



CONSTRUIRE  
3D DESIGN  
SYSTEM  
LE GUIDE **PAS-À-PAS**

Édition 2022



WORLD OF DIGITS

***Un Design  
System est un  
outil rationnel,  
fait sur mesure,  
qui répond à des  
besoins concrets.***



# Design System ?

Ces deux mots présents sur les lèvres de nombreuses équipes digitales font rêver beaucoup d'entreprises. En ces temps de transformation numérique, l'humain et la technique font équipe afin **d'avancer ensemble** et efficacement dans l'aboutissement des projets. La mise en place d'un Design System permet **cette synergie**.

Attractif à souhait, stratégique et complexe... Vous l'aurez compris, le Design System est LE sujet du moment. Le Gouvernement lui-même s'est emparé du sujet puisque l'État Français a créé son Design System (DesignGouv).

Mais quelle réalité se cache derrière ces termes ? En quoi consiste un Design System ? Est-ce véritablement utile et pour qui ? Autant de questions à se poser avant de se lancer.

Il faut en effet avoir bien en tête que mettre en place un Design System n'est pas chose aisée. Comme toute conception de produit, cela ne va pas de soi et résulte d'un **travail méthodique et rigoureux**. Ce travail nécessite préparation, **communication, organisation et structure**. Mettre en place un Design System impacte vos méthodes de travail ainsi que celles de vos départements digitaux tout entiers car il apporte un changement de **paradigme** global.

Un projet de Design System bien conduit vous offre de **nombreux avantages** : gain de temps et meilleure qualité de vos produits digitaux. Mais si au contraire, vous traitez le sujet à la légère, l'effet inverse risque vite de se produire...

## Un travail d'équipe sur un projet long-terme.

Partie prenante du projet, l'aspect humain doit être pris en compte car de nombreuses équipes et personnes vont travailler sur le projet. Toutes vivront un bel ascenseur émotionnel : échanges et désaccords, fatigue et enthousiasme, erreurs et solutions...

pour finalement **réaliser ensemble le Design System qui vous correspond, vous ressemble et vous rassemble.**

En bref : avoir entre vos mains **votre futur outil de conception** commun et quotidien !

Avec ce guide, nous vous proposons un document technique et holistique afin de vous aider à **construire concrètement votre Design System.**

Issu des enseignements tirés de nos multiples expériences sur le sujet, il repose autant sur nos succès que sur nos erreurs. Loin d'être une compilation de recettes miracles, il n'apporte pas non plus de réponse unique. Vous découvrirez qu'un Design System n'est en aucun cas générique mais qu'il **se construit sur-mesure** selon votre structure, vos équipes, vos besoins et vos contraintes.

Nous mettons donc ici à votre disposition un **document de travail didactique** qui vous permettra de vous poser les bonnes questions, choisir les bons processus et **avancer sereinement dans la construction de votre Design System.**



# Sommaire

<b>Un peu de théorie</b> .....	<b>9</b>
Qu'est ce qu'un Design System ? .....	11
De quoi est-il composé ? .....	11
Quels sont ses avantages ? .....	14
<b>Commencer son Design System</b> .....	<b>17</b>
Avez-vous besoin d'un Design System ? .....	19
Les questions à se poser, la capacité à mettre en place .....	19
Le voir comme un produit à part entière et non comme un projet ....	24
Définir la stratégie produit .....	26
<b>Structurer son Design System et son équipe</b> .....	<b>33</b>
Design System strict ou souple ? .....	35
Modèle de gouvernance : centralisé ou réparti ? .....	40
Constituer l'équipe et définir les rôles de chacun .....	43
La documentation, une étape importante ! .....	48
<b>S'organiser au quotidien</b> .....	<b>55</b>
Gérer son produit au quotidien .....	57
L'importance de communiquer .....	65
Travailler en équipe .....	72
La conception par composants .....	73
<b>Le déploiement et le faire vivre</b> .....	<b>83</b>
Le déployer dans votre écosystème .....	85
La gestion des releases .....	87
Mesurer l'impact .....	93
<b>Pour conclure</b> .....	<b>97</b>
<b>Glossaire</b> .....	<b>99</b>
<b>World of Digits, qui sommes-nous ?</b> .....	<b>115</b>
<b>Les contributeurs</b> .....	<b>117</b>





## Chapitre I

# Un peu de théorie

Avant de se pencher sur la construction d'un Design System, regardons ensemble ce que contient ce produit et en quoi il consiste.

“

***Un Design System est un outil évolutif et adaptatif qui regroupe tous les éléments utiles à la conception de produits numériques.***

”

## Qu'est ce qu'un Design System ?

Le Design System peut s'apparenter à une bibliothèque de ressources réutilisables : éléments graphiques, principes UX et composants développés, pour que designers et développeurs conçoivent ensemble et rapidement, des produits cohérents.

C'est un outil complet mis à disposition des équipes **pour concevoir efficacement des produits numériques cohérents entre eux.**

Cet outil contient une suite de livrables qui évolueront dans le temps, soit par l'arrivée de nouvelles technologies, de nouveaux frameworks, de nouveaux usages ou encore par l'évolution de votre marque ou de vos produits. Contrairement aux idées reçues, un Design System est un outil qui regroupe bien plus qu'une bibliothèque de composants (même si elle en est le cœur).

Utiliser un Design System, c'est embrasser un nouveau mode de fonctionnement et adopter un nouveau modèle organisationnel. Il ne suffit plus de penser ses produits numériques en termes de pages isolées ou de projets indépendants, mais bien en termes d'écosystèmes ; de réfléchir et concevoir ses produits de manière globale, à l'aide de composants transversaux.

Enfin, un Design System est, par définition, un **produit à long terme en perpétuelle évolution**, au même titre que vos infrastructures informatiques.

## De quoi est-il composé ?

Le but premier d'un Design System étant de faciliter le travail des équipes, il faut tout d'abord se questionner sur les besoins et exigences de ces dernières. Les outils clés contenus dans votre Design System seront donc les outils qui répondent aux besoins fonctionnels des équipes produits afin de leur permettre de concevoir leurs interfaces.

Cependant, afin d'être complet, un Design System doit au moins contenir :

### **Des objectifs**

Les objectifs seront à la base de votre ligne directrice, de votre vision et de vos stratégies. Ils doivent être parfaitement clairs et sans aucune ambiguïté.

Ils peuvent, bien sûr, être multiples comme par exemple contenir les objectifs de vos produits (être plus accessibles), de votre marque (créer une identité cohérente sur toutes les applications), etc.

### **Des valeurs**

Les valeurs sont les axiomes de votre Design System. Ce sont les éléments-clés à garder en tête lors de sa création. Elles guideront vos actions, la structure de vos fichiers, le choix de vos outils, votre communication et parfois même votre organisation.

En plus des valeurs de votre marque, votre Design System peut contenir les valeurs de votre équipe. C'est-à-dire les valeurs qui vont regrouper et fédérer votre équipe produit.

### **Les éléments d'identité de votre marque**

Tout ce qui compose votre marque : le logo, la typographie, les couleurs, les pictos, les illustrations, le Tone of Voice, votre stratégie de marque, etc.

Ils doivent être présents, accompagnés de leurs règles d'utilisation (vos guidelines) et adaptés à une utilisation numérique.

## Des principes de design

Cet ensemble représente un set de lois, de principes qui vont guider votre Design. Toute l'expérience autour de vos produits va dépendre de ces principes. Ils vont vous aider à prendre des décisions structurantes dans votre phase de conception.

Il existe des principes universels comme « la simplicité », « l'accessibilité » et d'autres qui seront plus spécifiques à votre produit.

## Une bibliothèque de composants

Les composants de vos interfaces représentent la partie visible de votre Design System. Ils sont présents sous forme graphique pour les designers et sous forme de code pour les développeurs. Une documentation fonctionnelle les accompagne.

## Des design Patterns

Un design pattern est un schéma de conception réutilisable qui résout efficacement un problème. Il prend en compte l'aspect graphique, technique et fonctionnel. Les documenter permet un gain de temps pour leurs utilisations futures et une homogénéité sur vos productions.

On note par exemple un pattern de connexion, un pattern d'erreur ou encore un pattern de feedback. La documentation représente le point névralgique de votre Design System. Elle contient tout le détail et les différents accès aux éléments définis plus haut.

## Une documentation

Le point névralgique de votre Design System : la documentation. Dans cette documentation se trouve tout le détail et les différents

accès aux éléments définis plus haut.

## Quels sont ses avantages ?

Que cela soit de manière directe ou indirecte, mettre en place un Design System apporte de nombreux avantages à vos équipes et vous fait réaliser des économies à long terme sur de nombreux points : des économies de temps, financières, d'échelle ou encore de main d'œuvre.

Grâce à cet outil, vous capitalisez sur ce qui est réalisé pour gagner en rapidité, performance et cohérence dans la conception d'un produit digital et / ou dans l'ajout de nouvelles fonctionnalités.

Le Design System met l'expérience utilisateur et le design au cœur de la conception. Il permet de faire travailler ensemble et plus aisément, des profils différents, dans la même direction.

Voici une liste de bénéfices que permet la création d'un Design System :

- **Une mise à l'échelle.**

Le Design System permet de produire plus de produits, plus de features sans augmentation exponentielle de l'équipe, ni sans devoir tout réinventer.

- **Une réduction des dettes Design et Technique.**

Grâce à la construction d'une bibliothèque de composants uniques.

- **Un partage de connaissances et une meilleure communication des valeurs de l'entreprise.**

Via une documentation, des principes et des valeurs partagées qui rendent possible une communication autour de la philosophie

et les bonnes pratiques de conception de design et de développement.

- **Une accélération des phases de conception.**

Notamment, des maquettes graphiques et du développement front des interfaces, ce qui permet de prendre «plus de temps» sur les phases de réflexion.

- **Une meilleure maintenance.**

Documentations, règles et assets étant centralisés, l'accès à leur modification/édition est facilitée. Aussi, l'homogénéisation des interfaces qui en découle permet une meilleure maintenance sur le long terme, maintenance aussi bien technique que graphique.

- **Une diminution du « facteur autobus ».**

Grâce à la documentation, à des standards et à une centralisation de l'information, le risque de perte de connaissances liées aux interfaces diminue. Il est alors plus facile de remplacer une personne manquante et un nouvel arrivant montera plus rapidement en connaissance sur le sujet.

Tout ceci limite l'impact du turn-over des équipes.

Pour conclure, un Design System est un outil, **vivant, agnostique, universel et inclusif** qui soutiendra autant la construction de vos projets que le maillage et les compétences de vos équipes.







Chapitre II

# Commencer son Design System

Construire son Design System représente un projet de grande ampleur. Il est alors primordial de se poser les bonnes questions et de s'interroger sur l'utilité de sa mise en place.

“

***La possession  
d'un Design  
System n'est pas  
une fin en soi.***

”

## Avez-vous besoin d'un Design System ?

Avant toute chose, c'est la question à se poser. **Elle est essentielle.**

En effet, toutes les entreprises, tous les projets ne nécessitent pas la construction d'un Design System. Parfois une simple librairie de composants développés, un kit UI (Interface Utilisateur), un bootstrap ou une charte graphique suffisent.

Construire un Design System ne semble pas justifié par exemple pour une simple landing page, ou une application temporaire.

Avant de vous lancer à corps perdu dans sa création, **prenez le temps** de vous questionner sur votre besoin réel, **sur la pertinence** de cet outil dans la résolution de vos problématiques.

**Interrogez-vous également sur votre volonté mais aussi sur votre capacité à le faire.**

## Les questions à se poser, la capacité à mettre en place

Vous avez répondu « oui » à la question précédente en concluant qu'un Design System serait une approche pertinente pour votre projet qui aiderait à la résolution de vos problèmes ?

Après la question primordiale « *ai-je besoin d'un design system ?* », viendront un ensemble d'interrogations (exposées ci-après) qui vous informeront sur votre capacité – ou non – à le mettre en place.

Les réponses dépendent de votre organisation, contexte, équipe ou de l'étape du projet à laquelle vous vous trouvez.

Bien entendu, non exhaustive, cette liste n'a pour but que de vous donner un ordre d'idées. Quoi qu'il en soit, essayez toujours d'être le plus honnête possible quand vous répondez à ces questions :

## Questions techniques

### **Mon environnement technique me permet-il de mettre en place un Design System ?**

Si cela n'est pas le cas, mettre en place un Design System, vous occasionnera au mieux de nombreux casse-tête, augmentant potentiellement la dette technique, ou démontrera au pire l'inadaptation de vos infrastructures pour la mise en place et / ou le bon fonctionnement d'un Design System.

### **Mon environnement technique est-il agnostique ?**

- S'il est agnostique, votre bibliothèque de composants front-end est déjà partagée par vos différents produits / applications / modules.
- S'il ne l'est pas, vous devrez faire en sorte de tous les réunir sous une bibliothèque commune, si tant est que cela soit possible.

### **Ai-je beaucoup de dette technique à résoudre ?**

Cela ne représente jamais une partie plaisante du travail de développement, mais une dette ignorée au départ devra fatalement être traitée en cours de route – ce qui survient généralement lorsqu'elle est devenue un obstacle trop évident pour être ignorée et qu'elle détériore la qualité de votre Design System et donc de votre produit final.

Appropriiez-vous ce problème en amont pour le traiter dans les meilleures conditions possibles.

## Questions liées à l'équipe et à l'environnement

### **Ai-je les ressources nécessaires (équipe, budget, temps) pour mener à bien la mise en place d'un Design System ?**

Vous trouverez, au fil de ce document, des exemples détaillés des ressources humaines et techniques qui vous seront nécessaires. Si l'un ou l'autre vous fait défaut et que vous ne pouvez développer correctement un Design System, des solutions plus légères peuvent vous permettre de vous approprier certains de ses bénéfices. Nous les mentionnons plus haut : une librairie de composants développés, un kit UI ou une charte graphique, selon vos besoins.

### **Mon équipe a-t-elle les compétences requises, et si non, peut-elle les acquérir ?**

Dans ce cas de figure, il faut mesurer l'étendue des connaissances à maîtriser, s'il est possible pour l'équipe de se former et si elle peut être totalement opérationnelle le moment venu.

Ce document liste de nombreux points à prendre en considération d'un aspect théorique (méthodologie du Design System) à pratique (mise en place et outils) et vous renseignera sur les compétences à développer.

### **Mon équipe comprend-elle la plus value du Design System ?**

Il s'avère bien entendu indispensable de présenter ce qu'est un Design System, ses caractéristiques et ses bénéfices : en d'autres termes, de faire preuve de pédagogie.

