GitHub Enterprise の活用事例

GMO ペパボの取り組み

GMOペパボ株式会社



自己紹介

self.introduce



=>
{
  name: “SHIBATA Hiroshi”,
  nickname: “hsbt”,
  title: “Chief engineer at GMO Pepabo, Inc.”,
  commit_bits: [“ruby”, “rake”, “rubygems”, “rdoc”, “tdiary”, “hiki”, “railsgirls”, “railsgirls-jp”, “jenkins”],
  sites: [“ruby-lang.org”, “rubyci.com”, “railsgirls.com”, “railsgirls.jp”],
}

会社組織

GMO ペパボ

- 本社: 東京都渋谷区、支社: 福岡県福岡市
- 設立日: 2003年1月10日
- 従業員数: 231名
- 企業理念: “もっとおもしろくできる”
- ミッション: “インターネットで可能性をつなげる、ひろげる”



事業部制

- サービスを担当する事業部
  - 本社事業部
  - EC事業部
  - ホスティング事業部
  - …
- バックオフィスを担当する事業部
  - 経営管理部
  - 経営戦略部

GitHub Enterprise

ペパボの GHE 導入の歴史(1)

- 2012: trac, redmine などサービスやプロジェクトごとに自由に選択
- 2012: 一部のプロジェクトで github.com/paperboy-* という org アカウントを作成して github の利用を開始
- 2013: 社内の半分程度のサービスが github.com を利用開始
- 2013: 一部のプロジェクトがセキュリティの問題で github.com を利用できないため GitHub Enterprise を導入

GitHub Enterprise 導入時の状況

- 社内であまっているサーバーに linux と virtualbox をインストールし headless で起動
- 監視については nagios/munin を自前で構築
- 部署として面倒を見ていくわけではなく、技術基盤チームの1エンジニアが導入と運用を担当

ペパボの GHE 導入の歴史(2)

- 2014: 社内の全てのサービスが github.com か GHE のいずれかを利用
- 2014: サービスだけではなくバックオフィスも利用を開始
- 2014: 全社員が GHE のアカウントを保有して情報共有するということに社長から方針決定(緩く)
- 2014: github.com を利用していたサービスもアカウント管理の煩雑さを解消するために GHE へ引っ越し

GHE 集約期の状況

- github.com も GHE も社員全員が広く情報にアクセスできるようにということで全員に write 権限を付与(200人…)
- アカウント管理用のスクリプトを作成し、定期実行

```
def users
    ghe = Octokit::Client.new(:access_token => configuration['access_token'])
    @_users ||= [0, 100, 200, 300, 400].each.with_object([]) do |start, memo|
      ghe.all_users(:since => start).map do |user|
        if !(partner_users + exclude_users).include?(user.login) && user.type == 'User'
          memo << user.login
        end
      end
    end
    @_users
  end
```

ペパボのGHE導入の歴史(3)

- 2015: 全サービスが GHE を用いて開発する状況になる
- 2015: 慢性的な高負荷と virtual box 脱却のために OpenStack 基盤への移設を実行(GHE 2)
- 2015: GHE 2.x への継続的なアップデート中

