

Wellcome design systems discovery

August 2023

Contents

- [Contents](#)
- [About this document](#)
- [Wellcome design systems background](#)
- [Aims](#)
- [Design systems at Wellcome today](#)
 - [Parts](#)
 - [People](#)
 - [Teams using design systems](#)
- [Findings and recommendations](#)
- [Summary](#)
- [Detailed findings and recommendations](#)
 - [A lack of governance is the biggest hindrance to design systems work](#)
 - [Standards and quality assurance are needed to support design system delivery and contribution](#)
 - [Wellcome wants a design system that supports creativity](#)
 - [People are missing the guidance they need to use the design system as intended](#)
- [Proposed roadmap](#)
- [Contact details](#)

frankly.

About this document

This document summarises findings from:

- a series of interviews with the users, contributors, and stakeholders of design systems at Wellcome between July and August 2023
- a review of Wellcome's public-facing digital products, as well as code and documentation pertaining to the organisation's design systems

Based on those findings, it includes a set of general recommendations and a loose roadmap for scaling and sustaining design systems across The Wellcome Trust and Wellcome Collection.

The content in this document was researched and authored by design systems consultants Amy Hupe and Heydon Pickering on behalf of [Frankly Design Ltd.](#)

Wellcome design systems background

Wellcome's digital presence is split across multiple websites, products and services.

The 2 main websites, [wellcome.org](https://www.wellcome.org) and [wellcomecollection.org](https://www.wellcomecollection.org), are run by different teams, with separate budgets and distinct brand values and assets.

As a consequence of this, 2 separate design systems have been established for The Wellcome Trust and Wellcome Collection.

Aims

At the time of writing this report, Wellcome Collection is undergoing a strategic realignment with Wellcome Trust.

Although there is some uncertainty about the eventual brand relationship between the 2 main websites, there is a significant enough degree of commonality between them to consider bringing the 2 design systems into closer alignment.

The aim of this discovery is to explore opportunities to better standardise:

- ways of working
- tooling
- documentation
- styles, components and design patterns
- digital accessibility best practices

Wellcome is keen to reduce friction in designers' and developers' workflows, and seeks guidance on improving and maintaining design systems within its existing capacity.

Design systems at Wellcome today

Parts

On the **Wellcome Trust side**, the design system is noticeably more pared back than its counterpart for the Wellcome Collection. However, there is more alignment between its constituent parts.

It consists of 3 main parts.

1. A collection of design foundations in Figma

[A core or "global" library of foundations in Figma](#) sets out:

- colours
- responsive spacing
- responsive typography
- icons

- buttons and links

These foundations are used across all products on the Trust side. In addition, each team uses a set of local Figma styles and components specific to their product.

2. A code library

The design system codebase is documented in [Storybook](#). It mostly maps to the styles and components set out in the global Figma library, with a few exceptions, including icons.

Developers in the Trust's product teams draw from this codebase when building out their UI by installing packages from the [Wellcome Design System npm](#).

There is currently no automation to keep the codebase in sync with what's in Figma, so alignment is achieved manually.

3. A wiki on Notion

[The Wellcome Design System wiki on Notion](#) sets out the design system's purpose, vision and some guidelines to support assets in Figma and Storybook, as well as links to them.

It also includes some lightweight accessibility and tone of voice guidelines.

Research participants commented that the wiki represented a useful place to start, but was not systematically maintained and some of its content was out of date.

On the **Wellcome Collection side**, the design system contains a much larger collection of styles and components than exists in the Wellcome Trust's Design system.

However, its 4 main constituent parts are much more varied in terms of size and maturity.

1. A component library in Storybook

Arguably the most comprehensive part of the Collection's design system is its [component library in Storybook. Cardigan.](#)

Cardigan documents a large number of components as well as icons, colours, spacing and typography,

Research participants told us that Cardigan serves as more of an inventory of components that exist at a product level, albeit not a complete one. Documentation is patchy and does not follow a consistent structure.

There is significant misalignment between the components and styles documented in Storybook, and those in Figma.

2. Figma library

Like the Trust team, the Wellcome Collection design system also includes a [Figma library of design principles, styles and form elements.](#)

To date, the library has been created, maintained and used by the sole designer on the Collection side, Dominique Marshall.

3. Documentation in Notion

As on the Trust side, the Wellcome Collection's design system inhabits a space in Notion, setting out documentation pages for:

- design principles
- foundations
- page types
- components
- patterns
- accessibility
- content

- developer resources
- structure
- resources
- roadmap
- design system checklist

While the list of contents appears comprehensive, many of the links reveal missing documentation (here's an [example](#)).

Research participants told us that the Notion documentation is outdated and needs reviewing and updating, although the current plan is to migrate this documentation to GitBook.

Documentation does not follow a specific structure even within a single category.

4. Documentation in Gitbook

Alongside the Collection's design system documentation in Notion is a subset of this in [Gitbook](#), namely:

- design principles
- foundations
 - responsive grid
 - design tokens
 - page types

A decision was taken to migrate documentation to Gitbook from Notion - research participants were not aligned or clear about the reason for this. This exercise was started around a year ago, and has not been completed.

As with Notion, participants commented that the documentation in Gitbook is somewhat outdated and needs reviewing and updating.

In addition to the Wellcome Trust and Collection design system resources, there is also a [brand book hosted on Frontify](#) for both parts of the organisation. The audience and scope for this overlaps with that of the design systems, but is broader.

Historically, the brand books have been geared towards brand designers working on offline assets, like print and exhibition materials.

Some effort has been made to map the styles set out in the Trust's brand book to their design system, although there is still some misalignment.

People

There's currently no dedicated full-time team working on design systems at Wellcome. A Product Designer, Jason Jaworski, has "design systems lead" appended to his job title.

However, it should be noted that he has requested to have design systems work explicitly removed from his performance objectives owing to a lack of resources and necessary support to move this work forward.

The parts that exist today have been created by Jason, and by others working on it voluntarily in their spare time.

Teams using design systems

At the moment, the design systems on the **Wellcome Trust** side are being used and contributed to by teams working on the following 4 products:

- [Wellcome.org](#)
- [Funding platform](#)
- [Data and Insights finder](#)
- [Trustnet, the organisation's intranet](#)

Designers on these teams are mostly adhering to the design foundations, namely colours, spacing, grid, typography set out in the Wellcome Trust design system's Figma file.

