

Modernising support comms

Practical ways to work outside the inbox





Hello!

I'm April

- Managing Director, Knowledge Bird
- Investment manager, Pick & Shovel Ventures

You can find me at [@knowledgebird](#)



1997 - Technical support & network operations at One.Tel

1999 - Broadband analyst, Optus@Home

2006 - IT support & batch operations, Police Bank

2011 - Knowledge management consultant & KCS trainer



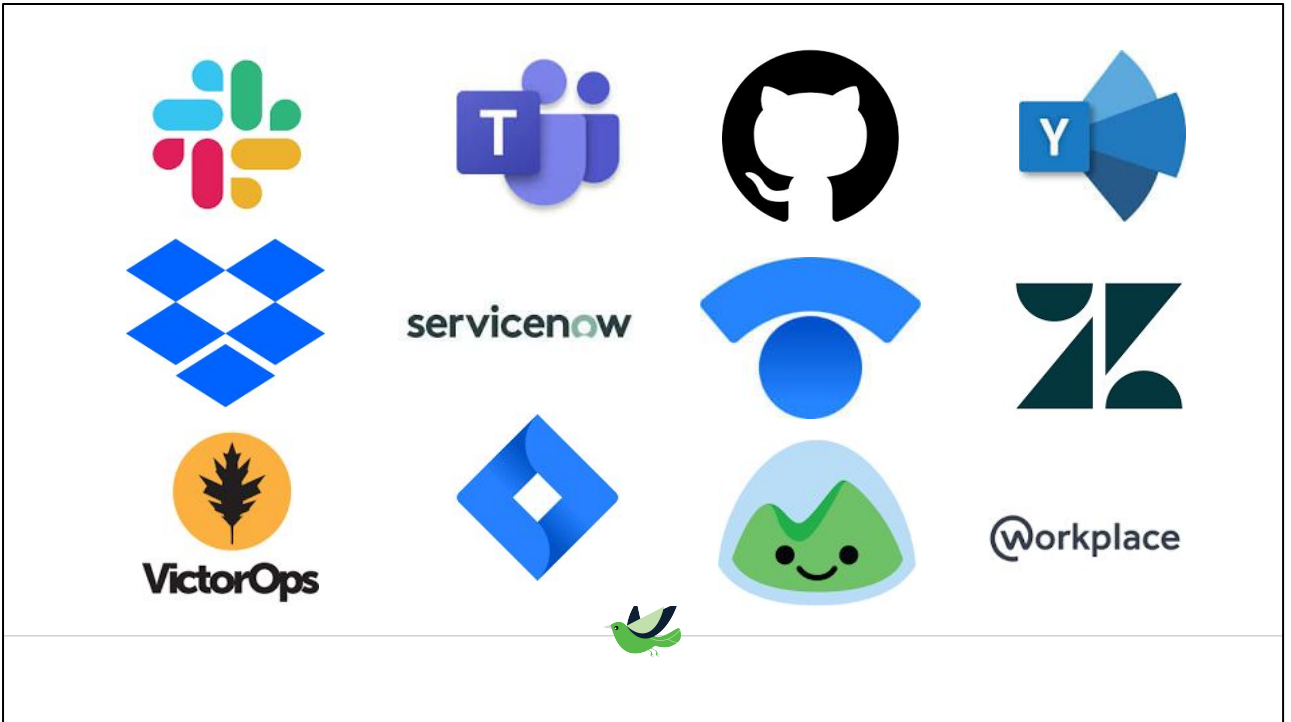


KM tip for online meetings

Use the chat function to facilitate questions & observations during the call

As someone speaks, post your questions and comments and wait for the meeting facilitator to call on you. This method streamlines the meeting and can provide a record of the key points captured.

Zoom tip from Patrick Lambe of Straits Knowledge. Also Zoom does allow you to save the chat as a text file.



Your knowledge management capability within IT considers how we interact with tools like these and how they interact with each other to get more from them --enabling better decisions, saving you time, and better serving the needs of your teams and your customers.

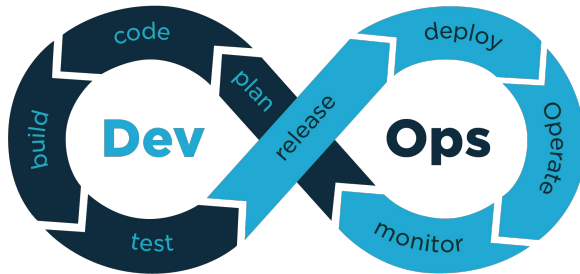


What we'll cover

- Chatops
- Communicating with customers
- Leveraging automation for your change board
- Speeding up resolution with swarming
- What makes Intelligent Swarming intelligent
- Gluing it all together - discussion & questions



DevOps



Streamlines the movement of software change through the build, validate, and deploy and delivery stages, while empowering cross-functional teams with full ownership of software applications – from design through production support.

