ENGINEERING

# Why engineering teams love Slack (and why yours might love it too).

The adaptive collaboration hub for software engineering

slack

# Whoa, that happened fast

**Back in 2009, we were a small team of software engineers, building a massive multiplayer game called Glitch.**

We started using Internet Relay Chat (remember IRC?) to keep everyone on the same page. As work on the game progressed, we kept wishing our IRC channel could do more than just basic communication. So we found ourselves tweaking it, adding to it and hacking new ways to get things done faster.

Well, the game flopped, so we decided to focus on this new collaboration thing we'd developed.

That was a good call. Because, we ended up with a great product-market fit: a really effective way for teams to work together.

Maybe that's because we weren't trying to "design" anything. There was no ego, and no speculation about some fictional user. We were the users.

That's how Slack was born.

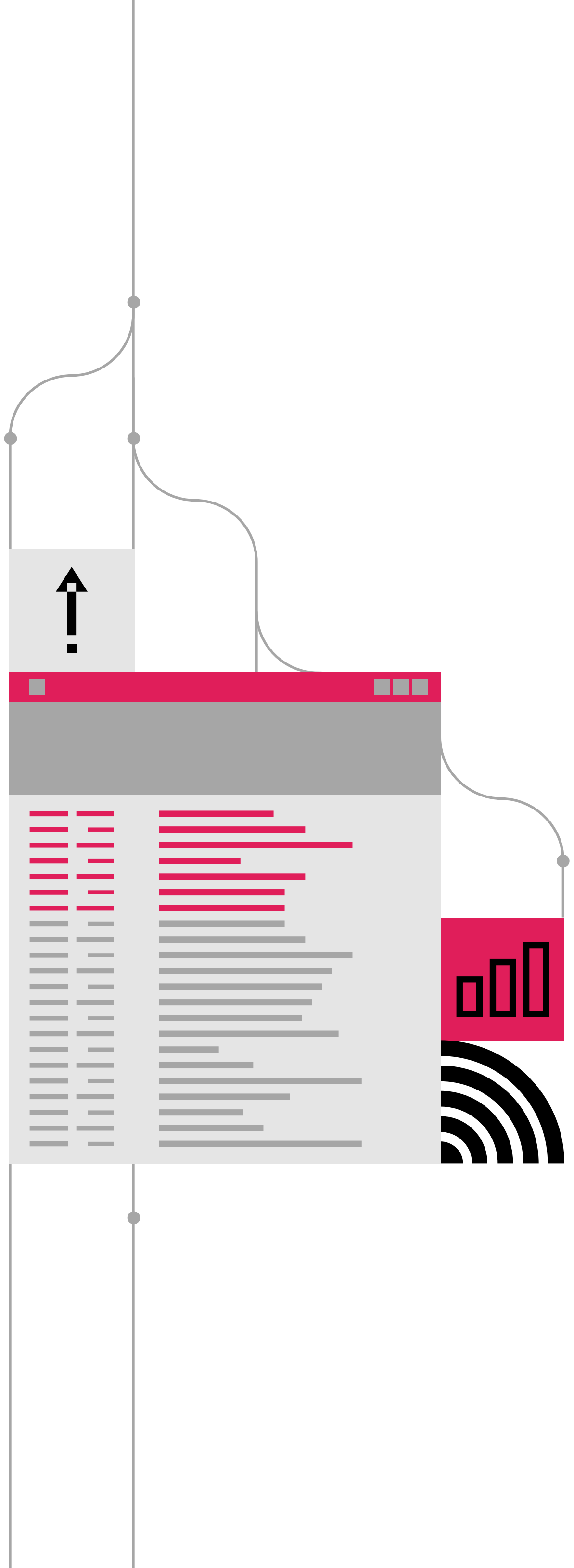
Today, it's become bigger than we'd ever imagined—and its adoption by dev teams of all sizes is a major reason.

Yes, Slack is used in pretty much every department and discipline, but software engineering is where it all started, and is still the source of a lot of the user love that keeps us coming to work smiling.

It's hugely gratifying that we're able to create something that makes a direct impact on the daily working lives of developers.

This e-book is a quick introduction to some of the ways Slack helps developers.

We hope it helps explain why software engineers seem to like Slack so much.



# Why Slack fits software engineering so well

**Slack is used by all kinds of non-technical teams, every day. It seems to organically adapt to the work it's supporting.**

But Slack seems to fit software engineering like a glove. After all, this is a pretty specialized kind of work.