Il demeure essentiel que l'équipe soit engagée et moteur sur le sujet pour maintenir un haut niveau d'exigence sur la durée.

### **Pouvons-nous mettre en place les outils nécessaires au déploiement d'un Design System ?**

Vous devez identifier les outils dont vous aurez besoin, vérifier si

vous pouvez les adopter, et si oui, prévoir un temps de prise en main.

Anticipez aussi l'aspect administratif : l'abonnement à des logiciels non présents dans les catalogues conventionnels de votre entreprise peut s'avérer fastidieux !

### Questions liées au projet

#### **Le projet en cours est-il un projet structurant ?**

---

Si tel est le cas, s'engager dans un chantier de Design System en même temps que votre projet actuel compliquera très probablement la mise à bien du Design System, mais également votre projet. Ces deux sujets risquent ainsi d'être compromis.

#### **Une période de transition et/ou cohabitation entre le produit actuel et les nouveaux composants est-elle possible ?**

---

Vous avez ici deux possibilités :

Si oui, en procédant par itération : c'est-à-dire en implémentant progressivement les nouveaux composants dans le produit actuel. Vous pouvez faire le choix de conserver l'ancien style graphique des composants (ceci rendant la transition invisible pour l'utilisateur) ou de le changer si votre objectif est également une repasse graphique de votre produit. Cependant, les deux styles devront cohabiter.

Dans cette situation, deux versions de votre projet cohabitent pendant une période pouvant aller de quelques mois à plus d'un an. Si vous faites évoluer le graphisme des composants au fur et à mesure, cette solution ne sera pas la plus élégante mais elle n'immobilisera pas les évolutions de votre projet. De plus les changements apportés seront directement visibles pour votre équipe.

Si non, en procédant à une refonte totale du produit, en préparant en amont un Design System complet qui sera utilisé à cet effet.

Solution la plus coûteuse en ressources, elle nécessite soit une immobilisation du projet, soit une équipe travaillant sur le produit existant et une autre dédiée à sa refonte en parallèle. Néanmoins, cette solution réduit considérablement le risque de dette technique ainsi que tous les aléas de la cohabitation entre l'ancienne et la nouvelle version du produit. Cette solution permet à votre produit d'exister et d'évoluer sur des bases saines.

### **Est-il possible d'attendre un moment plus opportun pour sa mise en place ?**

Si vous rencontrez ces difficultés, mais que vous savez que leurs résolutions n'est qu'une question de temps, reporter à plus tard. Le chantier du Design System peut être une solution. Mettez ce temps à profit pour vous organiser, mettre en place vos outils ou encore communiquer autour de votre Design System.

### **Questions liées à l'accompagnement**

Si vous êtes et restez coincé sur l'un des points précédents, être accompagné peut vous permettre de débloquer certaines situations.

Si, par exemple, votre écosystème est mature mais que votre équipe ne l'est pas, vous pouvez être accompagné pour de la formation ou avec un intervenant spécialisé pour vous aider à mettre en place les éléments requis : connaissances fonctionnelles, techniques, création de composants, maîtrise des outils...

À l'inverse, si c'est votre écosystème qui n'est pas au point, des ressources externes peuvent vous aider sur des questions d'organisation et de méthodologie.

## Le voir comme un produit à part en et non comme un projet

Nous voyons la mise en place d'un Design System comme la création d'un nouveau Produit (et non comme un projet avec une date de fin). Comme tout Produit, il doit répondre aux besoins d'une cible donnée (développeurs, designers, etc...). Il doit également apporter de la valeur dans la durée.

De plus, par définition, un Design System évolue avec le temps, il n'est jamais fini. Une fois la première version déployée, il entre ainsi dans des cycles itératifs continus.

En effet, **nous ne le construisons pas pour nous mais pour des utilisateurs**. Et ces utilisateurs sont les membres de vos équipes. Ce sont eux qui manient au quotidien les éléments du Design System.

De plus, il ne se limite pas à la conception de composants : il faut aussi savoir gérer le produit, le marketer, connaître ses utilisateurs et leurs besoins. C'est pour cela qu'il faut changer de perspective et le penser comme un **produit à part entière de votre organisation**. Un produit transversal, au service de vos équipes de conception.

Dans ce contexte, il ne faut surtout pas le considérer comme un projet secondaire pouvant s'intégrer dans un lot d'autres projets. Il ne faut pas non plus l'envisager comme un projet de conception à caler 6 mois dans un calendrier, sans itérations, et encore moins espérer qu'il se construise sur le temps libre des équipes ou entre deux projets.

**Non, c'est un produit qui doit refléter votre entreprise et évoluer avec elle.**

Un Design System doit posséder :



## Une vision produit

C'est-à-dire la réponse opérationnelle et la stratégie mise en place pour le construire. Cette vision produit correspond aux questions suivantes :

- Quels problèmes résolvez-vous avec le Design System ?
- Qui est la cible de notre Design System ?
- Comment allez-vous construire le Design System ?
- Comment allez-vous lancer le Design System ?
- Comment allez-vous mesurer le succès du Design System ?
- Comment allez-vous vendre le Design System ?

## Un rythme et une cadence dédiée

Nous vous conseillons d'utiliser des méthodes agiles (SCRUM notamment) pour instaurer un rythme de travail.

Découpez votre roadmap en plusieurs Sprints avec les cérémonies adaptées à votre équipe. Cela vous aidera à cadencer votre produit et à suivre plus facilement son développement.

## Une équipe produit

Qu'elle soit uniquement dédiée au Design System ou issue de plusieurs équipes, vous devez nommer une équipe garante et responsable du produit (PO, PM, designers, développeurs par exemple). Indispensable, ces personnes seront les garants en charge de maintenir cohérence et harmonie du Design System. Il accompagne les équipes dans l'utilisation de leur Design System, centralise les retours et gère les discussions et partages entre tous les corps de métier.

## Un espace de gestion de projet

Nous vous déconseillons de mélanger les tâches du Design System à celles de vos autres projets. Que votre outil soit JIRA, Asana, coda ou autre, le produit Design System doit avoir son propre espace, avec son propre Backlog, ses propres tâches, users stories et sprints.

## Une documentation dédiée

Sujet traité un peu plus tard dans ce guide, mais la documentation liée à votre Design System ne doit pas se mélanger avec vos autres documentations existantes.

Peu importe l'outil choisi, celui-ci doit posséder un espace consacré à la documentation de votre Design System.

## Définir la stratégie produit

Votre **stratégie produit** représente véritablement la première brique de votre Design System. Elle en pose les bases, va servir à planifier sa construction et à constituer votre équipe dédiée. Vous devez envisager l'élaboration de votre Stratégie comme une étape-clé de votre produit.

Prenez donc le temps nécessaire à son élaboration.

Plusieurs mois peuvent ainsi s'écouler entre le moment où la construction du produit Design System est actée et le début de sa conception.

Pour vous aider, voici les thématiques auxquelles votre stratégie doit répondre :

***Allez-y étape  
par étape,  
construire un  
Design System  
n'est pas un  
sprint mais  
une course de  
fond !***

## Les utilisateurs du Design System

Réalisez des entretiens avec les leaders et les contributeurs-clés, échangez autour de la culture et de la philosophie de conception des équipes. Comprenez leurs besoins, leurs outils, leurs priorités... Voyez ensemble quelles sont les ressources temps qu'elles peuvent accorder sur ce produit.

Définissez vos utilisateurs directs et indirects :

- Qui va l'utiliser (designers, développeurs...) ? Dans combien d'équipes ? Seulement des internes ou aussi des externes ?
- Sur quels types de supports ? Avec quelles technologies et quels devices ?
- À qui devez-vous vendre le Design System ?
- Qui sont vos sponsors ?
- Qui seront vos «early adopters» ?
- Quel est votre plan d'adoption ?

## Les objectifs du Design System

- Que souhaitez-vous accomplir avec le Design System ?
- Quels problèmes souhaitez-vous résoudre ?
- Que comptez-vous améliorer ?

Bien souvent, les objectifs répondent aux problèmes existants.

## L'équipe produit

Définissez votre mode de gouvernance ainsi que les membres de l'équipe qui deviendront les piliers de votre Design System. La construction d'un tel produit demande beaucoup de temps et d'énergie à votre équipe.

Pensez donc à ne pas solliciter des personnes déjà surchargées, ou alors attribuez-leur plus de temps (moins de projets en parallèle, sprints plus longs, etc.).

## Les outils

- Avec quels logiciels allez-vous concevoir vos composants ?
- Sous quel framework allez-vous les développer ?
- Avec quel outil allez-vous suivre le projet, le documenter ?

En bref, définissez bien en amont vos outils car ils impactent fortement le produit. Au cours de la phase de construction de votre Design System, un changement d'outil peut augmenter drastiquement la complexité et le temps passé.

## L'approche

Faut-il s'y atteler de manière incrémentale, en y allant lot par lot ou bien au contraire tout faire d'un coup ?

De votre approche, découlent votre roadmap et votre organisation, comme le découpage en différentes versions, le contenu de celles-ci, les milestones, etc.

Il importe donc de se poser les bonnes questions :

- De quoi avez-vous besoin tout de suite (prioritaire) ?

- Qu'est-ce qui peut attendre demain et après-demain ?

Votre stratégie Produit doit pouvoir expliciter clairement et simplement ce qu'est et ce que deviendra votre produit Design System. Elle vous servira à communiquer auprès de vos stakeholders, à tenir vos engagements envers votre équipe et à définir une vision commune et partagée de tous.

Présentez clairement votre stratégie à l'aide d'un support contenant :

- La définition d'un Design System ;
- Les problèmes existants et les réponses apportées par le design System ;
- Votre roadmap produit (macro) avec un calendrier précis et son périmètre ;
- La composition de votre équipe et son organisation ;
- Des exemples de Design System dans des secteurs proches ou identiques à votre organisation.







## Chapitre III

# Structurer son Design System et son équipe

Après avoir vu l'aspect théorique avec la méthodologie du Design System, place à la pratique avec sa mise en place et les outils à utiliser.

**“  
Comme  
tout produit  
numérique, votre  
Design System  
se structure et  
se construit en  
équipe.”**

## Design System strict ou souple ?

Tous vos produits numériques doivent-ils partager les mêmes normalisations, les mêmes principes d'expériences, les mêmes interfaces ?

Les réponses à ces questions permettent de savoir où situer votre gestion du Design System, entre l'un ou l'autre de ces deux modes de fonctionnement, strict ou souple.

### La gestion stricte

Le Design System très cadré et directif : il laisse peu de place à l'expérimentation et l'improvisation des équipes projet. Tout y est très documenté et les processus de validation y sont très forts, pour chaque évolution ou introduction d'un nouveau composant / pattern.

En conséquence : il n'existe pas de variation visuelle entre vos produits, tout est identique.

**Le point positif ?** C'est un Design System très exhaustif : il contient tous les cas possibles.

**Le point négatif ?** Les designers des équipes peuvent se sentir guidés à l'extrême dans la construction de leurs patterns et se sentir submergés face à une documentation conséquente (s'il en résulte une lourdeur / difficulté d'utilisation, un manque d'adhésion est à craindre

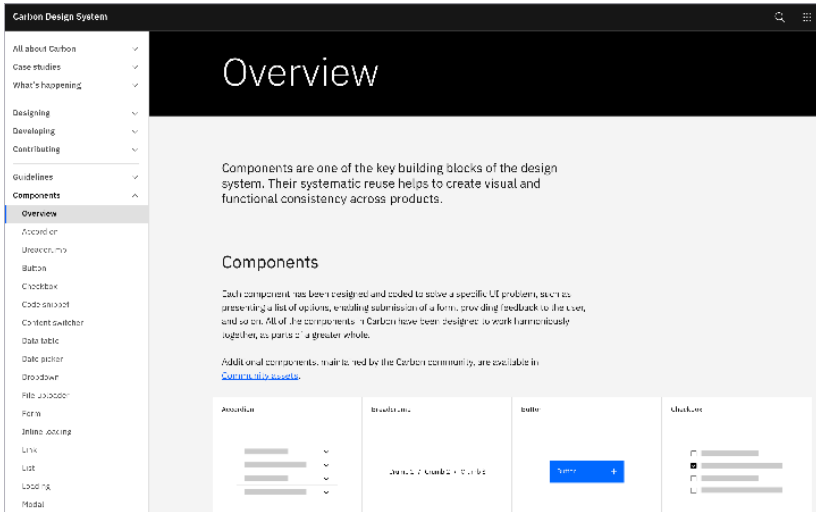
### Quand le privilégier ?

.....

Cette gestion s'avère s'avère particulièrement pertinente lorsque vous avez besoin de dupliquer un produit identique sur plusieurs plateformes. Ou quand vos équipes présentent des maturités très variables et ainsi qu'un fort turn-over.

## Quelle référence pour mieux l'appréhender ?

**IBM Carbon design system**, qui propose une documentation très complète de l'usage de chaque composant : un guide sur leur style, leur usage, leur code, et aussi des règles d'accessibilité.



*IBM Carbon design system, un Design System très complet balisant le maximum de besoins possible.*

## Qu'est-ce qu'une gestion souple/flexible ?

Bien que rigoureux, ce système laisse plus de libertés, plus de place à l'expérimentation pour les équipes qui l'utilisent, quant à l'usage des composants et à la construction des patterns d'interface.

À l'extrême, il est même possible d'avoir un système-mère et plusieurs systèmes-enfants (le système-enfant peut, par exemple, consister en un « thème UI » différent afin de donner un autre look à certaines interfaces).

Il s'avère ainsi possible de différencier visuellement des interfaces d'applications métier (notamment, par une couleur). Cela peut faciliter leur reconnaissance par des employés qui auraient à jongler

entre plusieurs applications à l'esthétique semblable.

