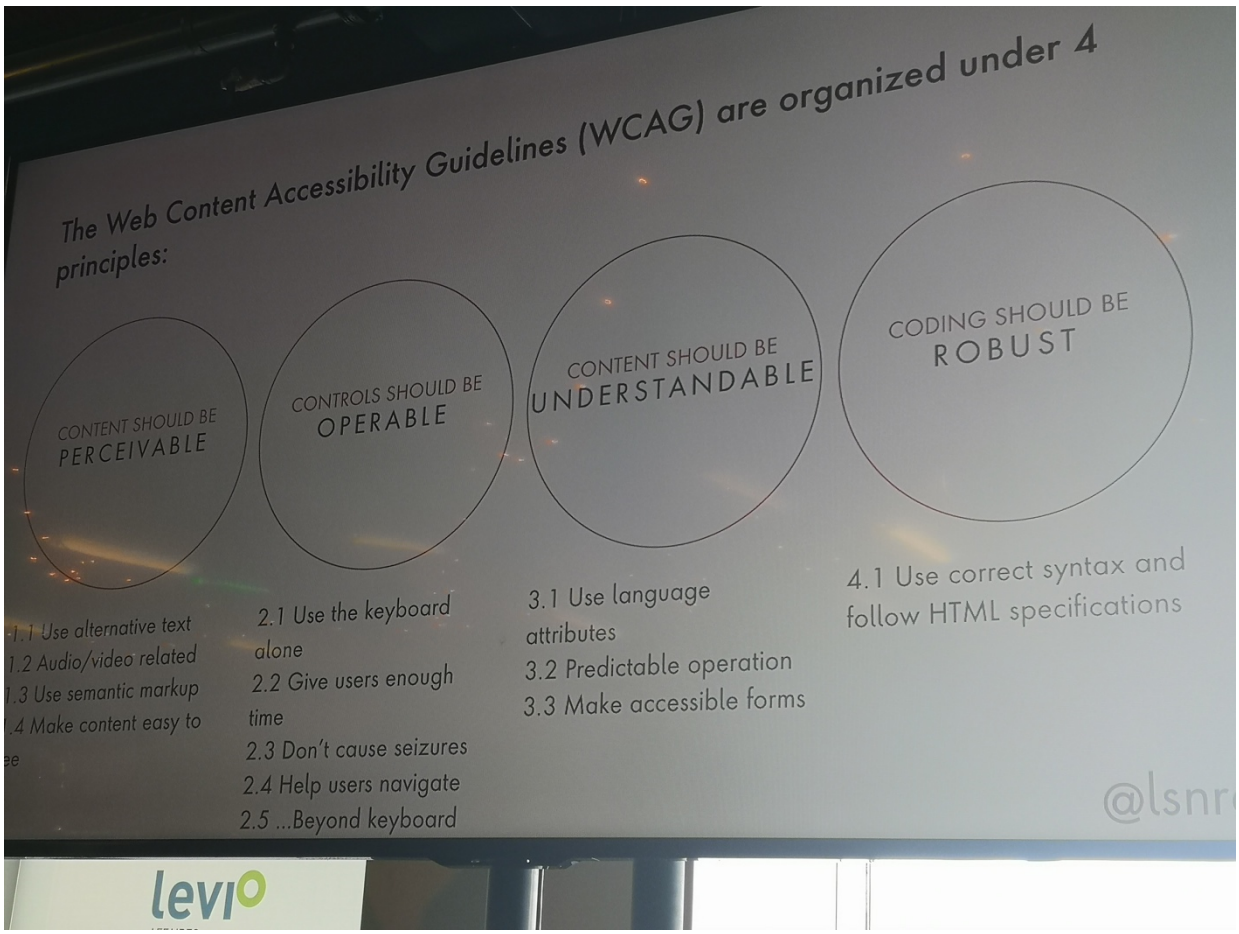
Compte-rendu de conférence WAQ 2019

Create accessible experiences

Alison Walden
Director of Technology, Publicis Sapient, Canada

Il y a beaucoup de populations à prendre en compte lorsqu'on parle d'accessibilité. Les personnes handicapées, mêmes temporairement, représente 16% de la population. Certaines peuvent être considérées comme des handicaps pour la navigation sur internet (affectant la vision ou la mobilité). L'accessibilité Web veut dire que les personnes ayant des handicaps peuvent percevoir, comprendre, naviguer et interagir avec Internet.

