International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 07 | July - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Avoiding Rigidity: Designing Systems That Evolve

Author: Siddhi Suman, Tom Murphy, Srivatsa Rao
User Experience Designer | UX & Design Systems

Abstract

Design systems are created to bring consistency, speed, and collaboration across digital products. But when systems become too rigid with strict templates, inflexible components, and one-size-fits-all rules, they can slow down innovation instead of enabling it. This whitepaper explores how to avoid that rigidity by designing systems that evolve alongside real-world needs.

1. Introduction

Design systems usually begin with good goals to make the experience consistent, speed up work, and help teams work better together. But many design systems fail when they become too strict, turning into rigid rules that hold things back instead of helping them grow.

A truly sustainable design system must be adaptive and structured enough to provide coherence, yet open enough to embrace innovation.

2. Research

*" **80% of usability issues can be uncovered with just 5 user tests**. "- Nielsen Norman Group*

Continuous UX research can reduce rework by up to 50% by catching issues early in the design cycle. So research should not be treated as a one-time task but as an ongoing part of the design system lifecycle. As user needs shift and products evolve, the system must adapt accordingly. Without continuous research, a design system risks becoming disconnected from real-world usage and team workflows.

Beginning with foundational research to understand how users interact with the product and similar tools in the ecosystem ensure components are grounded in actual behavior rather than assumptions. Beyond initial research, conducting regular usability testing, not only on the product but on the design system itself to evaluate whether the system is genuinely helping teams or introducing unnecessary complexity.

Establish clear and simple feedback loops to allow designers, developers, and product teams to report friction points or suggest improvements. Keeping the system connected to real experiences ensures it remains useful, flexible, and aligned with evolving needs.

3. Building Components

Components in design system work best when they are modular, composable, and adaptable. They should be treated as building blocks that can be reused in different ways, not as final, unchangeable designs. Each component is best designed with a single responsibility — it should do one specific thing well. Components that try to handle too many functions at once often become harder to use and maintain.

A composable structure helps teams combine smaller components to create more complex interfaces. This approach avoids locking teams into rigid templates and allows more freedom to shape layouts as needed.

Using a smart variant strategy can allow components to support different use cases like small, medium, and large buttons, without needing to create entirely new components for each version. This saves time and makes the system easier to manage. For example, instead of building a unique button for each use case, a base button with size, color, and icon variants serves majority of all product needs. This helps in reducing duplication, improves consistency and enables scalability.

Instead of forcing one-size-fits-all solutions, a set of adaptable base components should be provided. These can be customized and assembled in different ways to suit each product's unique needs. This approach makes the system more

flexible, helps teams move faster, and reduces the chances of creating components that are too specific or too difficult to reuse.

4. Tokenization

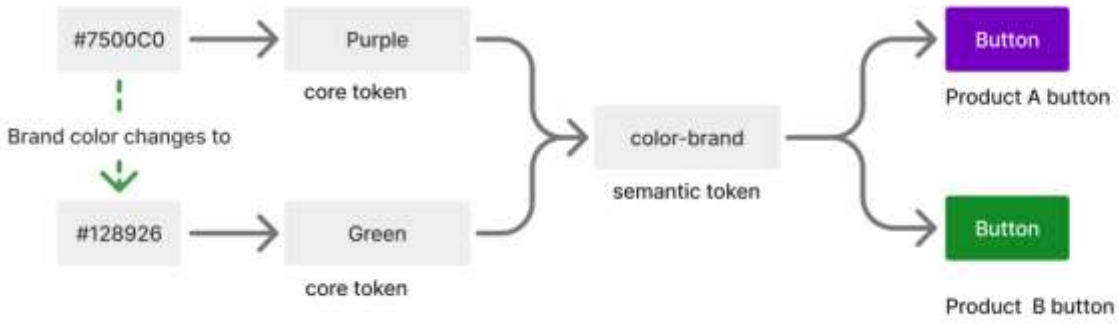
Design tokens act as the foundation for styling decisions across the system. They define values like color, spacing, and typography in a way that stays consistent and is easy to update. Tokenization makes it possible to clearly separate how something looks from how it works, which helps design systems scale and adapt more easily.

Semantic tokens are created using names based on purpose rather than appearance. This makes it easier to update the visual design without changing how components behave or what they mean.

Tokens also help support theming and accessibility. When visual properties are defined through tokens, it becomes easier to switch between light and dark modes, apply different brand themes, or improve visual contrast, all without needing to rewrite components. Tokenization makes maintenance simpler. When a design update is needed, changing a token, updates every place where the token is used. This saves time and lowers the risk of inconsistencies across the product. When the system is built around tokens, design becomes more flexible, reusable, and easier to scale across platforms and brands.

For example, as shown in Fig 1, if we decide to change error color “Red 300”, then changing a single design token updates all the UI elements instantly, across multiple products where the Red 300 color is used.

[Fig 1] Semantic token



[Fig 2] Color abstraction with tokens

As shown in Fig 2 above, project teams can easily reuse the button component without having to recreate unique buttons and properties for every unique product. They would need to simply change the property value, such as the hex value for a button color, for ‘color-brand’ to achieve the desired results.

5. Accessibility

Accessibility plays an important role in a design system and should be included from the very beginning. When accessibility is part of the foundation, the system becomes more usable for everyone.

