in

Guide To:

Benchmarking Your Design System
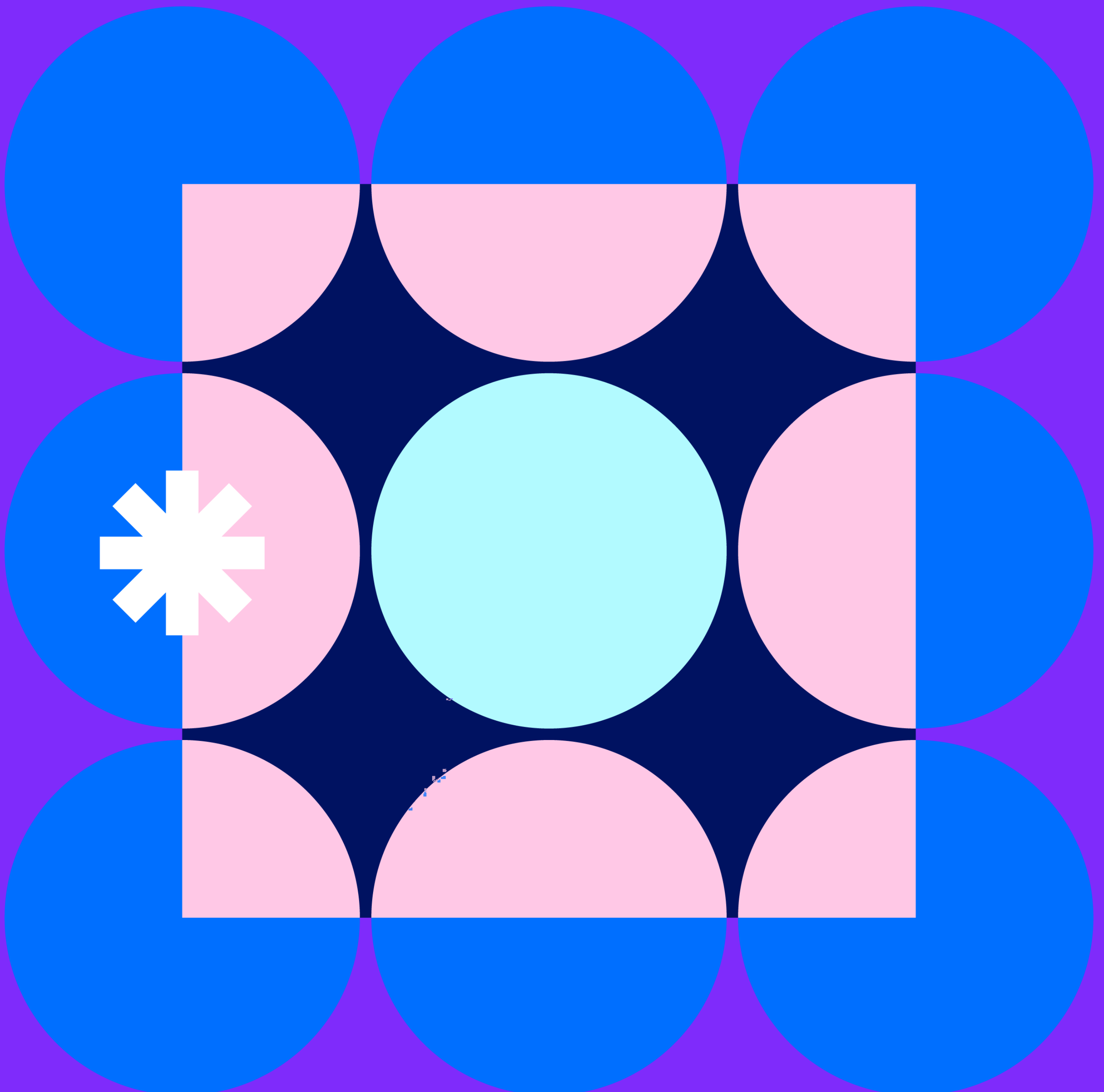
Table of Contents

Guide To:

Benchmarking Your Design System

Introduction

Digital product design and development is experiencing its own industrial revolution. Decades after IBM introduced the first commercial software in the late 1960's, the majority is still hand-crafted. Each button, form field, and navigational component is often custom designed and coded. It's as if every time a car was manufactured, each of the thousands of parts was designed and engineered individually, without any of the interchangeability and interoperability a modern assembly line affords.

This methodology is, of course, massively inefficient, and creates inconsistent user experiences. Design systems offer a solution. Through a library of visual guidelines, reusable components, and necessary documentation, they make the creation of new products efficient and cohesive.[1]

1. Adapted from Nathan Curtis's definition: https://medium.com/eightshapes-llc/defining-design-systems-6dd4b03e0ff6

What is a Design System?

