

クックパッドにおける Androidエンジニアの役割と その変遷



DroidKaigi 2/19

ROOM:A 11:30-12:00

ハッシュタグ： #DroidkaigiA



対象者

- **モバイル開発チームを率いている人**
- **参加している人**
- **これから作っていきこうと思っている人**



話すこと

- **少人数1チーム体制時代から多事業部多人数体制時代までAndroidエンジニアの役割が変遷していくなかでチームが直面した課題とそれを解決する為に構築してきた開発プロセスや習慣、仕組み**
- **すぐに導入できそうなものを中心に紹介**
- **※コードは出てきません。**



クックパッドにおける Androidエンジニアの役割と その変遷 を支えた開発プロセス



DroidKaigi 2/19

ROOM:A 11:30-12:00

ハッシュタグ： #droidkaigiA



自己紹介

- **八木 俊広 (@sys1yagi)**
 - ▶ **Android Engineer**
 - ▶ **クックパッド株式会社 買物情報事業部 所属**
 - ▶ **2013年 入社**
- **最近の趣味: Githubに草を生やし続ける**



レシピ関連サービス

みんなのレシピ

プロのレシピ

献立

料理動画

おいしい健康

話題のキッチン

プレミアムサービス

人気順検索

レシピランキング

殿堂入りレシピ

専門家厳選レシピ

プレミアム献立

もっと見る



2月18日の
おすすめ

母の味♪白菜と厚揚げの卵とじ

miyuki12



特売品でお得に作ろう！

サイズいろいろ卵 (10コ入) **108円(税抜)** タイヨー 東陽町店

ニュース



エビ好きも満足栄養満点エビ豆腐



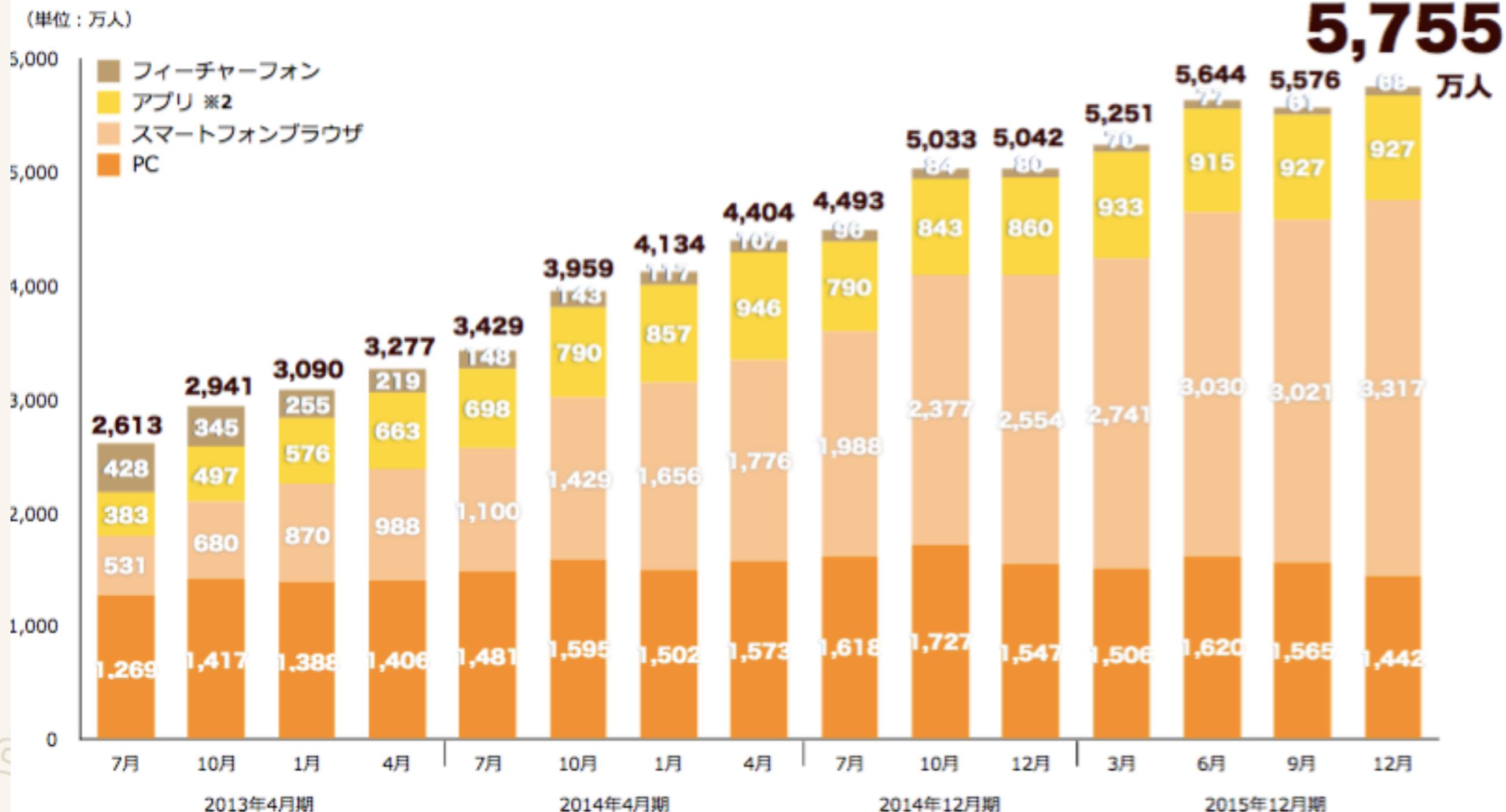
旬の

野菜や食材がそろう
マーケット



国内の「クックパッド」の利用状況

月間利用者数 ※1

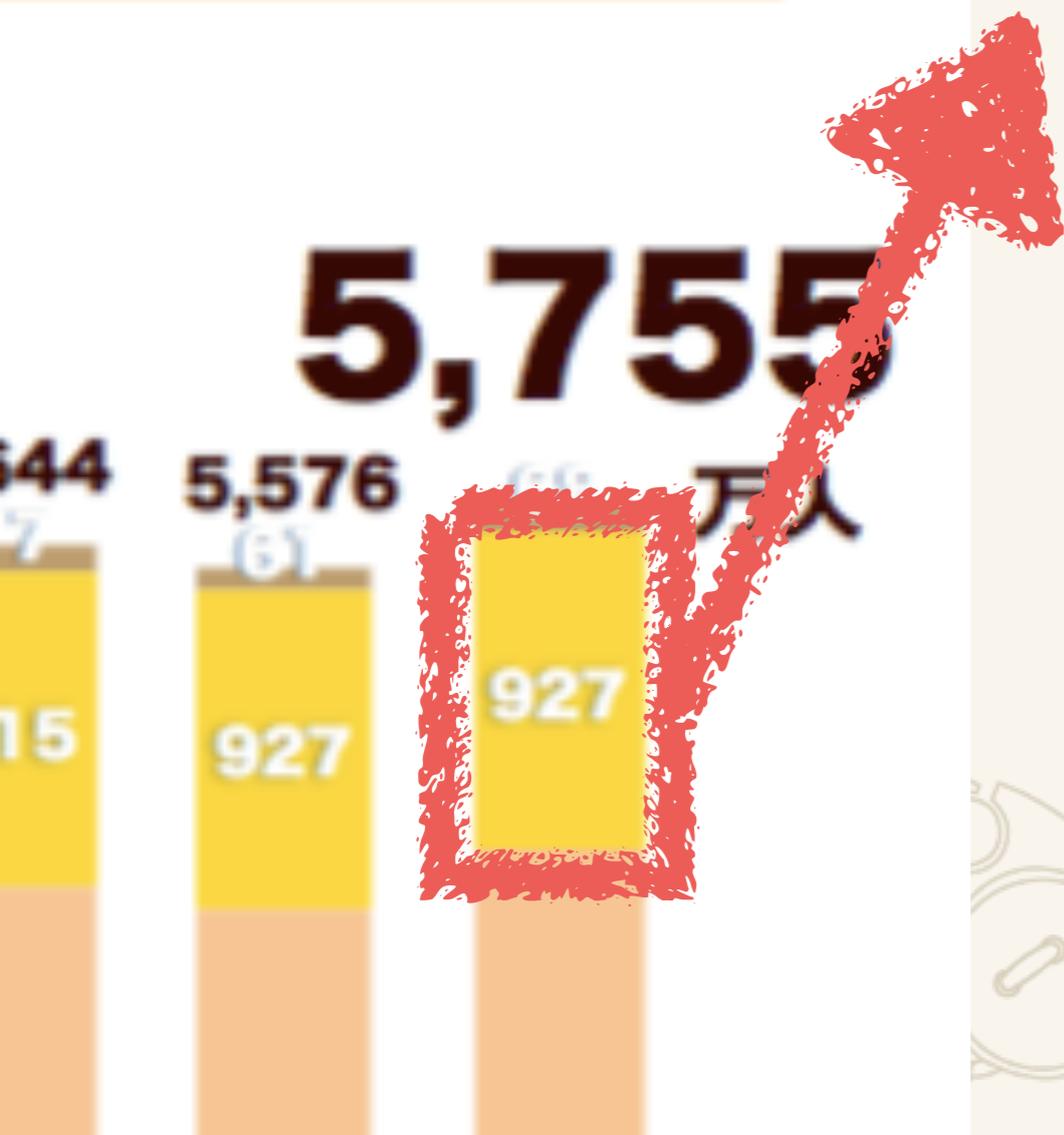


※1 月間利用者数はブラウザベースまたは端末ベースにより集計。

※2 2014年7月より計測方法を変更。

クックパッド Android

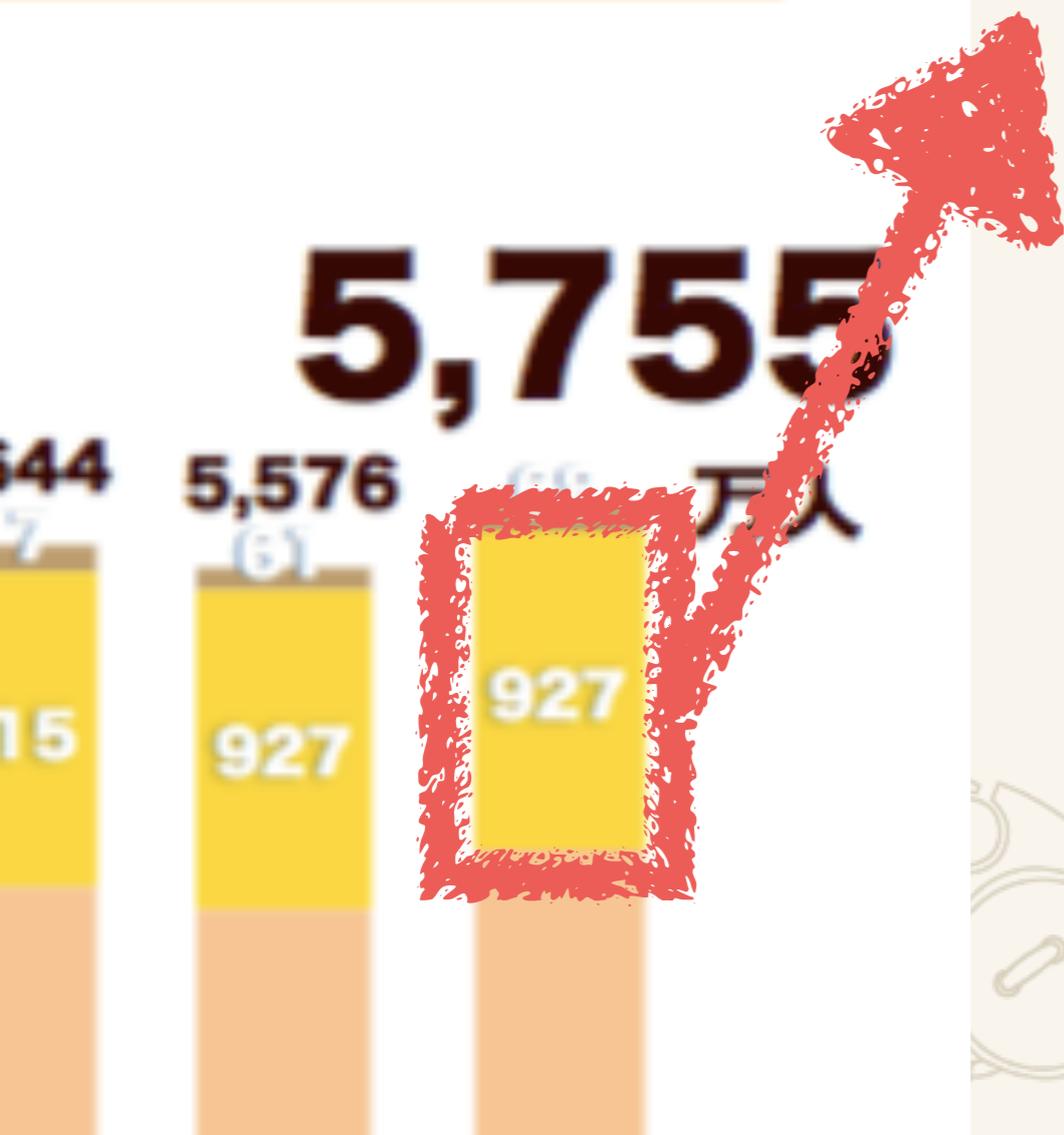
mobile app全体の
月間利用者数(MAU) 927万



※2015年12月時点

クックパッド Android

mobile app全体の
月間利用者数(MAU) 927万



の40%くらいがAndroid



※2015年12月時点