Software engineering is:

- **Complex**
  With lots of moving parts
- **Highly iterative**
  Proceeding in sprints and cycles
- **Collaborative**
  With design, devs, product people and QA folks in constant touch
- **Distributed**
  Often involving people in different places and time zones
- **Open**
  With sharing as a default setting
- **Increasingly automated**
  Supported by lots of tools and cloud services

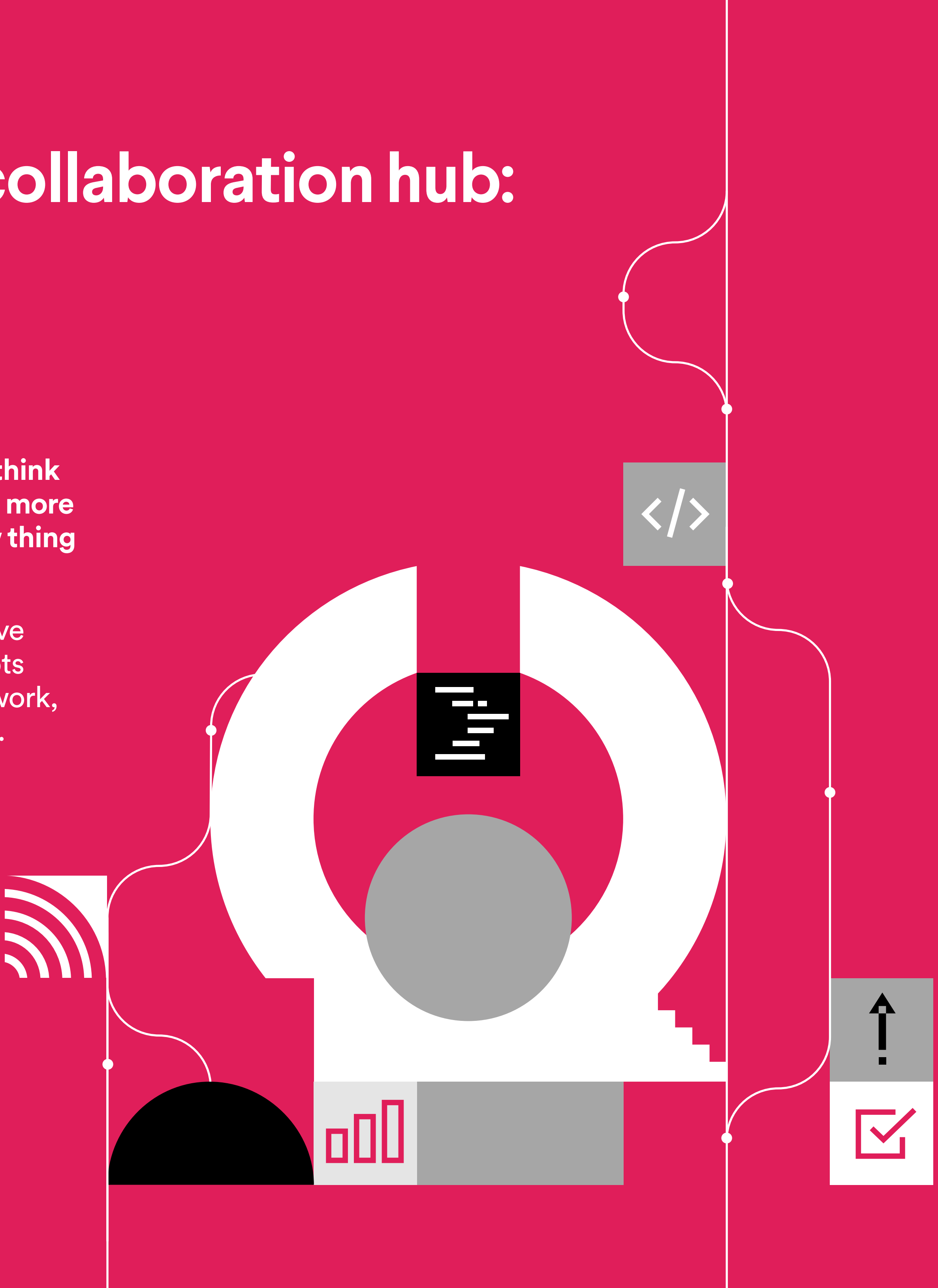
When you think about it, all work is getting more and more like this. But software engineering is an ideal use case. It demands a kind of collaboration that just can't happen with email and face-to-face meetings.

This kind of work needs a new kind of collaboration.

# The adaptive collaboration hub: a new thing

**People who've never used Slack think of it as a messaging app. It's way more than that. In fact, it's a whole new thing that hasn't existed before.**

We've taken to calling it an adaptive collaboration hub because it adapts to the way different teams like to work, to their existing software choices... and to change.



## An adaptive collaboration hub combines three things in one tool

### 1.

**Channel-based messaging**
This lets teams spin up channels dedicated to specific tasks, projects or issues. Like a #devel-new-site channel where all developers meet to work on the new website. Or a #triage-mobile-app channel where teams work together to squash bugs on the mobile app.

**Channels are way better than one-to-one messages or closed email threads because they make it easy to include the right people in the right topics at the right time.**

### 2.

**A searchable knowledge store**
A single place where anyone can find all relevant documents, conversations and decisions—like those product specs or the discussion around that new feature.

**Email attachments usually sink out of view to everyone but the people copied. Knowledge is only valuable if it's discoverable.**

### 3.

**An integration layer**
A place where the software your people spend the most time in (like GitHub, Jira, Jenkins and Trello) hooks into the place where work is discussed every day.

