Tallinn University

School of Digital Technologies

DESIGN SYSTEMS:

DEFINITIONS AND MAIN ELEMENTS

Master thesis

Author: Daniel Ranzi Werle

Supervisors: Sonia Sousa, PhD

Tallinn 2021

Author's Declaration

I declare that apart from work whose authors are clearly acknowledged, this manuscript is a product of the author's original work, and it has not been published before, neither is not currently being considered for publication elsewhere for any other comparable academic degree.

This master thesis document has been supervised by PhD Sonia Sousa (Tallinn University, Estonia).

Author: Daniel Ranzi Werle

Date: 05.01.2021

……...

(signature)

……..

(data)

Non-exclusive license to reproduce a thesis and make thesis available to public

I, xxx (date of birth: 28.06.1986), grant Tallinn University a permit (a non-exclusive license) to reproduce for free and make public in the repository of Tallinn University Academic Library a piece of work created by me,

“Design Systems: Definition and main elements ”,

supervised by Sonia Sousa.

I am aware of the fact that the author also retains the rights mentioned above.

I certify that granting the non-exclusive license does not infringe the intellectual property rights of other persons or the rights arising from the Personal Data Protection Act.

Tallinn, _____ 05.01.2021

Abstract

The digital transformation in our society led to the dramatic increase in the creation of digital experiences for an infinite number of applications. In order to keep up with the growth, digital product development teams have also grown exponentially. More products to be developed and more people developing them required new tools to ensure consistency in user experience and scale these experiences. Design Systems is a tool to help address these challenges. However, due to the novelty of the term, there is not yet a consensus of what defines a Design System and what elements should be contained in it. This research starts by briefly presenting what problem Design System trying to address, followed by a theoretical background review on three main areas of knowledge that contributed to the creation of Design Systems (visual design, human-computer interaction and web development), then moves on to a literature review. Based on this review, the research proposes a definition of the term 'Design System'. Then, the research presents the findings of an online questionnaire conducted on late 2020. The survey aimed at finding out what are the most important elements of Design Systems based on a list compiled by the researcher, combining both literature and elements from an online Design Systems database. The survey results are compiled and presented in a table of 18 elements, sorted by perceived importance, along with a discussion of the main findings. The research then concludes discussing its limitations and contributions.

Keywords: Design Systems, HCI, Human-Computer Interaction, User Experience, Web Development

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

It is fair to say that the digitalization of our world has transformed what were previously in-person services or interactions into digital products or experiences. Going to the movies has been transformed into watching a streamed film on your mobile device or smart TV; grocery shopping has become a purchase on an e-commerce platform on your laptop or mobile phone; meeting friends at a bar or going to a doctor's appointment has become a video conference call.

Although this phenomenon started years ago, the Covid-19 pandemic is amplifying the adoption of new technologies (Saliola, 2020). As expected, the restrictions imposed by governments in the attempt to deal the pandemic has made the usage of online services and internet connected devices surge sharply, pushing us to turn to digital services not just to work, but to play, to shop, to entertain ourselves and connect with other people (Koeze et Popper, 2020).

To this widespread adoption of Information and Communication Technologies (ICT), more specifically mobile devices and wireless communication, the term Digital Transformation emerged. Vial (2019) defines Digital transformation as the use of new digital technologies (social media, mobile, analytics or embedded devices) to enable major business improvements (such as enhancing customer experience, streamlining operations or creating new business models).

This Digital Transformation of businesses and experiences brought challenges not only for businesses as a whole, but also for user experience (UX) designers, who now have a seat at the table and are responsible for shaping these digital experiences, which have become a key factor in the digital economy. In this new setting, UX designers face multiple challenges.

The first challenge is the fact that digital experiences now have multiple touchpoints. A touchpoint is any instance of contact between the user and the service provider (Stickdorn, 2010). This term is widely used in Service Design, an emerging design field focused on the creation of well thought experiences using a combination of tangible and intangible mediums (Stickdorn, 2010). To exemplify this challenge, Roto et al (2016) provide this example of Service Design:

> *For example, when you are ill and need a prescription drug, you book an appointment from a doctor, see the doctor, get a prescription, visit a pharmacy to buy the drug, and use the drug. Digitalization can make many of these service moments online, which calls for UX design and service designers' understanding of the customer experiences throughout the whole customer journey.*

Further on this example, we can say that for a single service, these various online moments can happen by visiting different websites on a computer, or the same website on a mobile device, or even through a dedicated mobile app. To sum this up, Roto et al. define Multi-touchpoint experience design as a combination of service design (by designing for several touchpoints), experience design (by utilizing user needs) and omnichannel design (by utilizing brand identity) to define experience goals that address both the customer and the company needs.

Digital Transformation and the advances in ICT also changed the geography of jobs (Saliola, 2020). The exponential growth of technology companies required more talents, and ICT developments allowed people to work remotely and distributed across multiple locations. This posed another challenge for designers: maintaining consistency across these multi-touchpoint experiences.

In their Product & Engineering blog, the Design Team at HubSpot (an American developer of marketing software) Muscato (2018) describes the problems they faced in keeping their user experience consistent across multiple platforms and products. Across the company, there were over 40 product teams distributed in two continents, and from 2016 to 2018 their design team grew from 17 to 46 professionals.

In an early investigation of HubSpot's User Interface components, the team assigned to recreate HubSpot's design language found several user interface inconsistencies across their products, such as:

- 100+ shades of the colour grey
- 40+ text styles in 3 different fonts
- 16 different styles of modals
- 6 different primary buttons

- 5 different ways to filter a table
- 8 different date picker components

Figure 1 - Different examples of similar user interface components (buttons) found at HubSpot before their redesign. Source: Muscato (2018)

They identified that the reason there were so many variations of essentially the same styles and components was because HubSpot's organizational structure created visibility issues: it was very hard to discover what was already in play, and easier to just build something new. Muscato (2018) also explains that because teams were small, autonomous, and focused on

solving customer needs, it allowed them to ship products very quickly, but it meant that the different teams had a hard time keeping aligned, since each one was solely focused on their part of the product.

> *When you have 40+ product teams rapidly building, shipping, and iterating, it's actually pretty easy to lose sight of the overall customer experience. Being tightly focused on a specific problem often means you're putting on blinders to everything else. Because of these blinders, our designers and developers were unknowingly recreating existing elements, components, and patterns across our user interface. This led to a fragmented user experience and compounded design and tech debt.*

In order to fix these problems, HubSpot developed their own Design System called HubSpot Canvas. This very problem HubSpot faced can be applied to thousands of other organizations.

Stepanova (2018) summarizes Design systems as a systematic approach to creating and maintaining consistent user interfaces which coherently communicate the brand values and empower user experience. Google researchers Yew et al. (2020) define design systems as a repository of reusable components that follow a set of shared design principles that ensure design consistency and development efficiency across products. Kholmatova (2017) defines a Design System as a set of interconnected patterns and shared practices coherently organized to serve the purpose of a digital product. For Suarez, Anne et al. (2018), a design system unites product teams around a common visual language, reduces design debt, accelerates the design process, and builds bridges between teams working to bring products to life.

However, the idea of creating a methodology for defining and applying reusable patterns to solve design problems is not new (Fanguy, 2018). Perhaps the most notable example comes from Christopher Alexander's architectural work A Pattern Language: Towns, Buildings, Construction (1977), where he presents the idea of design pattern, which can be explained as a way to document a reusable solution to a common design problem. Design patterns are a universal resource to align best practices, describe the elements of good designs, and most importantly, provide a repository so that other people can easily reuse these solutions (Figma, 2020). These repositories were called pattern libraries, and an early example applied to interaction design is Tidwell's Common Ground: A Pattern Language for Human-Computer

Interface Design (1999), which then paved the way for her important book Designing Interfaces.

Over a decade later, with the rise of mobile technology and the need for responsive design, Brad Frost (2016) introduced the idea of atomic design, a methodology for creating design systems that breaks down user interface elements and patterns in five levels: atoms, molecules, organisms, templates and pages. This methodology became widely popular because it proposed a clear framework for creating user interface style guides that allowed designers to build upon and use repeatedly.

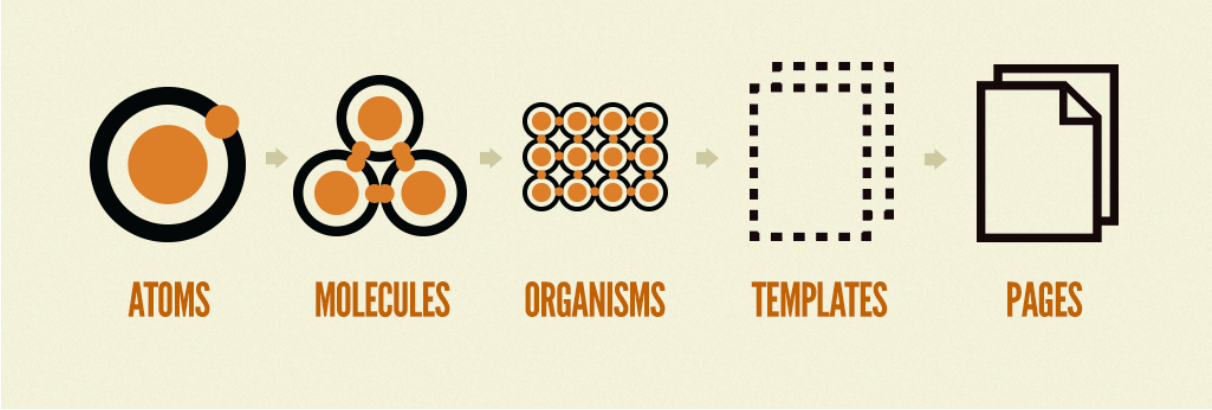


Figure 2 - Brad Frost's atomic design structure. (Frost, 2016)

Frost (2016) writes about the importance of defining and creating reusable user interface components that can be compounded into larger and more complex patterns, and that all these should be documented in a style guide (or pattern library) that contains specifications and examples of a given design language. Frost (2016) also elicits the main benefits of having a well-documented styleguide:

- **Consistency:** Web style guides promote consistency and cohesion across a user interface
- **Shared vocabulary:** Style guides establish a consistent, shared vocabulary between everyone involved in a project, encouraging collaboration between disciplines and reducing communication breakdowns.
- **Education:** A pattern library (manifested in a style guide) communicates the design language in a very tangible way, which helps stakeholders understand that an underlying system is determining the final interface.

- **Speed:** Once the pattern library is established, subsequent design and development becomes much faster
- **Testing:** A style guide allows you to view interface patterns in isolation, allowing developers to zero in on what's causing errors, browser inconsistencies, or performance issues

However, Frost's most important contribution to the Design System discussion is a shift in mentality: from having style guides (pattern libraries) as solely documents of good practices that were followed during the execution of a product (or to be followed on subsequent developments) as shown in Figure 3; to having pattern libraries that are actually live artifacts that reflect and document the underlying design system built using those design principles in mind, as shown in Figure XX.

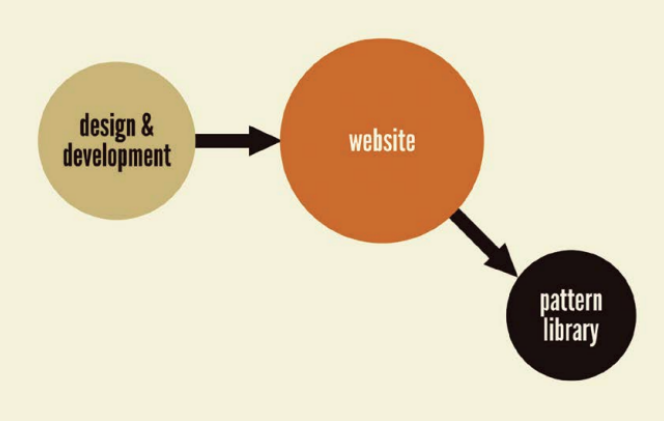


Figure 3 - Pattern Libraries as follow-up documentation after a project is done (Frost, 2016)

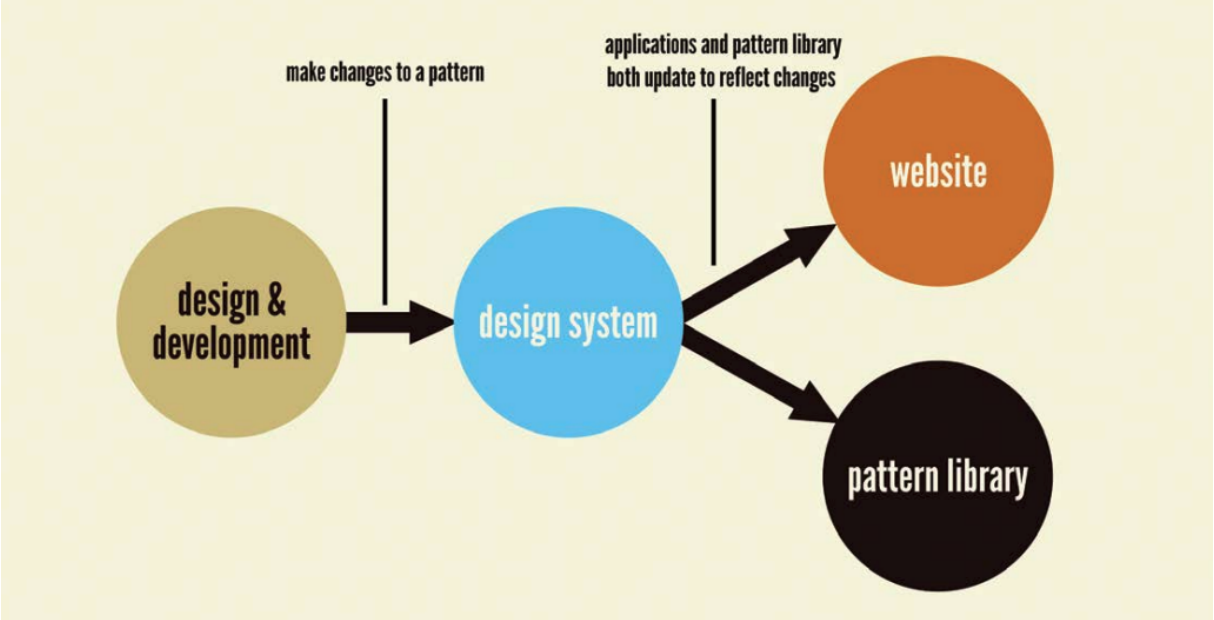


Figure 4 - Design Systems as underlying foundations for both products and pattern library documentation (Frost, 2016)

In Figure 4 we can understand 'Pattern Library' and 'Website' as separate products that are built using the same Design System as a foundation. In this sense Design Systems can be seen almost as products on their own, with their own development teams and release lifecycles.

And this is what happened in some cases. In 2014, Google launched its design system named Material Design, with the goal to unify the company's user experience across platforms, devices, and input methods (Google, 2020). On Material Design, Wilson (2018) wrote for Fast Company magazine:

> *The launch of Material Design became a landmark moment for the company. It was the first time that Google made an earnest attempt to unify all of its products with a common thread–the high-minded philosophy that a user interface should be treated as a tangible object, like paper. It actually took about four years for all of Google's services to adopt Material Design, but the response has been a win for the company. Developers implemented Material Design across millions of apps, while Google earned a reputation as a company that gets design every bit as well as Apple.*

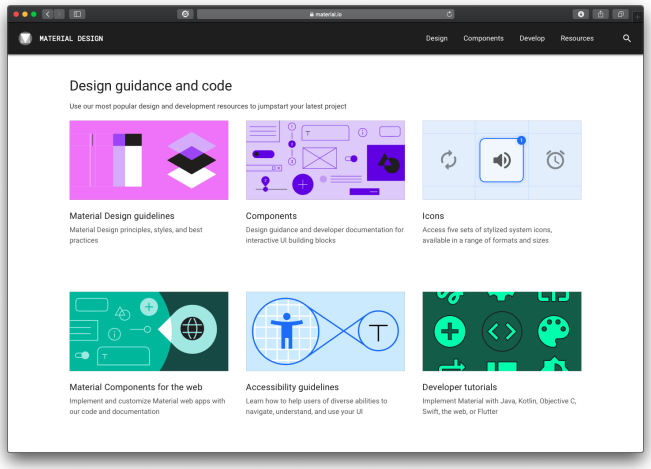


Figure 5 - screenshot of Google's popular Material Design system. Google (2020). Retrieved May 31, 2020 from https://www.material.io. Screenshot by author.