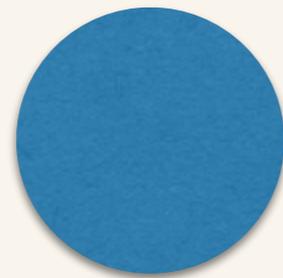
少人数体制時代

- 2013年10月~
- ハイブリッドアプリ->nativeアプリ化
- フルスクラッチ
- Androidエンジニア3人

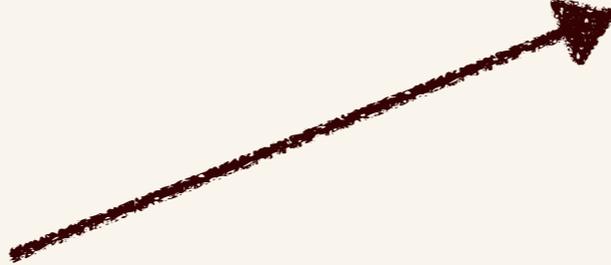
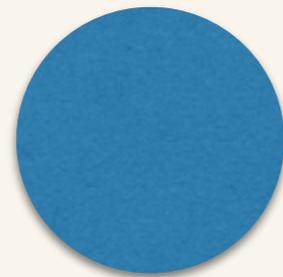
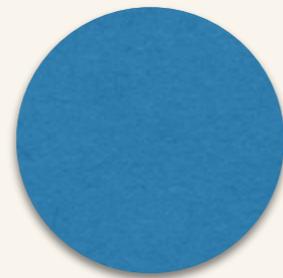


少人数体制時代

mobile group



dev



cookpad app



やっていたこと

- **Webで提供している機能をnativeに移植する**
- **必要なAPIの洗い出し、仕様作成、調整**
- **Android特有のデザインと体験を作る**
- **大量のissueをひたすら消化**



3人の開発スタイル

- **Github Enterpriseでissue管理**
- **Pull Request & Review**
- **CI(Jenkins)でのmaster branchビルド**
- **Deploygateへdebug版を配信**



Androidエンジニアとしての役割

- **Android appの開発力を発揮する**



2014年3月 リリリース



- 無事リリース
- 大きな問題はなし
- 滞在時間 up

2014年3月25日火曜日
■ 平均セッション時間: 00:02:26
■ Android: 00:02:26



2014年3月25日火曜日
■ 平均セッション時間: 00:02:58
■ Android: 00:02:58



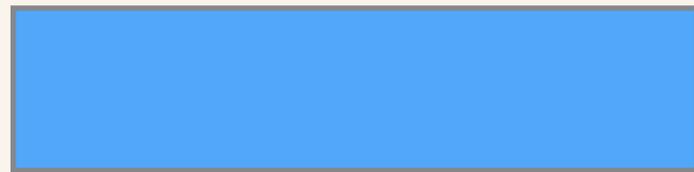
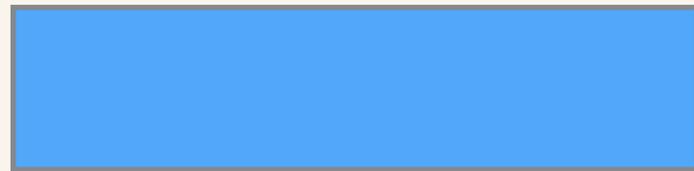
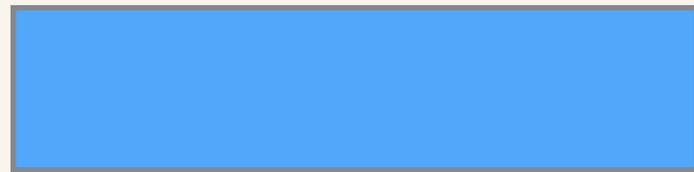
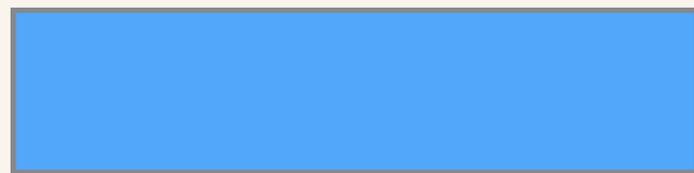
2014年3月 モバイルファースト室設立



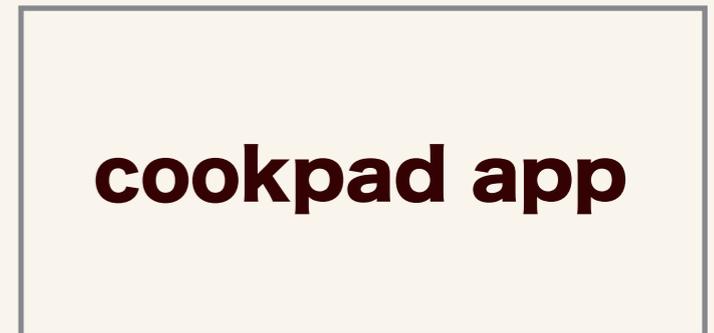
对多部署体制时代

division

mobile first team



dev



request

knowlage



当時の私の考え方

たぶん動くと思うから
リリースしようぜ



たぶん動くと思うから リリースしようぜ

- **どんどん機能を開発し**
- **bugが見つかったらガンガン直し**
- **スピーディにリリースする**
- **それが価値を提供する事に繋がる**



しかし数回アップデートすると...

