

Internet Week 2017

S15 高信頼性運用を実現するSREという新潮流

モンスターストライクの信頼性を支えるSREの組織化について

株式会社ミクシィ XFLAG スタジオ

ゲーム開発室 SREグループ

清水 勲



XFLAG

ケタハズレな冒険を。

自己紹介

清水 勲 / Isao SHIMIZU @isaoshimizu

株式会社ミクシィ XFLAG 事業本部 ゲーム開発室 SREグループ 所属

経歴

- SIerで受託開発、自社プロダクト開発、運用を約8年
- 株式会社ミクシィ
 - 2011.8～ 運用部 アプリ運用グループ所属、SNSの運用
 - 2014.4～ モンスターストライクの運用にジョイン
 - 2015.8～ XFLAG スタジオが創設される
 - 2016.7～ XFLAG スタジオにSRE グループ創設

ミクシィグループ

株式会社ミクシィ

SNS「mixi」や「モンスターストライク」を提供



株式会社ミクシィ・リクルートメント

WEB業界で働く人に特化した、企業と求職者をマッチングするプラットフォーム「Find Job!」を提供



株式会社ノハナ

写真を軸とした家族コミュニケーションを活性化するフォトブックサービス「ノハナ」を提供

nohana

株式会社Diverse

結婚支援を中心とした出会いのプラットフォームを提供



株式会社フンザ

チケットフリマサービス「チケットキャンプ」を提供



株式会社クト

価値観診断をベースとした相性マッチングアプリ「Pancy」を提供

Pancy

XFLAG スタジオ

- **スマートフォン向けゲーム**

- モンスターストライク (2013.10～)
- モンストスタジアム (2015.4～)
- ファイトリーグ (2017.6～)

- **動画**

- モンストアニメYouTube配信 (2017.6.14に世界累計再生回数2億回突破)
- 昨年末には劇場版も公開

- **XFLAG STORE SHIBUYA (常設店舗)**

- **XFLAG STORE (オンラインストア)**

- **その他**



XFLAG
ケタハズレな冒険を。



モンスト
MONSTER STRIKE **アニメ**

SRE という組織ができるまでの 変遷についてお話します

モンスターストライク以前

モンスターストライク以前

- かつては運用部という組織で、SNS「mixi」の運用に取り組んでいた
- インフラ、アプリ運用という2つのグループ
 - インフラ
 - サーバー調達、ネットワーク設計・構築などがメイン
 - インフラエンジニアと呼ばれたり
 - アプリ運用
 - サーバー構築、負荷対策、デプロイ、チューニングなどがメイン
 - 運用エンジニアと呼ばれたり
- 運用部と連携する「たんぽぽグループ」

たんぽぽグループ

- 2008年頃、「刺身の上にたんぽぽをのせる仕事」のような単純作業の仕事から社内開発者を解放しよう、というミッションのもと設立
- 「開発者のための開発」をおこなう組織
- サービスがどうあるべきかという大局的な視点に立って、すべてのシステムに横断的に関わる組織
- コアアーキテクチャの検討、開発工程の改善、改善のためのツールの導入検討、パフォーマンスチューニング、アルゴリズム改善、海外向けサービスプロジェクトのサポートなど
- 開発・運用がスムーズに進むように
- 現在においてもXFLAG スタジオ内で同様の取り組みをおこなっている

当時のシステム・インフラ

- ~2013年くらいの話
- SNSのシステム
 - サーバー
 - 1つのDCにオンプレミス（数千台） ※いまはAWSに移行済み
 - OS
 - Fedora 8から17へ ※いまはCentOS 7.1
 - プログラム言語
 - Perl 5.8系から5.14系へ
 - ミドルウェア
 - Apache 2.2系(mod_perl, mod_proxy)、Percona Server 5.1、Memcached 1.4.5
 - ソースコード管理
 - SubversionからGitへ(Gitolite → GitHub Enterprise)
- コミュニケーションツールはIRCがメインだった

