

The Future of Emulsify

Research Playback

September 9, 2022

Welcome

This report and corresponding playback culminate research on design systems, related tools, and open source communities.

We'll begin with an overview of the stakeholders, subject matter experts, Emulsify contributors, and users we spoke with as part of the research. Some of the statements they made are used as support. Note that we kept most of these quotes anonymous. Next is a section for alignment on language about design systems and how to think about maturity in the industry.

With a shared language, we jump into insights on Emulsify, the design systems world, and open source communities. The work concludes with a competitive analysis, the business model canvas's first draft, and the Emulsify SWOT analysis.

Our next step is to synthesize the insights from this work into needs statements which will drive ideation across multiple work streams and ultimately a revised roadmap for Emulsify.

Table of Contents

- Research
- Alignment
- Insights
- Competitive Analysis & Porters Five Forces
- Business Model Canvas
- SWOT Analysis

Project Team

A team of Four Kitchens leaders and independent consultants was pulled together to gain an understanding of how best to evolve and position Emulsify to be competitive according to user, business, and market needs. And to provide create a revised roadmap and operational understanding of the effort required to succeed.



Greg Storey
Project Lead



Colin Narver
Research Lead



Brett Harned
Research Lead



Randy Oest
Creative Director
Four Kitchens



Brian Lewis
Frontend Architect
and Senior Frontend
Engineer
Four Kitchens

Research

Primary Research

We had a series of conversations with project stakeholders, design system subject matter experts, Emulsify contributors and users.

In addition, a public survey was conducted on the week of July 25, 2022.



Randy Oest
Creative Director
Four Kitchens



Brian Lewis
Frontend Architect
and Senior Frontend
Engineer
Four Kitchens



Todd Ross Nienkerk
CEO, Owner and Co-
founder
Four Kitchens



Si Robins
Director of Business
Development
Four Kitchens



Nick Hahn
Former Director of
Design Systems
Consulting
InVision



Val Head
Co-Chair
W3C Design Token
Working Group



Ben Callahan
President
Sparkbox



Lauren Rice
Senior Visual
Designer, Carbon
Design System
IBM



Leslie Camacho
Former Product
Manager
User Arcade



Ryan Irelan
VP, Engineering
Pineworks



Caleb Ping
Enterprise Account
Exec
Knapsack



Dan Donald
Design Advocate
ZeroHeight



Franz Joseph Hartl
Webmaster
Yale University



Arek Susecki
Programista PHP
Fabrity



Kurt Trowbridge
Front-End Developer
Gravity Works



Evan Wiilite
Web Chef Emeritus
Four Kitchens



Andrii Shutov
Chief Software
Engineer
EPAM Systems

Secondary Research

Building on our own subject matter expertise, we dove into websites, reports, and books on the subject of design systems.

The latest version of Material Design is now available for Android
Learn about Material 3's new features and support for modern design & developer workflows

MATERIAL DESIGN

Material System

Introduction

Material studies

Design

Create intuitive and beautiful products with Material Design

press.stripe.com/working-in-public



Working in Public: The Making and Maintenance of Open Source Software
Nadia Eghbal

Over the last 20 years, open source software has undergone a significant shift—from providing an optimistic model for public collaboration to undergoing

Design Systems Survey 2022

Emulsify Design System

Emulsify DESIGN SYSTEM

Emulsify is an open-source tool for creating design systems with reusable components and clear guidelines for teams. Emulsify helps organizations scale their design while reducing cost, streamlining workflows, and improving accessibility.

Emulsify Design System contains multiple packages, which can be used individually to solve small problems or together to solve big ones. See below for some of the popular packages.

Starters

Starters contain application-specific configuration and files. For example, the Drupal starter contains the `.info.yml` file that defines the Drupal theme's name and other metadata as well as other Drupal-specific files.

Drupal

Emulsify Drupal is a full prototyping development environment using [Storybook](#) as a component library and [Webpack](#) as a build engine. It is also a [Drupal](#) theme. It can be used as a standalone prototyping tool or inside a Drupal installation.