In addition, each team has built on top of the core design foundations with local styles and components documented in their own Figma files.

On the development side, Wellcome.org consumes design system modules as part of its Next.js-based frontend. This is backed by Drupal.

In the **Wellcome Collection**, the design system is used to varying degrees by the designers and developers that look after wellcomecollection.org.

The Figma library was created, and is maintained and used by the sole designer on this side of the organisation, Dominique Marshall.

The codebase documented in Storybook is used and contributed to by multiple developers. Dominique refers to Storybook when designing to check for alignment.

The frontend is in Next.js, as on the Trust side, incorporating Typescript. In this case, content is rendered by querying the CMS Prismic's content model API to populate React components. There is not always a *"one-to-one mapping"* between components and Prismic's *"slices"* (typed content blocks).

Participants told us that components do not always match their requirements, and in these instances, local components are often created for specific product areas and may or may not be added back into the design system.

Findings and recommendations

Summary

Finding	Recommendation
<u>A lack of governance is the biggest hindrance to design systems work</u>	Set up a centralised or federated design system team to own the work
<u>Standards and quality assurance are needed to support design system delivery and contribution</u>	Define a set of success criteria and a review process for additions to the design system
<u>Wellcome want a design system that supports creativity</u>	Create a unified system of modular elements and rules, making component composition the responsibility of product teams
<u>People are missing the guidance they need to use the design system as intended</u>	Make the most of interdisciplinary review to document usage guidance and implementation detail in Figma

Detailed findings and recommendations

A lack of governance is the biggest hindrance to design systems work

“The design system is not an explicit responsibility of anyone’s role”

A universal view amongst the research participants we spoke to was that the lack of clear ownership and available time to dedicate to systems work was the biggest barrier to progress.

Sporadic efforts on the existing design systems has led to conspicuous gaps, out-of-date documentation and, on the Collection side in particular, an imbalance in the design and code sides of the system.

Decision-making is challenging in the absence of ownership. From one participant, we heard that, *"Wellcome has so many different levels of people making decisions. You sometimes go around in circles, as a designer, getting consensus and approval. I find that really quite annoying."* Another recalled a prolonged issue with an update to the colour palette because *"we didn't know who made the final decision"*.

Encouragingly, there was no question about the potential value of design systems. As one participant noted, *"I don't think there's anyone resistant to it. It is just a case of 'how do we fit it in?'"*. Another told us that they were keen to contribute more but *"would need to know who to approach [so] it's not opaque"*.

This suggests that previous efforts in advocacy have paid off, and that the work now lies in establishing strong governance, not in persuading designers and developers of the need for this work to happen.

Whilst there appears to be an appetite to develop design systems collaboratively, participants spoke of the need to assign responsibility and time for people to work on design systems.

Almost everyone we talked to was in favour of instating a dedicated design systems team. *"It needs to be living. It needs a dedicated team. It needs a product owner, designers, and developers."* However, many felt it unlikely that the organisation would invest in one.

Recommendations

There are various models for governing design systems, each with their own strengths and weaknesses. A useful overview is provided in Nathan Curtis's article, [Team models for scaling a design system](#).

In our experience, the most successful approach is to have a centralised and dedicated team owning the design system, working in close collaboration with designers and developers across an organisation.

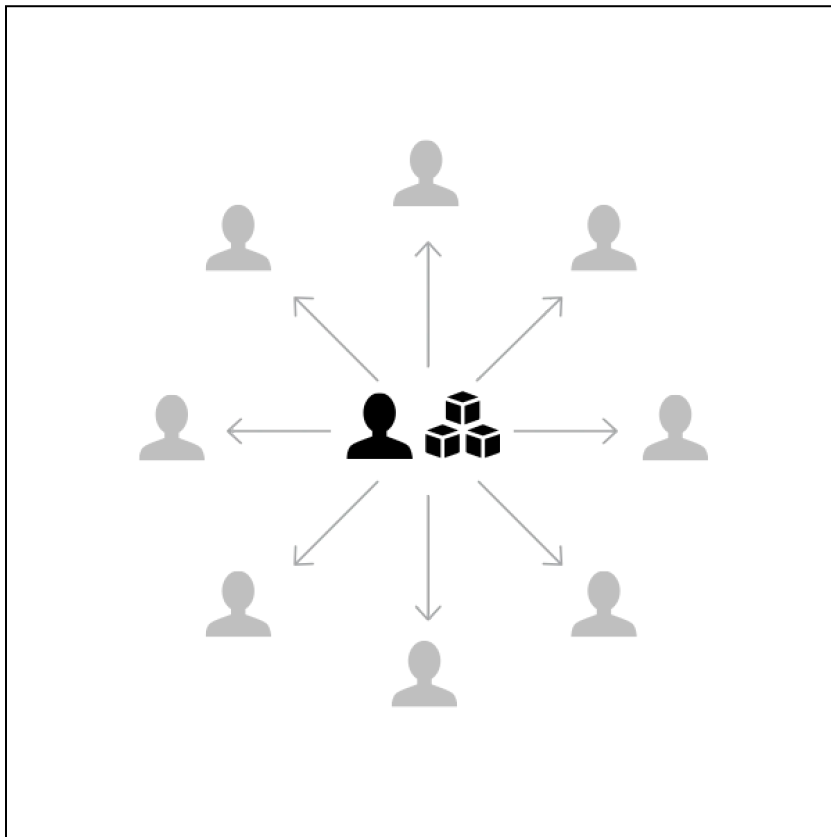


Illustration of a centralised design system team, Nathan Curtis

If Wellcome does decide to go down this route, we recommend the team is positioned at an organisation level, and not tied specifically to the Trust or the Collection.

frankly.

At this early stage, an adequate size and make-up for a design systems team would be:

- 1 team lead (this could be a product owner, or a senior designer or developer)
- 1 designer
- 1 frontend developer
- 1 technical writer / content designer

If hiring or reallocating permanent staff to a centralised team is not possible due to budget constraints, Wellcome may want to consider enlisting the support of specialist external contractors to fulfil some of these roles.