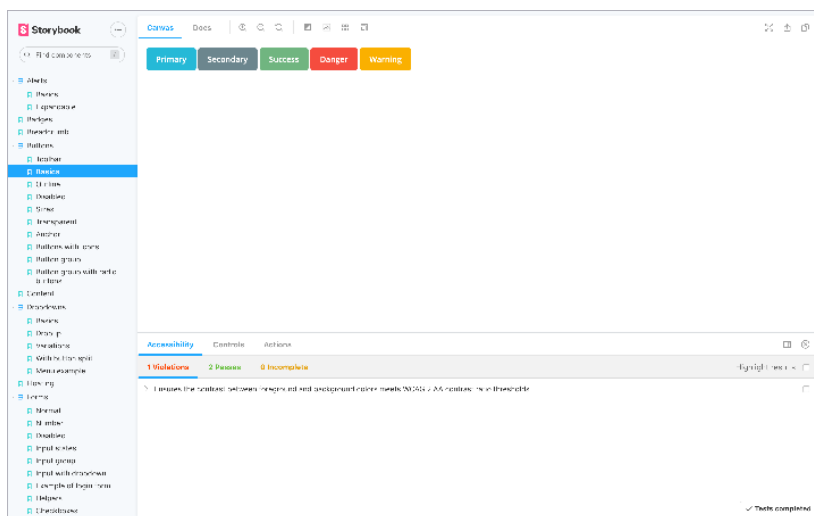
**N.B.** La liberté offerte par un Design System souple ne doit pas pour autant amener les équipes à créer ex nihilo des composants arbitrairement. Il faut alors rappeler que l'équipe référente se trouve à leur disposition pour les conseiller.

## Quand le privilégier ?

- Pour poser la première pierre de votre Design System ou pour des écosystèmes dont les produits doivent être très différents les uns des autres.
- Quand la maturité organisationnelle de l'entreprise ne permet pas d'imposer un système strict aux équipes.

## Quelle référence pour mieux l'appréhender ?

**Le kit UI de Prestashop™** qui, à ce jour, propose des composants sur Storybook sans en définir l'usage.



*Boutons de l'UI Kit Prestashop : une présentation simple des composants, sans documentation.*

## Comment choisir ?

Bien entendu, la structure de votre Design System peut être plus ou moins rigide ou souple. Vous choisissez où placer le curseur, selon vos besoins et vos capacités.

Voici quelques critères à prendre en compte pour vous aider dans cette décision. Chacun d'eux orientera plus ou moins votre choix vers strict ou souple.

### Le budget

**Budget important ?** Strict : documentez en détail tous les cas possibles.

**Budget serré ?** Souple : concentrez-vous sur la version MVP. Réalisez le minimum de composants nécessaire pour que votre Design System soit utilisable. Les templates, documentations ou recommandations UX viendront alors dans un second temps.

### Le temps

Si vous manquez de temps, visez là aussi le MVP avec le minimum de composants nécessaires pour que votre Design System soit directement utilisable par les équipes projets.

### La maturité de l'organisation

L'écosystème de l'entreprise s'avère-t-il favorable au développement de projets structurants ? Les porte-t-il (les impose-t-il) auprès des usagers ? Ou au contraire, faut-il se battre pour que ces projets soient utilisés ?

Les projets des équipes métiers sont-ils suffisamment concernés et engagés dans une optique d'écosystème d'entreprise ? Ou est-ce que l'usage est plutôt de faire « cavalier seul » et développer des projets seul, ex nihilo et avec un style original pour avancer sans contrainte ?

Si ces voyants sont au rouge, le Design System souple sera à privilégier.

Avant d'envisager une gestion stricte, il vous faudra faire campagne auprès de la hiérarchie afin qu'il s'inscrive dans la politique d'entreprise, sous peine de ne pas être utilisé.

### **La maturité des équipes (designers + développeurs)**

Un Design System strict demandera plus de travail et de qualité de la part de l'équipe Design System, mais cadrera d'autant plus les équipes projets.

À noter que nous parlons aussi bien de maturité sur le sujet Design System que de compétences dans leurs domaines respectifs.

### **Un turn-over fréquent des équipes**

Réalisez un Design System suffisamment strict pour guider et encadrer les nouveaux arrivants.

Travailler avec des prestataires (nombreux) peut aussi • conduire à une gestion stricte pour limiter les multiples apports d'intervenants externes (plus enclins à amener de la nouveauté et se sentant moins liés à la durée de vie d'un projet).

### **Des équipes multiples**

Plus les intervenants sont nombreux, plus le curseur doit tirer vers le strict sous peine de voir autant de versions différentes en ligne pour un même composant.

### **Des marques / produits multiples**

Un Design System souple leur donnera la latitude nécessaire pour se différencier des autres produits.

Un projet multi-produits réussi avec un Design System souple peut aussi aboutir à définir des normes de cadrage vers un Design System plus strict pour des projets semblables à venir.

## Une marge de manoeuvre stratégique/fonctionnelle

---

Le contexte politique dans lequel se développe votre Design System peut vous amener à le développer de manière plus ou moins importante.

Plus il cadrera de cas pratiques et plus son périmètre augmentera. Au point même qu'il pourrait bien empiéter sur le périmètre d'autres services. La phase de cadrage prend alors toute son sens afin de se prémunir de ce type de situation !

Il n'est pas nécessaire de « formaliser » que votre Design System est strict ou souple. C'est une direction donnée par l'équipe qui le construit, afin qu'elle augmente ou diminue les informations que le projet devra comporter.

## Modèle de gouvernance : centralisé ou réparti ?

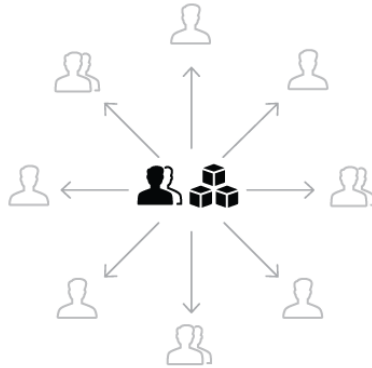
Choisir son mode de fonctionnement représente l'une des étapes-clés pour garantir une implémentation efficace d'un Design System. En ce sens, il importe de réfléchir en amont au modèle de gouvernance à mettre en place.

### Le modèle centralisé

Avec un **modèle centralisé**, une seule équipe dédiée gère le design system.

Cette équipe s'occupe de construire le système et de le diffuser au plus grand nombre. Elle s'occupe de prendre les décisions sur ce qui doit être intégré ou non, définit la vision et la Roadmap à suivre.





Ce modèle à plusieurs avantages :

- Une équipe à temps plein sur le sujet ;
- Moins de risque d'épuisement des équipes et donc un meilleur maintien des efforts de conception sur le long terme ;
- Une meilleure facilité dans la mise en place des processus et pratiques de conception ;

Mais présente aussi ses inconvénients :

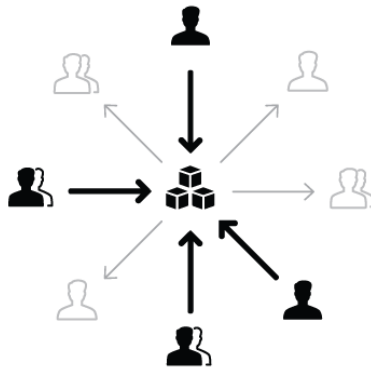
- L'équipe peut manquer de contexte sur les problématiques réelles des équipes produits ;
- Une appropriation du produit et donc biaiser sa conception ;
- Un isolement de cette équipe dû a un manque de pouvoir pour faire participer d'autres acteurs dans la conception du système.

### Le modèle réparti

Avec le **modèle réparti** des membres de plusieurs équipes produits gèrent le Design System.

Ils sont chargés du Design System et prennent ensemble les décisions le concernant.

Ce qui diffère avec le modèle centralisé, c'est que les membres de cette équipe ne sont pas à « temps plein » sur le sujet, ils continuent de travailler sur leurs produits respectifs.



Ce modèle a plusieurs avantages :

- Limite les biais de conception car ce sont les membres des différentes équipes produits qui conçoivent le Design System ;
- Facilite la diffusion du système au sein des équipes ;
- Ce modèle rend plus naturel les échanges autour de la conception.

Mais présente aussi ses inconvénients :

- Augmente les risques d'épuisement des membres car ils doivent gérer leurs produits en plus du Design System. Ce qui peut provoquer un abandon des équipes sur le sujet ;
- Les différentes équipes peuvent avoir des avis contradictoires et

être confuses sur ce qui doit être fait.

### Mais alors lequel choisir ?

Cela va dépendre d'éléments comme la quantité de produits que vous possédez, le nombre d'équipes associées et de personnes dédiées au Design System, ainsi que de vos objectifs. Il s'avère même fort possible que vous finissiez par adopter une variante qui deviendra votre propre modèle. Il n'existe pas de formule-clé.

Il peut être parfois compliqué de trouver un modèle de gouvernance idéal, nous vous conseillons de commencer par un atelier de type «Définition des rôles» pour mettre en lumière les différents rôles nécessaires à votre Design System.

## Constituer l'équipe et définir les rôles de chacun

Un Design System étant un produit, il requiert des compétences très horizontales.

Nous définirons l'équipe selon 2 aspects :

- **Les intervenants** : des représentants des métiers et compétences nécessaires à la réalisation d'un Design System. Ce sont des rôles fonctionnels.
- **Les rôles à endossement** : des fonctions décisionnelles indispensables au bon fonctionnement du projet.

À noter que 2 rôles peuvent parfois être tenus par une seule et même personne, en début de projet par exemple.

Il est fort possible que, parmi vos équipes, il y ait des profils possédant différents domaines d'expertises. Ainsi, un designer pourra endosser le rôle du P.O. tant que le sujet est à l'état de P.O.C. En effet, un designer sachant coder peut réaliser les premiers composants ou encore un designer UX/UI peut pleinement utiliser sa double compétence...

Capitalisez dessus au début ! Mais n'oubliez pas que cela apporte un travail plus important pour ces personnes : il faudra dans tous les cas les décharger de leurs expertises secondaires pour garantir une pérennité au Design System.

## Composition de l'équipe : les intervenants

### L'équipe principale

#### **Designer UI** (*obligatoire - Dès le début de projet*)

- Définit et assure le style graphique du Design System. Garantit le respect de l'identité de marque aux composants.
- Conçoit les composants et leurs déclinaisons destinés à la librairie de composants, les micro-interactions, etc.
- Garant de la qualité/fonctionnalité graphique des composants.
- Réalise des templates.

#### **Designer UX** (*obligatoire - Dès le début de projet*)

- Définit le cadre d'utilisation du produit et ses utilisateurs.
- Assure l'utilisabilité, la fonctionnalité et l'accessibilité (absolues et contextuelles) du produit.
- Définit les patterns.
- Facilite l'intégration des utilisateurs finaux du projet et leur compréhension.

- Teste les réalisations auprès des utilisateurs pour s'assurer de leur adéquation.

### **Dev front-end** (*obligatoire - Dès le début de projet*)

- Développe les composants UI conjointement avec les designers.
- Teste la bonne intégration technique des composants dans les différents éco-systèmes.

### **Les guests** .....

**Scrum master** : La méthodologie de travail étant Agile, les réalisations s'organisent par sprint. La présence d'un Scrum Master peut donc s'avérer bien utile, au moins en début de projet. Il peut ne pas être dédié uniquement à ce projet mais garantira le process agile et ses cérémonies.

**Directeur artistique** : Garant de l'image de la marque, il amène ses compétences sur l'aspect graphique et interactionnel.

**Copy Writer / U.X. Writer / Content Manager (SEO)** : Le ton employé par la marque est une partie identitaire importante et présente à chaque écran, il est donc vivement conseillé d'intégrer ces profils dans la construction de votre Design System.

**Expert accessibilité** : Il garantit l'accessibilité des produits réalisés (respect des engagements signés le cas échéant).

**Dev Ops** : Gère la migration des versions (jonction avec la production).

**Design Ops** : S'assure entre autre de l'organisation, fait le lien entre les différentes équipes, de la migration et du versioning côté Design.

**Designer de Service** : Son cœur de métier étant la construction de service, une telle ressource peut s'avérer très utile s'il en existe une dans la structure. Ce designer peut intervenir en début de projet pour aider à la mise en place de la méthode ou en aval sur l'appropriation

du Design System par les équipes et sa mise en place.

### Rôles à endossement et Hiérarchisation des rôles

Les rôles à endossement sont des responsabilités à attribuer aux différents membres de l'équipe, pour assurer le bon fonctionnement du projet. Plusieurs rôles peuvent être endossés par une même personne.

Si plusieurs métiers sont représentés dans l'équipe (exemple : 2 UI designers), un Lead doit être désigné pour arbitrer les décisions et veiller à l'homogénéisation des procédés.

#### Design System Manager

- Gère le spectre d'application du Design System (Exemple : Quels services / branches du groupe sont concernées ? Servira-t-il au web uniquement ou aussi pour les applications ?) ;
- Arbitre les choix stratégiques ;
- Liste et priorise les réalisations ;
- Responsable de la bibliothèque de composants et de ses mises à jour ;
- Veille à ce que les designers appliquent les règles définies (couleurs, typos, conception de templates et composants, méthodologie, etc) ;
- Arbitre les différents concernant le design visuel ;
- Contributeur de la documentation (librairie de patterns) ;
- Définit les outils design.

## Product Owner

---

- Veille au bon déroulement du développement des composants du Design System ;
- Veille au respect des délais ;
- Juge de la conformité/qualité du produit fini (User Stories, KPI, pré-requis, etc) ;
- Assure le lien entre les équipes.

## Lead développeur

---

- Veille aux bonnes pratiques de développement des composants du Design System ;
- Instruit la documentation du code dans le Design System ;
- Définit la stratégie de conception et les choix techniques de développements (langages, outils) ;
- Arbitre les différents sur le sujet développement.

Nous vous conseillons de commencer le travail de Design System avec une équipe plutôt orientée conception (designer(s) + développeur(s)).

Cela permet d'apporter de la valeur sur les choix réalisés en design et en développement. Puis, petit à petit, ajoutez des membres, comme un Product Owner pour la partie gestion et communication, un Design Ops / Dév Ops pour les parties livraisons du produit, un responsable produit, etc.

## La documentation, une étape importante !

La documentation, bien que souvent délaissée car jugée chronophage et peu consultée, représente l'un des piliers du bon fonctionnement de votre Design System.

Celle-ci comporte des bénéfices évidents :

- Elle répertorie ce qui est fait de ce qui reste à faire ;
- Définit les usages d'un composant, ses comportements, déclinaisons et caractéristiques graphiques, ergonomiques ou encore techniques ;
- Offre un **répertoire commun** qui définit les règles du Design System et non une utilisation basée sur une expérience commune transmise oralement qui tend à différer avec le temps ;
- Elle permet l'historisation du travail accompli et évite les doublons, débordements, et exceptions ;
- Elle peut être l'opportunité de jumeler les sujets de code, sujets graphiques et fonctionnels sur un même support, disponible pour tous (non-designers et non-développeurs compris).