WORLD DESIGN SYSTEMS WEEK 2022

September 19 – 23rd 2022

NDSWOWDSDWOWDSDW

superfriendly.com/design-systems/books/design-system-90

Design System

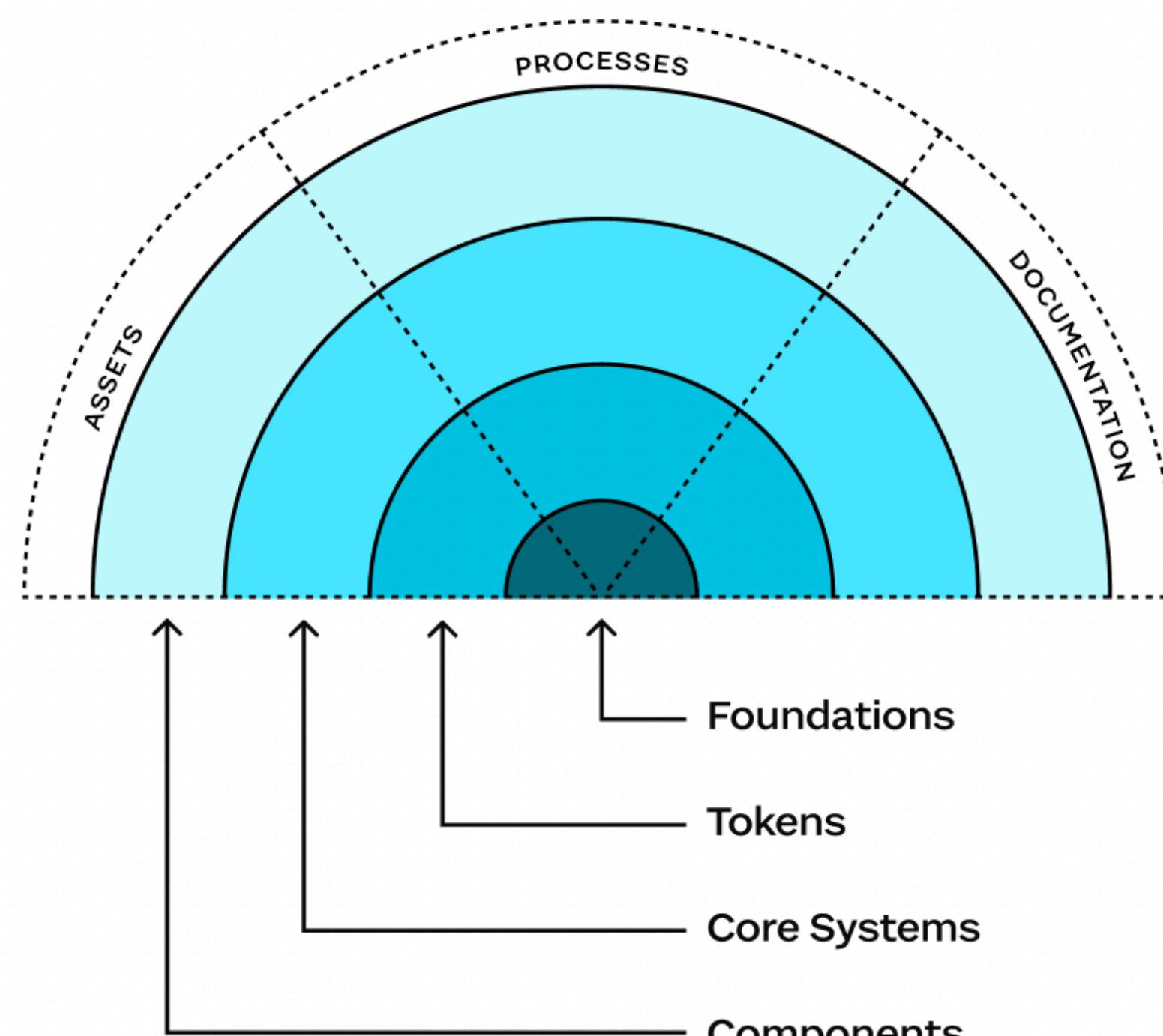
in

90

From your friends

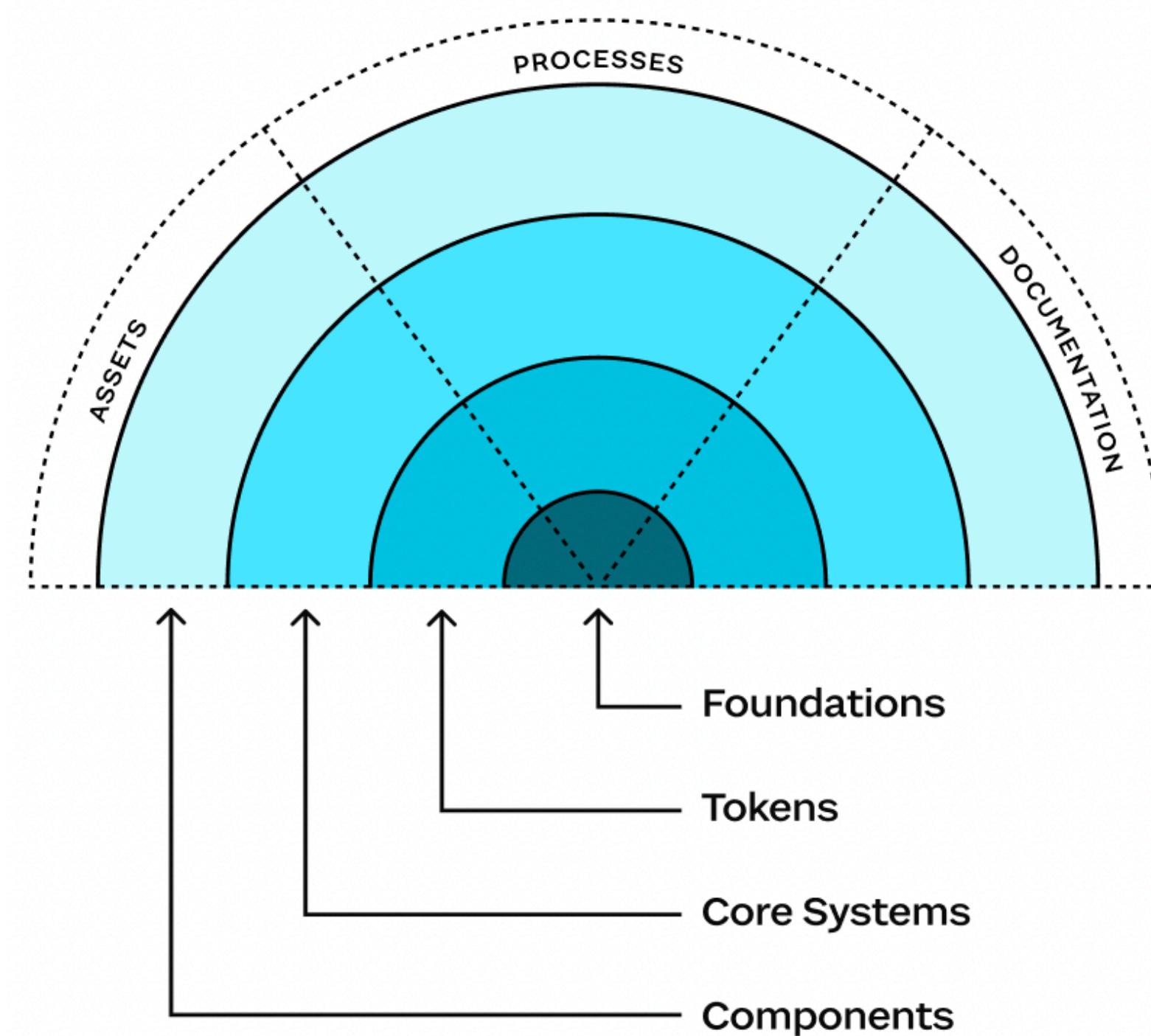
Alignment

What does design system maturity look like?



- **Foundations Layer** defines a subset of the organization's brand to be used in the design and development of digital interfaces.
- **Tokens Layer** takes the design concepts expressed in the Foundations Layer and codify them for use in constructing digital interfaces.
- **Core Systems Layer** offers a solution to common interface challenges such as layout, type scale, theming, etc. Core systems are the reusable “building-block” systems used to solve these challenges.
- **Components Layer** defines and make available the reusable parts of a digital interface.

What does design system maturity look like?



- **Assets** are the things that make each layer of the design system usable: design kits, code libraries, software tools, or even install scripts and repositories.
- **Documentation** is just what you would expect: a detailed explanation of why something is the way it is and a breakdown of when and how to use it.
- **Processes** define human behaviors as we work on and with the design system.

Design System Maturity Levels

Side Job

Bare bones effort created as a means to an end. Very informal with little to no budget. Ex: Style Guide.

Tool

The design system is a tool or a library, usually created by a small group of well-intentioned people. It's been helpful to them in their work, and they'd love to share the benefits. Ex. Style Guide with a coded Component Library.

Product

The design system becomes a product as more people recognize its importance and merits. It receives more attention, dedicated team members, funding, and a genuine attempt to make its use more widespread. Design System with tokens under internal governance. The beginnings of a variable system that affords changes in present and future state in design.

Platform

Tokens governed by cross-functional disciplines (e.g., Brand, marketing, dev, design, business, compliance, and legal).

These are four phases of design system maturity based on an amalgamation of findings from primary and secondary research. Few teams make it past the Product phase.

Design System Maturity Levels

Side Job

Bare bones effort created as a means to an end. Very informal with little to no budget. Ex: Style Guide.

Tool

The design system is a tool or a library, usually created by a small group of well-intentioned people. It's been helpful to them in their work, and they'd love to share the benefits. Ex. Style Guide with a coded Component Library.

