



Design Systems: A practical guide to creating design languages for digital products

Alla Kholmatova

Download now

Read Online ➔

Design Systems: A practical guide to creating design languages for digital products

Alla Kholmatova

Design Systems: A practical guide to creating design languages for digital products Alla Kholmatova

Not all design systems are equally effective. Some can generate coherent user experiences, others produce confusing patchwork designs. Some inspire teams to contribute to them, while others are neglected. Some get better with time, more cohesive and better functioning; others get worse, becoming bloated and cumbersome.

Throughout this book, Alla Kholmatova, previously a lead designer at FutureLearn, will share an approach and the key qualities of effective, enduring design systems. It's based on Alla's experiences, case-studies from AirBnB, Atlassian, Eurostar, TED, and Sipgate, plus 18 months of research and interviews — all attempting to figure out what works and what doesn't work in real-life products. It may not answer every question, but it will help you figure out just the right strategy for establishing and evolving a design system in your organization.

Design Systems: A practical guide to creating design languages for digital products **Details**

Date : Published September 2017 by Smashing Magazine

ISBN :

Author : Alla Kholmatova

Format : Paperback 288 pages

Genre : Design, Computer Science, Software, Business, Nonfiction

 [Download Design Systems: A practical guide to creating design la ...pdf](#)

 [Read Online Design Systems: A practical guide to creating design ...pdf](#)

Download and Read Free Online Design Systems: A practical guide to creating design languages for digital products Alla Kholmatova

From Reader Review Design Systems: A practical guide to creating design languages for digital products for online ebook

Mark DeHate says

As someone who's starting the process of creating a design system, this book has been very timely. It's packed with clear and straightforward examples from a wide variety of existing systems. It's full of real-world and concrete advice, as well as stories of design systems that could have gone better.

Highly recommended for anybody that's thinking about implementing a design system.

Vu Tran says

The author has laid out a nice foundation for any team that currently seeks create or manage a systems for their products. The book caters to designers, engineers, and writers as it provides different perspectives of how each member thinks when working in a systems team.

Simon Vandereecken says

Alors que je suis plongé en plein dans un projet de Design System, ce livre était un bon point de départ. Les idées qu'il donne font énormément dans ce sens ce genre de projet et permettent de développer de très bonnes fondations ainsi qu'une réflexion intéressante autour du projet. Il offre de plus un tour d'horizon assez intéressant, bien que trop succinct à mon goût, de divers systèmes de design utilisés par des firmes de tailles différentes. Bien qu'assez intéressant il reste extrêmement théorique, ne s'attardant pas énormément aux outils. Mais il restera ainsi intemporel et une bonne référence pour ce genre de projet, les outils étant appelés à se développer et à changer profondément.

John C says

Alla provides great insights and perspectives, not only for the design of interfaces themselves, but also what's necessary at a people level in order to have a strong philosophical foundation for tackling big projects and involve stakeholders across design, development, marketing, and more.

Nicholas De Lucena says

A very straightforward and practical guide to the subject. For those with a design or programming background, the book feels a little bit obvious at certain parts but overall it is a very useful guide on the subject.

Wilson Jimenez says

Review

Good insight on an interesting topic, a bit redundant but still good content. If you've some experience in the industry, some concepts will be somewhat clear by now like the importance of getting buy-in or the intricacies of coordinating work and communication between product teams within a company.

There are many references to great articles through out the book that I'd like to read next, would've liked to have them as an aggregated list at the end like a bibliography.

Notes

Design Systems

- Functional patterns generally have an HTML basis while perceptual ones a CSS basis.
- An effective design language bridges the gap between the system image and the assumed user model.
- Design systems are a combination of patterns and practices.

Design Principles

- Vague: "make it clear"
- * Practical: only one number one priority, what do you want your users to do first?
- Vague: "make it simple"
- * Practical: make it unbreakable, make sure it's designed for exploration and impossible to mis-tap.
- Vague: "make it useful"
- * Practical: start with the needs, do research, analyse data, don't make assumptions.