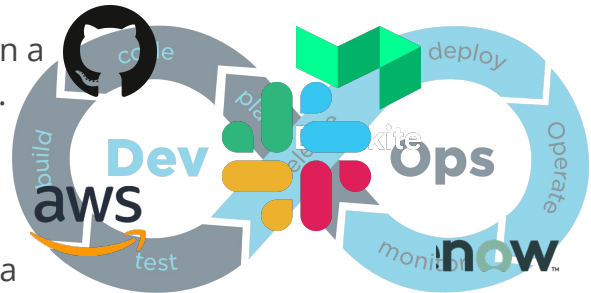
<https://medium.com/tech-tajawal/devops-in-a-scaling-environment-9d5416ecb928>



ChatOps

It's where notifications and systems commands occur in a collaborative chat platform.

It's a central place where teams can trigger automations and maintain a system of record.



The sort of automated interactions you could have within a ChatOps environment are automated tests and rollbacks, automated server builds & checks, automated environment config checks, automated change management for standard changes, backups...really, it's only limited by the constraints within your ecosystem.



Caveat

- Scripts/bots interacting with external APIs may be encountering errors without reporting.
- If you rely solely on ChatOps to trigger commands, you risk forgetting how to use native commands and interfaces.



It's all about me

Use status pages to provide proactive information about defects & degradations



Hosted status pages have become an industry standard. Companies like Apple, Reddit, Dropbox and New Relic all have status pages. It's a page customers can go to for a quick traffic light view to confirm whether or not they're being affected by any current degradation or service outage. It's a channel for transparent incident communication and it's effective at lowering operational costs because it reduces volume of repetitive calls and emails to the service desk, as long as it's communicated in a way that's relevant and understood by your customers and users. When customers have a clear understanding of whether or not, and how a degradation affects them, they feel more positive towards the service you're providing. It builds trust, and it means they can keep their own lines of communication open when they have others relying on them.



All services are online.

As of 22 April 2020, 09:50



■ 100% uptime ■ partial degradation ■ downtime

Twitter Updates



S As of 07:22 UTC, the ongoing issues for Instant Payouts are now resolved.

18 April 2020, 17:38

S We're currently investigating ongoing issues for Instant Payouts. We'll post an update soon.

18 April 2020, 16:49

S From April 14 22:43 to April 15 1:33 UTC, Instant Payouts were unavailable on the Dashboard. This issue is now resolved.

15 April 2020, 11:49

S We're currently investigating an issue with Instant Payouts being unavailable on the Dashboard. We'll post an update soon.

15 April 2020, 11:22

Updates provided via the [@stripestatus](#) twitter feed—follow us there!



Anatomy of a good status page

- A header with global status
- A set of services with individual status history
- A key to show what the colours mean
- A twitter feed (or similar) showing the latest incident responses

Service Subscriptions

When an outage is logged against a service, notifications are sent to people who have subscribed. Icons are shown to indicate your subscription to a service.

Subscribed to Updates
 Not Subscribed to Updates

Click below to Subscribe to All Services

Subscribe To All Services

Unsubscribe From All Services

Legend

- ✔ No issues
- i Planned maintenance
- Service degradation
- ! Outage
- + Multiple issues