Product

The design system becomes a product as more people recognize its importance and merits. It receives more attention, dedicated team members, funding, and a genuine attempt to make its use more widespread. Design System with tokens under internal governance. The beginnings of a variable system that affords changes in present and future state in design.

Platform

Tokens governed by cross-functional disciplines (e.g., Brand, marketing, dev, design, business, compliance, and legal).

Emulsify is here!



Emulsify is in-between the Tool and Product maturity levels based on capabilities, how contributors are using it, and how they rate the maturity of their own design system.

Insights

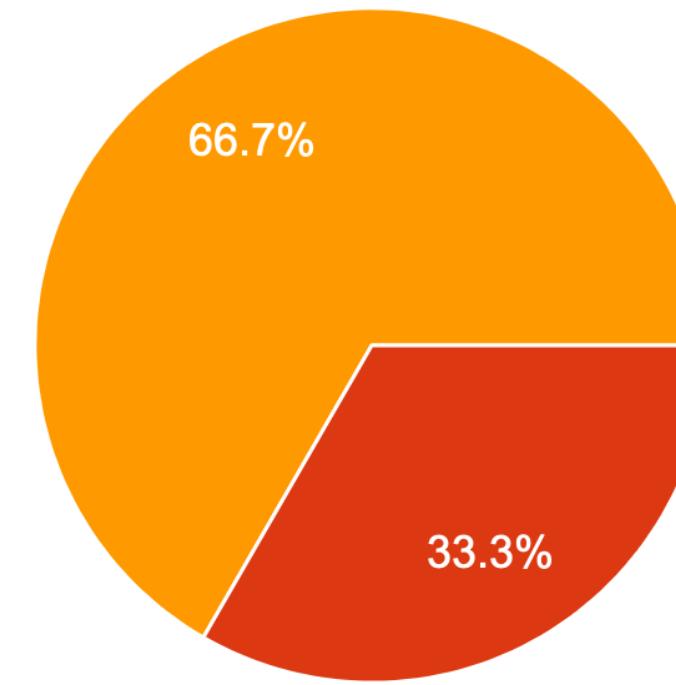
Insights: Emulsify

- There is confusion regarding Emulsify's past state as a design system, and current state as a bespoke design system builder. Today, developers see Emulsify as a Drupal Theme with Atomic components and basic tokenization.
- Current users see their design system as a tool.
- Investment in proper analytics is essential in order to assess progress towards milestones.
- Emulsify should continue development beyond Drupal, while engaging deeply with the Drupal community.

There is confusion regarding Emulsify's past state as a design system, and current state as a bespoke design system builder.

Today, developers see Emulsify is seen as a Drupal theme that supports "Atomic design."

How would you describe Emulsify to a colleague?



- It's a design system
- It's an editable Drupal theme with tokenization and Atomic components
- It's a mix of both

"I chose Emulsify because its based on Atomic Design and its very good integration with Drupal."

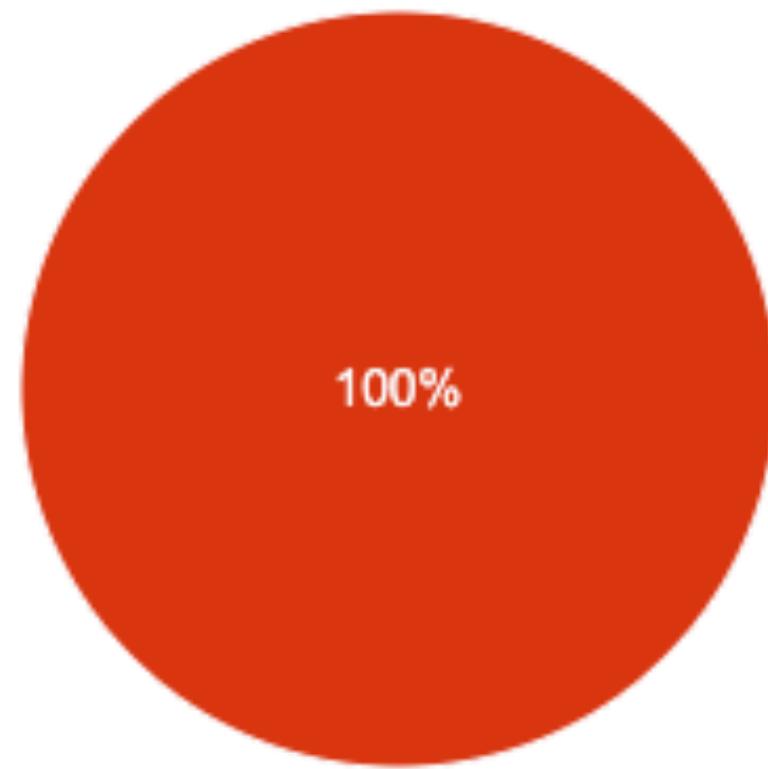
"Yes, it's a design system, but we think about it as a theme to address the needs of different departments, different voices."

"Calling [Emulsify] a design system was a function of seeing ten miles down the road."

Current users see their design system as a tool.

Very few Emulsify primary users have reached the product maturity level. Features that address “tool” problems, needs, and opportunities should be prioritized over greater ambitions until the user base and their design systems mature.

How would you rate the maturity level of your current design system?



- The design system is like a side job.
- The design system is a tool or a library. **100%**
- The design system is a "product" with a dedicated team.
- The design system is a practice that is co-owned by people outside of product, engineering, and design.

Investment in proper analytics is essential in order to assess progress towards milestones.

At the time of writing this report, there is incomplete data on distribution and downloads—No data through Github or CLI. This information is vital to define and track success for community growth and progression for users, contributors, and maintainers.

Emulsify should continue Wordpress development, but prioritize the Drupal version.

“Emulsify is “big” a differentiator in the Drupal space.”

“Wordpress is the big one. That’s next! I don’t want to, but I’m going to have some person come in and say, I like Wordpress.”

People want to contribute to Emulsify, but are confused as to what and how. Pull requests and contribution approvals take too long.

The community is ready and willing!

“Right now I don’t see a roadmap for Emulsify.”

“How do you get designers to contribute to Emulsify with as little burden on the backend as possible. Like the work of Emulsify in Figma. And what does that work flow look like for a designer.”

“Last pull request was one month ago.”

“It took a couple of weeks for the pull request to be improved, current request is at one month.”

“What is Compound?”

“I need to know the rules to contribute.”

Emulsify is a great way to introduce code cohesion, development principals, and process alignment between developers.

“Before Emulsify, FED work was random.”

This side-benefit should be considered for future marketing use.

FEDs want to use Emulsify as a way to improve their work with designers and get designers to be component driven (where it makes sense).

FEDs seek to bring design into parity with code through Emulsify. They also see the tool as a way to educate design on systems thinking and exposing the “bits and pieces” of design that often fall through the cracks, leaving it up to FEDs to make guesses on how to finish the job.

