Better Issue Management With ChatOps

RAYGUN

Table of Contents

Better Issue Managment With ChatOps by Raygun

Published by Raygun.com - www.raygun.com

© 2016 Raygun Limited - First Edition

All rights reserved. No portion of this book may be reproduced in any form without permission from the publisher, except as permitted by copyright law. For permissions contact: support@raygun.com

SECTION 1

Introduction

The rise of ChatOps seems incredibly simple if you think about it in a real world scenario.

How hard can it be to assemble a group of people and make them sit around in a circle to discuss their web development projects? In person, with everyone in the same place, this is a simple task. Doing this online shouldn't be too hard, right?

If we try and replicate this over the internet, with people in multiple locations and over different time zones, or even in different areas of our office, things are suddenly going to get a lot messier.

We can't always be sat together in one location and have complete context on everything that is happening in our projects. Even if our team is located on a single floor of an office building, communication actually becomes a challenge as a company grows, and this has been the problem that chat related software has aimed to solve for us for many years. How do we make communication online easier?

Gone are the days where employees all had to be located at the same physical location. In today's super-connected world, employees could be located anywhere.

ChatOps, is a term originally coined by GitHub. A play on the popular DevOps term that incorporates Development and Operations teams, naturally ChatOps mixes Chat with Operations, allowing you to move everyday tasks performed in other applications or processes into the chat application itself, rather than having to switch between multiple applications to get the job done.

However, ChatOps is not just great for remote or dispersed teams. Located in the same office or not, chat applications connect people, tools and automation into a transparent workflow that helps teams monitor, build, and ship their products with speed and agility.

ChatOps really is for all teams - whether dispersed or not, and can make a massive improvement in communication for your organization. Rather than each team member completing a task and having to communicate this to all other team members, a chat application can make everyone aware of team activity at the same time.

How ChatOps can breed transparency

ChatOps enables teams to have full transparency on tasks, issues and progress, whilst also promoting a company culture that values every team member's opinion. Some may question if chat applications provide a distraction for teams, reduce productivity, or simply stop people talking to each other. The opposite is often true for teams that fully embrace ChatOps and make it the central nervous system to their software development workflow.

An additional benefit is to encourage teams to share knowledge. More extroverted team members can often dominate discussions in group situations, leaving the opinions of the introverts less likely to be raised.

We've seen first hand at our own company (Raygun) that although people have incredible ideas, they are not able to raise them in a group situation due to being more introverted than their more extroverted colleagues.

Individually, or when behind the safe harbour of their keyboards and participating in online group chats, this becomes easier, and team members are far more willing to share and speak up with their opinions. The medium they use to communicate may be different, but communication actually improves within the team, rather be affected negatively with the use of team chat programs.

Issue management within ChatOps

Your entire organization can reap the rewards that ChatOps offers, especially when it comes to the thorny subject of issue management. Whether that's a small bug in your application that does not require an immediate fix, or a full blown and severe outage affecting thousands of customers, communication is key, so how calm will your team be as the ship starts to sink?

Issue management and resolution doesn't need to involve panic, blame and miscommunication. The days of scrambling to resolve a problem in a cold sweat are behind us.

If teams can embrace ChatOps within their own organizations and automate many processes with the use of bots, third-party plugins and tools, they stand to make big gains in the efficiency of their wider team, and at the same time, make the management of issues in their software applications stress free, collaborative and fully transparent.

This ebook will help you build a better issue management processes around bugs, errors, incidents and outages which embraces ChatOps at its very core, ensuring all issues, whenever they arise, are dealt with effectively with full visibility and collaboration from your entire team.

The history of chat applications for software development teams

The original Talkomatic was the world's first multi-user online chat system and from here we grew into the early 90's, where we experienced 'Chat Wars'.

Microsoft might have killed off its Messenger service in recent years, but back in 1999 it was the centerpiece of a heated war between the company and AOL. At the time, Microsoft had just introduced its MSN Messenger Service as a direct rival to AOL's popular AIM instant messenger.