Functional Patterns

- Can be seen as the nouns and verbs of a sentence.
- When creating a component, each part has to have a purpose, i.e.: in a card component, we may want the title to tell what the card is about, the image to have a quick preview and draw attention, the description to allow easy card scanning.
- Patterns change/evolve, behaviours stay.
- Don't try to design the most impressive header forgetting what the header component is for and how it affects the user journey.
- * Don't loose the connection between user behaviours and the patterns to encourage those.
- UI Inventory by Brad Frost: great way to modularise interfaces, you end you with a collection of parts grouped in categories.
- * Run inventory every few months.
- When defining a pattern, start with it's purpose, not what it is.
- * To broaden potential its use cases, try calling it with a verb, not a noun.
- * If you have a promotional module, if you call it Image Header or Course Promoter, you may be restricting its use to a particular context.
- ^ Better to name it after an action (behaviour) like Promote/Discover a Course, Encourage people to join, etc.
- Tweaking a module (like reducing font size) to allow a use case where we want the content to be smaller or not push another element out of view, might be the wrong call.

- * Maybe we're using the wrong patterns which aren't robust enough for the content behaviours.

Perceptual Patterns

- Use personas to come up with a list of them.
- * You can use ambience if personas are hard to come up with, i.e., what ambience suits your branding traits more?
- Come up with a 5-7 list of branding traits.
- * Like fun but not childish, informal but not sloppy, powerful but simple, etc.
- * These traits are brought to life in the UI through tone of voice, visuals, interactions, sounds.
- Techniques to assess patterns.
- * Mood boards: loose way of grouping related style elements together like on Pinterest.
- * Style tiles and element collage are more high fidelity options.
- * Don't force consistency every time, it can lead to a generic and rigid branding, there's a difference between consistency and uniformity.
- If you introduce a new element or pattern in one place, it may not feel as part of the brand or disconnected until you add it to other parts of the site.
- Some patterns should be used sparingly in order to keep the brand focus.
- * Like using too much of a colour or a triangle shapes when the brand uses squares.

Shared Language

- Patterns should connect through a shared language.
- Name things based on purpose not based on what they look.
- * Or how users call them, which forces engineers to get a user perspective.
- Try keeping a glossary as well.

Parameters of your System

- Strict <-> Loose
- Modular <-> Integrated
- Centralised <-> Distributed
- Strict pattern libraries may benefit bigger teams or companies that would like to provide precise, predictable and consistent outputs.
- * Loose libraries are more flexible by encouraging more exploration, design acumen and sensitivity to context; and may not have the need to thoroughly document every pattern.
- ^ Even though, clear, comprehensive and compelling documentation is fundamental for this type of systems in order to keep a solid foundation.
- Pattern library sketches and naming convention should always be in sync with its code counterpart.
- Modular systems can create more overhead while making parts reusable; are suited for designs that need to scale and evolve, repeated elements, teams working concurrently.
- Integrated systems are suited for one-offs like ad campaigns and fewer repeating parts.
- Don't sacrifice UX or potential impact for the sake of reusability.
- Design system organisation can be centralised (owned by a team) or distributed.

Planning and Practicalities

- Time and money is saved on reusing, you're not rebuilding the same button component every time.
- Easier site wide changes.

- Faster product launch: prototyping a new page out of a pattern library is usually a matter of days, a new design can take weeks.
- Brand unity at scale: you don't end up with different parts of the site not looking like part of the brand.

Systemising Functional Patterns

- Look at the main screens of the site, try to break them down into behaviours, group elements by purpose, draw patterns.
- Put stuff on a specificity scale, the more specific the less reusable it can be.
- Think about potential changes, if I change module A, do I want the change in other places?
- * Some modules might look similar but we may decide to treat them differently.

Systemising Perceptual Patterns

- Voice and tone.
 - * Direct, positive, encouraging.
 - * Friendly but reserved; "well done!", rather than "wohoo, you're a superstar!"
 - * Elements of humour; "we're hatching something new"
 - Colour.
 - * Primarily white colour palette with pink accents.
 - * Pink to blue gradient (celebration).
 - Shapes.
 - * Mostly square and occasionally circular.
 - * 1 px border, square icons.
 - Typography.
 - * Mostly large type face with focus on readability.
 - * High contrast, large headings in relation to body.
 - Animations
 - * Small to no movement.
 - * Pink to blue subtly hover, opacity, gradual colour change.
-

Milana Stanic says

This should be considered a handbook, not just a book, you can go through it and something will stick BUT ultimately you will have to make a specific case for your specific needs and this book as Santa's little helper will do it. It has completely changed my mind on some things and opened my eyes for others, I am amazed. So, why 4 then? The structure is weird for me, I am not sure how to explain it apart from, it is not chronologically set and that confuses me. All the operations I need to do I have to plan out by chasing bits and pieces, sure would help if I made notes XD

Awesome book overall, would recommend :)

Nick says

Alla writes a handsome book here, catering to the needs of visual and UX designers seeking a systematic and

modular approach to their work. There's much redundancy on basic design concepts, though: identifying principles, building consensus, clarifying functions, the difficulty of naming. And there are only passing references to front end implementation strategies, which has my interest. Recommended, for visual designers, but not developers per se.