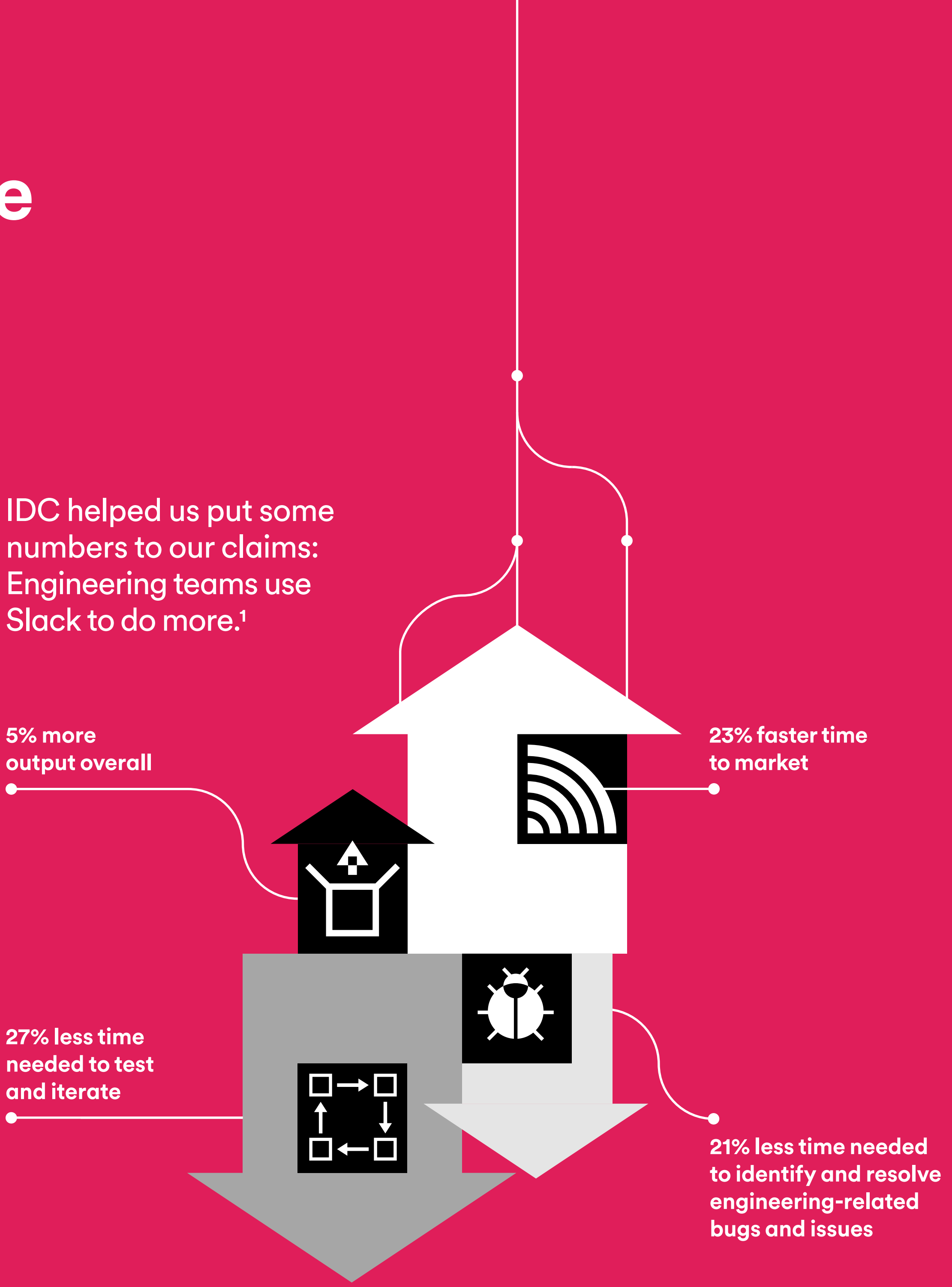
**This minimizes the constant context-switching that comes from working across many different apps. Instead of forcing your people to go to the apps, bring the apps to your people.**

**Note:**
**Bringing these three things together in one place makes each one of them far more powerful. The hub is far greater than the sum of its parts.**

# Benefits to software engineering teams

**The right collaboration hub directly influences the things that are most important to every engineering team: better code, delivered faster; more efficient bug squashing; and a better developer experience (so you keep your talent happy). Any software that helps you do all that is probably worth looking into.**

IDC helped us put some numbers to our claims: Engineering teams use Slack to do more.[1]