Ci-dessous : Les principes du WCAG (Web Content Accessiblity Guidelines)



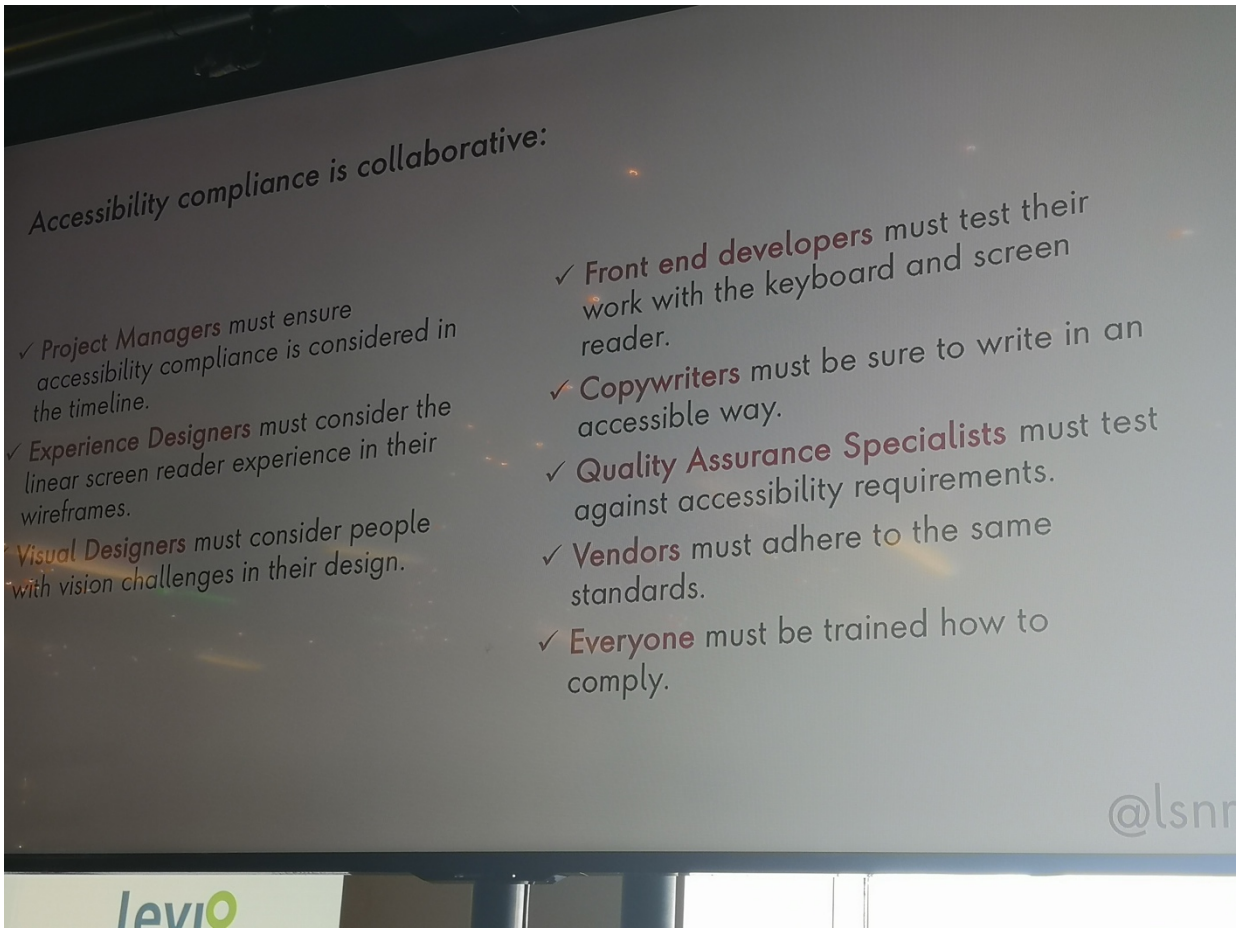
Le WCAG est divisé en trois niveaux de conformité : A > AA > AAA

AAA est bien trop difficile à mesurer et la plupart des critères sont trop subjectifs. Aussi, il est bon de viser pour le niveau AA dans un premier temps. Il faut savoir que moins d'1% des sites sont accessibles. Imaginez ne pas pouvoir être capable d'accéder à n'importe quelle information ?

