



<https://ja-jp.facebook.com/nasaearth/>

アジャイル・クオリティの探求

JaSST Niigata 2021

サイボウズ株式会社 アジャイル・クオリティ 永田 敦

永田 敦

サイボウズ株式会社

開発本部

アジャイル・クオリティ

アジャイルの品質保証サポート

JSTQB Advanced Level Test Manager
Agile Inspection Maestro

SQiP 研究会 研究コース 4
派生開発推進協議会運営委員

Agile流派 Evolutionary(EVO)



本日のアジェンダ

1. アジャイル・クオリティ
2. 品質って何だろう
3. 品質保証とは
4. アジャイル・プロセスにおけるサポート
5. アジャイル・ドキュメント
6. アジャイル・クオリティ・モデル
7. さらなる広がり
8. まとめ

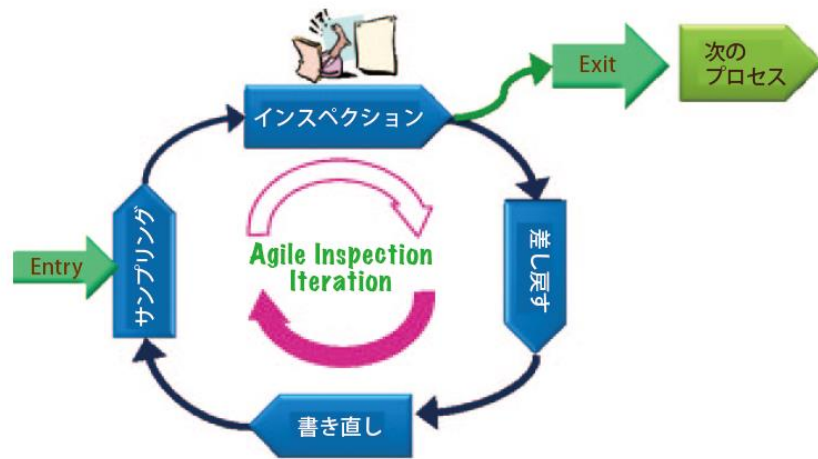
アジャイル・クオリティ

アジャイルの理解が深まるにつれて
これは、品質にとって有利だと思った

いろいろなプラクティスをやってみた

- アジャイル・インスペクション
- アジャイルRCA
- DevQA

アジャイル・インスペクション



レビューをアジャイルに行う手法

2010

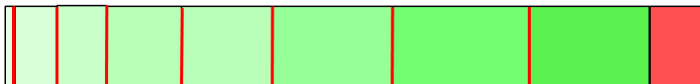
- ドキュメント全体の欠陥の閾値の目標を決めておく
- ドキュメントをサンプリングし、
- レビューして、フィードバックを返す
- また、欠陥を測定し、ドキュメント全体の欠陥を推定する。
- それを欠陥の閾値目標をクリアするまで、イテレーティブに繰り返す

バッチ的レビュー

数百ページのSRS

インクリメンタルなレビュー

初めの1ページからレビューを始めてしまう



- 1ページからインクリメンタルにレビューでき、その学びでドキュメントの質と書き手の技量が改善される
- 書き手にフィードバックすることにより、ドキュメントの品質を改善する効果がある。
- レビュー側もフィードバックの質が向上する効果が期待できる。

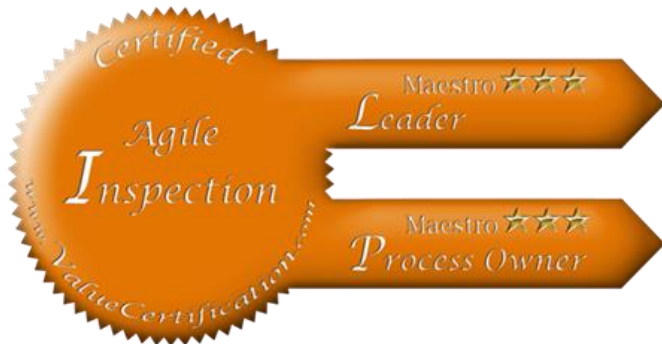
Tom Gilbさん



Agileの 生みの親の親

イテレーティブに行く

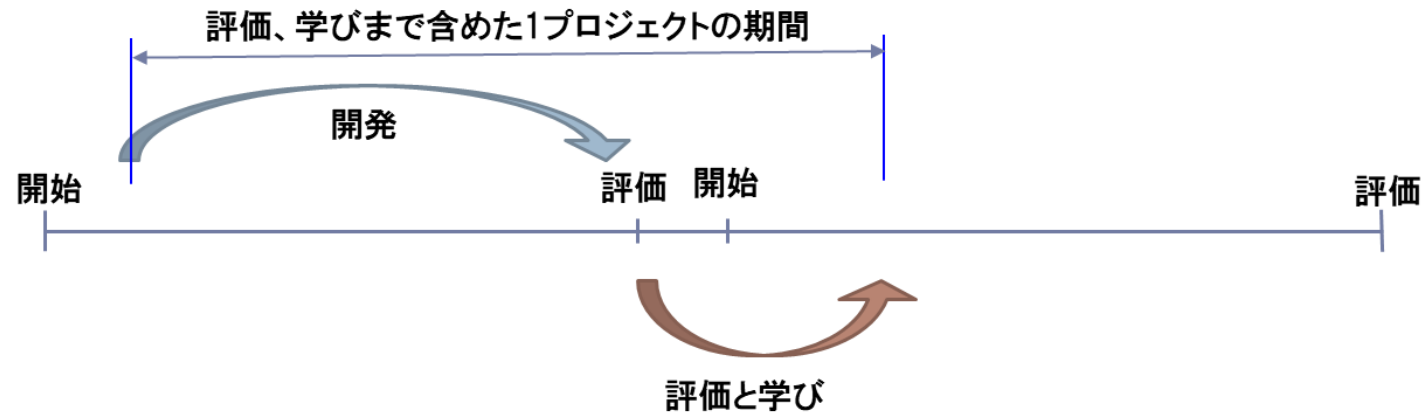
- 小さくレビュー
- 書き手にフィードバックする
- 測定し品質を判断する
- 改善をしてもらう



