INTEC
TIS INTEC Group
Go Beyond

ソースコードの品質向上に向けたレビュープロセスの改善について

～プルリクエストを利用したソースコードレビュー～

株式会社インテック
2018/10/11
ネットワークソリューション部
木村　慎吾

INTEC TIS INTEC Group Go Beyond

本日の目的

■主題

・新しい方法（プルリクエスト）でレビュープロセスを改善した結果のご紹介

■対象

・マネージャー・リーダ
・エンタプライズ系

※Web系開発者の中ではすでにデファクトの方法

自己紹介

木村慎吾

インテック
ネットワークソリューション部

スペシャリスト（ソフトウェア開発・ID管理）

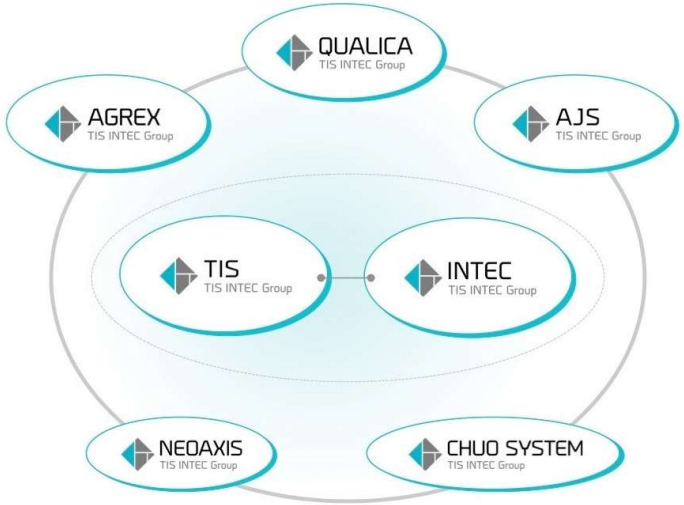
開発との関わり
- 主に開発チームのリーダとして従事
- アジャイル開発を実施

株式会社インテック　について

私たちTISインテックグループは、**システムインテグレーター**として第2位グループの売上規模を誇る、
独立系・プライムコントラクターです。グループ52社、従業員数約20,000人の規模を有しております。

2018年3月期　実績

売上高　4,056億円
営業利益　327億円



株式会社インテック

本社所在地	富山県富山市（本社） 東京都江東区（東京本社）
設立日	1964年1月11日
資本金	208億30百万円（2018年4月1日現在）
売上高	1,153億35百万円（2018年3月期）
従業員数	3,655名（2018年6月1日現在）

INTEC TIS INTEC Group | Go Beyond

ネットワーク＆アウトソーシング事業本部

EINS WAVE of Everything

いつでも、どこでも、お客さまに

SaaS Software as a Service
- EDIアウトソーシングサービス **EINS/EDI-HubNex**
- マーケティング基盤 **Callノート** **Callクレヨン**

PaaS Platform as a Service
- ID統合管理サービス **EINS/IAM**

IaaS Infrastructure as a Service
- クラウドサービス **EINS/SPS**
- データバックアップサービス **EINS/BRS**

データセンター
- 高機能データセンター **EINS/VDC**

Security
- エンドポイントセキュリティサービス **EINS/ISS EPSecurity**
- 不正アクセス監視サービス **EINS/MSS+**
- セキュリティ診断サービス **EINS/SVA+**
- 電子証明書発行サービス **EINS/PKI+**

