

Disrupting Developer Productivity One Bot at a Time



Margaret-Anne
Storey University of
Victoria



Alexey Zagalsky
University of
Victoria

(11月のFSE '16で発表予定)

botとは

- ✧ 一般的な定義

- ユーザとサービスをつなぐインターフェース

- ✧ 本研究でのbotの定義

- パーソナライズ可能な会話型ユーザーインターフェース (UI)

- Hubotとは

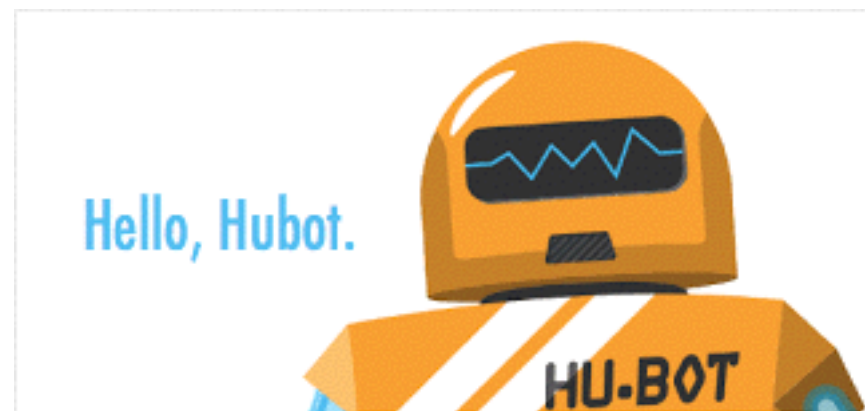
- ★ GitHubが提供する
botのフレームワーク
- ★ Hubot is your company's robot.
Install him in your company to
dramatically improve and reduce
employee efficiency.



<http://blog.mmmcorp.co.jp/blog/2015/02/11/hubot-hipchat/>

背景

- * Botは、ソフトウェア開発におけるつまらない仕事を支援（自動化）するために使用される
- * Botがより、ソフトウェア開発に普及すれば、ソフトウェアの品質や開発者の生産性がさらに向上していくと考えられる
- * Hubotのようなフレームワークが登場したことにより、開発者個人が利用するためのBotを開発できる



問題

- ✧ Botをどこに活用すれば，開発者個人やチームの生産性にどのような利益を得られるのかは，あまり知られていない



使えるところが
分かっているのは氷山の一角

<https://www.youtube.com/watch?v=RpbW-qBG9QM>

- ✧ Hubotはフレームワークであり， Bot自体は自分で設計しなければならない
 - どのように設計すればいいか不明確なままである

目的

❖ 目的

- Botがソフトウェア開発において支援できるドメインを示したフレームワークを提案する

❖ アプローチ

- 既存のBotに関する知見を参考に，ソフトウェア開発に必要なBotの役割 **(Developer Bot Roles)** を考えていく
- 既存のbotに関する知見
 - ★ Slack, HipChat(アトラシアン)に含まれているbotツールの仕様
 - ★ Sven Pealによる，アトラシアンでのbotの活用法

<https://www.youtube.com/watch?v=RpbW-qBG9QM>

Developer Bot Roles

- ❖ Code Bots
- ❖ Test Bots
- ❖ DevOps Bots
- ❖ Support Bots
- ❖ Documenting Bots

Code Bots

- ✧ コーディングを効率化することができるBot
 - どのようにして? : Botが1つのタスクをこなすことで, 開発者が2つ以上のタスクを同時に行えるようにする

既存のCode Bot	概要
BugBot	Githubでbug reportが作成された場合, Slackを通じて通知してくれる
Peer Review Bot	Pull Requestを自動でチェックし, 似ているようなものをマージしてくれる
Karan	Slackで, 開発メンバーに質問すると, その質問をStack Overflowでもしてくれる
Atlassian Bot	ビルドしている間にブランチを自動で切ったり, 複数のブランチに跨ったコミットをマージしてくれる

- ✧ プログラミングしたら済むような初歩的なものが多い
- ✧ 将来的には, コードの意図を理解した上で, 洗練されたアクションを起こせるようになっていく

Test Bots

✧ テストを効率的に行えるようにしてくれるBot

- どのようにして? : Botが静的解析を行い, テストに要する時間を減らす

既存のTest Bot	概要
FreudBot	FindBugs, CheckStyleなどの静的解析ツールを自動で実行し, 検出された結果に対応するPull Requestを示してくれる
Dr.Code	プロジェクトの健康状態をコードを通じて追跡&可視化することにより, "技術的負債"を検出してくれる
Compare Bot	画像(スクリーンショット)を用いてコードの変更を検出し, バグの可能性のある箇所をユーザに提示してくれる