A collection of ↓	Used by both ↓	Controlled with ↓

Foundations
Basic building blocks

A collection of	Used by both	Controlled with
Foundations Basic building blocks	**Design** Assets created and used in a design tool	**Guidelines** Written usage, rules and recommendations
Colors	Design Files	Accessibility
Typography	Libraries	Examples
Icons	Styles	Patterns
Spacing		Principles
Grids		Usage
Motion		Voice & Tone
Components Reusable UI elements made with foundations	**Engineering** Implementation in front-end development	**Governance** Adoption, maintenance processes, analysis
Buttons	APIs	Communication
Cards	Components	Contribution
Input fields	Tokens	Incentives
Pagination		Measurement
Tabs		Reviews
Tables		Versions

As changes in markets, regulations, and user behavior happen more rapidly, the need for more efficient design processes (and tools) becomes even more urgent. Product teams with mature design systems have the ability to pivot quickly, transforming the entire user experience in days, rather than weeks or months. That makes the entire enterprise more resilient–and

ultimately more successful–than organizations that lack the speed a healthy design system imparts.

Our team at InVision created this guide to share our point of view on where the design system industry is today, based on research and interactions with customers. **Here's what you can expect:**
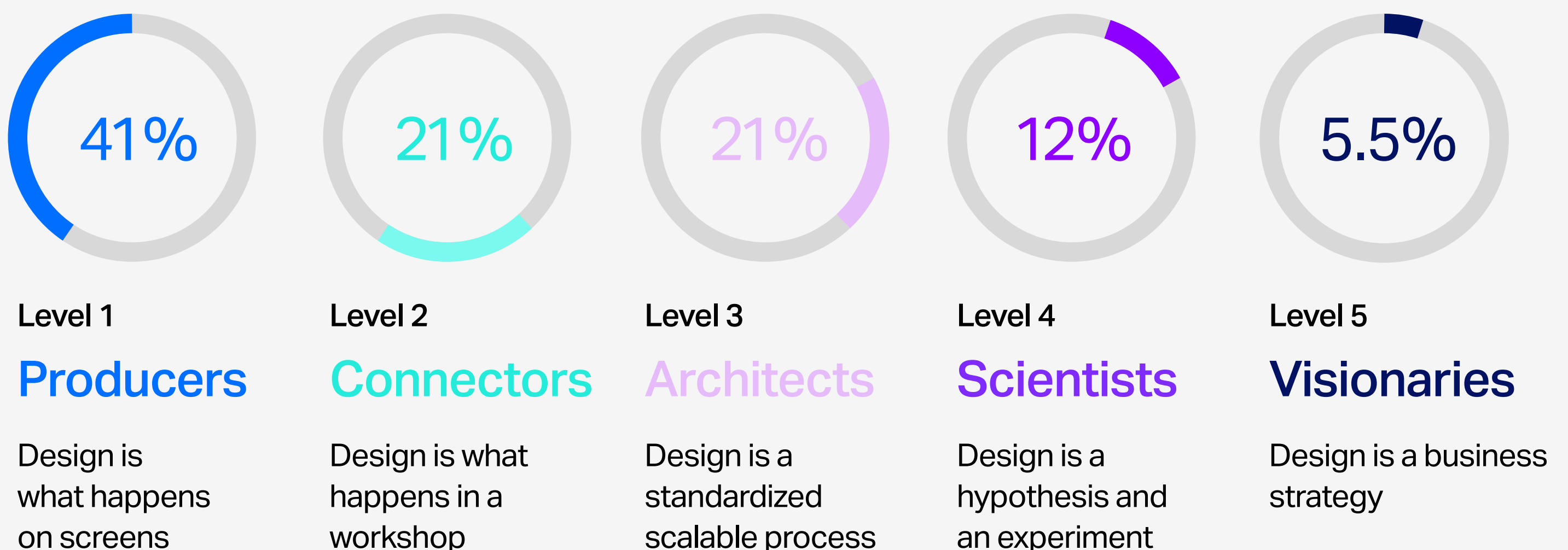
✓ We'll discuss a framework that our team has developed to help organizations benchmark the progress of their design systems, which you can use to assess your own team's progress.

✓ Based on your self-assessment, we'll provide concrete recommendations to help scale your design system across your product teams.

✓ We'll call out some ways that InVision's Design System Manager can help your team create, document, and evolve your design system.

✓ We'll finish by discussing why the teams creating design systems find them so challenging, often facing a feeling of "imposter syndrome," and what you can do to move past that.

This guide is for design system makers who are looking for resources to help them advance. It's also for design leaders who are looking to benchmark their own design system against the industry and invest in leveling up their teams to create more robust, efficient systems.

Where design systems are today.