“The thing we miss on our end in general is that [Emulsify] is development centric, but it doesn’t speak to design, and design does not map to development.”

“How do you get designers to contribute to Emulsify with as little burden on the backend as possible? Like the work of Emulsify in Figma. And what does that work flow look like for a designer?”

Insights: Design Systems

- Design systems are not universally considered the “Source of Truth.”
- Design systems often fail when they try to scale beyond a tool.
- The need to support compliance is important and will grow beyond accessibility.
- Carbon’s success was driven by global IBM’s enforced brand compliance.
- Design systems are the best way to scale brand and regulatory compliance across digital experiences.
- Token sophistication and their use are in early infancy but their utility and importance will become much greater in the future.
- As design systems mature to a platform state, the focus shifts away from design and development. Few companies have achieved a level of design system maturity where practice and core systems are meaningfully enabled.

Their is a lack of a universal definition for design systems and no current W3C standard in play.

There is an opportunity to add more perspective to this initiative.



Design Systems

A connected, package-managed, version-controlled, software product that contains the smallest set of components and guidelines an organization needs to make digital products consistently, efficiently, and happily.

Many people have different and useful definitions for design systems:

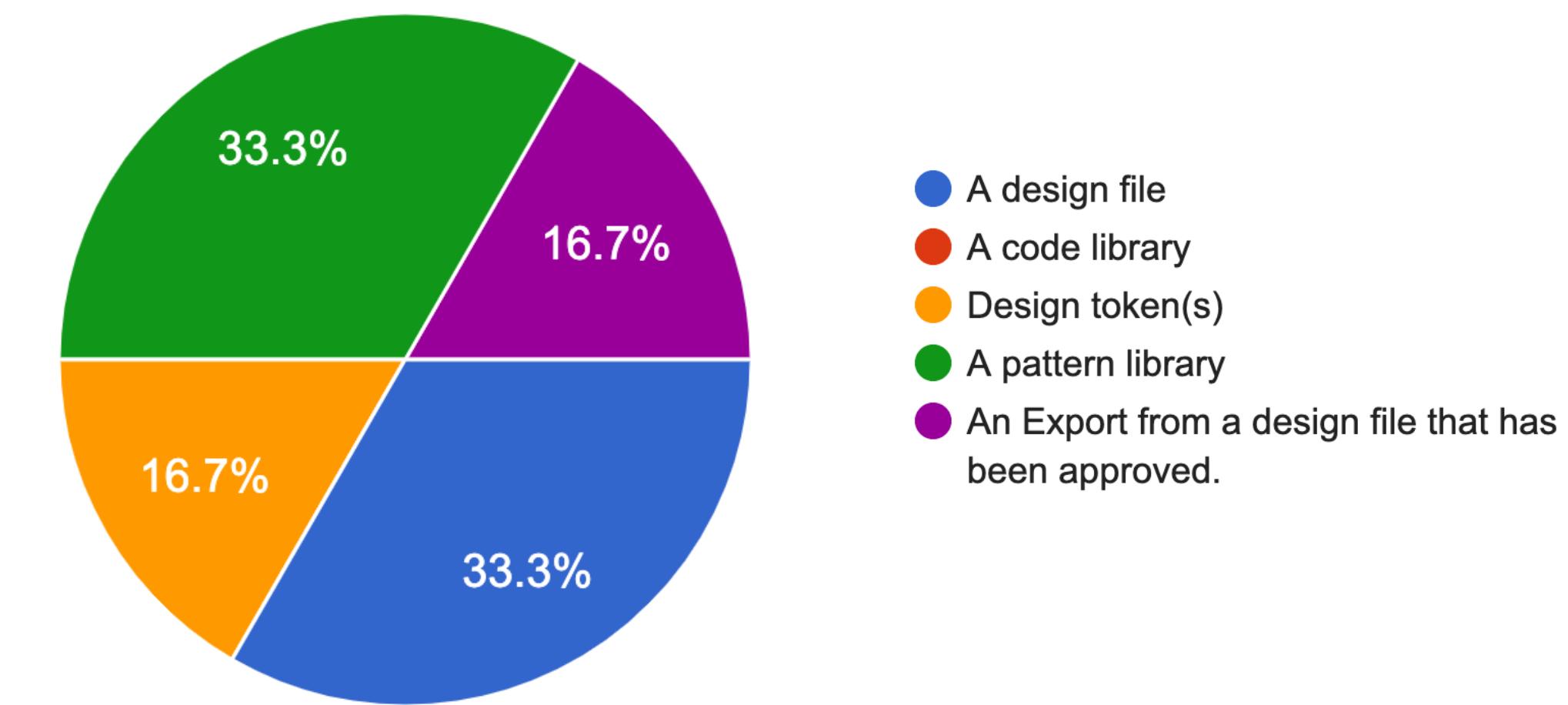
- Designer Alla Kholmatova calls design systems “a set of connected patterns and shared practices, coherently organized to serve the purposes of a digital product.”
- Designer/developer Sylvee L. says design systems are “collections of rules, constraints, and principles, implemented in design and code.”
- Designer Nathan Curtis says, “A design system offers a library of visual style, components, and other concerns documented and released by an individual, team or community as code and design tools so that adopting products can be more efficient and cohesive,” among other things.
- Developer Jeremy Keith says, “I think we lose sight of the real value of a design system when we focus too much on the components. The components are the trees. The design system is the forest.”
- Front-end designer Brad Frost says, “A design system [is] the official story of how your organization designs and builds digital interfaces.”

“There is an epidemic of syntax around Design Systems.”

Emulsify users do not have a shared definition for the “Source of Truth.”

This is a great opening opportunity for the new product leader to engage with the Emulsify community and evangelize best practices.

What do you consider to be the single source of truth?



Design systems often fail when they try to scale beyond a tool.

Multiple SMEs we interviewed for this project echoed problems we have seen across design programs at scale in the last three years. The pattern is consistent across hundreds of design programs at scale.