**"Slack is a living documentation hub, and everything is searchable."**

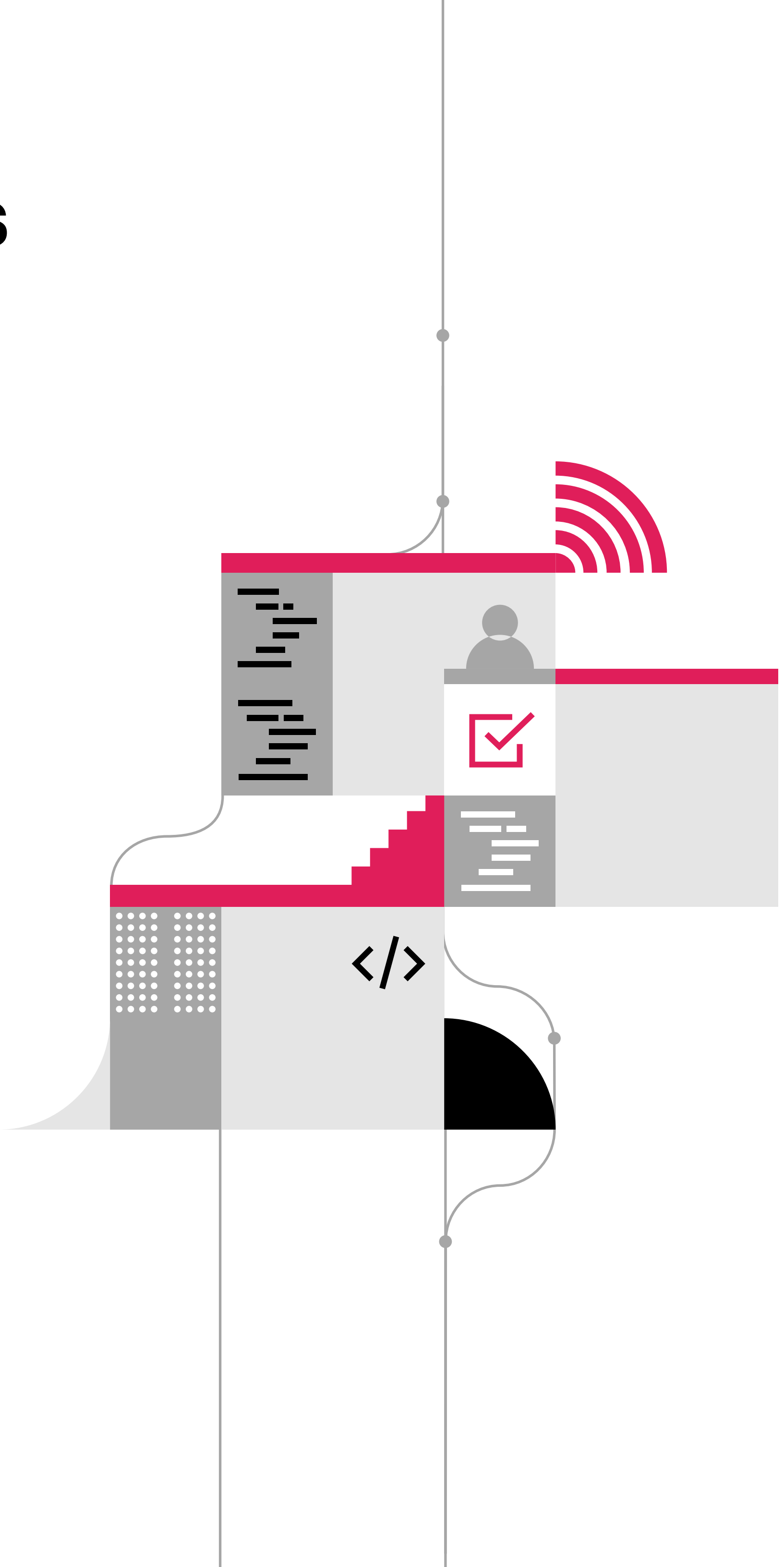
Malika Rajvanshy, senior engineer, Slack

1. IDC research, "The Business Value of Slack, 2017", sponsored by Slack

## How Slack streamlines the whole software engineering process

**We're undoubtedly some of the most sophisticated users of Slack for software development—all our dev teams live in it. But every day we still hear about new use cases and interesting apps and integrations that software teams are using.**

Let's walk through some of them, organized by stages in the software development cycle.



# Plan

## Slack helps product managers, designers and engineers agree on what they're building and why.

- **Kick off the whole process with a single channel for a new product or feature**
  Maybe it's called something like #feature-new-app.

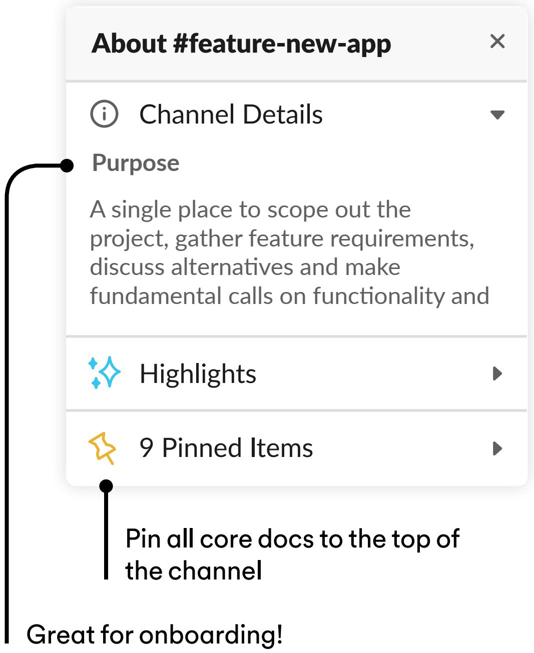
  Now there's a single place to scope out the project, gather feature requirements, discuss alternatives and make fundamental calls on functionality and UX.

- **Sharing documents here makes everything discoverable**
  For all contributors and any new joiners.

