

モブ・プログラミングを用いた アジャイル開発プロセスの1事例

サイボウズ株式会社
QA Community
永田 敦

モブ・プログラミングによるプロセスに対する効用

プロセスによるモブ・プログラミングに対する効用

SPI JAPAN 2020

この発表で伝えたいこと

- アジャイルにおいても、しっかりしたプロセスが育まれる
 - ”プロセスやツールよりも個人と対話を”ですが
- 特に計画のプロセスを大切にしている
 - “計画に従うことよりも変化への対応を“ですが
- そのプロセスで生まれるドキュメントの質は、モブプログラミングに影響する
 - バックログの品質が悪ければ、モブは混乱する
 - “包括的なドキュメントよりも動くソフトウェアを“ですが
- それにモブを用いることで、プロセスの成果物の質が初めから組み込まれるようになる

モブ・プログラミング

目的

- 複雑な、難しい問題を解決していく
 - 多様性を持ったチームが互いに創発して、問題を解決していく
 - チャレンジし、失敗し、成功する中で互いに学ぶ
 - 品質をチームで作ricoむ
- チームの育成に効果的なプラクティス

[4] パターンが見せるモブプログラミングの魅力と効果, 2020, Atsushi Nagata,
https://www.slideshare.net/atsushinagata2/effects-of-mob-programming-pattern?qid=fe4fdbf2-9f21-46fb-ad2c-b55ed6df14e9&v=&b=&from_search=1

モブの構成

サイボウズの場合

- 3名から5名程度の開発チーム
- ロール
 - ドライバ 1名
 - ナビゲータ
- チームで一つのバックログに対応
 - 同時に複数のバックログは対応しない
 - 1個流し
 - バックログは原則としてひとつずつDoneしていく
- リモート開発：Zoomを使う
 - ドライバの画面を共有する
 - バックログ、仕様書、ソースコード、テスト設計書を共有

