

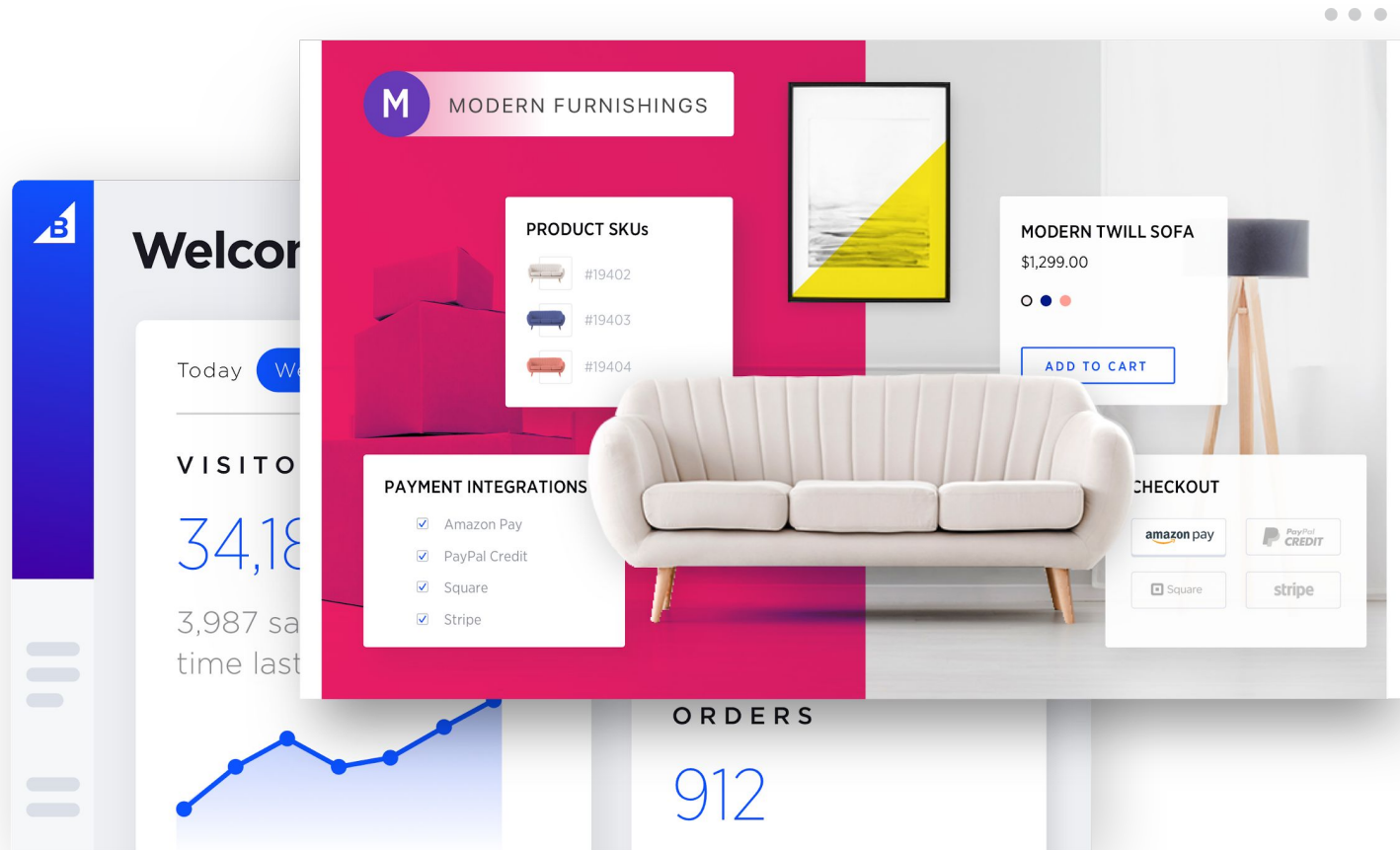
{Re}Designing a Developer Portal

That Your Developers Will Want to Use

Karen White

Developer Advocate
BigCommerce
@karen_pwhite



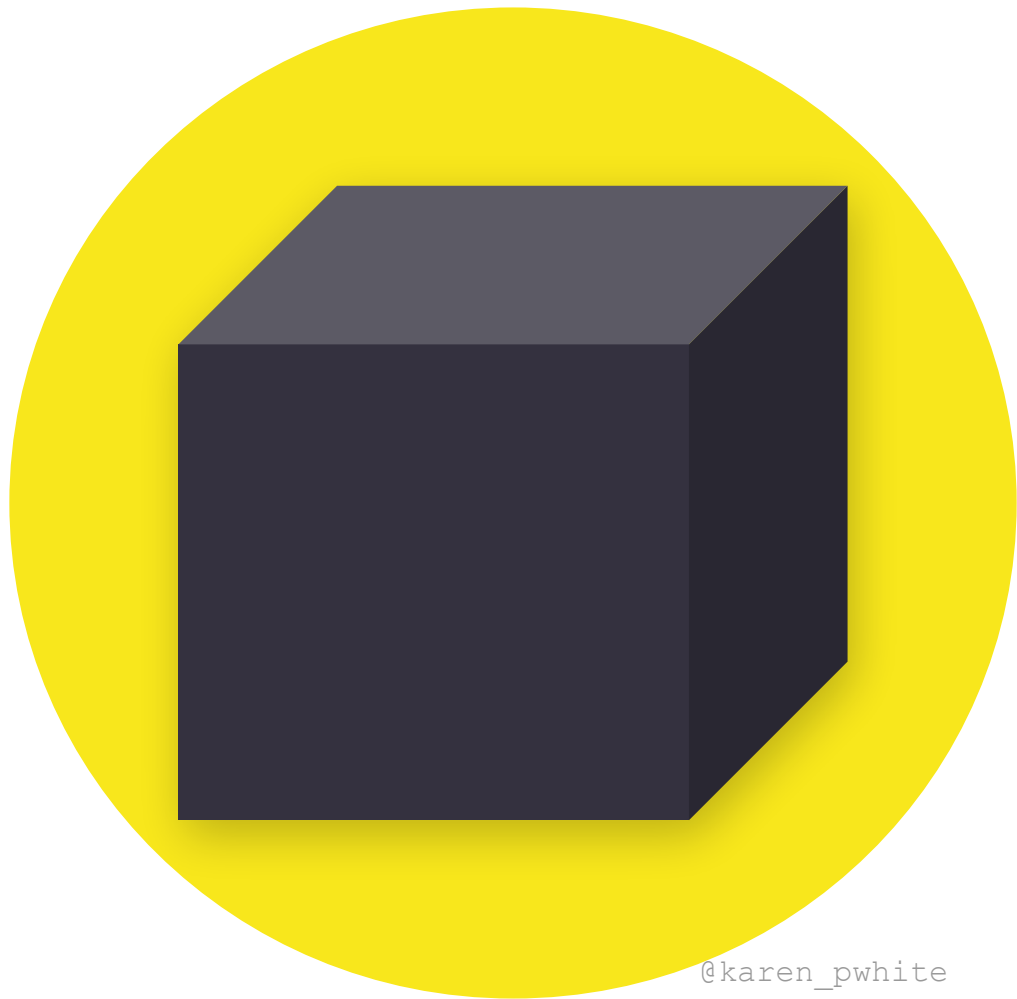


The Black Box



Opening the Black Box

- API First
- When building a new feature, it's not finished until there's a public API for the new functionality




Open SaaS



@karen_pwhite

Developer Portal Before

 DEVELOPERS

API


THEMES

CHANGELOG

SUPPORT


MY APPS


CREATE ACCOUNT




BUILD ON BIGCOMMERCE

Build apps and themes for thousands of successful merchants on our growing ecommerce platform




 See what kinds of apps you can build

Develop a custom integration, or build an app to sell in our marketplace. Our Stores API features a RESTful architecture that allows you to code in the language of your choice.


 See what kinds of themes you can design

Our Stencil theme framework is a powerful basis for creating beautiful, dynamic, and high-converting storefronts.