Since the bulk of the work involved in creating a system's foundations (governance, standards, base styles, guiding principles and documentation templates) happens up front, relying on temporary external support may actually be of benefit to Wellcome, provided that:

- contractors work very closely with in-house product teams to ensure knowledge and skills are passed across to Wellcome
- some ongoing support for the design system is provided by internal staff members once any contractual engagement has ended - although this may be scaled down compared with the setup phase

Instating such a team, even if it's modest in size, ensures standards are met and followed, contribution is coordinated, and support requests are dealt with. This tends to lead to greater trust and more consistent adoption.

It bears emphasising that centralised design system teams must work closely with their product counterparts to be successful. A team creating components in isolation will inevitably fail to meet the

frankly.

requirements of the teams it serves, and can foster mistrust and hinder adoption.

This risk was acknowledged by one research participant, who told us that any design system team must *“be in the weeds using stuff in production too”* and that failing to *“keep one foot in product work”* would render their work *“too theoretical”* and cause them to lose touch with reality.

A healthy design systems team encourages continuous input and contribution from its community of users.

If investment for a dedicated team is not possible, an alternative solution is the federated governance model. In this scenario, selected individuals are assigned design systems work as part of their existing roles. They are allocated time to contribute to the development of the design system, ensure its upkeep, and respond to support requests from design system users.

Federated teams are inherently more complex than their centralised counterparts, but can prove fruitful provided there is regular communication, time to work on the system, and senior support to protect this time.

A successful federated team:

- includes representatives from all flagship product teams
- contains a balance of designers, developers and content specialists (for example, technical writers or content designers)
- is allocated a clear and agreed amount of time to work on the system (for example, 1 day a week)
- assigns decision ownership and other responsibilities to specific individuals
- has a clear set of objectives / deliverables that are explicitly tied to its members' performance reviews

frankly.

- via its members, communicates its work outwards to product teams, and uses input from product teams to inform the system's development

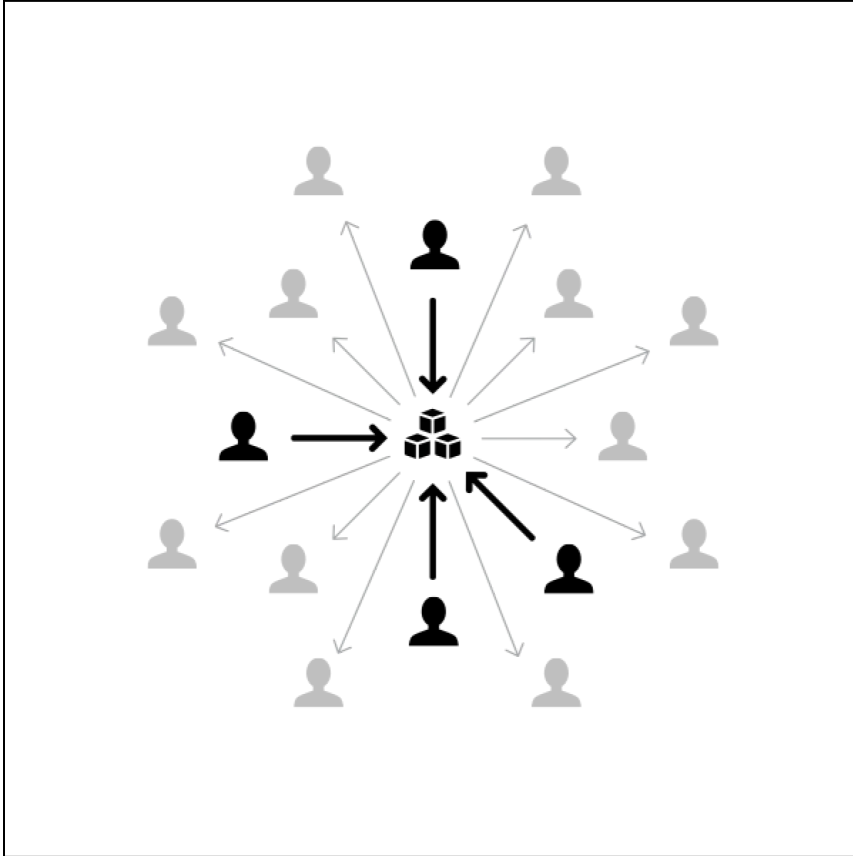


Illustration of a federated design system team, Nathan Curtis

To set up a federated team at Wellcome, we recommend the following:

1. Agree an initial timeframe for the federated team to work. At the end of this timeframe, review what the team has delivered against the intended outcomes. We recommend 3 to 6 months.
2. Identify the teams' members. As above, ensure representation from a balance of designers, developers and content specialists working across Wellcome's flagship products.
 - a. As mentioned above in reference to a centralised team, Wellcome may want to consider hiring external contractors

frankly.

to work on the design system to support the federated team during the first phase of work. This will reduce some of the reliance on reallocating internal staff.

3. Identify senior sponsors for the design system team. For efficiency and simplicity, we recommend no more than 2 sponsors. These should be individuals who can advocate for the team's work amongst product teams and the organisation's leadership, and with adequate authority to support the team with any blockers it faces.
4. Agree how much time the team's members can dedicate to design system work.
5. Together with relevant stakeholders, the teams' members agree what it aims to deliver in the allotted time. (We suggest some specific deliverables and focus areas later on in this report)
6. Hold a kick-off session to agree ways of working. This should cover, but may not be limited to:
 - a. how the team will communicate with each other
 - b. whether work will happen synchronously or asynchronously, or a through a combination of both
 - c. release processes including communication of releases to the systems' users and stakeholders
 - d. team ceremonies and planning processes
 - e. roles and responsibilities of the team's members beyond core specialisms (for example, who will run planning sessions? Who will write release notes? Who will provide support to system users?)
 - f. an initial roadmap
7. Agree regular review points to assess how the team is working and identify blockers. This could be done in the form of a team retro.

At the end of the agreed timeframe, the team may be scaled up or down as appropriate to the next phase of work. We have explored this

frankly.

in more detail in the later section, [Wellcome wants a design system that supports creativity](#).

Its members may be swapped for other individuals in the organisation if there is an appetite for rotation.