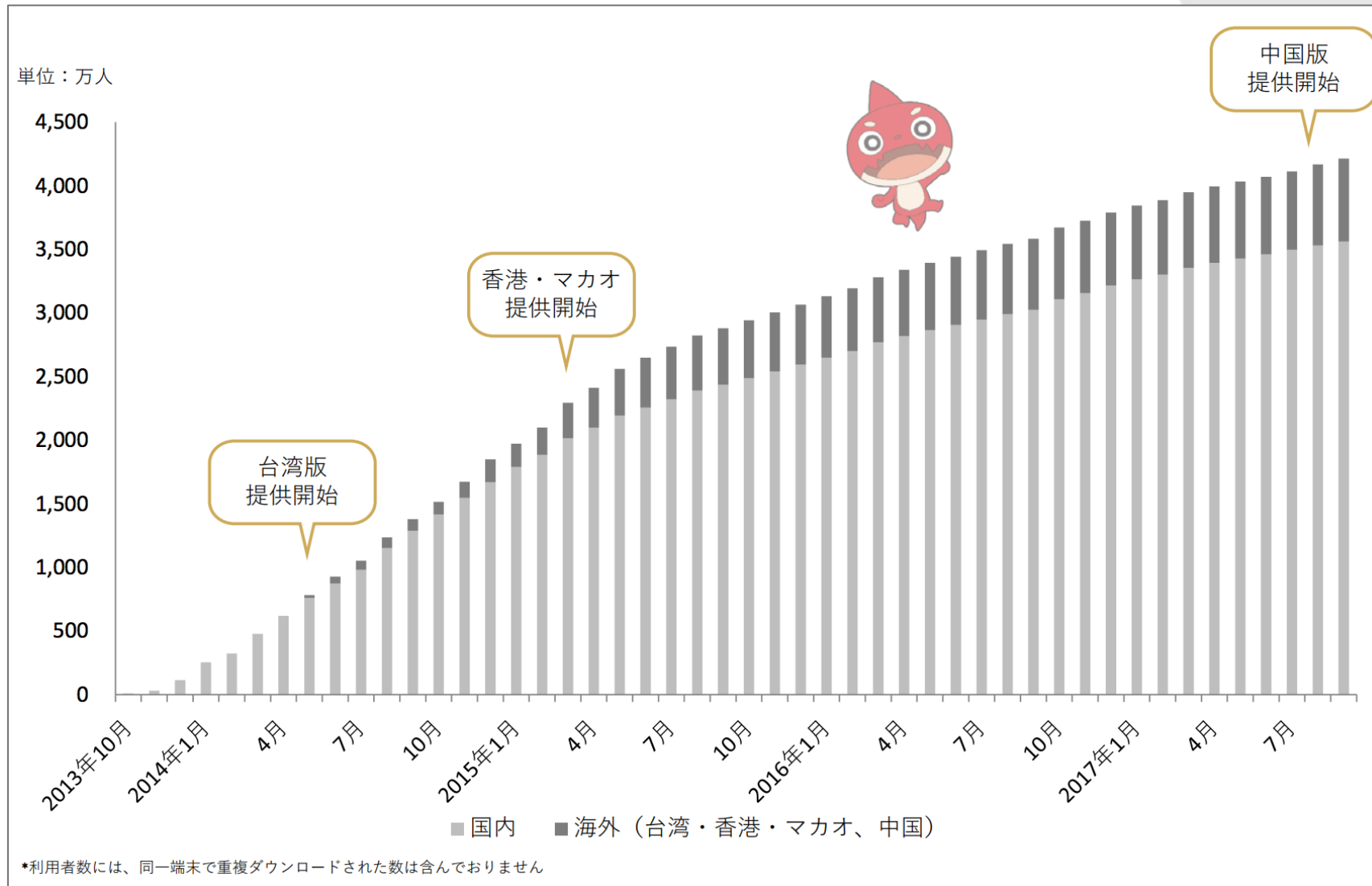
当時の課題

- **主に負荷対策、効率化、コスト削減**
- **課題、取り組んできたことの一例**
 - MySQLの負荷対策、ioDriveでの集約化
 - 古いOS、古いミドルウェア、Perlのアップデート
 - OpenStack導入
 - systemd対応
 - デプロイ、プロビジョニングの改善
 - コンテナ化
 - その他いろいろ
- **JIRAで課題管理、アサイン、作業の実施**
- **Confluenceでドキュメント作成**

モンスターストライクの登場

2013年10月

「モンスターストライク」利用者数推移



2017年11月8日 2018年3月期 第2四半期 決算説明会資料より抜粋

モンスターストライクリリース後

- **2014年前半からは、新機能の開発と平行して負荷対策に注力していた**
- **スケールアップ**
 - 高負荷のマシンはAWSからオンプレへ（自社インフラの活用）
 - AWSも併用する。Direct Connect（専用線）フル活用。
 - SSDやioDrive(PCIe SSD)の活用
- **スケールアウト**
 - DB分割（テーブル分割、シャーディング）
- **DBチューニング**
- **ソースコードの改善**
 - クエリ改善
 - キャッシュの活用（Memcached）
- **コミュニケーションはIRC → HipChat → Slackへ**
- **ソースコード管理はGitHubに統一**

