alliants

DIY Knowledge Bases - Taking Ownership And Control Of Your Knowledge Without Adding More SaaS

Sebastian Burridge - SysOps / DevOps Engineer

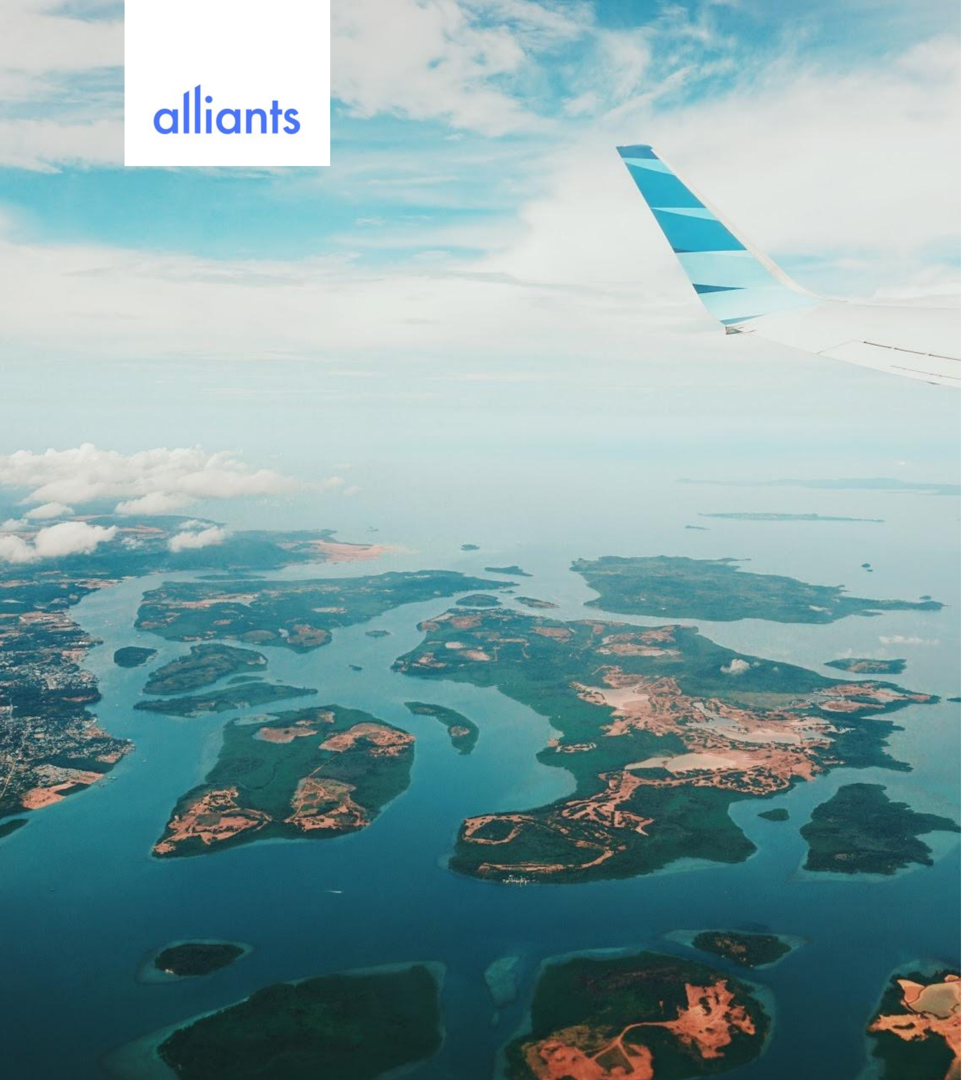
alliants

About Me

12 Years in IT Support - From IBM AS400 to AWS...

- Operations
- Servicedesk
- Datacentre NOC
- Application Support
- Agile Product Owner / Scrum Master
- Professional Services
- SysOps / DevOps

https://www.linkedin.com/in/sebastian-burridge-ab617041/



WHO ARE WE?

ALLIANTS ARE THE TRUSTED DIGITAL PARTNER OF CHOICE FOR MANY HOTEL BRANDS.

Founded in London in 2009.

Our team create, develop & deliver digital transformation strategies for guest and staff experiences.

We are now looking to build our own platform based on our experience to complement our existing offerings.

alliants

SOME OF OUR CLIENTS

FOUR SEASONS

SOHO HOUSE & CO

ACCOR

AVIS®

Jumeirah

the hoxton

GLENEAGLES

Travelodge

Extracting knowledge from developers:

- Readme.md

- Make sure Product Owner / Scrum Master allows time to complete

- Customer docs take precedent

- Don’t want a separate tool to do documentation

- “Developers speak in code while ServiceDesks speak in acronyms”

Knowledge Base Tool options:

- ServiceDesk - ITSM tool
- Developers - Confluence / Readme file
- Sales - Salesforce add-ons
- HR / Finance / Security - In-House
- Customers - Walk.me

So, do we really want to centralise all of this?