One final note regarding this recommendation is that any design system team, whether centralised or federated, must continue to be in place for the entire lifetime of the system. A system is a living entity that requires ongoing maintenance, support and development.

Therefore it is critical that as long as there is a design system, there is a team in place with ownership for that system.

Standards and quality assurance are needed to support design system delivery and contribution

Second to the lack of ownership and available time, the absence of a clear set of standards to work to and a process for assuring those standards was cited by research participants as the biggest blocker to design systems work.

One participant expressed frustration at the stagnation over the design system's evolution owing to the lack of contribution guidance: *"We need a clear process so people can get on with it."*

The absence of a set of standards to follow is evident in gaps and inconsistencies amongst the systems' contents. As one participant observed *"There are a lot of empty README files. A listing component that just says 'this is a listing component'"*.

With no clear definition of "done", styles and components within the system vary significantly when it comes to robustness and documentation.

Without standards to adhere to, design systems can easily become propagators of the needless inconsistencies they aim to counteract. One participant shared a particular instance of this happening at Wellcome *"We found out our headings were different, and not deliberately, in style. We use classes to style the headings. But if you have it in the body (different context) it becomes problematic."*

While those we spoke to acknowledged an open and collaborative mindset around design system contribution, the lack of a process to follow has created a perception - particularly amongst new starters - that the systems are static and cannot be changed. *"There's not a clear process. It's really chill but, for someone new, there's a lot of hesitation to touch it. They wouldn't know who to get it approved by."*

Despite a clear willingness to help develop the design system, people are unclear about how and what they should contribute, and who gets to make decisions.

Recommendations

In the short term, Wellcome should look to establish 3 things:

1. A set of criteria that sets out the necessary standards of design system entries
2. A checklist of deliverables for new additions to the design system (which should be retrospectively applied to existing things in the design systems)
3. A review process for ensuring criteria are met and work is complete before publishing

These standards and processes should be applied to the design systems in both the Wellcome Trust and the Wellcome Collection. This will support the organisation's endeavour to bring the respective systems into better alignment and standardise ways of working amongst the 2 teams.

In the case that a design system team (central or federated) is established, their own work should undergo the same assurance practices as the work of external contributors. This will ensure that standards are applied consistently, and will support trust—a system team who marks their own homework can rouse suspicion.

We recommend that to begin with, design system additions are only made by design system team members and not by external contributors. In our experience, trying to support contribution while establishing system norms, and indeed building the system, creates too much overhead on already-strained nascent design system teams.

Pausing contributions will therefore give the team space to bring design systems in line with agreed standards, and to test the efficacy of the assurance framework, without the demand of supporting unpredictable incoming contributions.

Wellcome should communicate that this is happening, as well as why and for how long, so that historic contributors are not taken aback.

To establish contribution criteria, Wellcome should ask 2 questions when it comes to adding something to a design system:

1. Should it be added at all?
2. What is the minimum viable level of quality it needs to meet before it can be published?

[The GOV.UK Design System has published its own contribution criteria](#) and this may serve as a useful starting point to guide such an effort at Wellcome.

Regarding what should actually be created when something new is added to the systems, we have outlined a set of suggested deliverables later in this document in the recommendations for the finding [People are missing the guidance they need to use the design system as intended](#).

frankly.

We propose that Wellcome converts these deliverables into a published checklist for design system maintainers and contributors to refer to when updating or adding something new to a design system. Given the organisation's existing use of Notion, we recommend using its database feature to publish this and track progress against it.

The final piece of this particular puzzle is a process for ensuring that design system additions meet the agreed standards. To be successful, this process should comprise the following.

1. A nominated reviewer or working group of reviewers

The best models we've seen here are multi-disciplinary, cross-organisational working groups that can sufficiently represent the needs of different disciplines and product teams.

Reviewers do not necessarily have to be members of a design system team. In fact, choosing reviewers who sit outside of the design system team can help to improve representation of different perspectives, and build trust on the basis that the design system team is not acting as an overlord.

However, if reviewers *are* external to the design system team, the team should have the final decision on whether and when to publish new entries. This must be used responsibly and with a commitment from the team to uphold the decision of the external reviewers as often as possible.

2. Pre-agreed points to review design system additions and contributions

These can be set at a specific cadence (for example, weekly) or can be dealt with ad-hoc, when a new contribution is made.

Our recommendation is to trial a regular review point to begin with, to set expectations for contributors and reviewers about timings.

frankly.

When setting the frequency of review points, consider the number of expected contributions and reasonable wait times for contributors. Carrying out reviews at least once a month ensures momentum is maintained.

3. An agreed decision-making protocol

In the case that a group of reviewers is established, is complete consensus required or will a majority vote suffice? Does each reviewer hold equal power or do some have more authority than others? Is there such a thing as a veto, in this model?

The specifics of the protocol are less important than the fact that one exists at all. Decide on a process to trial, and publish it alongside contribution guidelines to set clear expectations about how assurance is carried out.

Our last word on standards and assurance is that Wellcome should endeavour to publish its quality criteria, expected deliverables and review process somewhere that everybody in the organisation has access to, and somewhere in close proximity to other design system documentation.

Ensuring transparency around design system protocols will provide necessary reassurance that the systems are living entities, are dependable, and will chart a clear course for prospective contributors.

Wellcome wants a design system that supports creativity

Both designers, and developers on behalf of designers, talked about the need for a design system that enables the creation of fresh and compelling work. The term “*flexible*” came up multiple times. One participant expressed the aspiration that their design system would free everyone up to focus on the “*real decisions*”. It would, “*give us the*

frankly.

freedom to think through fresh problems." This sentiment was echoed by others.