En plus de ces avantages et dans des organisations où les processus de validation ne sont pas toujours respectés, elle permet également de **valider officiellement** les composants et leurs caractéristiques.

Elle grave ainsi dans le marbre la nomenclature adoptée : maintenant qu'un composant est répertorié sous un nom précis, cela devient son nom d'usage et évite les malentendus.



Il est important que votre documentation soit la plus claire et explicite possible : elle ne doit laisser aucun doute quant à l'usage des composants. Dès lors, évitez d'utiliser un langage trop hermétique, du jargon trop spécifique ou des tournures trop littéraires. Allez droit au but !

### Qui documente ?

Nous vous conseillons de commencer le travail sur votre Design System avec une équipe orientée conception : Designer(s) et Développeur(s). Cela apporte une vision plus pragmatique sur les choix réalisés en design et en développement. Cette façon de travailler permet également de se concentrer sur les besoins initiaux du Design System. Ce sont ces mêmes personnes qui documenteront le Design System, selon leur domaines d'expertises :

- Les designers documentent la structure, l'anatomie et les comportements des composants, les règles d'accessibilité, les indications à respecter et les interdits. Si besoin, ils peuvent également documenter leurs fichiers de design ;
- De la même manière, les développeurs documentent les spécifications techniques des composants, la manière de les implémenter et documentent leur code au besoin.

Tout comme pour la nomenclature, réfléchissez bien à la langue que vous utiliserez pour votre documentation. Par exemple, si vos équipes avec lesquelles vous intervenez sont à l'international. Ou si vous communiquez votre documentation à l'extérieur, à l'audience que vous pouvez toucher.

## Que documenter ?

Votre documentation contient plusieurs niveaux et éléments.

En voici une liste non exhaustive :

- Une section « **Comment utiliser le Design System ?** » qui comporte vos principes de Design, vos valeurs, la notice d'utilisation et d'import du Design System, les éléments nécessaires aux designers et développeurs. Et tout ce que vous trouvez utile pour permettre une bonne communication et utilisation de votre Design System ;
- Une section **Branding**, qui contient les éléments de votre marque, comme le « Tone of Voice » ou les logos par exemple ;
- Une ou deux sections liées à vos **composants et à vos fondamentaux** (typographies, couleurs, grille, iconographie, etc). C'est le cœur de votre documentation, n'hésitez donc pas à être exhaustif ;
- Une section listant **les évolutions** de votre Design System à chaque évolution mise en production ;
- Une section dédiée à vos **patterns**, qui contient le fonctionnement d'ensemble des patterns utilisés ;
- Une section **outils et ressources**, fournissant tous les éléments utiles comme les liens vers des documentations externes ou des fichiers à télécharger ;
- Nous vous conseillons aussi d'avoir une section **contact/support** pour préciser les modalités de contacts et retours / demandes à faire auprès de l'équipe Design System.

Présentez clairement votre stratégie à l'aide d'un support contenant :

- Son rôle ;
- Ses cas d'usages, ses DO et DON'T ;
- Son anatomie, sa construction ;
- Son fonctionnement, sa démo technique, son code ;
- Ses règles d'accessibilité ;
- Les micros-interactions.

Bien sûr cette organisation est propre à chaque Design System, son contenu et sa densité varient en fonction de vos besoins. Vous pouvez vous aider de la fiche atelier « Structurer sa documentation » en page X pour organiser votre documentation.

### Où documenter ?

Quel est le support de documentation le plus adapté à vos besoins ?

Ce choix dépendra de la façon dont vous souhaitez accéder - et donner accès - à cette documentation. Qui la rédige ? Qui en est le destinataire ? Comment sera-t-elle mise à jour ?

Mesurez votre audience et adaptez le support en conséquence : la documentation est-elle destinée à l'équipe, à de multiples services ou à l'entreprise entière ? Le Design System sera-t-il également communiqué à l'extérieur ? Selon le cas, votre documentation ne sera plus seulement un outil technique mais un support de démonstration, voire de communication (par exemple s'il vous prenait l'envie d'ouvrir votre Design System au public via [www.designsystems.fr](http://www.designsystems.fr)).

Pensez aussi à la maintenance et à l'évolution du support de documentation, si le projet prend de l'ampleur ou si votre structure est amenée à changer. Enfin, interrogez-vous sur la propriété du

document, et de ce fait, sur qui l'héberge, si vous jugez que vos données sont sensibles.

Tout comme pour la nomenclature, réfléchissez bien à la langue que vous utiliserez pour votre documentation. Par exemple, si vos équipes avec lesquelles vous intervenez sont à l'international. Ou si vous communiquez votre documentation à l'extérieur, à l'audience que vous pouvez toucher.

Soyez vigilant quant au choix du support pour éviter tout problème de versioning. Vous ne souhaitez pas que différentes versions de la documentation soient prises comme références par les différentes parties prenantes du projet. Tout d'abord pour les problèmes de production que cela générerait, mais aussi pour la perte de légitimité du document de référence.

**Attention** : les droits d'accès d'une solution qui engage des secteurs sensibles d'une entreprise (comme Confluence) peuvent empêcher des utilisateurs extérieurs à votre organisation de consulter votre documentation. Ces droits d'accès sont généralement définis par des politiques d'entreprise qui impliquent tout un département (par exemple une DSI). Et cela peut s'avérer extrêmement difficile, voire impossible d'avoir des passe-droit, si d'aventure vous décidiez de le partager).





## Chapitre IV

# S'organiser au quotidien

Pour suivre au quotidien la construction d'un Design System, une bonne organisation s'impose. Des processus rigoureux, une bonne communication et une cadence définie vous aideront à en maîtriser les multiples aspects.

***Une bonne  
organisation  
et une  
communication  
fluide sont des  
critères de  
réussites de votre  
Design System.***



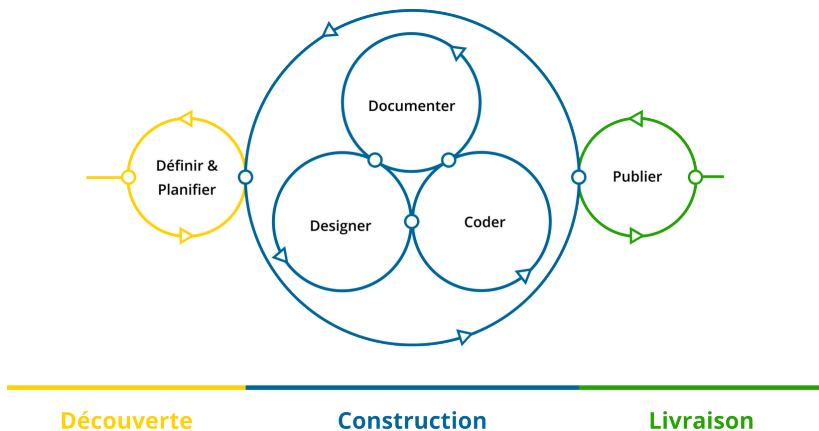
## Gérer son produit au quotidien

Les tâches qui accompagneront la gestion de ce produit sont les mêmes que pour n'importe quel autre : vous devrez identifier, prioriser, décider, découper, produire, recetter, livrer et recommencer !

### Le workflow

Le processus de construction se divise en 3 grandes phases : une phase de Découverte, une phase de Construction et une phase de Publication / Livraison.

Ces phases, elles-mêmes, peuvent être décomposées en 5 étapes reprenant les éléments-clés d'un découpage en User Stories ou en tâches, pour une bonne gestion au quotidien



### 1. Définition / Planification

Première étape dans laquelle vous allez étudier le besoin du composant, définir son périmètre, son coût, ses déclinaisons. Cela va passer par l'audit de l'existant, par des échanges avec

les parties prenantes, des ateliers, un découpage en US et de la planification.

En bref, tout ce qui s'avère utile au bon déroulement de la construction du composant.

**Membres concernés :** PO - PM - UX - UI - développeurs.

**Exemples d'activités :** Interviews, audit, analyse, création d'US, priorisation, etc.

**Exemples de « Definition of Done » :**

- L'élément étudié possède son utilité dans le Design System et son intégration y fait pleinement sens ;
- Toutes les US du sprint prochain sont prêtes et chiffrées.

## 2. Conception UX et UI

.....

Étape durant laquelle sera construit le composant avec ses variantes, ses différents états ainsi que ses interactions.

**Membres concernés :** UI - UX.

**Exemples d'activités :** Pair design, benchmark, review, présentations, etc.

**Exemples de « Definition of Done » :**

- Le composant et toutes ses variantes sont construits et validés par le Lead ;
- La conception proposée du composant résout le problème pour lequel il est défini.

## 3. Développement

.....

Cette étape correspond à celle où le composant va être codé et testé.

**Membres concernés :** Développeurs - QA.

**Exemples d'activités :** Pair développement, Tests Unitaires, review, présentations, etc.

**Exemples de « Definition of Done » :**

- Le code du composant correspond aux critères de développement et de qualité et se trouve validé par le Lead ;
- Le composant se conforme à sa spécification graphique et au fonctionnel défini.

#### 4. Documentation

.....

Cette documentation ne doit pas être confondue avec celle servant de spécification pour le passage en développement. Nous parlons ici de documentation d'usage du composant. Cette documentation explique comment l'utiliser et ce que l'on peut faire avec.

En somme, il s'agit ici de mettre en place une documentation donnant le bon niveau de compréhension à vos utilisateurs finaux afin qu'ils puissent utiliser votre Design System.

**Membres concernés :** UX - UI - développeurs.

**Exemples d'activités :** Présentations, review etc.

**Exemples de « Definition of Done » :**

- La documentation du composant est considérée comme complète et le niveau d'information jugé suffisant pour permettre d'utiliser le composant.

#### 5. Livraison

.....

Cette dernière étape présente différents types de livraisons :

- Une livraison design, soit l'intégration du composant dans la bibliothèque UI ;

- Une intégration du composant développé dans la bibliothèque (sous react, angular ou autre) ;
- Des publications de documentations comme celle concernant les règles d'usage du composant pour la partie Design ou encore le storybook pour la partie technique.

**Membres concernés :** UX - UI - développeurs.

**Exemples d'activités :** Migration, gestion de versioning, merge, review etc.

**Exemples de « Definition of Done » :**

- Le composant est ajouté dans l'outil FIGMA afin que l'équipe Design puisse l'utiliser sans souci ;
- Les branches git sont fusionnées et le composant s'inscrit dans la bibliothèque.

Bien que cela puisse sembler linéaire, comme une réalisation étape par étape, il est en fait très courant que les tâches se chevauchent voir se parallélisent, notamment dans la phase de construction.

Prenez le temps de définir les DoR (Definition of Ready) et les DoD (Definition of Done) de chaque étape, cela vous aidera à mieux vous structurer par la suite.

## Utiliser l'agile

Utiliser intelligemment les méthodes agiles vous permettra de **maîtriser l'avancée** de votre projet au quotidien.

En effet, vous ne construirez pas un Design System complet en une seule itération. Vous allez devoir séquencer et cadencer sa création. Les méthodes Agiles se révèlent très efficaces dans ces cas-là (et généralement déjà en place dans les entreprises et connues ou

pratiquées par vos collaborateurs.)

Regardons comment intégrer le workflow présenté précédemment, dans un fonctionnement en sprints.

### **Le découpage en sprints**

---

Avant de vous lancer dans le découpage en sprints, prenez le temps de réfléchir à votre V.alpha (ou Minimum Viable Product – « MVP » –, le terme importe peu) et à ce que vous souhaitez y intégrer.

Servez-vous d'un Sprint 0, avec une durée plus importante que celle de vos sprints à suivre. Celui-ci vous permettra de mettre en place vos outils, construire vos Users Stories, découper le contenu de votre V.alpha et l'intégrer dans les sprints futurs. Il vous aidera aussi à définir votre cadence (combien de sprints sont utiles pour sortir la V.alpha ? Quelle sera leur durée ? leurs objectifs ? etc).

### **Remplir les sprints**

---

Il va vous falloir un outil de gestion afin de créer des sprints, gérer un backlog, associer des tâches etc. Outil très puissant, JIRA convient parfaitement. Il est d'ailleurs souvent déjà utilisé dans les entreprises, mais en fonction de votre organisation, de la taille de votre équipe et de vos besoins, vos outils peuvent varier. Votre façon de structurer, de découper et de cadrer se révèle bien plus importante que l'outil en lui-même.

### **Mettre en place des rituels**

---

Daily, Retrospective, Grooming, démo de sprints etc. Ces rituels sont importants pour communiquer sur l'avancée du produit.

### **Structurer votre board**

Le board de votre outil de gestion de projet (JIRA, Asana, Trello ou tout autre outil) représente un allié précieux dans le suivi et la gestion

de votre Design System. Prenez donc le temps de bien le structurer et de l'organiser. N'oubliez pas que c'est un outil qui sert à l'équipe et pas uniquement aux PO / PM. Faites des ateliers avec les membres de l'équipe produit pour trouver la structure avec laquelle tout le monde se sent à l'aise.

Là aussi, vous pouvez y aller par itérations. N'ayez pas peur d'essayer, il est possible et même fort probable que votre façon de faire évoluera en même temps que votre prise de maturité sur le sujet.

Regardons ici comment le workflow présenté plus haut peut être traduit via un découpage en Epics, tâches ou Users Stories :

 Button [Analyse]	Alpha (0.0.1)	UI Component	DS-14
 Button [Design]	Alpha (0.0.1)	UI Component	DS-15
 Button [Code]	Alpha (0.0.1)	UI Component	DS-16
 Button [Doc]	Alpha (0.0.1)	UI Component	DS-17
 Button [Publish]	Alpha (0.0.1)	UI Component	DS-18

*Exemple d'US associées à l'épic Button. Epic liée à la construction du composant Button.*

## Créer des Epics

.....

Une Epic est un travail important qui peut être découpé en un certain nombre de petites stories. Construisez plusieurs Epics, une par composant. Pensez aussi à prévoir des Epics pour vos tâches liées à la **stratégie**, à la **communication** ou tout élément connexe nécessitant un découpage et un suivi dans le temps.

## Ajouter des tags

.....

Pensez à ajouter des tags, ils seront importants pour le suivi. Utilisez des tags pour vos composants, pour vos styles graphiques, vos éléments-clés (couleurs, typographies, icônes etc.) et tout autre élément pouvant être labellisé et regroupé dans une même

thématique.

### **Gérer vos versions**

---

En fonction de votre Roadmap, vous allez livrer au fur et à mesure des lots de différentes tailles.

