

# Sensible API document management guide

**Vanjikumaran Sivajothy**

Senior Manager - Product Software Architecture and Engineering  
Equinix Inc

# Agenda

01 API vs Products

02 Why is API documentation important?

03 Best practices for API content strategy

04 Building a Documentation Portal

05 Modern portal patterns in the industry

06 Time for Questions

# Impact of API Economy



# Battle between API vs Product



VS



## DIGITAL PRODUCT

# Why is API documentation important?



# Why is API documentation important?

***API Documentation is the Core Pillar of the Good Developer Experience!***



Awareness

Smooth experience in  
building apps

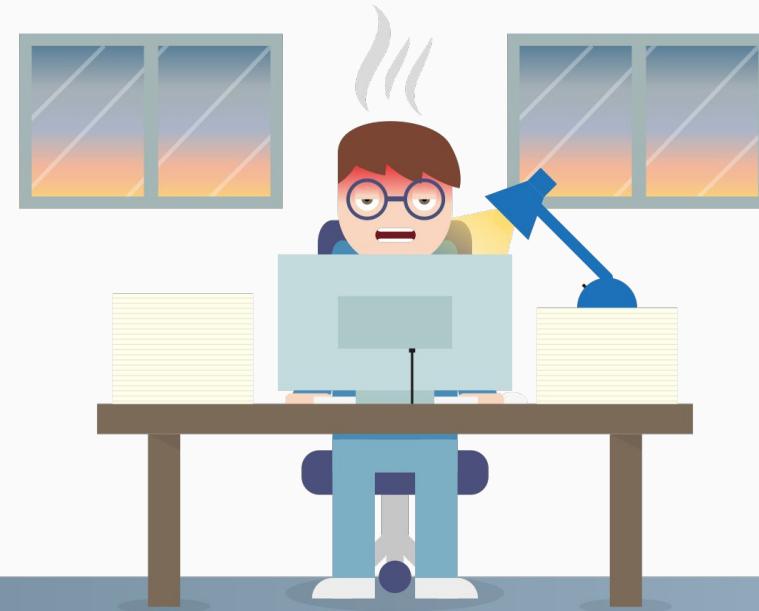
Self-sustainable  
API adoption

Easier product  
maintenance for  
internal teams

Saves time & costs of  
onboarding new  
developers

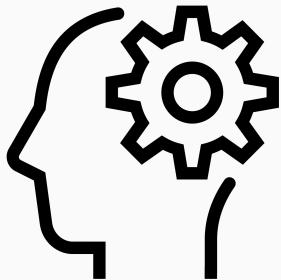


# But why so hard?



A unit of documentation. There are always three or more on a given item. One is on the shelf; someone has the others. The information you need is in the others.

— Raymond Simard —



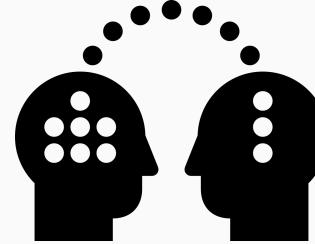
Tacit  
knowledge!

Documentation is  
everywhere!

Poor  
maintenance

## Sharing is not consistent

Different users  
business and  
technical

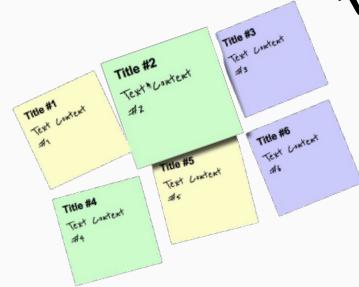


Stored in  
other places,  
separate  
PDFs

Ambiguous  
documentation

The guy who knows about  
computers is the last person you  
want to have creating  
documentation for people who  
don't understand computers.

— Adam Osborne —



Notes

# Personas



# Best Practices



# Understand their information interests



- Type of API
- Authentication
- How to accomplish what I need with your API
- Sample code for common use cases?
- Endpoint details & how-tos

Examples:

- Business use cases and examples
- Getting started guides, standards, Authentication guide
- Code samples & SDKs, API references

# Know your stakeholders / audience



- API Providers
- API Consumers/ App Developers
- API Product Managers
- Documentarians
- Developer Evangelists
- Support Team

# Determine their user journey



Figure out touchpoints with:

- Process
- People
- Information/ Artifacts

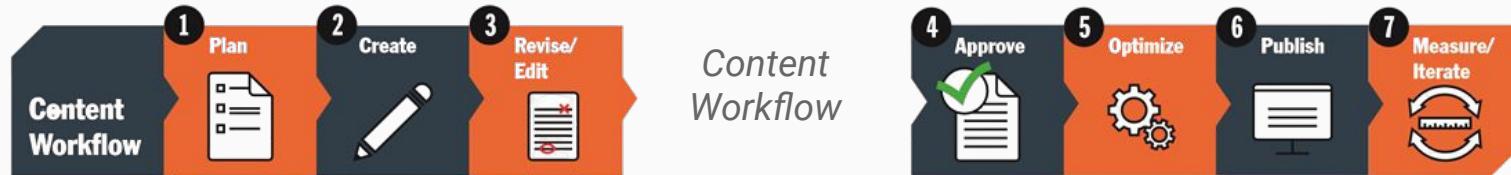
# Design the onboarding experience



Merge the user journey with:

- API Portal Technology
- Identify complexities of artifacts
- Design content workflows

# Determine the content workflow



- *What is the lifecycle of the content? (create, edit, review, approve, publish, measure, retire etc)*
- *What is the nature of your docs? Is it per API or per use case?*
- *Where is the content stored (single/ multiple/ existing sources) and format of your content?*
- *How is it maintained? Is it versioned?*

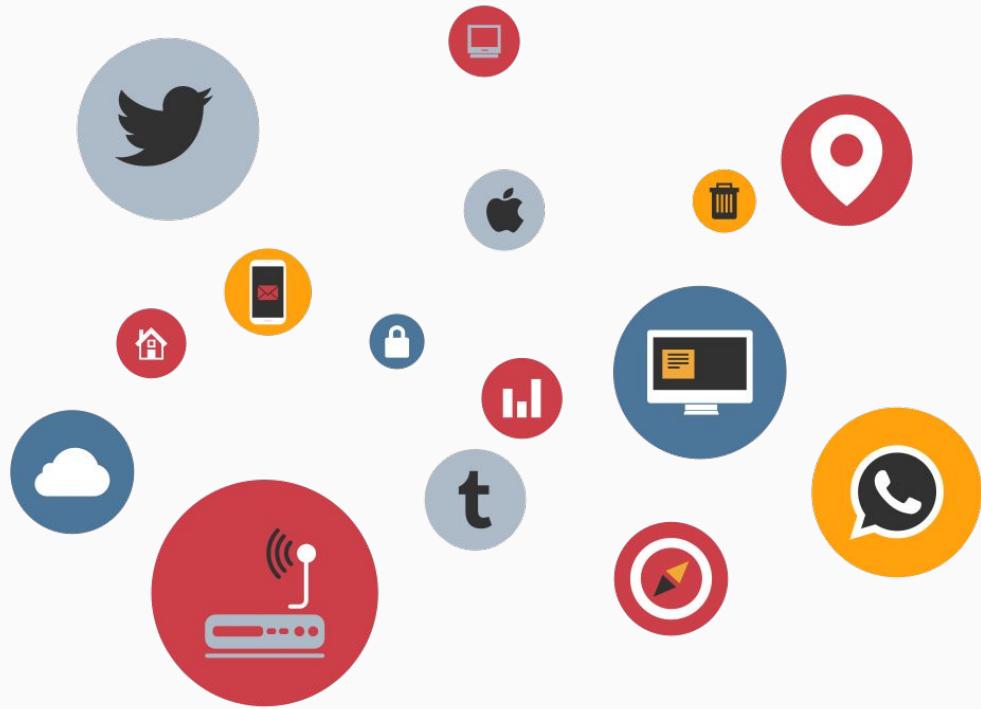
# Modern Portal Should focus on

- **Use Case Driven** : Encourages specific applications & more details
- **Getting Started Guide**: General information on getting started with a specific API
- **API Mapping**: Explaining information given out from the API
- **Tools** : Try it tool/ API sandbox to try APIs
- **Audience**: Focused on relevant target audiences
- **Simpler and clear interfaces**:: Simple catalog to understand and consume the APIs
- **Engage** : with providers & community via resources, forums/ hackathons

# Thanks!

[twitter.com/vanjikumaran](https://twitter.com/vanjikumaran)

[linkedin.com/in/vanjikumaran/](https://linkedin.com/in/vanjikumaran/)



# Question Time