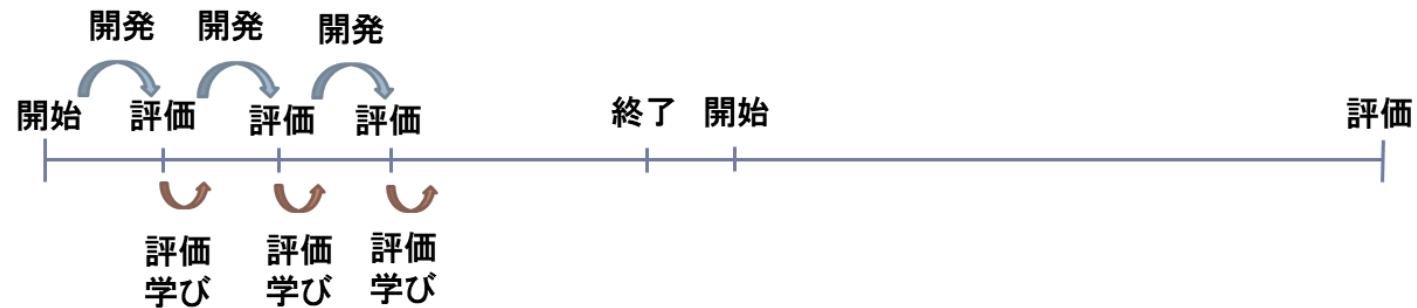
アジャイルなレビューにより早く**学び、書き手とドキュメントを改善する**

アジャイルの特長

ウォーターフォール

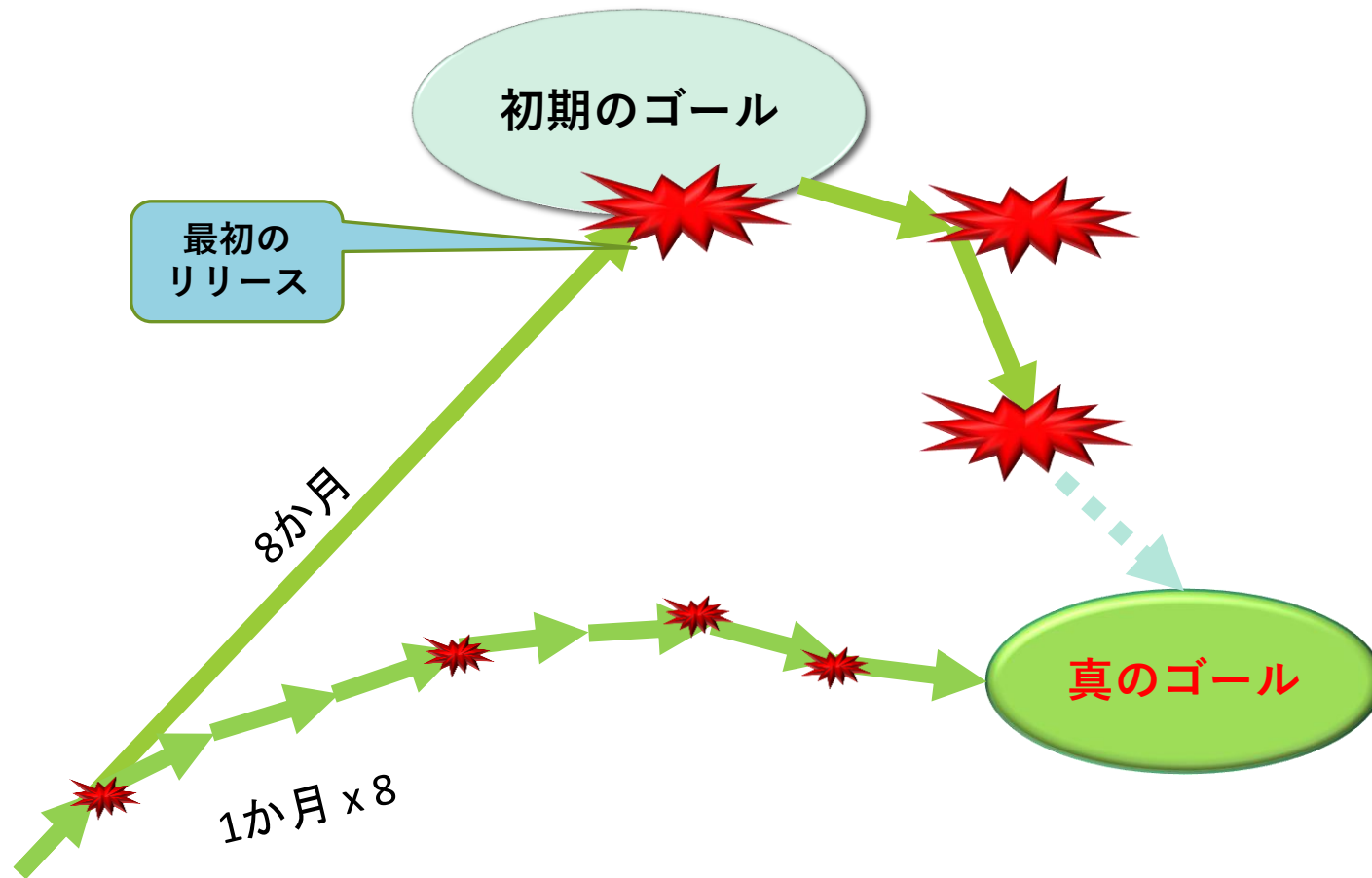


アジャイル

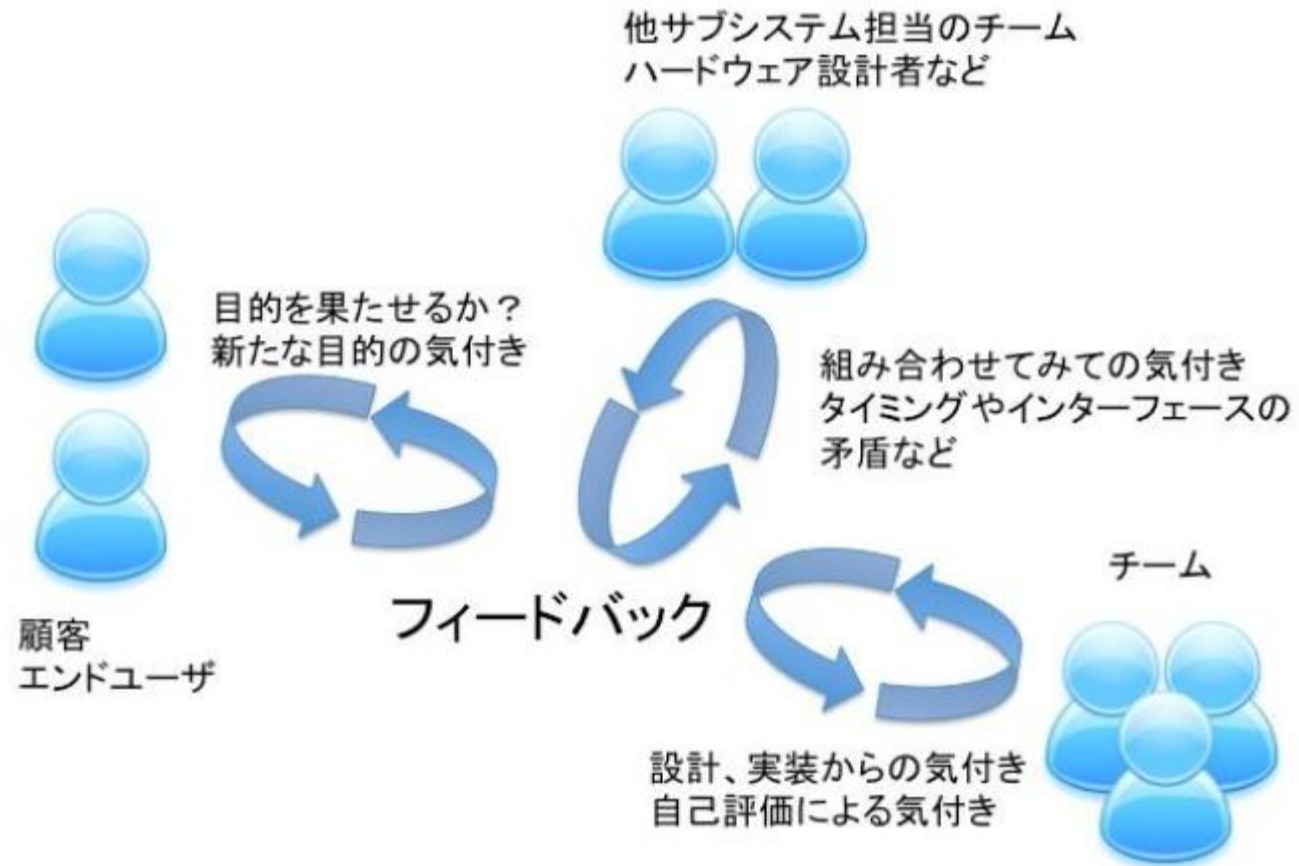


早く、そして小さく、イテレーティブに評価しそこから学ぶ

アジャイルは早い？

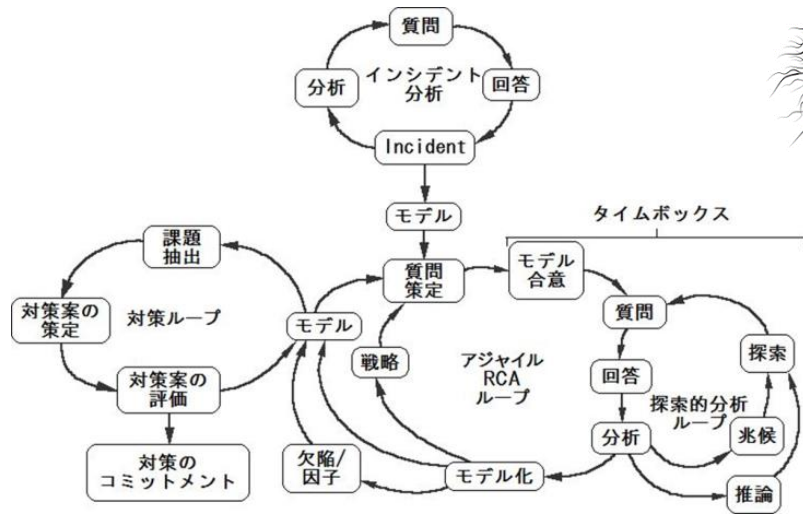


フィードバック獲得の設計



三菱電機 細谷泰夫: 斥候としてのアジャイルプロセス活用の提案 : SPI Japan 2012

アジャイルRCA

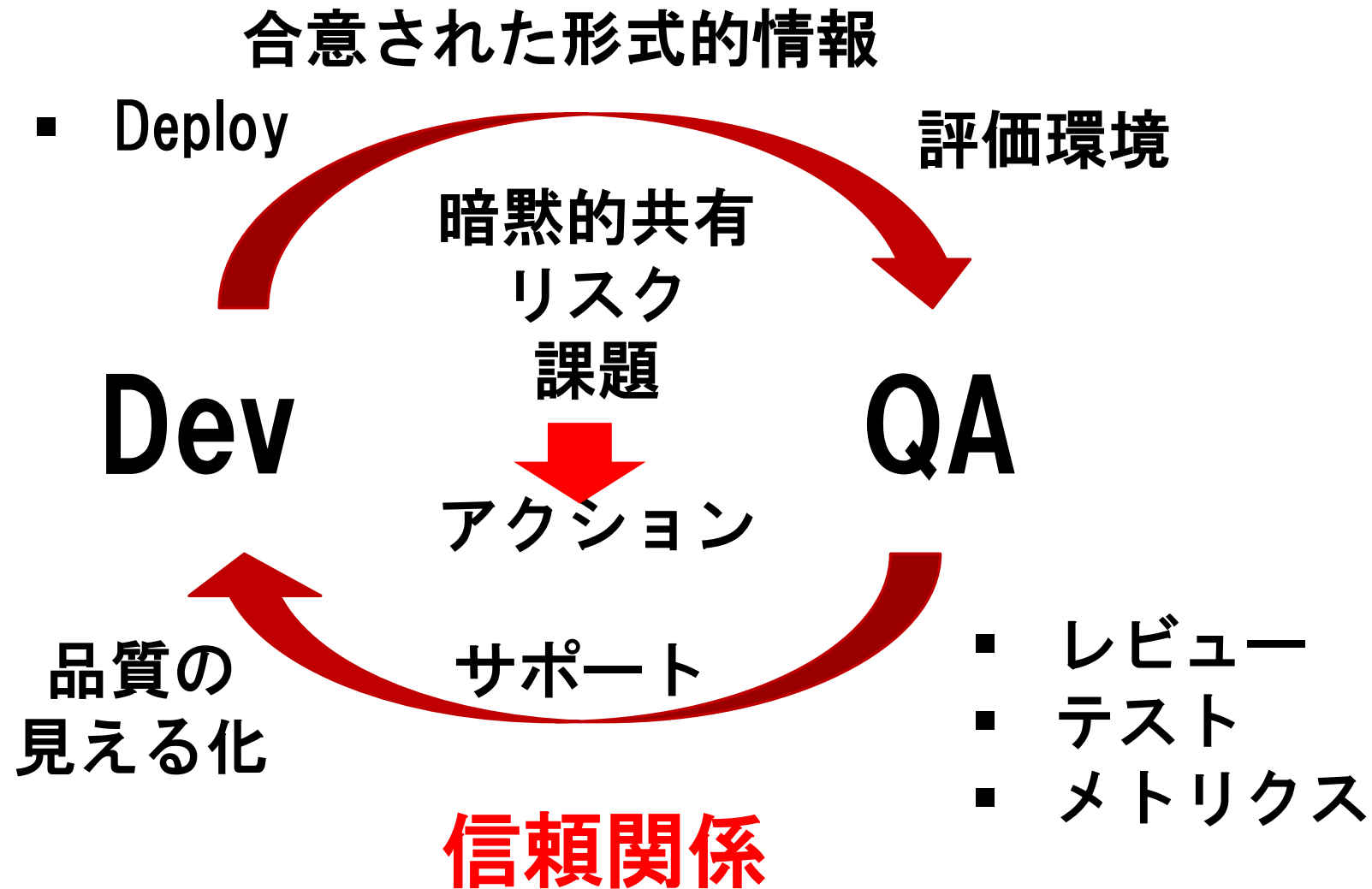


アジャイルで要因分析をする

2015

分析探索のインタビューループと、欠陥モデルを作るループの二つのフィードバックループをイテレーティブに繰り返して、モデルを成長させ、複雑な要因による障害の見える化を行う

- インクリメンタルに、モデルを成長させ**学びながら分析する**



品質って何だろう

品質・クオリティは誰かにとっての価値

品質は誰かにとっての価値である。

誰か＝使っていただいているお客様

[価値]

- ・ サービスを使って満足している、助かっている
- ・ その製品を欲しいと思う
- ・ 今後も使っていこうと思っている
- ・ もうこれなしでは仕事はできないと感じている
- ・ 効果の実績が出ている