Crashlytics



しかし数回アップデートすると…

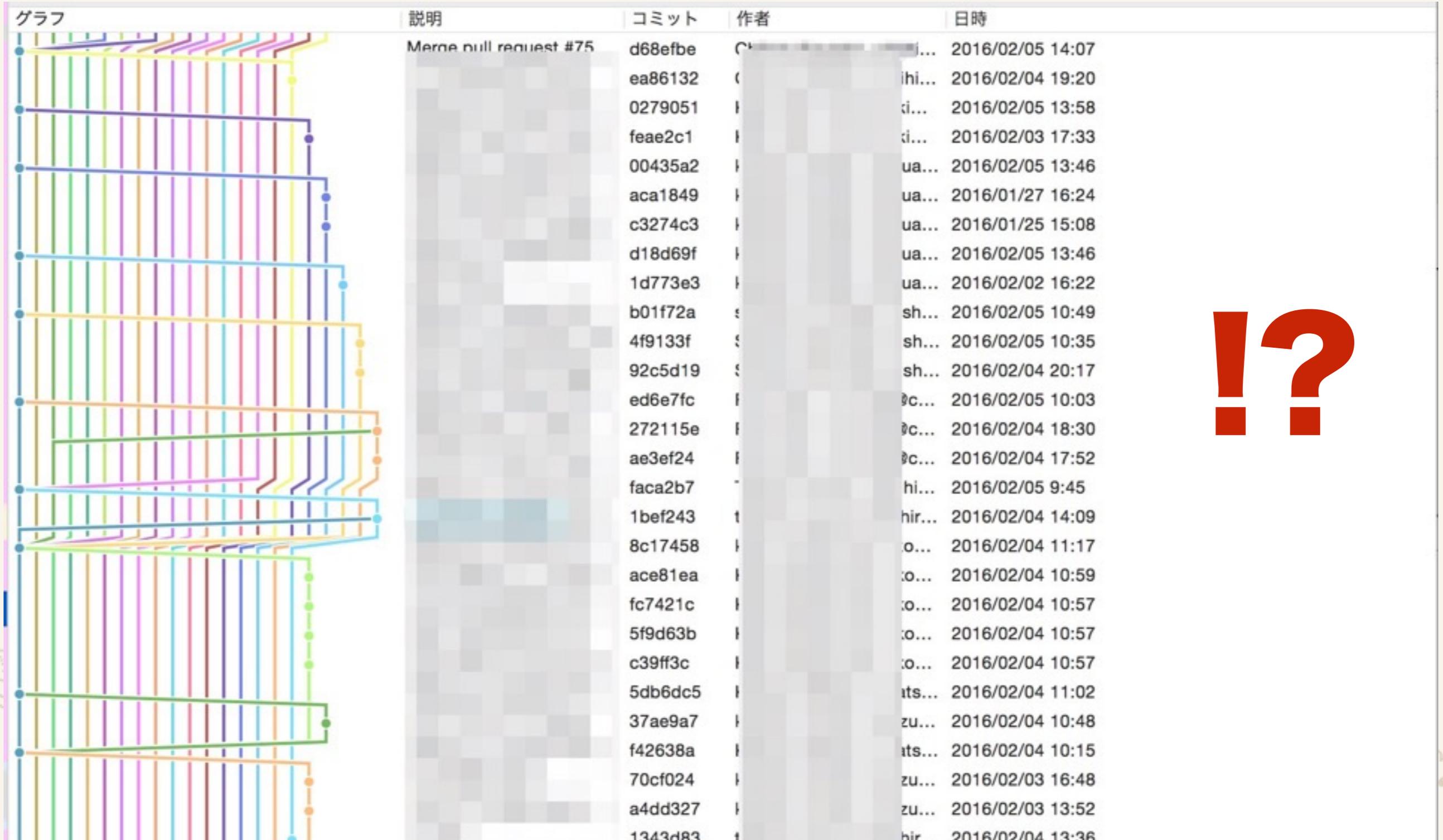
軽微なbugでも命取り…

14000 crash / day

12000 UUに影響

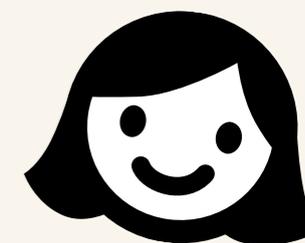


何をいつ変更したっけ？



いつできる？

〇〇機能を入りたいので予定を立てたい
けど次のバージョンはいつでますか？



わかりません…



課題

- **アップデートするとすぐに数十万人が利用しはじめる、軽微なbug
が大きな影響を与えてしまう**
- **mobile appはWebサービスとは異なり、一回リリースすると
取り消せない。品質によりシビアにならなければならない。**
- **品質upのために作業に没頭しがちでバージョン間の差分について
意識が回らない、告知やドキュメント等のコミュニケーションを
する暇がない**
- **リリースサイクルが明確ではないので各部署で予定がたたない**
- **あふれるissue**



たぶん動くと思っから
リリースしようぜ



Androidエンジニアとしての役割+

- 安定的なリリースを行うアプリケーション開発プロセスを構築し、多人数での開発を支える
- 品質向上の仕組み作り、ナレッジの蓄積をする
- でもスピードは下げない



リリース毎にKPTによる 振り返りを開始



<http://techlife.cookpad.com/entry/2014/10/31/093305>

リリース毎にKPTによる 振り返りを開始



**継続的な
改善**

- **Keep:**
 - 簡単な事でも出来たことを
どんどん出す
- **Problem:**
 - 解決すべき課題を洗い出す
- **Try:**
 - **Problem**を解決する具体的
な行動を定義する



無法地帯だった開発プロセスをコントロールできるルールを作る

リリーススケジュールの定義

- **開発期間：10営業日**
- **テスト：3営業日**
- **10%リリース 3営業日**
- **20%リリース 2営業日**
- **50%リリース 1営業日**
- **100%リリース 1営業日**



リリーススケジュールの定義

- **1リリース20営業日**
- **10%リリースから次の開発期間が始まる。**
リリースとリリースの間は2週間
- **頻度が多すぎず少なすぎないバランス**
- **スケジュールに沿って変更を考える**



開発終了日問題

開発開始

CF



開発期間 10営業日



開発終了日問題

開発開始

CF



開発期間 10営業日



Pull Request (^q^)



開発終了日問題

開発開始

CF



開発期間 10営業日



Pull Request (^q^)

コードフリーズとは



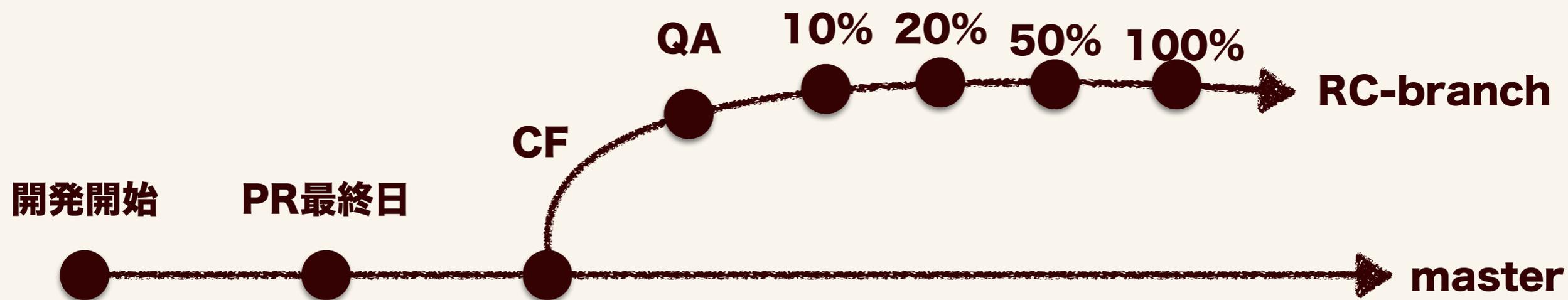
PR締め切りの導入



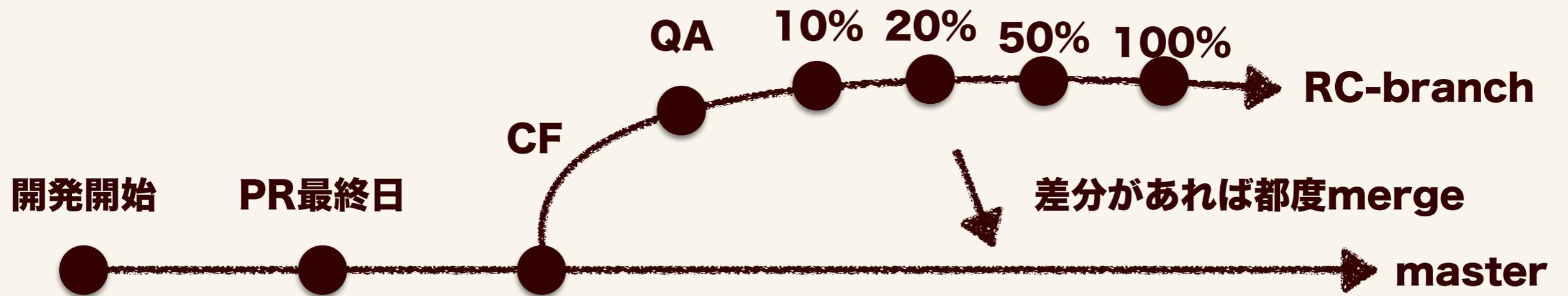
- レビュー時間、修正を考慮してPR締め切り日を開発終了3営業日前に置いた



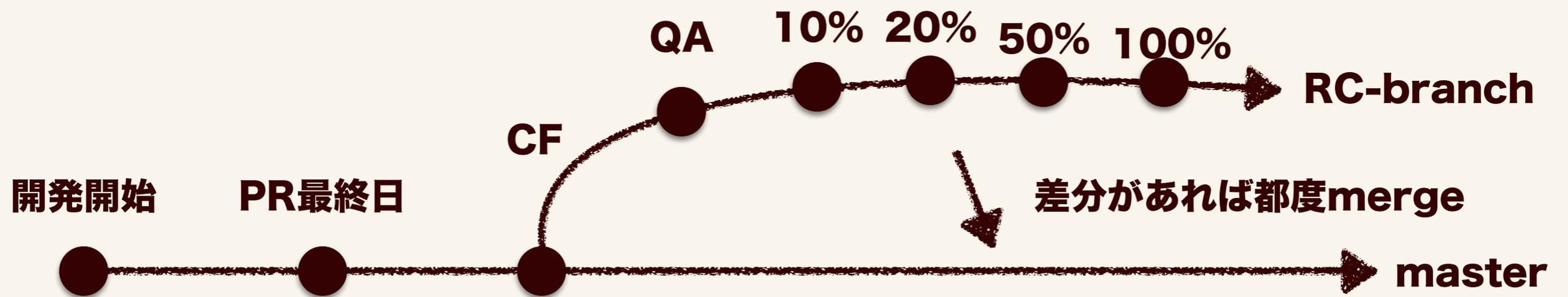
gitのブランチングの方針を定義



gitのブランチングの方針を定義



gitのブランチングの方針を定義



基本的に常にmasterを向いて開発すれば良い



変更の全てをissueにする

- **githubにバージョン毎のマイルストーンを作る**
- **部署ラベルを作成**
- **各バージョンに入れたい変更を全てissue化する**
- **PR締め切り前などに進捗を確認**
- **マイルストーンに入っているissueに関する変更を全てテスト**



コントロールラブルなリリース

- 部署間でのスケジュール調整が基本的に不要になる
- 直前の駆け込みPR等の防止、治安維持
- 変更差分がマイルストーンを見ればわかる
- 安定した開発が可能になった

品質を上げる
運用コストを下げる



リリース前

リリース後

- リリーススケジュールによる差分の入るタイミングの統制
- レビューによる検出
- マイルストーンによる全差分の可視化
- マイルストーン差分を中心に網羅的なテスト

?