Legacy systems are not accounted for

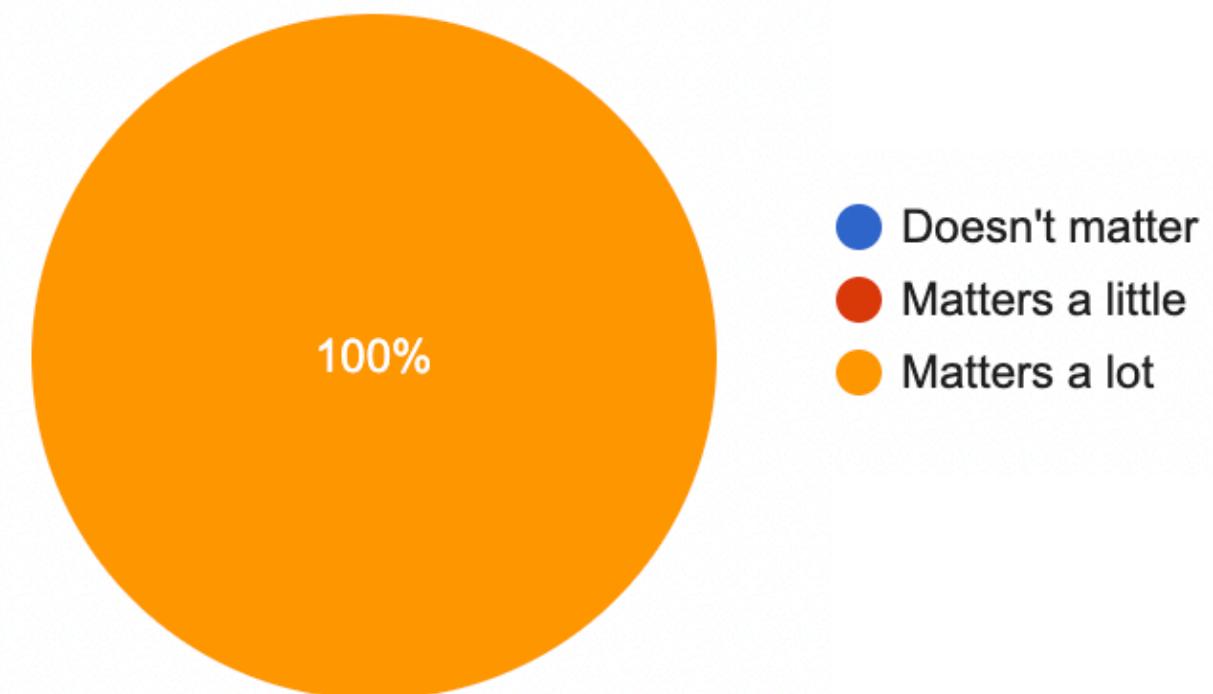
Cross-functional partners outside of design and development are not consulted or brought in to collaborate during the work

The support system required to scale to a product is beyond the core team's capabilities and expertise

The need to support compliance is important and will evolve beyond accessibility.

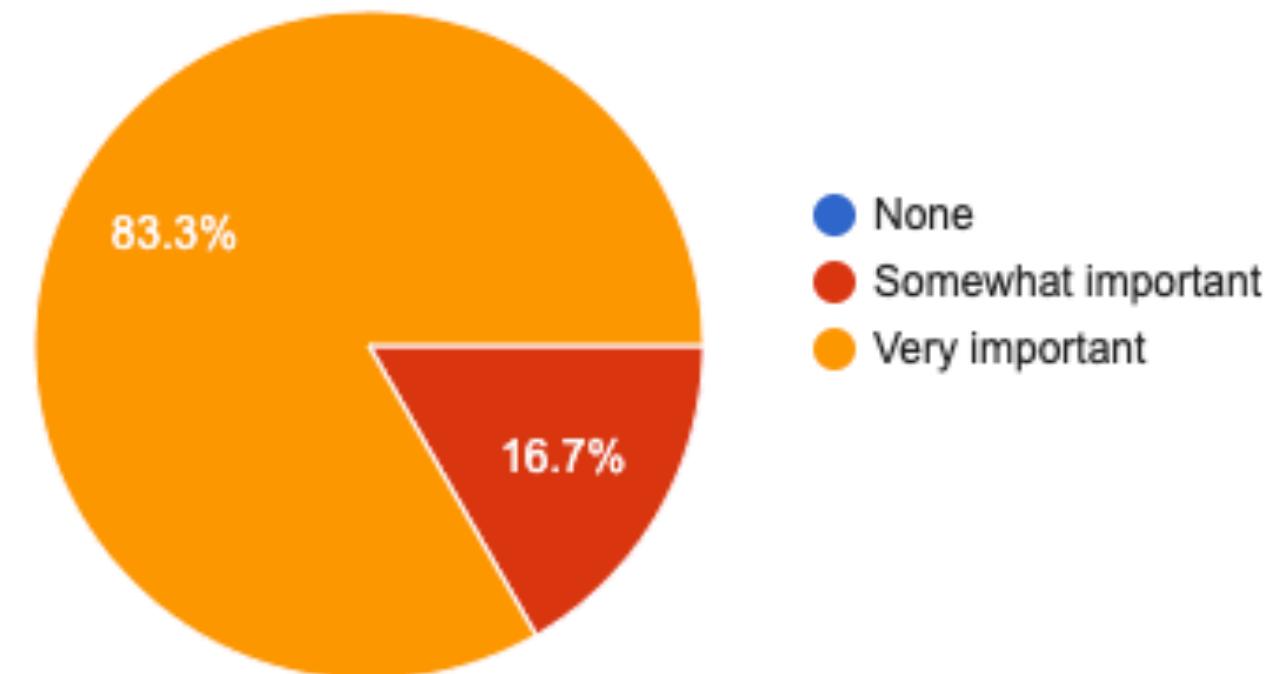
SMEs and contributors have a shared perspective on the needs and desires to use design systems as the best way to achieve brand cohesion and regulatory compliance across digital experiences.

How much does accessibility matter as part of your design system?



"The brand is everything! Accessibility is everything!"