AOL kept blocking MSN Messenger from accessing AIM, and Microsoft created new ways to work around the blocks put in place. The battle resulted in AOL using a security bug in its own AIM software to prevent Microsoft from accessing its AIM service, a move that forced the Microsoft team to give up on any dreams of interoperability between AIM and MSN Messenger, but not before an unknown Microsoft employee posed as software consultant to try and get security experts to detail the AOL flaw.

Software services like MSN Messenger and AIM paved the way for today's intelligent business tools that allow tech teams to work in symphony wherever they are in the world. Location is no longer seen as a barrier to most software development companies and there are even hugely successful tech companies that do not even have a physical location for their business. They are fully remote and rely on team chat software to do business and ship software collaboratively.

DevOps meets ChatOps

DevOps is about knocking down barriers between developers and operations teams, and in doing so reducing organizational friction wherever possible. That's certainly the spirit behind ChatOps – the term for the hyper-collaborative way of running DevOps, including operating aspects of systems and infrastructure, through online chat.

If we are to understand the benefits of ChatOps, it's worth understanding a bit more about it's relationship with DevOps. Although still a relatively new concept, DevOps (Development and Operations) has become a hugely recognized term in the software delivery world. DevOps has the aim of creating greater efficiency in an organization through the four cornerstones of culture, automation, measurement and sharing.

By creating a culture where teams strive to automate tasks, measure and analyze everything they can with their fellow team members, putting this inside a chat program seems like a no-brainer, embodying all of the methodologies that DevOps aims to achieve. Teams get all the benefits of DevOps practices, packaged up into a collaborative tool. ChatOps therefore may have stemmed from the DevOps movement, but the practices go hand in hand for software development teams.

Born within GitHub, ChatOps evolved from its open source chat bot, Hubot, which has changed the way the web-based Git repository runs operations through the automation of deployment, graphing, monitoring, provisioning, and even mitigating security events.

This new way of working is empowering teams to become more productive, through conversation-driven collaboration.

SECTION 2

The Rise of The Machines (AKA - Bots)

GitHub, Inc., wrote the first version of Hubot to automate their company chat room. Hubot knew how to deploy the site, automate a lot of tasks, and be a source of fun in the company. Eventually it grew to become a formidable force in GitHub. But he led a private, messy life, so they rewrote him and made him open source.

Written in CoffeeScript on Node.js, and easily deployed on platforms like Heroku, Hubot is a standardized way to share scripts between everyone's robots. Hubot can create more efficiency in your development team and become an extra member of your crew, but it's just one of many bots that can work within your ChatOps environment.

Innovative teams have devised a new way of collaborating. Everything that was previously done via email to run multi-million dollar businesses and associated operations can be done with messaging in a chat application. Teams have evolved from email to chat and have replaced repetitive tasks with automation.

Whilst in a chat room, team members can type commands that the bot is configured to listen out for and execute. These could include notifications of successful builds, errors identified in your codebase that are affecting users or just to display cute images of kittens with random cat facts. The entire team collaborates in real time as commands are executed.

Source - https://hubot.github.com/

James Fryman, Senior DevOps engineer at StackStorm recalls his colleague Mark Imbriaco, who on his way to a conference in 2014 to give a talk on the power of ChatOps, was alerted by his phone while he was in the taxi to the airport that the GitHub infrastructure was under attack by a DDoS. Instead of doing the traditional on-call scramble, tearing out a laptop, calling folks up to respond, he calmly pulled out his phone, opened up the main chatroom, asked the bot to effectively "raise shields." Within two minutes, the bot had opened a tracking issue, called up and notified team members in the company, updated the CORE ROUTING TABLES on the production routers to enable DDoS defenses (local and upstream scrubbing), and gave our customers a heads up that we knew the attack was underway and being dealt with. The rest of the ride to the taxi was probably spent between querying for graphs in ChatOps, seeing if the attack was over, and aimlessly playing a game.

Feature quote from ChatOps for Dummies

The far reaching benefits of using bots for automated tasks

Bots have turned traditional automation via scripting on its head (e.g. a system administrator with a deployment script) - and moved this to a shared command / control infrastructure.