  *Slack integrates beautifully with Google Docs so all docs are a click away.*

- **Got a question? Pop it into the channel**
  Launch a discussion and come to a resolution for all to see. Now there's a permanent record.

**A #feature_[name] channel is where it all happens**



About #feature-new-app

Channel Details

Purpose

A single place to scope out the project, gather feature requirements, discuss alternatives and make fundamental calls on functionality and

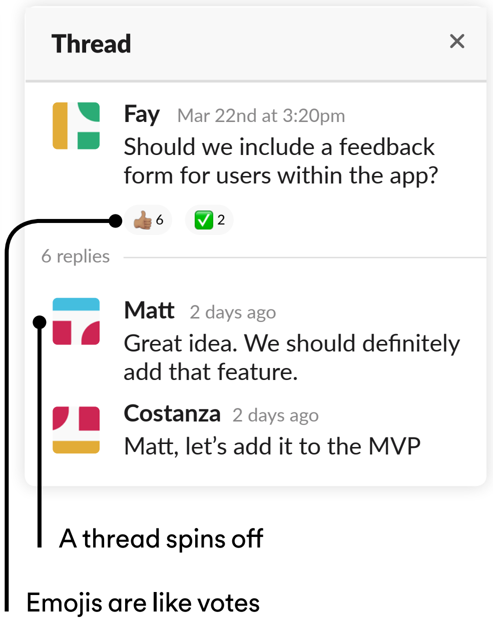
Highlights

9 Pinned Items

Pin all core docs to the top of the channel

Great for onboarding!

**Launch a discussion around a proposed new feature**



Thread

Fay Mar 22nd at 3:20pm
Should we include a feedback form for users within the app?
👍 6 ✅ 2

6 replies

Matt 2 days ago
Great idea. We should definitely add that feature.

Costanza 2 days ago
Matt, let's add it to the MVP

A thread spins off

Emojis are like votes

# Code

**Slack helps devs orchestrate the many moving parts of a large code base, speeding up development and improving quality.**

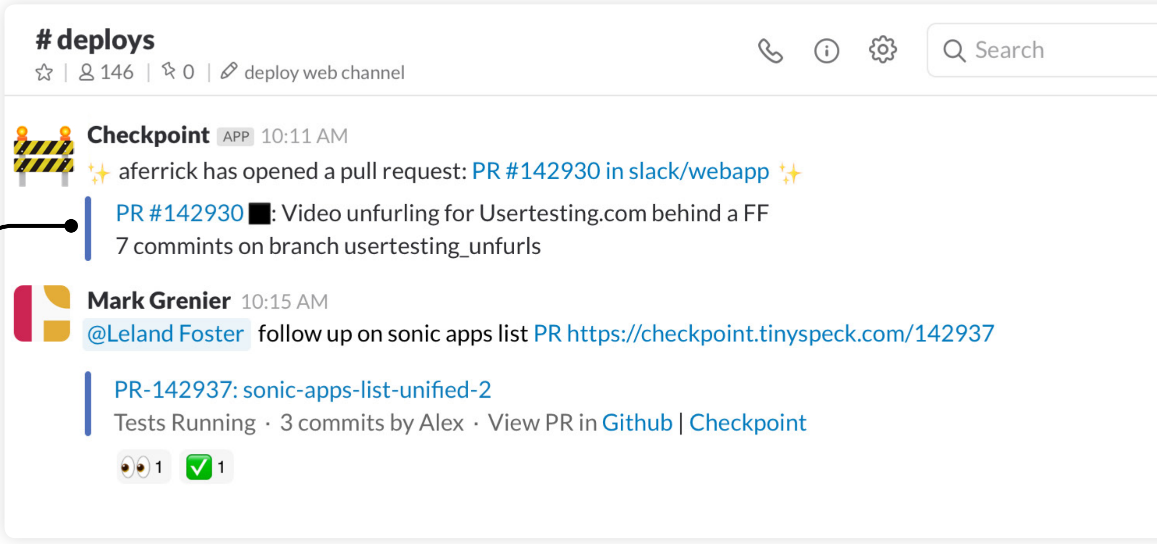
**When it's time to start coding, Slack makes sure the whole team is working together:**

- **A #devel-product-name channel is the home for everything**
  Including day-to-day work across engineering and QA; pull requests, code merges, design revisions, daily stand-ups, discussions, etc.

- **A central hub for code review**
  Slack supports whatever process you use for branching, merging, reviewing and releasing code, whether that's developing on version branches, feature branches or from a merged master.

  *Git integrations (with GitHub, Bitbucket or your chosen repository) bring all change alerts into Slack.*

**Git integrations keep everyone up-to-date**



# deploys
☆ | 146 | 0 | deploy web channel

Search

**Checkpoint** APP 10:11 AM
✨ aferrick has opened a pull request: PR #142930 in slack/webapp ✨
PR #142930 ■: Video unfurling for Usertesting.com behind a FF
7 commints on branch usertesting_unfurls

**Mark Grenier** 10:15 AM
@Leland Foster follow up on sonic apps list PR https://checkpoint.tinyspeck.com/142937
PR-142937: sonic-apps-list-unified-2
Tests Running · 3 commits by Alex · View PR in Github | Checkpoint
👀 1 ✅ 1

Open a pull request and you get pinged in Slack whenever someone comments on it

**A new kind of standup**
Standups are an important part of agile development but they don't have to be face-to-face. Dev teams use Slack for standups—whether every morning or every week—and have F2F meetings only when they make sense (for many devs, the best meeting is a canceled one).