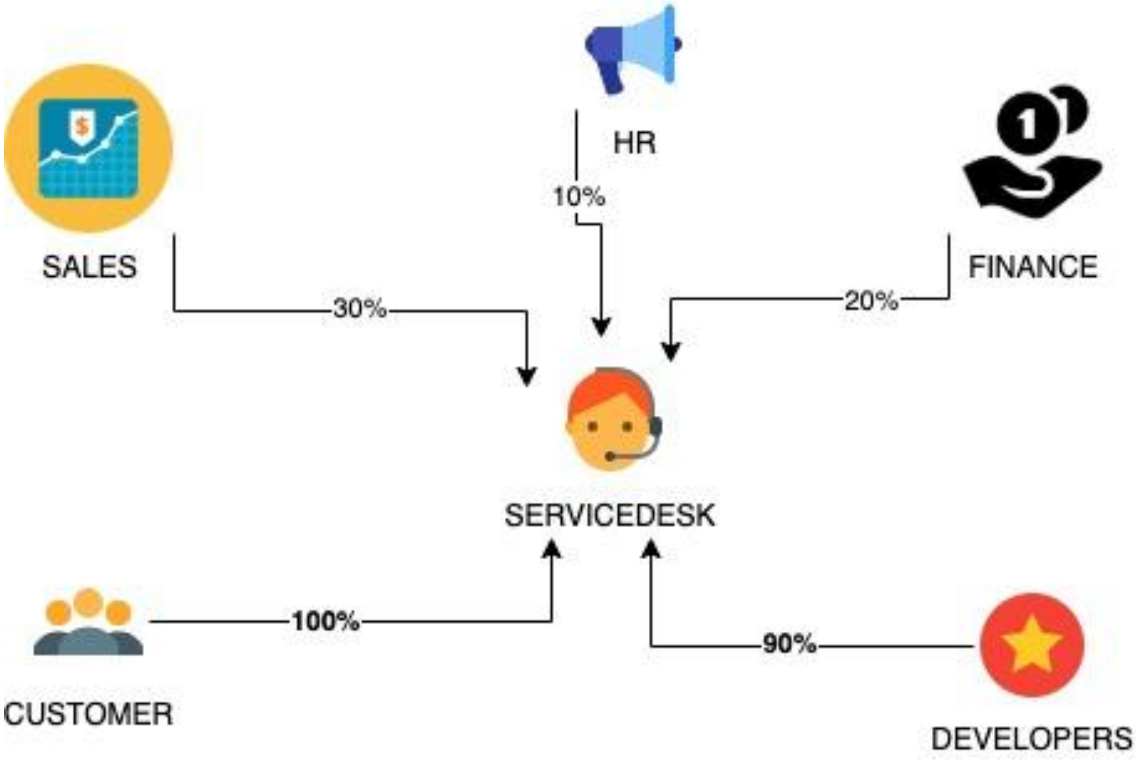
alliants

Knowledge Base Tool options:

- ServiceDesk - ITSM tool
- Developers - Confluence / Readme file
- Sales - Salesforce add-ons
- HR / Finance / Security - In-House
- Customers - Walk.me

So, do we really want to centralise all of this?

Not necessarily, but we do not want **silos** either…


SALES
HR
FINANCE
10%
30%
20%
SERVICEDESK
CUSTOMER
100%
90%
DEVELOPERS

GIT

GIT

“Git is a distributed version control system, meaning your local copy of code is a complete version control repository. These fully-functional local repositories make it is easy to work offline or remotely. You commit your work locally, and then sync your copy of the repository with the copy on the server. This paradigm differs from centralized version control where clients must synchronize code with a server before creating new versions of code.”

https://docs.microsoft.com/en-us/azure/devops/learn/git/what-is-git

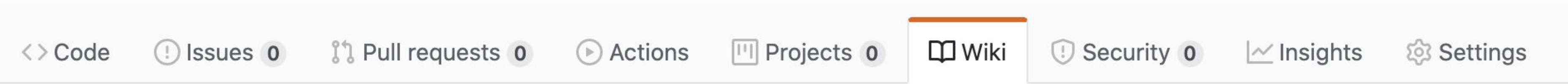
GitOps

“GitOps upholds the principle that Git is the one and only source of truth. GitOps requires the desired state of the system to be stored in version control such that anyone can view the entire audit trail of changes. All changes to the desired state are fully traceable commits associated with committer information, commit IDs and time stamps. This means that both the application and the infrastructure are now versioned artifacts and can be audited using the gold standards of software development and delivery.”

https://www.cloudbees.com/gitops/what-is-gitops

alliants

Git Wiki



https://help.github.com/en/github/building-a-strong-community/about-wikis

- Host multiple pages of content regarding the application
- Use a variety of writing methods (Markdown, AsciiDoc, MediaWiki)
- Sidebars and footers

HOWEVER:

- No tree structure
- Control is with those who can access repository

Gollum

https://github.com/gollum/gollum/wiki

The original tool within GitHub for Wikis - now diverged

- Open Source Wiki that uses Git repositories to version control.
- Controlled login / OAuth for those who can access and edit
- Two-way editing (i.e. changes in Git Repo go to Wiki Site and Vice-Versa)
- Open Source = free
- Simple, clean, easy to use
- Give Servicedesk ownership and autonomy

Gollum

https://github.com/gollum/gollum/wiki

Limitations:

- Not a knowledge base for customer facing
- Ideal for ServiceDesks with some technical knowledge
- Like some Wikis, can get messy if not maintained

alliants

LIVE DEMO

alliants

LINKS

https://projects.davidplanella.org/open-source/gollum-wiki-installation

https://grantwinney.com/hosting-a-github-wiki-remotely-on-ubuntu/

http://www.whiteboardcoder.com/2017/04/install-and-configure-gollum-on-ubuntu.html

https://www.mfoot.com/blog/2013/05/19/setting-up-a-personal-wiki-with-aws-and-gollum/

alliants

Conclusions from Alliants

- If you have development teams using Git - point them to using the Wiki features (get away from readmes)

- Service Desks that can control their knowledge Base easily and safely will engage with it far more

- New starter feedback has always been positive for documentation that they can influence, question and amend from day 1

- The closer you can bridge the Devs and ServiceDesk together, the more harmonious experience it is for customers

alliants

Thankyou :)