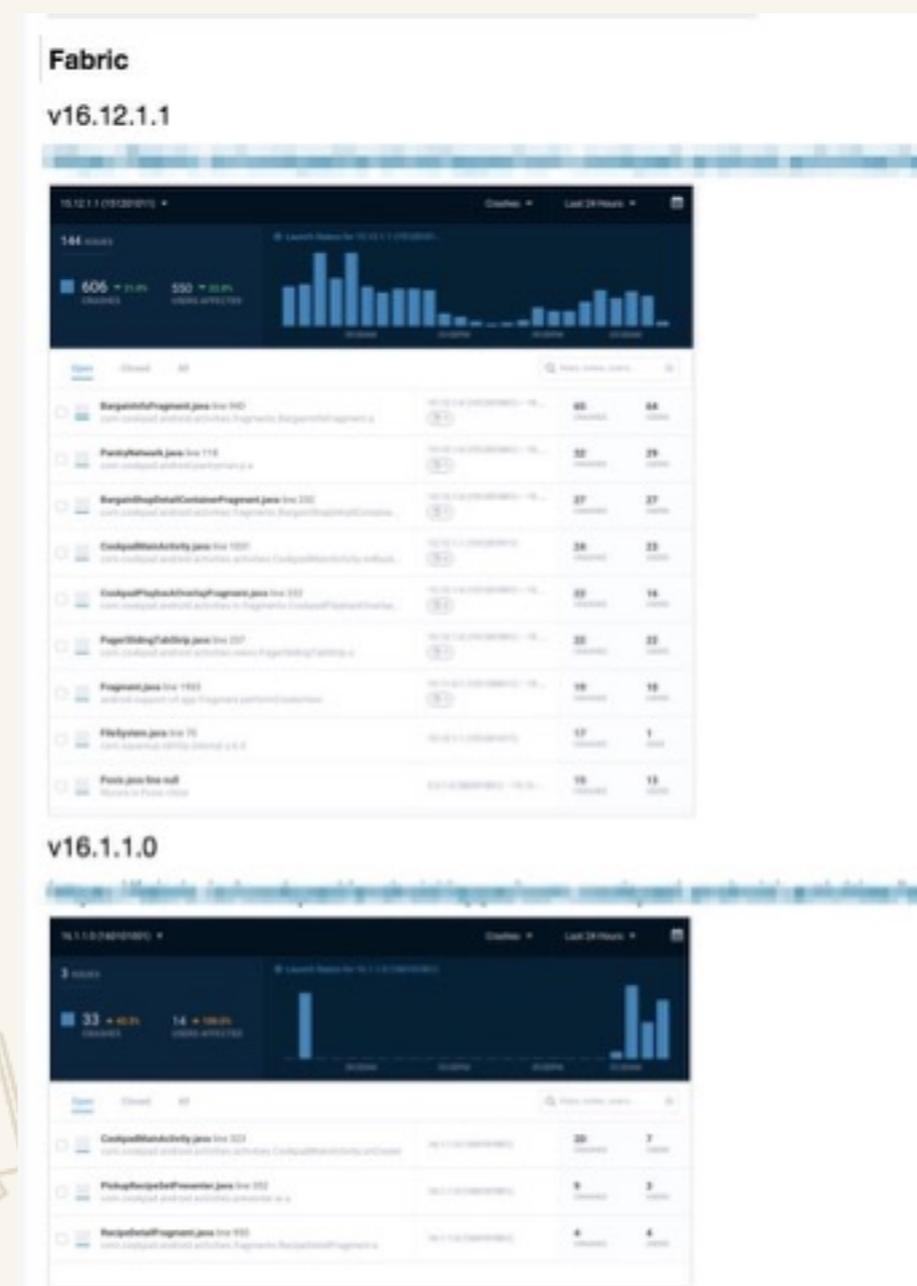
リリース後できるだけ早く見つけたい事

- **特定ケースでだけ再現するクラッシュ**
- **APIの予期せぬエラー**
 - ▶ **データ出向をミスって404**
 - ▶ **変なトークンでリクエストして401(ログアウト)**
- **リリースした機能やなくした機能に対する声**