 Try out our API


Run in Postman

Make a few calls to our API to see how it works.
(Learn how to authorize and use this Postman collection [here](#).)


 Sign up

CREATE ACCOUNT

Get a free BigCommerce user account.
Access your store, test apps, and reach thousands of merchants.

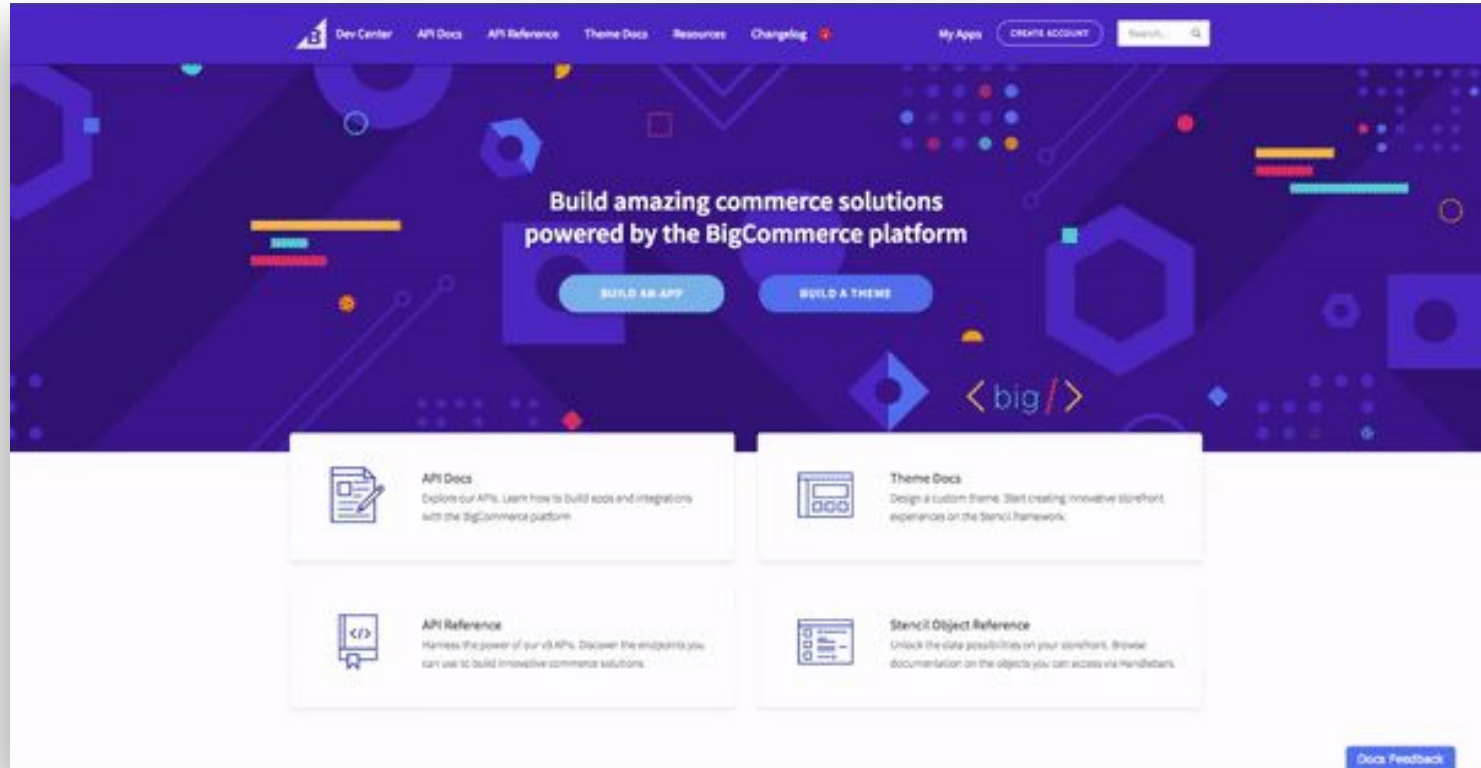
 View our API/apps documentation

Learn how to script a single store, or build apps for all our merchants.

 View our themes documentation

Learn how to build and customize responsive storefront themes.