Si vous travaillez avec un gestionnaire de projet qui permet de gérer des versions, créez donc autant de versions que nécessaire. Par exemple « Version Alpha (0.0.1) », « Version Beta (0.1.0) », « Non priorisé », « Next release », etc (pour plus de détails voir le Chapitre 5).

Si votre gestionnaire de projet ne vous le permet pas, traitez-les via les tags.

On pourrait penser que JIRA est le meilleur outil pour gérer votre Design System. Nous ne sommes pas si catégoriques : utilisez l'outil qui vous convient le mieux et qui semble le plus pertinent pour vos équipes et votre entreprise. Le plus important restera de bien le structurer.

### **Structurer vos fichiers**

Vous allez avoir beaucoup de fichiers, beaucoup de dossiers, que cela soit en développement, en UI et même dans vos espaces de stockage.

Réfléchir à la structure et à l'organisation de vos fichiers représente donc un gain de temps dans votre organisation quotidienne. En fonction de la densité de votre Design System, anticipez le fait que vous serez plusieurs à manipuler les documents de travail, voire même à travailler dessus en même temps. Pensez-donc à bien découper vos fichiers et leurs contenus pour gagner en productivité, en organisation et en rigueur.

## La nomenclature

La nomenclature, parlons-en !

Vous allez donc avoir énormément de fichiers et de différents types. Pour ne pas vous y perdre entre les exports « `img001.jpg` » et « `composant246.png` » sur votre documentation, vos calques « `layer24658` » et « `text245` » sur votre fichier de design ou encore un « correction du bug » dans vos commits, pensez à mettre des normes en place.

Nous vous conseillons de :

- Normer vos commits ;
- Normer le nommage de vos composants (aussi bien en développement qu'en UI) ;
- Normaliser votre code (classes CSS et noms de fonctions par exemple) ;
- Normer vos noms de fichiers ;
- Normer vos exports.

Et surtout normer votre façon de rédiger (exemple : éviter les anglicismes si vous faites une documentation en Français). Cela facilitera votre communication au quotidien et évitera les incompréhensions.

Bien que nous conseillons de ne pas utiliser des anglicismes pour une documentation en français, il est à noter que le design et le développement sont des domaines où l'anglais est intégré dans la pratique même de l'activité (`list`, `head`, `margin-bottom`, `font-family`,...), ou encore par convention pour nommer des composants ou des actions (`input`, `hover`,...). Un équilibre sera à trouver pour ne pas imposer des termes dans une langue qui ne serait pas naturelle à la pratique.



## L'importance de communiquer

Le mot d'ordre est : « É-van-gé-li-sa-tion » !

Avant même sa sortie, vous aurez besoin de faire connaître votre Design System aux équipes Produit qui l'utiliseront plus tard (et de manière plus globale, à la DSI).

Pour que ce produit soit accepté et utilisé une fois livré, vous devez communiquer sur son utilité, ses bénéfices et son usage, afin qu'ils y adhèrent le moment venu.

Avez-vous pensé à tous les cas de figure pour l'utilisation de votre Design System ? Ce que vous développez a-t-il vocation à être utilisé par toutes les branches de votre organisation ? Si oui, pensez bien à consulter les utilisateurs finaux afin de vous assurer que le Design System prend en compte leurs besoins, sous peine qu'ils ne l'utilisent pas !

### Définir une communication interne

Communiquez sur le fait qu'un outil structurant se trouve en construction et sera bientôt disponible pour réaliser vos produits numériques.

Nous vous conseillons de communiquer par étapes progressives et uniquement à partir du moment où la gouvernance du projet sera stable. Ceci pour les raisons suivantes :

- **Éviter des ingérences** qui pourraient ralentir le projet à un stade trop prématuré ;
- **Ne pas soulever à nouveau** les discussions **censées avoir été tranchées** en avant-projet ;

- Communiquer sur la thématique **dans un premier temps**, puis sur votre produit par la suite.

### Roadmap de communication

Construire un Design System représente la moitié du travail : faire en sorte qu'il soit utilisé constitue les 50% restants. Un Design System étant un produit tentaculaire, il devra bénéficier d'une communication planifiée et cadrée afin d'être sûr de toucher tous ses utilisateurs.

**Qui la réalise ? Le Design System Manager ou le P.O.** pourraient s'en charger.

Leur mission ? Définir les sujets sur lesquels communiquer ainsi que les moyens utilisés pour le faire. Ce temps dédié à la communication étant évidemment imputé sur le temps de production, identifiez au préalable les moments où vous devrez dégager du temps.

### Que communiquer ?

La première étape de communication sera dédiée à la présentation de ce qu'est un design system (introduisant l'utilité du produit, ses avantages...).

Ensuite, les communications seront fortement rythmées par l'avancée de votre Design System (ex : annonce de la sortie Bêta, annonce des releases suivantes, mise en avant d'ajouts attendus...).

Pour vous guider, voici quelques exemples de sujets de communication :

- **Présentation du Design System** : expliquer son intérêt, ses avantages, pourquoi c'est important, pourquoi en construire un, etc.
- **Outils** : décrire les outils nécessaires à l'utilisation du Design System

- **Releases** : montrer la progression du projet / les nouvelles mises en ligne disponibles
- **Présentation de l'équipe** : mettre des visages sur le produit est tout de suite plus sympathique (aide à l'adhésion) et vos interlocuteurs sauront vers qui se tourner en cas de besoin
- **Mise en avant de contenu** : tout le monde n'aura pas la curiosité d'aller voir votre Design System. Inclure dans votre mail des morceaux choisis sur une thématique précise peut forcer l'intérêt de certains. Cela peut aussi servir à lever des doutes que vous sauriez persistants chez certains utilisateurs réfractaires.

Par exemple, à destination des Product Managers : communiquer sur le fait que le Design System fournit tous les éléments pour concevoir des formulaires. Ou encore que les composants sont mobile-friendly ou parfaitement adaptés aux environnements web ou applicatifs...)

- **Principes UX** : si votre Design System en contient, c'est le genre de contenu qui peut intéresser les équipes projets pour leurs phases de conception.
- **Principes de développement** : annoncez les bonnes pratiques, la mise en avant des langages de code développés, les mises à jour de composants facilitées...
- **Un temps de questions / réponses** : des questions ressortiront sûrement des échanges que vous aurez en présentant le Design System à vos collaborateurs. En diffusant largement ces réponses, vous éliminez les idées reçues et touchez aussi les personnes qui s'interrogent sans oser vous contacter.
- **Mettre en avant des projets réalisés avec votre Design System** : faites appel au biais de conformité. Une personne peu sûre de la façon dont elle doit agir va se fier aux autres pour savoir quel comportement elle doit adopter. Ainsi, l'interview ou le témoignage d'une équipe ayant réalisé avec succès un projet en utilisant votre produit, rassurera les autres équipes et leur

engagera le pas.

### Comment communiquer ?

Le premier réflexe pourrait être de contacter les managers d'équipes afin de leur présenter le produit. Ce seront vos meilleurs alliés pour veiller à sa bonne mise en pratique par la suite (profitez-en pour identifier les personnes sur qui vous pourrez compter pour diffuser l'information).

Mais il vous faudra ensuite élargir au plus grand nombre. Voici quelques méthodes possibles :

- **Présentation PowerPoint** : simple et efficace, réalisez un document présentant ce qu'est un Design System, ses enjeux, ses avantages, sa méthodologie, ses impacts projet, les référents à contacter... Vous pouvez aussi ajouter des exemples d'entreprises qui l'ont déjà mis en place et par la suite de projets qui l'ont embarqué.

Commencez par de petits comités (10 personnes max) afin qu'ils s'expriment plus aisément. Finissez toujours par une session de questions à l'assemblée : l'échange permet d'identifier les doutes et les opportunités.

- **Mail groupé** : moyen de communication simple et efficace. Attention toutefois à l'effet d'accumulation : les mails adressés au plus grand nombre sont parfois ceux qui ne sont pas lus.
- **Newsletter** : à la différence du mail, il faut s'abonner : cela laisse donc supposer que le destinataire est déjà au courant de votre projet et souhaite le suivre. Ce service peut être diffusé via les autres moyens d'information, mais aussi être proposé sur le Design System.
- **Réseaux sociaux d'entreprise (Workplace, Yammer, Jamespot - Slack, Whatsapp...)** : si des moyens officiels de communication interne existent, ce sont de très bons porte-voix qui permettent le

relais de l'information. Un canal dédié pourrait même centraliser vos annonces.

- **Communication en mode Guérilla** : malgré tous vos efforts, il existera bien des personnes qui n'auront pas vu passer vos communications. N'hésitez pas à en parler durant vos pauses, au détour d'un couloir ou autre pour détecter ces personnes et leur proposer une présentation par la suite.

### Communiquer selon l'auditoire

Votre démarche sera de faire comprendre l'enjeu et les avantages d'un Design System. Mais il faut adapter votre discours à vos collaborateurs, selon les attentes et les craintes de chacun.

Voici un panel réaliste de ce que peuvent penser vos collaborateurs :

#### P.O.

.....

#### Ce qui peut les freiner :

- Une charge en plus dans des délais/budgets serrés ;
- Devoir refaire l'existant pour être conforme au Design System ;
- De possibles problèmes techniques (mélanges de code notamment) ;
- Besoin de certitude quant à l'aval du management pour l'ajout de cette nouvelle ressource / ce nouveau design (en cas de style graphique différent).

#### Ce qui peut les intéresser / pourquoi ils voudront l'intégrer :

- Un gain de temps de production design et développement non négligeable ;
- Un gain de temps dans la recette des composants à utiliser / des soucis fonctionnels et graphiques en moins ;

- La certitude de produire un projet ISO avec la marque entreprise ;
- Une mise à jour des projets en production plus simple et plus rapide ;
- Un cadrage des équipes projet via ce produit.

### **Designers**

.....

#### **Ce qui peut les freiner :**

- Manque de liberté / être relégué à de l'exécution ;
- Perte de créativité ;
- Usage du Design System trop contraignant (trop de règles) ;
- Devoir utiliser la librairie d'un logiciel de prototypage mal maîtrisé.

#### **Ce qui peut les intéresser / pourquoi ils voudront l'intégrer :**

- Gain de temps sur les éléments récurrents ;
- Gain de temps à consacrer à la résolution de problèmes et des changements de Direction Artistique ;
- Certitude de produire un projet ISO avec la marque entreprise ;
- Réduction du temps de recette ;
- L'outil de prototypage le plus commun à l'entreprise et / ou le plus adapté sera toujours pris en considération.

### **Développeurs front**

.....

#### **Ce qui peut les freiner**

- Comment implémenter le Design System dans les projets : par quoi commencer ;
- Difficile coexistence de plusieurs chartes et fastidieux ajout « composant après composant » dans le cadre d'une mise à jour

d'un site « petit à petit » ;

- Résistance au changement : devoir mettre en œuvre une méthodologie extérieure ;
- Que le Design System ne soit qu'une énième mise à jour de charte (répétition de tâche inutile) ;
- Travail inutile / désintérêt du management : peur que le projet soit abandonné en cours de route, augmentant la difficulté de gestion du site ;
- Difficulté d'usage avec trop de règles à prendre en considération pour les non-initiés ;
- Maîtrise de nomenclature / rigueur à tenir.

#### **Ce qui peut les intéresser / pourquoi ils voudront l'intégrer :**

- Moins d'erreurs et gain de temps de production (gain temps/ argent) ;
- Moins de retours sur LEUR production (erreurs de développement moins mises en avant) ;
- Facilité de maintenance ;
- Diminution de la dette technique par homogénéisation ;
- Un code plus propre / facilité de passation.

Rassurez vos interlocuteurs en leur rappelant qu'ils seront guidés sur la mise en place de ce produit et les pratiques à adopter. Que ce produit facilite les projets et soutient à terme la bonne évolution de toute l'entreprise.

## Travailler en équipe

La nature d'un Design System est **transversale, collaborative et pluridisciplinaire**. C'est une table ronde d'experts qui s'adapte aux évolutions des produits qu'elle sert.

Afin de bénéficier des compétences de chacun, voici quelques points à rappeler :

- S'appuyer sur des **outils de gestion de projet** assurera une bonne **transmission des informations** et une **trace de l'historique** (Confluence, Jira, Asana...);
- Si l'équipe grandit, plusieurs représentants d'un même métier devront collaborer. Il sera alors nécessaire de s'assurer qu'aucune tâche n'est réalisée en doublon et que les tâches parentes sont réalisées sur un même modèle ;
- Après toute conception, **une revue par un pair doit être O-BLI-GA-TOI-RE**. Gardez à l'esprit que toute erreur risque d'être embarrassante par la suite. C'est aussi le meilleur moyen d'homogénéiser les créations ;
- Certains sujets, comme la nomenclature de nommage par exemple, seront obligatoirement à réaliser en coédition : identifiez-les en amont ;
- **S'appuyer sur des rituels AGILE** (SCRUM par exemple) représente une bonne pratique **pour s'assurer que tous les membres possèdent le même niveau d'information** ainsi qu'une vision d'ensemble du projet ;
- Il est tentant de vouloir « mettre sa griffe » sur un tel projet, mais un design System, devant répondre aux besoins identifiés de ses utilisateurs finaux, ne doit pas supporter les choix non justifiés de ses contributeurs relevant de préférences personnelles.

Pour bénéficier des compétences de chacun, voici un rappel basique



de rapports professionnels (déjà connus mais facilement oubliables) à prendre en considération :

- **Favoriser la communication et ne pas sous-estimer les apports des collègues** : un produit réussi est un produit où chaque acteur de chaque expertise a pu donner le meilleur sur son périmètre. Une bonne écoute permettra au Design System d'atteindre tous ses objectifs, sur tous ses aspects.
- **Ne pas surestimer ce que nos collègues savent, pensent ou vont faire** : la validation par ses pairs se révèle essentielle pour s'assurer que chaque élément réalisé est aux normes attendues ISO. C'est primordial pour éviter les écueils et garder un cap identique, une qualité commune et une bonne efficience.
- **Casser la barrière des métiers** : entre collègues, il est toujours plus facile d'échanger avec ses pairs. On peut vite tomber dans un cloisonnement métier nous amenant à penser que les obligations des collaborateurs ralentissent, limitent, ou même dévient le projet.
- Pourtant, un produit réussi est un produit dont chaque métier aura su préserver les obligations de son périmètre. Un produit équilibré sera celui où chaque acteur considérera positivement les besoins de ses collaborateurs.
- **Laisser l'équipe s'exprimer** : le lead n'est pas une dictature. Tous les avis sont à considérer puis à évaluer afin de tirer le meilleur de chacun.