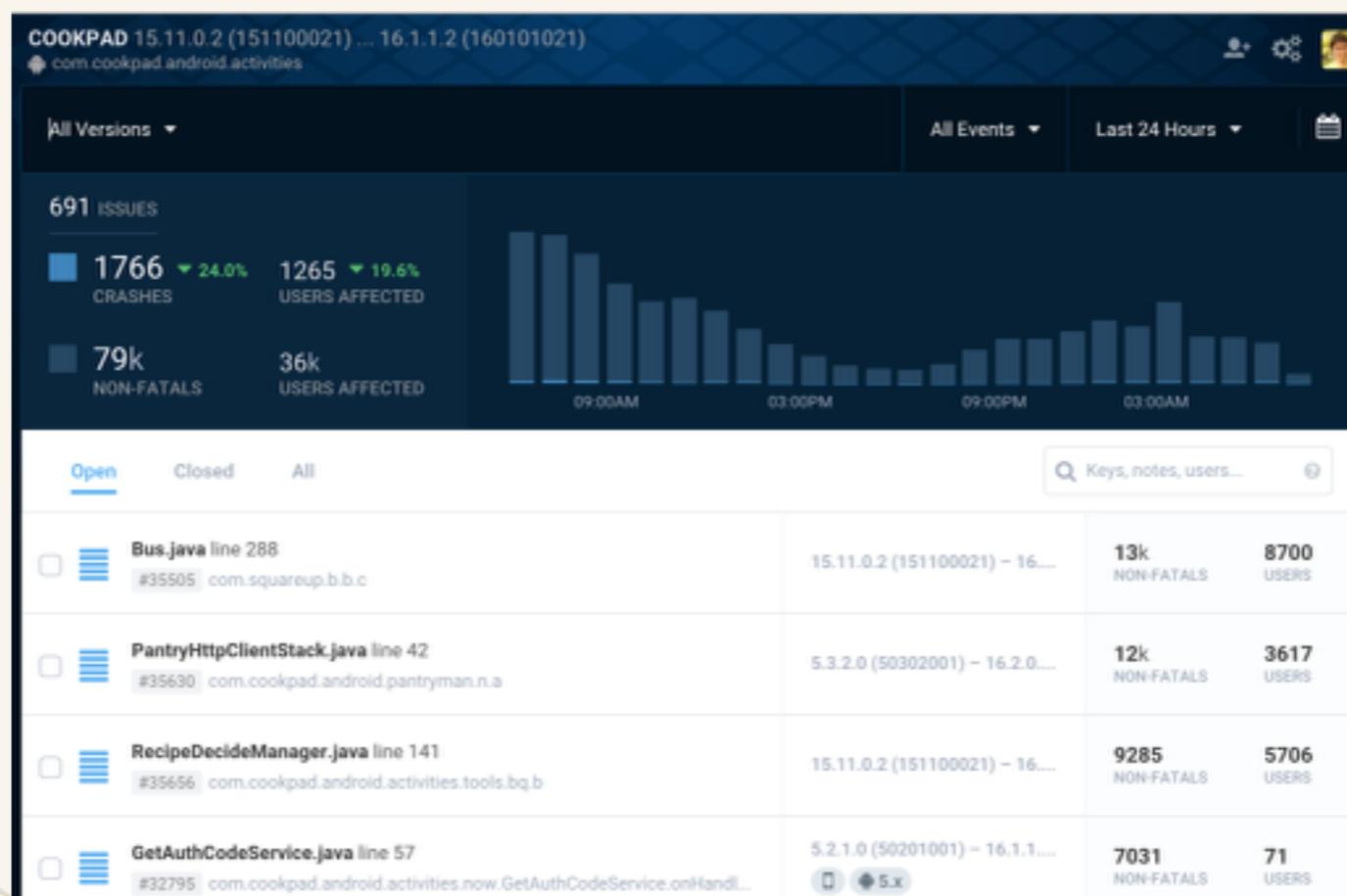
リリース後監視としてまとめてやる

リリースした3時間後、朝、夕方を3日間、
各種チェック結果を記録する



予期せぬクラッシュを素早く検出

Google AnalyticsのリアルタイムサマリーとCrashlyticsの状況を見る。



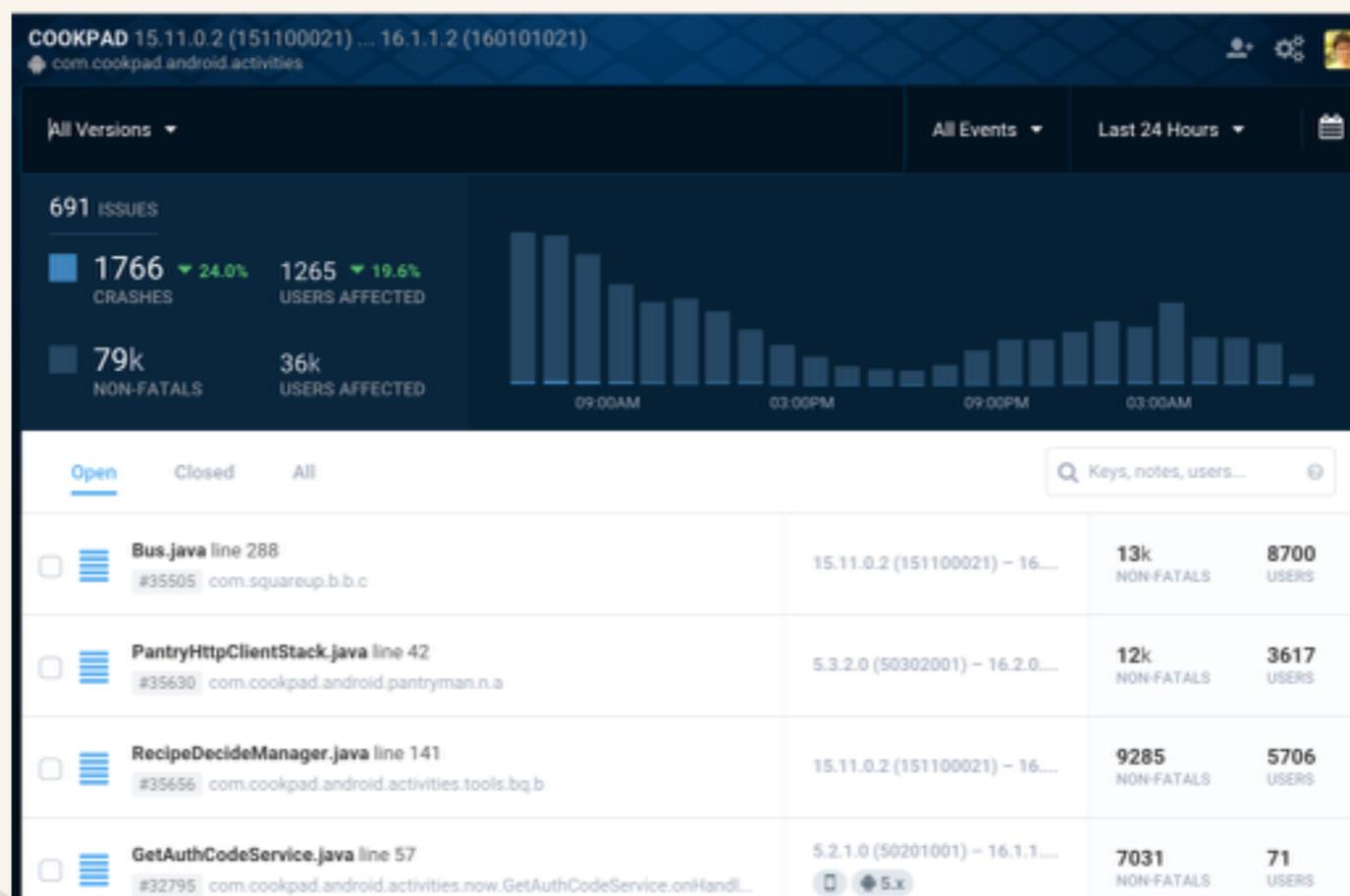
毎日の料理を楽しみに

cookpad

予期せぬクラッシュを素早く検出

Google AnalyticsのリアルタイムサマリーとCrashlyticsの状況を見る。

新バージョンの比率に対してクラッシュ数が多いか？



毎日の料理を楽しみに

cookpad

予期せぬエラーを素早く検出

APIのエラーログをKibana4に流しグラフ化。前バージョンとくらべてエラーが跳ねていないかを監視する



<https://www.elastic.co/downloads/kibana>

ユーザーの声を聞く

ご意見フォームに送られた特定バージョンの声をチェック

対象期間

~

絞込

Excel出力

1 2 3 4 5 ... ≥ >>

NEW 3195048

[Redacted text]

Android

[Redacted text] /160101021; Android/19; [Redacted text]

CDID

[Redacted text]

referrer

カテゴリ



障害タグ



リリース前

リリース後

- リリーススケジュールによる差分の入るタイミングの統制
- レビューによる検出
- マイルストーンによる全差分の可視化
- マイルストーン差分を中心に網羅的なテスト

- **Crashlytics**のチェック
- **Kibana4**でのAPIエラーチェック
- **ご意見**のチェック



リリース前後でやる事の沢山あるよね問題

- **マイルストーンに入れる
issueの準備**
- **バージョンコードの更新**
- **issueの進捗確認**
- **コードフリーズとRC-
branch作成**
- **差分のリストアップとテ
スト**
- **ライセンスのチェック**
- **更新文言**
- **リリース後監視**
- **KPTのスケジュール確保**
- **etc…**



リリース前後でやる事のチェックリスト をissue化する

Development

- 開発開始のアナウンスブログを書く
- 次のリリースに含めたい主要な機能や修正を決める
 -
- 次のリリースのページを GROUPAD に作る
 - 
 - [モバ] クックパッドアプリ[Android] v3.6.0 みたいなタイトルで。
 - < release_infoのURLを書く >
- 主要な機能を実装する
- バージョンアップPRを作る
- リリースカレンダーに記載する
 - 10%, 50%, 100%すべて書く
 - 月ごとにタブが異なるので注意
- KPTのIssueを作る
- Google Playに掲載する更新情報のissueを作る

リリース前後でやる事いつだれがやるの問題

あ、誰かやっているとってた

Development

- 開発開始のアナウンスブログを書く
- 次のリリースに含めたい主要な機能や修正を決める
 -
- 次のリリースのページを GROUPAD に作る
 - [\[モバ\] クックパッドアプリ \[Android\] v3.6.0](#) みたいなタイトルで。
 - < release_infoのURLを書く >
- 主要な機能を実装する
- バージョンアップPRを作る
- リリースカレンダーに記載する
 - 10%, 50%, 100%すべて書く
 - 月ごとにタブが異なるので注意
- KPTのIssueを作る
- Google Playに掲載する更新情報のissueを作る

あ...

リリースマネージャ

Development

- 開発開始のアナウンスブログを書く
- 次のリリースに含めたい主要な機能や修正を決める
 -
- 次のリリースのページを GROUPAD に作る
 - [https://groupad.cookpad.com/entry/2015/02/25/093000](#)
 - [モバ] クックパッドアプリ[Android] v3.6.0 みたいなタイトルで。
 - <release_infoのURLを書く>
- 主要な機能を実装する
- バージョンアップPRを作る
- リリースカレンダーに記載する
 - 10%, 50%, 100%すべて書く
 - 月ごとにタブが異なるので注意
- KPTのIssueを作る
- Google Playに掲載する更新情報のissueを作る

QA

- このバージョンのコミッターリストのissueを作りマージ後確認を依頼する
 - `git log --merges --pretty=format:"%C(red)%an,%C(green)%h,%Creset%s" released_branch...release_branch | sort -f`
- 更新内容の文言とスクリーンショットを用意する
 - コメントに更新内容の文言を書く
- アプリに出す更新内容を修正する
- 次のrelease issueを作る

10% Release

- Crashlyticsに怪しいクラッシュがないか確認する
- リリース対象のマイルストーンから作業手順issue以外のすべてのissueがcloseされていることを確認
- リリースビルドを用意する
 - <ビルドしたJenkinsのJobのURLを書く>
- 10%リリースする
- RCブランチをmasterにマージする
- GAにリリース日のメモをつける
- リリース後監視 1日目 3時間後
- リリース後監視 1日目 夕方
- リリース後監視 2日目 朝
- リリース後監視 2日目 夕方
- リリース後監視 3日目 朝
- リリース後監視 3日目 夕方

**必要な項目を
チェックリスト化し、
1リリースのあいだに
それを全て担当する
役割を作り、持ちまわる**

品質を上げる仕組み

運用コストを下げる仕組み

- リリース前後でエラー検出を素早く行えるようになった
- リリース前後にやるべき事をチェックリスト化する事で漏れぬけが無くなった
- それらをリリースマネージャが受け持つ事で開発者は開発に集中できるようになった

2014年末

- **リリースサイクルの安定(13回リリース)**
- **3500 crash / day (75%減)**



2015年1月 モバイルファースト室解散 モバイル基盤チーム設立



Androidエンジニアとしての役割++

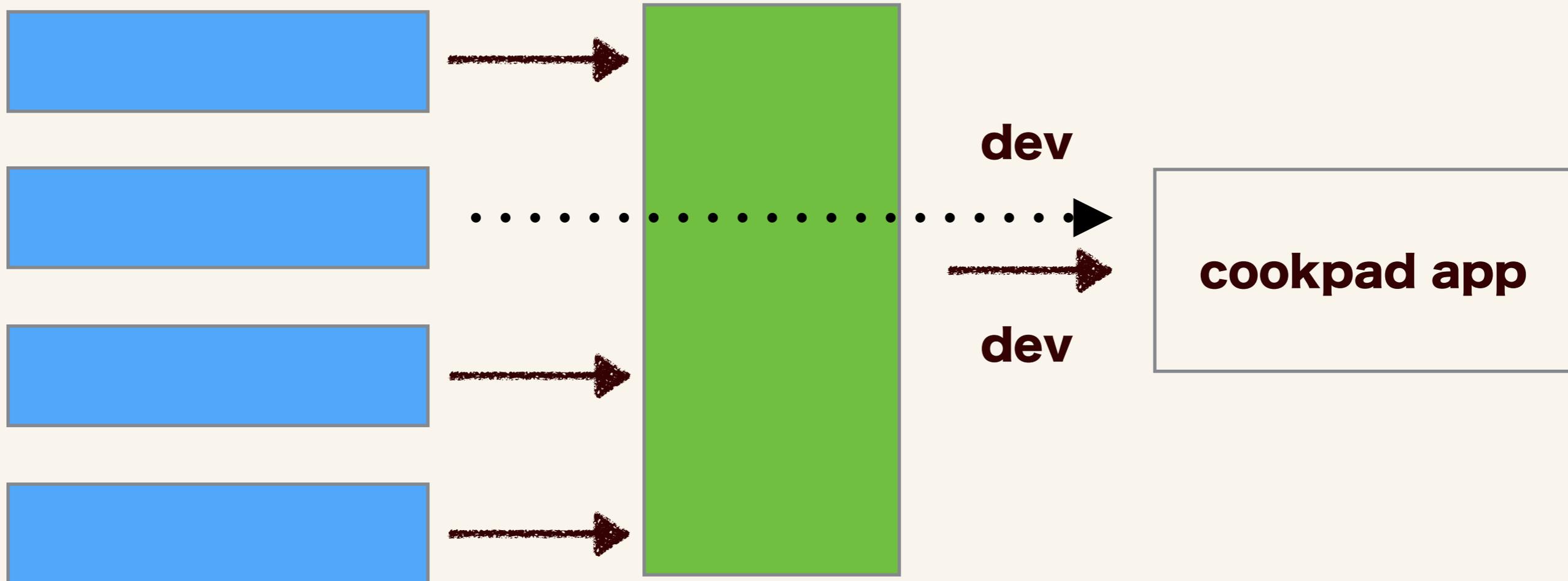
- **事業部内でサービスの成長に責任を持つ(主にAndroid部分)**
- **事業を加速させる開発基盤を強化していく**