We've known for a number of years now that design systems are critical to building digital products efficiently at scale, but when looked at across industries, there is still much progress to be made. In 2019, InVision released a report called *The New Design Frontier*, which surveyed thousands of companies to explore the relationship between design practices and business performance. From the survey data, we created a design maturity model which helps companies understand what differentiates high-performing design organizations, with the ultimate goal of providing the tools to evolve and improve their bottom line. The majority of the companies surveyed (41%) were still at the lowest maturity level (the "Producers") while only 5% where at the highest level (the "Visionaries").

Design maturity levels from *The New Design Frontier*

41%	21%	21%	12%	5.5%
Level 1	Level 2	Level 3	Level 4	Level 5
Producers	**Connectors**	**Architects**	**Scientists**	**Visionaries**
Design is what happens on screens	Design is what happens in a workshop	Design is a standardized scalable process	Design is a hypothesis and an experiment	Design is a business strategy

It was clear from the data that the higher performing, more mature teams invested heavily in design systems. **For example, 50% of companies in the Visionary category had dedicated design systems teams**, versus only 16% of Producers.

Design systems	lvl 1	lvl 2	lvl 3	lvl 4	lvl 5
A dedicated team to maintain the design system	16%	23%	26%	36%	50%
Code snippets, APIs, microservices, etc.	18%	28%	29%	34%	53%
Interaction patterns across the product suite and across multiple channels	16%	26%	27%	47%	52%
Content/editorial guidelines	28%	35%	30%	46%	52%
Application of elements from the design system	24%	35%	48%	50%	71%
Revisiting previously launched work and making continuous improvements	23%	39%	49%	59%	76%
An online tool that documents the design system and is accessible to others	30%	40%	45%	53%	59%
Updating elements in the design system	30%	51%	59%	66%	78%
Integrations between designer and developer tools (e.g., JIRA, Inspect, etc.)	41%	52%	59%	67%	78%
UI best practices (accessibility, internationalization, designing for web vs. mobile, etc.)	44%	60%	65%	80%	82%
Design principles (brand values, purpose, objective, product principles)	44%	59%	67%	79%	73%
Visual identity guidelines (voice and tone, color palette, fonts, etc.)	77%	81%	84%	94%	94%
Style guides, patterns, design language system	61%	76%	83%	89%	96%

A large majority of respondents were investing in related activities, if not an actual system. Across all levels of maturity, on average 81% of companies surveyed were using style guides, patterns, or a design language system of some kind. For the Visionaries, this was as high as 96%. In contrast, on average only 58% of companies surveyed were using an online tool that documents the design system and is accessible to others (and Visionaries only bumped that up by 1% to 59%).

Luckily, from our one-to-one interactions with customers out in the field, we're learning that the situation appears to be evolving rapidly. Several large banking institutions, in an industry that our data indicated is on average at a lower level of design maturity, have been using DSM as a tool to document their design system and keep it accessible to others. And large media organizations like ViacomCBS are using DSM to maintain a shared language between designers and developers.

" InVision DSM is the perfect home for any organization's design system. It provides a space to create and maintain a shared visual language between designers and developers at scale."