Network
- データセンター・クラウド・自社システムを結ぶ堅固なネットワークサービス **EINS/MOW DCAN**

今回の現場について

- パッケージ製品の開発チーム
- アジャイル開発
- 開発開始から9年経過
- チーム人数は3名～10名程度

背景

■悩み

・長年の開発によって、バグ修正や機能追加の時間が以前より掛かることが多くなってきた

■原因

・対応したい個所だけでなく関連する箇所など影響範囲が拡大してきた
・過去の開発においてソースレビュー不足でソースコードの品質に問題がでてきた

■対応

・ソースレビュー不足の改善

改善へ

ソースレビュー不足改善に向けて次の２点をポイントにした。

・レビュータイミング
・品質向上に向けた意識

改善へ（レビュータイミング）

レビュータイミングとは具体的に以下の改善を目指した。

・多くのメンバが締め切りまでローカル環境おくため、
マスタにぎりぎりにコミットされるので最低限のレビューだけになる

・リリース日が時間的プレッシャーとなり、
時間優先となりレビューなしで進められてしまう

改善へ（品質向上に向けた意識）

品質向上に向けた意識とは具体的に以下を目指した。

・レビューによって品質を作り込む
・レビューによってチームの暗黙知を共有させる

INTEC TIS INTEC Group Go Beyond

経緯

ソースレビュー不足改善が行われるまでに以下の経緯があった。

1 ：改善への悩み
2 ：プルリクエストとの出会い

経緯（１：改善への悩み）

過去にもレビュープロセスを改善したが継続しないことがあった。

・忙しくなるとレビューをしなくなる
・ソースをためてレビューするので一部集中になる

⇒仕組み（プロセス）として確立されていないと逸脱しやすい。。。

経緯（２：プルリクエストとの出会い）

プルリクエストという救世主登場

・開発者の中で流行ってきているやり方
・開発者からのボトムアップ的な提案を受けた

⇒この方法はプロセスに組み込みやすい

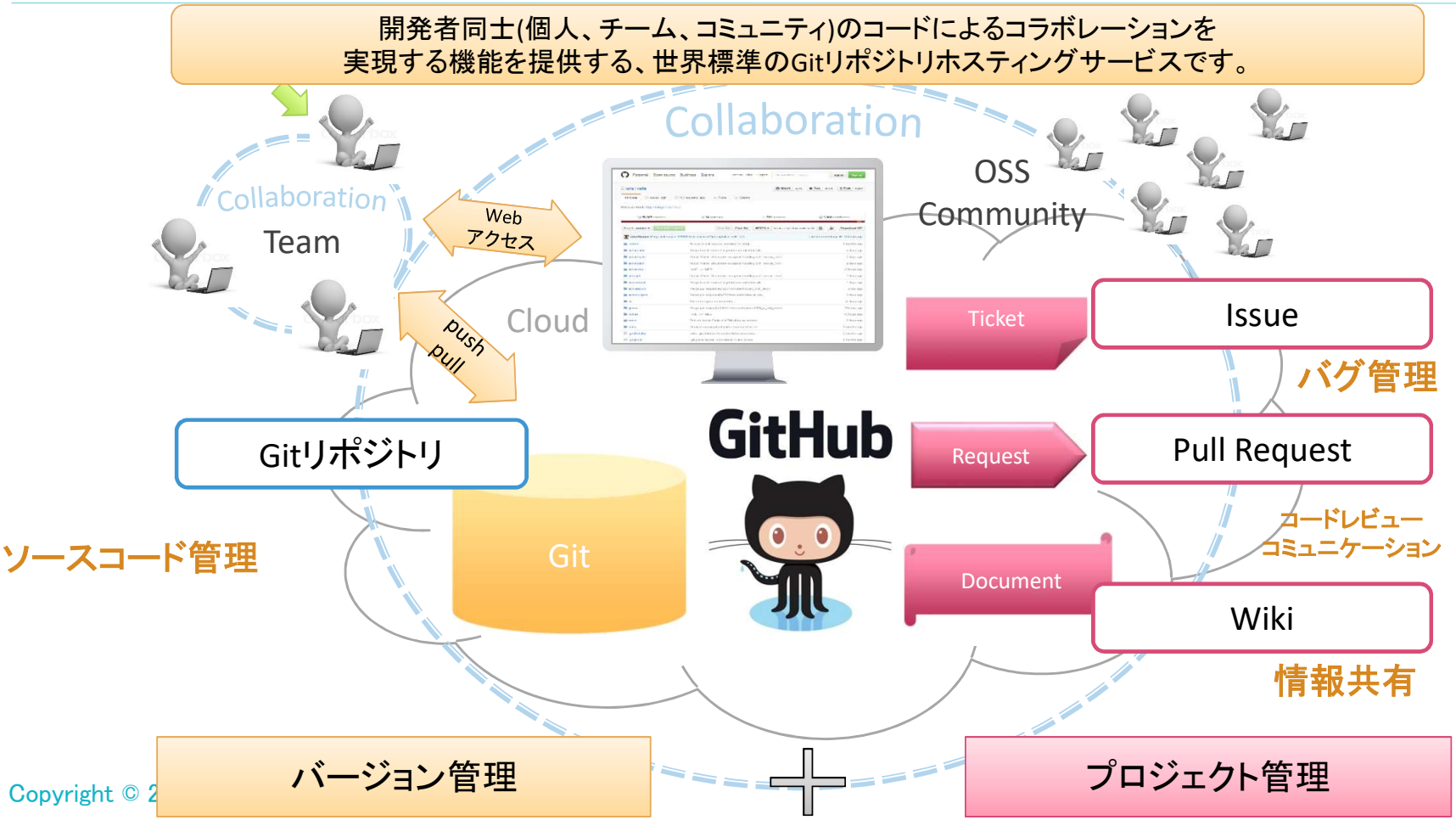
INTEC TIS INTEC Group Go Beyond

プルリクエストの概要

■方法について
・変更点を提出し、認めてもらうことで採用される

■機能について
・バージョン管理ツール（Git）そのもの機能ではない
・ツール・サービス（GitHub）などを使う必要がある

プルリクエストの概要（GitHubについて）



プルリクエストとは

プルリクエストの詳細ついて、以下の観点で説明する。

- 具体的な方法
- 効果
- 懸念点

プルリクエストとは（具体的な方法）

■具体的な方法

0：開発者が自分の環境で作成したコードを直接メインのマスタにコミットしない

1：一度別のブランチを作成し、その内容をアップする

2：アップしたことは他の開発者に通達される

3：その内容を他の開発者がレビューをして、問題なければマスタに反映させる

※これによって、マスタへコミットする前にソース公開しやすくなり、レビューできる
※先にマスタにコミットされると、ちゃんとしなければという意識でなかなかコミットできない