Nadya Tsech says

I would recommend this book to designers who haven't started to build their own library or just at the beginning. It has a lot of useful theory and some practical tips on building the library and less on maintaining and improving.

Matej Latin says

A few golden nuggets here and there but a lot of repetition. I feel it could be shorter and more focused. I loved the part of systemizing the functional and perceptual patterns, that was the best bit!

Stephanie Tanata says

Great book about general knowledge of design system and how to create one by examples from great design library in AirBnB, Atlassian and TED

Bram says

I think it's a great introduction to Design Systems. For me it lacked a bit of depth. But still, every product owner, designer and developer should read it..

Cat says

Instead of thinking in terms of static pages, teams are starting to approach design in a more systematic way. This book talks about how to make an effective design system that helps generate coherent user experiences, inspires teams to contribute to them, and get better with time. Here are my notes:

Design patterns are any repeating, reusable part of the interface (eg button, typography). A design system is a set of connected patterns and shared practices for a digital product.

The domain influences functional patterns, such as data fields and charts vs videos and discussion threads. Product brand influences perceptual patterns, like tone of voice and typography (differentiate products within the same space). Platform conventions also shape patterns (eg iOS vs Android).

Most design patterns are established and familiar since new patterns require users to learn and adopt them

first. Products are distinct because of the way patterns are applied, not because the patterns themselves are new.

One of the main goals of design systems is to get a group of people to follow a creative direction consistently. When you are approaching design with a system in mind, you can affect the design in more profound ways (changing across the entire system instead of one place at a time).

A pattern library is a tool to collect and share design patterns and principles for how to use them. Often these live in a document or wiki, but a better practice is to have a living library that contains the patterns as well as live code used to build them. Be aware that just having this living library won't guarantee that people use the components properly. Communication around the interpretation is key. Rigidity is a function of the design system behind it, not the library itself.

Effective design principles are:

- Authentic and genuine (eg think of "timeless, not cutting edge" compared to "simple, useful, enjoyable")
- Practical and actionable (eg "only one number 1 priority" vs "make it simple") - pair with an example of a specific part of your interface
- Have a point of view (help designers work out priority and balance when there are conflicting factors to consider - eg "direction over choice")
- Relatable and memorable (optimal number is between 3-5 for memory)

When defining design principles, start with the larger purpose of the product, look for shared themes among team members, focus on the right audience (the people directly involved with creating the product), test and evolve your principles (eg as part of design critiques).

Functional patterns enable certain user behaviours. Articulating the purpose of patterns early can help prevent duplication as the product grows.

Techniques for defining functional patterns:

- Create a pattern map by mapping core modules to sections of a user journey
- Conduct an interface inventory by grouping components in categories (eg navigation, buttons, lists, etc)
- View patterns as actions (eg "promote a course" vs "course banner")
- Sketch a pattern's content structure
- Place similar patterns on a scale (eg visual loudness)
- Treat content as a hypothesis

Understand how patterns link to the behaviours they were designed for (their purpose) since this will determine structure, content, and presentation.

Perceptual patterns are the aesthetic properties, like tone of voice, typography, colour palette, iconography, etc that help express brand image and differentiate products within the same domain. Perceptual patterns create visual coherence and connect parts across a system. Explore perceptual patterns using mood boards, style tiles, and element collages (each method distinguished by their relative fidelity). Expect to iterate and evolve styles as they're integrated into the product.

If a design only prioritizes perfect consistency, it can become rigid and generic. Leverage "signature moments" to add depth and meaning to an experience. Try small-scale experiments to see if elements work well before integrating changes into other areas of the interface. Be conscious of new requirements that demand custom patterns and one-off tweaks since they can dilute the brand.