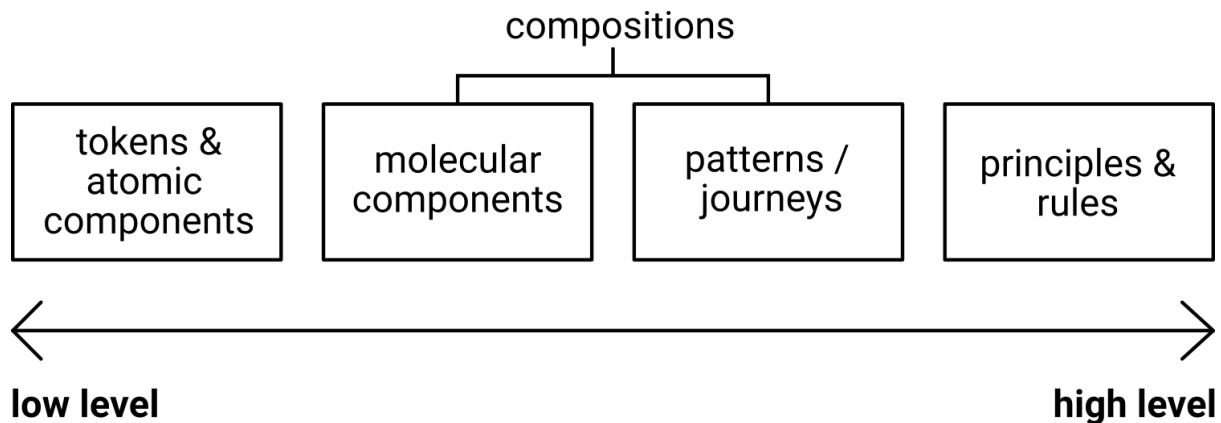
Developers spoke directly about flexibility. One expressed wariness of the stance *"we've created all these React components and you have to use them"*. They were concerned that in this scenario, product teams would have to *"fight [the components]"* to achieve what they wanted.

One designer noted the proliferation of pagination components, despite one existing in the design system (Figma) file already. As a clue about why this might be happening, another participant told us that the standard components are not always *"exactly what I need"* so they use them as a *"starting point"* instead. We learned that individual product teams have tended to build their own style guides and component libraries, on top of shared design system assets.

Some from the Trust side of the organisation felt that the pared-back nature of the design system supported flexibility well: *"I find it flexible enough sometimes."* *"You can make something quite distinctive with the building blocks."* One compared the Wellcome Trust system favourably to design systems at previous jobs, saying those were *"too bossy"* relatively speaking. A developer argued the Trust design system being limited is useful; it's not *"too prescriptive"*.

Recommendations

One of the (many!) ways you can divide up a design system is according to the granularity of what it offers. At one end are high-level things like guiding principles and at the other are small indivisible values and elements. The middle ground is occupied by components and sometimes larger "patterns".



(For the purposes of this recommendation, an atomic component is one that would typically map to an individual HTML element such as a button, link, or heading, as well as individual pieces of media such as logos and icons. A token would be a single value such as a colour or spacing size.)

We believe that Wellcome's time and resources would be best spent focusing on:

- **tokens and atomic components**
- **principles and rules**

What's useful for Wellcome to keep in mind is that components and patterns of the middle ground are necessarily *compositions*, made by combining the abstract and the atomic from either end of the spectrum. While they are often characterised as part of the design system, it's sometimes better to think of components as an *outcome* of the system. That is, given an agreed ethos and sufficient materials, you can create components systematically. From this perspective, components do not need to live *in* the system, as reusable assets.

At Wellcome, treating components as a (direct) consumable for product teams can lead to stifled creativity and (some have argued) insipid output. By the same token, where people have the autonomy to create their own compositions, they feel their work is of a higher quality.

frankly.

That components are frequently created at the level of individual products and not contributed back to the design system is not necessarily a problem.

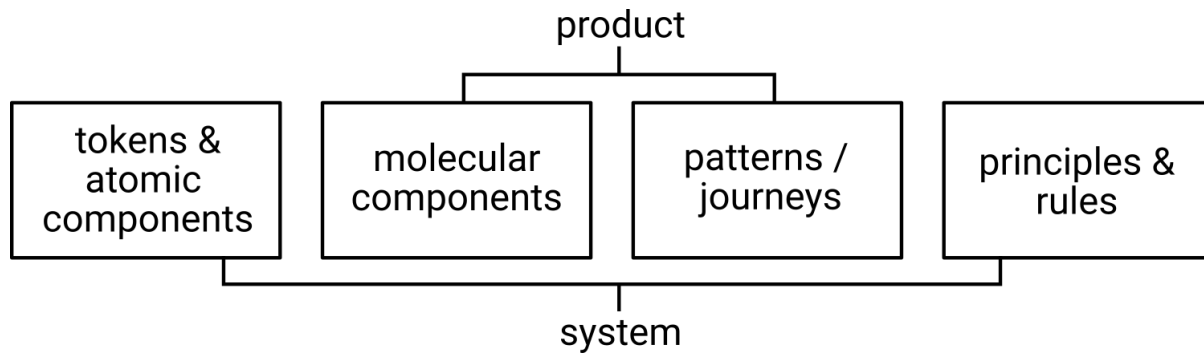
In fact, in the absence of dedicated people to develop and maintain the design system and given the work involved in creating and documenting system components, it's suitably resource efficient. And given the creativity with which the organisation wants components to be designed, it's unlikely that any existing component (of a reasonable complexity) is reused in its given form.

Making component creation the responsibility of product teams enables them to work autonomously and creatively. The judiciously limited offering of the Wellcome Trust system already has success in this area. However, the approach should be adopted more widely and with greater deliberation. The lack of guidance accompanying design system assets is a major theme covered elsewhere in this document. In this specific case, the following needs to be documented, as part of the high-level guidance:

1. The expectation that product teams are to create their own (complex) components in deference to the design system and the fundamentals it offers.
2. How product teams should approach composition such that they can exercise creativity without creating too much inconsistency.

In regards to (2), many participants noted a lack of "rules" for contributing to the design system, as well as a lack of time to do so. By providing axiomatic rules for working with the design system, you enable product designers and developers to engage with the system in the creation of original work without asking them to contribute everything back to the system directly and as a separate undertaking. The work itself exemplifies the system.

frankly.



Something to explore in terms of supporting component composition are layout primitives. Layout primitives are small, generic, and single purpose components that just handle layout. Perhaps the most well-known is [the stack](#), which apportions vertical (block) margin between successive elements.

Since margin is not rightfully a property of an element but rather a property of the relationship between 2 elements, a utility layout component takes responsibility for it. Layout primitives are a kind of glue that can bind atomic elements together to form molecular components in a consistent, efficient, and intrinsically responsive fashion. A number of these layout primitives are explored in [Every Layout](#). (Full disclosure: Heydon is a co-author of this resource.)