Jason Grotrian
Senior Director of Product Design
ViacomCBS

While the picture isn't entirely rosy, most teams are dealing with the same hurdles between design maturity levels, such as disagreement on standards and a highly-fragmented workflow between developers and designers. And although there's a lot written about the different approaches to building design systems, including InVision's Design Systems Handbook, there isn't much discussion about how they evolve over time. Because of this, even teams with high maturity wrestle with the feeling that other companies have more efficient systems, in a kind of organizational "imposter syndrome." Teams need a clear way to assess where they stand now, and how to level up.
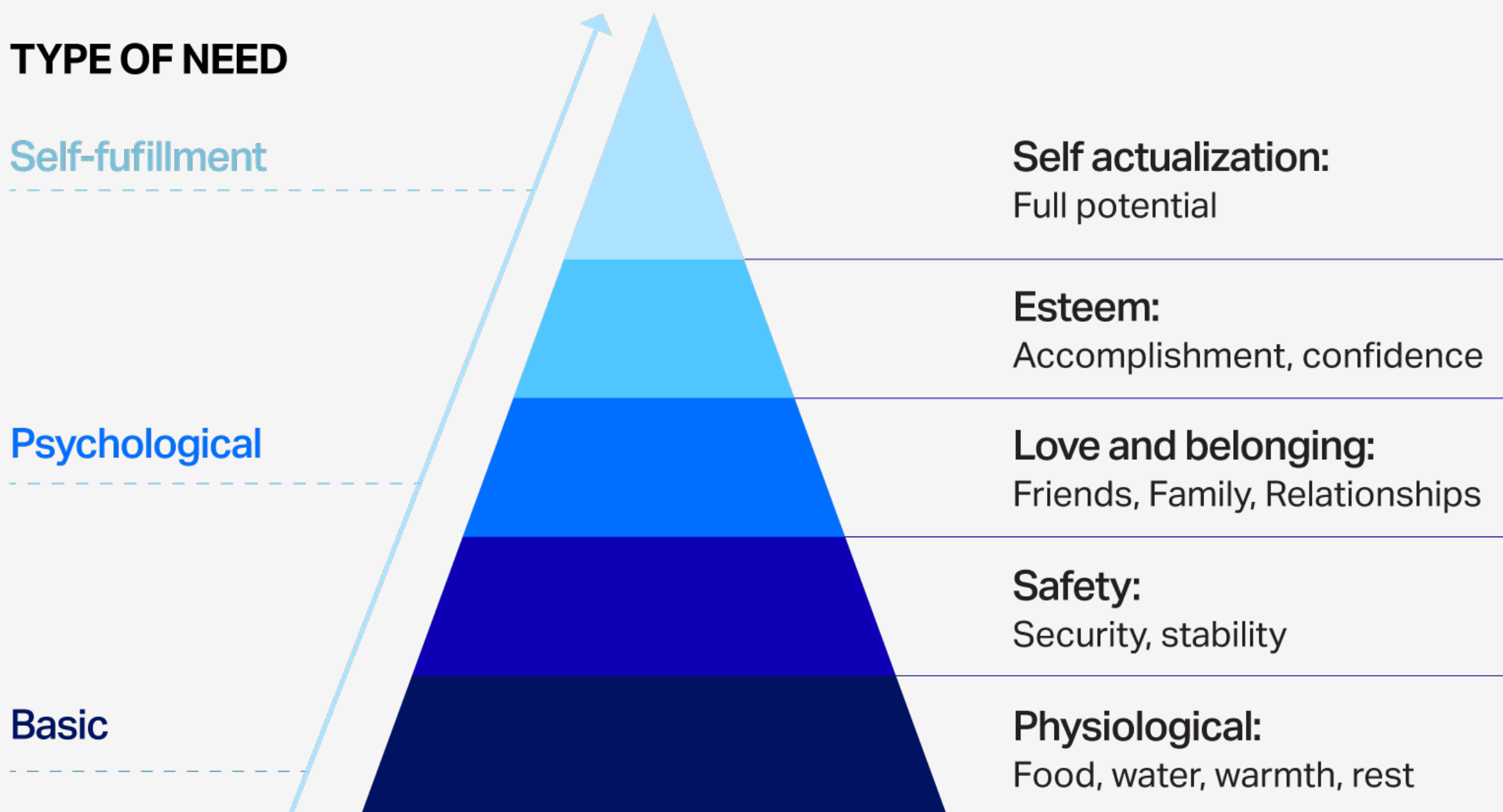
Benchmarking your design system.

Recently at many companies, the conversation has shifted from "why do we need a design system?" to "how do we build and/or scale our own?" Through our research and from speaking with customers, we can confidently say that in order to answer that question, you need a good understanding of where your design system currently stands. We find it works best through the context of a hierarchy of needs.

In 1943, the psychologist Abraham Maslow wrote a paper proposing that all humans are motivated by a

series of needs, and that each level of the needs must be met in order to progress to the next level.

Maslow's Hierarchy of Needs

Needs lower down in the hierarchy must be satisfied before individuals can attend to needs higher up.