一般で言われているような、一つのマシンを複数人で使う形ではない

そのやり取りを分析をする

モブ・プログラミングの 開発活動を分析する

プログラミング

複雑なシステム問題を
解決する認知活動

モブ・プログラミング

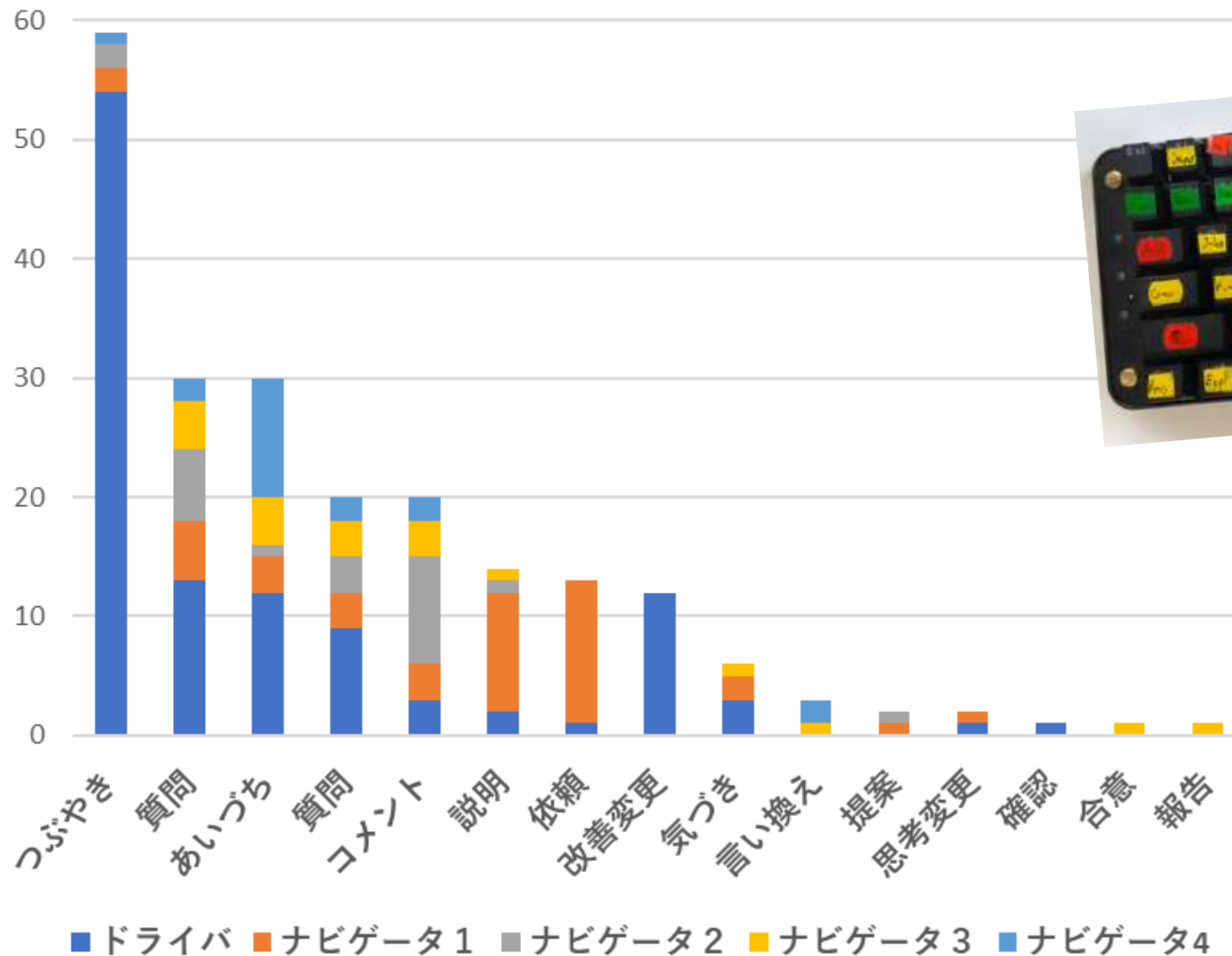
協調的な認知活動

チーム認知

チーム一体となって、状況を把握し、計画を立て、問題を解決し、設計し、判断する活動

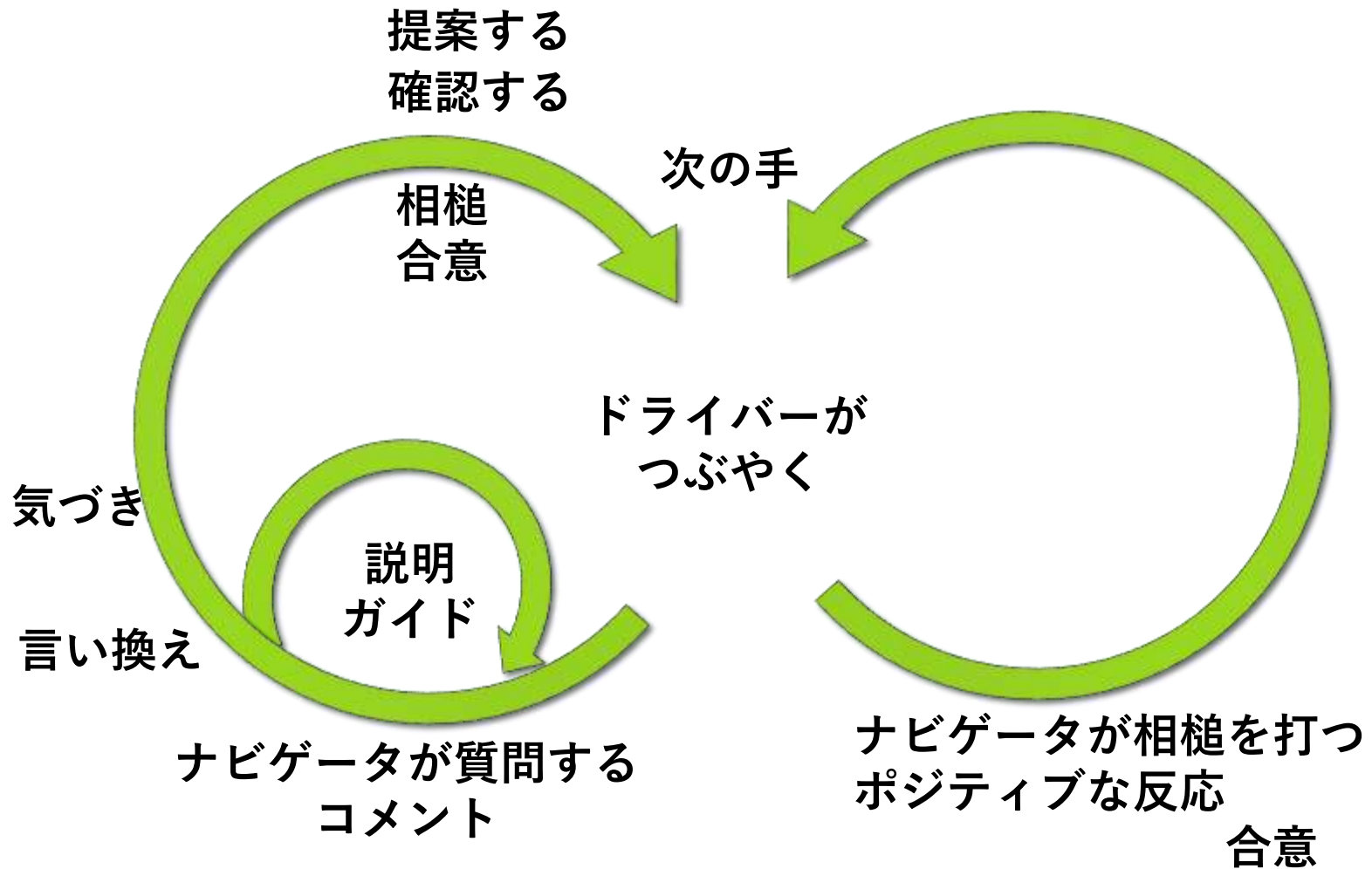
Kiekel, P. A., & Cooke, N. J. (2005). Human Factors Aspects of Team Cognition. In R. W. Proctor & K.-P. L. Vu (Eds.), Handbook of human factors in web design (p. 90–103). Lawrence Erlbaum Associates Publishers.