Developer Portal After



Deciding What to Build

**How does the Developer Portal fit into
your company's overall strategy?**

AAARRRP

Framework by [Phil Leggetter](#), Nexmo



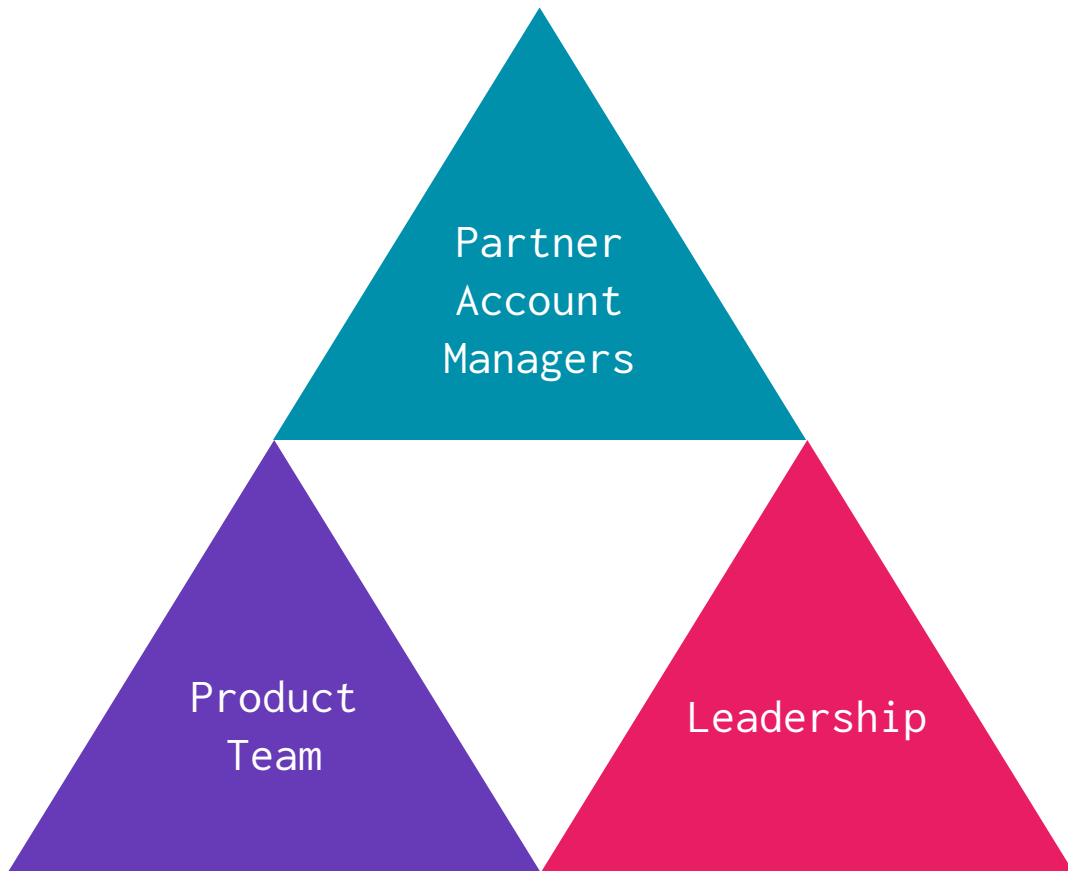
Awareness
Acquisition
Activation
Retention
Revenue
Referral
Product

You Can't Do it All Right Now



(and that's okay)

Who Are Your Stakeholders?



What Do Your Developers Want?

And more importantly... who ***are*** they?

Start with feedback interviews to hear pain points directly from developers

Creating Developer Personas

01

Gather Data

- Ask the right questions
- Survey the widest range of developers you can

02

Analyze

- Look for patterns in your data
- Identify predictive answers
- Form groupings based on similar priorities

03

Validate

- Present your findings to other internal teams who work with the developer ecosystem
- Get feedback - Do the personas ring true?

