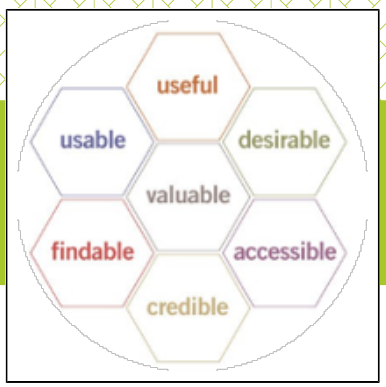


Programmiste, UX designer : même combat ?

Si vous suivez un peu les évolutions du digital, vous aurez constaté que depuis 2015, les deux lettres « UX » sont partout et sont souvent associées à « UX design », « design thinking », « UI Design » etc... A priori rien à voir entre le monde du bâtiment et celui du numérique. Et pourtant....
Le terme “Expérience Utilisateur”, ou UX, a été diffusé dans les années 90 par Donald Norman (auteur du célèbre “Design of everyday things” et aussi de la méthode de design centrée sur l’utilisateur). D’après la norme ISO ISO 9241-210, cela correspond « *aux réponses et aux perceptions d’une personne qui résultent de l’usage ou de l’anticipation de l’usage d’un produit, d’un service ou d’un système* ». Autrement dit, il s’agit de tenter de qualifier le résultat (bénéfice) et le ressenti de l'utilisateur (expérience) lors d'une manipulation (utilisation provisoire ou récurrente) d'un objet fonctionnel ou d'une interface homme-machine. Cette abréviation « UX » fait encore surtout référence au monde du digital et du numérique.

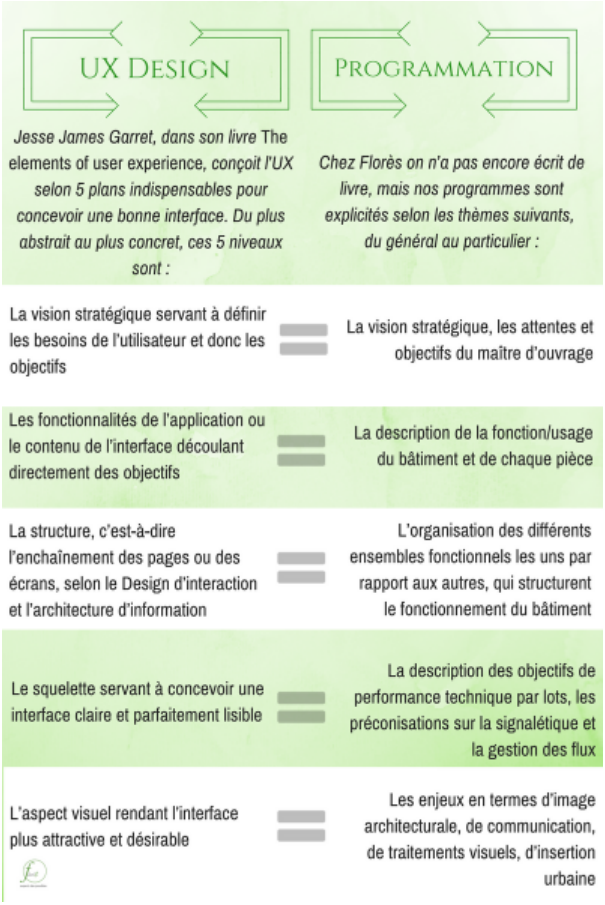
L’objectif ? Rendre les logiciels et sites internet :

- **plus utiles**, car toutes les fonctionnalités n’ont pas toujours leur place
- **plus utilisables**, car le logiciel doit être facile à utiliser
- **plus désirables**, car l’attrait visuel et émotionnel a beaucoup d’importance chez tout être humain
- **plus navigables**, car on ne veut pas perdre son temps à chercher des éléments
- **plus accessibles**, pour les personnes ayant des handicaps
- **plus crédibles**, parce qu’une bonne UX renforce la qualité perçue du logiciel