对多部署体制時代

division

mobile first team



request

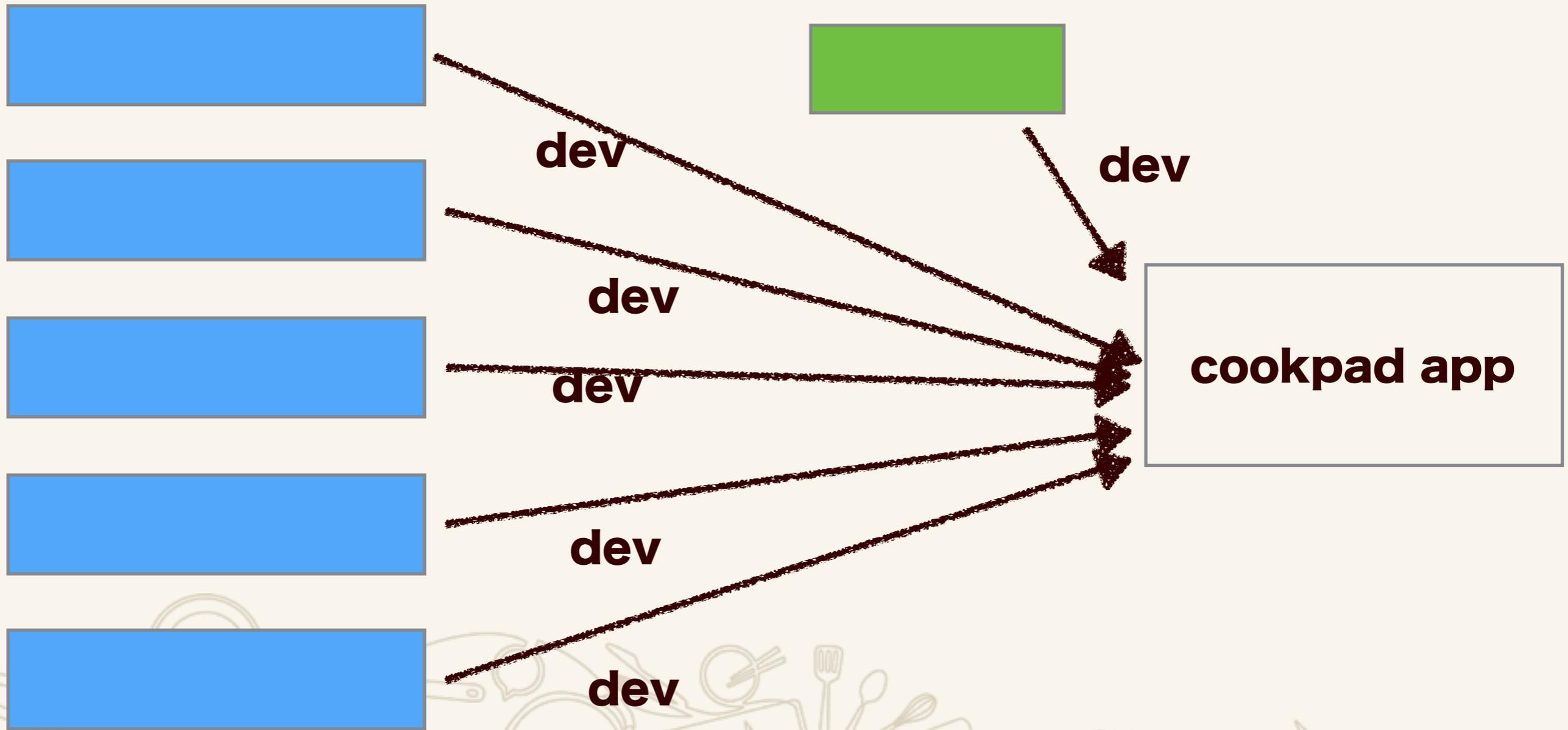
knowlage



多事業部多人数体制時代

division

mobile foundation team



検索結果画面

☰ しょうが焼き 🔍 詳細検索

新着順 | **🏆 人気順**

しょうが焼きのレシピ **8436**



🏆 殿堂入りレシピ
1000人が絶賛のしょうが焼きレシピ



今日の特売品 × しょうが焼きのレシピ >
近くのお店の特売情報



やわらか〜い豚の生姜焼き 📁+

豚ロース薄切り肉（しょうが焼用）、塩、こしょう、片栗粉、玉ねぎ（薄切り）、★しょう
by ラビ♡



我が家の野菜たっぷり♪簡単な生姜焼き☆ 📁+

豚こま肉or豚バラ肉、玉ねぎ、人参、ピーマン、☆醤油(薄口) ☆お酒 ☆みりん ☆砂糖 ☆

AD

毎日の料理を楽しみに

 cookpad

検索結果画面

☰ しょうが焼き 🔍 詳細検索

新着順 | **P** 人気順

しょうが焼きのレシピ 8436



P 殿堂入りレシピ
1000人が絶賛のしょうが焼きレシピ



今日の特売品 × しょうが焼きのレシピ >
近くのお店の特売情報



やわらか〜い豚の生姜焼き 📁
豚ロース薄切り肉 (しょうが焼用), 塩, こしょう, 片栗粉, 玉ねぎ (薄切り), ★しょう
by ラビ♡



我が家の野菜たっぷり♪簡単な生姜焼き☆ 📁
豚こま肉or豚バラ肉, 玉ねぎ, 人参, ピーマ
...

AD

会員事業部

買物情報事業部

検索編成部

投稿推進部

広告事業部



結構でかい

- コード:170,000行くらい
- クラス数: 1058くらい
- メソッド数: 93,000くらい
- (ライブラリ込み,proguard済)
- 画面数: 70くらい
- コード変更者: 1リリースで7-8人

Androidエンジニアとしての役割++

- **事業部内でサービスの成長に責任を持つ(主にAndroid部分)**
- **事業を加速させる開発基盤を強化していく**
- **多事業部多人数でもスピード、品質を落とさない体制作り**



How?