TYPE OF NEED

Self-fufillment

Self actualization:
Full potential

Esteem:
Accomplishment, confidence

Psychological

Love and belonging:
Friends, Family, Relationships

Safety:
Security, stability

Basic

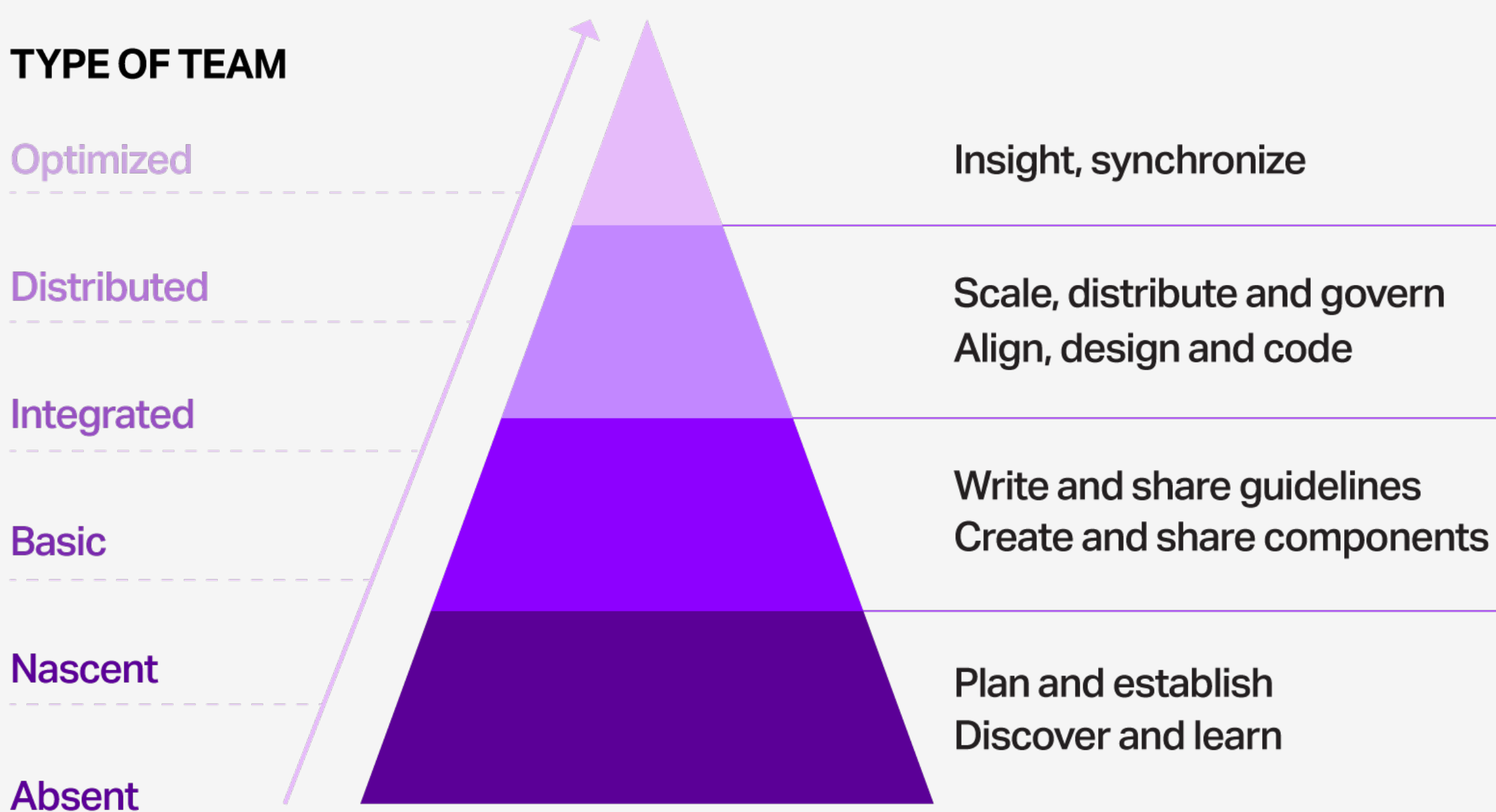**Physiological:**
Food, water, warmth, rest

From our research we've established a hypothesis that there is a similar hierarchy for design systems, where teams need to build a foundation before they can advance to more integrated workflows higher up in the hierarchy. Attempting to evolve your systems without a strong foundation can lead to frustration and failure, an experience we've heard echoed from many of our customers. Too often, teams put months or years of work into their design system, "launch" it, and then wonder why no one uses it. If

they skip steps (e.g. try to scale a system without first doing the necessary discovery work and audits), then the design system won't gain traction and will ultimately fail. We're excited to share our framework, and the accompanying recommendations, in the hopes you avoid that outcome.

Design System Hierarchy of Needs

Teams need to build their design systems foundation before they can advance to support more integrated workflows higher up.

TYPE OF TEAM

Optimized — Insight, synchronize

Distributed — Scale, distribute and govern
Align, design and code

Integrated

Basic — Write and share guidelines
Create and share components

Nascent — Plan and establish
Discover and learn

Absent

We'll assume that your team has already begun the work to discover and learn about design systems (you're reading this guide, right?) So to understand better where you sit, let's dive deeper into the other five benchmarks for design systems teams, talk about some of their associated behaviors, and offer some resources for upleveling your team:

Nascent → Basic → Integrated → Distributed → Optimized

Nascent teams are just getting started with their design systems and haven't yet standardized their approach. They may still be doing some research, but they've begun to put a plan in place to start creating and rolling out their own system.

Here's a checklist to see if you're at a *Nascent* stage:

✓ You've begun to evaluate the current state of your design and code assets by auditing your products.

✓ You've created a rough plan for getting your first design system in place.

✓ You've identified a starter team model to staff your design system, for example with a guild (see video below), or by following Nathan Curtis's approach to building a multidisciplinary team.

✓ You're working to set up and deploy tools to manage and use your design system.

Watch how LinkedIn uses Guilds to bring together their Design system.

RESOURCES

If you're at this stage, here are some resources that can help your team level up from Nascent to Basic:

Find the right team model.

If you haven't yet staffed your team, or it's still a work-in-progress, you can learn about the different models for design systems teams in Chapter 2 of the Design Systems Handbook.

Audit your patterns.

Learn how to audit your existing patterns and create a visual inventory in Chapter two of the Design Systems Handbook.

Create naming conventions.

Learn the best practices for defining your design system taxonomy.