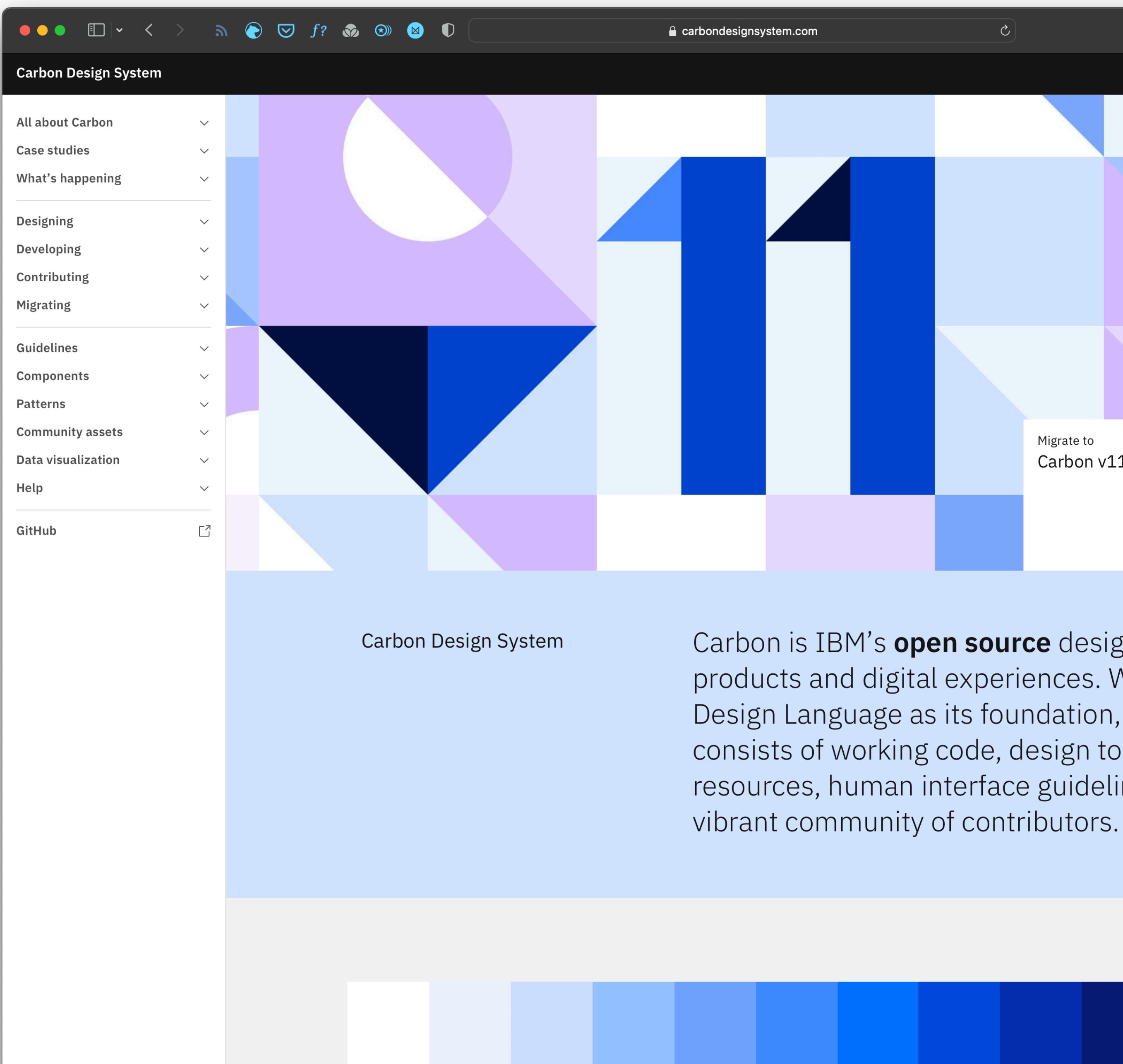
How important is brand cohesion to your stakeholders?



"I am responsible for providing a [branded] accessible experience on every view port imaginable."

After 10 versions, IBM Carbon only gained wide spread adoption due to new brand compliance.

Efficiency between design and code across hundreds of product teams was not enough to drive adoption for five years.



Carbon Design System

Carbon is IBM's **open source** design products and digital experiences. With Design Language as its foundation, consists of working code, design tools, resources, human interface guidelines, and a vibrant community of contributors.

IBM Carbon Timeline

Carbon v1.0

2016

IBM BlueMix creates the first “design system”

Carbon v11.0

2021

Carbon finally gains platform status

2016-2018

14 separate design systems created within IBM business verticals

2019

IBM launches revised global brand and compliance requirements

Carbon began as a design library for IBM Bluemix. Product teams forked this system to produce their own with branded elements. Within two years, there were 14 separate design systems within IBM. A new brand language with bespoke typefaces and grid systems drove higher adoption of Carbon, as it was the fastest way for teams to achieve corporate compliance.

Design systems are the best way to scale brand concession and regulatory compliance across digital experiences.

There is an opportunity to develop thought leadership, language, evangelism, education, and value proposition around this universal need at scale.

“If I’m looking for centralized funding, then the story is about compliance: brand and accessibility. To the end user, I’m saying you are empowered while staying compliant.”

“We don’t talk about the brand police, we talk about ease-of-authoring experience. Enabling of rapid prototyping. Try to avoid restrictive language. Provide the illusion of choice.”

“You’re already compliant if you use the design system.”

“One of the promises of Emulsify: To provide brand consistency across an entire organization.”

Token sophistication and adoption are in early infancy, but their utility and importance will become much greater in the future.

Token standards will drive adoption while providing opportunities for innovation and automation. Companies with platforms stand to benefit the most from future token development.

Immature

- Tokens are used as part of a delivery system.
- Tokens are used to store CSS variables.

Mature

- Multiple teams leveraging tokens across multiple design systems.
- Tokens represent cross-functional disciplines: Brand, marketing, legal, compliance, business, etc.
- Tokens are less about code delivery than decision velocity.

“What tokens are and what they do is not universally understood.”

— Val Head

— Maturity definitions taken from an interview with Leslie Camacho

As design systems mature to a platform state, the focus shifts away from design and development.

Once design systems reach the product-to-platform level, the majority of work is in processes. Few companies have achieved a level of design system maturity where practices and core systems are meaningfully enabled.

“The more mature your design system is, the more it is representative of multiple disciplines (brand, marketing, compliance, legal). It becomes less about code delivery than decision velocity.”

“Two-thirds of the work in design systems is not building design systems but governing them.”

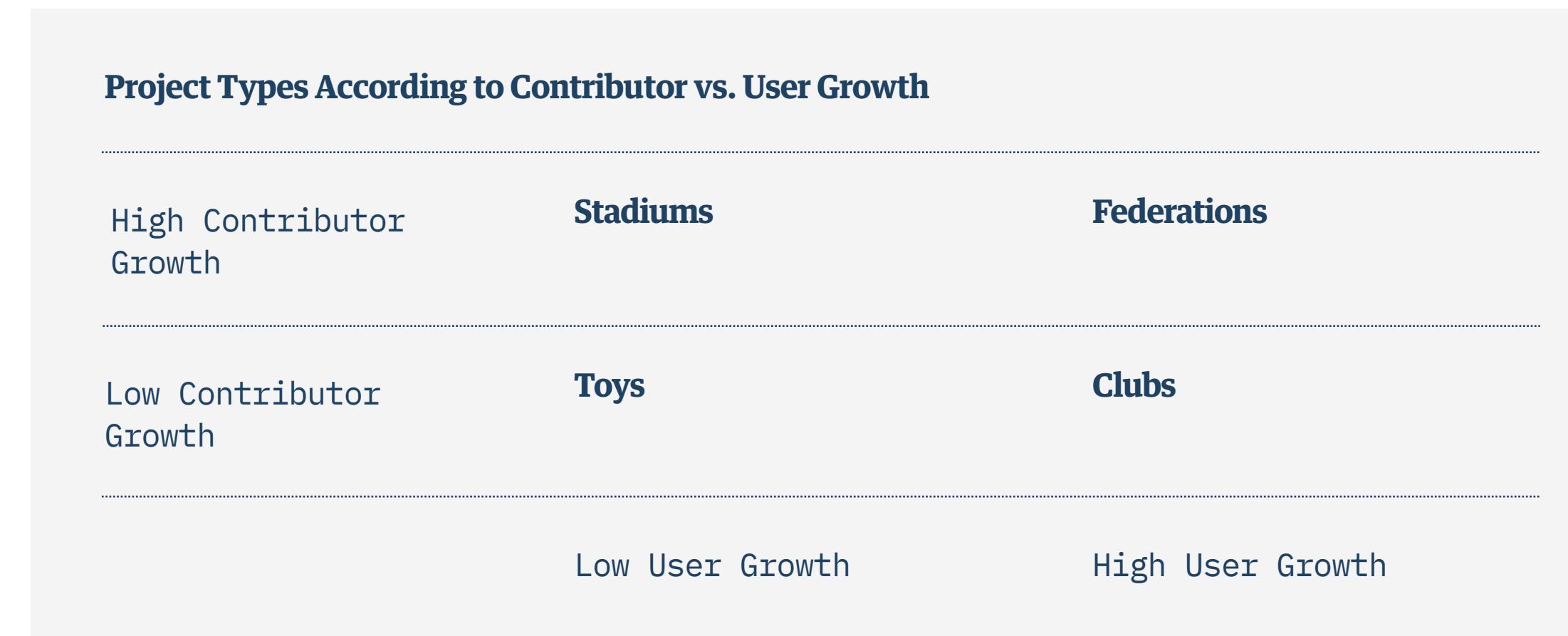
Mapping Emulsify as a Project Type

Toys are projects with low contributor growth and low user growth.