Creating cohesive design systems requires a shared language and a shared approach to naming patterns. Names should be focused, memorable, and embody the purpose of the modules they represent. Instead of focusing on presentational attributes (eg circle with a dot), look for metaphors from other industries like architecture or publishing. Good names communicate purpose - if you're struggling to come up with a name, maybe the purpose isn't clear. Get a diversity of perspectives when naming patterns (eg content, development, design, research). Make design patterns visible (eg putting up posters on a wall) and refer to objects by their names. Take new employees through the story of how the guidelines were created so they can understand how decisions have been made. Have regular design system catch-ups to keep everyone on the same page as the system evolves. Keep a glossary of the terms you use (this can be an easily accessible pattern library).

Design systems can be characterized along three attributes: rules (strict/loose), parts (modular/integrated), and organization (centralized/distributed).

- Stricter systems provide precise and predictable outcomes, but sometimes result in making UX compromises for the sake of consistency. Allow for opportunities outside the systems such as creative experiments and side projects to understand and challenge the rules. To have a looser system, the team needs to be aligned on a product's purpose and the design approach.
- In modular systems the parts are interchangeable and can be assembled in various ways to meet user goals. A modular approach is agile because multiple teams can design and build in parallel, is cost effective because modules are reused, and is easy to maintain. On the other hand, integrated designs can be more specific to particular content, context, or art direction and are actually quicker to build as one-offs because you don't have to spend time making them reusable. The degree of modularization can change over time.
- Rules and patterns can be managed primarily by one group of people, which helps with ownership and focus, but sometimes results in a bottleneck. Alternatively, in a distributed model, everyone who uses the system is also responsible for maintaining and evolving it. A distributed model is more agile and resilient, but work can sometimes slow down without someone in charge.

Get support for a design system from senior management by demonstrating the time saved on designing and building modules, time saved on making site-wide changes, faster product launch, and brand/visual consistency. Start by agreeing on goals and objectives, make progress transparent (eg blog posts), and plan tasks with the most benefit upfront (eg quickly document patterns through screenshots and go back to add the living code after).

Systemize functional patterns by starting with a purpose-directed inventory. The goal is to define the most essential design patterns, not account for all visual inconsistencies. This process should happen after the foundational UX work is complete (research, content strategy, information architecture, and design direction). Identify key behaviours, group existing elements by purpose (at the same level of granularity), and define patterns (when to merge and when to keep separate - look at content structure).

Systemize perceptual patterns by looking beyond style properties (like colour values and text sizes), and focus on having a consistent meaning throughout the interface. Identify signature patterns by having team members write down the most memorable elements that make your product feel a certain way. Itemize styles, such as colour, interactive states, animations, typography, etc. For each style, start with the purpose, collect and group existing elements, define patterns and building blocks, and agree on the guiding principles. Document examples using screenshots to show how the style is used. Be aware of how decisions in favour of consistency can alter the overall aesthetic of the site and go back to signature patterns to help find the right balance. Test colour contrast for accessibility. Consider using a base value and providing several incremental steps - for colour or animations (eg how fast something animates).

When building a pattern library, focus on the content rather than which tool you use to document it. Think about the structure in terms of components and styles (functional and perceptual patterns). Document functional patterns with name, purpose (be concise), example (visual and code), and variants (show how they differ). Consider documenting related patterns to show alternatives for when the pattern is not quite what someone is looking for. When documenting perceptual patterns, show how the properties are used and how they work together. Agree on how new patterns will be added to the system. A template for pattern submissions can be useful if contributions come from all over the organization. Agree on the role of the design systems team (curator vs producer) and ensure the systems team are seen as partners rather than police.

Design systems free our time and energy to solve bigger and more meaningful problems.

Jean-Francois Potvin says

Learned a lot on pattern libraries and design systems. Interestingly enough, our team is starting to build the equivalent for our digital products. Will definitely continue to read on that subject and try to put in practice some of the advices provided in the book. Next up : Inclusive Design Patterns.

Ardavan Mir says

I rated 5 because this book served the purpose it promised to me. It exceeded my expectation. Alla creates a simple and comprehensive categorization for design patterns which she explores each one more in details throughout the book. The book is full of different examples of processes and practices of different teams.

I also appreciate the way Alla avoids complication of an issue that could easily get overcomplicated.