One huge advantage of allowing bots to take care of tasks within your team, especially larger sized ones, is in simplifying user permissions. The granting of permissions for individuals can be a manual task for any senior level engineer to undertake, and surely they have better things to be working on. With several people (even hundreds) on the team, it becomes a more complex task to keep track of which team members have permissions for different parts of the system.

People don’t need access to production environments directly if it is managed through a centralised bot. If the bot has the necessary permissions, anyone on the team can deploy their code using the right commands, new team members can be up and running within minutes rather than hours and user permissions become encapsulated by the bot itself.

Bots can require some initial setup, but it’s usually a fun task that developers enjoy taking on, with not too much effort involved to get up and running. Any investment in automation can also be reused or repurposed by the organization, and as it resides within the company infrastructure, the knowledge of how it operates does not disappear with certain employees who may come and go.

Ultimately, bots can create common and consistent workflows. They are reliable servants to your team’s commands and reduce the overheads and complexities human involvement can bring.

No matter how many processes you put in place, human error will always be a factor, so using deterministic scripts via a centralised command interface reduces this risk greatly.

Increased visibility using plugins and third party integrations

If you already use chat applications, are you making the most of third party plugins?

Many DevOps Software-as-a-Service (SaaS) products now integrate with ChatOps services via plugins and integrations. With just a few clicks, these can be set up to keep your whole team in the loop during the software development process.

These integrations help bring all notifications and alerts into a consistent place.

Things such as build status, deployment status, commits and merges in source control can all be given visibility to everyone on your team instantly. Constant visibility within your team ensures there is no confusion as to the status of any particular task along the software development production line.

Tasks that used to be done as a manual process can now be automated, and the benefits come in many shapes and sizes. No longer will you be exposed to human error, but you'll also not need to employ a human's time to move things forward. You can essentially put an end to error-prone hand-typed SQL statements or automate tests around repeated commands.

If you employ a continuous delivery process, this becomes even more powerful, as you can easily understand who and when a deployment was started without any extra follow ups or inefficient loss of time, because it's made visible to your entire team in one swoop of an automated bot.

This has a huge efficiency benefit for teams that deploy code continuously throughout the day.

ChatOps and bots also benefit other areas of your company. You can ensure marketing and sales know when new features are shipped, get sales information piped through to give transparency on deal size value and support teams can have overviews on incidents as they happen.

If your company can embrace the value that ChatOps bots can offer, there are huge gains to be made in efficiency and team communication, and the best part, it's completely automated.

While chat bots do the commanding, sometimes you also have deployment servers listening for these commands, doing the heavy lifting of executing deployment tasks as background jobs.

Have you ever had the pain of trying to find out who a command was run by on your team? ChatOps gives your team full visibility of what's going on during your software development process with searchable history. This helps teams remain transparent and put everyone's actions in full view of their colleagues.

The results? Visibility across the board.

SECTION 3

Better Issue Management With ChatOps

Even the greatest software teams and products experience errors, issues and outages at some point - it’s an unfortunate but inevitable fact, no matter how hard we all try to avoid these situations. I’m yet to meet a software development team who are actively looking for ways to take the product offline or cause a user unnecessary misery when their app won’t work, but what separates the best from the rest when it comes to creating reliable and robust software is how teams deal with those problems when they arise and how they minimise the affect on end users.

When it comes to incident management, you should have your own processes in place (hopefully), but we all dread that notification ping at some unreasonable hour of the night. Every minute counts trying to rectify the issue.

Alerts are just the beginning

It’s one thing to be alerted to an issue, that’s just the notification part, but without the diagnostic details about the problem, we can spend a whole bunch of time just trying to work out the cause whilst our users continue to see problems, support queues start to fill up and the issue grows into a major company wide incident.

If we can get this information immediately at the time the error notification comes through, fixing and debugging can be done at record speed and your whole company can benefit, as well as your end users.

Many developers may automatically assume this only extends to servers falling over, database timeouts and more, but it can also include production errors your users are encountering everyday. If you knew that a recent update has broken your billing system and customers couldn’t pay you, surely you’d be scrambling to fix that pretty quickly!