Google's Material Design was so successful as a Design System that other companies followed suit and made their Design Systems available online for the wider community, and not only for their internal teams.

Design Systems as a tool and as a process has grown so much in importance within the professional interaction design community that there are now blogs, podcasts, ebooks and conferences dedicated exclusively to this topic, not to mention dedicated teams within companies for building Design Systems.

However, because of the novelty of the term, there is not much academic research around this topic yet, nor a clear definition of what constitutes a Design System. There is even less academic research and evidence of what makes a good design system. These are the two main problems this research will attempt to address.

1.1 Research Problem and Significance

Because of the relatively recent emergence of 'Design System' as a term and as a tool between practitioner UX Designers, there isn't yet a consensus of what a Design System should contain, or what structure it should have. This makes it difficult for designers who are creating their own tools or choosing between existing Design Systems to assess these tools.

Konaté (2018), Dukic (2020), Kholmatova (2017) agree that there is no dictionary or otherwise standardized definition for Design System, or adjacent terms such as design pattern, and UX

guidelines. Konaté (2018) states that designers are left to create their own definitions for these terms and to select a naming policy for whichever documents they use to communicate with their colleagues regarding principles of design.

Immich (2019) goes as far as saying that there are few standards for establishing effective design systems, as these depend very much on the processes and organizational structures of the implementing organization. The author also states that too often, conventional UI style guides that do not go beyond the definition of visual details and design elements are simply renamed "Design System" without getting to the real core and benefit of the topic.

Yew et al. (2020) states that design system practitioners could benefit from guidance from HCI researchers, as they explore new tools such as Design Systems without consulting the literature or carrying out rigorous research.

Therefore, once established that there isn't in the literature a solid agreement of what a Design System is and that there is even less data on what it should contain to be considered effective, this research will attempt to shed some light on these issues.

1.2 RESEARCH MOTIVATION

The purpose of this study is to understand, using both literature and user-collected data, what comprises a good Design System. Based on this finding, the expected outcome of this research is the formulation of criteria and a list of contents that Design Systems should meet.

Given how little research has currently been carried out in this topic, and given how important Design Systems have become as a topic in User Experience discussions as well as a tool in most software product development teams, this research aims at bringing a more scientific and data-driven approach to this discussion.

1.3 RESEARCH QUESTION

The main research questions of this study are:

RQ1: How can Designs Systems be defined?

RQ2: What are the common attributes of Design Systems?

RQ3: What are the most important elements of Designs Systems according to UX professionals and practitioners?

1.4 RESEARCH PROCEDURE

The research methodology is organized in order to fulfil the research goals listed above. The following table summarizes the research procedures that will be developed on Chapter 3.

Steps of Research	Research aim	Research question	Method	Outcomes
Step One: Theoretical Background study	Compile a definition for Design Systems	How can Designs Systems be defined?	Literature review: papers, blog posts, articles, books	Summarized definition of 'Design Systems' based on literature
Step Two: Comparative analysis from dataset	Define attributes of design systems for survey, based on dataset	What are the common attributes of Design Systems?	Comparative analysis, comparative benchmark, comparative review from Design Systems dataset	Aggregated list of Design System elements
Step Three: Survey	Have the community rank design system attributes according to importance	What are the most important elements of Designs Systems according to UX professionals and practitioners?	Online survey with close-ended questions	Ranking of most important Design System Elements

Table 1 - Research Procedure

1.5 STRUCTURE OF THE THESIS

This thesis is divided into four chapters.

This Chapter 1 focused on staging the case for the whole research. It describes the context, the research problem and significance, research goal and motivation, research question as well as the research procedure.

Chapter 2 presents the theoretical background of the research, elucidating necessary concepts for better understanding the research and the historical developments behind the phenomenon being studied.

In Chapter 3 the research methodology is introduced along with the steps that the researcher undertook and the main findings that were uncovered from this process.

Finally, in Chapter 4 the researcher discussed the results achieved, analyses the results and proposes new developments.

The list of references and appendix are also provided at the end of the thesis.

2 THEORETICAL BACKGROUND

This section will explore existing literature written about Design System and other product development topics that are relevant to understanding Design Systems.

2.1 MANUALS, VISUAL LANGUAGE AND BRAND IDENTITY

> *All design systems start the same: as an attempt at making order out of chaos. Categorizing and organizing elements makes them easy to find and replicate. It is a necessary utility, born from the tech industry's need to move fast and innovate even faster. However, design systems are so much more than this. They define the language we use to create new elements and experiences. (Vesselov & Davis, 2019)*

Design Guidelines and manuals have been around for quite some time, ever since the need to create and communicate visual standards for diverse applications. The earliest examples can be traced back to the Bauhaus, a German design school and movement in the early 1920 that defended that rather than focusing on decorative elements, the primary facet was functional simplicity. The notion that all the parts on a page must have a function led to a focus on proportions, grids, and colour theory (Vesselov & Davis, 2019)

An early example of a visual design language and specification is the London Underground visual identity, with its classic *bullseye* design from circa 1925, its ground-breaking representation of the rail network from 1931 and the now traditional typographic typeface Johnston Sans, from 1916 (Damon, 2016)

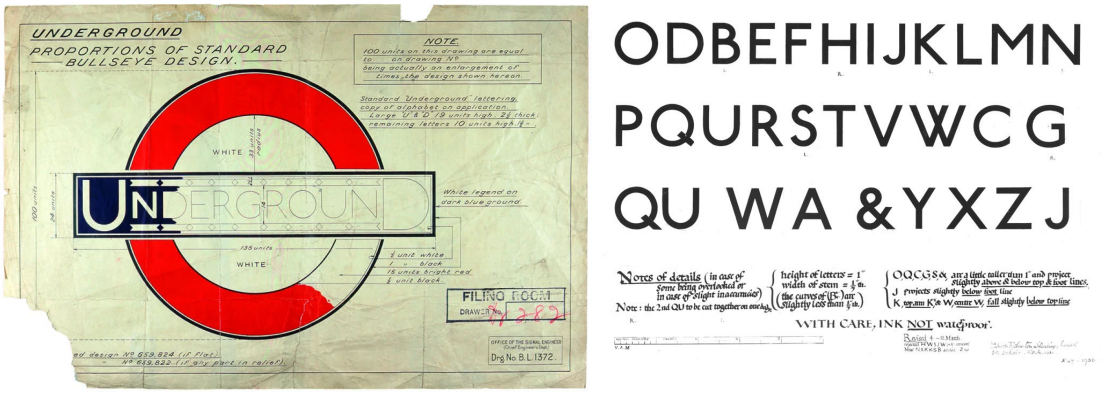


Figure 6 -Johnston Sans, London Underground's iconic fontface (Damon, 2016)

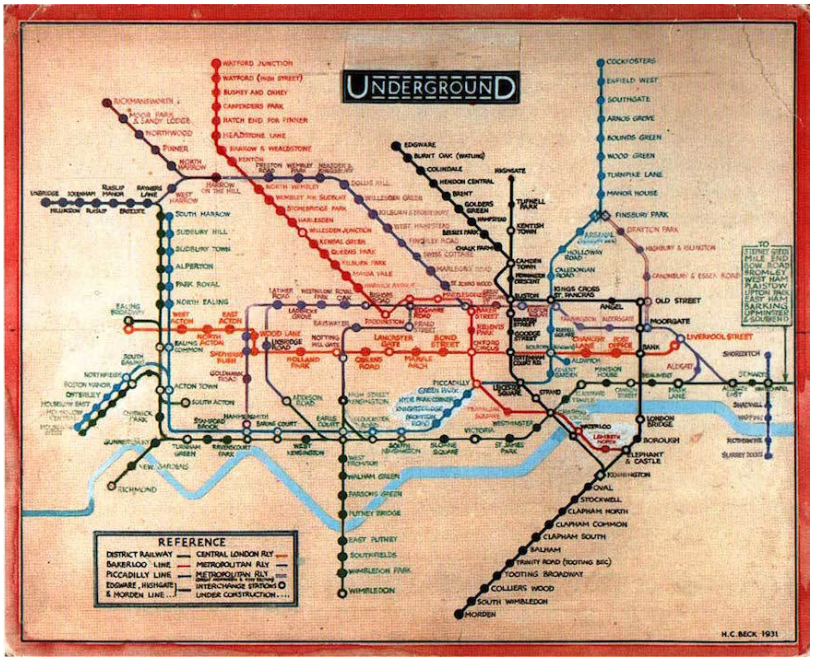


Figure 7 -Early prints of London Underground's map drawn in schematic straight lines (Damon, 2016)

As these visual design elements started being planned systematically, corporations created manuals and guidelines. One of the most emblematic examples is NASA's Standards Manual, a very comprehensive document released in 1976, detailing all visual manifestations of the NASA visual identity; or New York City Transit Authority's Graphics Standards Manual from 1970.

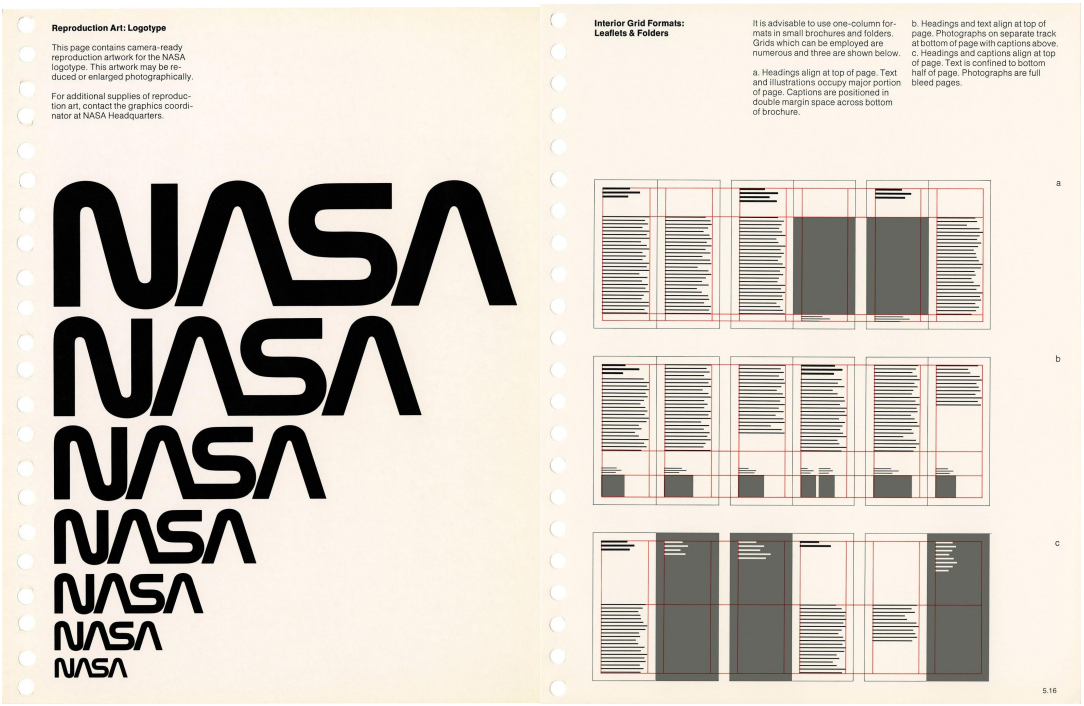


Figure 8 – Sample of NASA's Visual Identity Guidelines from the 70's (NASA, n.d.)

Figure 9 - New York City Transit Authority Visual Identity manual (NYCTA, n.d.)

The goal of these manuals was to create a shared visual language across all elements of the system. By language, (also visual language),we mean any system of communication that uses signs arranged in a particular way (Saint-Martin, 1990). However, instead of organizing nouns, verbs and adjectives manifested in sounds or words, designers organize shapes, colour, letters, signals and spacing as ways to convey meaning (Dondis, 1997).

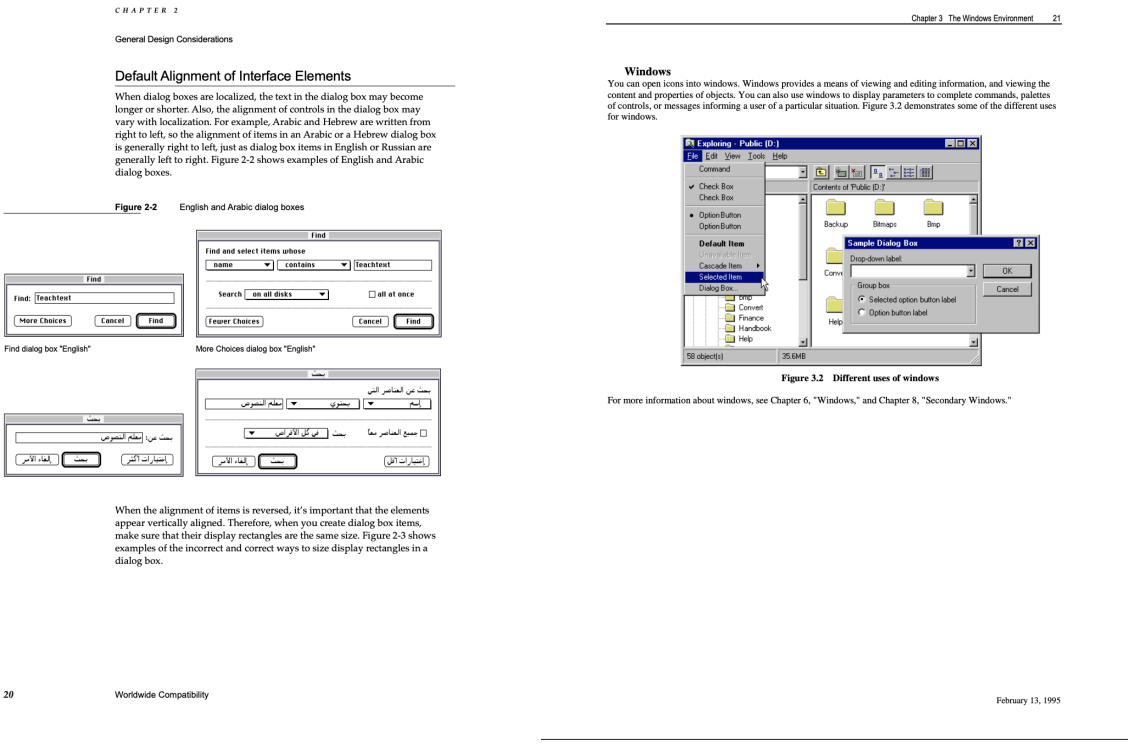
The use of these elements in carefully designed visual and physical products in a repeated and consistent way creates a visual identity, that is therefore used by organizations and brands to strengthen their corporate identity. Expanding this notion more broadly, we can say that the visual identity of a brand along with other brand manifestations such as the brand's logo, communication campaigns and even their websites and apps are part of the brand's identity. Brand identity is defined as the uniqueness and essential idea of the brand (Da Silveira, 2013)

Before the internet became ubiquitous as nowadays, NASA and Transport for London had to specify how their corporate and brand identities would reflect in products such as maps, signs, uniforms and even spacecrafts. Now, with the digital transformation of the late 20th and 21st centuries, brands and corporations need to think and specify how their brands will be manifested in the digital realm of apps, websites and digital experiences. Therefore, one of the aspects that Design Systems must embrace is how brands manifest their presence and identities in this digital setting: in sum, companies and organizations must define how the digital experiences they provide must look like.