## La conception par composants

### Approche modulaire, approche intégrée... qu'est-ce que c'est ?

Dans une approche modulaire, les composants partagent certains paramètres que l'on peut aussi appeler variables - qu'il s'agisse du

*Déjà entendu  
au détour d'un couloir...*

**“ Les développeurs vont  
encore dire que c'est trop  
compliqué.**

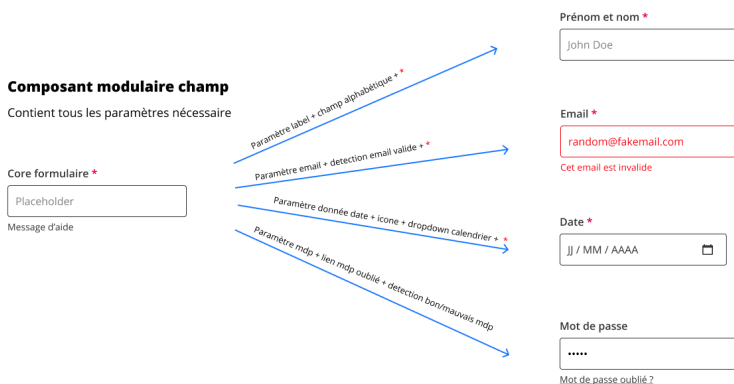
**C'est des trucs de créa  
ça, c'est juste un coup de  
peinture.**

**Il est pénible ce P.O. à  
demander ou j'en suis sur  
mon travail.**”

style : une couleur, une police de caractère, un contour mais aussi des paramètres fonctionnels (comportement, états). Le changement de l'une de ces variables s'applique à tous les composants. Cette approche permet donc de faire évoluer tous les composants concomitamment en modifiant ces paramètres.

À l'inverse, dans une approche intégrée, chaque composant est créé de manière unitaire et sans être relié aux autres par des paramètres dynamiques. Cela signifie ainsi qu'un composant existe pour un rôle et un usage unique. Quand bien même les composants respectent des guidelines ou une charte, la modification d'un composant spécifique n'a pas d'incidence sur le reste de l'écosystème.

Si nous prenons l'exemple d'un champ de formulaire : dans une approche modulaire, le composant sera paramétré selon le besoin : présence ou non de label, d'icône, est-ce que le champ est un champ de texte classique ou spécifique pour un mot de passe, une adresse email, une date...



Dans une approche intégrée, ce composant sera créé spécifiquement pour ce cas de figure. Pour un besoin différent, il faudrait alors concevoir un autre composant, si l'un de ces paramètres venait à être différent. Cela n'empêche pour autant pas ces composants d'avoir un CSS commun et qu'un changement de style s'applique à tous les composants concernés.

Composant intégré champ alphabétique

Prénom et nom \*

Composant intégré email

Email \*

Cet email est invalide

Composant intégré datepicker

Date \*

Composant intégré mot de passe

Mot de passe

Mot de passe oublié ?

## Quelle approche choisir ?

Certaines des caractéristiques de ces 2 approches peuvent être des qualités essentielles ou des défauts rédhibitoires, selon le contexte du produit.

Ainsi, **une approche intégrée** permet de mettre en place un Design System opérationnel bien plus rapidement et sa flexibilité permet des compromis parfois nécessaires. **\*\*Une approche modulaire\*\***, quant à elle, est recommandée si l'automatisation se révèle un besoin essentiel pour vous et que vos interfaces ne peuvent en aucun cas, être hétérogènes.

## Mais alors, quelle approche choisir, pour quelle situation, quel contexte, quels besoins ?

L'approche intégrée se prête bien à des projets de petite ou moyenne ampleur de par sa rapidité de mise en place. Mais elle convient aussi à des situations qui regorgent de cas particuliers et où la communication et la gouvernance entre les différents services, produits ou équipes ne permet pas de prise de décision collégiale. De même, cette approche s'avère pertinente si la scalabilité n'est pas à l'ordre du jour.

Quant à l'approche modulaire, bien que plus complexe à mettre en place, elle vous offrira bien plus de satisfaction sur le long terme. Elle se révèle particulièrement adaptée si votre parc applicatif est très important, si l'automatisation est votre maître mot. Ou si vous souhaitez confier la gestion du Design System à une équipe centralisée plutôt que de la répartir au sein de différentes équipes et

d'en multiplier les instances (et fatalement, les divergences), toujours dans un même souci de maintenance et de scalabilité.

Si votre Design System doit vite faire ses preuves, l'approche intégrée sera la plus adaptée du fait de sa plus grande rapidité de déploiement et de sa souplesse relative.

En résumé :

### **L'approche modulaire**

---

#### **Avantages :**

- Garantit l'homogénéité des composants du fait de la centralisation de leurs paramètres ;
- Automatise leur évolution par héritage ;
- Facilite la maintenance.

#### **Inconvénients :**

- Nécessite énormément plus de rigueur à la conception ;
- Demande beaucoup plus de communication et de synchronisation entre les équipes ;
- Est nettement moins souple à la conception : les composants obéissent à des règles ;
- Chaque modification implique d'en mesurer les impacts potentiels ;
- Plus coûteuse à mettre en place.

### **L'approche intégrée**

---

#### **Avantages :**

- Est plus rapide à implémenter ;
- Composants est bien plus souple ;
- Moins onéreuse à court terme (temps passé > résultat).

**Inconvénients :**

- Les évolutions et la maintenance du Design System prennent plus de temps ;
- Beaucoup plus de contrôle est nécessaire pour le maintien de la cohérence et de l'homogénéité des interfaces ;
- Si la souplesse est une force, elle peut conduire à de multiples exceptions et dérives si elle est utilisée à mauvais escient, menant à une expérience et des interfaces hétérogènes.

**Commencer à concevoir**

Parce qu'un Design System est une solution sur-mesure et adaptée à des besoins précis, il est impossible de proposer une solution clé en main. Pour définir les composants nécessaires, il faut étudier, auditer, et faire l'inventaire de son produit pour estimer les besoins auxquels le Design System répondra. Cependant, il existe des éléments-clés par lesquels commencer sa conception.

**Les Core Values**  
.....

Avant même de construire vos composants, vous aurez besoin d'éléments essentiels, la matière de base pour commencer à concevoir, les fondations sur lesquelles reposent tout votre Design System.

Nous appelons ces éléments les Core Values. Il s'agit :

- Des styles de texte ;
- Des couleurs ;

- Des éléments liés à la marque ;
- Des grilles de composition.

### Les Core Components

---

Une fois les Core Values définies, vous pourrez débiter la construction des composants indispensables à toute interface, qui vous serviront par la suite pour la construction de composants plus complexes et propres à votre produit. Nous appelons ces composants indispensables, les Core Components. Ce sont :

- Les liens et boutons ;
- Les champs ;
- Les checkboxes, radio buttons, toggles, sélecteurs ;
- Les inputs & form types ;
- Les spacings ;
- Les icônes.

### Construire et décliner les Core Values & Core Components

**Hiérarchiser les styles de texte** : titres, textes courants, catégories...

Le découpage proposé par Material Design est un excellent modèle à suivre : headline, body, subtitle, button, caption, overline.

Attention à ne pas nommer ses titres de la même manière que des balises html : les styles que nous définissons ici sont indépendants de la structure technique de la page et ne doivent pas être liés à cette dernière.

**Classer les couleurs par catégories** : les couleurs principales liées à la marque et / ou au produit, les couleurs contextuelles (feedback positif ou négatif, erreurs, alertes...), les couleurs de fond, les couleurs de texte...

Il est important de les classer pour préciser leur cas d'usage et éviter les confusions ou erreurs.

**Mettre en place un système de grilles** : définir une grille pour différents formats de devices ou plusieurs grilles si des cas particuliers récurrents se présentent.

**Décliner les liens et boutons** : set de boutons primaires, secondaires et éventuellement tertiaires, variations de tailles liées au contexte, états (survol, actif, désactivé, focus).

**Décliner les champs et comportements associés** : champs de saisie de texte, champs de recherche et résultats, pré-saisie, formulaires.

**Identifier les types d'inputs et form types nécessaires** : menus horizontaux et verticaux, sélecteur de date, champs avec sélection, combobox...

**Définir un standard d'espacements** : il peut soit s'agir de créer un set défini d'espacements (par exemple 4, 8, 16, 24, 40, 64, 80px) soit de s'accorder sur une unité de référence (par exemple, tous les espaces sont un multiple de 8px).

Note : la grille de 8px est devenue très populaire puisqu'une très grande majorité d'écrans tous types confondus ont une largeur divisible par 8.

**Mettre à disposition un set d'icônes** : qu'il s'agisse d'icônes génériques ou spécifiques au projet. Il peut être judicieux de créer une librairie séparée pour les icônes et de classer celles-ci par typologie.

**Mettre en place un système de grilles** : définir une grille pour différents formats de devices, ou plusieurs grilles si des cas particuliers récurrents se présentent.



Bien que nous ayons mentionné le principe d'atomic design, nous ne pouvons pas commencer à découper nos composants en suivant les catégories atome / molécule / organisme, ces catégories n'exprimant pas explicitement les éléments d'interface. Contentons-nous de considérer l'Atomic Design comme un concept pour comprendre le principe de composants imbriqués à différents niveaux.

Il est important de **décliner les éléments selon ses besoins et en se restreignant à l'essentiel**. L'idée est de parer à toutes les cas d'usage avérés et ne pas se poser de questions et non pas de proposer des choix multiples préventifs qui complexifient le processus de décision.

À cette étape, la **nomenclature** des composants s'avère essentielle : elle doit être claire et explicite. Peu importe que le composant soit un atome, une molécule ou un organisme, son intitulé doit être descriptif et permettre de le nommer et de le retrouver facilement.

Nous favorisons par exemple un naming de type « bouton\_primaire\_XL\_icône-gauche plutôt que atom\_bouton\_3 ». La nomenclature des composants fait donc bien plus que vous aider à organiser un fichier de design. En adoptant un langage commun pour parler d'un composant, elle fluidifie les échanges avec le développement et évite, encore une fois, les confusions et malentendus.

Prenez du temps avec votre équipe de développeurs pour connaître leurs contraintes techniques et pour faire l'inventaire des paramètres possibles à implémenter pour les composants que vous créez. Pour un bouton, par exemple : contour, fond, texte, ombre, icône, espacements...

Plus le fonctionnement de vos composants sera proche entre le design et le développement (en plus de la nomenclature !) plus les échanges et leur mise en place seront fluides.



## Chapitre V

# Le déployer et le faire vivre

Votre Design System est en ligne, il commence à être utilisé : bien joué !

Cependant, ce n'est pas fini, ce n'est que le début.

Comme nous l'avons affirmé plus tôt, un Design System est par définition scalable, il va donc falloir mesurer régulièrement sa capacité à s'adapter, communiquer sur son évolution, gérer ses versions.

***“ Le travail d’un Design System ne s’arrête pas à son déploiement, il y a tout un travail de maintenance et d’évolutions à prendre en compte. ”***

## Le déployer dans votre écosystème

Pour déployer un Design System, il ne suffit pas de cliquer sur «publier» ou de mettre du code en ligne.

Un déploiement se prépare, se planifie. Il faut ainsi se charger des nombreux éléments à publier, préparer et diffuser la communication autour de votre Design System. Il faut également gérer les différents retours.

### Que déployer ?

Dans la majorité des cas, 4 grandes topologies de fichiers sont à déployer :

#### 1. Du code.

Cela constitue votre librairie de composants, votre site de démonstration, votre storybook, etc.

#### 2. Du design.

Qui se rapporte à vos fichiers pour construire votre librairie UI, vos fichiers typographiques, vos icônes, vos design Tokens etc.

#### 3. De la documentation.

Il s'agit de votre documentation d'usage et documentation technique.

#### 4. Des supports de communication.

Mettre à jour votre patch note, vos présentations, préparer vos mails.

Ils permettent de mettre à jour votre patch note, vos présentations, préparer vos mails.

Tous ces fichiers ne se déploient pas de la même manière, ni avec les mêmes outils et pas nécessairement en même temps.

### Pre-versions 1.0.0

Avant de déployer votre première version - la version officielle qui sera communiquée à toute votre entité - vous vous exposez à déployer plusieurs versions intermédiaires moins complètes.

Dans la majorité des cas, nous parlons de version **Alpha** ou version **Beta**. Celle-ci s'emploie pour effectuer les premiers tests en conditions réelles et roder votre processus. Votre version alpha peut ne contenir que des composants UI sans code ni documentation, un **POC en code** ou même juste des cas d'usages. Cela dépend de votre besoin, du temps alloué et de votre maturité. Cela peut être tout aussi bien un travail en interne, qu'une diffusion à des **bêta-testeurs** ou bien à vos **early-adopters**.

Cela permet d'obtenir des retours et de tester très rapidement votre méthodologie et vos composants.

### Version 1.0.0

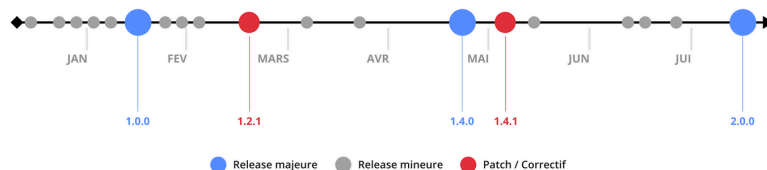
Votre **version 1.0.0** lance le premier déploiement officiel ! Cela représente un moment très important car votre Design System s'ouvre au grand public.

Vous pouvez parvenir à cette version de manière incrémentale c'est-à-dire en agrémentant votre version alpha petit à petit (V.0.0.1, V.0.1.2, V.0.12.45...) ou bien passer directement de la version alpha à la V.1.0.0 (bien que plus risqué).

Déploiement assez dense, cette version comporte tous les éléments nécessaires à l'usage de votre Design System : design, code, documentation, supports de communication, formations, démos...

## La gestion des releases

La diffusion de votre version 1.0.0 ne représente qu'une étape dans



le cycle de vie de votre Design System. Vous allez le faire vivre en livrant régulièrement évolutions, ajouts et correctifs.

Prenons le temps de nous arrêter sur un élément essentiel : **le versioning**. Pour ce faire, nous utilisons une convention nommée **SemVer (Semantic Versioning)**. Cette gestion sémantique des versions permet de les numéroter de façon logique et significative. Une version se rapporte à un produit, une application, une

**2 . 1 . 3**

Major                      Minor                      Patch

bibliothèque, un OS, tout ce qui comporte une progression en informatique. Cela aide à définir l'avancée du produit.