その[価値]の対価として、お金をいただいている

ローレックスはなぜ売れる



<https://housekihiroba.jp/shop/c/c01rxnw01/>



<https://www.casio.com/jp/watches/casio/product.LTP-1129AA-7B/>

人は、機能だけで価値を決めていない

1000万円を超えるトゥールビヨン

<https://www.jackroad.co.jp/blog/post/tourbillon>





さて、
これは、いくら値を付けます？

私たちが売っているものは、
ソフトウェアという形がないロジックにより提供されるサービスの“価値”です。

皆さんは、その[**価値**]の対価として、お金をいただいてる、プロフェッショナルです

お客様にとっての価値＝品質

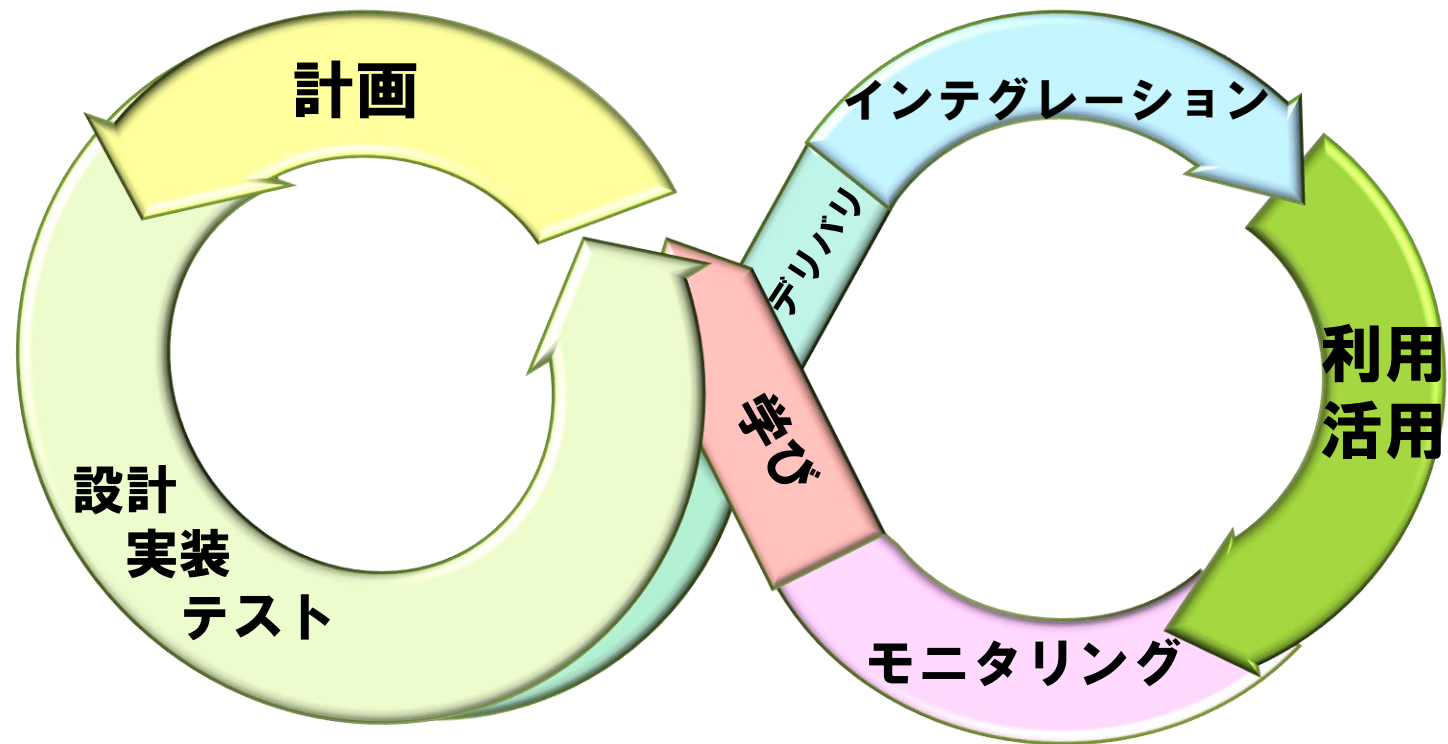
品質が良い悪いという評価は
誰ができるのでしょうか？

お客様

では、どんどんデリバリして、お客様に使ってもらえば
品質がいいかわかるじゃないか
悪ければ、すぐ直してすぐデリバリすればいい

DevOps

開発とオペレーションのフリクションをなくし
CI/CDのプロセスを自動化



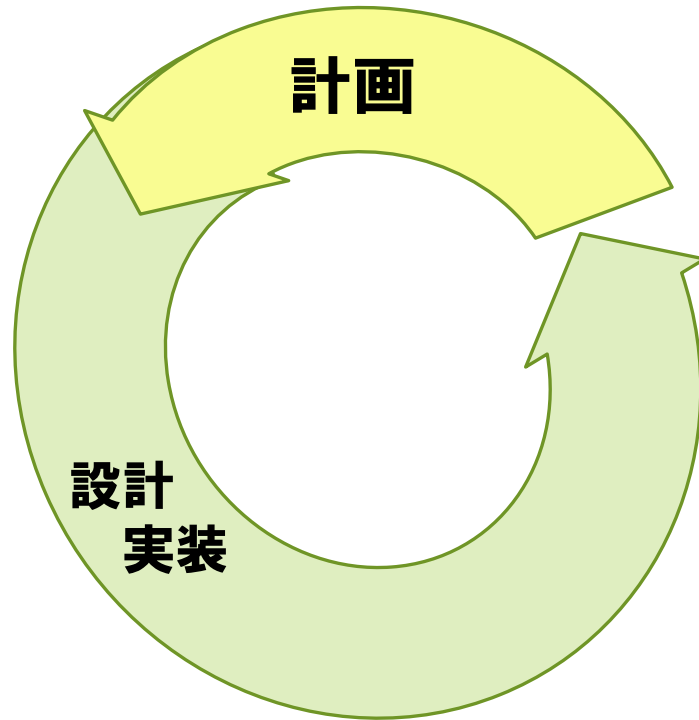
価値の創造と流れ：バリューチェーン

- どんな価値を提供すればよいのか



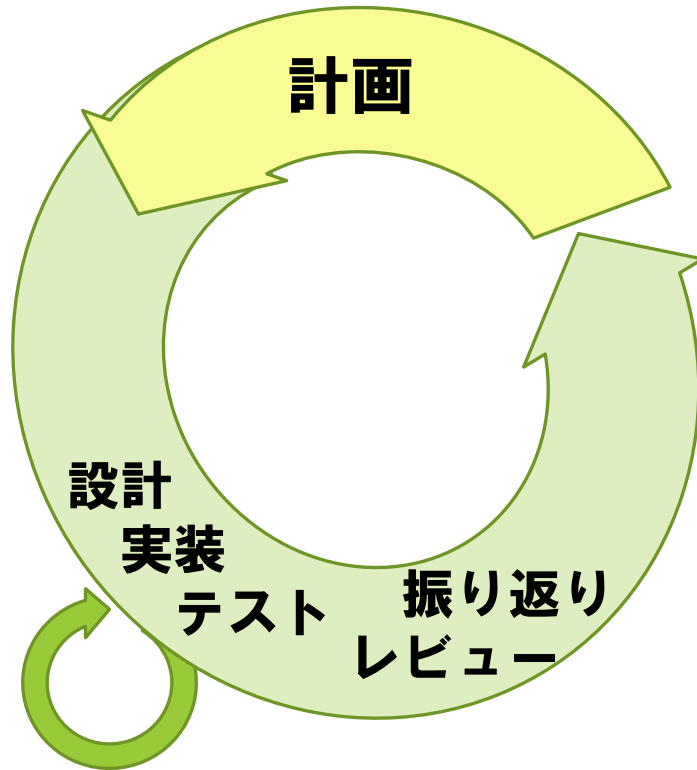
価値の創造と流れ：バリューストリーム

- どのようにして
その価値を
作り込むのか



価値の創造と流れ：バリューストリーム

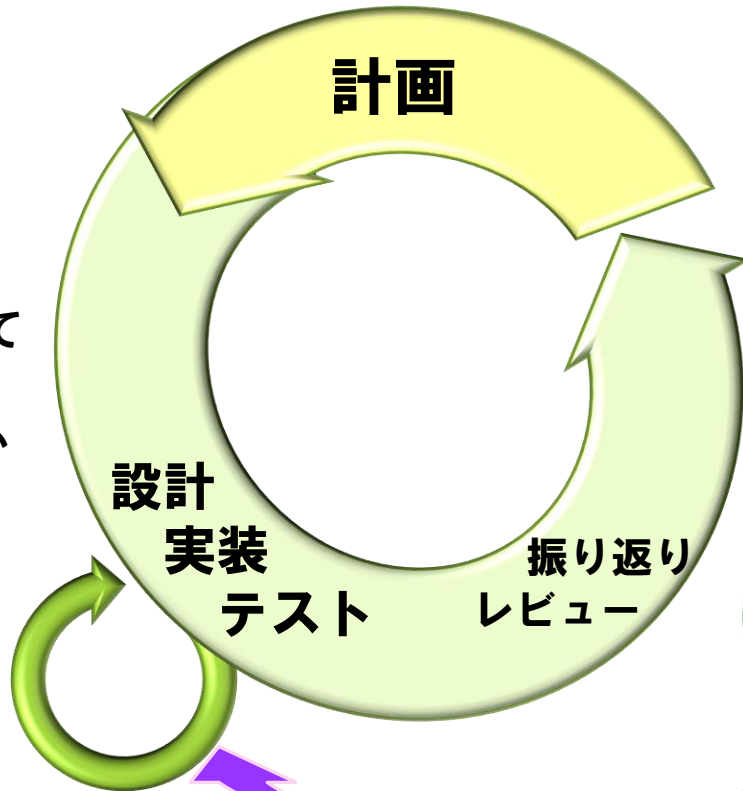
- どのようにしてその価値を
作り上げるのか



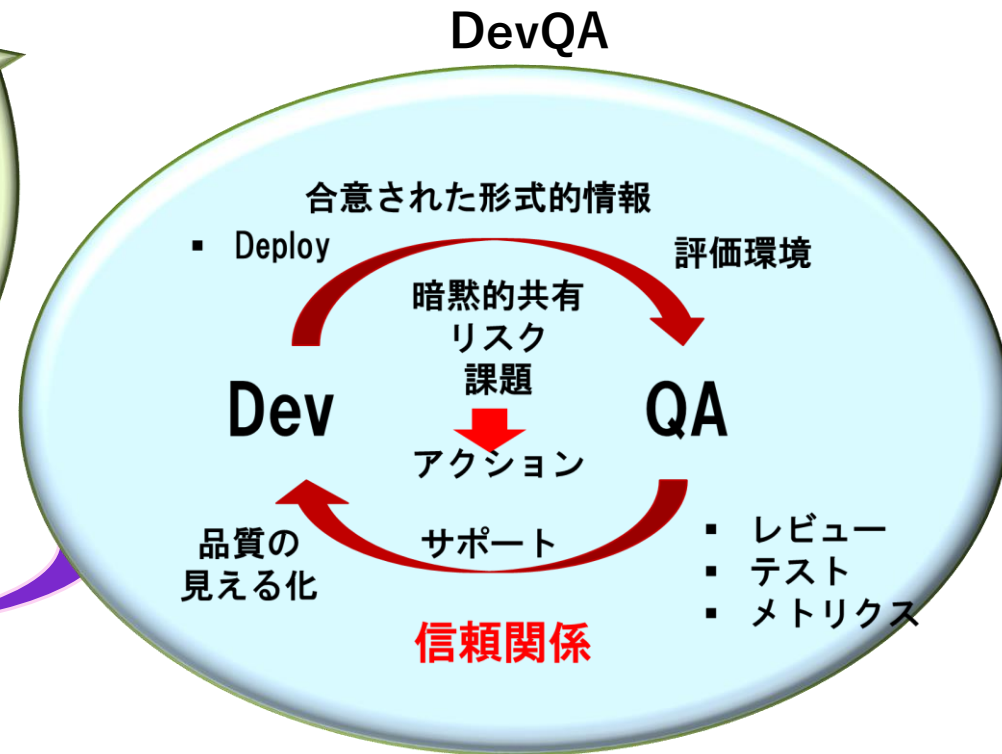
- 作り上げたものは、
その価値を満足するものなのか

価値の創造と流れ：バリューストリーム

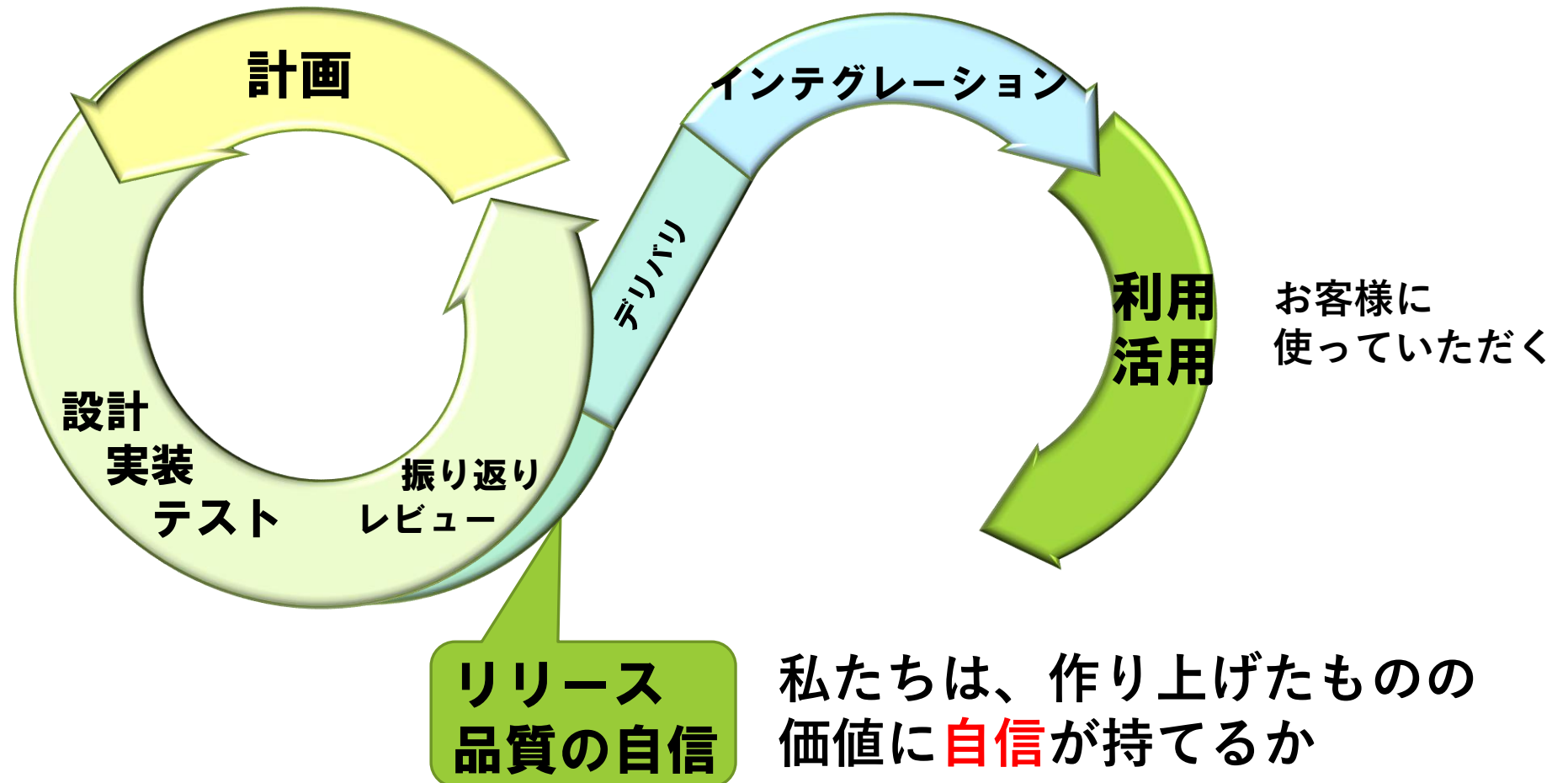
- どのようにしてその価値を
作り上げるのか



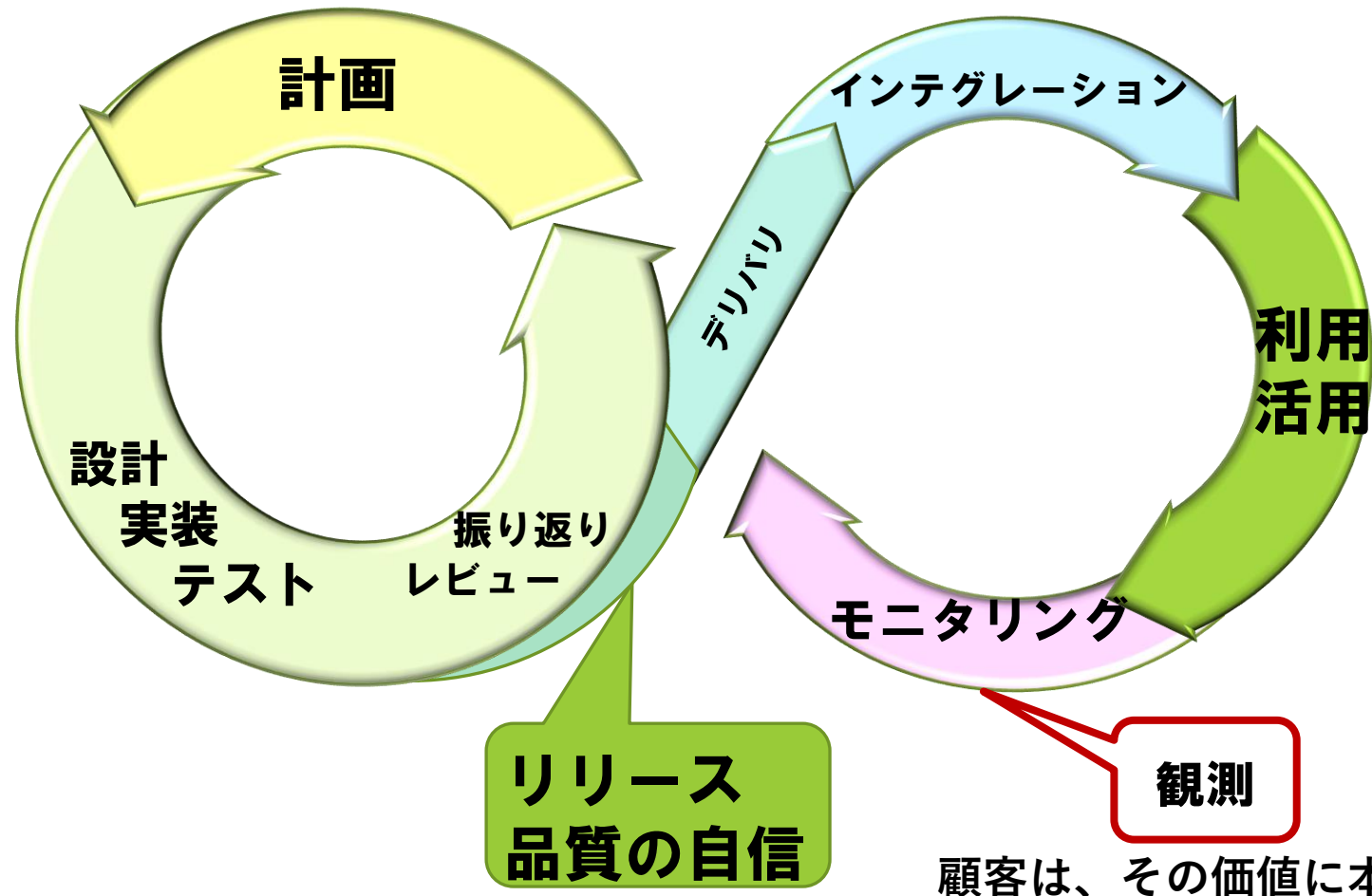
- 作り上げたものは、
その価値を満足するものなのか



価値の創造と流れ：バリューストリーム

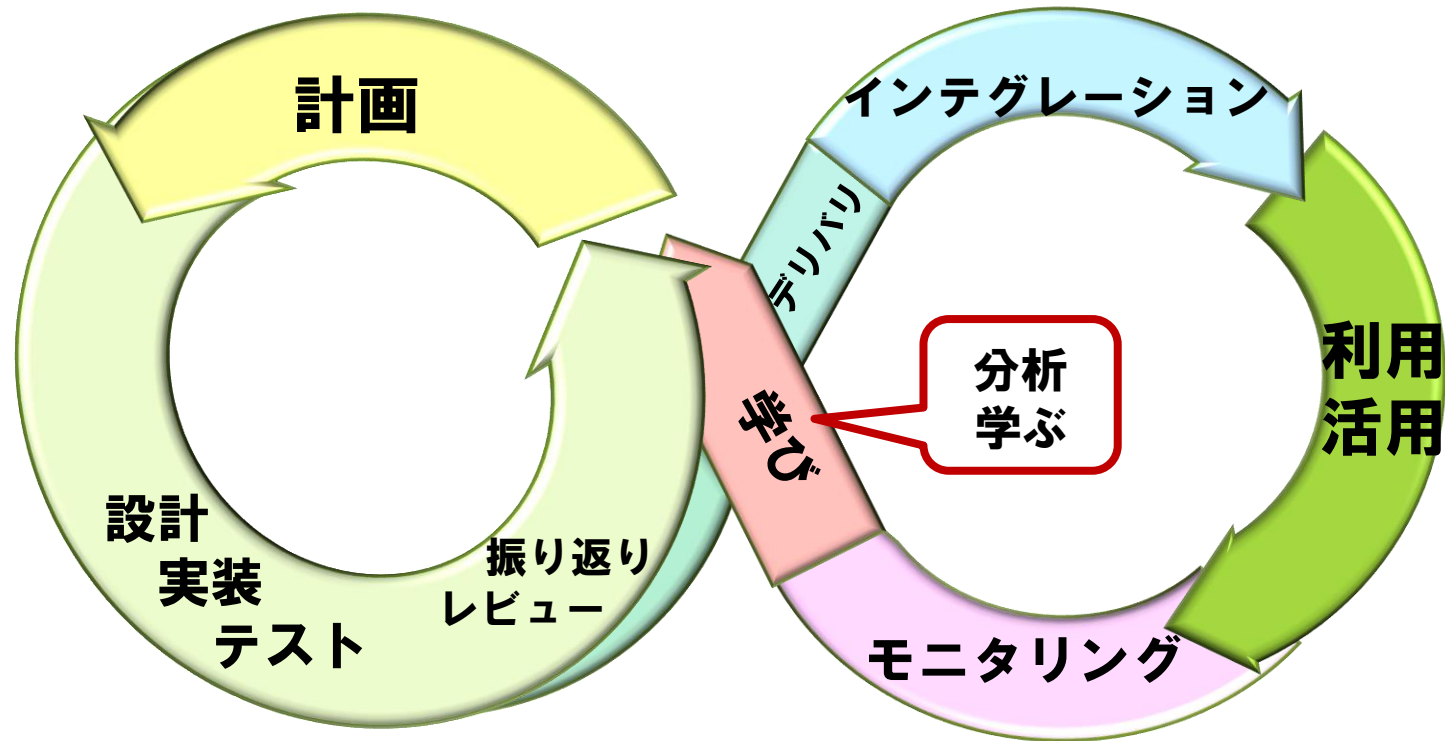


価値の創造と流れ：バリューチェーン



価値の創造と流れ：バリューチェーン

- よりよい価値を、よりよく提供するにはどうしたらよいか



顧客からのフィードバックから
何を学ぶか

品質＝価値をどのようにして提供し続けるのか

計画

- どんな価値を提供すればよいのか

設計・実装 テスト

- どのようにしてその価値を上げるか
- 作り上げたものは、その価値を満足するものなのか

品質の自信

⇒ 私たちは、作り上げたものの価値に自信が持てるか

モニタリング

- 顧客は、その価値に本当に満足しているか

学び

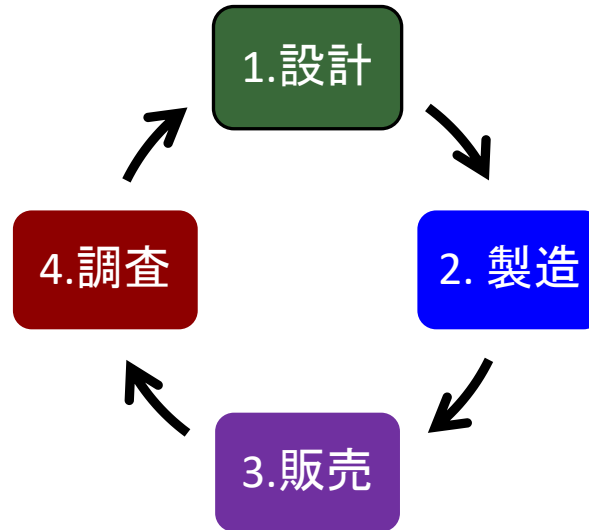
⇒ 顧客からのフィードバックから何を学ぶか

改善

- よりよい価値を、よりよく提供するにはどうしたらよいか

Deming Wheel

1950年 デミングサイクル (Deming Wheelともいう)