2.2 PATTERNS, INTERFACES AND USER EXPERIENCE

In the following decades, computers became a household item, new technology and new terms emerged. The first aspect of computing technology where Design had a role to play was the GUI (Graphical User Interfaces), where users could see and directly manipulate representations of objects on the computer screen, rather than addressing the objects through an intervening command language code (Barnes, 2010). For example, Microsoft Windows allows users to point and click with a mouse on visual representations of documents to directly open the document without typing in a keyboard command.

For computers to reach the masses, they had to be *user-friendly*, so technology companies such as Microsoft, Apple and IBM created guidelines for developers with best practices on how to create interfaces for their systems. The "Macintosh Human Interface Guidelines" (Apple Computer, Inc., 1995), "Object-Oriented Interface Design: IBM Common User Access Guidelines" (IBM, 1992), and the "The Windows Interface Guidelines — A Guide for Designing Software" (Microsoft, 1995) are some examples of platform specific guides that provided guidance for designers in the developments (Ribeiro, 2020)



CHAPTER 2

General Design Considerations

Default Alignment of Interface Elements

When dialog boxes are localized, the text in the dialog box may become longer or shorter. Also, the alignment of controls in the dialog box may vary with localization. For example, Arabic and Hebrew are written from right to left, so the alignment of items in an Arabic or a Hebrew dialog box is generally right to left, just as dialog box items in English or Russian are generally left to right. Figure 2-2 shows examples of English and Arabic dialog boxes.

Figure 2-2 English and Arabic dialog boxes

When the alignment of items is reversed, it's important that the elements appear vertically aligned. Therefore, when you create dialog box items, make sure that their display rectangles are the same size. Figure 2-3 shows examples of the incorrect and correct ways to size display rectangles in a dialog box.

20 Worldwide Compatibility

Chapter 3 The Windows Environment 21

Windows

You can open icons into windows. Windows provides a means of viewing and editing information, and viewing the content and properties of objects. You can also use windows to display parameters to complete commands, palettes of controls, or messages informing a user of a particular situation. Figure 3.2 demonstrates some of the different uses for windows.

Figure 3.2 Different uses of windows

For more information about windows, see Chapter 6, "Windows," and Chapter 8, "Secondary Windows."

February 13, 1995

Figure 10 - Example of user interface guideline (Apple Computer, Inc., 1995)

These HCI manuals contained several design patterns that software developers and designers had to follow to keep consistency amongst products operating in that system. The concept of Design Patterns first introduced by Alexander (1977) is defined by Tidwell (2010) as a description of best practices within a given design domain; they capture common solutions to design problems. When defining Design Patterns applied to interfaces, Tidwell (2010) lists some characteristics of Design Patterns:

- Concrete, not general: patterns are concrete enough instructions to fill the gap between high-level principles and low-level "grammar" of the user interface
- Valid across different platforms: ideally, with minor changes, each pattern should work on different medias to solve the same problem
- Products, not processes unlike heuristics or principles, which advise on how to find a solution, design patterns *are* possible solutions to a design problem
- Relationships among elements, not single elements: a text field is not a pattern, but the spatial relationship between a text field and a piece of help text near it might be a pattern

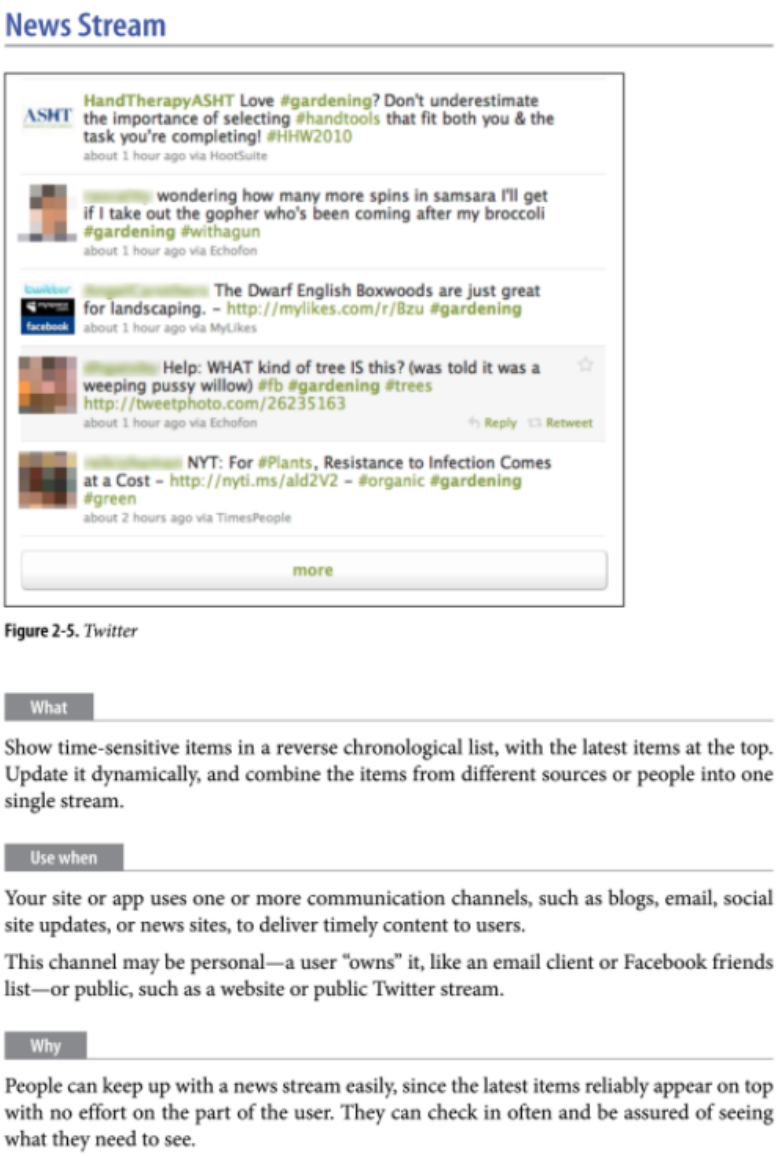


Figure 2-5. *Twitter*

What

Show time-sensitive items in a reverse chronological list, with the latest items at the top. Update it dynamically, and combine the items from different sources or people into one single stream.

Use when

Your site or app uses one or more communication channels, such as blogs, email, social site updates, or news sites, to deliver timely content to users.

This channel may be personal—a user "owns" it, like an email client or Facebook friends list—or public, such as a website or public Twitter stream.

Why

People can keep up with a news stream easily, since the latest items reliably appear on top with no effort on the part of the user. They can check in often and be assured of seeing what they need to see.

Figure 11 - Example of 'News Stream' pattern by Tidwell (2010, p. 34)

At its core, the end goal of a Design Pattern is to document a certain user interface behaviour that HCI researchers or practitioners believe will provide users with a good experience when attempting to perform a task using their software. However, it is important to understand what is understood by a good user experience.

2.3 USABILITY VS. USER EXPERIENCE

The field of Human-Computer Interaction was for a long time seen as the field of Usability Engineering (Karapanos, 2013). Early literature from the field was strongly focused on the objective aspects of usability, as the ISO 9241-11 standard defines usability:

> *"the extend to which a product can be used by specific users to achieve specific goals with effectiveness, efficiency and satisfaction in a specified context of use*

Karapanos (2013) notes that effectiveness (the accuracy and completeness with which users achieve a goal) and efficiency (resources, usually time, users spend to achieve a goal) represent the objective side of usability, whereas user satisfaction represents a more subjective side of usability. Kapanos (2013) cites several studies, conducted in different cultural backgrounds, that suggest subjective perceptions of usability are generally not correlated with objective measures and seem to measure something else than merely effectiveness and efficiency. The author also concludes:

> *These findings also suggested that users' experiences with products go beyond the effectiveness and efficiency in product usage. Consequently, the field of Human-Computer Interaction quested for new concepts, measures and methods in capturing a more holistic view on user experience. This development has gone hand-in-hand with a shift in the contexts of study, from professional to personal and social, and in the design paradigm from product to experience design.*

This led to the rise of User Experience as an important academic discipline and professional field that also extrapolates Human Computer Interaction. ISO 9241-210 defines user experience as *a person's perceptions and responses resulting from the use and/or anticipated use of a product, system or service.* Norman and Nielsen (2020) define user experience as encompassing all aspects of the end-user's interaction with the company, its services, and its products.

Hartson & Pyla (2012) highlight that user experience is the totality of the effect or effects felt (experienced) internally by a user as a result of interaction with, and the usage of, a system, device or product. The authors stress that the term user experience refers to what the user feels internally, including the effects of usability, usefulness and emotional impact. Therefore, the *user experience* cannot be designed, as it resides *internally* in the user. Because the user

experience is the result of the user's interaction with the product and the context of this usage, only the artifacts can be designed, not the experience itself.

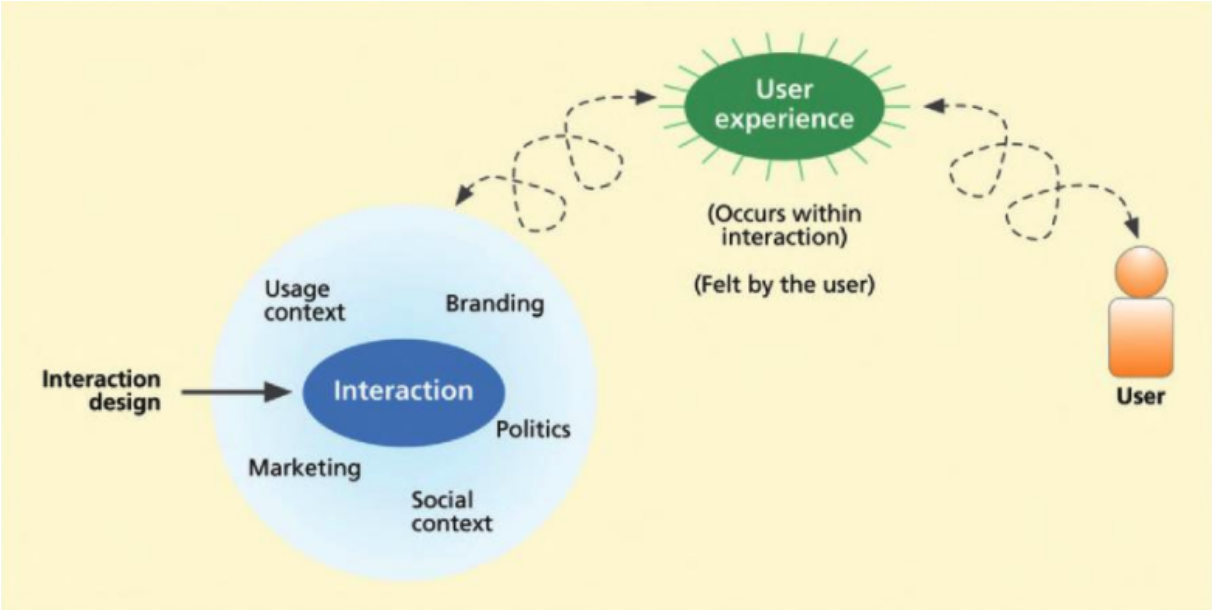


Figure 12 - User experience occurrence in time (Authors: Hartson & Pyla, 2020)

This is an important distinction because it shifts the focus from the uncontrollable feelings, emotions and experiences inside the user's mind and places it in the artifacts and environments that designers have control over.

Therefore, given the digital transformation phenomenon mentioned in the previous chapter but also the challenges that have emerged from it, if organizations and teams seek to provide exceptional user experiences in their products, they must define not only how their products will look - as described in the section 2.1 - but also how their products will *behave*, and that is one of the reasons why Design Systems have been established.

2.4 WEB DEVELOPMENT

It is impossible to write about User Experience without mentioning the Internet or its most popular manifestation: the World Wide Web.

In the web's infancy, during the 90's and early 00's, web pages would initially allow very limited interaction and visual possibilities, being compared to *glorified* Word documents (Vesselov & Davis, 2019). Pages were mostly published documents that did not allow for

much interactivity. This period of the World Wide Web was called Web 1.0 (Choudhury, 2014).

The next 'version' of the Web - Web 2.0 - a term popularized and explained by Tim O'Reilly (2009), described the web as a platform of communication and continuous delivery of solutions. It meant users now engaged in the production of the content, not just its consumption, and were constantly connected to one another.

Finally, Tim Bernards-Lee coined the expression 'Semantic Web', which other authors associated with a version of the Web: Web 3.0 or the *executable web* (Choudhury, 2014). The main idea of Web 3.0 is that now data can be structured, made available and linked in order to allow more effective discovery, automation, integration, and reuse across various applications.

WEB 1.0	**WEB 2.0**	**WEB 3.0**
1996 – 2004	2004 -2016	2016+
The Hypertext Web	The Social Web	The Semantic Web
Read Only	Read and Write Web	Executable Web
Millions of User	Billions of User	Trillions+ of Users
Echo System	Participation and Interaction	Understanding self
One Directional	Bi-Directional	Multi-user Virtual environment
Companies Publish Content	People Publish Content	People build application through which people interact and publish content.
Static content.	Dynamic content.	Web 3.0 is curiously undefined. AI and 3D,The web learning

Table 2 - Comparison of Web 1.0, Web 2.0 and Web 3.0 (Choudhury, 2014)

The data that allows for this cross-communication between applications and users is made available through an application programming interface, or an API. Mozilla Foundation (2020) defines API as constructs made available in programming languages to allow

developers to create complex functionality more easily, since it abstracts more complex code away, providing some easier syntax to use in its place. In its documentation, Mozilla (the foundation that develops Firefox browser) provides the following example to explain APIs:

> *As a real-world example, think about the electricity supply in your house, apartment, or other dwellings. If you want to use an appliance in your house, you plug it into a plug socket and it works. You don't try to wire it directly into the power supply — to do so would be really inefficient and, if you are not an electrician, difficult and dangerous to attempt. In the same way, if you want to say, program some 3D graphics, it is a lot easier to do it using an API written in a higher-level language such as JavaScript or Python, rather than try to directly write low level code (say C or C++) that directly controls the computer's GPU or other graphics functions.*

This is an important understanding for Design Systems, because this new architecture of the Web 3.0, followed by technological advances in internet browsers and in the JavaScript programming language allowed for a loosely coupling of systems (O'Reilly, 2009) allowed for much of the software to run on client (the browser or an app) rather than the server.

2.4.1 CLIENT VS. SERVER-SIDE RENDERING

Part of the reason why the web has become such a competent platform for applications is because of current client-side technologies. When the word 'client' is used, it means the web browser, and browsers render HTML documents (web pages) that users read and interact with, whereas 'server' is the computer that stores and processes the application that is being served.

Server-side rendering technologies are conceived to render HTML code on the server (Beke, 2018). It means that the server fetches the data from the database, creates the HTML that will be used to render the page, and sends it back to the client. Examples of server-side programming languages include Java, PHP, C+ and Node.js

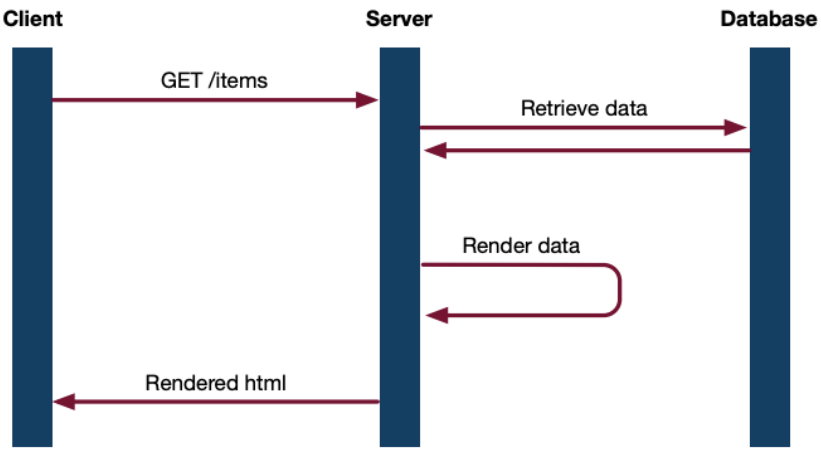


Figure 13 - Server-side rendering example schematics (Beke, 2018)

However, in modern web development, the data is not rendered in the server, rather it is rendered in the client (the browser). Client-side programming, also called front-end development, is based on three concerns, each one with its language (W3C, 2014):

- Content: HTML, used for organizing a page's content
- Presentation: CSS, used for defining content presentation
- Behaviour: JavaScript (ECMAscript), used for defining how the content behaves and interacts with the user

Beke (2018) explains that in this architecture, the server exposes its data through an API, the client fetches the data, fetches the raw HTML, CSS and JavaScript code that will generate the page the user will see, combines it and then renders it.

