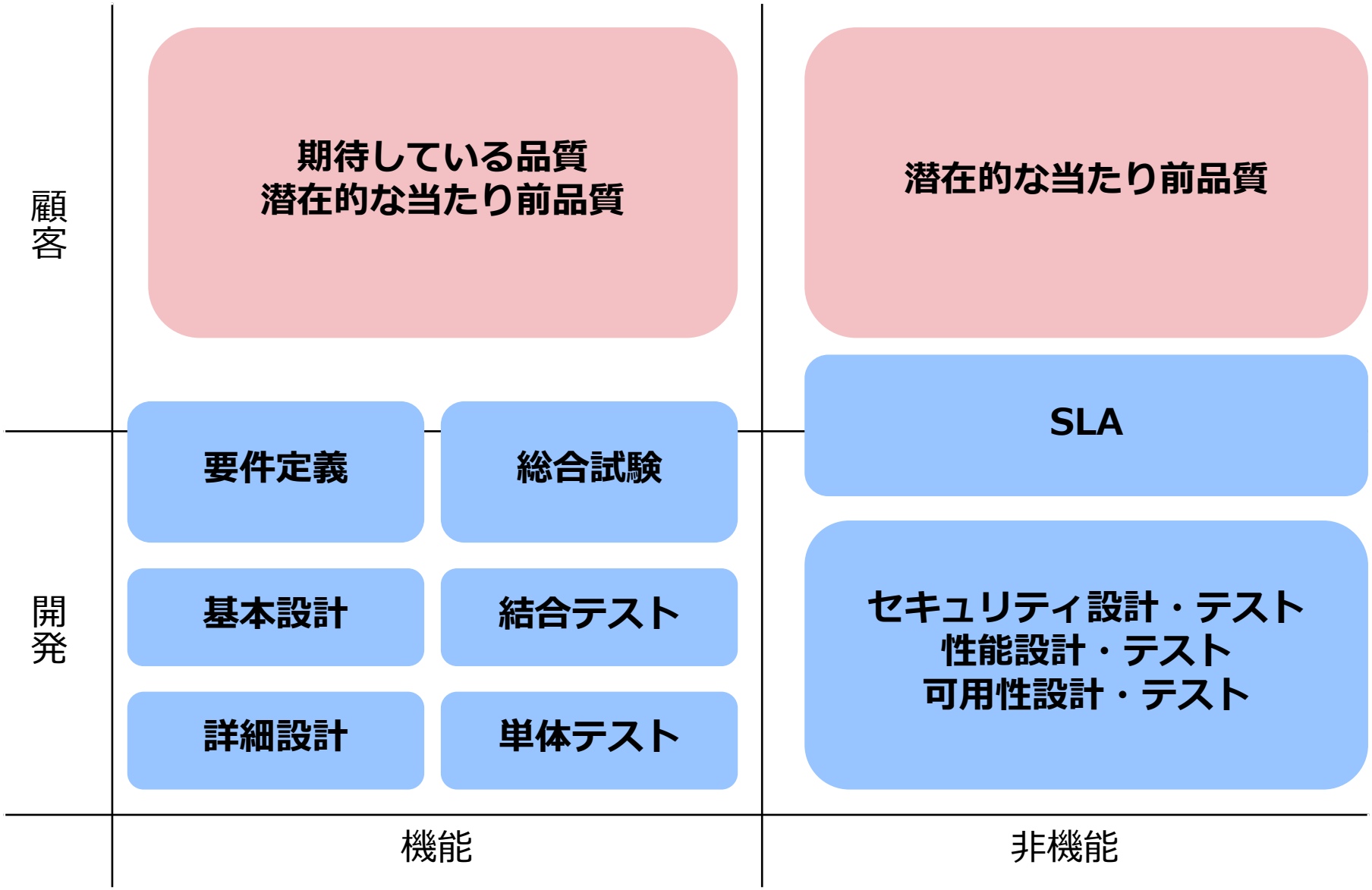
リモートチームでのスクラム

- 可能？不可能？効率悪い？どんなツールを使う？

複数チームでのスクラム

- どこから始める？どこが難しい？

私が考える品質とは？



品質を支える取り組み

無駄を徹底的に減らしていく

- 顧客の理解に力を使うため、その他の作業は最大限に効率化させる

各種テスト	→	**ほぼすべてテストコード化** 単体・結合・総合・負荷など
設計書	→	**設計は大事だが書類はいらない** 一時的なメモのみを作成 ドキュメントになるソースを作る
仕様書	→	**できるところは自動化** 画面のハードコピー、入力項目をコードから摘出

自動化を中心に今までとは違う取り組みを進め、改善させていく

第三者による品質管理では数字で示せる指標が求められる・・・

循環的複雑度を使った定量評価

循環的複雑度とは？

プログラムの分岐数を数値化し、プログラムの複雑度を定量化する方法

```
if method == "agile":
    success = True
else:
    success = False
```

1 + 1(ifの数) = 2

指標１　循環的複雑度が、各関数内で、10以下であること

- プログラム自身の複雑度を下げていく効果あり
- 変化させやすいプログラムが作られる可能性があがる

指標２　循環的複雑度の数の合計値より、テスト数が多いこと

- テストを減らしたいという気持ちが指標１以上にプログラムをシンプルにさせる効果
- ライン数によるテスト密度より正確で、カバレッジより曖昧な基準

まとめ

顧客が考える品質のために時間を最大限使う

- 逆に自分たちが管理できることには最大限、効率化を検討する

循環的複雑度を使った定量評価の取り組み

- 複数のチームで実施した結果、プログラム自身の品質が上がり、全体のバグ数は激減
- スキルが高くないと難易度高めの指標・・・・

指標はチームで決める

- 何がプログラムの品質に貢献するかは、チームのスキルにも依存

組織として品質を一定にするには工夫が必要

- 各個人のスキルの統一
- 各技術の統一
- **継続的な改善を行う仕組み**


Orchestrating a brighter world
NEC