Sam

LEAD DEV



Age 33

Job Title Lead Developer

Languages PHP, React., Node, CSS

ABOUT

Sam leads a team of 4 developers at Atomic, an agency serving mostly mid-market B2B businesses with a global footprint. She specializes in building custom backend integrations, although she sometimes does frontend work if needed.

Sam double majored in Chemistry and Computer Science in college. After college, she wrote software for the oil and gas industry before moving into web development.

BEHAVIORS

Sam has contributed to open source projects in the past, but mainly to submit bug fixes. If she uses a project and sees something is broken, it makes sense to just submit a fix herself.

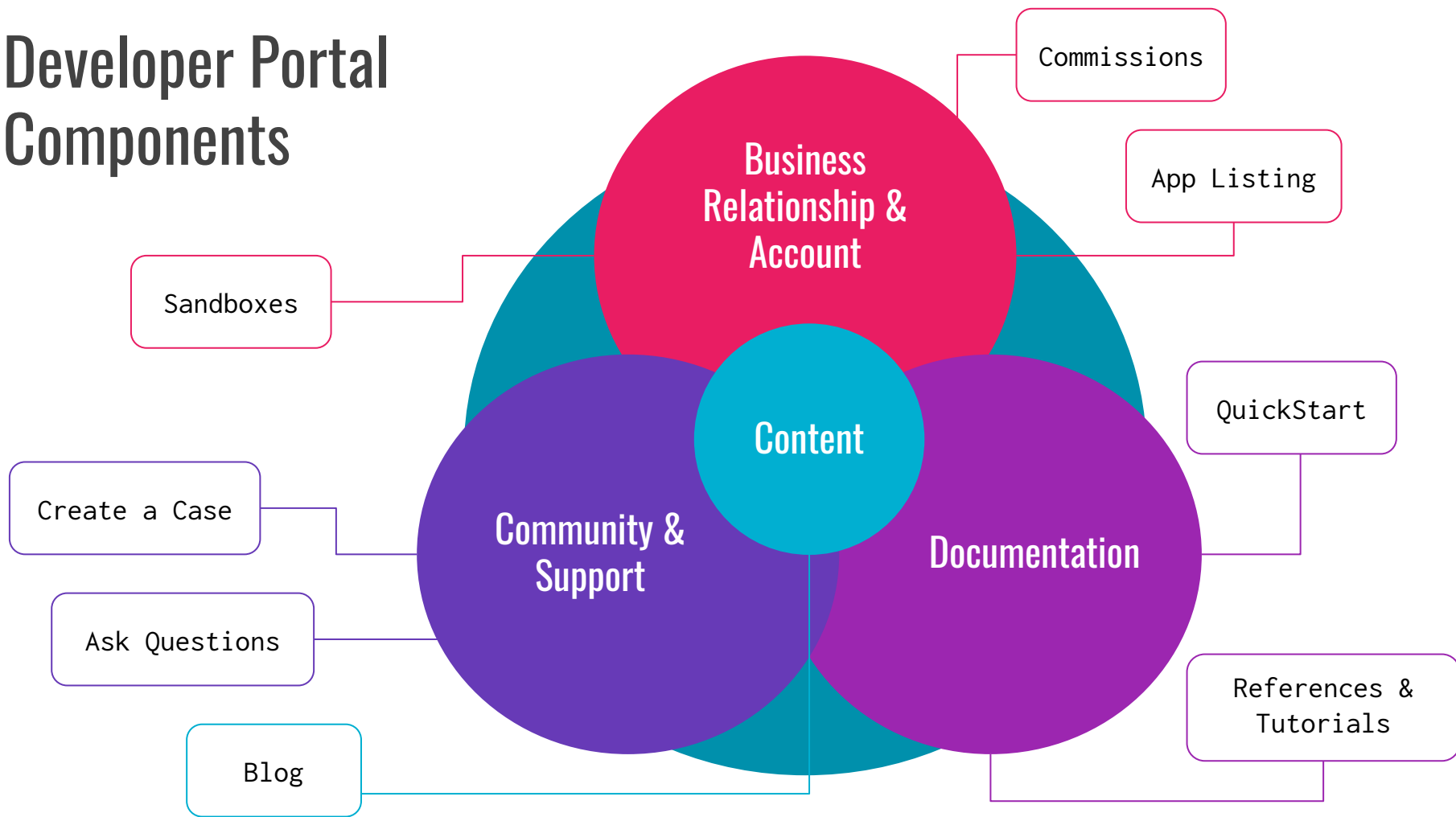
GOALS

Sam is always looking for new ways to improve her team's processes. Recently, she began adopting containerized environments and now her team is using Docker for all application development.