SRE グループ^o設立

2016年7月

SRE について

- **Site Reliability Engineering**

- Googleが提唱し、Facebook、Dropbox、メルカリ、クックパッド、サイボウズなど、最近では多くの企業を取り入れてきた（組織として存在する）
- システム運用、可用性向上、人手で行ってきたことの効率化・自動化など
- 運用業務よりもソフトウェアエンジニアリングに割く時間の割合が多め
- 書籍「SRE サイトリライアビリティエンジニアリング——Googleの信頼性を支えるエンジニアリングチーム」
 - <https://www.oreilly.co.jp/books/9784873117911/>
 - Googleのサイトでは英語版が無料で読める <http://landing.google.com/sre/book.html>
- 最近では、SREに関するイベントや勉強会も増えてきている

XFLAG スタジオにおけるSRE グループについて

- **求められること**
 - サービスに何が起きていて、何をすべきか理解すること
 - 当たり前前のことを優先度付けして能動的にやれること
 - 視野を広くして俯瞰して見られること
 - ソフトウェア・エンジニアリングによって徹底的に信頼性を向上させること
- **変わったこと**
 - 社内&社外からもわかりやすい組織体制
 - ゲームに関わる機能開発からの分離
 - 負荷対策、効率化、自動化などに注力
- **従来と変わらないこと**
 - メンバーの得意不得意を相互に補完
 - 運用業務

XFLAG スタジオにおけるSRE の業務内容

- **モンスターストライクの負荷対策**
 - クエリ改善、キャッシュ利用の効率化、DB分割、チューニングなど
 - リソース見積もり、ベンチマーキング
- **可用性向上（壊れにくいハードウェア選定、ミドルウェア構成）**
- **データのバックアップ**
- **リリースエンジニアリング（デプロイ、プロビジョニング）**
- **物理マシン、クラウドのリソース設計と最適化**
- **自動化、ツール開発**
- **監視、モニタリング改善**
- **各種Webサイト構築**
- **新規案件相談、モック開発**
- **セキュリティ対策**
- **障害対応（オンコール対応）**
- **その他**

オンコール対応

- 定時外、休日の緊急時に一次対応するための制度（当番制）
- 2007年頃から制度化
- 2名体制、1週間でローテーション
- 対応例
 - ハードウェア故障対応（メモリ、電源、SAS、SSDなど）
 - 負荷増への対応
 - クラウド障害対応
- いまではPagerDutyフル活用
 - 以前はNagiosからのメール受信のみだった
 - 様々な通知（電話、メール、プッシュなど）
 - 当番が通知に気づかなかった場合、当番外へ自動エスカレーション

仕事の進め方

- **それぞれが能動的に行動し、今やるべき仕事は自分で見つける**
 - マネージャーからはチームの方向性の調整のみ
- **使いたい技術はメンバーの合意を得て積極的に導入する**
 - 例えば、プログラム言語、クラウド、ミドルウェア、ツールなど
- **Slackのチャンネルで議論**
 - ダイレクトメッセージではなくチームのチャンネルでおこなう
- **GitHub上でのコードレビュー必須**
 - Pull Request、Issue上で大半の課題は解決する
- **テストとレビューが通り、masterにマージされたらデプロイする**
 - SREに限らず、XFLAG スタジオで統一されたやり方

SRE の評価ポイント

- 何にどのくらい時間をかけて、何に対して貢献したのか
- 与えられた仕事だけしても評価はされない
- **技術**
 - 技術によってどんな課題を解決できたのか
 - 今その技術を選ぶ必要があったのかどうか
- **アウトプット**
 - 何を作ったのか、そのモノの価値はどうか
 - 生産性は高かったのか
- **事業貢献**
 - 事業、プロダクト・サービスへの程度貢献できたのか
 - なぜそれに貢献したのか
- **グループ貢献**
 - グループに与えた影響はどんなものだったのか
 - メンバーに対してどんな行動をして、何が変わったのか