チーム認知活動；プロトコル分析

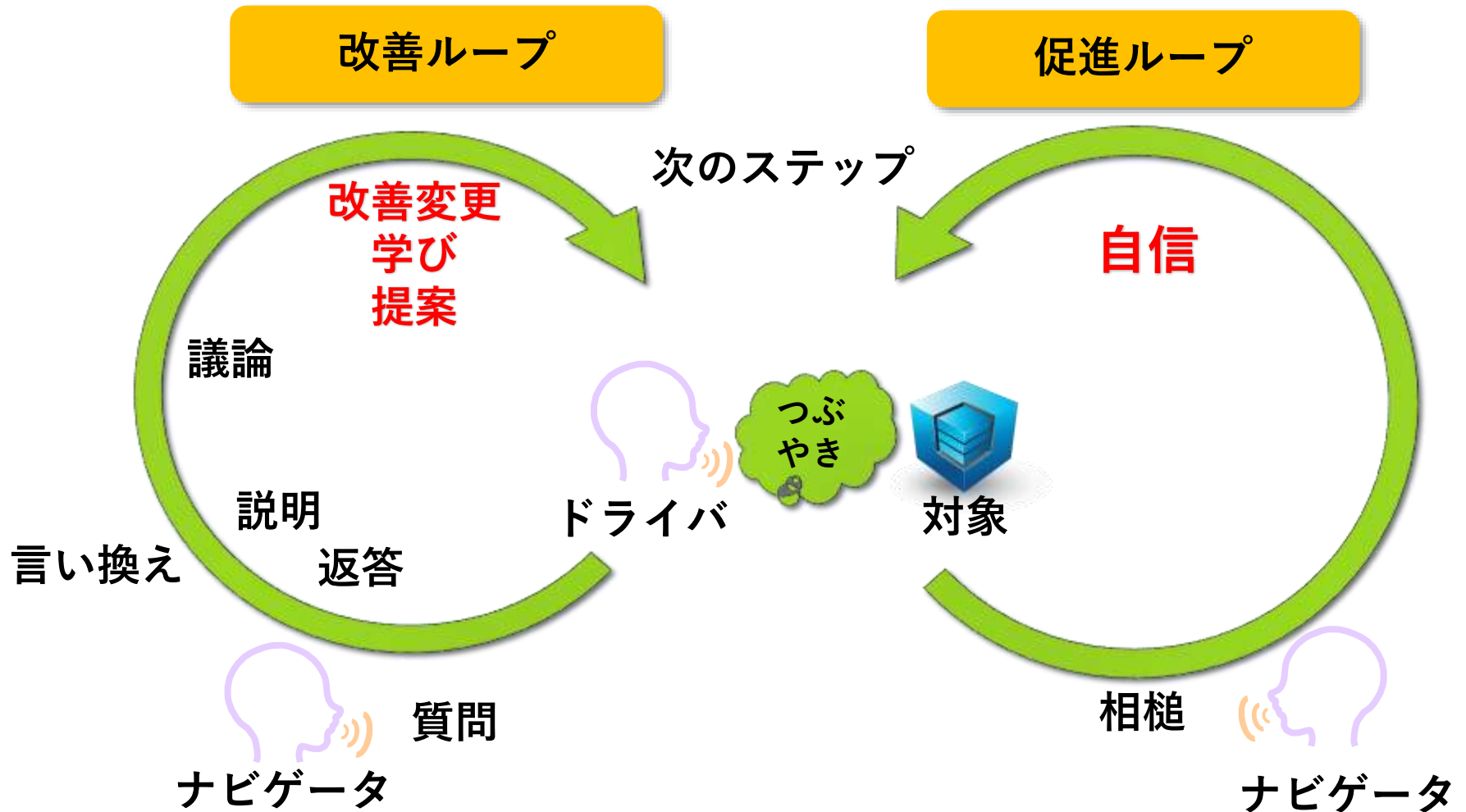


時間長 39:14 ， イベント数 214

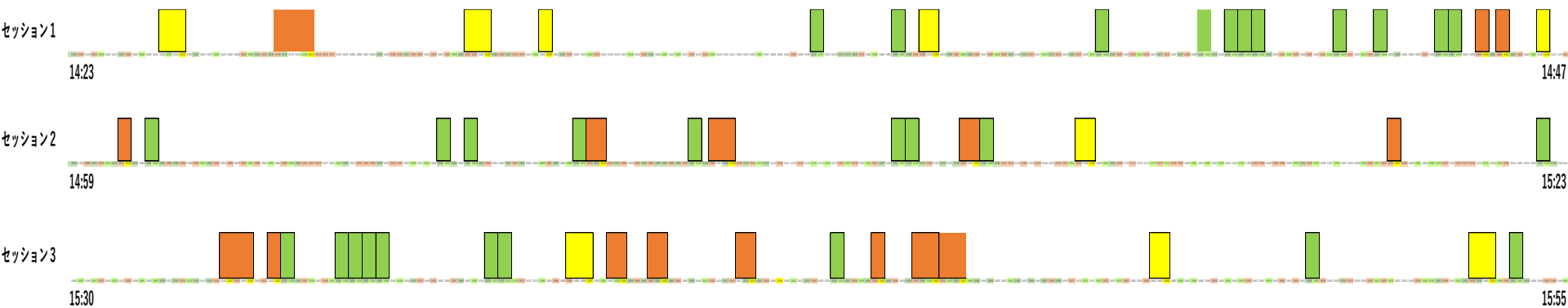
モブのフィードバックループ



モブ・メンタルモデル



モブのやり取り



合意
“いいね”

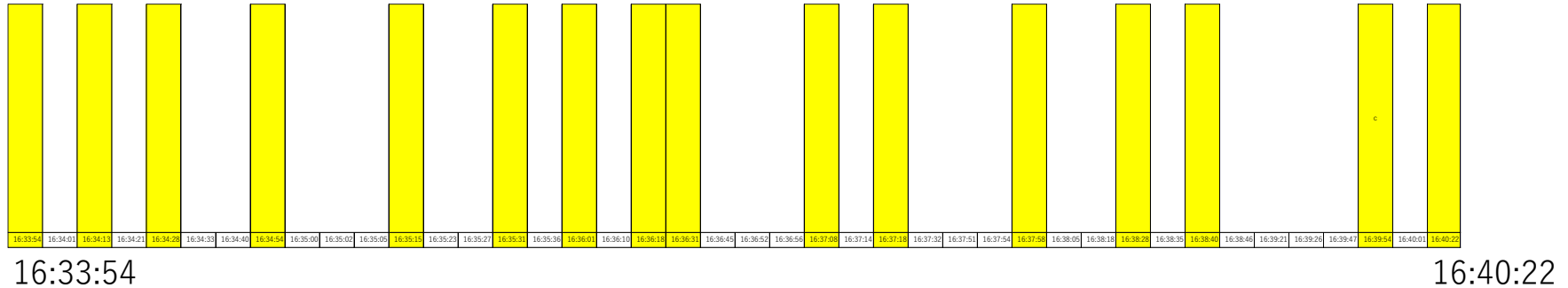


改善
”いいので
変えたよ”



学習
“いいので
学んだよ”

事例：コメントの投げ合い



コメントを投げ合って、モブが膠着した

モブの参加者それぞれが憶測でこうあるべきという意見が続出し
しかも、互いに納得しないために膠着した

バックログの品質に問題
このバックログでの
目的、スコープ、前提条件、制約事項、終了条件が不明確

チーム認知活動の前提条件

チーム認知の観点から、私たちの議論は、問題が何であることを共有して理解しなければ、チームは間違っただけでなく、リソースを十分に活用することができないということである。

Fiore, S. M., & Schooler, J. W. (2004). Process mapping and shared cognition: Teamwork and the development of shared problem models