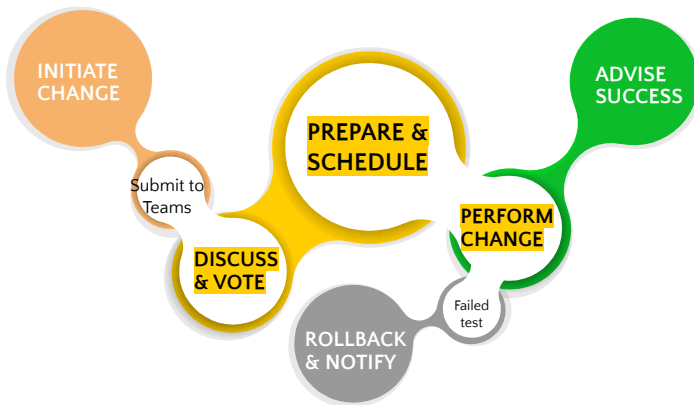
PeopleSoft Reporting	✉	✔	✔	✔	✔	✔	Operational
PeopleSoft Supply Chain Management	✉	✔	✔	✔	✔	✔	Operational
Retail	✉	✔	✔	✔	✔	✔	Operational
Retail Adding Points	✉	✔	✔	✔	✔	✔	Operational
Retail Client Lookup	✉	✔	✔	✔	✔	✔	Operational
Retail Client Registration	✉	✔	✔	✔	✔	✔	Operational
Retail POS (Point of Sale)	✉	✔	✔	✔	✔	✔	Operational
SAP Controlling	✉	✔	✔	✔	✔	✔	Operational
SAP Enterprise Services	✉	✔	✔	✔	✔	✔	Operational
SAP Financial Accounting	✉	✔	✔	✔	✔	✔	Operational
SAP Human Resources	✉	-	-	-	-	-	Degradation
SAP Labor Distribution	✉	✔	✔	✔	✔	✔	Operational
SAP Materials Management	✉	!	!	!	!	!	Outage
SAP Payroll	✉	✔	✔	✔	✔	✔	Operational
SAP Plant Maintenance	✉	✔	✔	✔	✔	✔	Operational
SAP Sales and Distribution	✉	✔	✔	✔	✔	✔	Operational
Securities Lending	✉	✔	✔	✔	✔	✔	Operational
Windows Mobile	✉	✔	✔	✔	✔	✔	Operational
Apache Web Hosting	✉	✔	✔	✔	✔	✔	Operational
Bond Trading - DR	✉	✔	✔	✔	✔	✔	Operational
Campus Network (LAN)	✉	✔	✔	✔	✔	✔	Operational
Desktop Support	✉	✔	✔	✔	✔	✔	Operational

Hosted status page services emerged because of the same reason we keep offsite backups and infrastructure redundancy - if we've got an outage, our own status page may go down, too. So there are a few commercial services out there - statuspage.io, status.io, statisfy.... And a couple of others. ServiceNow has the System Status Page, which intends to do the same thing... but you'd be doing yourself a favour by extending the basic OOB functionality with the System Status v2 Service Portal widget in the ServiceNow developer projects.

https://developer.servicenow.com/connect.do#!/share/contents/5434060_system_status_v2?v=1.01&t=PRODUCT_DETAILS



Change management



Embrace O365

Streamline your working groups and regular change board meetings by introducing online discussion, voting and automated notification & document storage steps.

Example: Using PowerApps to bring your change submissions into Teams for discussion and voting on implementation. You could use PowerAutomate to send technical documentation to SharePoint

https://resources.techcommunity.microsoft.com/case-studies/defense-contractors-learn-on-powerapps-and-flow-powering-a-change-control-board-built-by-summit-7-systems/?_sft_products=microsoft-teams

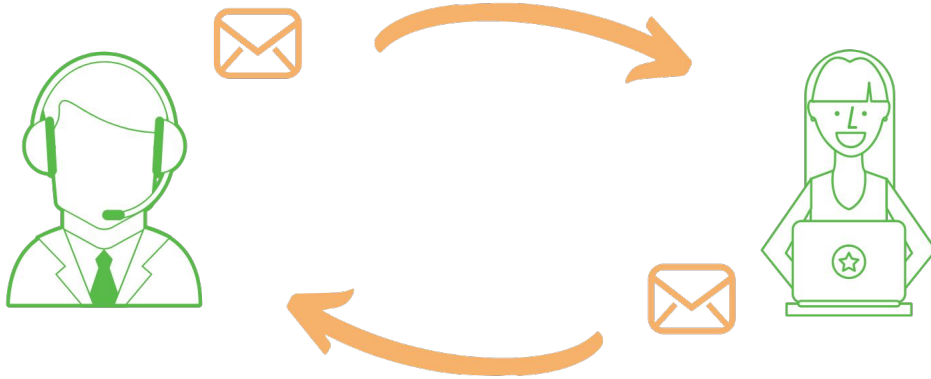


Working collaboratively

Moving from the inbox to a shared workspace



Escalation



Tier 1 > 2 > 3 Streaming... outcomes of working this way - likelihood of overloading specialists, key man risk, no exposure to problem solving process, and for the customer their issue bounces back and forth and tends to take longer because of that.