Stadiums are projects with low contributor growth and high user growth.

Clubs are projects with high contributor growth and low user growth, leading to a roughly overlapping group of contributors and users.

Federations are projects with high contributor growth and high user growth. Federations are similar to companies or NGOs. They're more complex to manage from a governance standpoint, so they tend to develop processes—voting, leadership positions, foundations, working groups, and technical councils—that address coordination issues within their contributor community. These contributors, in turn, make decisions for a broader constituency of passive users.



Successful clubs are highly sticky, retaining a large portion of their contributor base, even if they don't attract as many new ones. So long as those contributors stick around, and clubs add just enough new contributors to stay active, these projects will continue to exist as their own, self-sustaining commons.

— Eghbal, Nadia. *Working in Public: The Making and Maintenance of Open Source Software*

Using Nadia Eghbal's definitions of "project" types (the entire bundle of community, code, and communication and developer tools that support its underlying production¹), Emulsify is a "club" with more than 200k downloads.

1. Eghbal, Nadia. *Working in Public: The Making and Maintenance of Open Source Software* (p. 54). Stripe Press.

How FOSS projects change over time

Creation

The earliest stages of a project, there are probably one or a few developers who are writing code in a fairly closed state of development.

Evangelism

At this stage, the goal is distribution—getting new code in the hands of other developers—so the project transitions to a more open state of development.

Growth

One heuristic for when this transition occurs is when maintainers start doing more non-code than code work on the project, such as triaging issues and reviewing others' pull requests.

“Growth” means that more developers might be using the project, but that doesn’t necessarily mean that more developers are making meaningful contributions, or that there are more maintainers meeting this level of demand (as one might expect a successful company to grow employee head count).

A project’s contributor growth is a function of its technical scope, support required, ease of participation, and user adoption.

– Eghbal, Nadia. *Working in Public: The Making and Maintenance of Open Source Software*

Emulsify is in-between the Creation and Evangelism state.

Competitive Analysis & Porters Five Forces

Competitors

- ✓ Capability
- ✓ Nascent Capability

	Basic Tokens	Components	Core Systems	Assets	Documentation	Process
Emulsify	✓	✓		✓	✓	
Knapsack	✓	✓	✓	✓	✓	
Fractal		✓		✓		
Arcade	✓			✓		
Blacklight	✓	✓		✓	✓	
Supernova	✓	✓	✓	✓	✓	
Zero Height	✓	✓		✓	✓	
Invision DSM	✓	✓		✓	✓	

None of the competitors provided represent a significant market share. It is still early days in this product space as most offerings have only been in existence for two-plus years. Tools that were early to market (e.g., InVision DSM) are already falling out of favor. There is room to innovate in the core systems and token sophistication. It's unclear how tools in this space can have an impact on "Processes" because so few teams have reached that level of maturity.

Porter's Five Forces

Competitive Rivalry - *The strength of competition in the industry.*

- Early days, with lots of products and services launching to support design system management (e.g., Luro). The floodgates are open.
- Even established players (InVision) lack dominance. Employees left to create Knapsack.

Suppliers Power - *Examine the ability of your suppliers to increase their prices.*

- Ratio of buyers to suppliers is high (lots of buyers/consumers of design system tool).

Buyer Power - *The strength of your customers to drive down your prices.*

- Customers have access to lots of alternatives, cost is a factor and changes given scale (small team vs. Enterprise pricing model).
- The more that Emulsify is used, the burden for supporting and improving falls to Four Kitchens unless grows contributors.

Porter's Five Forces

Threats of Substitute Products and Services - *The extent to which different products and services can be used in place of your own.*

- High number of substitute products available at root tech stack level (Drupal, Wordpress etc) and top of stack
- Switching costs are low
- Design system management tools like Knapsack leverage pre-existing production tech (Figma, Github, NPM) which makes it easy to adopt
- Biggest risk is if Wordpress or Drupal bring design system functionality “in-house” natively with tokens etc.

Threats of New Entrants - Evaluate how difficult it is for new competitors to enter the market.

- No unified definition (no set standard) of a “design system” means it is more difficult to respond to tactical threats (moving targets/definitions)
- Easy for SaaS competitors to access market
- If you’re going to be competitive in the design systems marketplace, you need to support Figma and Github
- Loyalty for tools in the digital space is short lived and tied to parent organization’s speed to respond to new threats (case studies are InVision and Adobe)
- Adobe stands to gain the most from token standardization and maturity

SWOT Analysis

Emulsify SWOT Analysis: Strengths

- Able to flexibly support dynamic content distribution needs at scale.
- Sweet spot for higher education, government, and media organizations with integration to Drupal.
- Open source solution built for the Drupal community by the Drupal community.
- Emulsify is proven, dependable tool currently deployed at all levels of scale.

Emulsify SWOT Analysis: Weaknesses

- Identity Crisis: Developers are unsure what Emulsify's grand intention is - a design system or a tokenized theme or both.
- Uncertain if decoupling design components from starter makes sense to anyone outside of Four Kitchens.
- Lack of dedicated evangelism.
- Contributors have to wait too long for pull requests and submissions.
- Incomplete data on distribution and downloads, No data on downloads through Github or CLI.
- Emulsify needs better representation on the Drupal website, a primary marketing tool.
- The Emulsify website needs improvements to reach parity with other design system management and template/theme products.

“I stumbled upon Emulsify through the Drupal community documentation.”

Emulsify SWOT Analysis: Opportunities

- Design systems and tokens are still in infancy in their shared definition and successful use at scale.
- Grow thought leadership through teaching and advocating best practices.
- Increasing clarity of what and how people can contribute to Emulsify.
- Evangelism on design systems and Emulsify at-large.
- Bring designers and their work into the product.
- Find ways to bring designers into the project as contributors.
- Create a multi-tenant tool that works across teams and tech stacks.

“How do you get designers involved without creating burden on the backend?”