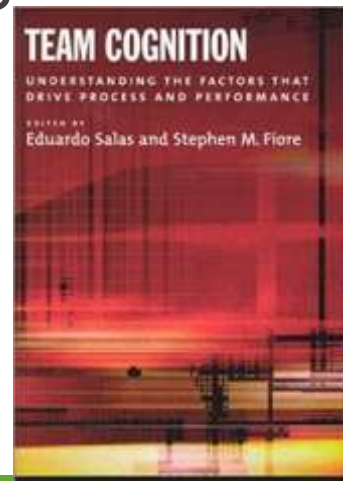
PO(プロダクトオーナー)とチームはバックログの品質を継続して改善している

チーム (含QA)はバックログの内容を理解し共有している

チーム (含QA)はリスクの識別、分析をし共有している

チーム (含QA)は仕様を作成しPOと共有している

QAはテスト設計を行い、チーム内で共有している

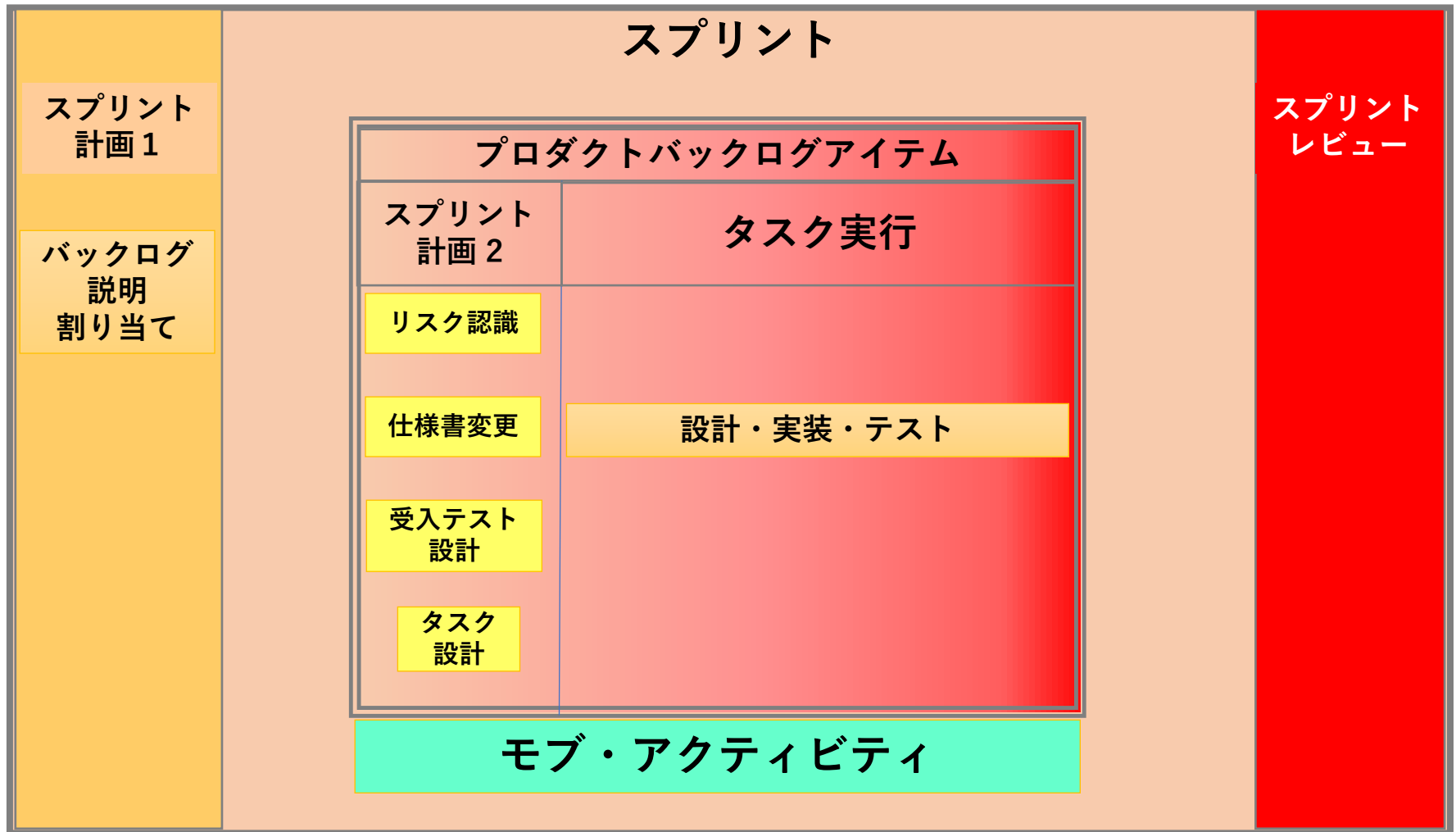


アジャイルプロセスと チーム認知活動

チーム認知の活動（モブ・プログラミング）と、アジャイル開発のプロセスはどのように影響を与え合っているか