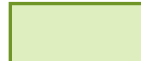
1. 製品を設計する
2. 製品を製造ラインで製造し、テストをする
3. マーケットに投入する
4. マーケットリサーチを通して使用される中でテストする。そしてユーザーは製品をどう思っているか、買わない人はなぜ買わないのかを解明する。そして1に戻る:顧客の品質や価格に対する反応の観点から製品を設計し直す。これをぐるぐると回す

品質に対しての良いフィードバックを得たい

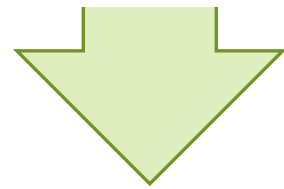
よい魅力的品質価値を生み出していきたい



お客様の判断＝フィードバックを手に入れたい



よいフィードバック



デリバリーする製品・サービスの品質が
良くなければならない

よいフィードバックとは

価値のアイデア：

こうすると、このようになってもっと嬉しい



顧客からの信頼、期待がある状態・関係

価値を提供できているからこそつかめる関係

品質保証とは

そして、品質保証組織、メンバーの役割、目的

品質保証とは (私の定義)

”顧客価値を製品、サービスに組み込むしくみ及び活動のこと”

なので、その活動はチームすべてのロールに分散している。 = Whole Team

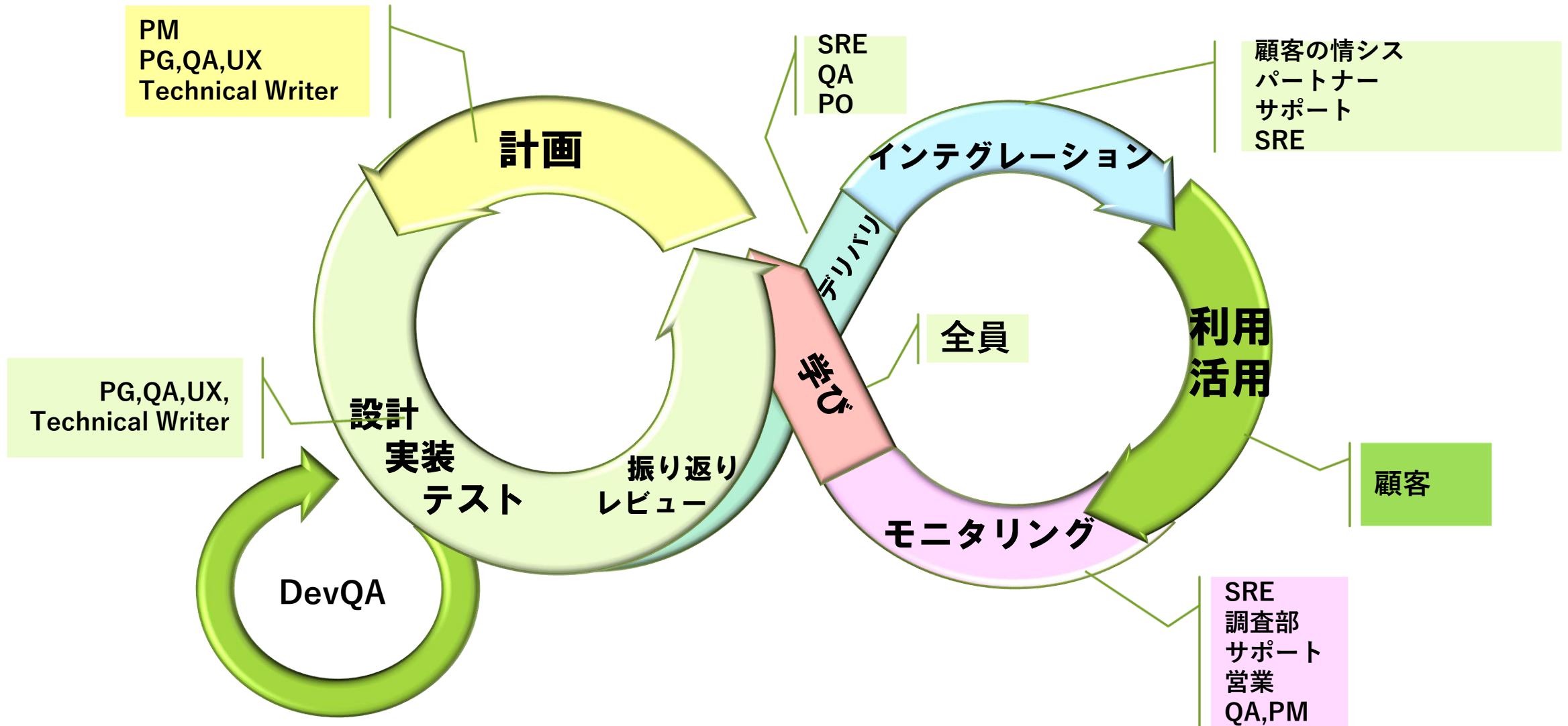
なお、顧客価値も、ここではバックログアイテム（要求）のことで、仮説であり、それを主に作るPOもPBIに品質を組み込む品質保証を行っている。

カスタマーサポート、顧客のモニター、顧客の調査分析チームも、仮説である顧客価値が、本当に顧客にとって価値があるのかという情報を開発チームにフィードバックしてくれる。これも品質保証である。

チーム全員が品質の組み込みにかかわり、つまり、品質保証を行っているといえる。

この活動の結果：顧客がこちらの目論見通り満足し、
あるいは安心して使って“うれしい”と言っていただけ

すべての人がかかわるバリューチェーン

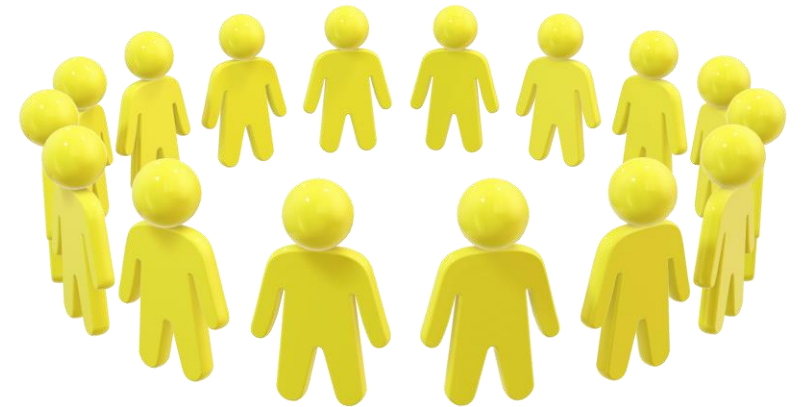


Whole Team パターン

“だれも”が品質＝価値を作り上げることに貢献している

それぞれの役割において

多様な観点によって、補い、コラボし



”提供する価値”に責任を持っている

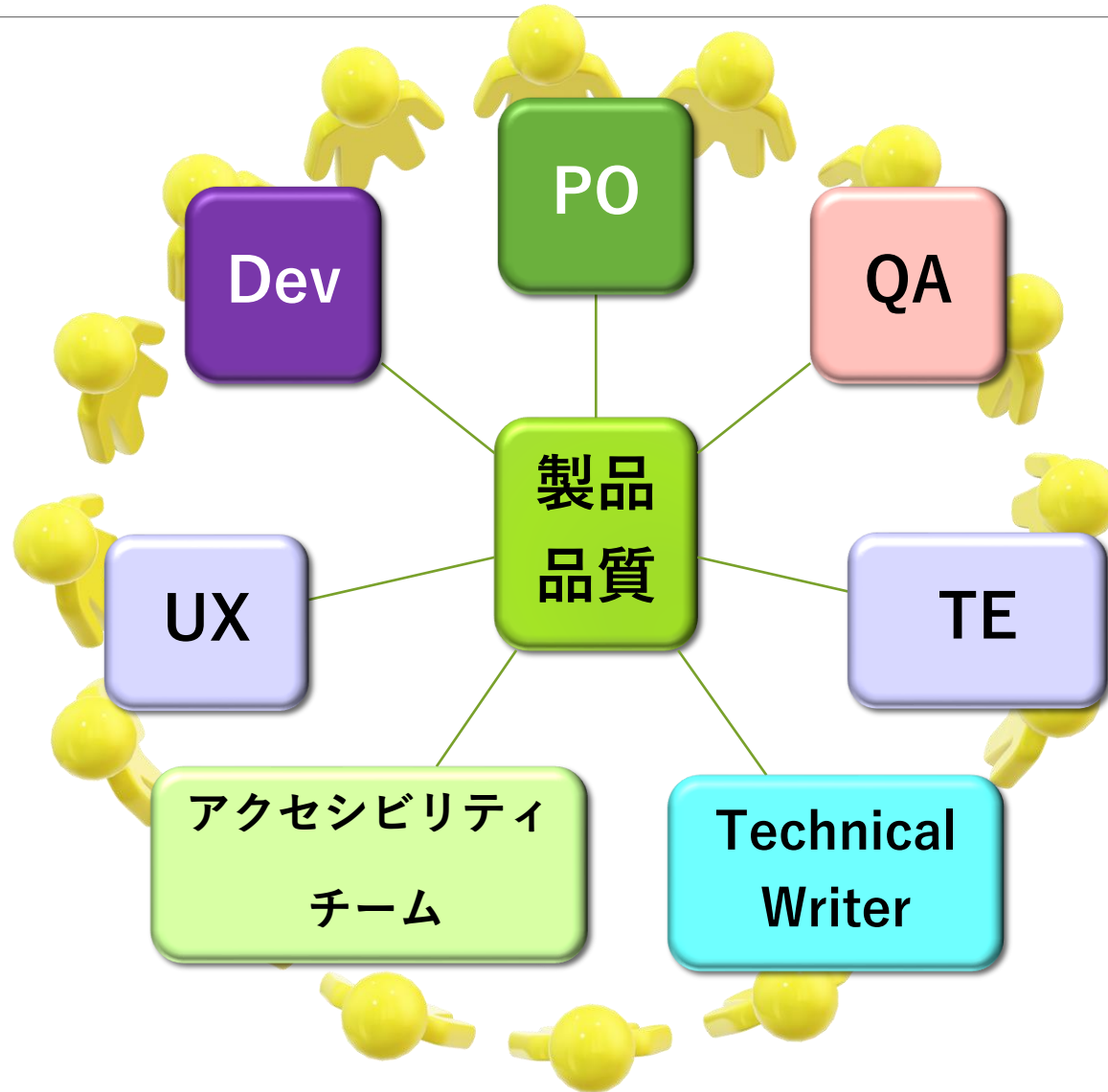
多様性が品質を支える

観点

情報

知識

専門性



QAに求められているもの

ソフトウェア品質に特化した多様性

謙虚なDevの話

謙虚なDevの話：私たちは、与えられた問題をどう実現していくか、そして実現を遂行することに集中したい。集中するが上に、見落とすもの、過ちはある。それを、違う品質の側面でフィードバックしてほしい



QAは、ソフトウェア品質に特化した多様性を持ち、それを常に磨かなければならない

品質保証メンバーのロール

QAは、

- ソフトウェア品質のスペシャリストとして、
- 品質のビックピクチャの観点を持ちながら、
- 製品の品質を顧客視点で肌で感じ、
- 製品の品質とプロセスの品質とバランスをもって
- Whole Teamが支える品質保証が進化改善していくよう
- サポートするロールである。

品質＝価値をどのようにして提供し続けるのか

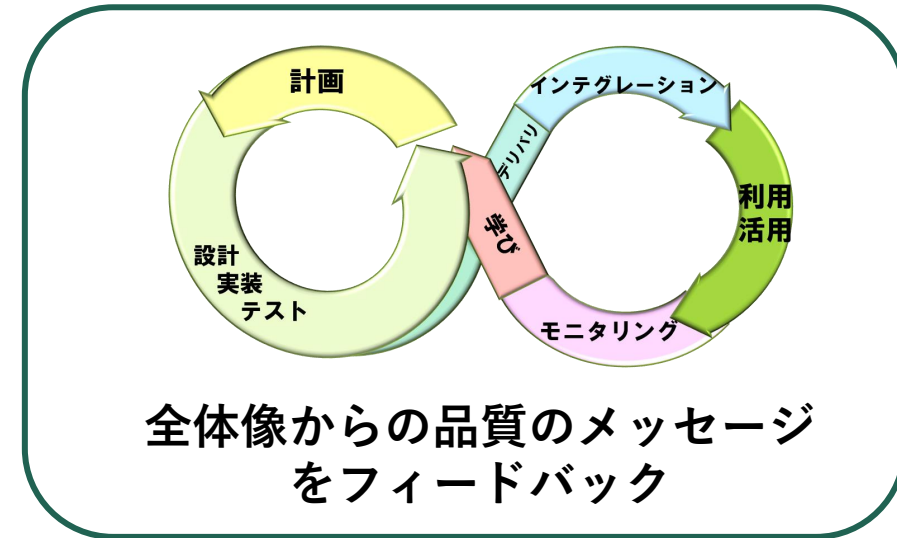
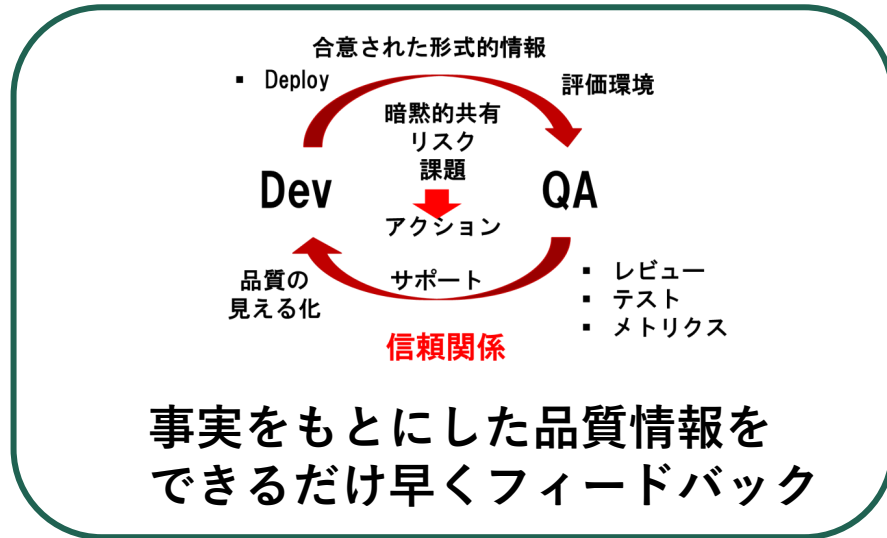
よりよい価値を、よりよく提供するにはどうしたらよいか