Accessible color tokens that meet WCAG contrast standards can be used to make sure that text, buttons, and other UI elements are easy to read, including for users with low vision. Color combinations need to be chosen carefully, and color alone should not be used to show meaning — icons, labels, or patterns can be added to help with clarity.

Component design can also include support for keyboard use and screen readers. Elements like modals, dropdowns, and forms should have clear focus states, follow a logical tab order, and use proper labels. This often involves working with developers to make sure the visual design supports how the component works.

Accessibility can also mean being flexible with different user needs. Consistent spacing, readable typography, and layouts that work at different sizes can support users with cognitive or motor difficulties.

Making accessibility a core part of the design system, instead of adding it later, helps create more inclusive experiences and sets up a strong foundation that grows with the product over time.

For example, when a design system uses color tokens (like color-font-primary, color-background-default) instead of direct color values, it's easier to ensure that all text and UI elements meet accessibility contrast standards right from the start. When color tokens are pre-validated to conform with WCAG AA contrast standards (minimum 4.5:1 for normal text, 3:1 for large text [2]), all interface elements referencing these tokens inherently satisfy accessibility compliance. This reduces the need for later-stage audits and prevents the manual remediation of accessibility issues across numerous individual elements.

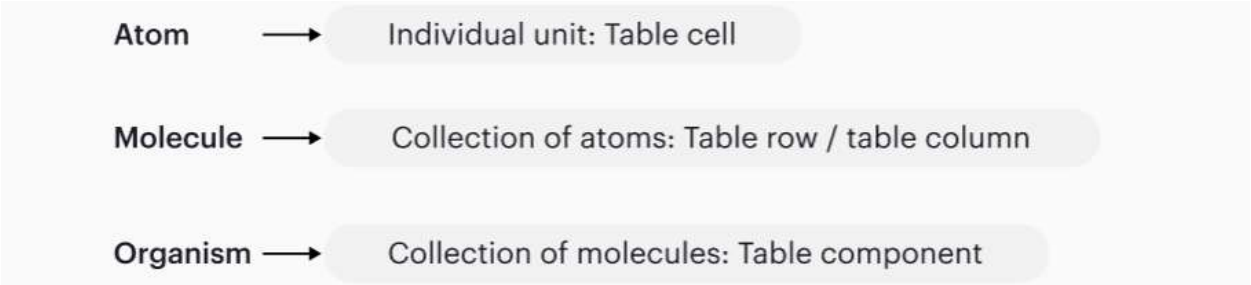
6. Atomic Design

Atomic Design offers a clear way to structure a design system by breaking the interface into smaller, reusable parts that can be combined to build consistent and scalable experiences.

It begins with atoms — the simplest elements such as buttons, inputs, and labels. These are the foundational components used throughout the system. Atoms are grouped into molecules, which are small functional units like a search bar that includes an input field and a button.

Several molecules can be combined to form organisms, which are full sections of an interface, such as a navigation bar or a product card. These organisms fit into templates, which show the layout of a page without final content. Finally, pages are created by filling templates with real content to show how everything works together in practice.

This method makes it easier to keep everything consistent and organized. A change to an atom like updating a button style will automatically apply wherever that atom is used. Atomic Design supports consistency, speeds up workflows, and enables systems to evolve in a structured and predictable way.



[Fig 3] Table component built using Atomic Design

In line with atomic design principles, each table cell represents an atom, the most basic unit of the component. Rows and columns are constructed as molecules, composed of these atomic cells. The full table, as an organism, is assembled from these molecular structures. When a project requires a table configuration not available as a predefined component in the design system, teams can compose it using these atomic units. This approach preserves consistency, ensures reusability, and maintains alignment with the system's underlying architecture.

7. Realizing the Benefits of Modular Design

In building design systems, the goal is not just consistency but adaptability. This whitepaper has outlined a set of principles and practices to help avoid rigidity and design systems that evolve with user needs, product goals, and technological shifts.

By grounding decisions in continuous user research, systems remain connected to real-world behaviours. Modular components and variant strategies ensure that UI patterns can scale without becoming restrictive. Through tokenization, style and structure are clearly separated—making maintenance, theming, and accessibility simpler. Embedding accessibility from the start supports inclusive experiences, and following atomic design principles offers a repeatable, scalable approach to building UI across products. Together, these methods create design systems that are not only efficient, but resilient—able to grow, adapt, and thrive in fast-changing environments. The focus moves from enforcing control to enabling creativity, making the system a living foundation that serves both users and teams over time.

7. Conclusion

Creating a design system that remains effective over time requires more than just a set of predefined rules or components. It demands an approach that values flexibility, user-centered research, and scalability. By focusing on continuous research, modular components, tokenization, accessibility, and atomic design, a system can stay adaptable, inclusive, and relevant as needs evolve.

With a modular and composable architecture, components can be mixed and matched to suit different contexts, while tokenization ensures that visual updates can happen effortlessly. Accessibility should be integrated into the system from the start, making sure that it serves all users without extra work down the line. And by applying atomic design principles, the system remains consistent yet flexible, supporting both small tweaks and large redesigns.

Ultimately, a design system should never be static or rigid. Instead, it should grow and evolve alongside the product it supports, staying aligned with user's needs and technological advancements. Building design system with this mindset ensures that the system remains a powerful tool for the team, enhancing creativity, collaboration, and consistency.

8. References

Brad Frost *Atomic Design

W3C Web Accessibility

Nathan Curtis *Modular Design Systems

NN Group