以下に、事例としてデータを示す

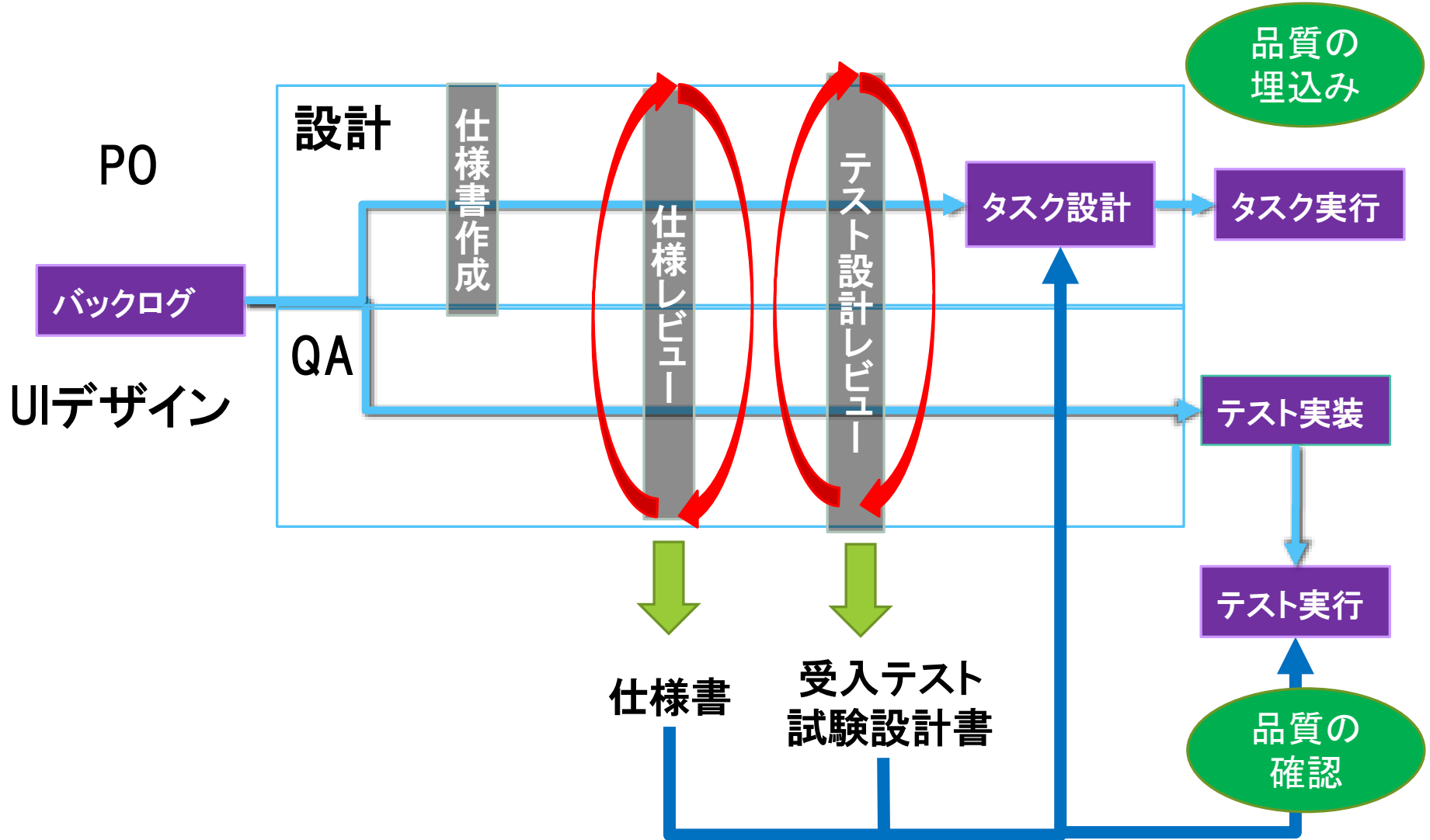
開発プロセス：LeSSフレームワーク



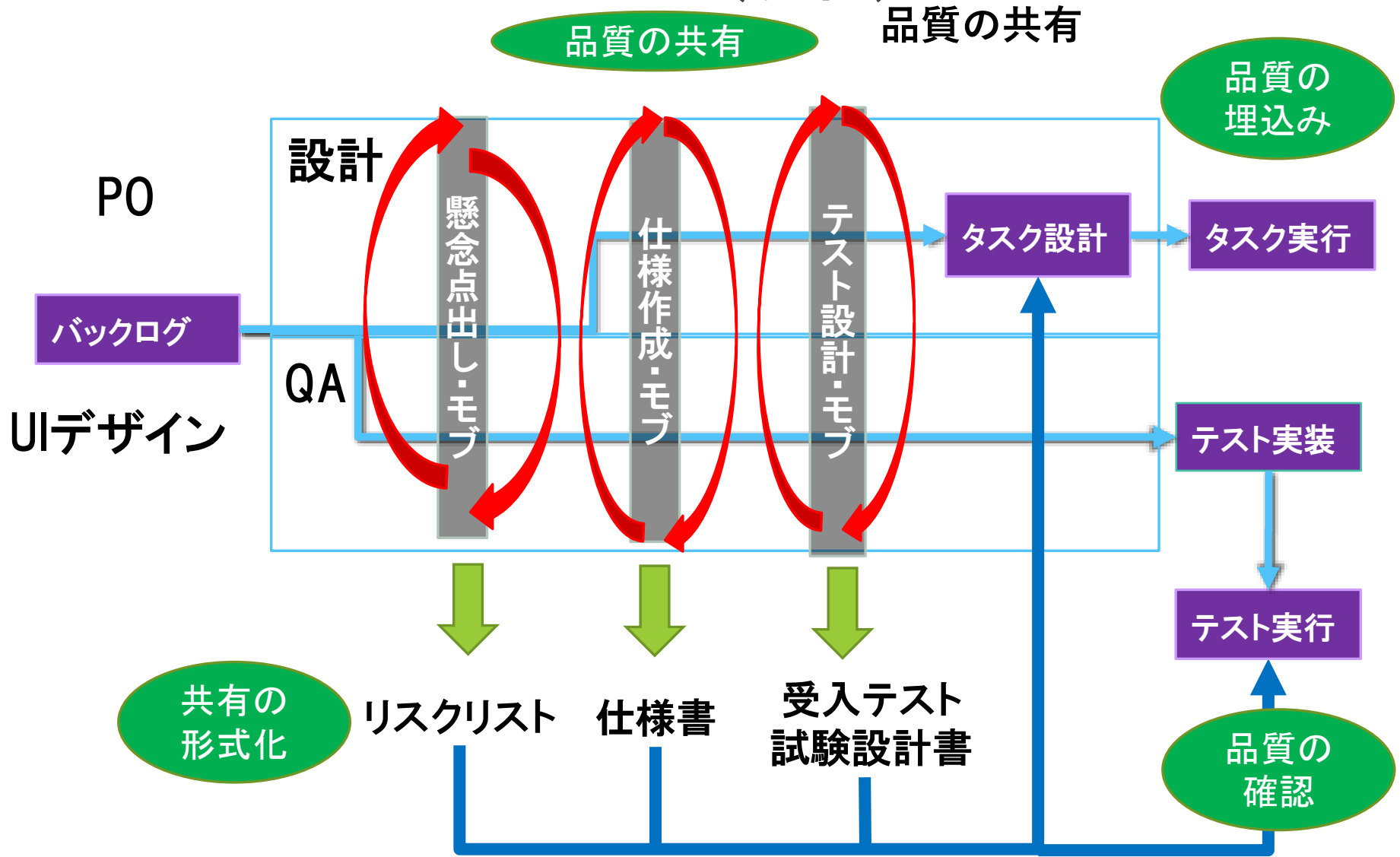
LeSS : Large Scale Scrum <https://less.works/>

PBI
Done

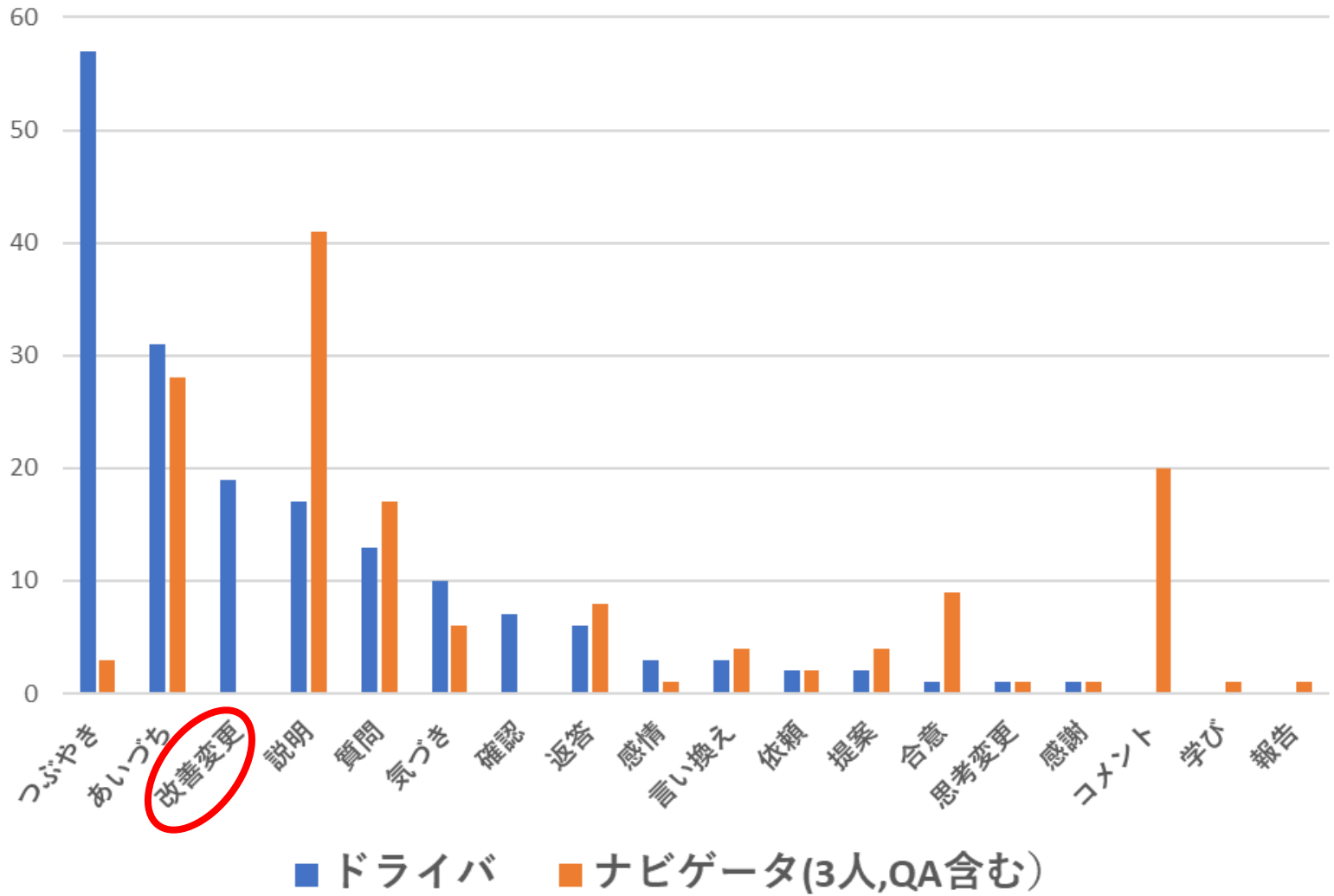
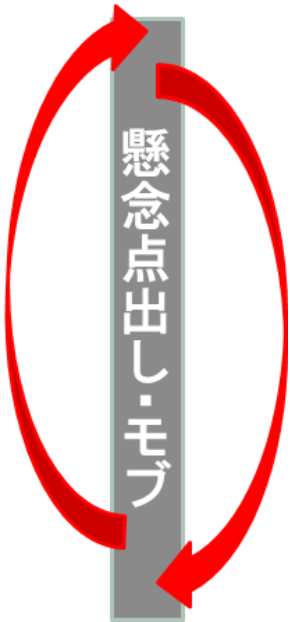
プランニング 2 (2018年初頭)