Whole Team

- どのようにしてその価値を上げるか
- 作り上げたものは、その価値を満足するものなのか
 - ⇒ 私たちは、作り上げたものの価値に自信が持てるか
- 顧客は、その価値に本当に満足しているか
 - ⇒ 顧客からのフィードバックから何を学ぶか

QAは、バリューストリームに対してどういう貢献ができるか

QAは、短い周期で品質のフィードバックをする機会を繰り返し持つことができる



品質保証をサポートする

チームは、より早く気づきを得て
品質の作り込みを改善する

チームは、その品質の見える化により、
成果物の品質に自信を持つことができる

アジャイル・プロセスにおけるサポート

アジャイルでは、質の高いプロセスが必要

アジャイルにおけるプロセス

決められたプロセスが守られているかどうかを管理することよりも

プロセスをつかさどる“人”と、人どうしのインタラクションがより重要なんだ



プロセスで何をするか、中身、質が重要だ

要求毎に、必要とするプロセスは変わる。それが適切にできるか

どのようなインプットが必要か： 調査の必要性、プロトタイピング

どのようなタスクを、どんな順番で行うか：段取り、PFD (Process Flow Diagram)

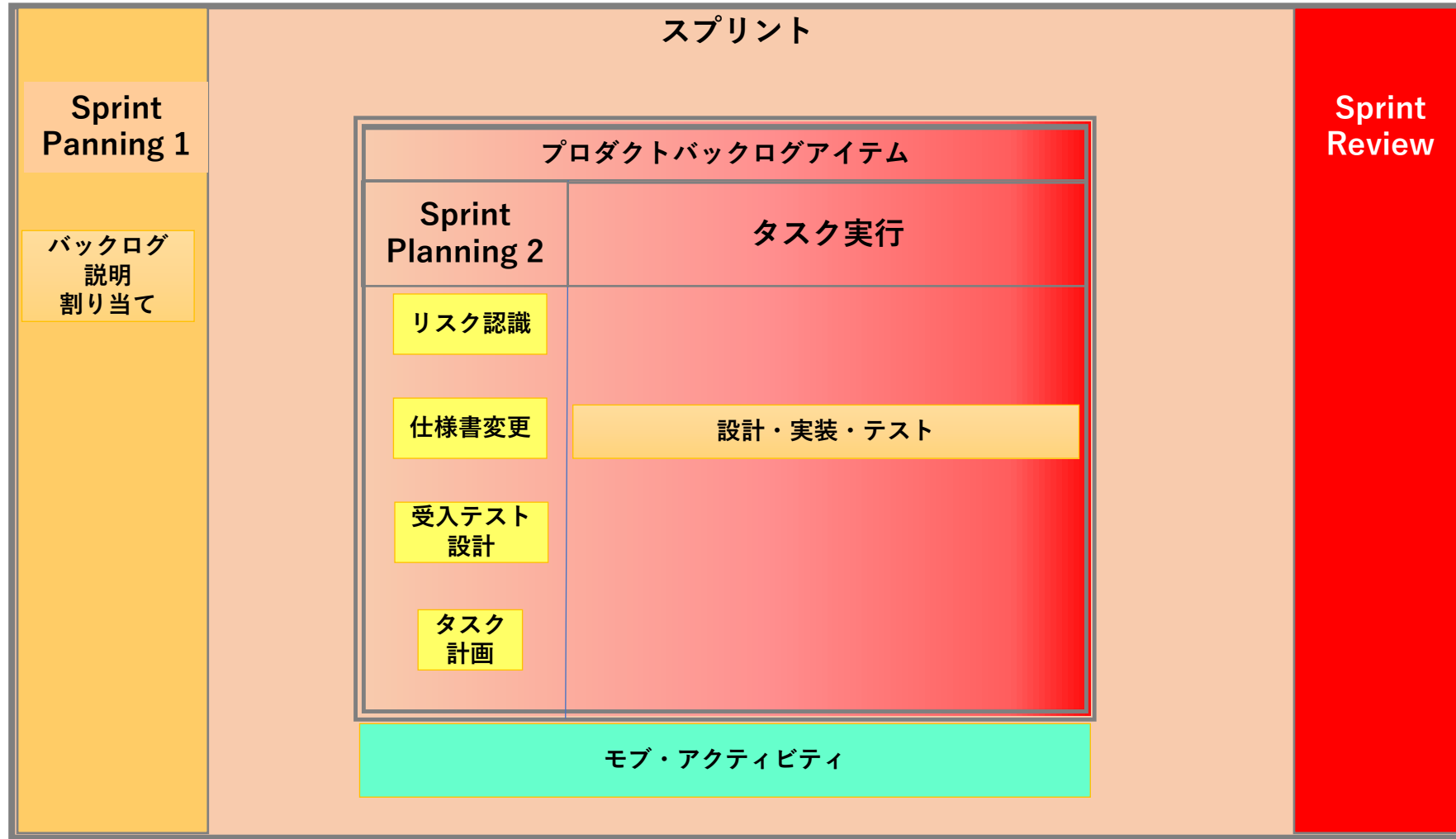
成果物は、何が必要か：モデリング、ドキュメント、コード

アジャイルはプロセス

プロセスを守るのではなく、要求の開発作業に適合する、最適なプロセスを設計する。
バックログ(要求) 毎に、プロセスを設計する

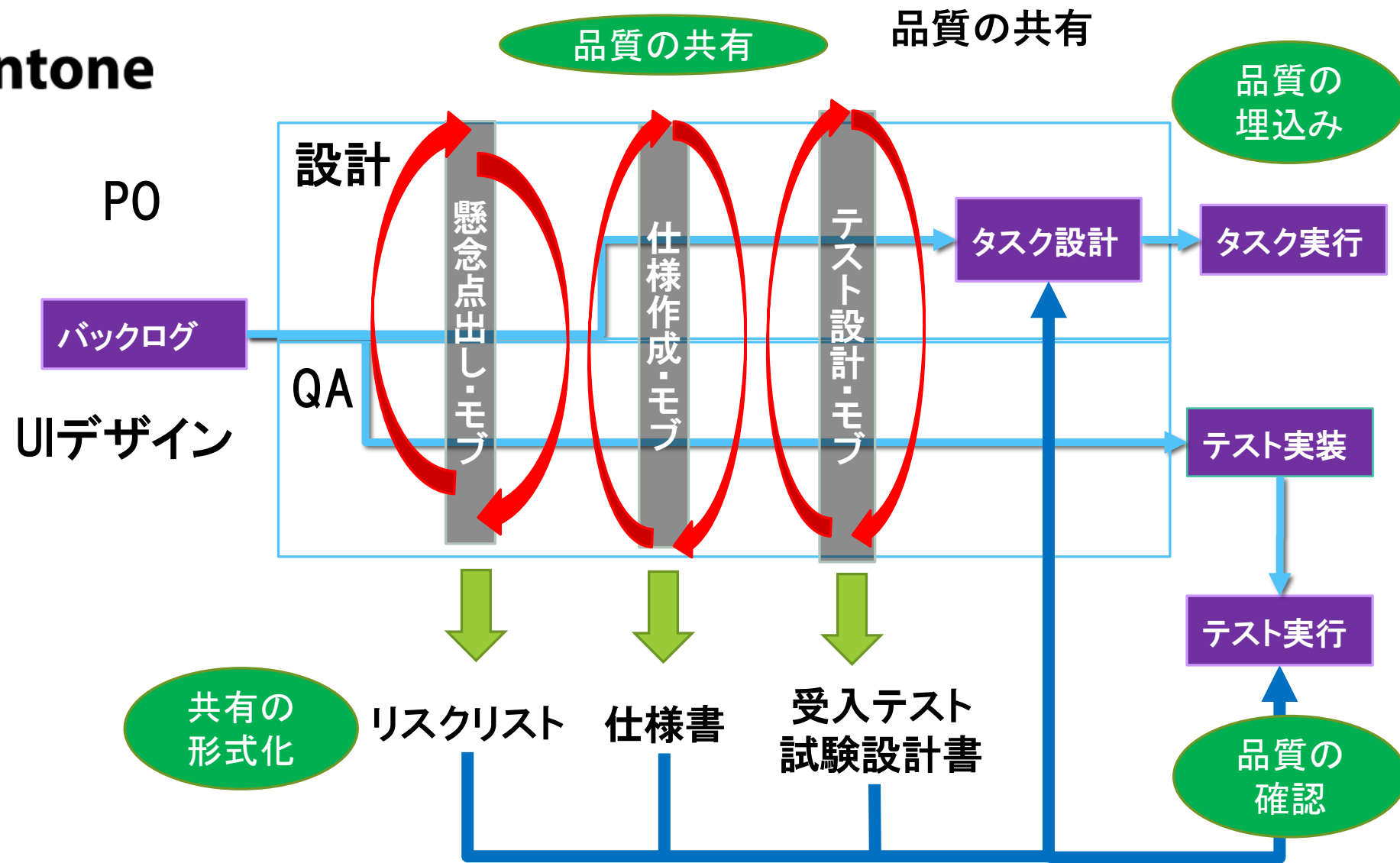
プロセスは、チームが決めるもの
チームが設計していくべきもの

開発プロセス：LeSSフレームワーク例



PBI
Done

スプリントプランニング 2



懸念出し

リスクを洗い出す

未知：わからないこと

調査
試行

懸念：起こるかもしれない問題

懸念出し ～例

- 仕様の不明点
- 設計・実装上の懸念
 - 初めて、久しぶり
 - 機能、コード、ライブラリ、言語
 - わからないこと
 - 構造、責務
 - オプション
- エラー処理
 - 条件、出かた、優先度、検出
- スコープ
- どこで実現するか
- 影響箇所、error prone、複雑度
- 非機能要求

懸念出し

- ・ チームでモブによりリスクを出し整理をする
- ・ QAもモブに参加して、QA観点のリスクを共有する

多様性によりリスクの漏れを防ぐ

リスクへの対策のモチベーションがうまれる

必要十分なテストが設計できる

仕様書修正

仕様：システムの振る舞い

開発：システムの振る舞いを作りこむ作業

仕様に誤り（欠陥）があれば

システムの振る舞いが想定と違う



不具合

仕様書修正

仕様書の修正は、QAとモブで行う

- モブの強力なレビュー効果で仕様に誤り（欠陥）を入れない
- QAからも、品質観点が仕様に盛り込まれる

必要ならば、PMも呼ぶこともある

仕様の品質が上がる

品質が組み込まれる

仕様書修正 ～例

- タイミングを表現に加える
 - 条件を加える
 - どのように表現するか
 - 言葉
 - イメージ
 - 誤解・非曖昧性防ぐ
 - 書かなくてもわかるか
 - 暗黙の了解（しつこくなる）
 - 仕様書のどの章で書くか（構成）
-
- POにレビューしてもらう
 - マージして構成管理をする

受け入れ試験設計

QAが主体となってテスト設計をモブで行う

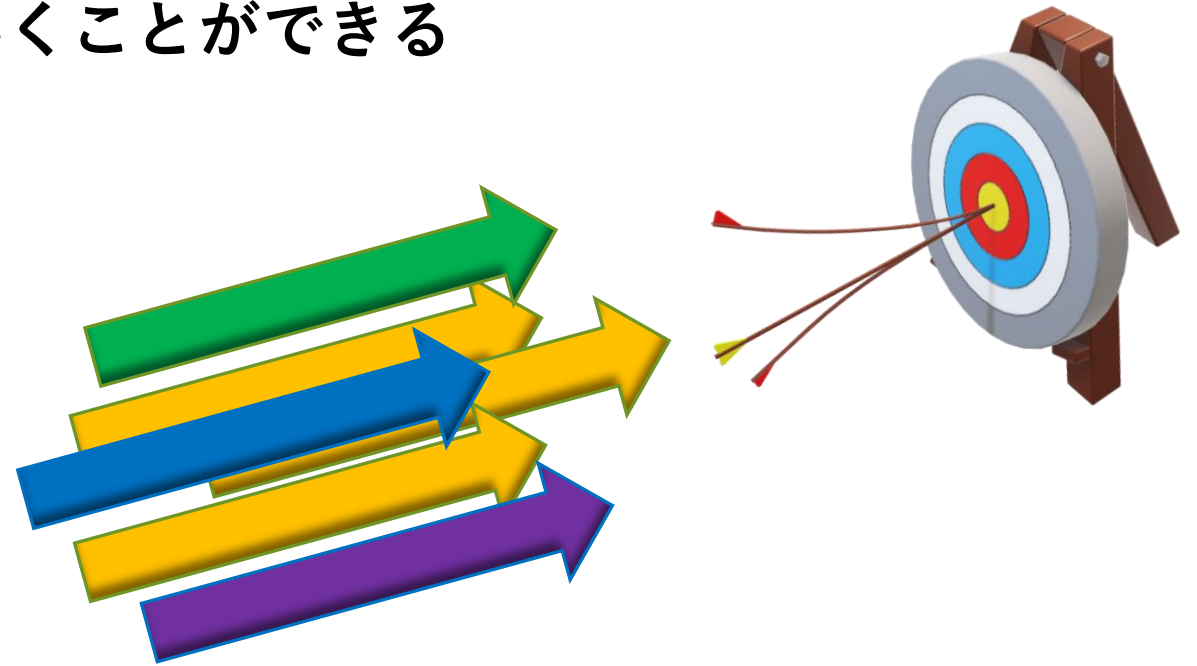
テスト設計＝開発の品質ゴールの明記とその共有

- ユーザストーリーと受け入れ条件をどのようにテストしていくかを設計していく
- QA⇒Dev
 - 新たなリスクへの気づき
- Dev⇒QA
 - コードの構造をモデルなどで説明
 - グレーボックステスト
- 自動テストと手動テストの仕分け
- DevとQAとのテストの仕分け

これによって出来上がる状態

QAメンバーも開発メンバーも、同じチームとなって、
同じゴールに向かって、走っていくことができる

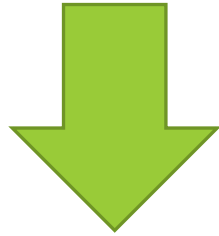
Whole Teamで
上流での
品質を組み込みを行う



タスクの洗い出し

リスク分析
仕様書の確定
受入テスト設計の共有

スプリントバックログの設計



PBIごとに最適なプロセスを設計する

どのような段取りで、**品質を組み込むか**の設計（計画）を
チーム（モブ）で行う

プリントバックログ：カンバン

合計時間: 0h	TODO						実装	レビュー	完了		
									13013	13010	13015
									13014 随時性情報確認		
	12906 日本橋駅主入口	12907 日本橋駅主入口	12908 駅1 試験実施/手 書	12909 Greenで動作確認 す	12910 PM・デザイン/監 修	12911 仕様書をまとめ る	12912 WIPコミットを監 修	12904 デザイン/監修			
	12990 PM仕様書/監修	12994 駅15588のデ ザイン	12995 disabled 対応	12996 システム管理の デザイン	12997 コンパイル田 デザイン/監修	12998 デザイン/監修	12999 駅1 試験実施工 事	12993 K-15603 駅15588のデ ザイン		12992 K-15588 駅15588のデ ザイン	
	13000	13001	13002	13003	13004	13005	13006				