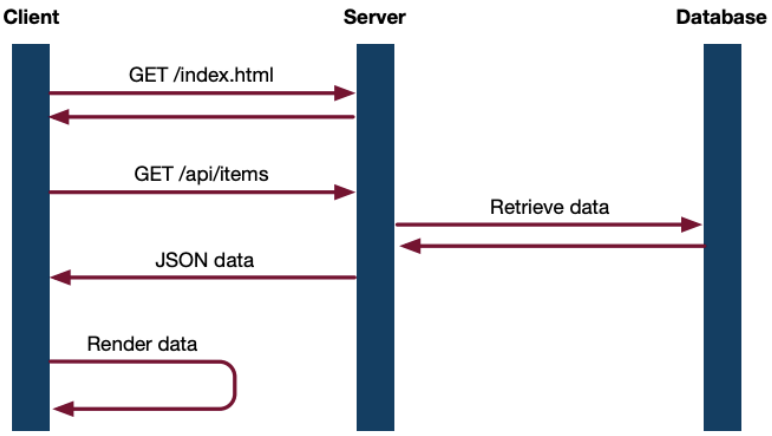


Figure 14 - Client-side rendering example schematics (Beke, 2018)

Therefore, much of the processing previously done in the server now happens in the client-side (in the user's browser). And because much of the logic is interpreted and rendered locally in the client, the user experience tends to be a lot faster and more fluid, as there's no need for the user to perform a request, wait for the server to get the data, render the page and send it to the client. Now, the presentation code is rendered in the client, and only the necessary data is transferred between the client and the server. This is usually done with the help of client-side frameworks such as React or Angular.

In practical terms, it means the client-side (front-end) of web applications can now be totally decoupled from the server-side (back-end). This led to web app interfaces becoming much more engaging, responsive, user friendly and also easier for front-end developers to create them.

As front-end development teams grew in order to meet the needs of product development, the problems mentioned in chapter 1 such as multiple implementations of the same components, visual discrepancies and need for a centralized source of information (single source of truth) became evident.

These single sources of truth are usually centralized code repository platforms (like GitHub) and code distribution platforms (such as GitHub Packages). These platforms allow teams to work distributed but have their final output centralized in one place, and also to control the way the end result is distributed.

The problems faced by companies in the scenario of multiple teams having to develop multiple different products, that end up generating multiple different experiences only reinforced the need for companies that create digital products to define not only how their products would look and behave, but also how they *should be built*.

The technological improvements that allowed for code to be centralized and also centrally distributed, plus the decoupling and advances in front-end technology made Design Systems be this instrument where this front-end knowledge is centralized and distributed.

2.5 FINAL CONSIDERATIONS

The theoretical background presented in this chapter aimed to explain what the context for this research was and to present the main concepts regarding Human Computer Interaction, User Experience and Web Development that led to the creation of the Design System as a tool for defining how digital experiences should look, behave and be built, given the context and the research problem presented in chapter 1.

The next encompasses the research methodology definition and further research on the topic of Design Systems, where the researcher collects and presents definition and data regarding the topic.

3 THE STUDY

As addressed in the introduction, the research objective of this study is to understand what comprises a good Design System.

As so, the main research questions of this study are:

RQ1: How can Designs Systems be defined?

RQ2: What are the common attributes of Design Systems?

RQ3: What are the most important elements of Designs Systems according to UX professionals and practitioners?

In order to answer the three research questions above, this research is divided into two main sections: the first section features a *literature review* that aimed mainly to compile a definition for Design Systems (RQ1). The second step features a c*omparative analysis* to define attributes of design systems, based on dataset and literature (RQ2), that will be used to formulate the questions for the survey. Finally, *an online survey* was conducted, and its results show what the most important elements of Designs Systems are according to UX professionals and practitioners (RQ3).

3.1 DESIGN SYSTEMS DEFINITION

In order to answer the first research question of this project, a thorough literature review was conducted on Design Systems in order to find and present definitions for the term. This literature review was adapted by the methodology used by Vial (2019) in a 5-step process:

1. Define the scope of the review;
2. Search the literature;
3. Select the final sample;
4. Analyse the corpus:
    a. Extract tags and keywords from each quote;

b. Group similar keywords to generate main concepts;

5. Present the findings;
6. Summarize the results in a new definition.

The scope of the literature review was to find authors who presented definitions for the term 'Design System'. However, this posed a challenge because 'Design Systems' is a very generic term that can be applied to several different areas of knowledge. In our preliminary research, it was perceived that the term 'Design System' or 'Design Systems' is also largely used in Engineering, which made searching the literature more difficult. In order to narrow down the amount of publication to be searched, the following criteria were used:

- Only articles from 2016 onwards
- Initially searched for terms "Design System" or "Design Systems"
- Then combined "Design System" or "Design Systems" with related terms: "HCI", "Human-computer Interaction", "UX", "User Experience", "UI", "User Interface"

Regarding the databases and the types of publication searched:

- Scientific databases for articles, books and conference papers: ScienceDirect, ACM Digital Library, Scopus, Scielo;
- Google Scholar for books, articles, papers and other resources;
- Google general search for blog posts and other website articles;
- Specific libraries of different universities: Aalto, Gothenburg, UFSC, Stanford, etc.

The results found are presented in the next section.

1.1.1 FINDINGS

This research yielded 39 analysed results, from which we were able to find 20 pieces of work that contained original definitions of Design Systems. From these 20, 4 were academic papers, 5 were blog posts, 8 were book (or e-book) chapters and 3 were other academic Master's thesis.

Following Vial's (2019) methodology, the definitions found in the literature review for 'Design Systems' were laid out in a table detailing their authors, the extracted quote and the kind of publication where it was found.

On a first round of analysis, each quote was coded with tags expressing their primary ideas. In a second round, these tags were then grouped by similarity and recoded, so that they could all be organized by each concept they presented.

Author and type	Quote	Concepts
Curtis, 2018: Blog post	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.	library components, team, code, design tool, cohesion, proucts
Hacq, 2018: blog post	A Design System is the single source of truth which groups all the elements that will allow the teams to design, realize and develop a product.	single source of truth, teams
Suarez, 2017: book	A design system is a collection of reusable components, guided by clear standards, that can be assembled together to build any number of applications.	reusable components, standards, build applications
Suarez, 2017: book	Design systems enable teams to build better products faster by making design reusable—reusability makes scale possible. This is the heart and primary value of design systems.	teams, build products, reusable
Stevens, 2020: book	Design system is a set of clearly defined guidelines that ensure a visual consistency across a product that allows for scalability	guidelines, visual consistency, scalability
Ruissalo, 2018: thesis	Design systems aim to salvage the aforementioned problems by linking designer style guides and programming component libraries together in a more systematic, practical manner, and focusing on the organisation-wide applicability.	style guide, component library, organization-wide
Huang (2019): thesis	...design system as a shared visual language for the team, which helps to avoid miscommunication in order to build better, faster and create more scalable products	shared visual language, team communication, scalability
Slifka 2020: article	A way of how to represent a graphical user interface style and behaviour	GUI style

Churchill, 2019: article https://dl.acm.org/doi/fullHtml/10.1145/3352681#R1	Design systems lay out the core elements and components of visual and interactive design for the devices we use every day—cellphones, tablets, laptops, desktops, and car dashboards, to name just a few. They offer guidance for layout, shape/form, color palettes, buttons, forms, tables, typography, user task flows, and can offer advice for content, branding, tone, and personality. Design systems enable designers and developers to create a consistent and cohesive product or suite of products.	components, visual design, guidance, consistency
Dukic 2020: thesis	A Design System is a single source of truth for designers and developers that evolves together with the product(s) over time, facilitates team work by reducing the layers of translation between design and development and reflects the organisation and its culture. It should include both the more abstract styles that represent the brand identity, the organisations' design principles and guides on best practices along with a well documented concrete representation of designs in code as components and patterns	single source of truth, teamwork, reflects organization/culture, styles, principles, documentation
MacDonald 2019: book	Design system is a single source of truth for shared parts and processes to build consistent products that is tailored to organizational needs and reflects the culture, team values, and visual language of an organization.	single source of truth, shared parts, consistency, reflect organization, visual language
Pyrhönen 2019: book	Design system is a living system of guidelines, reusable code and design assets, and tools that helps organizations deliver consistent, on-brand experiences at scale and over time.	living system, reusable, consistent, scalable
Vesselov & Davis (2019): book	A series of documented elements, components, and regions that include both design and front-end guidelines. The documentation contains live code examples, allowing cross-functional teams to easily reuse styles and components in several instances across an application. A design system also includes underlying design principles, rules, and guidelines that help a team build one or multiple products.	components, guidelines, documentation, code, principles, products
Kholmatova, 2017: book	*A design system is a set of interconnected patterns and shared practices coherently organized to serve the purpose of a digital product. Patterns are the repeating elements that we combine to create an interface: things like user flows, itneractions, buttons, text fields, icons, colors, typography, microcopy. Practices are how we choose to create, capture, share anduse those patterns, particularly when working in a team.*	patterns, practices, digital product
Punchev, 2019	*A design system is the single source of truth for digital products, and should build a place for every team to refer back to when they're creating anything to put into the world on a digital device*	single source of truth, team, digital device

Toman 2017: blog	*Style guide is a set of rules which defines the basics. You can find colors here, typography, brand, icons, etc. You can even find a grid here. This is the most abstract part of UI. Component library is storage for your components — articles, headers, galleries and many more. Each component is categorized, well-documented and responsive. The design system defines the principles relating to the way in which components should work together... how you can combine it all together.]*	rules, design tokens, component library, principles
Edelberg, 2020 Kilrain, 2020: artcile	*A design system is a set of both tangible items, and non-tangible deliverables that form a systematic language and architecture. Some of the tangible deliverables are branding guidelines, color palettes, button styles, patterns, and UI/UX components. Non-Tangible elements are about the brand values, beliefs, and best practices.*	systematic language, guidelines, components
Yew 2020: article	*A repository of reusable components that follow a set of shared design principles*	reusable, components, design principles
Vacchar, Maritan, 2019: blog	*A Design System is a systematic approach to product development — complete with guidelines, principles, philosophies, and code. It shines a spotlight on how a team designs, making it a fantastic tool for scaling your design practice, reducing the need for hand-off and promoting more collaboration.*	product development, guidelines, code, team, scale, collaboration
Stepanova, 2018: blog	*Design systems is a systematic approach to creating and maintaining consistent user interfaces which coherently communicate the brand values and empower user experience.*	systematic approach, consistent user interface

Table 3 – Literature review findings with definitions for Design System

The keywords were extracted and these main nine categories were identified:

Tag	Concepts
Component library	library component, component library, components, shared parts
Code assets	code artefacts, code
Centralized repository	single source of truth,living system, documentation
Consistency	cohesion, consistency, visual consistency, styleguide, consistent user interface

Scalability	reusable, systematic language, systematic approach, scalability, scalable, scale
Team-oriented	team, communication, organization-wide, team communication, teamwork, organization culture, reflect organization values, collaboration
Guideline	standard, guideline, guidance, principle, patterns, rules, design principles, practices
Product development	products, build products, digital products, digital device, product development
Visual identity	shared visual language, GUI style, visual design, styles, design token

Table 4 –Tags used to organize definition concepts

Below, each one of the categories is explained, both according to literature and the researcher's perceptions:

Centralized repository: When authors mention a 'single source of truth' they speak of the fact that Design Systems work as a central repository of information, and that information and assets contained there should be considered the latest, most up-to-date and correct information regarding the components in a Design System. It also means that once a change is made to the Design System, it automatically affects all products that rely on it (Hacq, 2018; Dukic, 2020; Punchev, 2019),.

Component library: Design Systems document the various components used in the system itself . These components are usually User Interface components, such as buttons, headings, etc., but they can also cover other pieces of information meant to be consumed from a single source, such as 'Design Tokens'. Authors also refer to it as a 'shared' because these components are meant to be shared between the various products that use this design system. (Ruissalo, 2018; Tomen, 2017).

Code assets: Besides containing documentation regarding the components specifications, Design Systems also provide the actual code libraries to be used by developers. It is often done by making the code assets of designs systems available in either public code repository services or code distribution services, such as GitHub and npm. (Curtis, 2018; Dukic, 2020; Pyrhönen, 2019).

Consistency: Because Design Systems can be shared by multiple products and updates pushed to the Design System can affect products instantly, this ensures the consistency of the components used across different products, thus creating coherent experiences (Stevens, 2020; Macdonald, 2019);

Scalability: Authors also define Design System as a scalability tool because it solves very common problems for software development teams. It prevents teams from redefining attributes already defined previously, and by allowing teams to reuse artifacts already created, hence scaling the speed of developing new products (Huang, 2019; Stevens, 2020);

Team-oriented: Authors also stress the importance of Design Systems being a tool used for team communication and collaboration. As Design Systems are available for the entire organization, they allow knowledge sharing and provide visibility to important information. It should be also stated that most Design Systems are created and maintained by internal teams, so the internal communication processes also contribute to keeping Design Systems updated (Vesselov & Davis, 2019; Kholmatova, 2017;);

Guidelines: Design Systems are also a centralized repository not only for UI components but also for documentation on guidelines, principles and rules developers and designers should follow to adhere to the organization's best practices and desired user experience, not to mention that the artifacts contained in the Design System (such as UI components) should also abide by these guidelines (Edelberg, 2020; Vacchar, 2019; Stevens, 2020; Pyrhönen, 2019);

Product development: Authors emphasize that a Design System is a tool created for and used primarily by software product development teams (Punchev, 2019; Vacchar, 2019);

Visual Identity: In its components and documentation, Design Systems should contain all the visual identity elements that should be used in the digital products of a given organization and apply them to the components the Design System provides or allow these pieces of information (Design Tokens) to be accessible (Dukic, 2020).

Based on the findings above, the researcher proposes the following definition for Design Systems:

> *Design Systems are centralized repositories used by product development teams to manage, distribute and communicate user interface assets, code,*

documentation, guidelines and principles in a scalable way in order to ensure user experience consistency.

3.2 DESIGN SYSTEMS ELEMENTS

Aiming at fulfilling the second objective of this research - to identify the most important elements of a design system - a survey was created.

In order to create the survey instrument, it was necessary to elicit the most common elements found in design systems and categorize them.

A similar research was previously conducted by Yew et al. (2020), where authors carried out a survey during Clarity 2018, a Design Systems conference. Amongst the questions asked in this survey was *what artifacts respondents thought should belong to a design system.* The responses were:

1. component library (87.4%)
2. style guide (85.9%)
3. design guidelines (62.8%)
4. content guidelines (50.86%)
5. "Other" (8.4%)

Although these results do not fully provide the answer to the question this research aims at answering, it provides one set of categories to be used for this survey.