Design is not just what it feels like, it's how it works

Steve Jobs

Le problème aujourd'hui est qu'on demande a des designers et développeurs parfaitement valides, qui n'ont jamais utilisé qu'une souris et un clavier pour naviguer, de mettre en place des expériences de navigation qu'ils ne connaissent pas (*screen reader*, sans souris, etc)

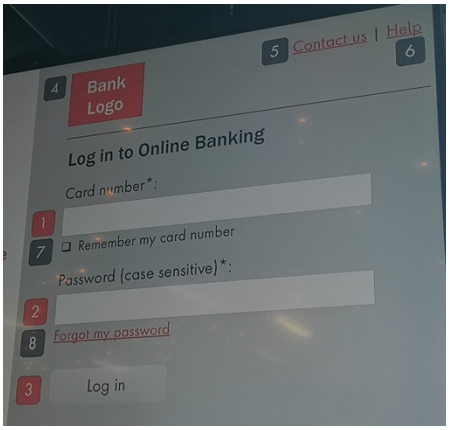


Les problèmes les plus communs :

1. Le menu qui ne se ferme pas en sortie avec *tab*.
2. La modale qui ne se ferme jamais avec seulement le clavier

3. Un *markeup* HTML invalide, avec des liens vides (ajouter *title* ou un *alt* ou même du texte *hidden* dans le span)
4. Des formulaires inaccessibles
    - Tous les champs doivent avoir des labels en *markeup* naturel et non ajoutées programmatiquement
    - Même les champs sans labels visibles doivent avoir un label pour le lecteur d'écran
    - Les messages d'erreurs doivent être lisibles par les lecteurs d'écrans
    - Les utilisateurs peuvent soumettre le formulaire même si cela donne une erreur
    - Le bouton de soumission à l'apparence d'un bouton *disabled* mais ne l'est par vraiment car le lecteur d'écran ne pourra jamais passer dessus.
5. Ne pas séparer le style de la structure.
    - Par exemple ne pas styliser les *h1*, *h2*, *h3* mais donner des niveaux de style afin de pouvoir avoir un h2 avec un niveau de style 3 mais respecter quand même l'ordre des *headings*
6. Surutilisation de *ARIA* quand le HTML sémantique simple suffit
    - Par *exemple <a href=""role="link">* alors que c'est DÉJÀ un lien...
7. Mal utiliser les attributs *ARIA* et descriptions
    - Les attributs *ARIA* ne remplacent pas un nom de bouton ou de lien. Ex : *<button aria-label="four dour vehicule highlights" ...> 4doors </button>*
8. Ne pas utiliser les *captions* et *scopes attributes* pour un tableau de données
    - *<table><caption>Statement</caption><tbody><tr><th scope="col">etc.*
9. Des calendriers inaccessibles
    - Vraiment difficile à rendre accessibles. voir *airbnb.io/react-dates/*
    - Ne jamais forcer à utiliser le *date picker* mais pouvoir écrire la date

10. L'ordre des focus n'est pas logique (navigation clavier)
    - "Logique" dans le sens de lecture naturel du site dans sa langue
    - Attention aux sauts de fonctionnalités
    - Exemple image ici-bas



Toutes ces choses doivent être livrées dans les wireframes !

Outils et méthodes :

- WAVE
- Google Lighthouse
- Axe accessibility by Deque
- Achecker

Comment tester manuellement l'accessibilité d'un site :

- Construire la fonctionnalité
- Tester avec le clavier seul
- Tester avec le clavier et le lecteur d'écran

Si vous avez des questions et/ou voulez parler de cette conférence, venez me voir !

Jules.