*Integrations with software like Standuply automatically push summary reports into Slack so your teams can share goals and tasks; track business metrics; post meeting notes; and monitor the team's progress and happiness.*

**Promote code reuse**
Code reuse is a core principle of efficient engineering teams, but it's a challenge when you have hundreds of developers contributing to many different products.Before writing any new code, your devs can search across all Slack channels to see if anyone else has already built something similar. Next step: ask in the right channels, "Has anyone made a date picker yet?" Stop re-coding the wheel.

**Create and share code using snippets**
Snippets make it easy to share code, configuration files and log files directly in Slack. Colleagues can download them, view the raw file, and leave comments.

## Slack in action

# Extensibility at its core

**Slack is a collaboration hub. That's what it's great at. It doesn't try to do the work of the software your teams already use—like Trello, GitHub or Jenkins.**

Instead, Slack simply unites all these different apps, bringing the relevant information from them into the channels where the work is being discussed (and inviting actions in those applications, triggered from inside Slack).

These integrations help developers do what they love to do: *create systems that just work.*

The examples shared throughout this e-book are just that: examples. There are as many ways to use Slack as there are software teams using it.

**Bots: apps get conversational**
A bot is a type of Slack app designed to interact with users via conversation. It's the same as a regular app. It can access the same range of APIs and do all the things that a Slack app can do. But when you build a bot for your Slack App, you're giving that app a face, a name and a personality, and encouraging users to *talk* to it.

Your bot can send DMs, it can be mentioned by users, it can post messages or upload files, and it can be invited to channels (or kicked out).

**"Anytime I've seen a Slack integration I've turned it on. It's provided so much value and helped us save so many extra steps in our process."**

Thomas Lawless, senior software engineer, IBM

# Test

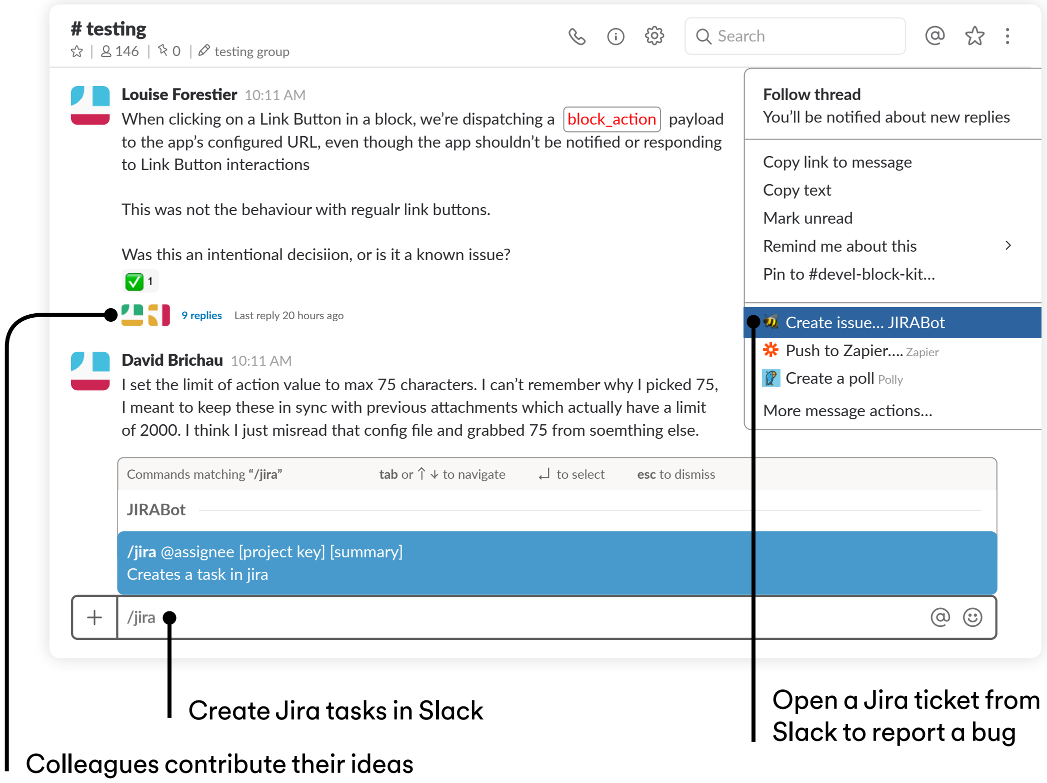
## Testing is woven into the modern development/ deployment process. Slack supports a dynamic, collaborative, transparent approach to testing.

**Continuous integration runs your testing suite against every merge with each new chunk of code. Slack streamlines the process in lots of ways, big and small:**

- **A #testing–feature channel coordinates QA**
  Let the QA team collaborate with devs in an open forum.
- **Jira integration automates test workflows**
  Capture issues in Slack and get them into the pipeline automatically. Send customizable notifications from Jira into your channels. Quickly assign issues to people and know they're recorded where they belong.