How DSM Can Help.

If you're just getting started, InVision Design System Manager (DSM) can give a big head-start to your process. A major news organization recently used DSM to stand up a secure design system documentation site within hours, customizing it to their unique needs, work that has taken other teams months to do. If you're interested in learning more about creating custom documentation in DSM, you can read more on our support site, or talk to one of our experts today.

Nascent → **Basic** → Integrated → Distributed → Optimized

At the Basic stage, teams have begun to build components and usage guidelines for designers or developers, but have not yet integrated them cross-functionally so that both teams are using a shared language. Many teams at this stage also have an online tool, like DSM, that documents the design system and is accessible to other parts of the organization.

Here's a checklist to see if you're at a *Basic* stage:

- ✓ Your developer teams and/or your design teams have built components and usage guidelines, but they aren't integrated with each other.

- ✓ You've created a basic UI kit in a design tool to share reusable design assets.

- ✓ You have an initial definition for stylistic elements such as colors, typography, and icons that are part of your design system.

- ✓ You've started to iterate on your system based on team feedback.

- ✓ You're starting to evolve governance practices to understand what's working and not working with the system.

Rethinking designer-developer collaboration

Brad Frost and Dan Mall discuss how design systems can be a middle ground for designers and developers.

RESOURCES

To level up to the next stage, it's important to start building a shared language between designers and developers. Here are some resources that can help your team:

Iterate on your audit to build shared language.

You may have already done a basic audit in the Nascent stage, but you can uplevel your audit with an InVision Board, calling out the names and descriptions of each section as the beginning of a shared language. As part of this practice, discuss inconsistencies, get design and dev involved, and see where language diverges.

Establish your purpose and principles.

Design systems pioneer Nathan Curtis makes a good argument for creating design system principles. At the highest level, this allows for a shared language and set of criteria that all elements should adhere to. When making decisions about what should/shouldn't be included in the system, you now have a shared vocabulary to facilitate

decision-making. An extensive set of example design system principles can be found at Design Principles FTW.

Create a contribution guide.

A contribution guide allows people across the organization to have a clear set of requirements in order to contribute back to the system, which will help prevent duplicated effort (e.g. recreating components that already exist) and can also help ensure additions to the system are reusable, accessible, etc. In this article on *Inside Design*, you can see how Apartments.com used a simple google survey for team contributions.

How DSM Can Help.

With InVision DSM, designers can quickly access the most up-to-date design assets and versions right in Sketch while using native design libraries. They also have easy access to rich documentation guidelines on how to use the system.

Design systems teams can maintain design libraries in just one place, and seamlessly publish updates to DSM. Teams can share and showcase branded design system documentation that is flexible and easy to maintain from the web.

In conjunction with the DSM toolset, our consulting and training teams can help you level up as part of a complete package. If you're interested in learning more, you can talk to one of our DSM experts today.

Nascent → Basic → **Integrated** → Distributed → Optimized

Teams that are Integrated have established processes to keep design and coded libraries aligned and up to date (e.g. sets of symbols in a Sketch file are mapped to components in code). The team also has processes to keep design assets, coded assets, and documentation aligned, and so they have a regular release cadence to version their design system that both designers and developers can use. Additionally, they may have integrations between designer and developer tools like JIRA, Inspect, or others. There likely is a lot of manual maintenance and communication to keep things running smoothly at this stage.

Here's a checklist to see if you're at the *Integrated* stage:

✓ Your design system components reflect both design assets and coded components.

✓ You've established processes to update both design and coded assets once an update is made.

✓ You've defined a checklist or rules necessary to evolve and contribute to the design system.

✓ You've established communication processes to inform your team about design system updates.

✓ You have started to implement processes that identify conflicts between design and code.

To level up to being a Distributed team, you need to have processes in place that will help you start to scale your design system across multiple product teams, geographies, and platforms. You'll also need to have a more robust versioning mechanism in place, and processes that help you enable contributions back to the system. Here are some resources to help your team:

RESOURCES

Define a decision tree or component lifecycle process.

As author Brad Frost writes, "You can have a comprehensive design system that contains a slew of well-structured components, thorough documentation, thoughtful guidelines, and a well-considered design language. But if a design system user can't get done what they're trying to get done, the whole system risks obsolescence." Coming up with a system of governance and decision tree, as Frost shares in the above article (or in this example from Cap Watkins), can help users understand what to do when they get stuck.

Upgrade to a tiered system (or system of systems).

As design systems scale across an organization, it is often necessary to use a tiered approach (or system of systems). Nathan Curtis of Eightshapes outlines this approach, which

includes developing a "core," relevant to the entire organization, and tiers within business units below it.

Establish design tokens.

Design tokens keep code in sync as design standards change and help your team focus less on coordination and more on building experiences your users love. Watch Brad Frost, Dan Mall, and Josh Clark discuss how they approach design tokens, or take a look at Amazon's Style Dictionary to see them in action.