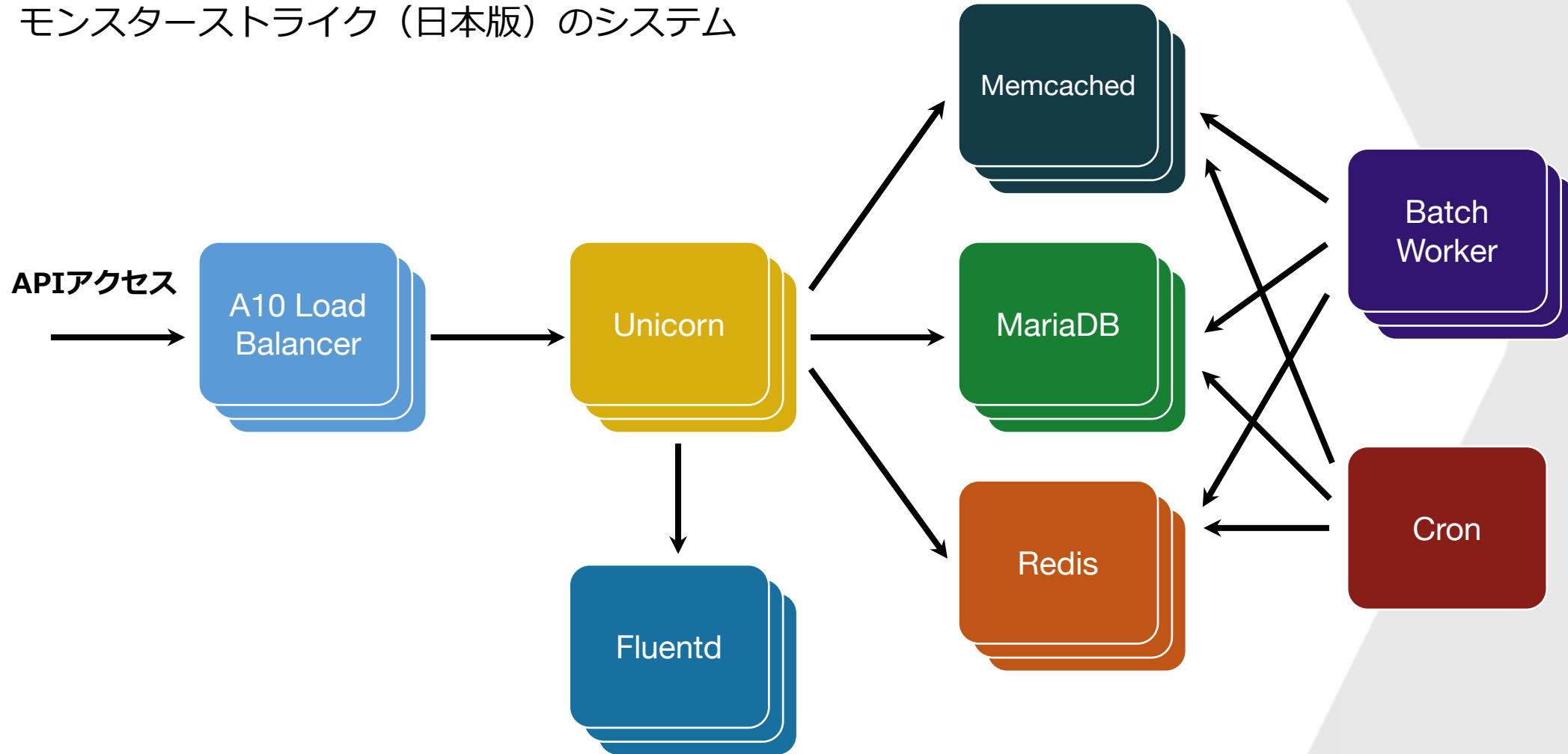
現在のシステム・インフラ

現在のシステム・インフラ概要

- **モンスターストライク（日本版）のシステムの例**
 - サーバー（現在1,100台くらい）
 - オンプレミス（2つのDC）
 - DC単位での冗長構成（片方のDCが死んでもサービス継続できるように）
 - マルチクラウド（AWS, GMO, GCP）
 - レイテンシ（RTT） 1ms以下を目指したい
 - 適材適所、その時に最適なものを使う
 - OS
 - Ubuntu Server
 - プログラム言語
 - Ruby
 - ミドルウェア
 - unicorn、nginx、MariaDB、Memcached、Redis
 - ソースコード管理
 - GitHub

現在のシステム・インフラ概略図

モンスターストライク（日本版）のシステム



現在の課題

- **負荷対策はずっと続いていく**
 - さらなる高負荷が想定される企画、事業にに応じていく
- **古いものを捨てて新しいものを使う**
 - ハードウェア、ソフトウェア、アーキテクチャ
 - 入れ替えしやすい環境作り
- **人間の手による作業を減らす**
 - ミスを減らす、作業時間を減らす
 - 例えばクラウドのコンソール画面をポチポチする作業
 - APIを使ったツールの開発
- **ハードウェアのパワーに頼りすぎない**
 - ソフトウェアで解決できることを探す
 - 耐障害性向上、コスト削減につながる

まとめ

まとめ

XFLAG スタジオにおけるSRE(Site Reliability Engineer)

- いまままでの運用業務もやりながら、ソフトウェアを作る、使うことで課題を解決していく
- 能動的に、広い視点で、最適な技術を使って、いまやるべきことにおいて価値を生み出す
- 新しいものを取り入れ、新しいことに挑戦し、事業に貢献していく