Another advantage of focusing on the atomic parts of the design system is that these are parts—fonts, colours, breakpoints, etc—typically common to all corners of the Wellcome organisation (the Trust and Collection). They also tend to be parts relevant to digital as well as brand. We understand alignment between brand (print) and digital is something actively being worked on already, with the brand side reaching out to digital and vice versa.

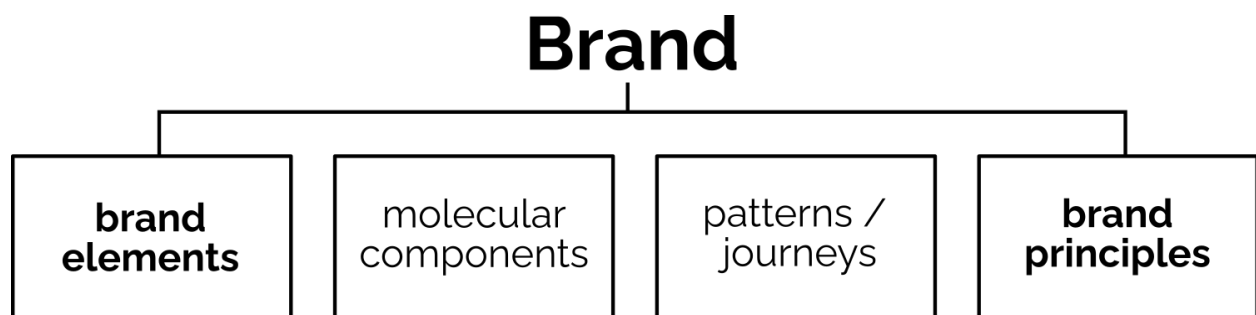
As one participant noted of the sibling design systems, there's currently *"a lot of duplication of effort."* One way to demonstrate the current simultaneous commonality yet disparity between Wellcome Trust and

frankly.

the Collection is by using Project Wallace, a CSS analyser. As you can see, broadly the same set of fonts are imported, just in different ways:

- wellcome.org
- wellcomecollection.org

There's an opportunity to define a system with the flexibility to serve everyone, without diminishing or restricting anyone's work. As one participant said, the system should be flexible enough that, while there are constraints, *"we can still get enthusiastic and creative. 10 projects can share stuff but still have their own identity."* Given an existing culture of knowledge sharing at Wellcome, building a design system of shared purpose does not seem like an unattainable endeavour. Especially where it is no longer an exercise dominated by accumulating and maintaining complex components.

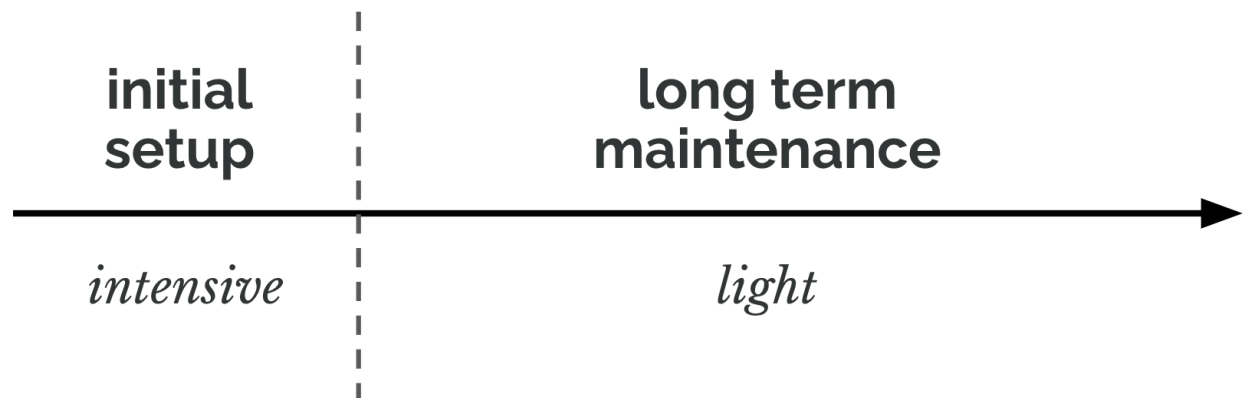


There was some enthusiasm about aligning the Trust and Collection systems, with one participant saying, *"I think it would be amazing if we were all using the same system"* and another saying aligning design systems would be *"the next best thing"* to aligning ways of working.

But others were more cautious. One felt quite *"prickly"* and was concerned aligning the 2 systems would *"water down"* the work on either side. By limiting the design system offering and freeing product teams to generate components from the fundamentals it offers, we believe nobody will feel *"watered down"* or at risk of losing product identity.

frankly.

Since big, abstract things like guiding principles and small things like spacing values or colours tend to change little over time, intensive work on developing a shared design system could be confined to a timescale.



This is not to say that ongoing responsibilities towards the design system (things like user support, managing contribution and system maintenance) should not be defined, perhaps on a percentage-of-role basis—as one participant suggested. It's just that an initial push should create a lot of inertia.

People are missing the guidance they need to use the design system as intended

"You can't [currently] say 'here's how we create a cohesive product.'"

One of the largest trends we identified was a lack of confidence in how to apply design system assets correctly. When asked to describe a successful design system at Wellcome, one participant replied, *"in my mind it is the components, how they are glued together, and rules, ownership and governance."* They pointedly added, *"we are good at the first part."* Another noted, *"we've never really nailed documentation for component usage."* Unfortunately, adding usage guidance, *"just didn't happen."*

frankly.

On the Trust side of the organisation, for example, there was uncertainty about when and when not to employ icons: *“Do I get points for using icons or lose points for not using all of them?”* From a Collection participant, it was reported, *“we don’t really understand where to use the different [colours].”* One participant suggested the organisation may need to take on *“specialist technical writers”*, because there are currently only *“implied rules for components etc”* and not always anything *“explicit”*.

One participant was concerned *“we’re not sharing enough”*, meaning, for consistency's sake, *“we have to do that stuff”* (have repeated cross-team meetings). This was echoed by another participant who felt, *“getting everyone’s thoughts”* got in the way of getting work done.