How DSM Can Help.

InVision DSM can help Integrated design teams level up to Distributed teams in three key ways: First, through the creation of live, coded components (via a deep Storybook integration) mapped to design assets. Second, DSM offers a draft mode or "staging environment" to align design, code, and documentation before releasing a complete version of these elements as a bundled unit. Third, DSM offers integration with InVision's Inspect product, where DSM components are surfaced right in the tool. This makes all the work of synchronizing designed and coded assets pay off, by reinforcing a shared language between designers and developers. If you're interested in learning more, you can talk to one of our experts today.

Nascent → Basic → Integrated → **Distributed** → Optimized

Distributed teams have scaled their design system assets across multiple product teams, sub-brands, and/or platforms. They have contribution processes in place for designers and developers to feel a sense of ownership over the design system.

Here's a checklist to see if you're at a _Distributed_ stage in your design system:

✓ You've set up more than one design system to reflect unique needs across your organization.

✓ You have shared your design system publicly.

✓ You have decentralized contributions to democratize the evolution of the design system.

✓ You've set up a governance model that identifies whether elements belong in a core or sub-system.

To graduate to Optimized, the final stage, your team will need to build custom workflows to eliminate manual work and keep design and code in sync. You'll know how your design system is being used across both design and code, and you'll have visibility into gaps within the system. Here are some resources to help:

RESOURCES

Keep a pulse on your team model.

Most design systems teams start with a centralized or federated approach and evolve towards a hybrid or cyclical model as they become more distributed, moving towards being Optimized. Now is a good time to review your team model to make sure it aligns with where you are as an organization, as the most effective model will likely change as the design system scales and becomes more broadly adopted.

Continue evolving your people practices.

Design systems will only be effective if they are adopted by the people that need to use them. Learn how the design team at Etsy fostered collective ownership of their system.

How DSM Can Help.

If you're at a Distributed stage, moving towards Optimized, DSM can assist you today in a few critical areas, and will soon be even more helpful. In DSM now, you can set roles and permissions to configure who has viewing vs. editing rights. You can also set up a versioning workflow to create holistic snapshots of your design/code assets and documentation, view a change log to compare versions, and have the ability to revert/switch versions on demand. Soon, you'll be able to manage design tokens (platform-agnostic variables that represent the look and feel of your brand and product) directly in DSM. To learn more about tokens, visit Salesforce's Lightning Design System Design Tokens. And if you're interested in learning more, you can talk to one of our experts today.

Nascent → Basic → Integrated → Distributed → **Optimized**

Optimized teams have automated workflows to keep design and code synchronized and up to date across multiple product teams, sub-brands, and/or platforms. They have analytics and reporting that provide insights into design system usage and adoption.

Here's a checklist to see if you're at an *Optimized* stage in your design system:

✓ You've built custom workflows that eliminate manual work to keep design and code in sync.

✓ You know how your design system is being used across both design and code.

✓ You have visibility into gaps within the design system.

✓ You've set up incentives that encourage design system evolution (such as finding ways to make your design system a living entity through continual contributions from designers and developers)–this might be incentivized by things like stickers, digital leaderboards, or tying contribution activity to performance reviews.