プランニング2 (現在)



懸念出しモブ



Duration : 54:41

改善変更 : 19 件

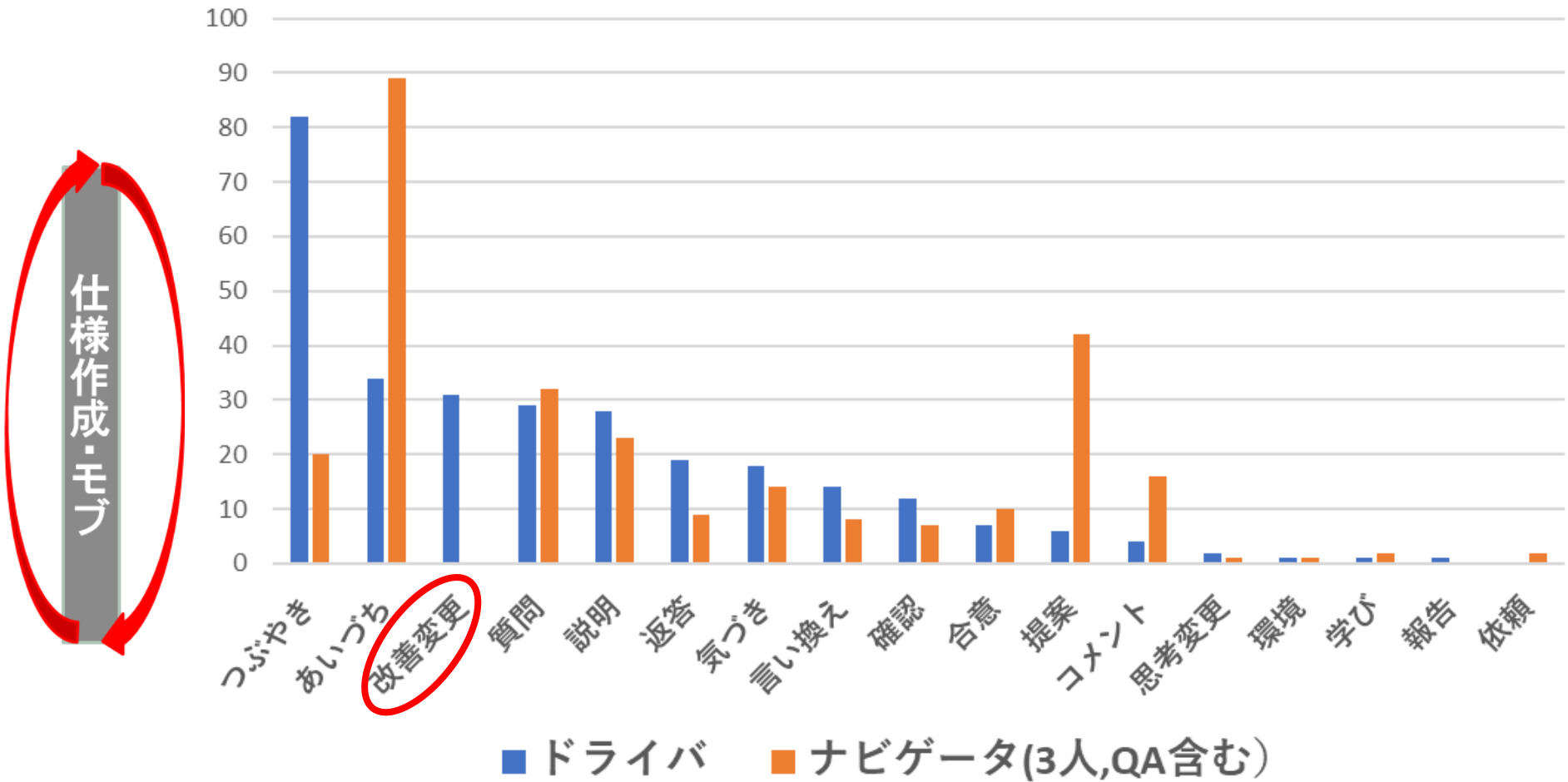
懸念出しモブ

- 懸念点出し・モブで、リスクを開発者、QAで整理する（リスク識別、分析、対応計画）
 - リスクリストの作成：カテゴリー
 - 要求（バックログ）、仕様
 - 実装（プログラム構造、影響箇所）
 - テスト（機能、非機能）



- **改善変更により、リスク記述の質が改善される**
 - 明確になることにより、対応を優先度をつけて立てる
 - 要求の不明点や仕様の見解をPOに聞く⇒仕様書作成
 - 不明なところを解決するタスクを積む⇒タスク設計
 - 必要なテストを検討する⇒QA⇒テスト設計
 - 非機能要求の考慮⇒仕様書作成、テスト設計