- これらは, テストすべき課題を検出・隔離し, テストに要する時間を一時的に減らす
- しかし, botにより隔離されたテスト課題は, 後に開発者が検証しなければならない

DevOps Bots

- ✧ DevOps Botはコードによる開発のスピードを上げる
 - どのようにして? : 開発者によるコードのフィードバックをBotに命令して伝えてもらう

既存のDevOps Bot	概要
Pager Duty	ツールが発したアラートを集約し、予め命令しておいたルールに従い、そのアラートに対するアクションを実行してくれる。
Pagerbot	開発者が不在のとき、Botが代わりにアラートに対して返信してくれる
Deploy Bot	Slackから命令するだけで、デプロイ(Webサービスを利用可能状態にする)してくれる デプロイしたいという意志や、デプロイしたことをSlackで全員に知らせてくれる

- 開発にチャットを導入することは一見奇妙に見えるかもしれないが、メリットがわかりやすい伝えてもらう
- デプロイはたくさんの人から許可をもらわないといけないので、botによる効果は分かりやすい

Support Bots

- * Support Botは、ユーザと開発者間に存在するコミュニケーションギャップを埋める
 - どのようにして? : ユーザから寄せられた大量のレポートの中で、多くを占めているものはBotが自動で答える

既存のDevOps Bot

概要

Zen Desk

Intercom

Smooch

カスタマーのフィードバックに対し、用意された回答を自動で提示する

- Support Botは、時間とともに、過剰なコミュニケーションを減らしたり、UXを向上させるかもしれない
- 似たようなBotは、昔からその他の分野でも使用されてきたMOOCs : 生徒の質問に対する回答を先生に代わって行う

Documenting Bots

- ❖ Documenting Botsは退屈なドキュメントの作成を支援する
 - どのようにして?: ドキュメントをBotが生成する

既存のDevOps Bot	概要
Atlassian Bot	リリースノート作成に必要な情報をコミットやIssueのコメントから吸い上げる。また、過去のリリースノートからいい定型文を抽出する
Translation Bot	他言語のドキュメントを自動生成する

- 将来的には、スタックオーバーフローのようなソースからドキュメントを生成できるようになると考えられる

どのようなBot が必要か？

- ✧ 開発者の生産性を向上させるために必要なBotとはどのようなものかを考える
- ✧ 生産性を持つ2つの側面
 - 効率的：Efficiency (do things faster)
 - 効果的：Effectiveness (towards meaningful goals)
 - それぞれのどのエリアに既存のbotが該当するか検証する

Efficiency (do things faster)

✧ タスクを自動化する

- つまらないタスク (e.g., 変更箇所をスクリーンショットからみつける)
- 反復的なタスク (e.g., 複数のブランチに跨った変更のマージ)
- 余分なタスク (e.g., ユーザのクエスチョンに回答する)

✧ 邪魔を排除し、開発者のフロー状態を保つ

- フロー状態：極限的な集中力により、極限的な技術力を生み出す状態、楽しさのあまり時間を忘れるのが特徴的
- (e.g., Botが代わりにアラートに対して返信する)

Effectiveness (towards meaningful goals)

✧ 決定方法の改善

- 決定に関連する情報をまとめる (e.g., ユーザのフィードバック) (e.g., リリースノート作成に必要な情報を集める)

✧ チームの認識（アウェアネス）の共有やコミュニケーションの促進

- Botは、チームが作成したコミットを把握していて、そのコミットがテスト通らなかったことも把握することができる
- コミュニケーションを支援するBotがないため、望まれる

✧ タスクや目標の管理

- 効率のいい開発者は、タスクや目標の設定に拘っている[2]
- 進捗管理ができるbotや、タスクをコーディネートできるbotがのぞまれる

[2] [M. Arciniegas-Mendez, A. Zagalsky, M.-A. Storey, and A. F. Hadwin. Using the model of regulation to understand software development collaboration practices and tool support. In Proceedings of the 20th ACM Conference on Computer Supported Cooperative Work & Social Computing (to appear). ACM, 2017.

まとめ

✧ 明らかになったこと

- 既存のBotの仕様から，ソフトウェア開発におけるBotの役割を明確にした
- 既存のBotは，タスクを効率化させるものが多く，タスクを効果的にするものが少ない

✧ 不明確なままのところ

- Botをどのように設計すればよいか
- Botをどのように学習させればよいか