A developer experienced the need to give repeated feedback over small mistakes. *“If teams are not using the standardised variables/tokens it’s frustrating.”* By the same token, a designer hoped for *“a day where I don’t have to look out for 1px mistakes”*.

It’s felt that not only a lack of guidance but not knowing where to look for guidance has led to an over-reliance on direct communication. *“As you’ve probably noticed, there’s a lot of documentation in a lot of different places”*. It was explained a designer being onboarded would have to be sent to Figma to see UI designs but also Notion for the brand guidelines. Another participant: *“From what I understand we used to go to Notion, but now we’re moving to Gitbook.”* The problem is exacerbated by the disparity between developer and designer content: *“what’s in the code and the design files is wildly different”*.

Recommendations

A system is at once:

1. A set of things connected and working together

2. A methodology for making a set of things connect and work together

A successful design system can be (2) alone or (2) incorporating (1) for the purpose of exemplification. However, (1) without (2) makes for a black box. A design system that does not reveal and explain itself cannot be easily used or extended. It becomes a catalogue of context-free past decisions, successful or otherwise, rather than a tool for successful decision making. Particularly on the Collection side, it appears that products inform the design system rather than the other way around. We're told the components are created in products first and added to Storybook.

Wellcome is able to subsist on design systems lacking comprehensive (and easily locatable) guidance thanks to a healthy culture of close and regular communication. In fact, when asked what's going best in the organisation, "*communication*" was the consensus.

While this culture should continue to be nurtured, relying too heavily on direct communication—crits, meetings, ad hoc discussions—carries certain risks. More than one participant pointed to work being left incomplete because an outgoing employee did not leave a paper trail. A lack of documented standards also creates inefficiency since—as noted in the preamble to this section—the same mistakes and omissions need to be repeatedly pointed out during review.

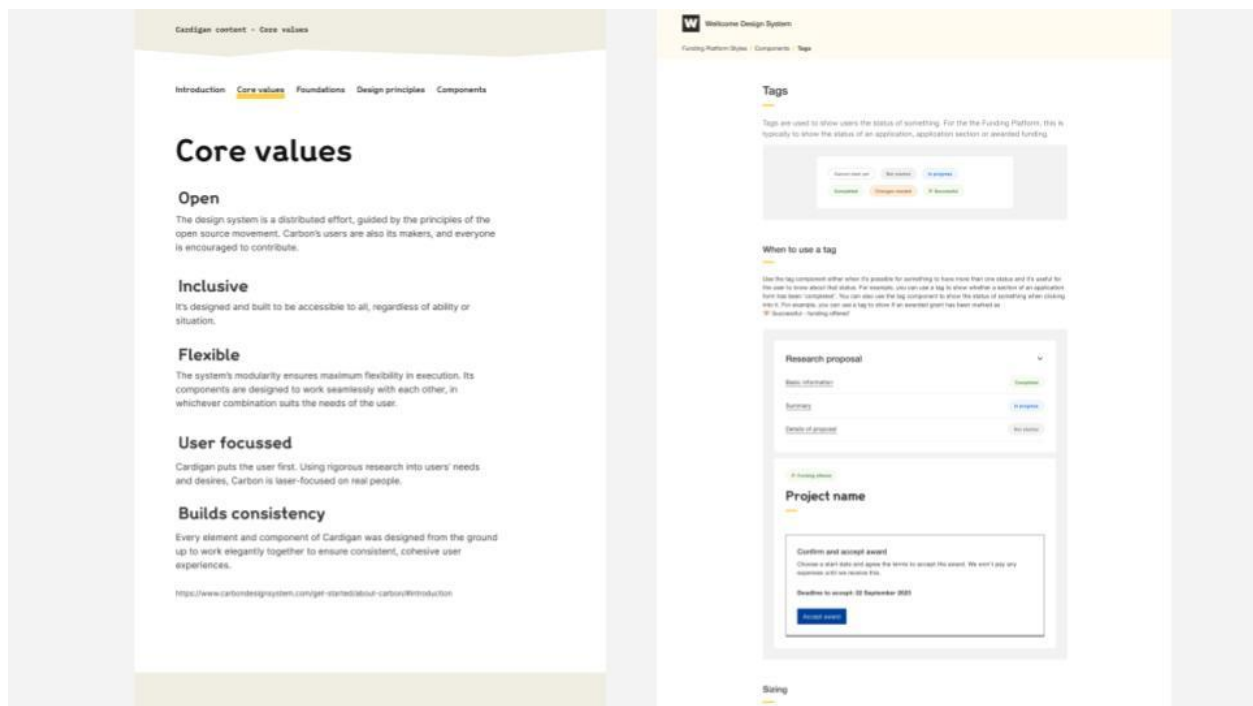
Establishing the need for considered usage guidance is less difficult than knowing where it should live and in what form. One participant favoured a "*Google-like*" one-stop design system website but most expressed preferences among the existing locations (Figma, Storybook, Notion, Gitbook).

On balance, we believe Figma is the most suitable "home" for design system usage guidance compared with any other existing location at Wellcome. Figma is where interface designs are realised, before being

frankly.

implemented and appearing in Storybook. While designers we spoke to tended to think of code as the rightful “*source of truth*”, there is generally a hand-off from designers to developers; Figma to CSS/React/Storybook.

There is already a precedent for publishing guidance in Figma, especially that of a higher level, on both the Trust and Collection sides. Designers from both teams shared prosaic Figma content to contextualise the styles and components they had been working on.



Given the primacy of Figma and the widely established practice of designers and developers pairing over Figma as part of hand-off (or “handshake” as one participant called it), we recommend the following. For the purposes of this recommendation, read “asset” to mean token/graphic/element/component.

1. Agree a template for documenting design system content in Figma. Each asset should be documented using a common structure. It should be a template with space allotted for:

frankly.