KPT, KPT, KPT



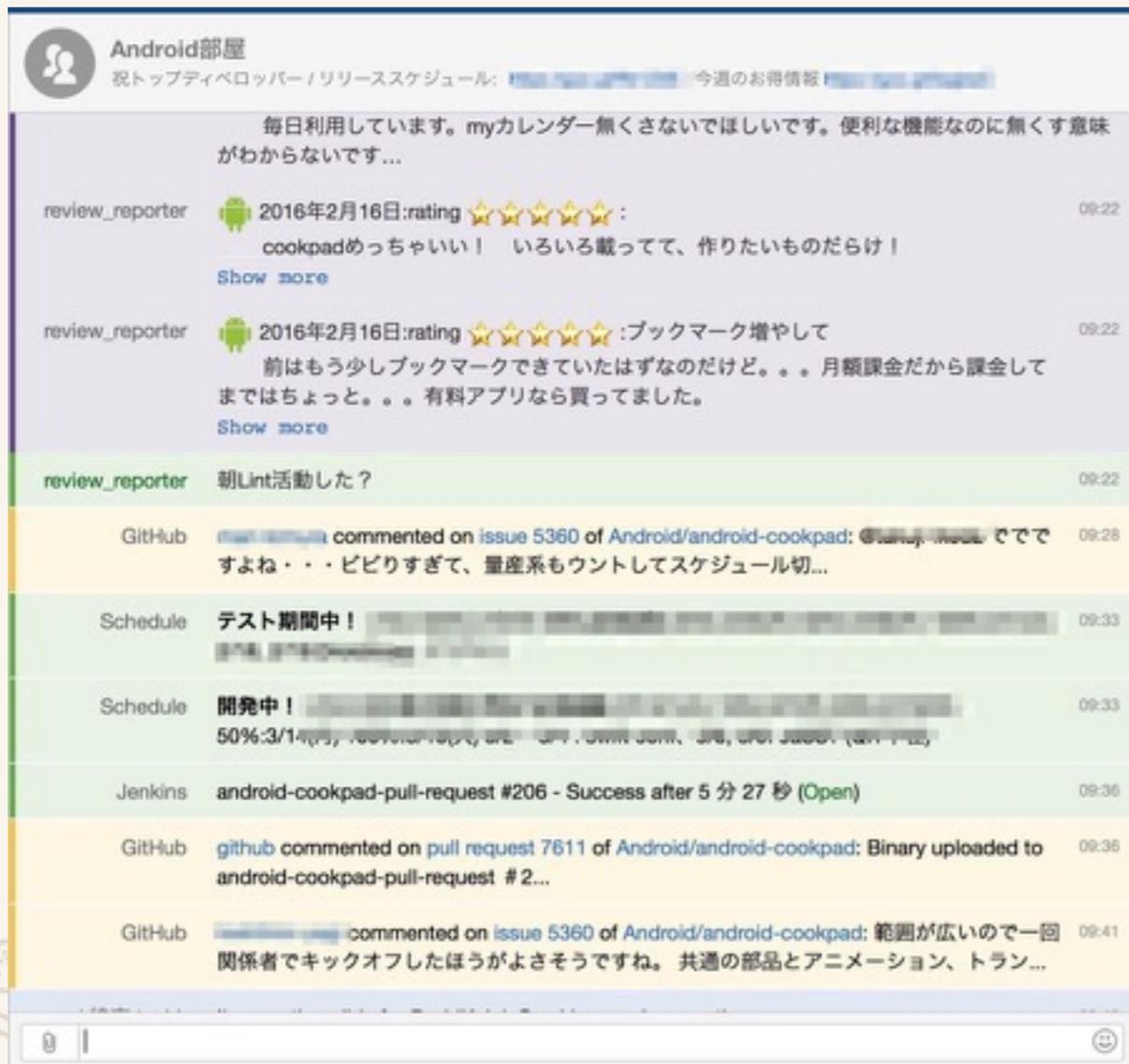
可視化とコミュニケーション

Android朝会

- **そのバージョンに関わる開発者がモバイル基盤を中心に部署横断的に集まり話す。10分程度。**
- **スケジュール、相談、周知などなど**



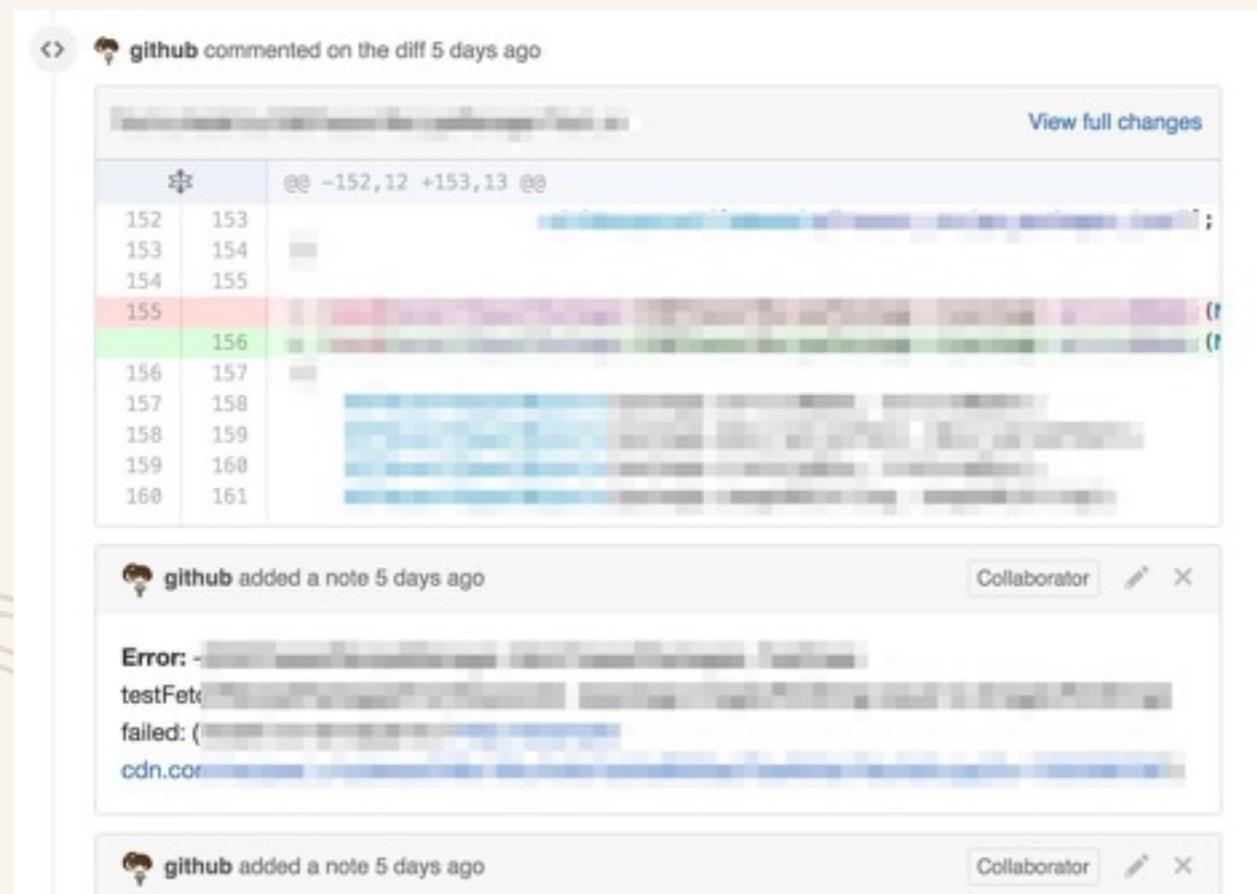
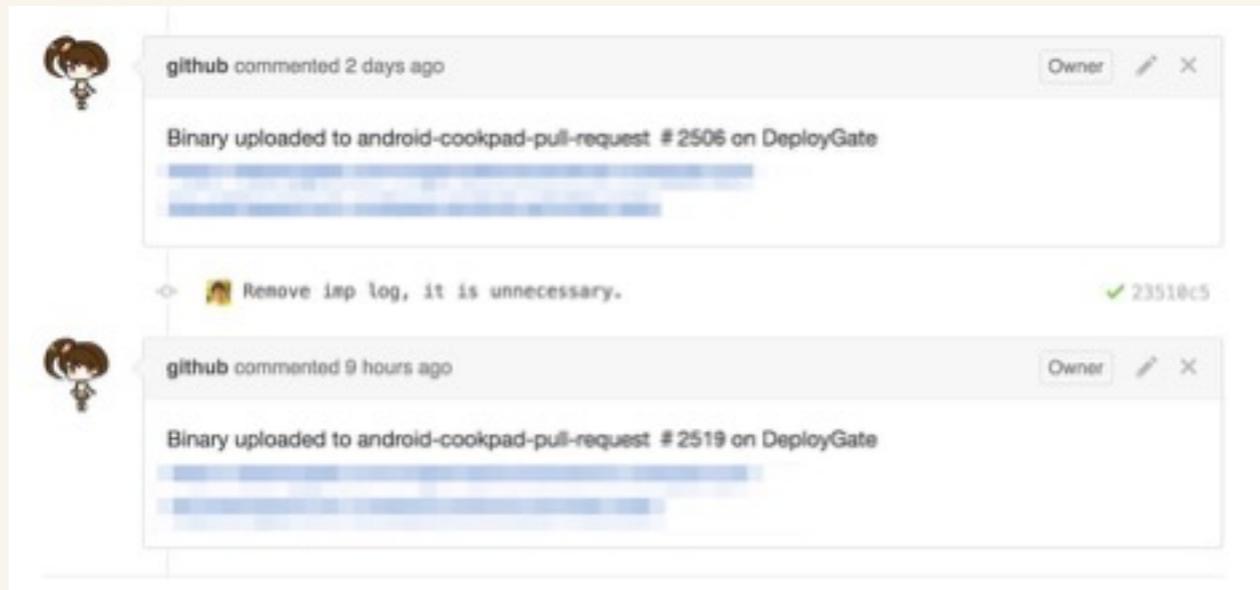
チャットでの通知を強化



- **Github上のコメント、PR、merge等のイベント通知**
- **開発スケジュールを毎日通知**
- **CI Statusを通知**



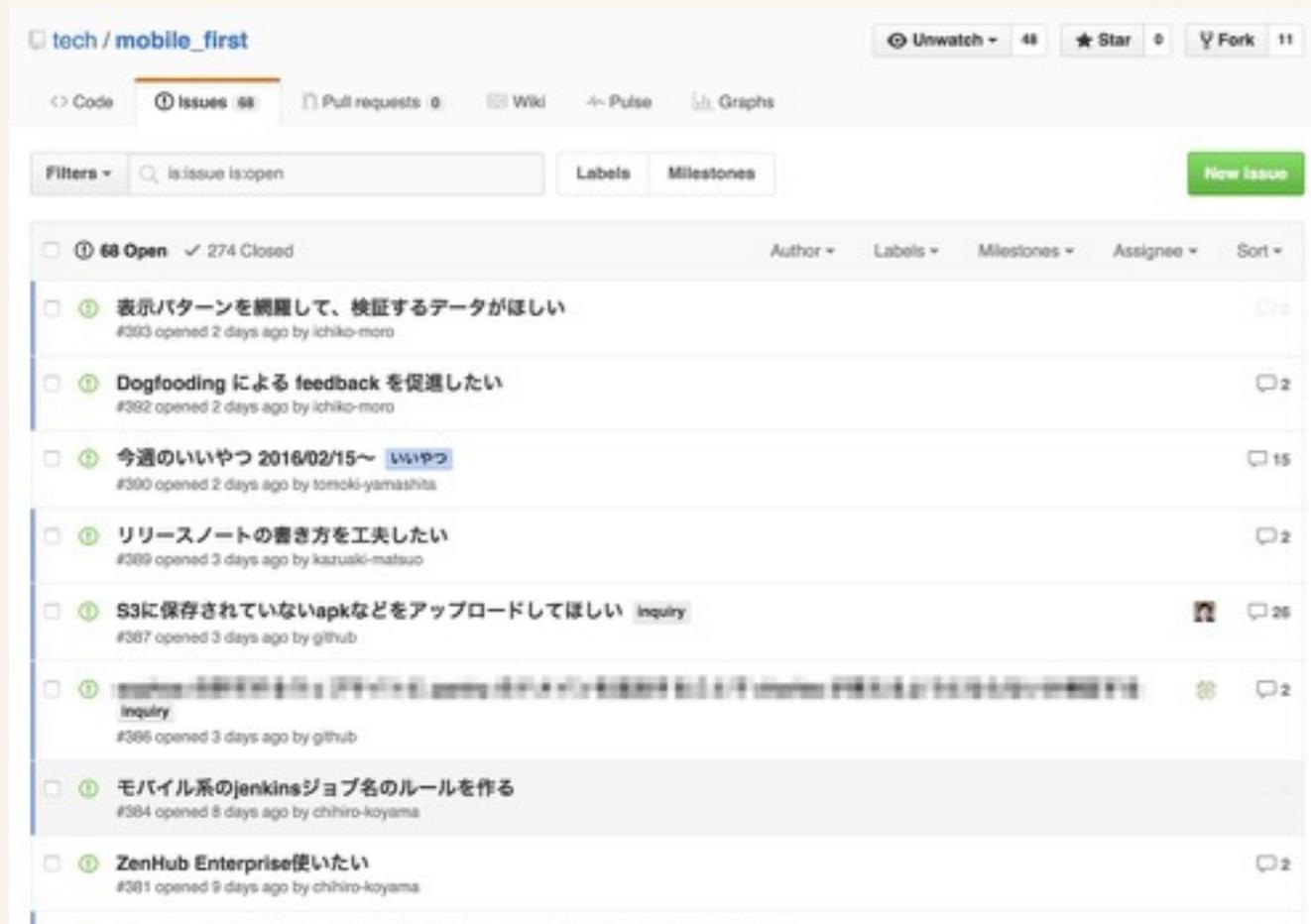
Pull Request効率化



- Pull Request出すとCIでビルドしてDeploygateに配信
- デザイン確認等捗る
- Dokumi (<https://github.com/cookpad/dokumi>) を使って自動的に細かいbug等を指摘
- findbugs, android lint