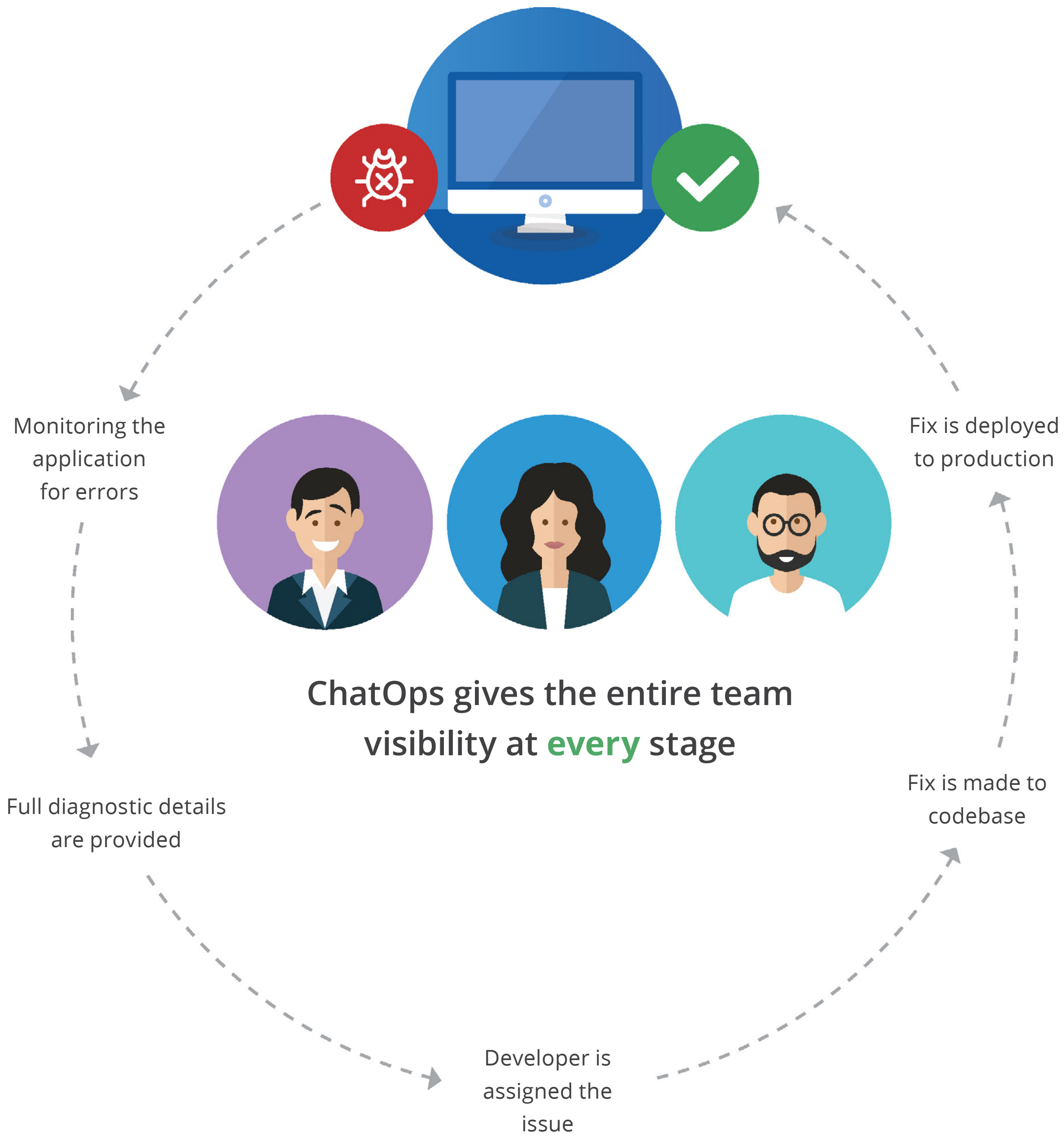
If you already use team chat applications in your business, you might already have several pieces needed to get an awesome issue management workflow operational, connecting these pieces together can create a world-class incident management workflow and ensure our entire team is following along from first alert to resolution.