全社員が使う

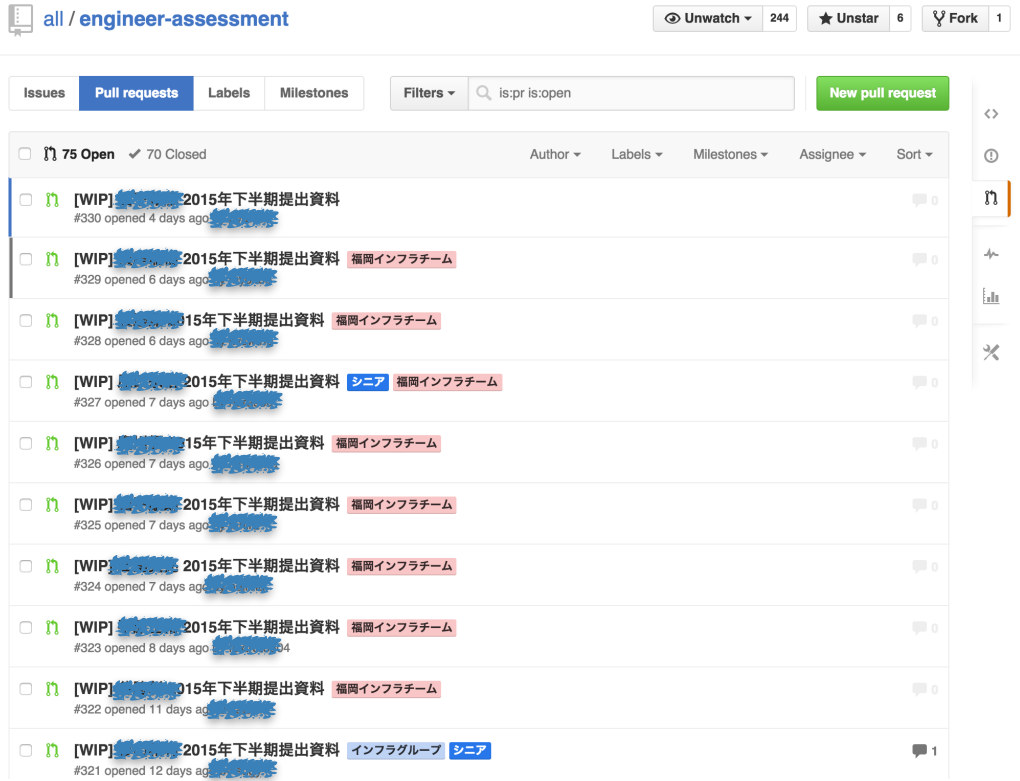
GHE

エンジニア評価資料を pull request で作成

誰でも評価資料を閲覧することができる
誰でも評価資料にコメントすることができる
誰でも評価結果を閲覧することができる

↓

透明性

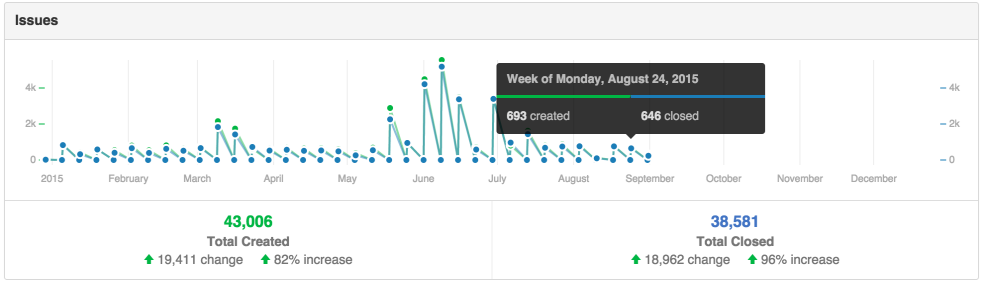


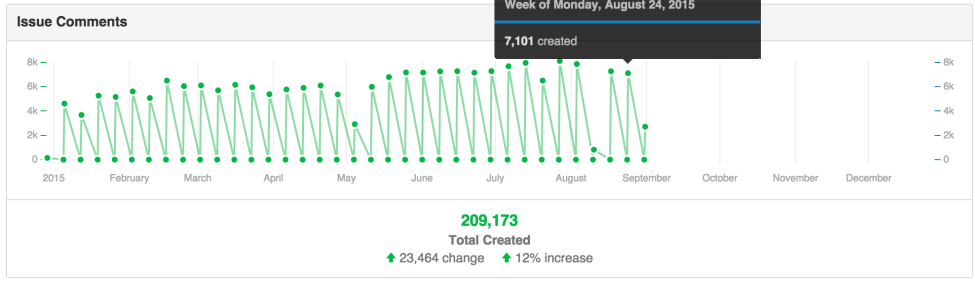
開発とCSと企画のコミュニケーション

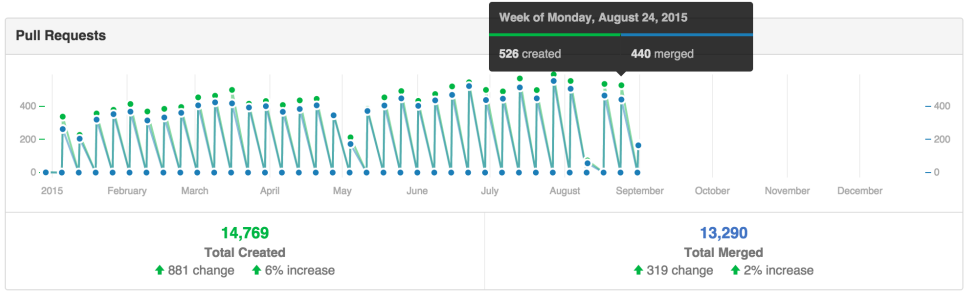


- “仕事”の情報のほとんどを GHE に集約
- 仕事が始まった/終わったということを可視化
- 仕事がどういう状態にあるのかということをはっきりさせる

全社員のアクティビティ = 開発力







1週間の

- issue 作成と close は 650 前後
- issue コメントは 7100
- pull request の作成は 520超、merge は 440