- a. The asset itself (in its default form), annotated as necessary to explain its anatomy and note implementation details.
 - b. Comprehensive documentation for the asset's role in the design system: what it is; what it isn't; which brand principles apply; who it's for; when you should use it; when you mustn't use it; what you can use instead (link to other, related assets)
 - c. Any acceptable variants (such as different colours and sizes for buttons) explained by accompanying use cases.
 - d. Any applicable states (such as hover, focus, empty).
2. Organise the assets into pages within Figma such that the navigation is like that of a design system website with tiered/nested menus where necessary (participants have said they are aspiring to this already).
3. Make sure Figma assets link to their code counterparts and the code counterparts (in Storybook, currently) link back to the Figma assets.
4. Leverage Figma's "dev mode" and complementary plugins:
 - a. Use the [Figma Github integration](#) to link Figma assets to their code counterparts.
 - b. Developers: since those we asked said they used VS Code, install and connect the [Figma VS Code plugin](#). This plugin lets developers discuss and implement Figma components directly in the VS code editor. It also exposes variables/tokens defined in Figma to editor suggestions.
5. If 4(b) is not sufficient, make Figma the source of truth for single value tokens/variables and integrate directly with code using a tool like [Design Tokens \(for Figma\)](#) or [Specify](#).
6. Use READMEs (and their rendering in Storybook) for development specific information only. Things like:
 - a. Installation and testing instructions
 - b. API information: events, properties ("props") etc.

7. Use Notion, and its database feature, specifically for tracking progress on the unified system. Make the most of its board and scheduling features.

Regarding (1): We are conscious of the accessibility issue of documenting your design system assets in Figma. Figma's interface is that of an artboard rather than a semantically tagged document. We reached out to a Figma representative to see if there is anything on the roadmap to enable semantic tagging and this is not the case. Given Wellcome's culture of cross-disciplinary collaboration over Figma, we believe it is still the best place to write documentation. However, for the sake of supporting disabled staff members present and future, the same documentation should be made available in Gitbook, which we have assessed as a satisfactorily accessible platform. The Collection side has already begun to migrate high level guidance to Gitbook, but an organisation-level instance should be instated.

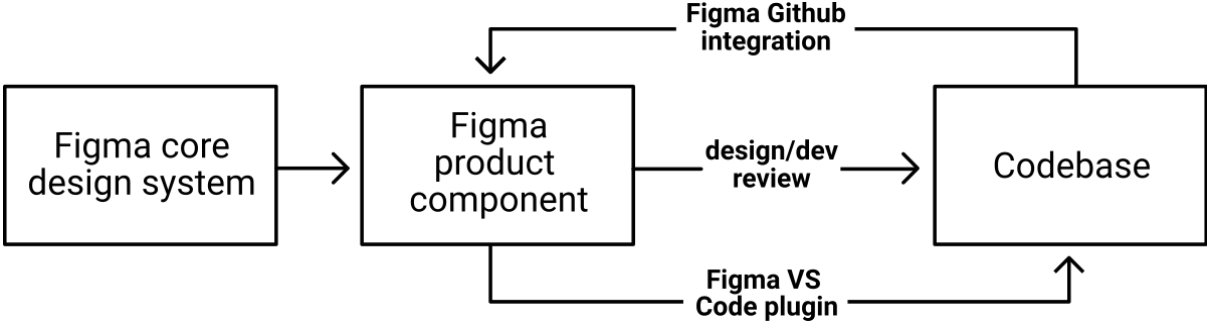
Regarding (4): Proprietary plugins and extensions alone cannot replace a well defined and documented process, as set out in [Standards and quality assurance are needed to support design system delivery and contribution](#). Instead, such tools are suggested only as a supplement, to help streamline said processes.

Importantly, make filling out the documentation for a Figma asset part of the existing pairing exercise between designers and developers during co-design and hand-off/handshake. Ensure the implementation details raised by the dev are included in the annotations. A prosaic example: if it is a font size variable, document which HTML heading level it should map to and why. This process may be improved by observing (4).

In the recommendations accompanying the finding [Wellcome wants a design system that supports creativity](#), we've argued that complex components should not form part of a maintainable and flexible core

frankly.

design system at Wellcome. Instead, focus on providing the kind of guidance covered in this recommendation for fundamental assets such as tokens and elements (atomic components).



That is not to say compositions of these assets should not be documented (as components) outside of the core system, in the ownership of specific product teams. In fact, the systematic and standardised approach to documenting core design system assets should be a model for product teams and their component documentation.

Review the design system offering in Figma to ensure the library of components/assets and accompanying variables are available and exposed to assist product teams in achieving consistent and well-documented components and patterns.

Proposed roadmap

This roadmap sets out a suggested sequence, not timescales, for the next phase of design systems work. The order is based on the respective urgency of each task, as well as the flow of dependencies between them.

Now

- Agree on governance model (centralised or federated team)
- Set up design system team and ways of working, including how updates will be released and announced
- Agree standards and assurance process for design system updates and additions
- Create an organisation-level Notion space that documents the above decisions and processes and sets out what's **Next** in the roadmap. Remove outdated Notion and Gitbook content relating to design systems and their management.

Next

- Using standards and assurance process:
 - review and deprecate outdated and redundant components
 - update and standardise documentation across remaining tokens and atomic components
- Agree a template for documenting design system assets (and components composed from those assets) in Figma, emphasising usage guidance. Mirror this guidance in an organisation-level Gitbook instance so there is an accessible alternative available. The developer working on the component would be responsible for keeping Gitbook in sync. The aim should be to demarcate documentation as follows, moving forward:
 - Figma: brand principles, contribution guidelines, design system assets alongside usage guidance.
 - Gitbook: accessible/semantic copy of the prosaic content in Figma (with Figma embeds for illustration where appropriate and possible).
 - Notion: tracking of work on the design system (making good use of databases and boards).

- Establish an architecture for a shared design system as documented in Figma, with clear navigation.
- Include high-level guidance (brand principles etc) in Figma, as part of the new architecture. Some guidance resembling this has [already been written](#), by the designer on the Collection team.
- Migrate any existing usage guidance from Storybook to Figma and cross link between Storybook and Figma for each asset.

Later

- Reintroduce external contribution to the design system
- Reorganise and resize design system team as necessary to support next phase of work

Contact details

This report was researched and written on behalf of [Frankly Design Ltd](#) by:

- Amy Hupe, Design Systems Consultant
amy.l.hupe@gmail.com
- Heydon Pickering, Design Systems Consultant
heydon@heydonworks.com

Please email us if you have any questions about its contents, or if you need our services to support the delivery of these recommendations.