Another research was conducted by Ribeiro (2020) as a master thesis project on Design Systems. Ribeiro initially analysed 28 publicly available design systems from Fortune Global 500 companies. From these 28, the researcher conducted a thematic analysis and identified *the elements that all effective design systems should contain: usage guidelines, colours, typography, grid or spacing, UI components, design patterns, design resources, and code resources.*

The author then synthesized these elements into 7 themes that were later defined by him as the *elements that make an effective design system*:

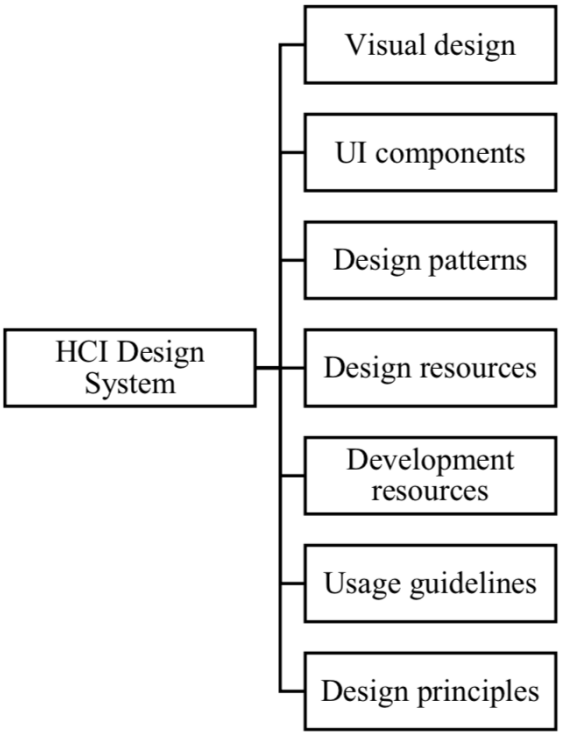


Figure 15 – Design system main elements (Ribeiro, 2020)

Ribeiro (2020) then explained each of the elements according to his interviews and briefly explained what contents each element could contain. The table above summarizes Ribeiro's findings:

Category (element)	Content
Visual design	• Smallest units used to build components: • colors, typography, and shapes • brand colors, color palettes, and color schemes. • Typography, font families and styles • Shape styles, icons • Spacing or grid systems • Logos, photos, or illustration.

UI Components	• Reusable elements that can be put together to create more extensive experiences. • Made of colors, typography, shapes, and assets • Can be nested in other components, and visual or interactive design patterns. • Should have built in accessibility features as well as responsive behaviors. • Component documentation should include all component states, configuration variables, and examples. • Examples: buttons, tooltips, modals, input fields, and navigation
Design patterns	• Design patterns provide solutions to common problems in a given context. • Visual design patterns describe ways to present information spatially on a page or section of a page. • Interactive patterns describe user flows and how the user interacts with the product, feature or component. • Patterns can support graphical interfaces as well as voice interfaces. • Examples: login flow, password reset flow, shopping cart, loading states, error states, or add users
Design resources	• Design resources cover design tools, design documentation, and research findings. • Figma files, Sketch files or other UI Kits
Development resources	• Development resources should include working code, “how to start guides”, and code formatting guidelines.
Usage guidelines	• Usage guidelines are instructions to people that consume or contribute to the design system. • They should be based on user centered design and they include brand guidelines, instructions on how to create and write content, and tone of voice • Usage guidelines provide instructions on how to use the design system elements such as colors, typography and shapes, components, and patterns • Also covers naming conventions, how to consider accessibility, and how to handle responsiveness.
Design principles	• Guiding factors that help teams solve problems in a consistent way, reducing discussions around design decisions

Table 5 – Categories of Design Systems contents (Ribeiro, 2020)

Although very insightful, Ribeiro's examples for the contents of each of the outlined categories were not as specific as necessary for this research, and more datasets had to be analysed.

One of the biggest repositories of Design Systems publicly available is Adele, a project created by UxPin (a technology company that develops a prototyping app under the same name). As of December 2020, Adele listed over 100 publicly available Design Systems.

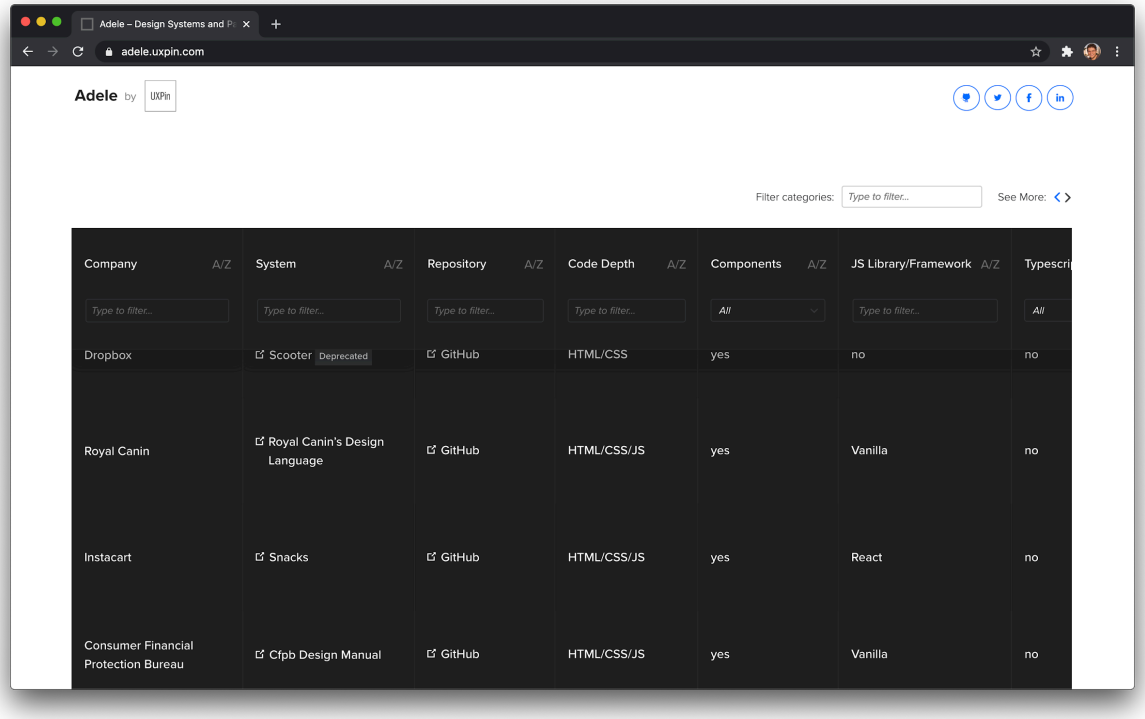


Figure 16 – Screenshot of Adele Design System Repo (UXPin, 2020)

Adele uses a set of 31 criteria to analyse Design Systems:

- SystemRepository
- Code Depth
- UI Components
- JS Framework?
- TypeScript
- Web Components
- Tests
- Linter
- CSS
- CSS in JS
- Bundle Manager
- Design Tokens
- UI Kit
- Brand Guidelines
- Colour Pallete
- Colour Naming
- Contrast Analysis
- Typography
- Icons
- Space/Grid

- Illustration
- Data Visualization
- Animation
- Voice & Tone
- Accessibility
- Guidelines
- Design Principles
- Documentation Website
- Code Documentation
- Storybook
- Distribution

The criteria used by Adele is more focused, itemized and direct, not to mention it has been applied to an analysis of over 100 design systems.

To generate the actual elements to be used in the research, the researcher attempted to match the criteria used by Adele and cross-reference it to Ribeiro's 7 categories for Design System elements

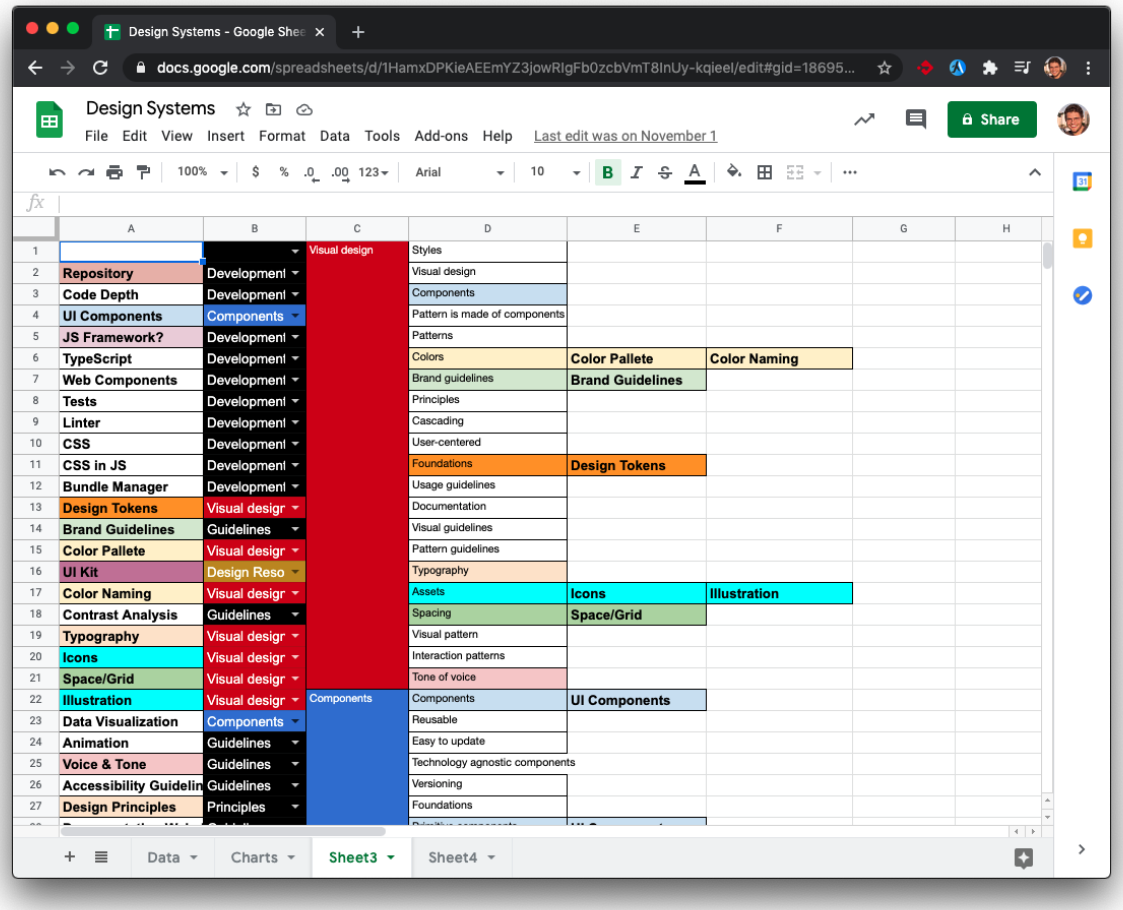


Figure 17 - researcher categorizing Design System elements

Once the elements were organized according to those seven categories by Ribeiro (2020), they were transferred to Lime Survey, a web application for survey creation.

The survey created consisted of two blocks: one for user segmentation and another for eliciting importance of Design System elements.

In the first block, four questions were asked:

1. Where are you from?
2. Are you familiar with Design Systems?
3. What option(s) best describes your role in your organization?
4. How many years of experience do you have in the role?

In the second block, the defined Design System elements were grouped in the seven categories and laid out in a 3-point scale with the following values: Not Important; Somewhat Important; Very Important.

Because the research goal is to elicit the most important elements of Design Systems, the researcher opted for a simpler scale than Likert's 5-point scale.

Once ready, the questionnaire was then pretested. According to Hilton (2017), pretesting is a method of checking that questions work as intended and are understood by those individuals who are likely to respond to them, with the ultimate capacity to reduce sampling error and increase questionnaire response rates. Because this survey was planned to be answered online and unattended by the researcher, it was critical that it is well-understood by respondents.

Three designers accepted to participate in the pre-test, which was conducted synchronously over Google Meet, an online conference service. The users were introduced to the research, and the researcher asked them to voice their thoughts and opinions about the survey as they progressed through the questionnaire. Respondents took an average of 10 minutes to complete the survey, and the need for several improvements were raised. The table below lists the problems found, as well as the measures taken to address each of the problems found.

Problem found	**Measure taken**

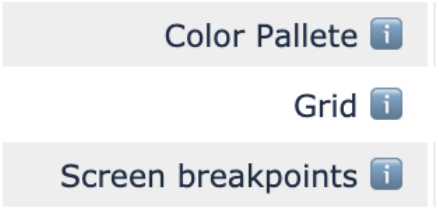
Survey objective was not clear at the beginning of the questionnaire	Research objective was detailed in the introduction block of the research
No context was given to what context the Design System in the survey would be used for	The following text was added at the beginning of the questionnaire: "Imagine you are starting to create a new Design System for your current organization, or that you are choosing an existing Design System to be used in your current context of work."
UI component elements and Pattern elements were not clear or easily understandable	Cues were added with examples of UI elements: 
There was no neutral/negative option	A fourth option was added: "I don't know or does not apply"
The difference between Guideline and Principle was not clearly understood	The difference between Guidelines and Principles was added to the survey as understood by the researcher

Table 6 – Summary of pre-testing problems found and solutions applied

After the corrections were applied, the final questionnaire was configured in Lime Survey according to the descriptions and options below. Every option was accompanied by a scale, where the user could choose between: **Not important; somewhat important; very important; I don't know or does not apply.** Also, an open-ended question was added at the end of every question, so users could add more options that they deemed important but that were not listed. Figure XX shows an example of the final question:

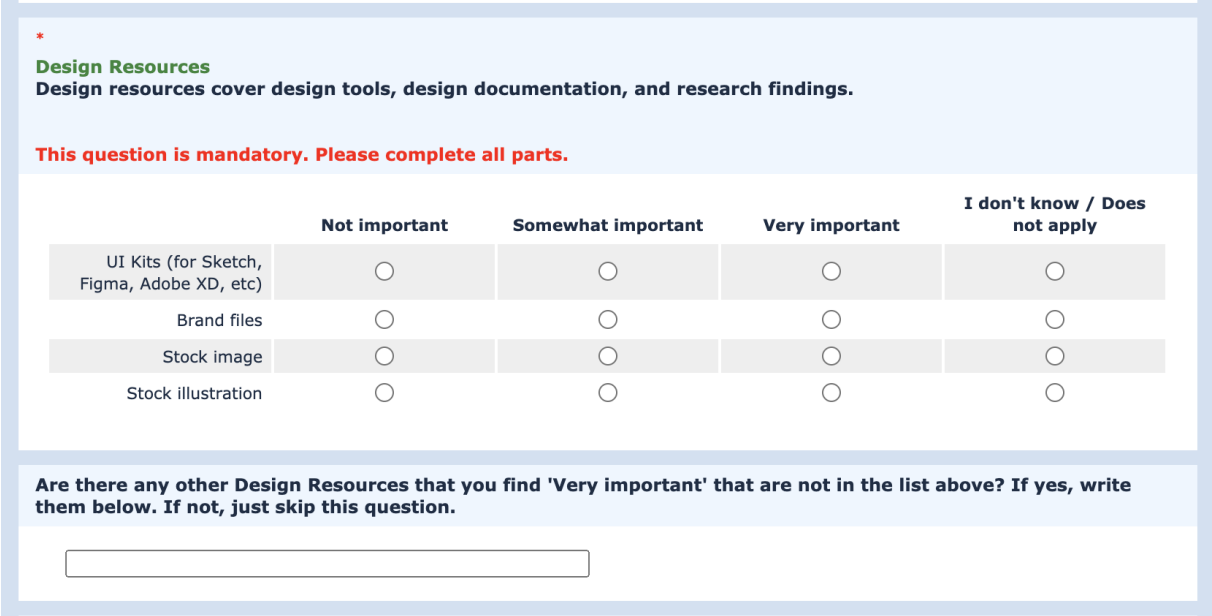


Figure 18 - Example of question and possible answers in the questionnaire, plus example of open-ended question for other options

The following table summarizes the options provided in the survey, the criteria in which they were grouped, as well as the description text used for each of the questions:

Criteria grouping and explanation	Options for each criteria
Visual Design (Design Tokens) Design tokens are the smallest visual design units used to build components. Visual design elements are informed by user centered design and include colors, typography, and shapes. Colors cover brand colors, color palettes, and color schemes. Typography covers all the different font families and styles. Shapes cover shape styles, icons, and spacing or grid systems to layout elements spatially on a page.	• Colour Pallete • Grid • Screen breakpoints • Shapes and radius • Spacing • Typography
UI Components UI components are usually the smaller user interface element that the user interacts with. They are reusable elements that can be put together to create more extensive experiences. They are made of colors, typography, shapes, and assets, and can be nested in other components, and visual or interactive design patterns. Components should have built in accessibility features as well as responsive behaviors. Component documentation should include all the component states, configuration variables, and examples. Examples of components are buttons, tooltips, modals, input fields, etc.	• Accordions • Alerts • Breadcrumb • Buttons • Chip/Pill/Badge • Carousel/Slider • Dropdowns • Form elements (inputs, checkboxes, selects, radios) • Icons

	• Modal • Pagination • Progress • Spinners/Loaders • Tables • Tabs • Tooltips/Popovers
Patterns Design patterns provide solutions to common problems in a given context. Design patterns can be visual or interactive. Visual design patterns describe ways to present information spatially on a page or section of a page. Interactive patterns describe user flows and how the user interacts with the product, feature or component. Patterns can support graphical interfaces as well as voice interfaces.	• Page layouts • Form validation • Navigation (Navbars) • Data grids • Cards • Data visualization
Design resources Design resources cover design tools, design documentation, and research findings.	• UI Kits (for Sketch, Figma, Adobe XD, etc) • Brand files • Stock image • Stock illustration
Development resources Development resources should include working code, “how to start guides”, and code formatting guidelines.	• Code Repository • Code Documentation/Samples • Integrated JS Library/Framework • Extensible CSS styles (SASS, Less, etc) • Distributed/Installable packages (npm)
Guidelines Guidelines are instructions, or specific set of rules to be followed by people that consume or contribute to the design system. They are lower-level, more specific than principles. Guidelines should provide do's and don'ts and also examples on several areas such as: brand guidelines, how to create and write content, tone of voice, how to combine elements such as colors, typography and shapes, components, and patterns, how to consider accessibility, how to handle responsiveness, etc.	• Accessibility guidelines • Tone of voice • Brand guidelines • Coding guidelines • Animation guidelines • Responsiveness guidelines

Design principles Design principles are very broad statements that provide advice on design issues and problems, or high-level ideas and metaphors that were used to conceive a design system. While principles are more generic, abstract and open to interpretation, guidelines and rules are more concrete, specific and focused on the application. Eg.: "Design to avoid errors" (Principle) -> “Make all unavailable menu choices gray and not selectable.” (Guideline)	• Design principles • Coding principles

Table 7 - Summary of Design System Elements

1.1.2 FINDINGS

The survey received 42 responses between November 16th and December 2nd of 2020. From these responses, 20 were incomplete. For this analysis, only the 22 valid responses were considered.

There were responses from Brazil, Croatia, Serbia, Argentina, Czech Republic, Albania, Spain, Ukraine and Germany. However, out of the 22 responses, 12 were from Brazil.

Almost 91% of the respondents said they were familiar with Design Systems. Also, 50% of the respondents were developers, 40% were designers and the remaining 10% had management roles

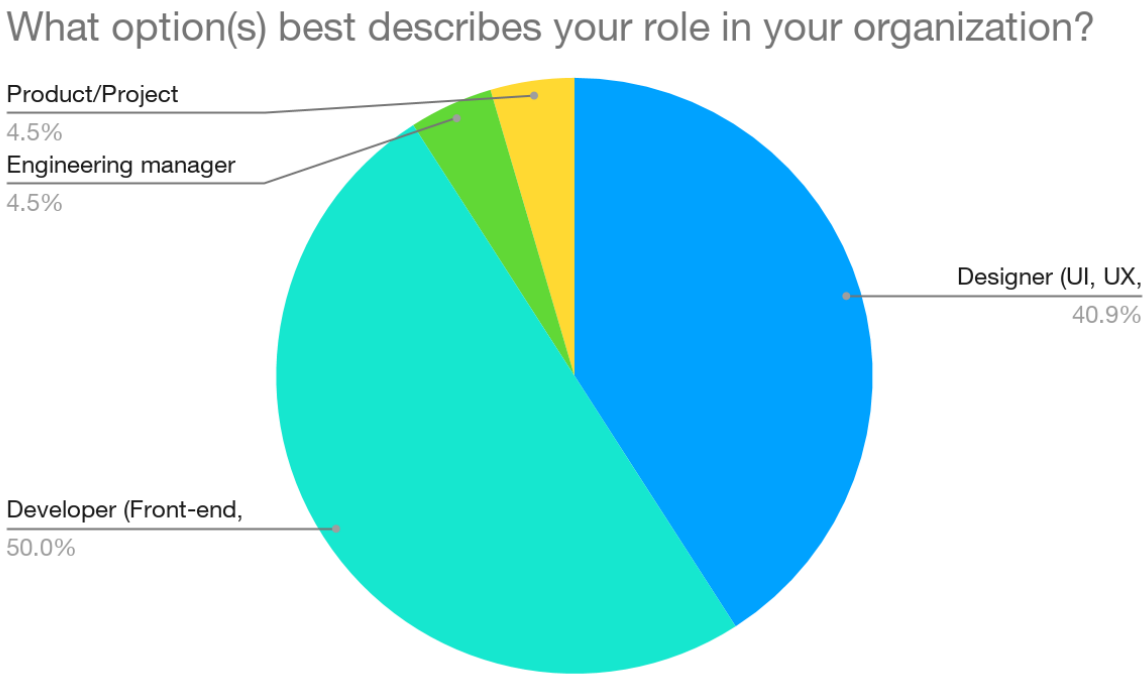


Figure 19 – Chart of respondents' roles in their organizations

45% of respondents had over 11 years of experience in their declared roles, whereas the remaining had either less than 5 years or between 6 and 10 years of experience in the field.

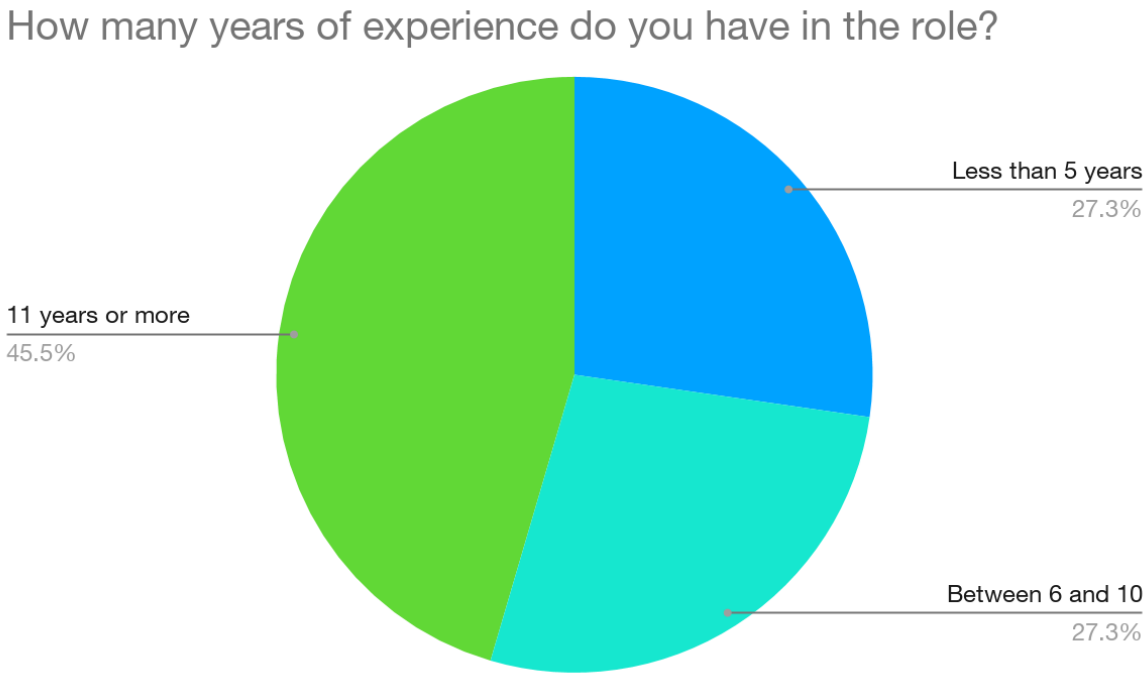


Figure 20 – Chart of respondents'experience with the research topic

In order to be able to analyse the answers, a scoring system was established to allow sorting the results by importance. Each option of the scale provided to respondents was assigned a value, from zero to three. To calculate the score of each Design System element, the value assigned was multiplied by the count of responses each element received, then summed up, and then multiplied by a fractional number of the maximum possible score each option could have received. The formulae below express these calculations:

Label	Count of responses checked as Very important (V)	Count of responses checked as Somewhat important (S)	Count of responses checked as Not important (N)	Count of responses checked as I don't know or does not apply (Z)
Value	3	2	1	0

Table 8 – Scoring system numeric values

$$Score = [(V \times 3) + (S \times 2) + (N \times 1) + (Z \times 0)] \times \frac{100}{Maximum\ Score}$$

$$Maximum\ Score = Number\ of\ Respondents \times Highest\ value = 22 \times 3 = 66$$

The numbers were rounded up to show only one decimal. The totality of the results is shown in the table below, ordering the results from highest to lowest score:

Element	Category	Score	V	S	N	Z
Buttons	UI Components	98.5	95%	5%	0%	0%
Form elements (inputs, checkboxes, selects, radios)	UI Components	95.5	86%	14%	0%	0%
Page layouts	Patterns	95.5	86%	14%	0%	0%
Form validation	Patterns	95.5	86%	14%	0%	0%
Navigation (Navbars)	Patterns	93.9	82%	18%	0%	0%
Coding principles	Principles	92.4	82%	14%	5%	0%
Color Pallete	Visual Design	90.9	77%	18%	5%	0%
Dropdowns	UI Components	90.9	73%	27%	0%	0%
Design principles	Principles	90.9	77%	18%	5%	0%
Typography	Visual Design	89.4	77%	18%	0%	5%
Alerts	UI Components	89.4	73%	23%	5%	0%
Icons	UI Components	89.4	68%	32%	0%	0%
Code Repository	Development Resources	89.4	86%	5%	0%	9%
Code Documentation/Samples	Development Resources	89.4	86%	5%	0%	9%
Grid	Visual Design	87.9	73%	18%	9%	0%
Spacing	Visual Design	87.9	73%	23%	0%	5%
Modal	UI Components	87.9	64%	36%	0%	0%
React, Angular, etc	Development Resources	87.9	82%	9%	0%	9%
Spinners/Loaders	UI Components	86.4	59%	41%	0%	0%
Distributed/Installable packages (npm)	Development Resources	86.4	77%	14%	0%	9%
Tabs	UI Components	84.8	59%	36%	5%	0%
Tooltips/Popovers	UI Components	84.8	55%	45%	0%	0%
Cards	Patterns	84.8	59%	36%	5%	0%
Screen breakpoints	Visual Design	83.3	64%	27%	5%	5%
Tables	UI Components	83.3	59%	32%	9%	0%
Data grids	Patterns	81.8	50%	45%	5%	0%
Data visualization	Patterns	81.8	50%	45%	5%	0%
Accessibility guidelines	Guidelines	80.3	64%	23%	5%	9%
Coding guidelines	Guidelines	80.3	64%	23%	5%	9%
Progress	UI Components	78.8	41%	55%	5%	0%

Extensible CSS styles (SASS, Less, etc)	Development Resources	78.8	59%	27%	5%	9%
Responsiveness guidelines	Guidelines	78.8	64%	23%	0%	14%
UI Kits (for Sketch, Figma, Adobe XD, etc)	Design Resources	77.3	55%	32%	5%	9%
Shapes and radius	Visual Design	75.8	41%	50%	5%	5%
Pagination	UI Components	74.2	41%	45%	9%	5%
Chip/Pill/Badge	UI Components	72.7	32%	55%	14%	0%
Brand guidelines	Guidelines	72.7	41%	45%	5%	9%
Breadcrumb	UI Components	66.7	27%	50%	18%	5%
Brand files	Design Resources	66.7	45%	27%	9%	18%
Accordions	UI Components	65.2	23%	59%	9%	9%
Tone of voice	Guidelines	65.2	32%	45%	9%	14%
Animation guidelines	Guidelines	65.2	32%	45%	9%	14%
Carousel/Slider	UI Components	59.1	14%	55%	27%	5%
Stock image	Design Resources	57.6	27%	32%	27%	14%
Stock illustration	Design Resources	56.1	18%	45%	23%	14%

Table 9 – Summary of survey results

In a quick statistical analysis, the median value of the score dataset is 84.8, with an average score of 81.6.

VISUAL DESIGN

Visual Design elements had an average score of 85.9. All Visual Design elements had a high score (above 80), with over 50% of respondents saying these elements are very important, except 'Shapes and radius', with a score of 75.8. There was no significant difference of attributed scores between designers and developers.

One respondent suggested adding 'z-index' values as a Design Token.

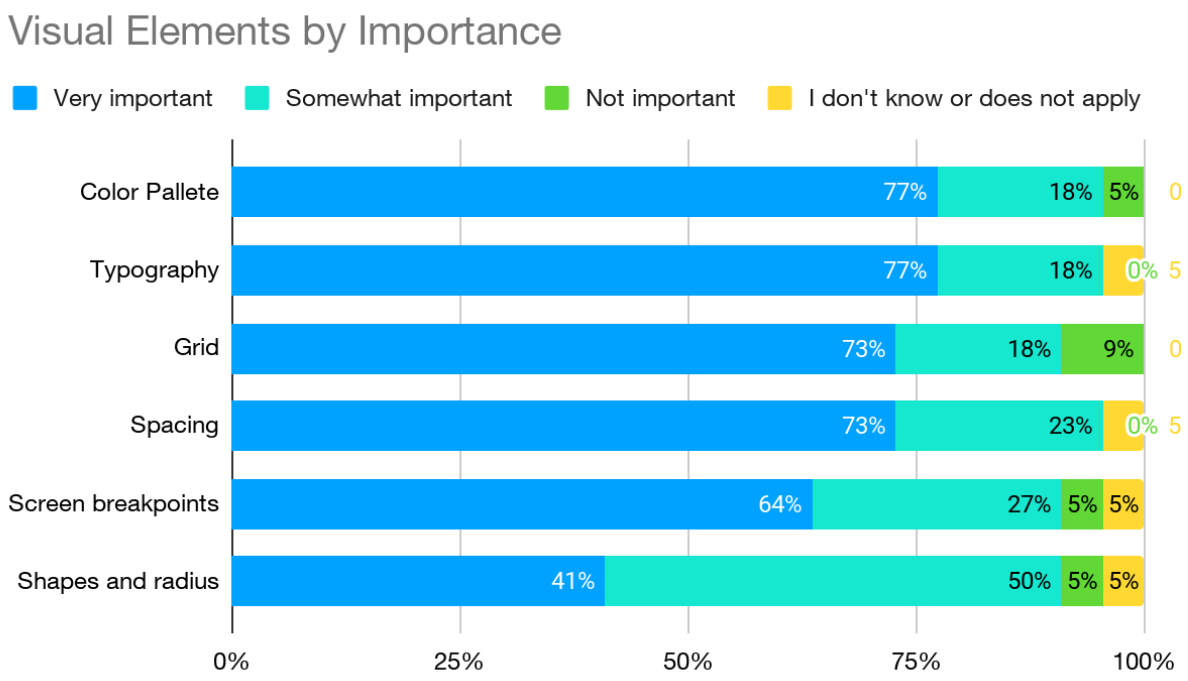


Figure 21 – Visual Elements by Importance

UI Elements

This category had the most elements to be analysed. It also turned out to be the one with the biggest difference in the number of respondents assigning 'very important' values: whereas buttons were said to be 'very important' by 95% of respondents, carousels were said to be very important by only 14% of respondents. There was also no significant difference in calculated scores between designers and developers for this category.