ExcelをKintoneアプリに読み込んでカンバンを作り、スプリントバックログを管理する

アジャイル・ドキュメント

筆まめのすすめ

アジャイル・ドキュメント

アジャイル開発では、何をドキュメントとして書くべきか

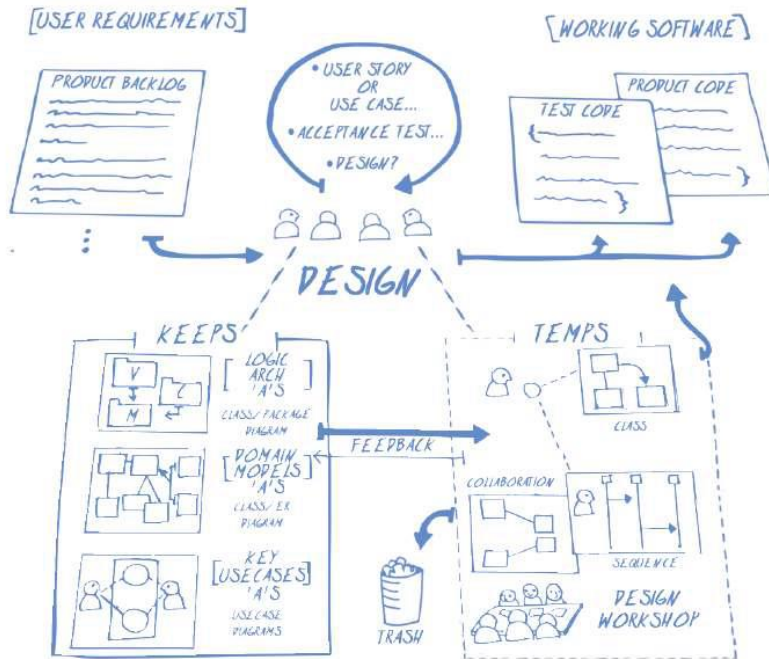
目的は二つ

1. 開発で使うため
 - Devが参照する
 - QAが参照する
 - 保守、サービスが参照する
2. 成果物として、デリバリーするため
 - マニュアルなど

アジャイル・モデリング

KeepsとTempsは、平鍋さんのアジャイル・モデリングのお話からいただいた

もちろん、モデルもアジャイル開発において重要なドキュメントである。



- Keeps
 - 開発の基盤となるモデル
 - 変化はあまりない
 - コードの動機をとり、保守をする
 - 構成管理をする
- Temps
 - コミュニケーション、共有理解のためのモデル
 - テンポラリで保守は行わない

“アジャイル時代のモデリング”, 2014, 平鍋健児、チェンジビジョン

アジャイル・ドキュメント

開発で使うドキュメントの種類

- Keeps
 - ・ 開発の軸となるドキュメント
 - ・ 保守をする＝コードと同期をとる
 - ・ 構成管理をする
 - ・ 例：機能仕様書、テスト仕様書
- Temps
 - ・ コミュニケーション、共有情報のためのドキュメント
 - ・ テンポラリで保守は行わない
 - ・ その開発バージョンのためのドキュメント
 - ・ 例：PBI、スプリントバックログ, 設計メモ
 - ・ バックログなどに紐づけて、検索できるようにしておく
とよい

ドキュメントの例: kintone開発

ドキュメント	種類	利用	作成型態	レビュー	備考
PBI	Temps	バックログのライフサイクルで参照	モブ	チーム全員	リファインメント
リスクリスト	Temps	タスクを設計する際に用いる	モブ	Dev,QA	
仕様書	Keeps	計画、開発のいろいろな場面で頻繁に参照される	モブ	Dev,QA,PO	POが後でレビュー
受け入れ試験設計書	Temps	受け入れ試験設計の共有のため	モブ	Dev,QA	
テスト仕様書	Keeps	機能試験設計や回帰テスト設計で参照される	QA個人	QA	
タスク設計	Temps	スプリントバックログアイテム策定のため	モブ	Dev,QA	
アジャイル・モデル	Temps	設計の際の理解のための共有情報	モブ	Dev	QAも参照あり