仕様書作成モブ



Duration : 1:16:28

改善変更 : 31 件

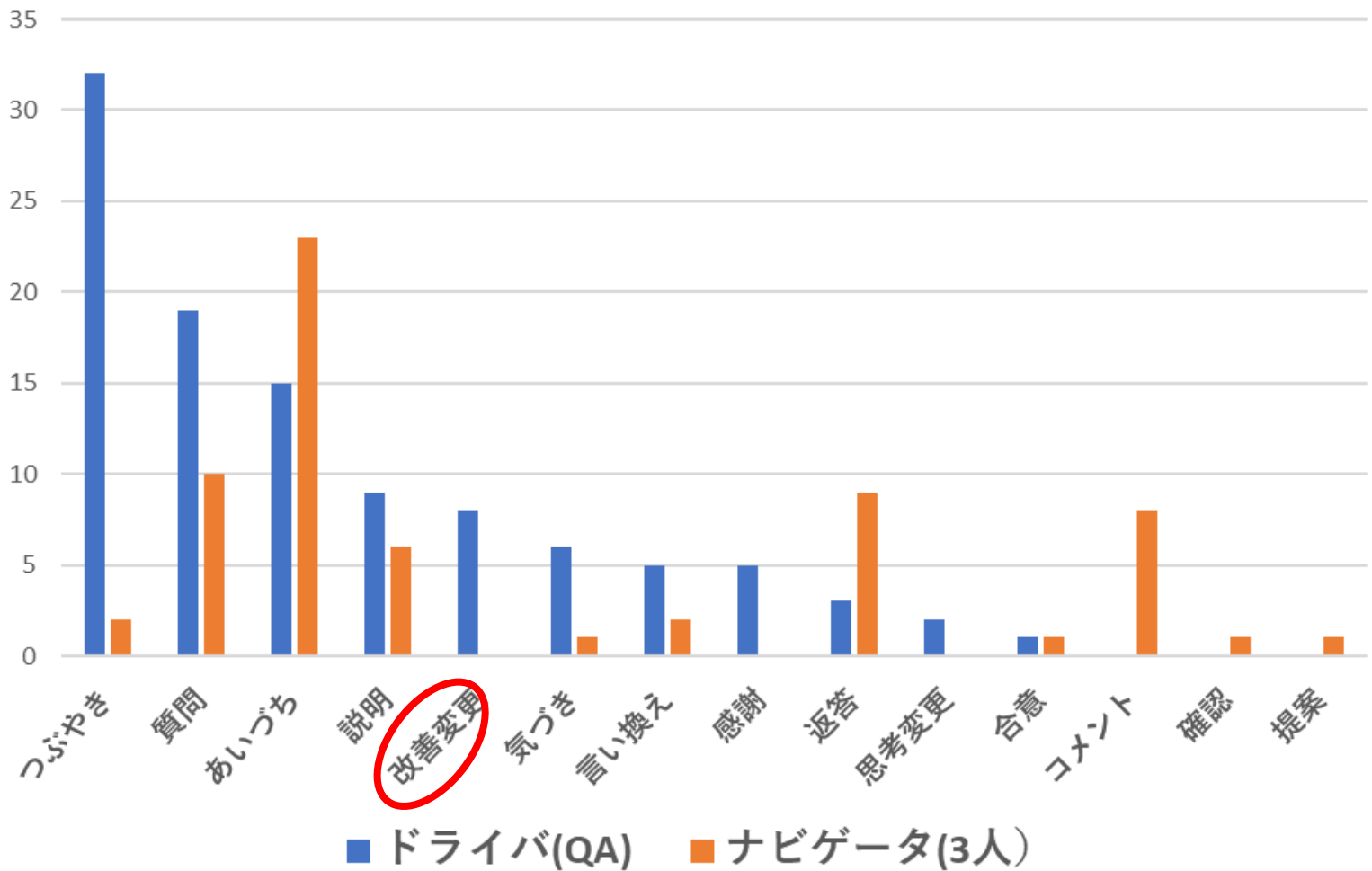
仕様書作成モブ

- 仕様書作成・モブで、仕様書の修正、追加、削除を開発者、QAが行う
 - 要求に対する仕様（システムの振る舞い）を特定する
 - 開発のゴールが特定する⇒開発者
 - テストベースが確定する⇒QA



- 改善変更により、仕様書の記述の質が改善される
 - 設計・実装の品質が上がる
 - 何をどのように作るかに集中できる⇒タスク設計
 - エラー処理、例外処理、非機能要求の漏れがなくなる
 - この後続くバックログ開発の質の高いリファレンスとなる