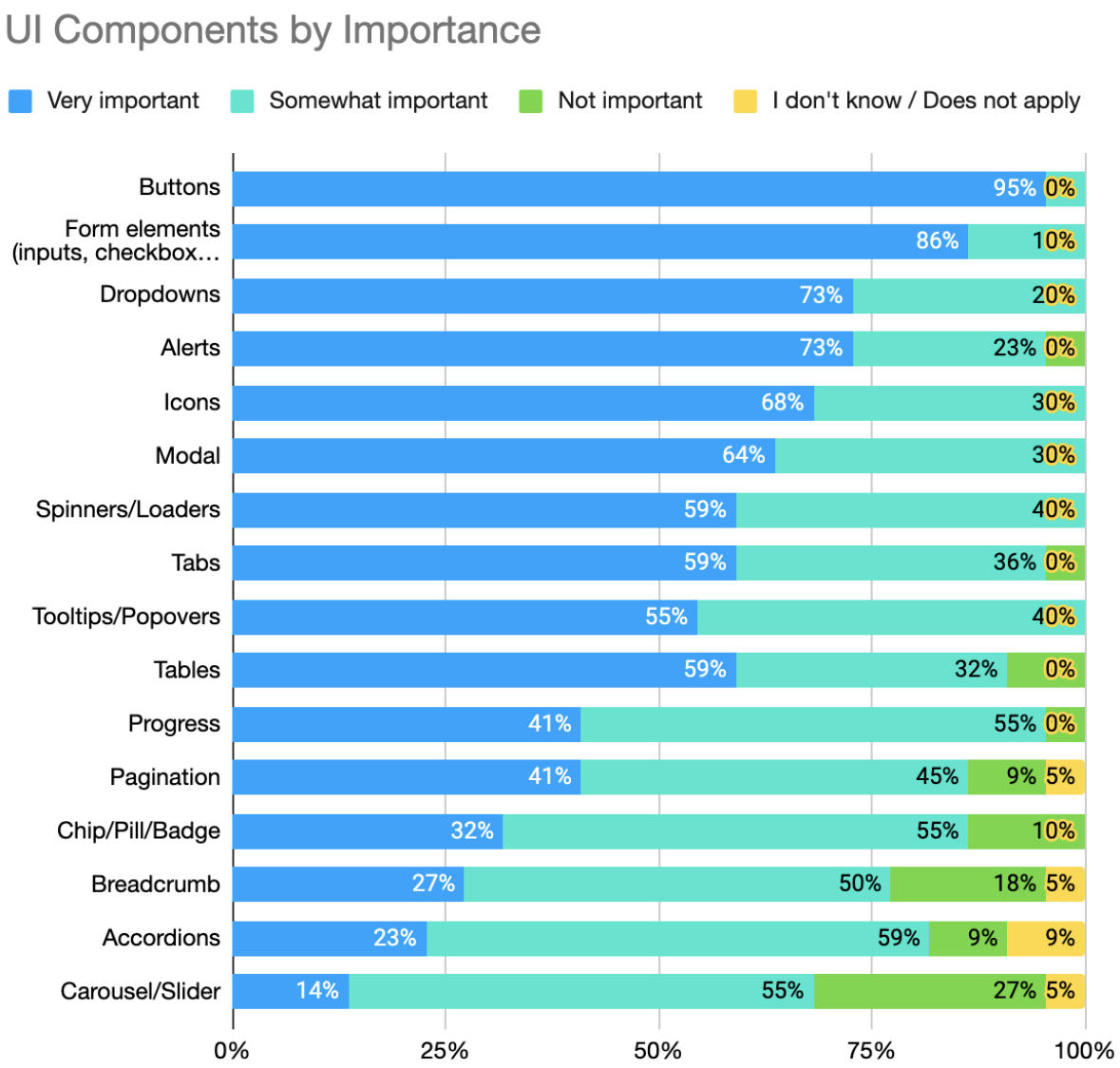


Figure 22 – UI Elements by importance

PATTERNS

Over 80% of respondents said that page layouts, form validation and navigation bars were very important in a Design System. There was also no major variance between answers given by designers and by developers.

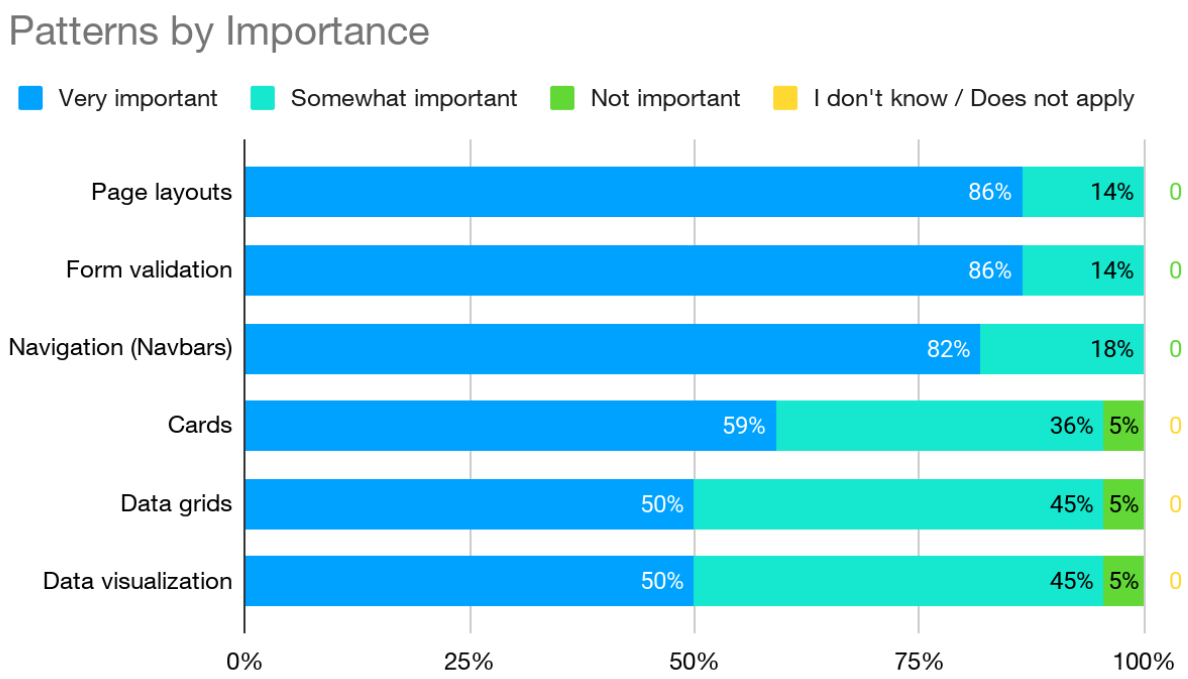


Figure 23 – UI Patterns by importance

DESIGN RESOURCES

This was the category with the lowest average score amongst all categories: 64.4, well below other categories. It is also interesting to note the difference between the average score given by designers versus the one given by developers: whereas the average score given by designers for this category was 75, the ones given by developers was 53.78. Through these results we can conclude, as expected, that designers give higher importance to design resources than developers.

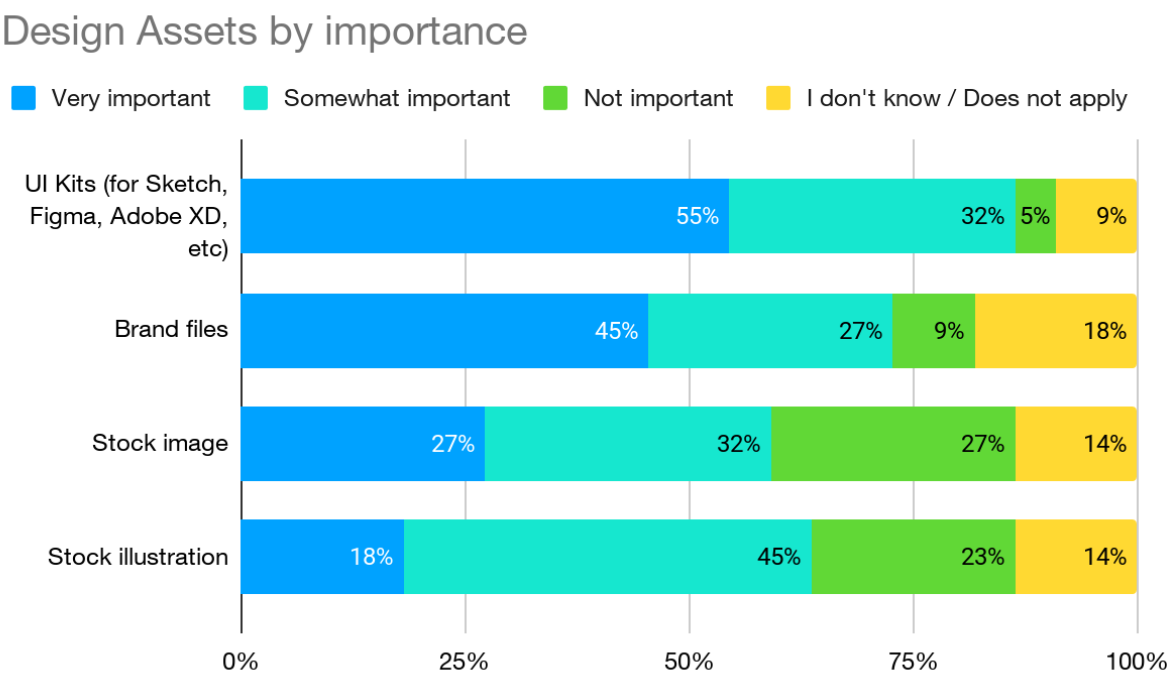


Figure 24 – Design Assets by importance

DEVELOPER RESOURCES

The same phenomenon observed in the Design Resources can be seen here: developers give more importance to developer resources than designers do. The average importance score given by developers was 93.93, whereas by developers it was 74.81. The main difference here is the higher base score: the lowest score in the previous section was 53, whereas here it was 74. Therefore we can conclude that even though each audience perceives the importance of Design Resources and Development Resources differently, designers consider both to be important, whereas developers consider Design Resources to be less important.

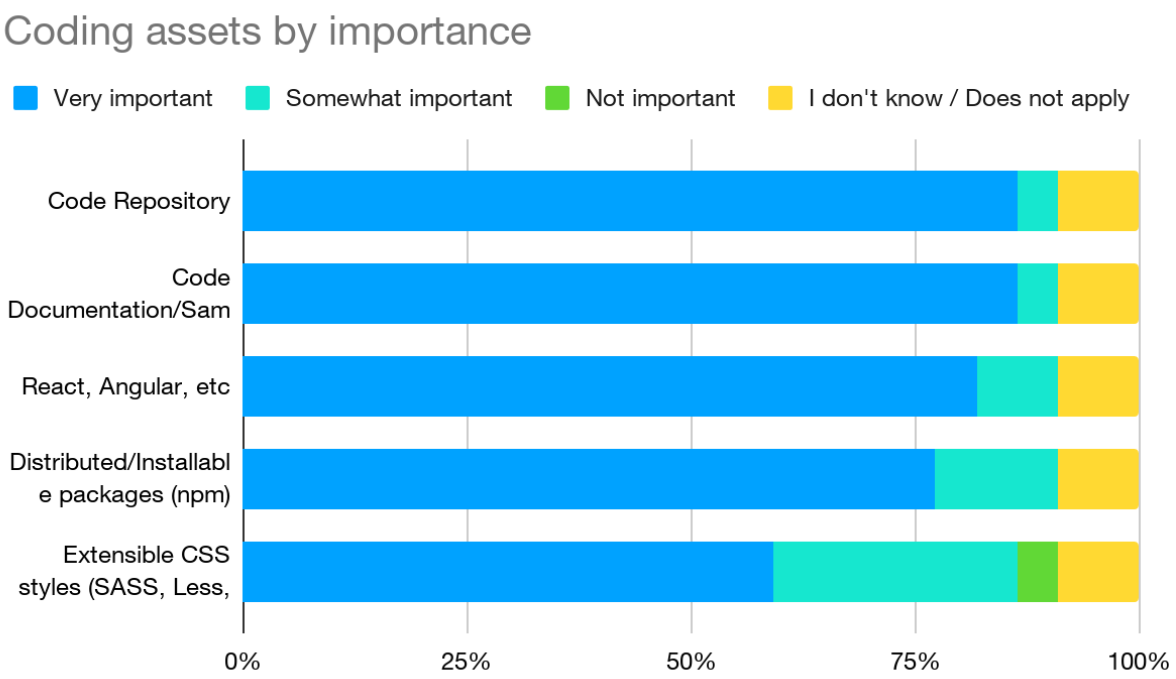


Figure 25 - Developer resources by importance

GUIDELINES

Guideline elements had an average score of 73.7, being Accessibility, Coding and Responsiveness guidelines, the most important guidelines assessed. Here it was also found Guidelines were perceived as 20% more important for Designers (average 85.18 score) than for developers (63.13).

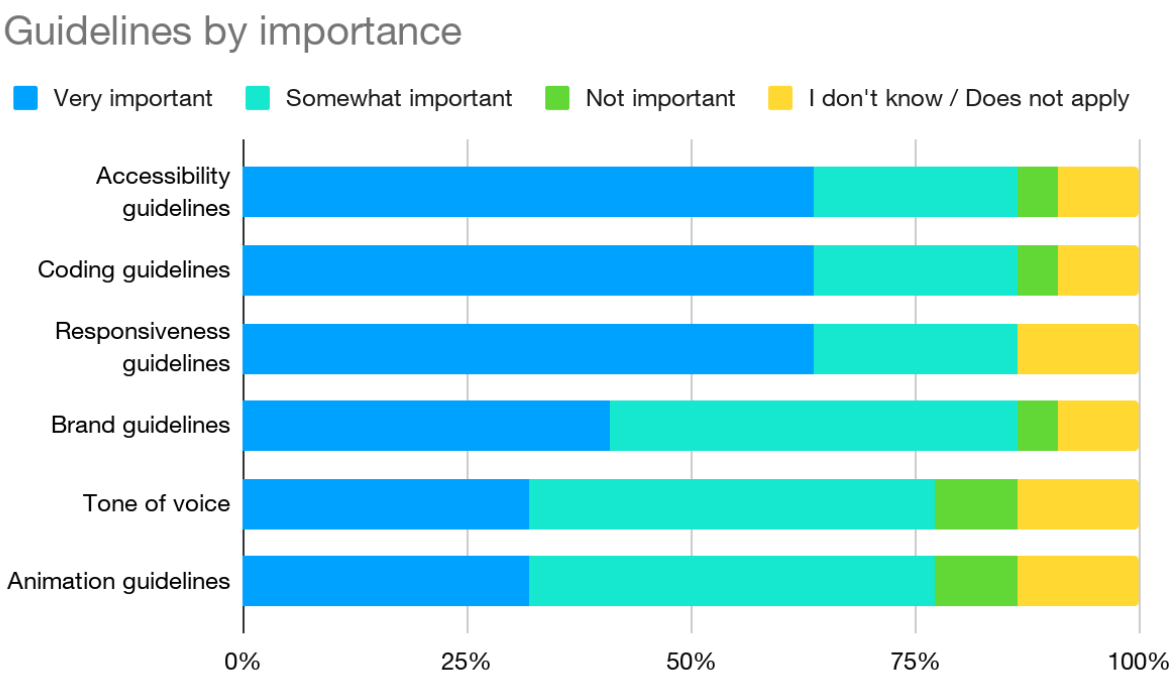


Figure 26 – Guidelines by importance

PRINCIPLES

This category had the actual highest average score for a category: 91.7, with no significant difference between answers given by designers and developers. The researcher believes that a possible reason why this category had the highest average score is because the elements contained very broad and generic terms, which led respondents to understand them differently.