If you're here, congratulations! Only a very small subset of the teams with advanced design systems have achieved this stage of performance. And even if you are here, the work isn't over. InVision gets to spend a lot of time with large tech companies that are at the bleeding edge of design systems technology and would be ranked as Optimized. But even these companies have insecurities about their design system.
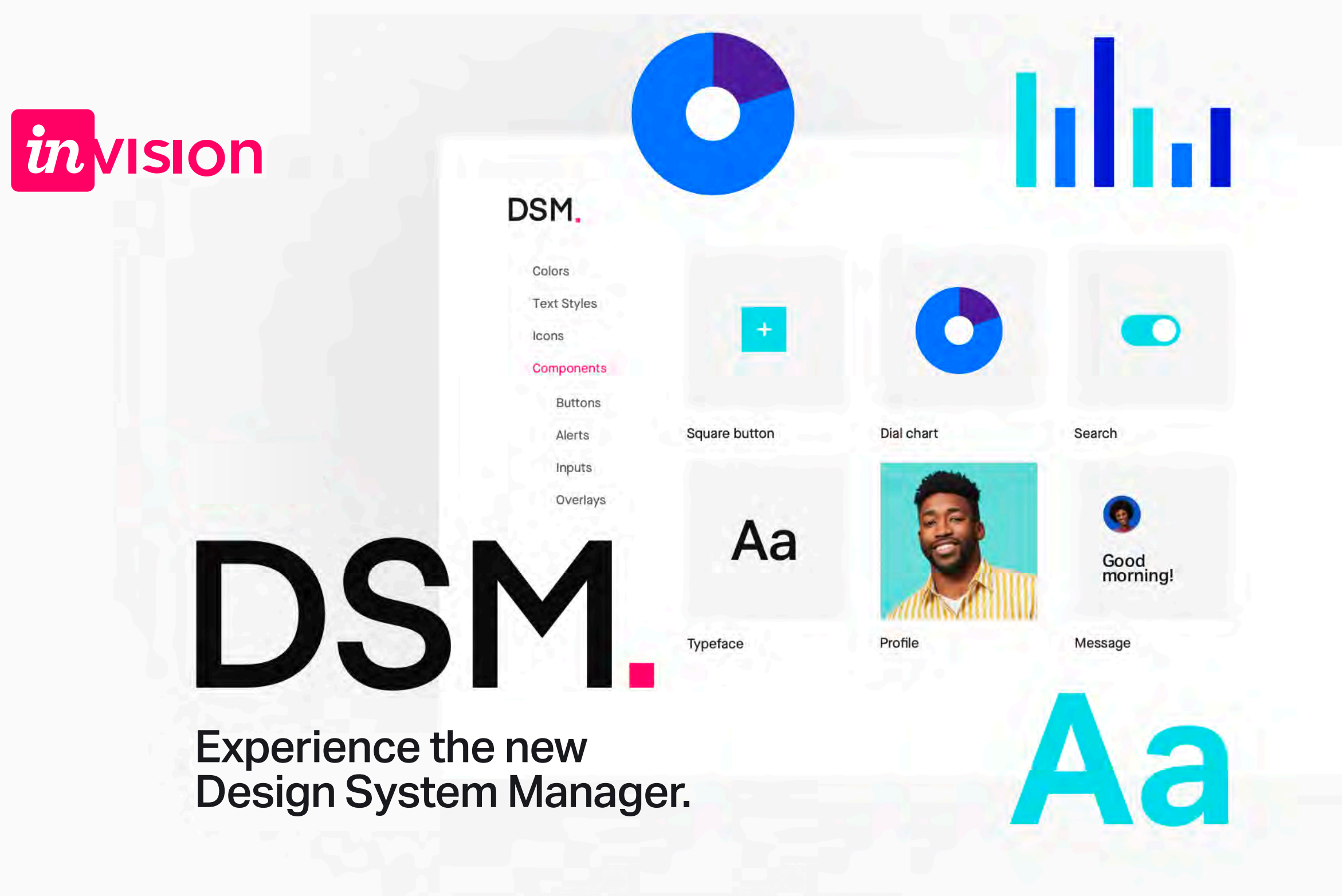
The truth is, building, deploying, and maintaining a design system is extremely hard, and because of the rate of change in the space, even organizations that have access to the most high performing teams and technology struggle at times. But it is also one of the best places to demonstrate the value of the work that your team does to your organization.

Moving your design system into the future.

One of the key benefits of evolving your design system is that it will allow your design teams to focus on efforts that directly impact the business. If you're no longer focused on repetitive UI design tasks, your team can spend more time with customers and build better experiences for them. If you're able to demonstrate the product development efficiency that comes along with a design
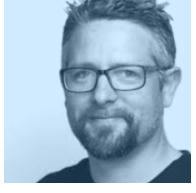
system—some teams see gains of 25% or more—you'll have an opportunity to engage more with business leaders and have a seat at the table for strategic decisions. This creates a virtuous cycle—as a design team you'll have a chance to not only continue to evolve your design system, but uplevel the design maturity of your whole organization. It's an exciting opportunity, and now is a great time to leverage the technology, tools, and community knowledge to move your design system forward.

Whatever stage you're at, we have your back. With InVision DSM, we can help you connect design and code so teams can work smarter, faster, and more in sync. Reach out any time to our experts to help you move your design system into the future

DSM.

Experience the new
Design System Manager.

Contributors

Eli Woolery, Design Stories Director
Researching and sharing best practices to guide digital product design. Co-host of the Design Better Podcast.

Marci Pasenello, Director of Product
Product lead for InVision's DSM who's been experimenting with the product development process for many years.

Nick Hahn, Director, Design Systems Consultant
Excitedly helping teams evolve their designs systems and bring forth the design system revolution as Director of Design Systems Consulting.

Micky Teng, Director of Product Marketing
Marketing InVision DSM to help customers solve key design system challenges.

Rebecca Kerr, Principal Conversation & Content Strategist
Exploring major product team transformations through interviews, stories, and conversation.

Catriona Shedd, Director of Product Design
Design lead for InVision DSM and design system advocate specializing in helping organizations improve design system workflows to increase adoption, foster collaboration, and elevate the efficacy of design systems at scale.

Zoë Adelman, Design System Consultant
Helping InVision customers create and evolve design systems for scale.

Ehud Halberstam, Product Manager
Working on InVision's DSM and these days focusing on developer-designer collaboration.