

Docs-as-Code

Table of Contents

Les supports de cours sont bons quand :	1
Quelle est l'approche de <i>Docs as Code</i> ?	1
Avantages de <i>Docs as Code</i>	1
Le Doc as Code intègre :	2
Pour synthétiser :	2
Apports pour O'Clock ☐	3
Voir plus loin pour O'Clock ☐	3
Et encore plus loin pour O'Clock ☐	3
Et toujours plus haut pour O'Clock ☐	4
Documentation complète	4
Logique d'organisation des cours	4
Logique d'organisation d'un module	4

Les supports de cours sont bons quand :

- Ils sont à jour
- Ils sont faciles à parcourir
- Ils sont faciles à gérer

Quelle est l'approche de *Docs as Code* ?

Ecrire de la documentation avec les mêmes outils que le code.

NOTE

Documentation as Code (*Docs as Code*) fait référence à une philosophie selon laquelle on doit écrire de la documentation avec les mêmes outils que le code.

Avantages de *Docs as Code*

- Qualité et Efficacité

Les avantages d'un tel dispositif sont nombreux :

NOTE

- Le rédacteur se concentre uniquement sur le contenu et non sur la mise en forme comme c'est malheureusement trop souvent le cas avec les éditeurs de texte ou slide type Google Slide.
- Le même code source permet de générer des formats différents : pdf, html, slides...
- Le travail collaboratif est grandement simplifié : code review, merge request, etc.
- Gestion des versions par branches ou tags git
- Edition de la documentation accessible à tous avec un simple éditeur de texte puisque la documentation est composée de fichiers texte.
- Génération de supports de qualité apportant une satisfaction des utilisateurs (apprenants, formateurs...).
- Génération de supports accessibles aux malentendants et non voyants.
- Mise en place de pattern de contenu de formations et d'organisation.

MONTRER LE RESULTAT + PATTERN

Le Doc as Code intègre :

- Des langages performants
- Des outils performants
- Du chainage d'outils
- Des méthodologies efficaces

NOTE

- Des langages performant : AsciiDoc mais le Markdown est aussi possible
- Des outils performants : VSCode, AsciiDoctor, Drawio...
- Du chainage d'outils : Github pour l'évolution, le versionning, l'automatisation, la génération, le partage...
- Méthodologie efficace : Agile, Pull Request

MONTRER LE CODE

Pour synthétiser :

- Drag and Drop Website Builder → Google Slide
- HTML → markdown
- PHP → asciidoc
- PHP/Symfony → asciidoc/asciiDoctor

NOTE

- Le HTML est au web ce que le markdown est à la documentation.
- Le PHP est au web ce que le asciidoc est à la documentation.
- Le PHP avec Symfony est au web ce que Asciidoc avec AsciiDoctor est à la documentation.
- Le tout dans github et la magie s'opère...

Apports pour O'Clock ☐

- Création, Evolution, Montée en compétence plus rapide
- Interopérabilité
- Mise en place simple

Voir plus loin pour O'Clock ☐

- Formateurs externes compatibles

NOTE

- Création des cours plus rapides donc moins de JH pour cette tâche.
- Evolution plus rapide donc moins de JH pour cette tâche.
- Montée en compétence plus rapide donc moins de JH pour cette tâche.
- Utilisation de standards et de pratiques efficaces donc reduction de la dette technique.
- Interopérable avec le fonctionnement actuelle donc changement en douceur et donc pas de perte lié au changement.
- Supports facilement exploitables pour des formateurs externes donc on peut facilement se faire un pool de formateurs externes 'ejectable' en cas de baisse soudaine d'activité. Pas de JH inactif. (De plus la trame permet de piloter l'orientation du cours et on s'assure que le formateurs exetrene respecte le programme)

Et encore plus loin pour O'Clock ☐

- Vidéos commercialisables en env clos
 - Pas de tournage à faire
 - L'apprenant est en autonomie
 - L'aprenant a accès à un tuteur
 - L'apprenant rempli des Quizz
 - L'apprenant est pris par un système de gamification

Et toujours plus haut pour O’Clock ☐

- Internationalisation simplifiée des supports

Documentation complète

- <https://docs.asciidoctor.org/asciidoc/latest/>

Logique d’organisation des cours

- Saison
- Une saison contient des épisodes
- Un épisode contient des modules
- Un module contient des démonstrations

Logique d’organisation d’un module

- Une module traite d’un sujet homogène
- Un module contient un sommaire
- Un concept doit être associé à
- 1 slide théorie
- 1 slide syntaxe
- 1 slide démonstration
- Un module contient une conclusion
- Un module contient une slide objectif/niveau