Reaching issue management heaven

How do we use ChatOps to create a faster, easier and more transparent workflow around issue management and fixes? After all, most teams will have #war-room or #devops team room set up in their ChatOps solution. If you don’t, maybe it’s time to create one? But the conversation can often be one of confusion, blame or panic when incidents occur.

Here are the steps that will ensure issues are dealt with in an effective manner, no matter what time of the day or night your team is having to deal with them.

This simple diagram shows how popular DevOps and ChatOps tools can streamline your workflow.


Monitoring the application for errors
Full diagnostic details are provided
Developer is assigned the issue
Fix is made to codebase
Fix is deployed to production
ChatOps gives the entire team visibility at every stage

If we are using an automatic and real time error monitoring and crash reporting software within our application, we can have issues raised to us automatically. Crash reporting software is your safety net against the huge risks of exposing your customers to broken software.

About Raygun Crash Reporting

Raygun's error and crash reporting software silently monitors your web and mobile applications, collecting all error and crash events that are affecting your customers. When issues are found they are presented on your crash reporting dashboard, with detailed diagnostic reports about every single error and crash, making digging through log files and trying to replicate issues a thing of the past.

Contextual information about the problem is made available to your entire team instantly, including which specific users of your application have been affected. Raygun then provides your team with a seamless workflow to solve the problem quickly. Should the inevitable happen and users are exposed to outages, bugs, errors, crashes or bad deploys, your entire team will know about it instantly with Raygun's smart alerts via email or team chat notifications, giving you all the diagnostic information you need to fix the problem quickly and efficiently.

"Without Raygun we had no clear window into what errors our applications were throwing. We only knew by digging through logs when we got the time, or by our customers letting us know something was wrong. Raygun takes the unknown and makes it known"

Daniel Hoenig - Software Architect at Schneider Electric

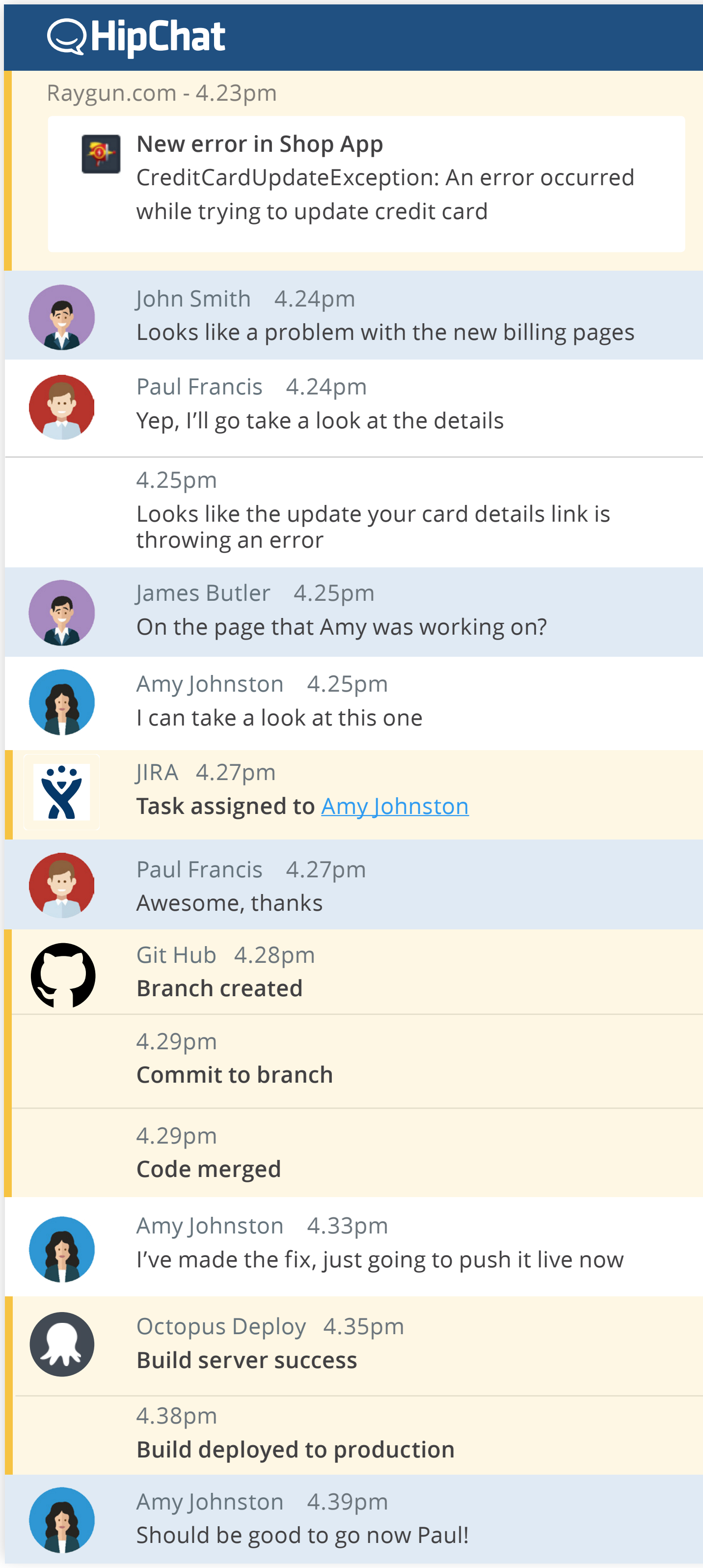
When an issue is identified in your application, this could mean that a single user has been affected by a bug, or it could mean that our entire application is currently experiencing a severe issue which is affecting thousands of users.

