

# 継続的なバージョンアップを前提としたメトリクス 駆動開発モデルの確立

リコーITソリューションズ株式会社

図司 達

toru.zushi@jp.ricoh.com

## 開発における問題点

クラウドサービスはリリース後も継続的にバージョンアップを行い機能改善を続ける事で完成度を高める。

バージョンアップの内容は期待効果、費用、品質、納期を加味した上で判断するが、ステークホルダー毎に重要視する項目が異なり、かつステークホルダー間で共有されていないため、調整に時間が掛かる。

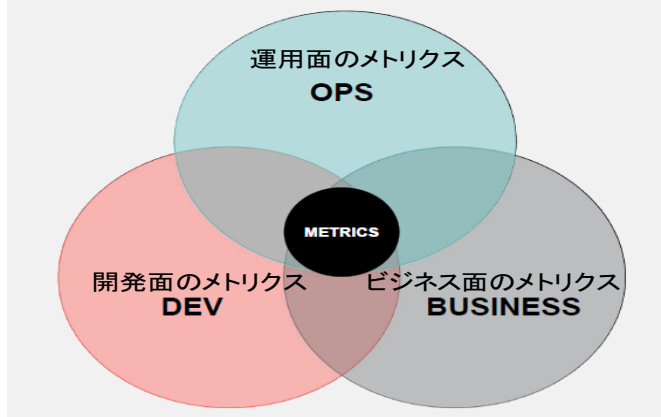
## 手法・ツールの適用による解決

ステークホルダー毎の重要メトリクスを抽出し共有できる仕組みを構築する事で速度/品質を維持したままバージョンアップする手法(メトリクス駆動開発)を提案する。

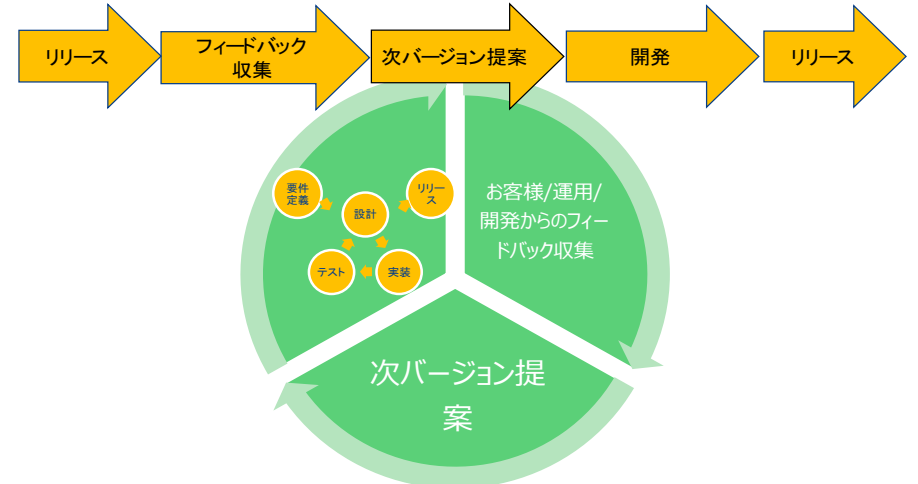
## メトリクス駆動開発

・メトリクス駆動開発  
各観点の指標を定義してリリース基準とする。

### Metrics Driven Development (MDD)



・メトリクス駆動開発の適用範囲  
初回リリース後、フィードバックを次バージョン反映する部分。



## メトリクスの定義

従来のDevopsの指標にビジネス観点のメトリクスを追加してビジネス/開発/運用の観点からバージョンアップの妥当性を判断する。

開発・CI	QA	デプロイ	リリース	運用	ビジネス
開発のリードタイム	非稼働時間	デプロイのリードタイム	リリース頻度	MTTR	従業員の業務負担削減による経費削減
欠陥・ビルド失敗・システムダウンに伴う再作業	検出・見通した欠陥の割合、および欠陥の影響度合い	デプロイ頻度と掛かる時間	リリース毎の時間とコストの割合	システムダウン時のコストと頻度の割合	サービスの価値向上による売上拡大
非稼働時間	MTTD (検知にかかる時間)	システム変更の成功率	(リリース成功の)予測可能性	営業時間後の緊急呼び出しの頻度	
進行中の作業と技術負債		MTTR		性能と利用時間の割合	
サイクル・タイム					

## まとめ・考察

ビジネス観点も踏まえた、メトリクスを定義し、ステークホルダー間で共有する仕組みを構築した上で、リリース基準を決める事で速度/品質を維持したままサービスを提供する手法を提案した。

また、上記の手法を実際の案件に適用しビジネス観点のメトリクスを取得/検証する仕組みを構築した。その結果、サンプル数は少ないものの、メトリクス駆動開発の有効性を検証する事が出来た。

今後の展開として、今回の実験ではサンプル数が少ないため、より精度を高めるためにサンプル数を増やして検証する必要がある。