Emulsify SWOT Analysis: Threats

- Too much confusion on what Emulsify is and what it does (design system tool vs. design system).
- Current users are dominantly FEDs who only see their design system as a tool.
- As most Emulsify users see the product as a Drupal theme, there are a lot of competing “design systems” for Drupal.
- Investment costs in Emulsify are low, switching tools is too easy.

The Future of Emulsify

Research Playback

###

FOSS Operations

Assessing the Health of Emulsify

Casual contributors are often aware that they have little context for what is going on behind the scenes of the project. But more importantly, they do not want to spend time familiarizing themselves with a project's goals, roadmap, and contribution process. These developers primarily see themselves as users of the project; they do not think of themselves as part of a "contributor community." The role of a maintainer is evolving. Rather than coordinating with a group of developers, these maintainers are defined by the need for curation: sifting through the noise of interactions, such as user questions, bug reports, and feature requests, which compete for their attention.

it's not the excessive consumption of code but the excessive participation from users vying for a maintainer's attention that has made the work untenable for maintainers today.

But over the last twenty years, open source inexplicably skewed from a collaborative to a solo endeavor.

Eghbal, Nadia. *Working in Public: The Making and Maintenance of Open Source Software* (p. 15). Stripe Press. Kindle Edition.

today's developers hardly even notice "open source" as a concept anymore. They just want to write and publish their code,

Eghbal, Nadia. *Working in Public: The Making and Maintenance of Open Source Software* (p. 28). Stripe Press. Kindle Edition.

Assessing the Health of Emulsify

This speaks to the need for community guidelines and, perhaps, a code of conduct.

Sarah Drasner, a maintainer of Vue, another popular frontend framework, cowrote an “open source etiquette guide” with Kent C. Dodds, in which they emphasize that contributors should “be polite, respectful, and kind.”

Eghbal, Nadia. Working in Public: The Making and Maintenance of Open Source Software (p. 38). Stripe Press. Kindle Edition.

Because GitHub is so easy to use, the hurdle to opening an issue or asking questions is low, and users ask maintainers for help with everything: how to resolve merge conflicts, how to write a test. These skills aren’t specific to any one open source project; they’re general “how do I open source” questions.

The need for onboarding documents and education.

Open source maintainers have become the de facto teachers for developers who are learning how to contribute.

Eghbal, Nadia. Working in Public: The Making and Maintenance of Open Source Software (p. 40). Stripe Press. Kindle Edition.

Assessing the Health of Emulsify

Does Emulsify do this?

Open source projects are called “projects,” rather than just code. While code is the final output of a project, the term “project” refers to the entire bundle of community, code, and communication and developer tools that support its underlying production.

A framework for contributing directly

Open source projects hosted on GitHub can be broken into three parts: code (the final output of a project), an issue tracker (a way to discuss changes), and pull requests (a way to make changes).

Issues tend to fall into three categories: bug reports, feature requests, and questions.

What, how, and where to communicate

Projects often utilize both synchronous communication channels (like IRC, Slack, or Discord) and asynchronous ones (like mailing lists, Reddit, or GitHub issues) in order to talk to users and other contributors.

These channels are used for all sorts of conversations, including: Asking and answering questions among both users and contributors (“How do I use the project to do X?” as well as “How do I make Y contribution?”) Coordinating work among maintainers Providing a community space Holding office hours Educating users (e.g., teaching or demonstrating something)

Eghbal, Nadia. Working in Public: The Making and Maintenance of Open Source Software (p. 52). Stripe Press. Kindle Edition.

Classifying Project Types

Technical scope refers to the size and complexity of a project's codebase: in other words, how much more there is to do. A project that feels feature-complete won't attract as many contributors as one that's extensible and customizable.

Eghbal, Nadia. *Working in Public: The Making and Maintenance of Open Source Software* (p. 56). Stripe Press. Kindle Edition.

Support required refers not just to writing code for the project but also to supporting tasks, like responding to open issues or reviewing others' pull requests. The former is considered to be "fun" work that potential contributors clamor to take on, but the latter is necessary work that tends to fall exclusively on maintainers.

Eghbal, Nadia. *Working in Public: The Making and Maintenance of Open Source Software* (p. 57). Stripe Press. Kindle Edition.

Ease of participation refers to how easy it is for someone to contribute to the project. Whether a project is easy to contribute to is partly a function of its technical scope, but there are other factors involved: the quality of its documentation, how responsive the maintainers are, the perceived hostility (or lack thereof) in social interactions, and the tools or preexisting skills required to make a contribution. Most important is the question of whether the project is on GitHub or not.

Eghbal, Nadia. *Working in Public: The Making and Maintenance of Open Source Software* (p. 57). Stripe Press. Kindle Edition.

User adoption refers to a project's reach: What is the total addressable market of contributors? How many people could potentially contribute to this project? Since most contributors start out as users of an open source project, user adoption is one heuristic for estimating a project's total potential contributor base.

Eghbal, Nadia. *Working in Public: The Making and Maintenance of Open Source Software* (p. 58). Stripe Press. Kindle Edition.

Ideas on how to break down tasks on the road map

A few of the conditions that Benkler identifies as necessary to pull off commons-based peer production are intrinsic motivation, modular and granular tasks, and low coordination costs.

Intrinsic motivation is the currency of the commons: members do the work because they want to do it. In the case of open source, it's assumed that developers participate because they enjoy writing code.

Modularity refers to how the project is organized. Can it be broken into clear subcomponents, and do they fit easily back together?

Granularity refers to the size of each module. It should be easy for anyone to jump in and complete the task without too much preexisting knowledge.

Assessing the Health of Emulsify

Ways of contributing beyond code

- EDUCATION, in the form of live-coding, or making videos, blog posts, or tutorials about the project
- SPREADING THE WORD, in the form of organizing events, posting about the project online, or giving talks at meetups or conferences
- SUPPORT, in the form of answering user questions on mailing lists, issue trackers, or chat
- BUG REPORTS, in the form of filing an issue with the project after discovering a bug**

How to Measure the Health of the community

- NUMBER OF OPEN ISSUES AND PULL REQUESTS: A high number of open issues or pull requests suggests the project is not well maintained, although projects have different philosophies as to whether they prefer to keep issues open or closed.
- AVERAGE TIME TO FIRST RESPONSE, from a maintainer, on a new issue or pull request. (Sometimes, maintainers use bots as first responders.)
- AVERAGE TIME TO CLOSE AN ISSUE OR PULL REQUEST.¹⁸¹ Activity that directly feeds back into “work done” (meaning, changes to the code) should weigh more heavily than issues and questions. The latter might help us measure a project’s popularity, but a project that’s all talk and no action (lots of discussion, but no commits) is not necessarily healthy. By contrast, “issue zero” and “pull request zero,” with a steady rate of issues and pull requests opened and closed, would suggest that a project is in comparatively better shape.
- Eghbal, Nadia. Working in Public: The Making and Maintenance of Open Source Software (p. 112). Stripe Press. Kindle Edition.