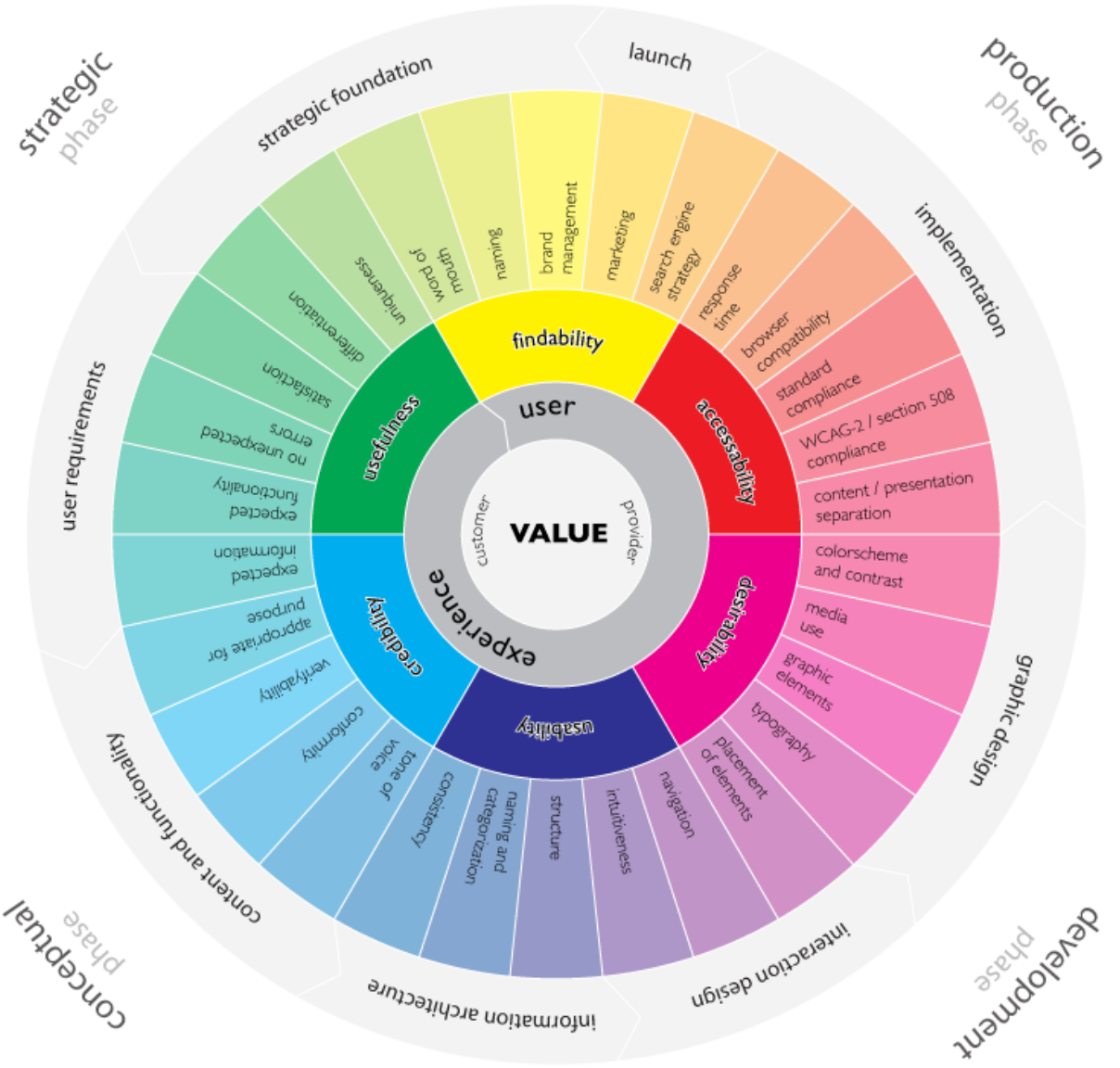
et ainsi fortement augmenter la valeur globale du logiciel.



On s'aperçoit donc facilement que le champ d'action de l'Expérience Utilisateur est en réalité très vaste et requiert de nombreuses compétences transversales comme l'ergonomie, la sociologie, la psychologie, l'ingénierie web, la marqueterie, le graphisme, la statistique etc. Donc finalement toute expérience vécue, en interaction avec un dispositif, qu'il soit digital ou non, ce serait de l'UX ! Amusez-vous donc maintenant à remplacer le mot « logiciel » par « bâtiment » dans la liste ci-dessus et vous comprendrez mieux le titre de cet article. En comparant les étapes à suivre pour bien concevoir l'UX et celles à suivre pour faire un bon programme on trouve vite les parallèles :



Le programmiste se frotte quotidiennement à l'Expérience Utilisateur : dans sa démarche de concertation initiale et tout au long de l'étude : en phase de diagnostic de site par exemple. Pour repérer les dysfonctionnements ou les points forts d'un bâtiment, il peut avoir à se glisser dans la peau d'un utilisateur lambda, en immersion complète dans la vie du lieu, pour s'imprégner compléter du contexte du projet, emprunter les mêmes circuits et se poser les mêmes questions qu'un UX designer : le chemin est-il clairement indiqué ? Le cheminement est-il intuitif ? Suis-je facilement arrivé à destination ? **Programmation architecturale et UX design semblent donc se baser sur une méthodologie semblable. Et si le croisement de ces deux mondes pouvait les enrichir l'un et l'autre ? Nous essaierons de répondre à cette question dans un prochain article. En attendant, nous vous mettons au défi de trouver quel serait le pendant de la roue de l'Expérience Utilisateur, pour un projet de bâtiment !**


strategic phase
strategic foundation
launch
production phase
implementation
graphic design
interaction design
development phase
information architecture
content and functionality
conceptual phase
user requirements
user
VALUE
customer
provider
experience
findability
accessability
desirability
usability
credibility
usefulness
naming
brand management
marketing
search engine strategy
response time
browser compatibility
standard compliance
WCAG-2 / section 508 compliance
content / presentation separation
colorscheme and contrast
media use
graphic elements
typography
placement of elements
navigation
intuitiveness
structure
naming and categorization
consistency
tone of voice
conformity
verifyability
appropriate for purpose
expected information
expected functionality
no unexpected errors
satisfaction
differentiation
uniqueness
word of mouth

C.G. et N.J.