As errors are being pulled into Raygun, we can also see real-time notifications appear in our HipChat team room by connecting the two services through their integration. This allows your entire team to have visibility of the issue and a handy link to the diagnostic details for each error occurrence.

When everyone has visibility of issues affecting your applications, communication becomes seamless.

Issues can be created in project management tools like JIRA and priority levels set. If we have a plugin enabled we can also send through JIRA activity into our team's HipChat room.

We can make then make the necessary changes in our codebase. As Raygun has given us the diagnostic information, right down to the line of code the issue stemmed from, finding out exactly what was causing the problem can take


HipChat

Raygun.com - 4.23pm

New error in Shop App
CreditCardUpdateException: An error occurred while trying to update credit card

John Smith 4.24pm
Looks like a problem with the new billing pages

Paul Francis 4.24pm
Yep, I'll go take a look at the details

4.25pm
Looks like the update your card details link is throwing an error

James Butler 4.25pm
On the page that Amy was working on?

Amy Johnston 4.25pm
I can take a look at this one

JIRA 4.27pm
Task assigned to Amy Johnston

Paul Francis 4.27pm
Awesome, thanks

Git Hub 4.28pm
Branch created

4.29pm
Commit to branch

4.29pm
Code merged

Amy Johnston 4.33pm
I've made the fix, just going to push it live now

Octopus Deploy 4.35pm
Build server success

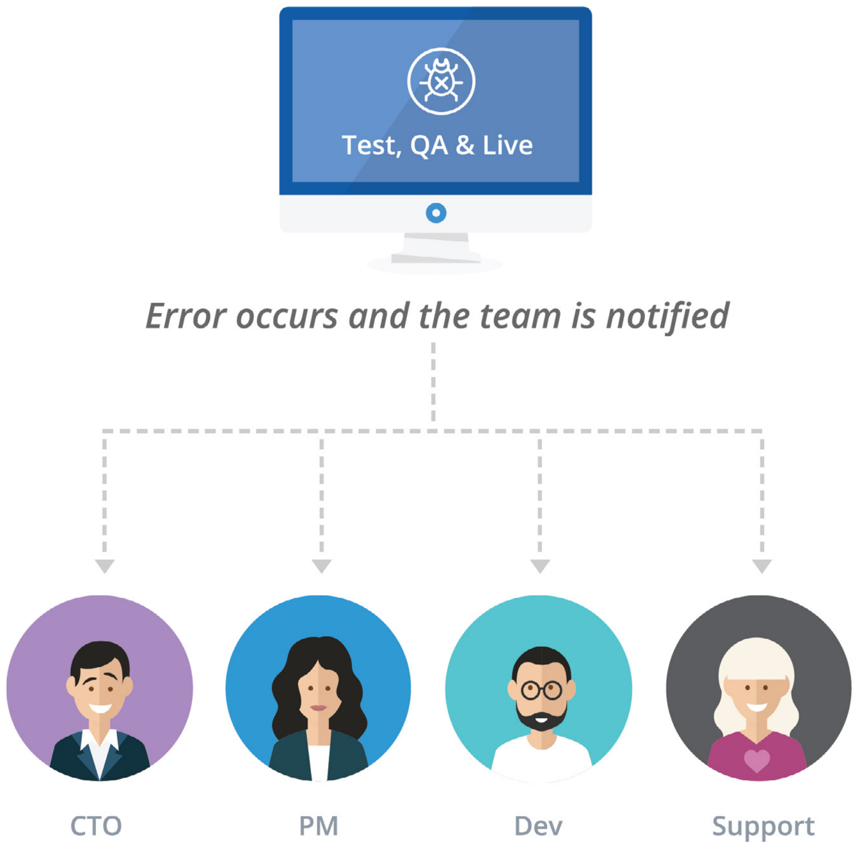
4.38pm
Build deployed to production

Amy Johnston 4.39pm
Should be good to go now Paul!

a few minutes. If you've ever had a major incident with time pressures to put out a fix, you're going to be thankful you had this kind of solution in place!

As we deploy the code to production, our automated bot can manage the build and deployment process. You'll notice here that there is no Quality Assurance (QA) step involved. Utilizing this workflow, if the issue re-occurs or causes further problems, Raygun will still be listening out for errors in production, effectively removing the need for a QA step.

We've now gone from discovery of a software issue to a resolution, with our entire team being given full visibility of the situation. ChatOps made this whole process easy, transparent and stress-free.



SECTION 4

Summary

Software development teams globally have been embracing ChatOps for many years now, but at the same time, many teams are still yet to optimize their ChatOps set up to become more productive as a team and increase the visibility of issues across their entire organization. They may already use team chat software, but the use of automated tools like error monitoring and crash reporting software, issue management, deployments and custom bots is not fully utilized.