受入テスト設計モブ



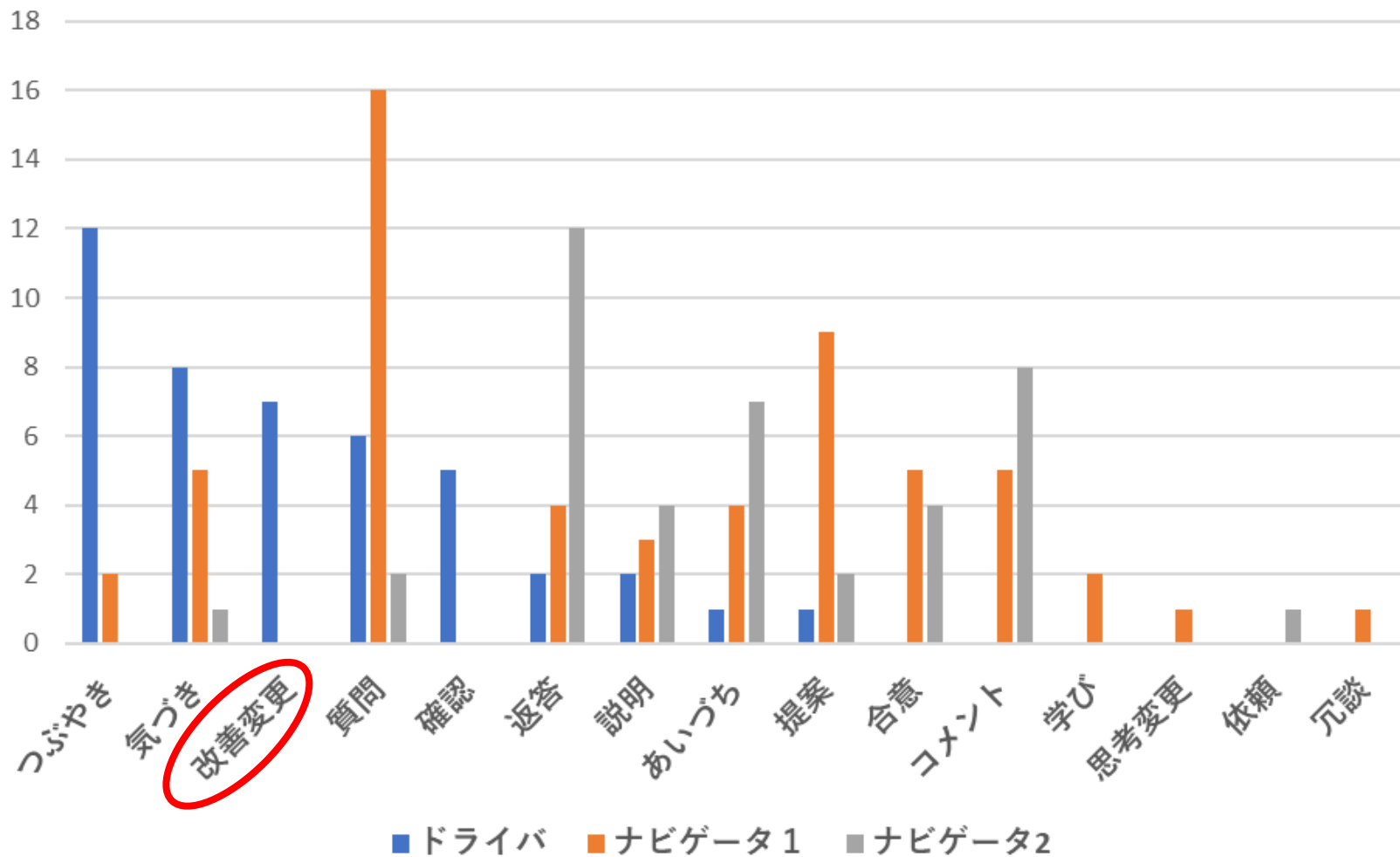
Duration : 33:01
改善変更 : 8 件

受入テスト設計モブ

- QAがドライバ、開発者はナビゲータになって、受入テストを設計する。
 - バックログの受け入れ条件をもとにテスト設計をする
 - テスト設計から、テストの意図を開発者に伝える
 - 開発者は、テストの意図から品質の気づきを得る←相槌、気づき
- どのテストをだれがやるか、手動テスト、自動テストの議論をする



- 開発者からのフィードバックにより、テスト設計が改善変更される
 - QAは開発者が気づいたテストの観点を得る
 - 開発者から、ソフトウェアの構造やその変化についての情報を得ることにより、QAはリスクベーステスト設計を考慮する



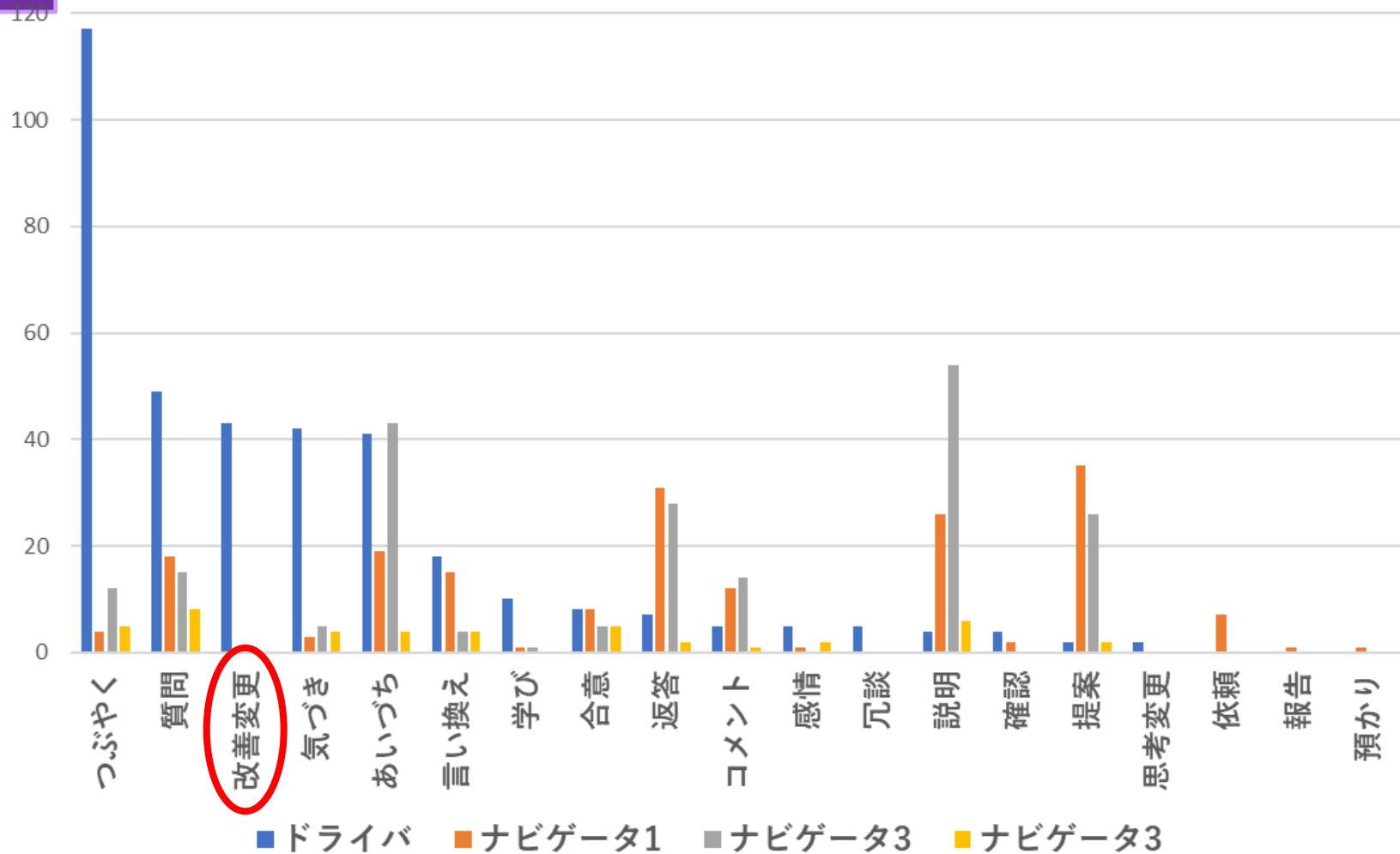
Duration : 15:24
改善変更 : 7 件

タスク設計モブ（タスク分割）

- バックログ、リスク記述、仕様書、テスト設計書をもとに、開発で行うべきタスクを洗い出す。
 - 具体的な設計・実装タスクの策定
 - 定番のタスクはテンプレートをベースに取捨選択する
 - テンプレートをツール(Kintone)に読み込みカンバンで管理する



- モブの多様性の特性から、コメントや質問が促され、改善変更が起こり
 - 隠れたタスクの漏れを防ぐ
 - タスクの目的やスコープがより明確になる
 - タスク実行の実行において、モブ・プログラミングのチーム認知活動が、より活性化し、問題の解決を促進する



Duration : 56:34
 改善変更 : 43 件
 イベント数 : 796

タスク実行モブ

- 改善変更：“提案”、“説明”、“コメント”によるのコードの変更
- 言い換えや学び“ ⇒ チームどうしの学習
- うなづきによる肯定的なフィードバック



活発なチーム認知活動

“モブ感すごいですね、今回のバックログ”

まとめ

- アジャイルにおいても、しっかりしたプロセスが育まれる
- 特に計画のプロセスはプロセスの中心的な役割を持っている
- そのプロセスで生まれるドキュメントの質は、モブプログラミングに影響する
- そのプロセス自身にもブの活動を用いることで、プロセスの成果物の品質改善が初めから組み込まれるようになる

ご清聴ありがとうございました

サイボウズ
QA Community
永田 敦