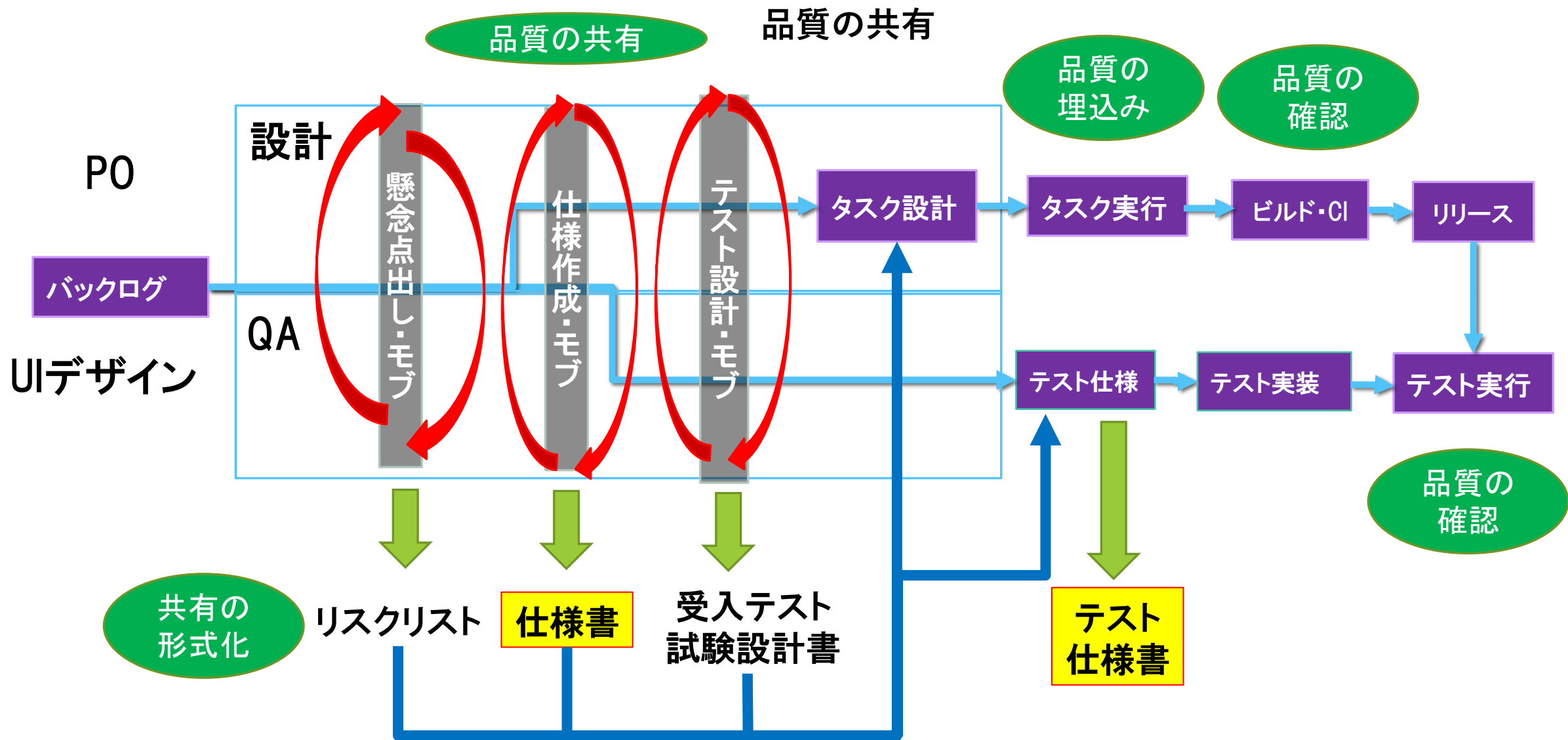
仕様書は、設計実装の前に、DevとQAが、モブで修正・追加変更される

Dev: 設計実装において、振る舞いのブレもぬけもれもない

QA : 仕様書という重要なテストベースを手に入れる

仕様書の品質がここで決まり、上流での品質が担保される

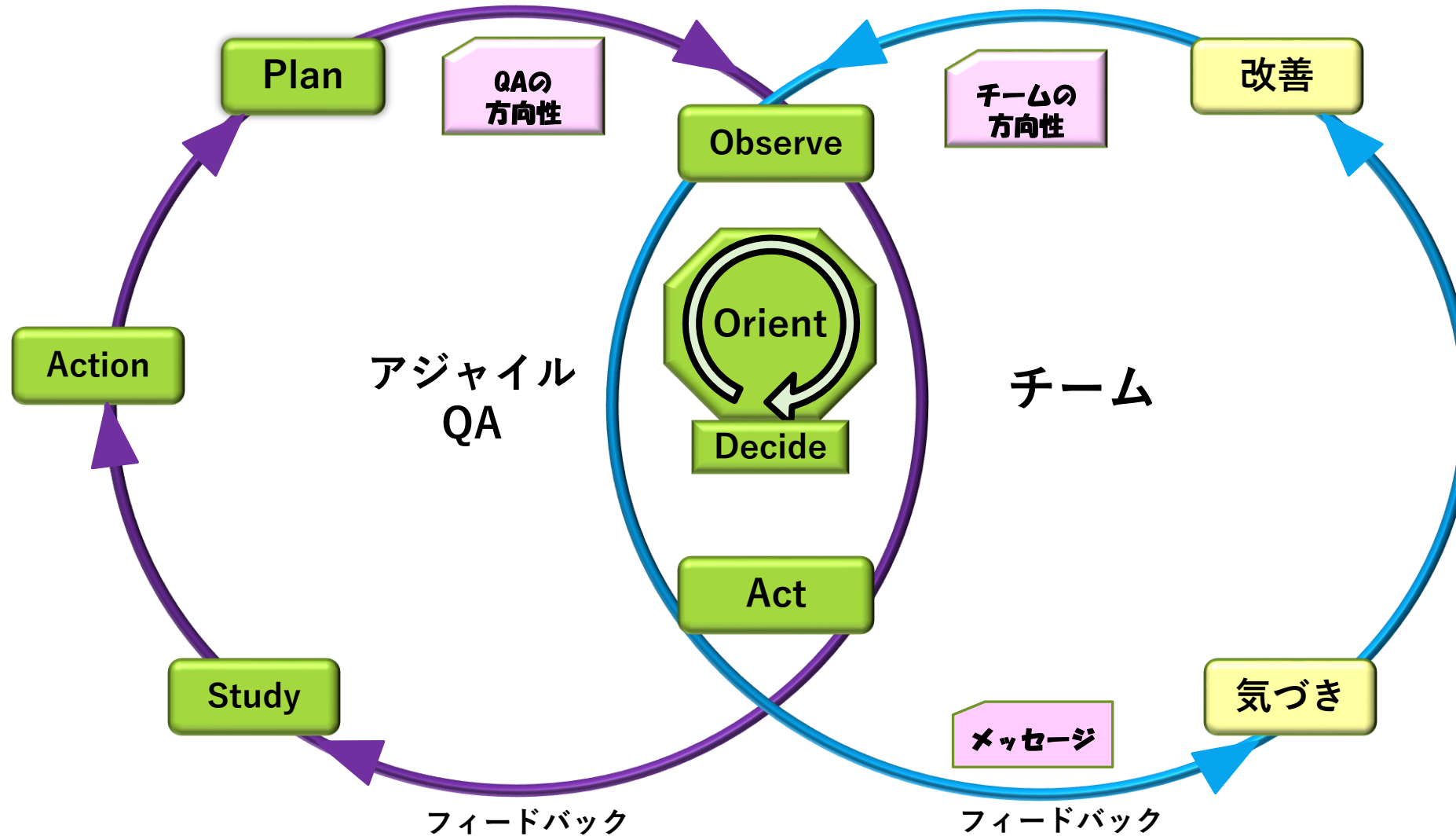
スプリントプランニング 2



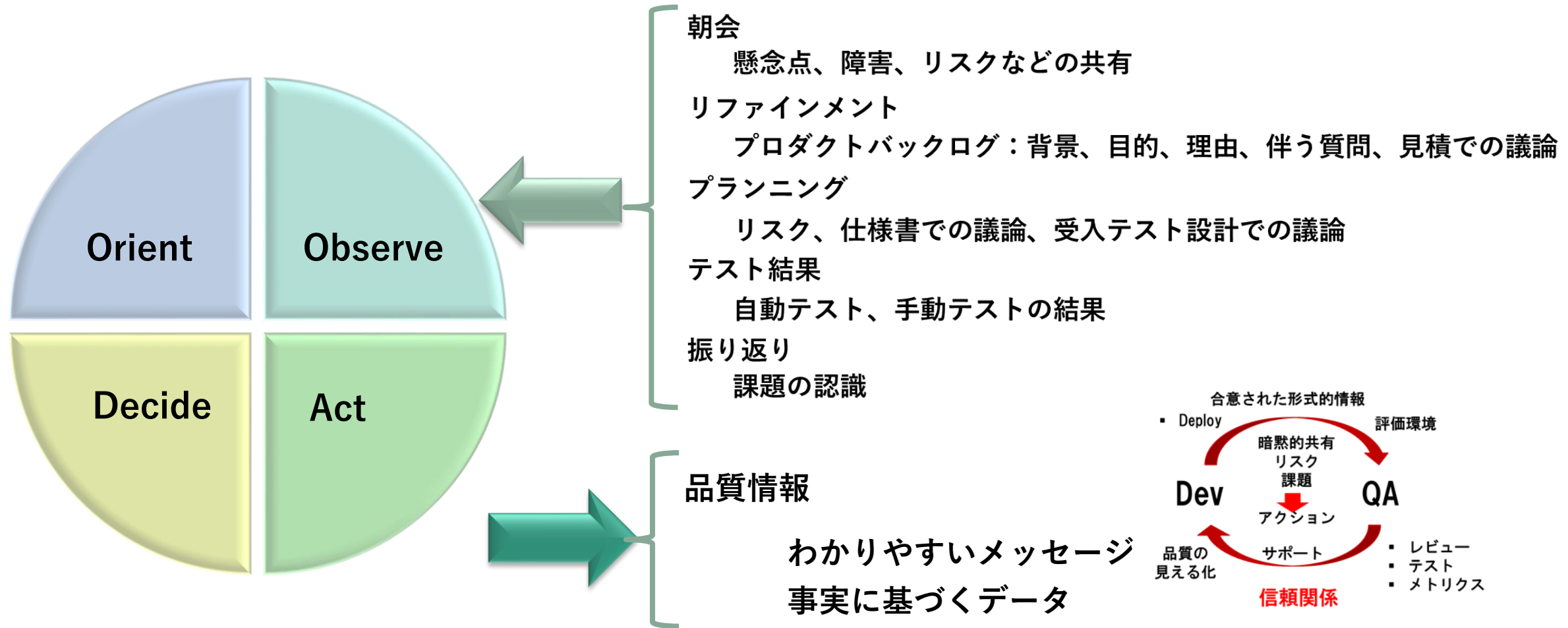
アジャイル・クオリティ・モデル

サイボウズのアジャイル品質保証のアプローチ

アジャイル・クオリティ・モデル

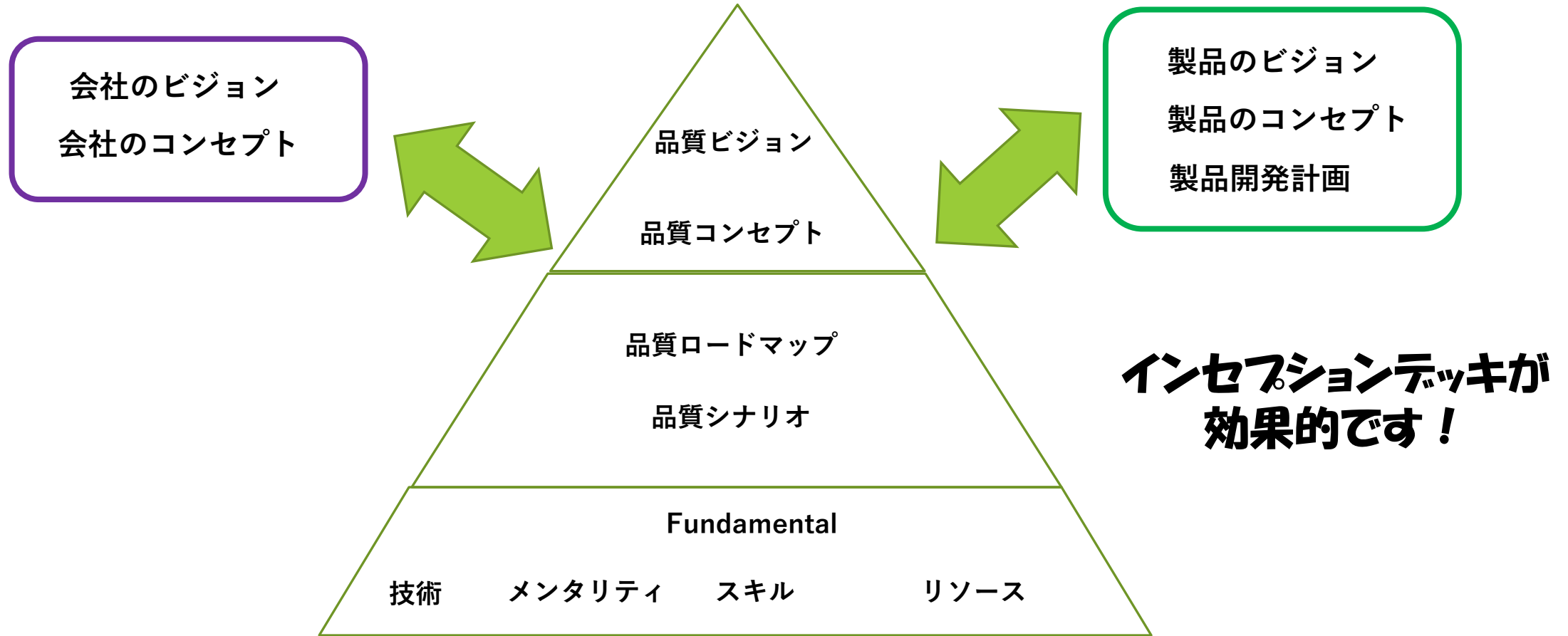


OODA : QAがチームをサポートする



Planning : アジャイルQAの計画

品質ビックピクチャ





企業理念 2020



Purpose (存在意義)

チームワークあふれる社会を創る

Culture (文化)



理想への共感



多様な個性を重視



公明正大



自立と議論



サイボウズのメンタリティ

○ 心理的安全性が高い

○ お互いをリスペクトする: 謙虚

○ 発言平等性が高い

○ 新しいものを受け入れる: 好奇心

Study

アジャイルQAは現場から学ぶ

現場の判断、アイデアをリスペクトする

現場はそんな単純なものではない
多くの課題が複合的にあり、制約条件もある

過去の成功事例や知見、外部の知見は、現在の現場にとっては仮説にすぎない。

現場から、もっと価値ある知見を学ぶことができる

それにより、さらに外部からの知見からも学ぶモチベーションが起こる

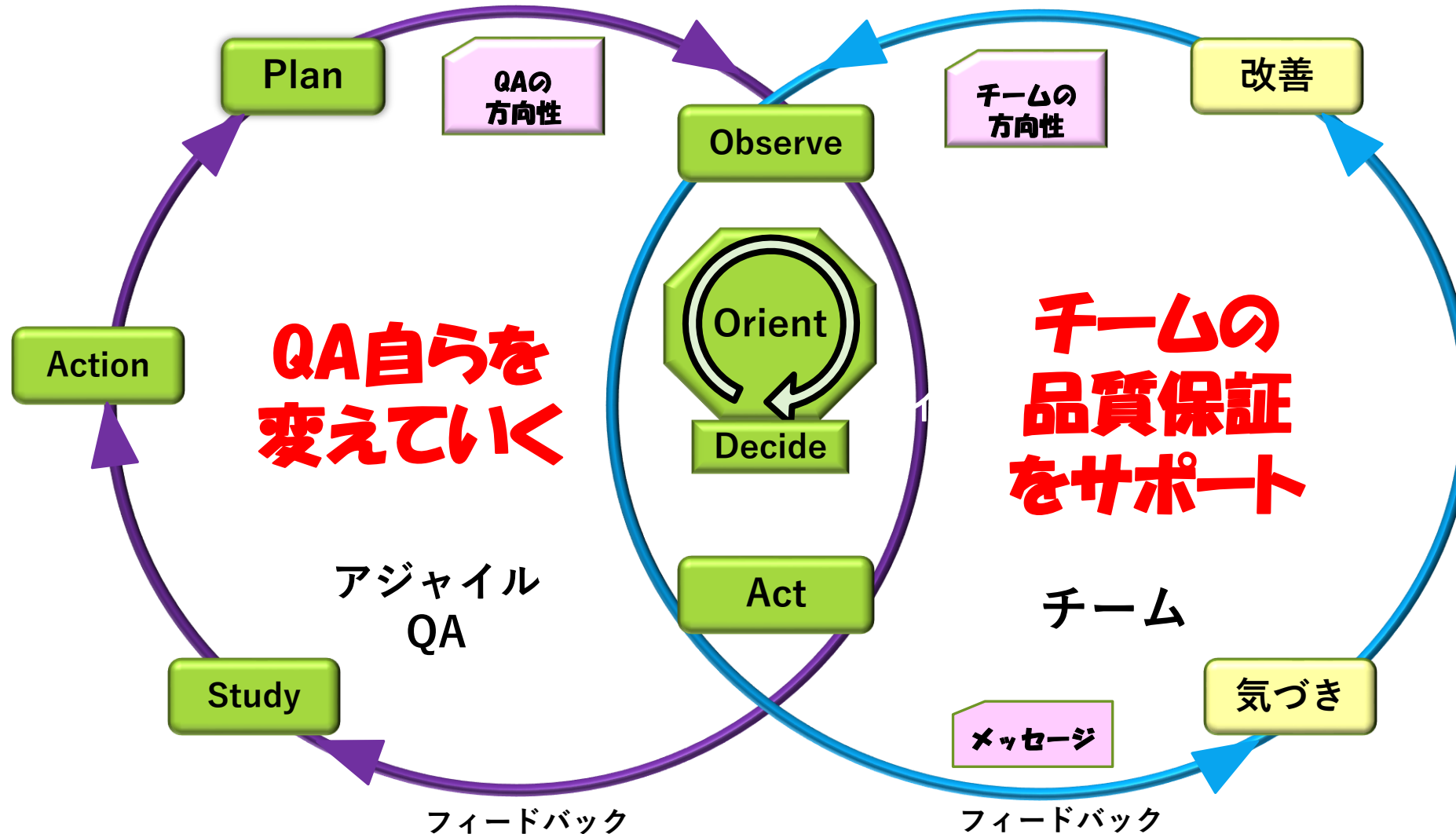
Action

学んだことで、QA自らの行動を変えていかなければならない

プランニングを見直すこと：Re-Planning

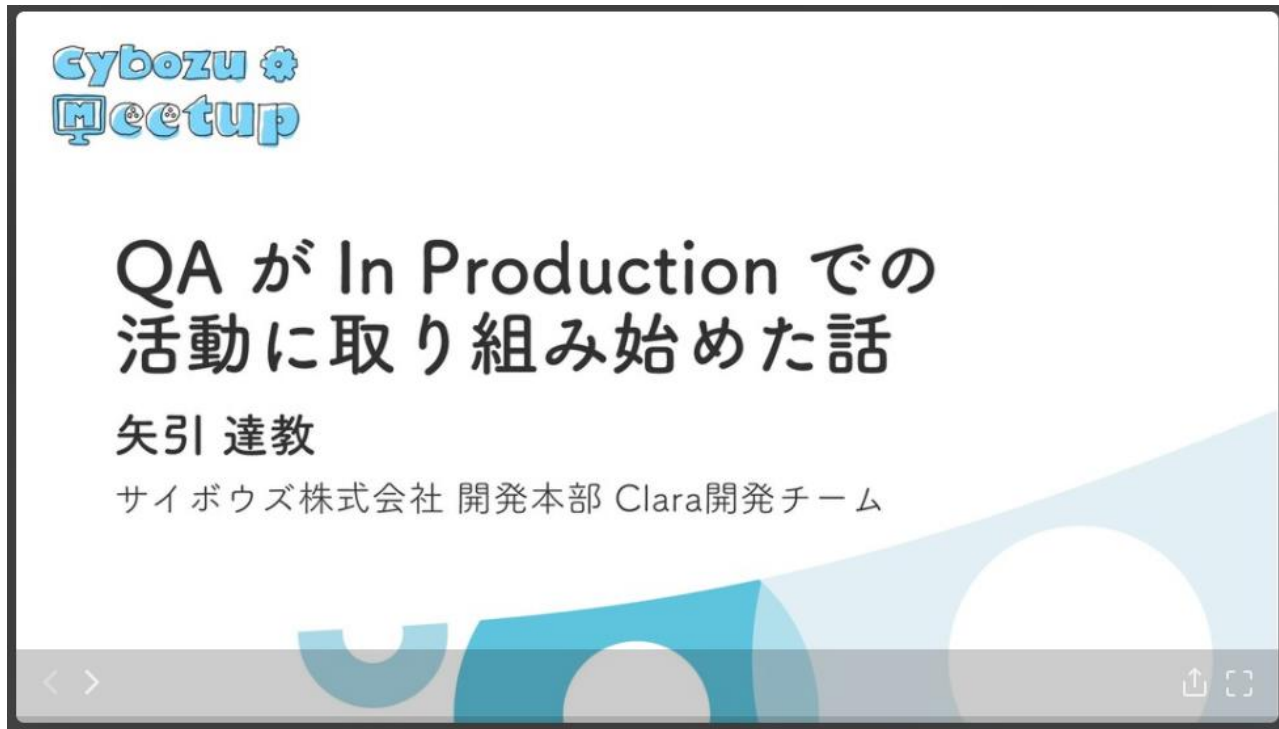
- どのような変化が必要か策定する
- ビジョン、コンセプト、目的、目標の調整
- ロードマップ
- 手法の変更

アジャイル・クオリティ・モデル



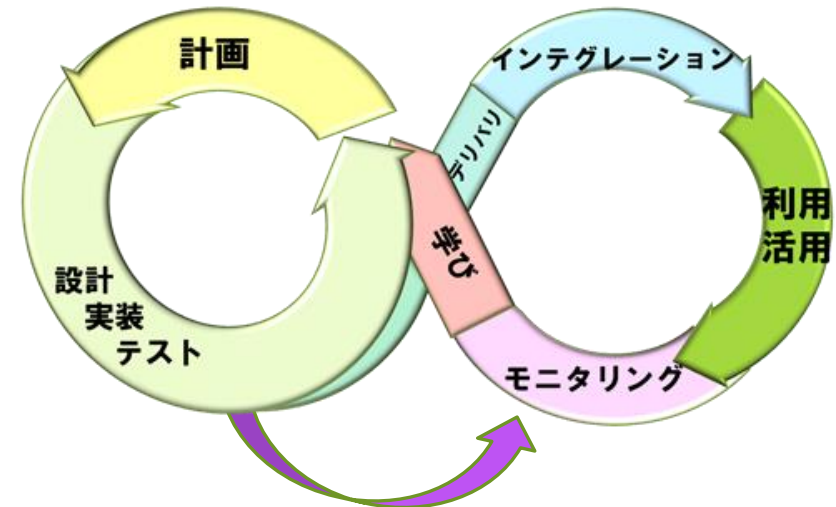
さらなる広がり

活動範囲を広げるQA



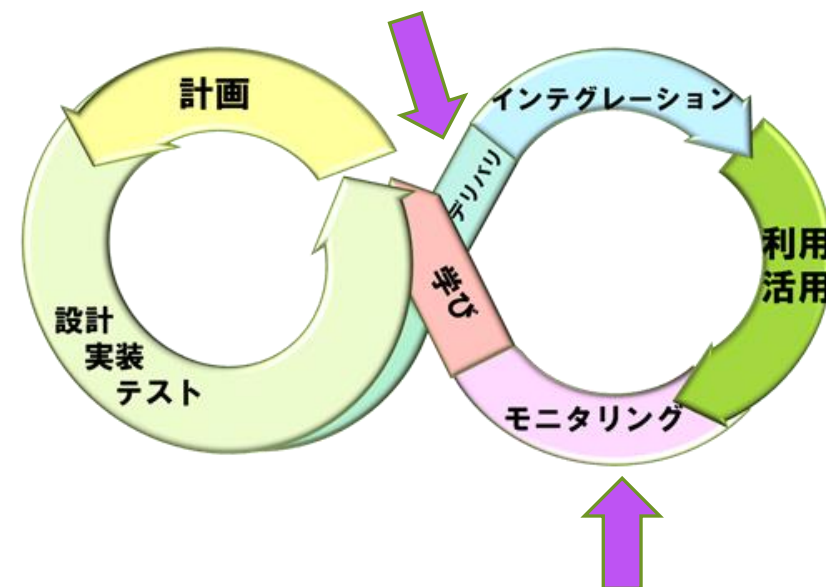
2020/08/19

<https://speakerdeck.com/yabbysan/qa-ga-in-production-defalse-huo-dong-niqu-rizu-mishi-metahua>



サイボウズ Data Center を支える インフラ QA の挑戦

サイボウズ株式会社 運用本部 山田 高大



2020/08/19

<https://speakerdeck.com/cybozuinsideout/saibouzudata-centerwozhi-eruinhuraqafalsetiao-zhan>