  Some teams use Slack to automatically move change requests into a new channel, updating Trello or Asana at the same time.
- **Break out a channel for each client**
  With dedicated testing channels for iOS, Android and web.

**Integrate with Jira to surface project progress**



**Louise Forestier** 10:11 AM
When clicking on a Link Button in a block, we're dispatching a `block_action` payload to the app's configured URL, even though the app shouldn't be notified or responding to Link Button interactions

This was not the behaviour with regualr link buttons.

Was this an intentional decisiion, or is it a known issue?
✅ 1

**David Brichau** 10:11 AM
I set the limit of action value to max 75 characters. I can't remember why I picked 75, I meant to keep these in sync with previous attachments which actually have a limit of 2000. I think I just misread that config file and grabbed 75 from soemthing else.

## Slack in action

# Working with Jenkins

**A lot of teams use Jenkins as their continuous integration server. It didn't take long for them to work out new ways to integrate Jenkins with Slack to automate all sorts of routine development tasks.**

One example: A software team's custom Slack integration spins up a Jenkins server running a big testing suite whenever a developer opens a pull request.

When the tests have been run, the notifications pop up in the right Slack channels. If the code fails the test, a notification is sent to the developer.

**Pipeline traceability**
Running code reviews, testing and deployment from Slack creates and maintains an up-to-date record of the entire pipeline from the beginning. If you hit a problem, the team can review the tests and merges, and roll back to a happier state until the code is fixed.

When code reviews are being performed or pull requests are being checked, comments and discussion happen on GitHub. But any comment is also posted into Slack and alerts the owner of that pull request. So the exercise is done on GitHub, then preserved as a copy in Slack.

# Release

## Slack helps push code to production by helping automate the workflows and notifications.

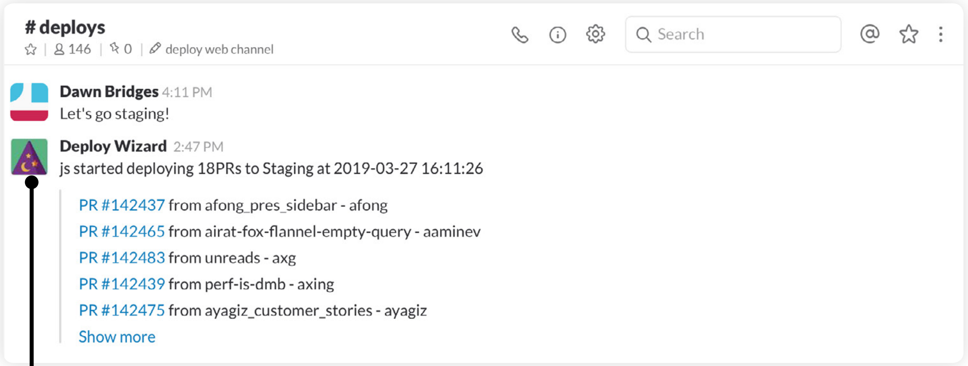
**Continuous delivery always calls for lots of small code releases, deployed frequently. Slack helps engineering teams streamline some of that.**

An example: One of our own software teams wrote an app, called Deploy Wizard, that integrates with ops and communicates the status of the code in channel. It starts with a "canary" stage (a tiny release to catch any sudden fails), then progresses through 10, 25, 75 and 100% of the user base.

Deploy Wizard pings the right developers and channels in Slack as the deployment progresses. The whole thing is managed by the on-duty deploy commanders (trained engineers, working on three-hour shifts).

If developers want to test their code in the staging environment, they specify that with their merge request. The deploy will stop in staging until a developer reports that he or she has tested the code into the deploys channel.

### A deployment progresses to full production



Dawn Bridges 4:11 PM
Let's go staging!

Deploy Wizard 2:47 PM
js started deploying 18PRs to Staging at 2019-03-27 16:11:26

PR #142437 from afong_pres_sidebar - afong
PR #142465 from airat-fox-flannel-empty-query - aaminev
PR #142483 from unreads - axg
PR #142439 from perf-is-dmb - axing
PR #142475 from ayagiz_customer_stories - ayagiz
Show more

Deploy Wizard automates notifications: All is going smoothly!

**Deploy right from Slack**
Some dev teams use slash commands (like /deploy_productname_staging) to trigger a deploy right from Slack. Automated messages show when the deployment has succeeded, with a link to go and check it (or a button to push it to production).