GHEで
開発速度を
上げていく

OpenStack

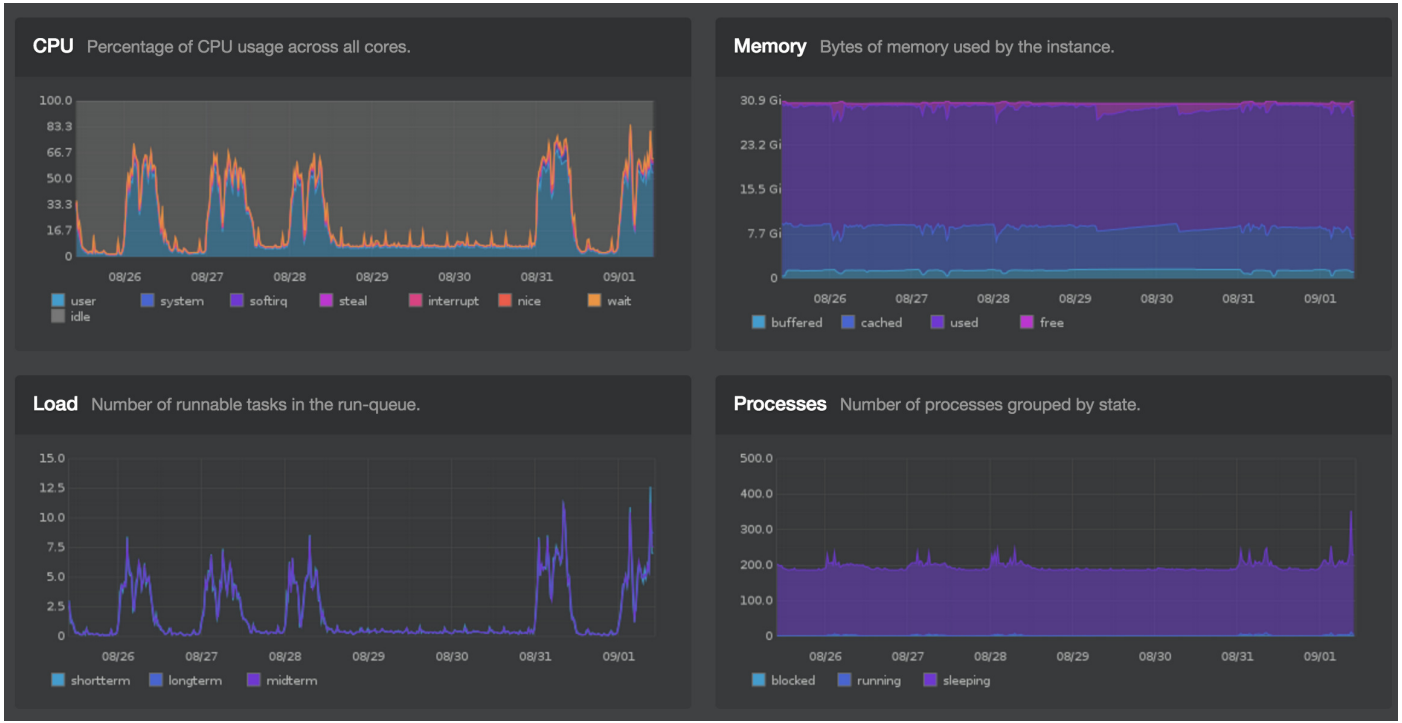
GHE on OpenStack

OpenStack の説明は省略

- vCPU 4, 32GB RAM, 85GB root drive で作成
  - 80GB と手順書に書いてあったが起動できなかったので OpenStack のホストマシンにログインしてデバッグ…
- Cinder(AWS の EBS) を構築して GHE 2.1 で無事 OpenStack で起動

OpenStack での利用状況

利用者増とGHEの普及の広がりにともないリソース不足気味



困っていること

アカウントのステータス管理

全社員が使うようになると社員のステータスと GHE のステータスのミスマッチが発生する

(入社、退職、休職) and (正社員、派遣社員、アルバイト)

社員は全 orgs に API を使って write 付与、派遣社員は... 休職になったら...という作業が若干煩雑

orgs を超えたアカウントのグルーピングがあると嬉しい

情報共有手段と通知

issue, wiki, repository(git) バラバラに情報を記入している

→ qiita や esa のようなドキュメンテーションツールが gist のように使えると統一できそう

notification が default on なので興味がないリポジトリでも write 権限付与の時点でどんどんメールが飛んでくる

→ default off にしたい

良かったこと

GHE のデプロイ、運用の仕組み

- CoreOS のようなボリューム入れ替えアップグレードの仕組み
- db:migrate を job worker で実行するような無人アップグレードの手順
- pt-achiver, pt-online-schema-change のような db オペレーションツールの具体的な使い方

…etc…etc

社員募集中!

技術基盤チーム & 情報システムグループのエンジニア

主なミッション

- 全社横断の開発基盤の構築、運用(OpenStack、GHEなど)
- 社内システムの構築、移設、刷新
- セキュリティ、監査、内部統制と開発スピードのバランスをとったソリューションの構築

毎日 GitHub を使って開発・運用をやりたい人募集してます!