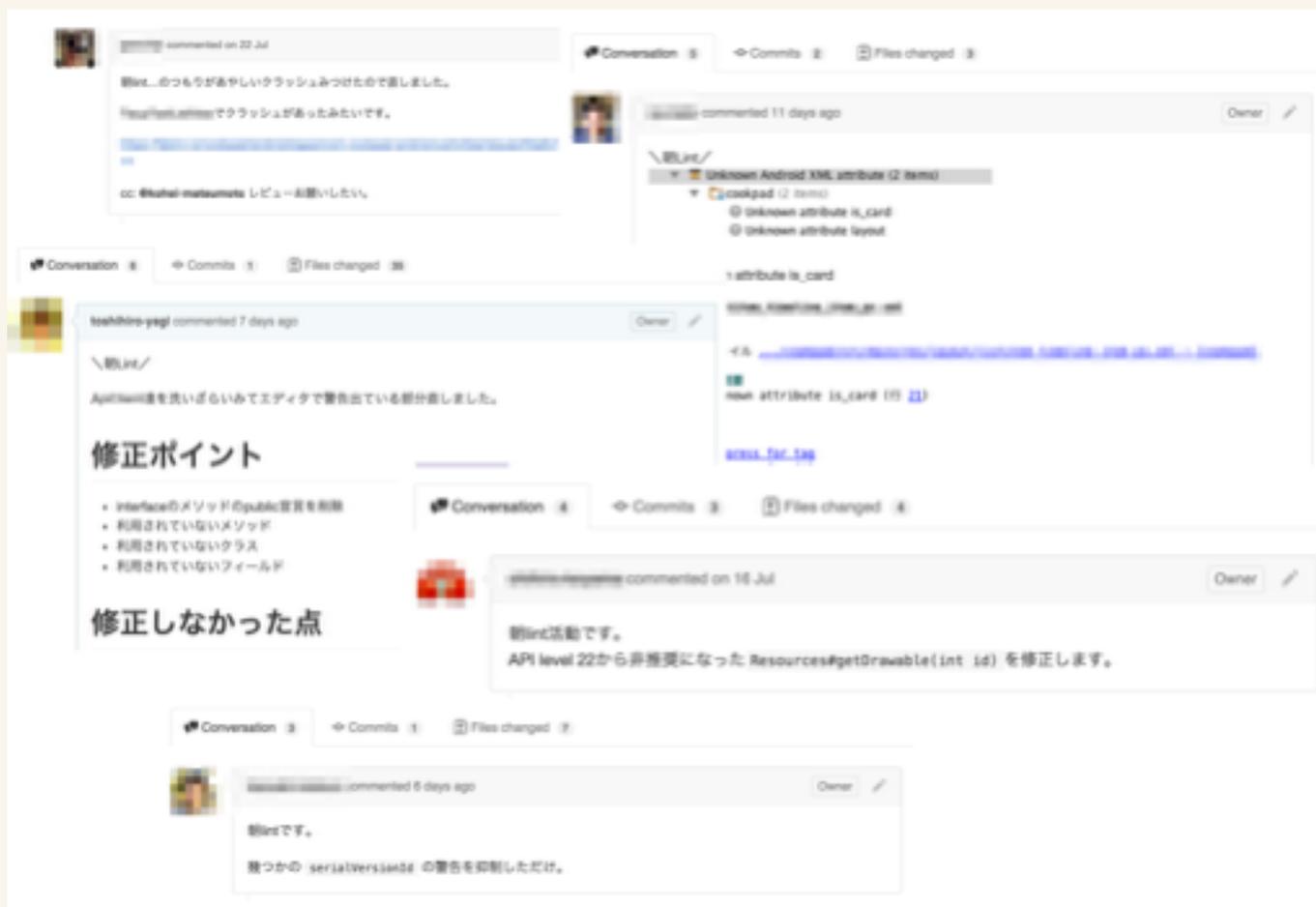
zatsu リポジトリ



- **モバイル開発の全般的な課題や要望、やりたいこと、夢などを雑に書く専用リポジトリ**
- **モバイル基盤チームが引き取る場合もあればissueを書いた人がゴリッとやる場合もある**
- **社内勉強会の検討issueや飲みに行く予定を詰めるissueなども作られる**

雑な発想を活かすチーム作り:<http://techlife.cookpad.com/entry/2015/03/25/202709>

朝Lintによる変更ハードル低減



- typoやメソッドの構造の改善やコンパイラが吐く警告の修正など優先度が低いものでもどんどん修正する習慣
- 小さなものをつぶしていった大きなものにぶつかった場合はissue化
- 部署横断的な活動
- 今では朝Lintと銘打たなくてもやるようになった

朝Lint活動で細かな技術的負債を返済する : <http://techlife.cookpad.com/entry/2015/09/15/190000>

ライブラリ導入による効率化 アーキテクチャの改善

- **Retrolambda, RxJava, RxT4A, Data Binding Library, Easy Parcelable, kvs-schema, fragment-creator, Android Orma, Permissions Dispatcher, Robolectric**
- **導入時にはpros/consを議論するissueを立てたり勉強会を開いたりする**
- **筋肉、QAフェーズに徹底的にテストをする**

同じ船に乗る

- 関係者の日々の継続的な会話で意識がズレない
- チャットでのイベントの可視化による議論の活性化
- やりたい事や要望を話せる雰囲気
- 本当にやるべきかを真剣に議論しあえる雰囲気
- 部署横断的に改善を続ける雰囲気

毎日の料理を楽しむに

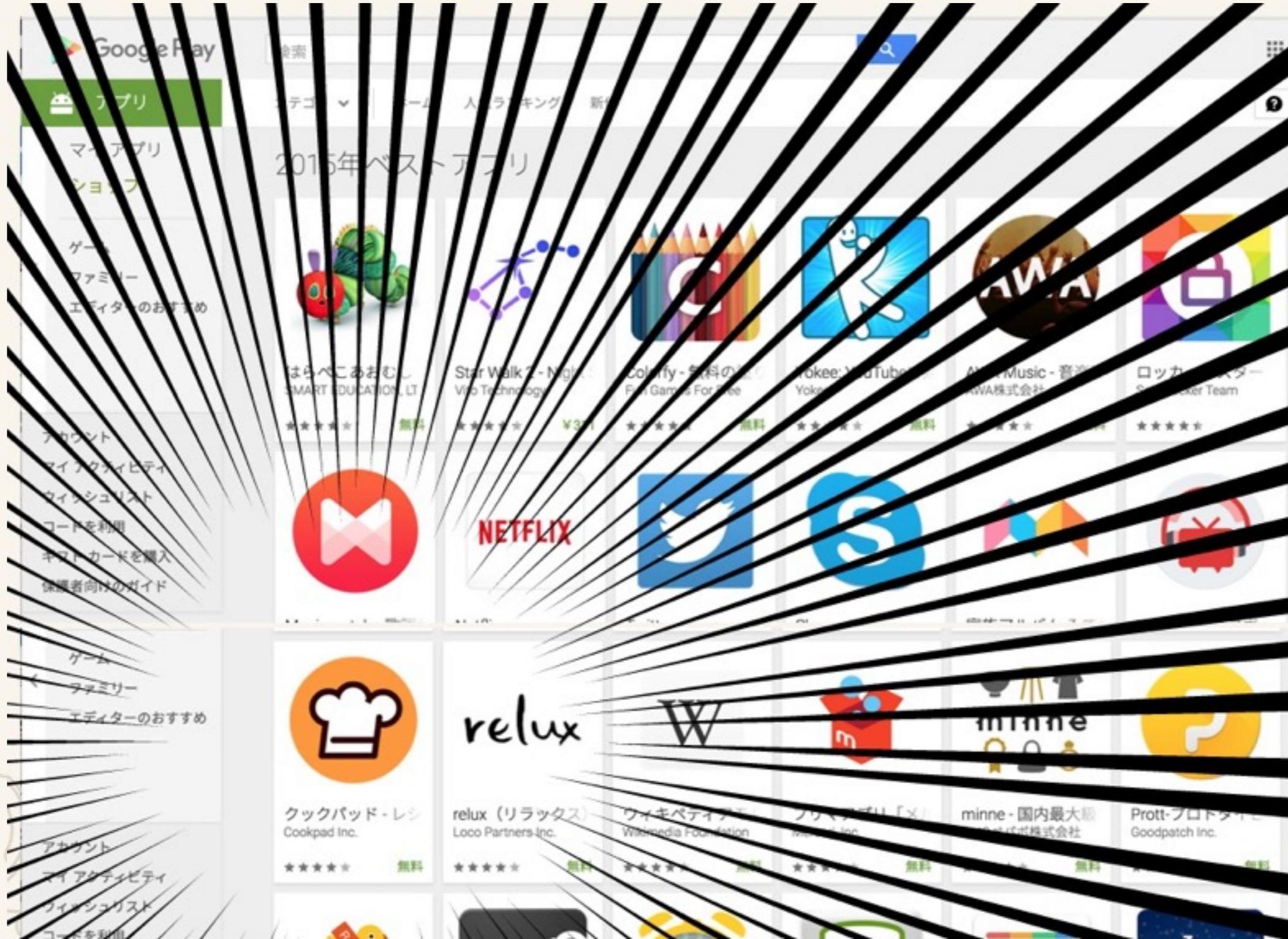
cookpad

2015年末

- **リリースサイクルの維持(17回のリリース)**
- **スピードの維持(bug fix, 設計改善, 新機能etc…)**
- **1800 crash / day (更に品質アップ)**



2015年ベストアプリ



トップデベロッパー



クックパッド - No.1レ ◆ トップデベロッパー
シピ検索 & スーパーのチラシア
プリ

Cookpad Inc. ライフスタイル

★★★★★ 49,290 

3+

アプリ内購入あり

 このアプリはお使いのすべての端末に対応しています。

インストール済み



2016~



まだまだやることが沢山





User First

すべてはユーザのために

毎日の料理を楽しみに

 cookpad



cookpad

**毎日の料理を楽しみにすること
とで、心からの笑顔を増やす**



We're hiring.

Let's enjoy Android together.



質疑応答