FRUSTRATIONS

SaaS platforms feel like a “black box” that she can't peer into. Sam prefers to solve problems by looking directly at the source code.

Developer Portal Components



What Should You Consider?

PERMISSIONS:

Different roles within an org may need access to different areas of the Portal. Decide what should be gated, and what shouldn't

PROVIDE PATHS:

Help developers make decisions about where they should go next

EXPLORATION MATERIALS:

First time visitors want to see success right away

REFERENCE MATERIALS:

Returning users want to go directly to the info that they already know they need

Developer Portal Maturity Model

Credit: Jenny Wanger + Ben Rodriguez

<http://jennywanger.com/speaking/dx-maturity-model/>

Level 1

- Closed system
- Docs may be lacking information needed to successfully make an API call
- May be unclear which APIs correspond to which features
- Some system to respond to inquiries/support requests. 1 week response time

Level 2

- Portal is self-service, but may be fragmented
- Quick-start guides, changelog and tutorials provided
- Can make an API call within 1 day
- Interactive docs
- Community portal or Stack Overflow presence
- Questions answered in 2-3 days

Level 3

- Entirely self-service
- Can make first API call within 10 minutes
- Docs in multiple languages
- Sandbox and production env provided
- Code samples and libraries
- Active community
- Questions answered within 24 hours

Level 4

- Can make first API call within minutes
- Guided walkthroughs
- Certification program
- Single, unified portal with personalized experience upon login

Developer Portal Maturity Model

Where we are today

Level 1

- Closed system
- Docs may be lacking information needed to successfully make an API call
- May be unclear which APIs correspond to which features
- Some system to respond to inquiries/support requests. 1 week response time

Level 2

- Portal is self-service, but may be fragmented
- Quick-start guides, changelog and tutorials provided
- Can make an API call within 1 day
- Interactive docs
- Community portal or Stack Overflow presence
- Questions answered in 2-3 days

Level 3

- Entirely self-service
- Can make first API call within 10 minutes
- Docs in multiple languages
- Sandbox and production env provided
- Code samples and libraries
- Active community
- Questions answered within 24 hours

Level 4

- Can make first API call within minutes
- Guided walkthroughs
- Certification program
- Single, unified portal with personalized experience upon login

Developer Portal Scorecard

Functional Unit	Level 1	Level 2	Level 3	Level 4
Openness & Accessibility	Portal Exists, manually request access	Portal is Self-service, but fragmented	Entirely Self-service	Roles & permissions
Environment & Docs	Docs insufficient to make API call	Interactive docs. Sandbox environment	Code examples and libraries available	Guided, interactive tutorials
API Quality/ Ease of Use	May be unclear what APIs can do	Regular updates. Clear error codes	Efficient. Minimal effort to integrate	REST & Storefront/GraphQL
Community & Support	Some system in place	Community portal, 2-3 day SLA	Active community. 24 hour SLA	Personalization, dev advocate program w/ KPIs

Measuring Success

Adapted from: Jenny Wanger + Ben Rodriguez
<http://jennywanger.com/speaking/dx-maturity-model/>

Functional Unit	Tactical Metrics
Openness & Accessibility	<ul style="list-style-type: none">• Account Signups• Ease of getting API keys• Time to First API call
Environment & Docs	<ul style="list-style-type: none">• Traffic• Quality rating
API Quality/ Ease of Use	<ul style="list-style-type: none">• API usage• Learning curve
Community & Support	<ul style="list-style-type: none">• Community engagement & support metrics

Changing Behavior Takes Time

Summary

Align on goals
and identify
stakeholders

Know your
users

Benchmark
progress
against
maturity model

Questions?