Swarming



Swarming brings people who may be able to solve the problem in together to work collaboratively with the requestor, and potentially the customer as well, to resolve the issue. ITIL4 describes swarming as a method of managing work with a variety of specialists until it becomes apparent who is best to proceed, at which point others are free to move on. It's an alternative to hierarchical escalation, and saves issues from bouncing up and down between tiers. It also means more people are exposed to the problem solving process, which expands learning at every level.

Swarms are self-organising and there's no definitive approach for how an organisation may make it work. Often it's hybrid, where there's swarming within certain teams and hierarchical escalation for other cases. Regardless, it's a touch-and-hold situation where the requestor stays with the issue through to completion and we rely on them to recognise when they need to raise their hand and trigger a swarm.

Swarms can manifest in a few different ways:

Dispatch swarms meet frequently through the day, review incoming work, select quick-to-solve items, and validates the information available is complete for any items that need help.

Backlog swarms meet as required by product or service specialists to review and resolve more complex items.

Drop-in swarms occur when specialists monitor open issues and jump in to assist.

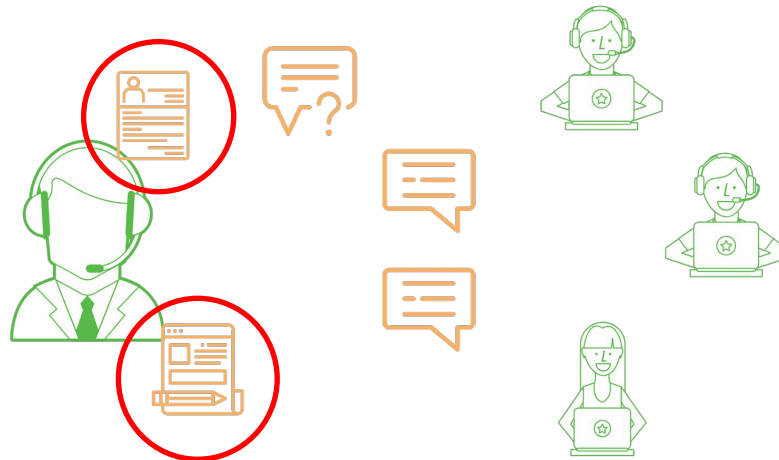
Jon Hall has written an extensive article on how swarming corresponds with the way DevOps works

<https://medium.com/@JonHall/itsm-devops-and-why-the-three-tier-structure-must-be>

[-replaced-with-swarming-91e76ba22304](#)



Intelligent SwarmingSM



Intelligent Swarming goes a few steps further. Intelligent Swarming guidance is being developed by the Consortium for Service Innovation, the same people who have researched and developed the Knowledge Centered Service methodology.

A - people profiles. Profiles contain who people are, the skills we have-both deep and broad-, AND the skills we WANT TO LEARN and be exposed to. Our profiles have a reputation model attached to them so that we have an understanding of our own history of value creation.

B - the requestor maintains ownership of the issue through to resolution and is then responsible for creating and updating the knowledge article associated with the incident.

So, for swarming to be intelligent, you need to have the ability to store and search expertise, and that may be automated from the content and classification of tickets - that's called Intelligent Matching; tickets, the knowledge base, and the work in progress need to be visible to those participating in swarming- it can't be a black box; and there needs to be this collaborative platform in which to work -- it may be Slack, Teams, or something else along those lines.



A real example

Zendesk have adopted KCS[®] & Intelligent SwarmingSM, internally. Engineers were already swarming around issues within Slack, even before KCS & Intelligent Swarming, so Zendesk leveraged that behaviour and developed an integration to create a process.



Things to note

- Intelligent Swarming must be opt-in.
- Best for organisations with a high number of complex or severe issues.
- The goal is to get known issues through self-service and new issues into a swarm.
- Don't over-engineer the process, the people profiles or the tool. Start manually and update frequently.
- Collaboration isn't always necessary. If the system is good at matching the best responder on the first touch, then multiple people won't be needed.

https://library.serviceinnovation.org/Intelligent_Swarming



Gluing it all together

Discussion: Questions and sharing ideas
about scenarios

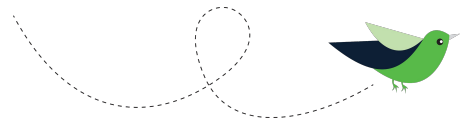




Thanks!

Any questions ?

aprill@knowledgebird.com



Ask me about running an Intelligent Swarming Insights workshop for your team or KCS adoption and training in your organisation.