Ce système propose un format de version en MAJOR.MINOR.PATCH :

- **MAJOR** : correspond à des modifications non rétrocompatibles. Par exemple vous changez votre façon de concevoir un composant.
- **MINOR** : correspond à des modifications rétrocompatibles. Par exemple l'ajout d'une nouvelle variante sur un composant.
- **PATCH** : correspond à des corrections rétrocompatibles. Il s'agit, par exemple de la correction d'un arrondi en CSS sur un composant suite à une erreur.

Cette nomenclature vous permet de savoir précisément à quel niveau d'évolution vous vous situez sur votre Design System.

En effet, utilisée sur vos fichiers de design et sur votre librairie, vous pouvez voir rapidement s'il existe un décalage entre le design et le développement.

Par exemple, un bouton ayant pour version 1.2.0 en design et 1.1.2 en développement vous informe sur le décalage entre la partie design (en avance) et la partie développement.

Il est important pour un Designer d'apprendre cette nomenclature et de s'en servir au quotidien, même si elle vient du monde du développement.

Vous trouverez 2 grandes façons de gérer votre versioning sur un Design System :

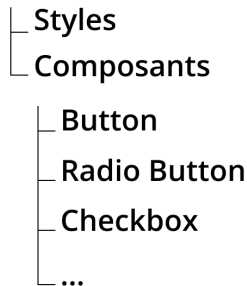
### **Au niveau bibliothèque**

La gestion de version se fait au niveau Librairie. Ainsi, celle-ci évolue à chaque évolution de composant.

Ce système présente de l'intérêt si vous utilisez une librairie HTML / CSS ou si votre Design System ne connaîtra que peu d'évolutions sur



## Design System 1.2.0



le long terme (1 à 2 composants par an).

### Les points positifs :

- Nul besoin de se questionner sur la différence de version entre chaque composant ;
- Cela simplifie la gestion à chaque release car toute la librairie augmente de version.

### Les points négatifs :

- Manque de précision sur la version réelle du composant ;
- Rend abstraite la distinction entre différentes versions d'un même composant.

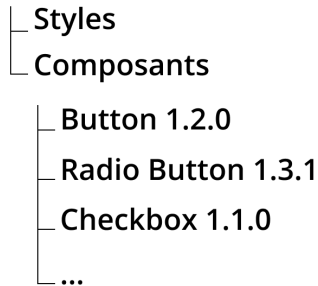
### Au niveau composant

.....

La gestion se fait au niveau composant. Chaque composant évolue indépendamment des autres.

Ce système se révèle pertinent lorsque vous employez des frameworks comme Angular, React et View.

## Design System



### Les points positifs :

- Chaque composant possède sa propre version ;
- Permet de voir le delta de versions entre chaque composant ;
- Rend possible la coexistence de différentes versions des composants ;
- Les composants peuvent être livrés indépendamment les uns des autres ;
- Le numéro de version de la librairie se trouve identique au numéro de release.

### Les points négatifs :

- Difficile à utiliser si votre bibliothèque existe uniquement en HTML / CSS ;
- Le numéro de version de vos releases différera du numéro de version de vos composants.

Vous l'aurez donc compris chaque release sera accompagnée d'un

patch notes résumant son contenu et d'un numéro de version.

Par ailleurs la nature de vos releases n'est pas forcément technique. Ainsi, vous pouvez livrer une V.1.1.2 contenant un nouveau composant en développement et une V.1.2.0 ne comportant que du Design. Vous pouvez aussi choisir d'attendre l'entièreté du composant (Design + Code) avant d'incrémenter la version. Libre à vous de voir ce qui vous correspond le mieux.

Concernant vos releases vous allez avoir :

- **Des releases majeures.**

Elles concernent des modifications importantes de votre Design System, comme par exemple l'ajout d'un nouveau composant ou d'une nouvelle librairie, un passage d'une version 1.0.0 à 2.0.0.

- **Des releases mineures.**

Cela correspond à de petites évolutions comme l'ajout d'un élément dans la documentation, l'ajout d'un cas mineur en design, la modification de la façon de coder un composant, etc.

- **Des patches/correctifs.**

Il s'agit de releases mineures dédiées à des correctifs (graphique/ technique/doc).



Attardons-nous maintenant sur les 2 principales cadences de releases :

### Releases par incrémentation (Releases By Increment).

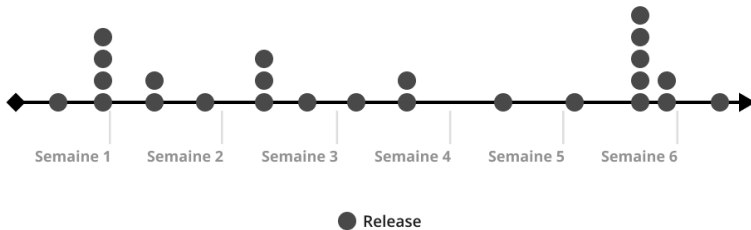
La cadence se définit par le rythme de vos sprints, par exemple une release par sprint ou tous les X sprints.

Cette cadence apporte une structure et un rythme dans votre gestion des releases. Vos utilisateurs savent qu'une livraison se fait toutes les X semaines. Par contre, elle vous demande une bonne préparation : vous devez ainsi définir le contenu de plusieurs sprints en avance.

Elle fonctionne très bien avec un versioning tant au niveau composant qu'au niveau de la bibliothèque.

Cependant, avec ce système, vous êtes moins réactif face aux aléas, telles que des erreurs mises en production. Chaque release se trouvant cadencée, un correctif doit attendre la prochaine release pour être mis en production. De plus, en fonction du correctif à apporter, cela peut décaler la suite de vos releases.

À long terme, cette cadence peut se révéler difficile à suivre si vous



ne possédez pas une équipe dédiée au produit Design System.

### Releases continues (Continuous Releases).

Cette cadence vous permet de livrer dès que c'est prêt, sans devoir attendre un moment précis ! Les livraisons peuvent se faire en continu, plusieurs fois par jour si nécessaire.

Ce système peut se révéler avantageux si vous devez livrer beaucoup d'éléments ou si vous souhaitez un découpage fin de chaque release

(une release par item). Il fonctionne très bien avec un versioning niveau librairie et se complexifie avec un versioning niveau composant.

Attention cependant, en livrant au fil de l'eau, vous risquez de casser plus régulièrement et plus facilement les éléments de votre Design System. Attention aussi à ne pas tomber dans une course où vous livrez tout et n'importe quoi.

En contrepartie, vous disposez d'une réactivité plus forte car vous ne devez pas attendre la release prévue au sprint suivant pour effectuer une correction.

Il n'existe pas de cadence meilleure qu'une autre. Elle se choisit en fonction de votre contexte, votre équipe et vos habitudes.

Nous vous conseillons de commencer par des releases par incrémentation. Cela vous permet d'apporter un cadre et un rythme à la construction de votre Design System.

## Mesurer l'impact

Un Design System étant un outil de production, son évaluation doit se faire selon des objectifs mesurables et quantifiables définis en amont.

Dans la majorité des cas, celle-ci se fait sous 2 angles :

- D'un côté les KPIs en lien avec l'opérationnel, comme le temps gagné sur la production.
- Et d'un autre, les KPIs en lien avec vos utilisateurs (interne & externe) comme par exemple, leur satisfaction à utiliser votre Design System.

**N.B.** Il est important de définir vos KPIs dès le début de votre projet

afin de pouvoir initialement mesurer l'impact de ce dernier. Pour ce faire, vous pouvez vous appuyer sur notre fiche Atelier : Définition des critères de succès en page X. Cela vous permet également de pouvoir mesurer la progression.

De nombreux outils (dont certains déjà connus par vous et vos équipes) permettent de suivre la performance de votre Design System :

- La mesure de la vélocité de vos équipes, une fois le Design System implémenté ;
- La mesure du Time-to-Market ;
- Les OKR de google ;
- Le ratio de jours nécessaires pour construire un même élément avant et après la mise en place d'un Design System ;
- Le mesure du taux de couverture du Design System. Il s'agit de quantifier le pourcentage de votre parc applicatif couvert avec le Design System à des dates T (à 3 mois, 6 mois, etc.).  
Les échelles de mesure de l'expérience utilisateur comme le SUS (System Usability Scale), le DEEP (Design-oriented Evaluation of Perceived Usability) ou l'AttrakDiff par exemple ;
- L'évolution du Design System, comme la diminution du Backlog au fur et à mesure de l'avancée du produit ;
- Des questionnaires de satisfaction utilisateurs.

## Quelques cas d'objectifs et de KPI correspondants

### Objectif 1

Mesurer le gain de production avec la mise en place du Design System.

### Résultats clés dans un délai de 6 mois :

- Une division par 2 du temps passé en moyenne sur le front ;
- Moins d'US front-end avec une complexité supérieure à 5 ;
- Une diminution des bugs liés au front de 60% ;
- L'équipe UI est capable de prendre 3 US en plus par sprint.

### Objectif 2

Mesurer le taux d'adoption du produit Design System par les équipes.

### Résultats clés dans un délai de 12 mois :

- Toutes les nouvelles fonctionnalités clés sont basées sur le Design System ;
- Toutes les équipes ont installé les dépendances du Design System ;
- 100% des produits ont vus leurs couleurs et typographies passées sur le Design System ;
- 70% des équipes ont jugé « facile » et 30% « très facile » la transition vers le Design System.

Ne mesurez pas pour mesurer, ces indicateurs doivent vous aider à facilement identifier ce qui fonctionne ou ne fonctionne pas avec votre Design System. Les chiffres doivent vous aider à avancer sur le sujet, à motiver vos équipes et parfois même à prioriser vos évolutions. Ces analyses font ainsi remonter les retours des utilisateurs et permettent d'ajuster nos outils pour qu'ils correspondent au mieux à leurs besoins.

N'oubliez pas : les chiffres seuls ne veulent rien dire. Associez-les toujours à un objectif. Ils vous aideront à partager le succès de votre Design System de manière chiffrée !





## Pour conclure

Comme vous avez pu le constater, construire un Design System représente un sujet bien plus complexe et complet qu'une simple bibliothèque de composants. Cela nécessite **du temps et des ressources**.

Nous vous conseillons d'avancer **à votre rythme**, avec vos moyens et vos outils sans chercher à obtenir le système parfait dès la première version. Découpez-le en étapes et allez-y progressivement. L'objectif n'est pas de concevoir le produit le plus rapidement possible, mais de créer un produit solide et pérenne.

Gardez bien en tête que votre Design System est un **outil vivant** qui doit **s'adapter à votre entreprise et à sa culture** et non l'inverse. Il met à l'honneur **vos valeurs**.

Ce **produit sur-mesure** évolue en fonction de vos besoins. Il doit remporter l'adhésion des parties prenantes. Il vous faut construire un Design System pour qu'il soit utilisé dans la durée.

Si vous ne deviez retenir que peu de choses de ce guide, cela serait :

- Gérez votre Design System comme un produit.
- Communiquez et vendez votre Design System.
- Travaillez en équipe, ne vous silotez pas.
- Mesurez et testez, n'ayez pas peur d'essayer.



# Glossaire

## **Definition of Done**

Correspond à l'ensemble de critères définis par la Feature Team qui permet de déterminer si l'User Story est considérée comme traitée.

## **Definition of Ready**

Correspond à l'ensemble de critères définis par la Feature Team qui permet de déterminer si tous pré-requis sont présents pour réaliser la User Story est prête à être prise dans le sprint.

## **DesignOps**

Membre de l'équipe qui a pour charge de mettre en place des processus de travail efficaces pour les Designers.

## **Dette technique**

Elle peut être défini comme le décalage entre l'état de l'art et l'existant. Elle est présente à cause de malfaçons souvent dues au non-respect de la conception et des règles de codage etc. Une dette technique peut-être intentionnelle ou non intentionnelle et impact la qualité du produit final.

## **DevOps**

Membre de l'équipe qui a pour charge d'améliorer la collaboration entre les développeurs et les équipes de déploiement.

## **Early adopters**

Désigne un client précoce d'un produit ou d'une technologie.

## **Micro-interactions**

Terme défini par Dan Saffer, les micro-interactions sont de petits instants pendant lesquels l'utilisateur et l'interface interagissent.

### **Facteur Autobus**

Le facteur d'autobus (de l'anglais bus factor) est une mesure du risque dû à l'absence de partage d'informations et de compétences entre les membres d'une équipe.

Le terme vient de la phrase « Combien de personnes clés dans votre équipe peuvent se faire renverser par un autobus avant que votre projet échoue ? ».

### **Framework**

Un framework correspond à un "cadre de travail". Il permet de simplifier le travail des développeurs en leur offrant une architecture "prête à l'emploi" qui évite de repartir de zéro à chaque nouveau projet.

### **P.O.C**

Renvoie au terme Proof of Concept, ou preuve de concept en Français. C'est une étape de validation qui permet de démontrer la viabilité du concept.

### **Scabilité**

Tiré du verbe anglais « to scale », qui signifie « changer d'échelle ou de dimension ». La scalabilité est la faculté d'un produit à s'adapter aux fluctuations de la demande et du marché, tout en conservant ses fonctionnalités de base. Notamment en cas de forte demande ou d'accroissement rapide.

### **Stakeholders**

Désigne d'une manière générale l'ensemble de tous les acteurs et de toutes les parties prenantes qui présentent un intérêt dans une entreprise.

### **Time to Market**

Le Time To Market correspond à la durée nécessaire pour la mise sur le marché d'un produit. Plus ce délai est court, plus vous serez en capacité de devancer vos concurrents et d'être rentable.

### **Tone of Voice**

Le Tone of Voice désigne la façon avec laquelle l'entreprise communique ses valeurs et sa vision à son audience cible et comment cela influence la façon dont celle-ci perçoit le message. C'est en quelque sorte la personnalité de vos produits.

### **User Story**

Une user story est une explication non formelle, générale d'une fonctionnalité logicielle écrite du point de vue de l'utilisateur final. Son but est d'expliquer comment une fonctionnalité logicielle apportera de la valeur au client.

### **Utilisabilité**

Correspond à l'aptitude d'un objet ou service à être utilisé facilement par une personne, conformément à ce pour quoi il a été conçu.