Cybozu Tech Meetup「活動領域を広げるQA」

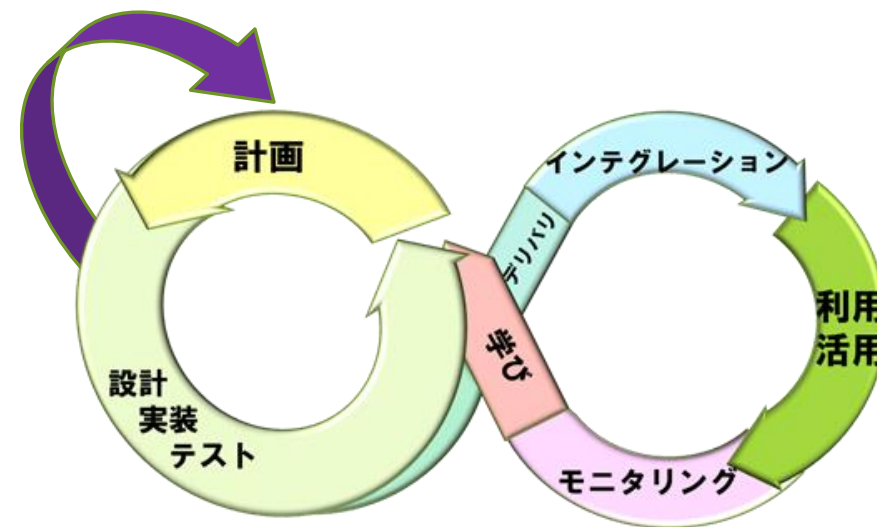
QAがバックログ作成をやってみた話

サイボウズ株式会社 開発本部
田村 ななみ

1

2020/08/19

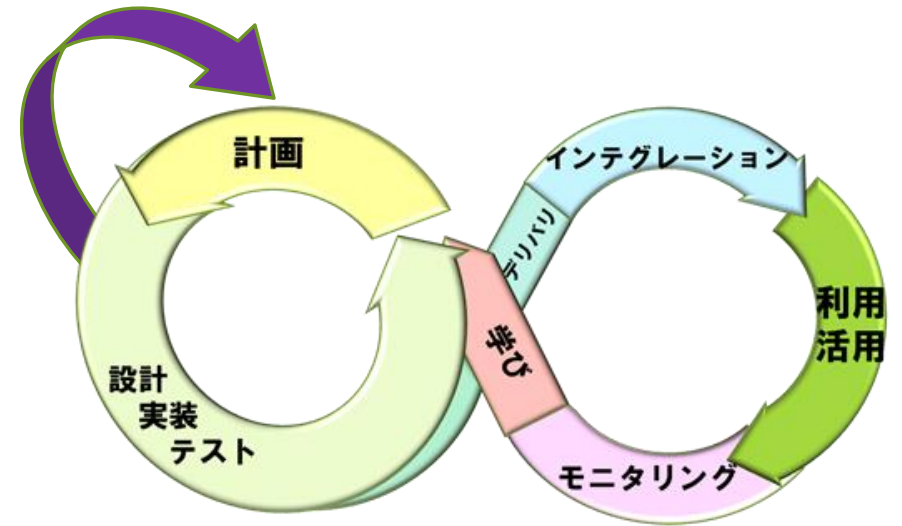
<https://speakerdeck.com/cybozuinsideout/qagabatukuroguzuo-cheng-woyatutemitahua>



まとめ

- QAがバックログ作成をすると
 - QA視点を活かすことができる
 - ユーザーを意識した試験ができるようになる
 - 製品理解が深まって楽しい！
- これからも職能にとらわれずに品質向上を探究していきたい

25



2020/08/19

<https://speakerdeck.com/cybozuinsideout/qagabatukuroguzuo-cheng-woyatutemitahua>

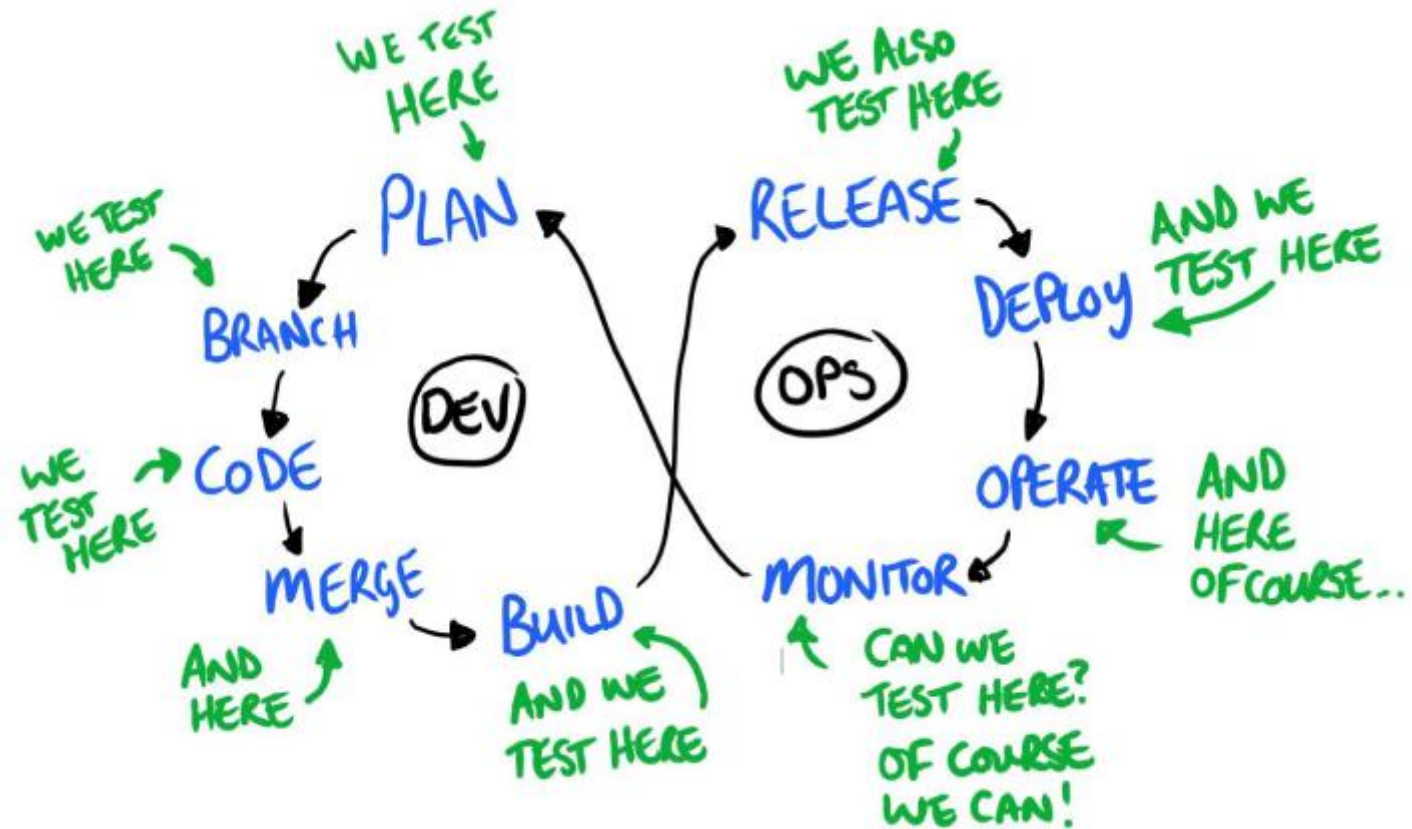
まとめ

アジャイル・クオリティの品質保証

顧客価値を、製品及びサービスに組み込む活動

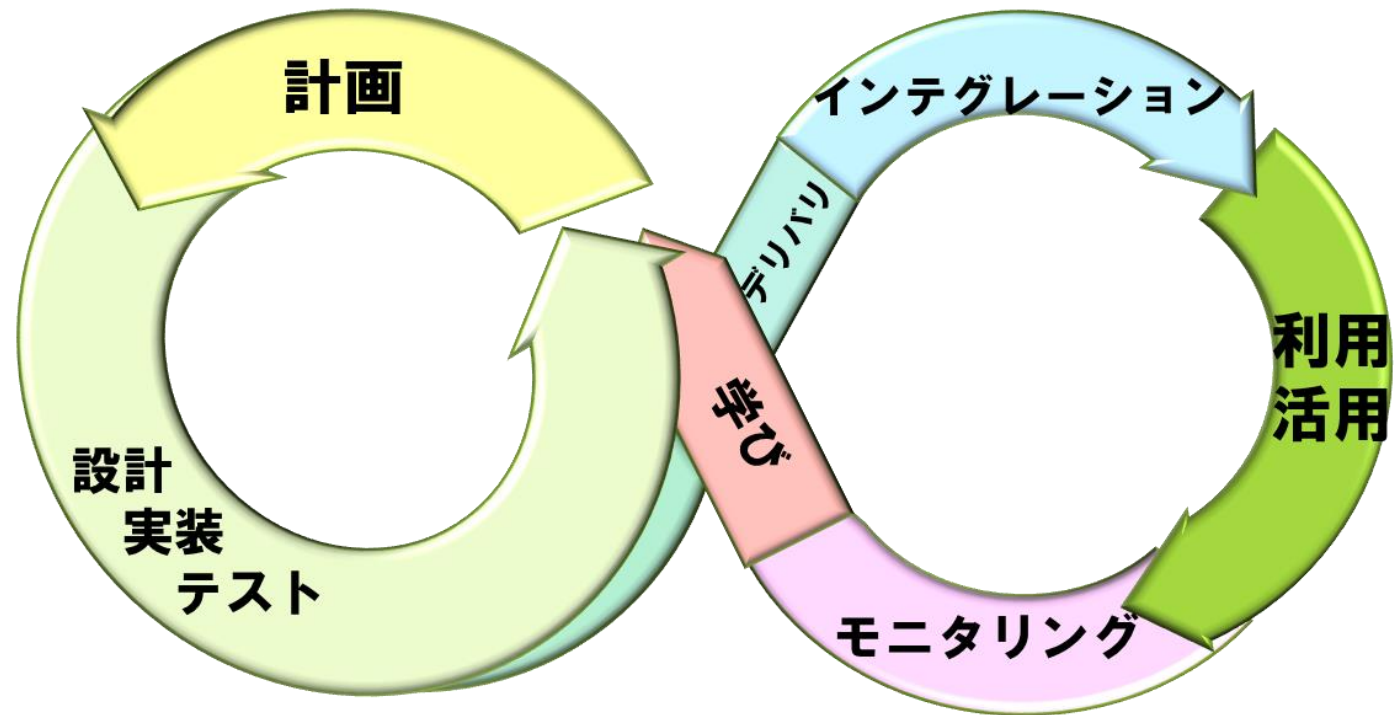
チーム全員が品質の組み込みにかかわり、
つまり、Whole Teamで品質保証を行っている

その中でQAは、
ソフトウェア品質のスペシャリストとして、
品質のビクピクチャの観点を持ちながら、
“製品の品質”を顧客視点で肌で感じ、
“プロセスの品質”とバランスをもって
品質保証が進化改善していくようサポートするロールである。

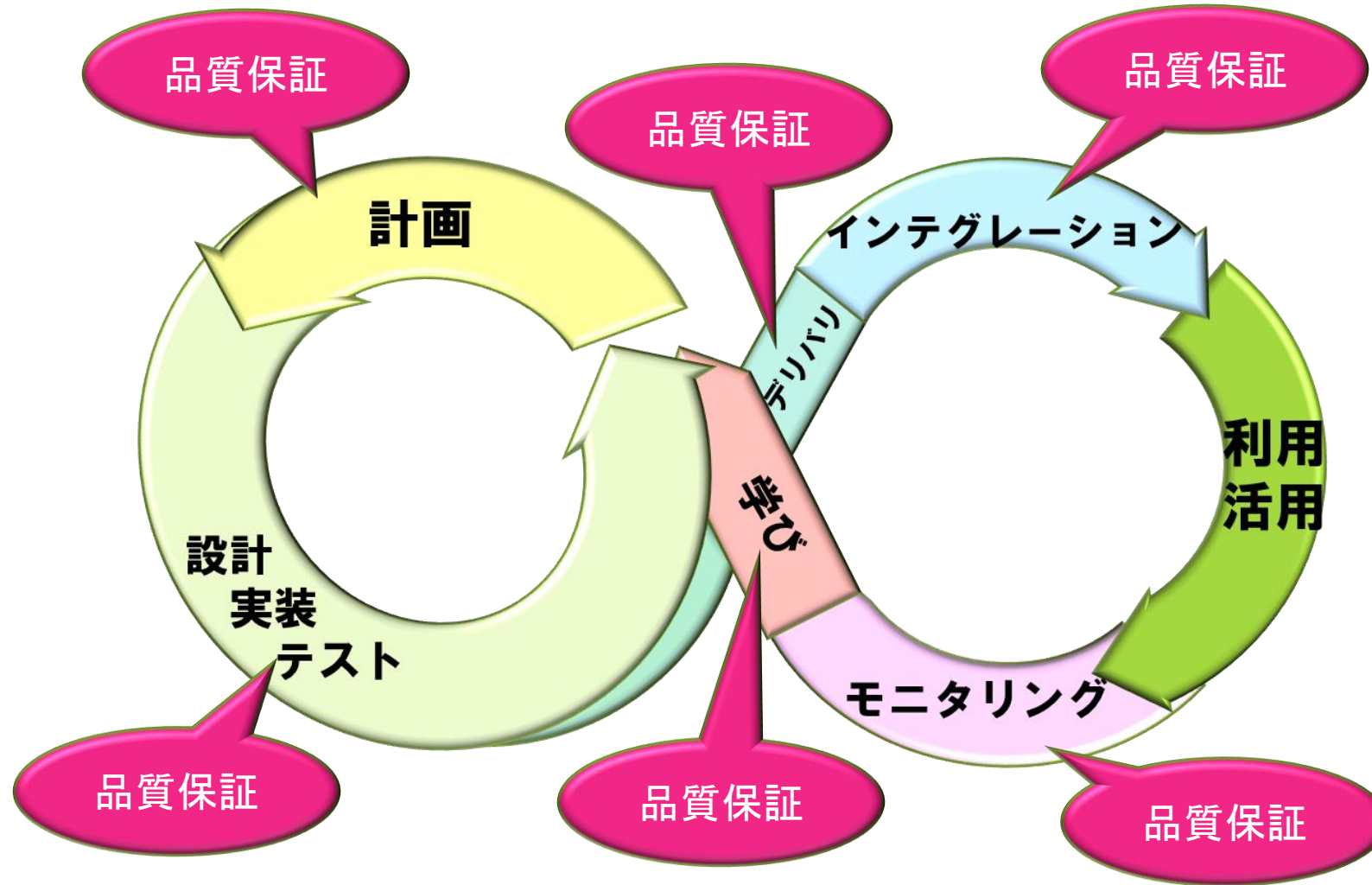


Testing in DevOps World, Amir Ghahrai, 2019, <https://devqa.io/testing-in-devops/>

バリューストリーム：顧客の価値を組み込んでいく活動の流れ



アジャイル・クォリティ



ご清聴ありがとうございました

サイボウズ株式会社

開発本部 アジャイルクオリティ 永田 敦

nagnagworld@gmail.com