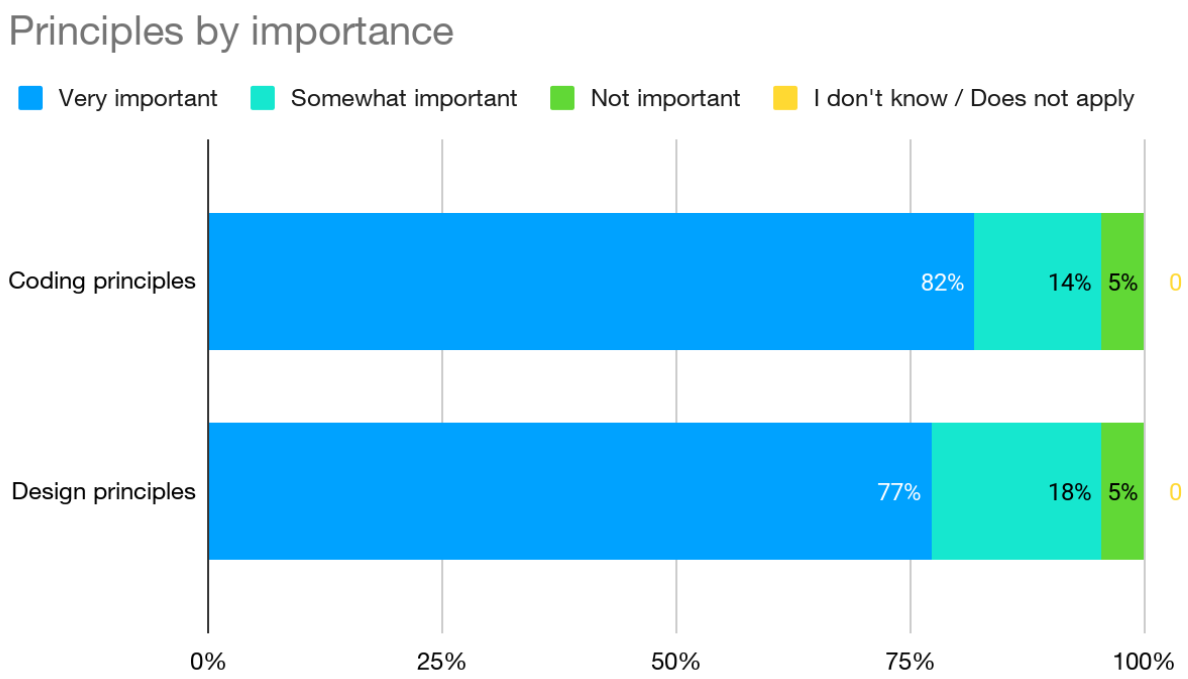


Figure 27 – Principles by importance

3.3 FINAL CONSIDERATIONS

After analysing the results, these average scores were found for each category:

Category	Average score
Principles	91.7
Patterns	88.9
Development Resources	86.4
Visual Design	85.9
UI Components	81.7
Guidelines	73.7
Design Resources	64.4

Figure 28 – Average Score for each category

A final narrowed down list of the most important elements will differ on the criteria, if either analysed by score or by percentage of 'Very important' responses. If analysed by percentage, the following list represents the most important elements of a Design System, assuming all elements with a 'Very important' percentage over 70%.

Element	Category	Score	Very important
Buttons	UI Components	98.5	95%
Form elements (inputs, checkboxes, selects, radios)	UI Components	95.5	86%
Page layouts	Patterns	95.5	86%
Form validation	Patterns	95.5	86%
Code Repository	Development Resources	89.4	86%
Code Documentation/Samples	Development Resources	89.4	86%
Navigation (Navbars)	Patterns	93.9	82%
Coding principles	Principles	92.4	82%
React, Angular, etc	Development Resources	87.9	82%
Color Pallete	Visual Design	90.9	77%
Design principles	Principles	90.9	77%
Typography	Visual Design	89.4	77%
Distributed/Installable packages (npm)	Development Resources	86.4	77%
Dropdowns	UI Components	90.9	73%
Alerts	UI Components	89.4	73%
Grid	Visual Design	87.9	73%
Spacing	Visual Design	87.9	73%

Table 10 - Most important attributes of Design Systems

When Ribeiro (2020) ran an analysis on the publicly available design systems of Fortune 500 companies, the researcher found out that only 50% of those 23 design systems detailed the underlying 'Design Principles' for a given system. Running the same analysis on the Adele Design System Database, only 38% of Design Systems available there had contents on Design Principles.

Given the importance attributed to both design and coding principles by respondents of this research, besides listing expected UI Components and other more obvious user interface elements, it is also important for Design System creators to explicit and elaborate on the guiding principles for each system.

4 DISCUSSION

This section outlines the answers to the reserved questions posed initially, suggests future research in this area and discusses the limitations of this study.

4.1 CONCLUSIONS

The main results achieved and contributions regarding,

- RQ1: How can Designs Systems be defined?
- RQ2: What are the common attributes of Design Systems?
- RQ3: What are the most important elements of Designs Systems according to UX professionals and practitioners?

HOW CAN DESIGN SYSTEMS BE DEFINED?

A literature review was conducted to find how practitioners and academics define Design Systems. Although it is a novel term and the literature is very scarce, the review identified 20 definitions that could be used for analysis. These were analysed, their main concepts extracted and explained, and a summarized definition was presented.

Design Systems are centralized repositories used by product development teams to manage, distribute and communicate user interface assets, code, documentation, guidelines and principles in a scalable way in order to ensure user experience consistency.

WHAT ARE THE COMMON ATTRIBUTES OF DESIGN SYSTEMS?

In order to answer this question, a comparative analysis of how authors and databases list Design System attributes. A summarized list was created based on the findings of Ribeiro (2020), Yew (2020) and UXPin's (2020) Adele Database. These are the common attributes of Design Systems found in this research.

Category	**Components (or Attributes)**	
Visual Design (Design Tokens)	● Colour Pallete ● Grid ● Screen breakpoints	● Shapes and radius ● Spacing ● Typography
UI Components	● Accordions ● Alerts ● Breadcrumb ● Buttons ● Chip/Pill/Badge ● Carousel/Slider ● Dropdowns ● Form elements (inputs, checkboxes, selects, radios)	● Icons ● Modal ● Pagination ● Progress ● Spinners/Loaders ● Tables ● Tabs ● Tooltips/Popovers
Patterns	● Page layouts ● Form validation ● Navigation (Navbars)	● Data grids ● Cards ● Data visualization
Design resources	● UI Kits (for Sketch, Figma, Adobe XD, etc) ● Brand files	● Stock image ● Stock illustration
Development resources	● Code Repository ● Code Documentation/Samples ● Integrated JS Library/Framework	● Extensible CSS styles (SASS, Less, etc) ● Distributed/Installable packages (npm)
Guidelines	● Accessibility guidelines ● Tone of voice	● Coding guidelines ● Animation guidelines

	• Brand guidelines	• Responsiveness guidelines
Principles	• Design principles • Coding principles	

Table 11 - Common attributes of Design Systems

WHAT ARE THE MOST IMPORTANT ELEMENTS OF DESIGNS SYSTEMS ACCORDING TO UX PROFESSIONALS AND PRACTITIONERS?

Based on the attributes above, an online questionnaire was created where each of the attributes was placed in a scale with 3 values: 'Very Important', 'Somewhat Important' and 'Not Important'. There were 22 respondents from 9 countries. A scoring system was defined so that elements could be ranked. The full list ordered by score can be found on Table 9, on page 54. The results where respondents ranked elements with at least 70% 'Very Important' is documented on Table 10, on page 62.

4.2 LIMITATIONS

This research faced several limitations throughout its course. The first that must be noted is the low volume of academic publications covering the topic of Design Systems. This impacted the possibility of finding more academic literature on the topic. So, in order to conduct the necessary literature review, the researcher had to rely less on academic literature and more on informal texts such as blogs or online books. This also reinforces the relevance of the topic and the importance of further research.

Another limitation was in the collection of survey responses. Independently of the scoring system used and the variables related, the number of respondents was lower than expected, which might have impacted the representativeness of the findings.

Finally, it must be mentioned that the COVID-19 pandemic also made any research during this atypical year very challenging, which might have impacted both on the time available for this research, and on the amount of respondents.

4.3 FURTHER STUDIES

Considering the limitations faced by this study, it is strongly recommended that more research is carried out in this field of study, especially in attempting to increase the number of respondents so that results are more statistically relevant. Also, as mentioned by some respondents, different Design Systems, for different applications, will have different needs. It would be valuable to study the application of this tool in different verticals across the technology industry.

5 REFERENCES

Barnes, Susan B. (2010) "User Friendly: A Short History of the Graphical User Interface," Sacred Heart University Review: Vol. 16 : Iss. 1 , Article 4. https://digitalcommons.sacredheart.edu/shureview/vol16/iss1/4

Churchill, E. F. (2019). Scaling UX with design systems. Interactions, 26(5), 22–23. https://doi.org/10.1145/3352681

Curtis, N. (2018, June 15). Defining Design Systems. Medium. https://medium.com/eightshapes-llc/defining-design-systems-6dd4b03e0ff6

Da Silveira, C., Lages, C., & Simões, C. (2013). Reconceptualizing brand identity in a dynamic environment. Journal of Business Research, 66(1), 28-36.

Damon, Dan. (2016). Johnston Sans: The Tube typeface that changed everything. BBC, Retrieved from https://www.bbc.com/news/magazine-35916807

Dondis, D. A., & Camargo, J. L. (1997). Sintaxe da linguagem visual. São Paulo: Martins fontes.

Dukic, N. (2020). Design Systems for improved development efficiency in Software Startups. (Master thesis, Aalto University, Finland). Retrieved from https://aaltodoc.aalto.fi/handle/123456789/46035.

Edelberg, J., & Kilrain, J. (2020). Design Systems. Proceedings of the 38th ACM International Conference on Design of Communication, 1. https://doi.org/10.1145/3380851.3416743

Frost, B. (2016). Atomic design. Pittsburgh: Brad Frost.

Hacq, A. (2020, February 11). Everything you need to know about Design Systems. Medium. https://uxdesign.cc/everything-you-need-to-know-about-design-systems-54b109851969

Hartson, R., & Pyla, P. S. (2012). The UX Book: Process and guidelines for ensuring a quality user experience. Elsevier.

Hilton, C. E. (2017). The importance of pretesting questionnaires: a field research example of cognitive pretesting the Exercise referral Quality of Life Scale (ER-QLS). International Journal of Social Research Methodology, 20(1), 21-34.

Huang, Y. (2019). Developing a design system for an e-commerce website (Master's thesis). Aalto University. https://aaltodoc.aalto.fi/handle/123456789/38465

Immich, T., (2019). Viel System, aber wenig Design? Wie Design Systeme adaptiver werden können, um guter UX nicht eher im Weg zu stehen. In: Fischer, H. & Hess, S. (Hrsg.), Mensch und Computer 2019 - Usability Professionals. Bonn: Gesellschaft für Informatik e.V. Und German UPA e.V.. DOI: 10.18420/muc2019-up-0296. Retrieved from https://dl.gi.de/handle/20.500.12116/24470

Karapanos, E. (2012). Modeling users' experiences with interactive systems (Vol. 436). Springer.

Kholmatova, A. (2017). Design Systems: A practical guide to creating design languages for digital products. Smashing Media.

Koeze, E., & Popper, N. (2020). The virus changed the way we internet. The New York Times.

Konaté, A. (2018). Design systems at work: Optimizing design processes and aligning design work to company identity. (Master thesis, Aalto University, Finland). Retrieved from https://aaltodoc.aalto.fi/handle/123456789/41399.

MacDonald, D. (2019). Practical UI Patterns for Design Systems: Fast-Track Interaction Design for a Seamless User Experience (1st ed.). Apress.

Mirnig, A. G., Meschtscherjakov, A., Wurhofer, D., Meneweger, T., & Tscheligi, M. (2015, April). A formal analysis of the ISO 9241-210 definition of user experience. In Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (pp. 437-450).

Moscato, M., (2018). How building a design system empowers your team to focus on people — not pixels. Retrieved from https://product.hubspot.com/blog/how-building-a-design-system-empowers-your-team-to-focus-on-people-not-pixels

Mozilla Foundation: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Introduction

N.A.S.A. (n.d.-a). Photos of the NASA Graphics. Standards Manual. Retrieved January 4, 2021, from https://standardsmanual.com/pages/original-nasa

NYCTA (n.d.). Photos of the NYCTA Graphics. Standards Manual. Retrieved January 4, 2021, from https://standardsmanual.com/pages/original-nycta

O'reilly, T. (2009). What is web 2.0. " O'Reilly Media, Inc.".

Punchev, Y., & Williams, O. (2019). Getting Started with Design Systems. https://Marvelapp.Com/Ebook-Design-Systems. https://marvelapp.com/ebook-design-systems

Pyrhonen, E., Lahdesmaki, R., Villarreal, F., Reyes, M., Clouin, P. H., Knutsson, M., & Torgersen, B. (2019). Hack the Design System: Revolutionize the way your organization scales design (Idean Design Books). Independently published. https://www.idean.com/micro/invent-design-systems

Rosenberg, G. (1997, March 31). Electronic discovery proves an effective legal weapon. The New York Times, Retrieved from http://www.nytimes.com

Roto, V., Väätäjä, H., Law, E., & Powers, R. (2016, October). Experience design for multiple customer touchpoints. In Proceedings of the 9th Nordic conference on human-computer interaction (pp. 1-3).

Ruissalo, M. (2018). Operating a design system in a large software company (Master's thesis). Aalto University. https://aaltodoc.aalto.fi/handle/123456789/32488

Saint-Martin, F. (1990). Semiotics of visual language. Indiana University Press.

Saliola (2020), https://hbr.org/2020/09/how-to-harness-the-digital-transformation-of-the-covid-era

Slifka, J., & Pergl, R. (2020, April). Laying the Foundation for Design System Ontology. In World Conference on Information Systems and Technologies (pp. 778-787). Springer, Cham.

Stepanova, V. (2018). Design Systems — review of vocabulary and terms. https://varya.me/blog/design-systems-review/

Stephanova, V. (2018). Design Systems - review of vocabulary and terms. Retrieved May 31, 2020, from https://varya.me/blog/design-systems-review/

Stevens, R. (2020). Powered by Design. Rocky Nook, Inc. https://books.google.com.br/books?id=BArYDwAAQBAJ&lr=&hl=pt-BR&source=gbs_navlinks_s

Stickdorn, M., & Schneider, J. (2010). This Is Service Design Thinking: Basics, Tools, Cases. Amsterdam: BIS Publishers.

Suarez M., Anne J., Sylor-Miller K., Mounter D., and Stanfield R. (2017) Design System Handbook, https://www.designbetter.co/design-systems-handbook

Toman, J. (2019, May 3). Design systems, style guides, pattern libraries. What the hell is the difference? Medium. https://product-unicorn.com/design-systems-style-guides-all-those-libraries-what-the-hell-is-the-difference-4c2741193fdc

Tombari, M. L., Fitzpatrick, S. J., & Childress, W. (1985). Using computers as contingency managers in self-monitoring interventions: A case study. Computers in Human Behavior, 1(1),

Vachhar, V., & Maritan, C. (2019, July 29). What is a Design System? https://rangle.io/blog/what-is-a-design-system/

Vesselov, S., & Davis, T. (2019). Building Design Systems: Unify User Experiences through a Shared Design Language (1st ed.). Apress.

Vial, G. (2019). Understanding digital transformation: A review and a research agenda. The Journal of Strategic Information Systems, 28(2), 118-144.

UXPin. (2020). Adele. Adele - Design Systems and Patterns Repository. https://adele.uxpin.com/

W3C: https://www.w3.org/wiki/The_web_standards_model_-_HTML_CSS_and_JavaScript

Yew, J., Convertino, G., Hamilton, A., & Churchill, E. (2020, April). Design Systems: A Community Case Study. In Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (pp. 1-8). Retrieved from https://dl.acm.org/doi/abs/10.1145/3334480.3375204