Organizations may have doubts over the benefits ChatOps bots and plugins can offer their software development and wider team members. After all, how can a bot make a team more efficient? There are many advantages to having a bot take over some simple tasks in your workflows.

Bots for one, don't get upset or have opinions on the right or wrong way to do things. They stick to the simple fact and logic based commands that they are instructed to execute. Nobody can point to a bot as a source of blame for oversights and incidents that would have typically been caused by human error. Bots take human error out of the picture so to speak and will always be there to do their dedicated tasks reliably and with precision.

Bots also don't need to take holidays, they don't get sick, require training or deep knowledge sharing to get them up to speed with the process. Should only one member have a deep understanding of your build process or could be the only one to react to issues on your infrastructure, your organization is at risk should that individual team member be on holiday, out of office or completely leave the company.

A bot that can respond to and execute laborious but complex tasks and steps, which could well end up saving your team huge amounts of work.

The aim of ChatOps is to make your team more productive and do most of the heavy lifting within the team chat tool itself. Utilizing DevOps tools that integrate with your existing team chat provider (or new one) will make your team's life so much easier, you'll wonder how you ever went without the smart automation and time savings they can offer.

The future of ChatOps will be interesting, with so many tools connecting together through plugins and APIs, perhaps we will see the day when all software development is completely automated. Until then we can rely on ChatOps to provide us with greater productivity, better team communication and full visibility of issues when they occur in our applications.

SECTION 5

Resources and further reading

- What is ChatOps? A guide to its evolution, adoption, and significance
- History of Chat Inforgraphic by Sameroom.io
- ChatOps for Dummies ebook
- Chat Wars - Microsoft vs. AOL
- ChatOps: How Software Teams Manage Issues
- Inside Atlassian: How IT and SRE use ChatOps to run incident management
- Some common chat bots that are all open source
  Hubot: GitHub's bot written in CoffeeScript and Node.js
  Lita: Written in Ruby
  Err: Written in Python