プルリクエストとは（効果）

■効果

・アップされたソースコードの変更点などがわかりやすく表示される
・コメントを簡単に書けるのでレビューがやりやすくなる
・リアルタイムにレビューできるのでコミュニケーションも活発になる

プルリクエストとは（懸念点）

■懸念点

・プルリクエストを採用してもルール次第でいろいろできる

■私たちの工夫のご紹介

チームとしてルール作りが必要になり、以下の工夫をした。

・継続インテグレーション（CI）のテストが通過ことでマスタに反映できるようにする
・さらに、最低1名がレビューしOKを出すことで、マスタに反映できるようにする

実現方法

本番環境でいきなり導入・運用するが心配だったため、
プロジェクトへの導入は次の２ステップで進めることにした。

１：プルリクエストを利用した開発プロセスの習得
２：バージョン管理ツール・開発ツール変更の検討

実現方法 （１：プルリクエストを利用した開発プロセスの習得）

■懸念点

・チームメンバ全員がちゃんと理解できていないと、逆にバージョン管理環境を荒らしてしまう

■対応

・準備段階のフェーズを設けた

■実施

・まずは基本的な使い方の勉強会を2週間ほど実施して、概念やツールの使い方を学習させた
・その後、製品開発から派生した小さなプロジェクト使って検証を行った

■結果

・これらによりプロセスの概念を押さえることができ、製品開発へ導入する目途がたった

実現方法（２：バージョン管理ツール・開発ツール変更の検討）

■懸念点

・プルリクエストを採用するためには利用しているツールを変更する必要があった
・ソース自体を新たなリポジトリに移行しなければならなかった

■現状

・開発のバージョン開発ツールにSubversion（SVN）を利用していた

■対策

・新しい環境としてGit対応のGitHubの利用を検討していたため、そのGitHubへの移行が必要になった
・SVNからGitへ変換するツールがあり、それを活用して移行できることがわかった

■結果

・移行検討期間で3週間程度使ったが、移行自体はスムーズに完了し、すぐに利用可能な環境を作れた

結果

約半年運用した結果について、以下の観点でまとめてみた。

- 効果
- 変化
- 問題点

INTEC TIS INTEC Group Go Beyond

結果（効果）

■効果：１　レビュー効率が上がった

・リアルタイムにコード変更をWeb上で簡単に確認できるようになり頻度が上がった
・問題個所（コード）にピンポイントで指摘でき、効率よくレビューできるようになった
・作業ごとにソースコードと指摘がまとまっているため、過去の検討事項も見つけやすくなった

■効果：２　早めにレビューできるようになった

・途中段階でソースコードをアップできるため、早めのレビューが出来るようになった
・これによって、教育的な指摘も多くできるようなった

結果（変化）

■ 変化

- OSSに対してパッチ提供できるメンバも出た
  - プルリクエストを使った開発がOSS界隈ではデファクトスタンダード方法
  - お作法を覚えたことで対応できるようになった
  - このように技術力の向上においても大きな影響を与えることができた

結果（問題点）

実際に運用していく中で課題も発生した。
その中でよく起こりそうな２つの事例を紹介する。

結果（問題点：１）

■問題１

・レビューのやりとりはWeb上の文章になるため、指摘が思ったよりきつく感じてしまう

■改善１

・文章だとニュアンスが伝わりづらいので語句を工夫するようにした
  ・文字だと伝わりにくい気持ち的なものは☺マークなどいれて、気持ちの共有をする
  ・指摘が一方的にならないように、「対応ありがとう」とか一言いれるようにする

結果（問題点：２）

■問題２

・テスト時間が掛かりすぎるため、マスタ反映への待ち時間が増えた
  ・CIで設定されているテストを通過しないとコミットできないようにしたため、レビューがOKでもマスタに反映できない
  ・複数の作業を同時にしていると反映待ちが増えてしまった

■改善２

・以下の工夫を実施した
  ・速度が向上するようにCI環境自体をリプレイスした
  ・テスト範囲を調整し、不必要なテストの見直しを実施した

INTEC TIS INTEC Group Go Beyond

まとめ

改善のポイントに上げた2点は以下のように対応できたと考えている。

１：レビュータイミング

・制約を入れることでレビューなしでマスタに反映されるソースコードが0になった
・マスタを汚さないため、早めにソースが公開でき、ぎりぎりでの公開はなくなった

２：品質向上への意識

・レビュー記録が書きやすくなったため、後輩への指導が簡単となり教育場として機能した
・プルリクエストのコメントとして仕様を決めた経緯などもかけるため、
修正の意図が後からでもわかりやすくなった

INTEC TIS INTEC Group Go Beyond

最後に

プルリクエストを用いた開発の導入をぜひ検討してみてください。

導入コストがかかるなどいいことばかりではありませんが
それを上回る以下のメリットが大きいと考えています。

・レビューを自然に開発プロセスに組み込める
・今どきの開発手法によるモチベーションアップ