# Operations

## Dev teams use Slack to triage trouble tickets, swarm around issues and squash bugs.

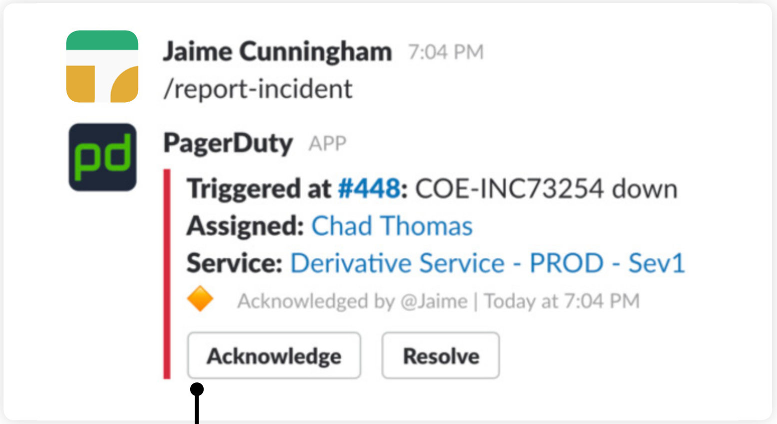
- **All issues flow through the #triage product-name channel**
  Including reports from customer support (manual or via integrations with tools like Zendesk).

- **Integrations bring all alerts into one place**
  Instead of expecting devs to monitor email or check into dashboards, Slack becomes the single place where all alerts find the best people to respond.

  Aggregating PagerDuty events or Asana tickets and posting them to the right channels reduces incident resolution times and creates a triage trail. Team members can work together on triggering, viewing, acknowledging and resolving incidents right from Slack.

  Similarly, Slack can pull all web, transaction, server and mobile alerts from New Relic into a Slack channel for fast response. Anyone curious about the incident can just pop into the channel and read about it. This reduces managers interrupting the incident responders for constant updates. It's all there.

PagerDuty events reach the right people fast


**Jaime Cunningham** 7:04 PM
/report-incident

**PagerDuty** APP
**Triggered at #448:** COE-INC73254 down
**Assigned:** Chad Thomas
**Service:** Derivative Service - PROD - Sev1
Acknowledged by @Jaime | Today at 7:04 PM

Action buttons accelerate resolutions right from within Slack

**Emojis and reacjis help triage issues and trigger workflows**
Reacjis are an efficient way to capture the responses of team members—but they're also a way to trigger automated workflows.

An app collects these so they can be aggregated, flagged and actioned. Any open issues (eye emoji but no check mark) are shown in PagerDuty.

**At Slack, we use:**

👀 = "I'm looking"

✅ = "Resolved" (for triage)
or "Approved" (for pull requests)

And we built a bot that collects and reports on these in a dedicated channel.

**For sharing bug reports, we also use:**

🔴 = Urgent!

🔵 = A question or non-urgent problem

⚪ = Sharing feedback, no immediate action

**An automated #decisions channel**
Some teams use the gavel emoji to indicate when a decision has been made. A bot then pushes all these decisions to a #decisions channel, where management can see the flow of decisions—and team members can easily search.

## Slack in action

## The people side

**Software engineers are in demand. To keep your talent, you need to give them the best employee experience you can.**

The right collaboration software can play a big role in this: helping reduce work friction, foster transparency, automate routine tasks and help work across teams.

Talk to any software engineering team that uses Slack.

Ask to see how they use channels, apps and integrations.

Then ask what they'd do without it.

## Slack in action

# Onboarding new devs

**Two new developers join the team. How do you get them up to speed?**

**Old way:** Lots of onboarding meetings and a bunch of forwarded email threads to try to figure out. Good luck with that.

**New way:** Invite them to the #dev–new–product channel, to review the pinned posts, like:

- The product spec
- The tech spec
- The designs

*(If these are Google Docs, DropBox or OneDrive, they'll always be kept up to date).*

They can also scan through all previous conversations, decisions and the people involved.

Now *that's* how you onboard a new dev.

# That's how software engineers use Slack

**So that's our quick tour of how Slack helps software teams streamline, automate and accelerate their work.**

We hope we got across the main points:

- **This is a new thing**
  An adaptive collaboration hub helps engineers work in new ways. It's way more than a messaging app.
- **It's super-flexible**
  Letting your teams "make it their own" with workspaces, channels, apps and integrations that reflect the way they like to work.
- **It helps you get more from your existing software**
  From GitHub and Bitbucket to Jenkins, Jira, PagerDuty, New Relic, Zendesk… whatever your devs, product, QA and support people use, they'll use those tools more efficiently by bringing their work together in Slack.
- **It adds value at every stage of the development cycle**
  From planning to developing, testing, operating, deploying and bug squashing.
- **Software engineers love it**
  Which means they'll adopt it and expand their use over time (delivering more value to the business).

If you'd like to see more, set up a demo—or ask one of our devs to show you around our own Slack instance. We're proud of it.

**"We have what we like to call an 'end-to-end delivery pipeline' that starts with source code and goes all the way through to *production* deployment. And now we have Slack integrated into all the key milestones in that process."**

Thomas Lawless, senior software engineer, IBM

# Learn more

**How Slack helps devs squash bugs**
Learn how dev teams manage incidents and outages better with Slack

**The Slack App Directory**
Take a browse

Or maybe we should talk.
**Schedule time with our team.**

LET'S GO

# About Slack

Slack is a layer of the business technology stack that brings together people, data and applications—a single place where people can effectively work together, find important information, and access hundreds of thousands of critical applications and services to do their best work.

From global Fortune 100 companies to corner markets, businesses and teams of all kinds use Slack to bring the right people together with all the right information.

slack