# World of Digits, qui sommes-nous ?

Chez World of Digits, nous plaçons **l'expérience utilisateur, au cœur de toute stratégie de transformation digitale.**

Cabinet de conseil digital international, nous vous accompagnons dans **le développement de vos produits et services digitaux.** Nos équipes sont expertes dans l'optimisation de l'expérience utilisateur, de la transformation Agile, Product Management, Customer Analytics et Project management.

Notre cœur de métier est de **porter la voix du client** en mettant au centre de toute démarche les besoins des utilisateurs.

Notre tribu de spécialistes combine **passion, expertise et stratégie** pour emmener **vos projets vers le succès !**

## Nos valeurs ?

- Mettre l'innovation au service de l'utilisateur ;
- Vous offrir des solutions pour relever vos défis ;
- Vous garantir la meilleure expérience possible ;
- Vous faire grandir ;

Vous aussi, faites appel à nous ! Nous sommes prêts pour collaborer avec vous :

- Définir, concevoir et **soutenir vos projets digitaux** ;
- Choisir et décider ensemble les ressources, méthodologies et technologies pour **réaliser vos objectifs.**

**Contactez-nous ici :** <https://worldofdigits.com/fr/contact/>

# Les contributeurs

Nous tenons à remercier chaleureusement toutes les personnes ayant contribué à la réalisation de ce document :

**Grégory Buron**  
UX Architect - UX Lead

**Quentin Dubas**  
UI Designer

**Bruno Tomasso**  
UX/UI Designer

**Paul Muller**  
UI Designer

Mais aussi tous les membres de World Of Digits qui nous ont aidé dans la réalisation de cet ouvrage grâce à leurs commentaires, révisions et contributions :

Benjamin Faurie, Natalia Devos, Jessica Rodrigues, Carmen Martin, David Jaumes, Loic Djili, Johann Leroux





# Inventaire de l'existant

## DURÉE

1h30 – 2h

## MATÉRIEL

Ciseaux Feuilles A3  
Colle Tableau blanc  
Post-it

## PARTICIPANTS

Multi-profils (développeurs,  
designers, PO, PM)  
1 animateur

## Objectif

Lister les différents éléments présents sur vos interfaces pour avoir une vision d'ensemble des composants que vous utilisez le plus.

## Préparation de l'atelier

Après avoir recensé et exporté les écrans principaux de vos différentes interfaces, toutes plateformes et devices, imprimez-les en plusieurs exemplaires (1 lot par équipe).

## Variantes

Pour diverses raisons (peu de créneaux disponibles, parc applicatif trop grand, etc.) il est possible que vous ne puissiez pas allouer 1h30 à 2h sur ce type d'atelier.

Dans ce cas, vous pouvez :

- Faire plusieurs ateliers de ce type, mais plus courts : 1h maximum.  
En découpant application par application.
- De même, si votre interface se révèle complexe, vous pouvez limiter le nombre de pages traitées lors de cet atelier. Vous pouvez alors réaliser plusieurs ateliers de ce type, à la suite

## Déroulé de l'atelier

### ● Introduction (10min)

Énoncez le but de l'atelier, constituez des groupes et donnez à chacun un lot d'impressions.

Ensuite, expliquez à chaque groupe qu'ils doivent chercher, regrouper et nommer un type de composant.

### ● Travail en groupe (45min)

Chaque groupe recherche un type d'élément/de composant : le groupe A recherche par exemple les différents boutons, le groupe B les différents liens, le C les menus, etc. Un groupe peut avoir pour tâche de rechercher plusieurs éléments.

Par la suite, le groupe doit regrouper les éléments d'une même famille.

Nous vous conseillons de vous focaliser sur le rôle de l'élément et non sur sa représentation graphique.

### ● Mise en commun (20min)

Chaque groupe présente son résultat aux autres en explicitant sa réflexion et en mettant en avant les divergences et les ressemblances des éléments de leurs groupes. C'est un moment d'échanges.

### ● Lister et voter (15min)

L'animateur note sur un tableau blanc les principaux groupes évoqués. Par la suite, chaque participant vote pour définir quels groupes doivent être réalisés en priorité.

# Prioriser ses composants

## DURÉE

1h30

## MATÉRIEL

Stylo      Feuilles blanches  
Scotch     Tableau blanc  
Post-it     Des matrices imprimés

## PARTICIPANTS

Multi-profils (développeurs, designers, PO, PM) de l'équipe Design System et celles participants à la conception  
1 animateur / facilitateur

## Objectif

Classer les composants à construire par priorité.

C'est-à-dire lister les composants en termes de fréquences et de facilité de construction pour définir un backlog et organiser les sprints de développement.

## Préparation de l'atelier

Imprimez des planches avec les composants à construire. Préparez des matrices sur de grandes feuilles (ou encore sur un tableau), dont l'axe vertical représentera la fréquence des composants dans vos interfaces, et l'axe horizontal représentera leur complexité de développement.

Préparez en amont des post-it contenant les noms des composants à construire. Cela fera gagner du temps aux équipes lors de l'atelier.

Veillez à avoir une salle assez grande pour disposer de plusieurs groupes de travail.

## Déroulé de l'atelier

### ● Introduction (10min)

Expliquez le but de l'atelier, faites des groupes multidisciplinaires et donnez à chacun une matrice et un lot contenant la liste des composants à construire.

Afin d'éviter les biais, pensez à définir la notion de simple/complexe, fréquent/rare avant de vous lancer dans l'atelier.

### ● Travail en groupe (30min)

Chaque groupe va devoir estimer la fréquence et la complexité de chaque composant à construire et les placer sur la matrice.

### ● Mise en commun (20min)

Accrochez chaque matrice sur un mur. Positionnez ensuite chaque composant sur une matrice finale vierge

S'il existe une grande disparité de fréquence / complexité sur un composant, laissez les équipes s'exprimer et débattre.

À la fin, vous obtiendrez une priorisation claire et précise de l'ordre dans lequel construire les composants.

Les plus simples et fréquents pouvant être réalisés «tout de suite» alors que les moins fréquents et les plus complexes nécessitent plus de travail et devront donc être réalisés plus tard.

# Rôles & Process

## DURÉE

1h – 1h30  
selon le nombre  
de participants

## MATÉRIEL

Stylo  
Feuilles blanches  
Cartes rôles

## PARTICIPANTS

Échantillons représentatifs de chaque rôle et profil. Minimum 8 personnes. L'idéal est de pouvoir avoir entre 2 & 5 groupes de 4 à 6 personnes  
1 animateur / facilitateur

## Objectif

Identifier les rôles de chacun ainsi que les tâches associées à ces rôles.

## Préparation de l'atelier

Préparez les cartes : vous pouvez les écrire à la main ou alors imprimer les modèles disponibles en page (lien vers les modèles). Les questions doivent être de la forme «En tant que..., comment puis-je... ?».

Il y a 4 rôles principaux, indépendants des expertises métiers :

- Les « Contributeurs » sont les utilisateurs qui participent à la construction ;
- Les « Utilisateurs » sont ceux qui utilisent votre Design System sans y contribuer ;
- Les « Leaders » décident et valident ce qui doit être ou non dans le Design System ;
- Et les « Sponsors » sont le plus généralement les Directeurs qui valident les budgets et sponsorisent le produit.

## Déroulé de l'atelier

### ● Introduction (5min)

Expliquez le but de l'atelier en explicitant les termes Contributeurs / Utilisateurs / Leaders / Sponsors.

### ● Travail en groupe (40min)

1. Faites des groupes de 4 à 6 personnes en prenant soin de mélanger les personnes ayant des profils et des rôles différents et distribuez-leur les cartes. Chaque groupe doit disposer de cartes reprenant les 4 rôles.
2. Chaque membre de chaque groupe va remplir individuellement les cartes qui correspondent à son rôle (10min). Puis, vient une mise en commun au sein de chaque groupe (30mn).

### ● Mise en commun (15min)

Chaque groupe présente ensuite ses résultats aux autres groupes. Il faudra décider quelles sont les propositions qui peuvent être mises en place tout suite, dans un temps 1, 2, etc.

Si vous disposez de plus de temps, préparez un tableau pour lister ce qui doit être pris en compte maintenant puis à la sortie de la V.0, à la V.1, etc.

Certaines propositions se révéleront alors inutiles au début mais deviendront nécessaires, par la suite et vice-versa.

# Définir les critères de succès

## DURÉE

1h30 – 2h

## MATÉRIEL

Stylo

Post-it

Tableau blanc

Feuilles blanches

Gomettes pour voter

## PARTICIPANTS

Leads designers

Leads Développeurs

PO / PM

Stackholders

1 Animateur / Facilitateur.

## Objectif

Définir quels sont les objectifs principaux de votre Design System et quels métriques/indicateurs permettent de mesurer si c'est un succès ou non ?

## Préparation de l'atelier

Lors de la mise en place de votre Design System, vous avez réalisé des interviews avec les différents membres des équipes pour soulever et lister les problèmes à résoudre. C'est le moment de ressortir votre restitution et de la compléter au besoin.



## Déroulé de l'atelier

### ● Introduction (15min)

Présentez votre restitution pour introduire l'atelier et donner le même niveau d'information à tous les participants.

### ● Travail individuel (15min)

Demandez aux participants de définir 5 objectifs maximum qui permettront de mesurer la réussite du Design System.

### ● Mise en commun (45min)

1. Inviter les participants à présenter leurs objectifs et à les expliquer.
2. Faire voter les participants sur les objectifs les plus pertinents en leur donnant la possibilité de ne voter que pour 5 objectifs au total. Ceux-ci deviendront les objectifs prioritaires.

N'ajoutez pas des objectifs pour en ajouter. Si vous n'en avez que 1 ou 2, c'est très bien, le principal c'est qu'ils soient réalistes et réalisables !

### ● Travail en groupe (30min)

Constituez deux groupes qui réfléchiront sur les indicateurs mesurables qui permettraient d'évaluer le succès de votre Design System.

### ● Mise en commun (30min)

Chaque groupe partage ses indicateurs, les valident ou les réfutent. Pensez aussi à définir les outils qui peuvent vous aider à les mesurer.

# Définir le niveau de maturité de la documentation

## DURÉE

1h

## MATÉRIEL

Stylo  
Post-it (idéalement, 2 couleurs)  
Tableau blanc

## PARTICIPANTS

Designers  
Développeurs  
Utilisateurs du Design System  
PO / PM du Design System (facultatif)  
1 animateur / facilitateur

## Objectif

Évaluer la maturité de la documentation du Design System au niveau graphique, fonctionnel et technique. Puis, définir en équipe les prochaines étapes ou axes d'amélioration.

## Préparation de l'atelier

Pas de préparation nécessaire.

À la fin de cet atelier, se présenteront plusieurs issues possibles :

- La documentation est incomplète ou insuffisante et l'équipe se répartit les actions nécessaires pour la compléter.
- L'équipe constate un besoin d'égalisation du niveau de la documentation car l'un des domaines se trouve en retard par rapport à l'autre. Dans ce cas, les efforts vont se concentrer pour rattraper cet écart.
- Les critères essentiels ont été remplis. Il est alors possible de refaire le même atelier en se questionnant sur d'éventuels manques ou sur les critères qui n'ont pas été retenus lors du vote.
- Il se trouve que vous avez répondu à tous vos besoins et que la documentation est complète. Félicitations !

## Déroulé de l'atelier

### ● Introduction (10min)

Demandez aux participants d'écrire sur les post-it les éléments indispensables de la documentation (de façon propre au projet) dans chacun des deux domaines (graphique et fonctionnel / technique). Utilisez une couleur par domaine.

### ● Mise en commun (25min)

1. Les participants présentent leurs critères et les expliquent.
2. Quand tout le monde est passé, les participants votent pour les critères les plus pertinents. Chaque participant a une limite de 5 points. Les critères les plus votés serviront de base de référence pour évaluer la maturité de la documentation.

### ● Évaluation (20min)

1. Dessinez sur le tableau un axe vertical et un axe horizontal axe X et Y. L'axe horizontal figure le niveau de maturité de la documentation UI/UX, et l'axe vertical représente le niveau de maturité de la documentation technique. Faire voter les participants sur les objectifs les plus pertinents en leur donnant la possibilité de ne voter que pour 5 objectifs au total. Ceux-ci deviendront les objectifs prioritaires.
2. Chaque participant positionne sur le graphique le niveau où il considère que l'équipe se situe en termes de documentation, sur la base des critères votés.
3. Une fois chaque participant passé, l'équipe définit les actions nécessaires à attribuer aux membres de l'équipe pour améliorer la documentation.

# Structurer sa documentation

## DURÉE

1h30 – 2h

## MATÉRIEL

Stylo  
Post-it  
Tableau blanc

## PARTICIPANTS

Équipe du Design System  
Utilisateur finaux de la documentation  
1 animateur / facilitateur

## Objectif

Cet atelier fonctionne comme un tri par cartes. Il permet de lister le contenu nécessaire à votre documentation, de catégoriser et de nommer ses différentes parties.

## Préparation de l'atelier

Pas de préparation nécessaire si l'atelier s'effectue en présentiel. Si vous le faites en ligne, pensez à préparer votre board numérique.

## Déroulé de l'atelier

### ● Introduction (10min)

Expliquez le but de l'atelier, faites des groupes multidisciplinaires et donnez à chacun un lot de post-it. Si vous n'êtes pas assez nombreux pour constituer des groupes, vous pouvez le faire de manière individuelle.

### ● Travail en groupe (30min)

Chaque groupe va noter sur des post-it les éléments qu'il juge pertinents pour une documentation.

### ● Mise en commun (45min)

Chaque groupe passe l'un après l'autre pour ajouter ses post-it. A la fin de la mise en commun, vous devriez commencer à voir un regroupement qui se construit.

Une fois chaque groupe passé, prenez une dizaine de minutes pour voir si un réarrangement/découpage se révèle utile.

### ● Travail en groupe (10min)

Demandez à chaque groupe de trouver un nom pour chaque catégorie construite.

### ● Mise en commun (20min)

Chaque groupe va passer chacun son tour pour nommer chaque regroupement. Dans le cas où plusieurs noms coexistent pour une même catégorie, faites voter les participants.

À la fin, vous obtiendrez les différents éléments nécessaires à votre documentation, ainsi que son regroupement et une nomenclature pertinente.

**WORLD OF DIGITS**

255 boulevard Pereire 75017 PARIS - TVA FR 45277780800 -

+33 1 82 64 51 51

[www.worldofdigits.com](http://www.worldofdigits.com)

Tous droits réservés. Cet ouvrage ne peut être revendu ou utilisé  
sans la permission de la société World of digits.