



**HAL**  
open science

# Repenser les chaînes de publication par l'intégration des pratiques du développement logiciel

Antoine Fauchié, Thomas Parisot

## ► To cite this version:

Antoine Fauchié, Thomas Parisot. Repenser les chaînes de publication par l'intégration des pratiques du développement logiciel. *Sciences du Design*, 2018, 8 (2), pp.45-56. 10.3917/sdd.008.0045 . hal-04336801

**HAL Id: hal-04336801**

**<https://hal.science/hal-04336801v1>**

Submitted on 11 Dec 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - ShareAlike 4.0 International License

# Repenser les chaînes de publication par l'intégration des pratiques du développement logiciel

**Antoine Fauchié**

Enseignant à l'IUT2 de Grenoble (Université Grenoble Alpes)  
Responsable d'Unité d'Enseignement du Master Publication de l'École Nationale Supérieure des Sciences de l'Information et des Bibliothèques  
antoine@quaternum.net

**Thomas Parisot**

Développeur, facilitateur, recherche en expérience utilisateur dans la coopérative dtc innovation.  
hi@oncletom.io

**Mots-clés**

**AsciiDoc  
Web  
Agilité  
Chaîne de publication  
Design itératif**

**Keywords**

**AsciiDoc  
Web  
Agile Practices  
Publishing Toolchain  
Iterative Design**

## Résumé

La rencontre des univers du livre et du développement logiciel est à l'œuvre dans plusieurs expérimentations de chaînes éditoriales, faisant apparaître de nouvelles approches et de nouvelles pratiques de design. Les étapes du processus d'édition sont réévaluées, repensées, re-conçues, notamment par l'avènement d'un environnement profondément numérique. Quelles sont les influences des méthodes et des outils du développement web sur les chaînes de publication des livres ? Nous nous focaliserons ici sur quatre aspects : la modularité des étapes et outils d'édition, l'ouverture des formats, la réduction de la distance entre le contenu et ses usages et l'économie acquise qui concerne principalement l'humain. Cet article constitue un bref panorama des efforts nécessaires pour envisager une évolution des chaînes d'édition, en sollicitant les pensées d'Ivan Illich et de Gilbert Simondon. Nous interrogerons des systèmes innovants inspirés de la méthode Agile, chère au monde du développement logiciel.

## Abstract

The book printing and the web development universes are currently sitting next to each other as new publishing toolchain are being trialed, showcasing new design approaches. The various steps of the publishing toolchain are being reevaluated, overhauled, redesigned, especially to welcome a more digital environment. What is the influence of web development practices amongst traditional publishing toolchains ? We will investigate this through four key aspects : the toolchain modularity, open formats, the shortening of the feedback loop and the benefits for us, people. This article will present a brief panorama of publishing toolchains alongside specific theories in philosophy of technical studies. We will challenge the introduction of innovative systems inspired by Agile methods which are popular in web development.

La publication de texte sur le Web repose sur un ensemble de méthodes et de technologies parmi lesquelles se trouvent des systèmes de gestion de contenu, des plateformes documentaires et des services en ligne d'édition de sites web. Les modes de conception, de fabrication, de production et de diffusion de ce domaine connaissent un renouveau qui s'illustre par exemple dans la mouvance des sites web sans base de données (Taillandier, 2016) ou dans l'immédiateté des processus de publication. Ce mouvement s'inspire des techniques issues de l'ingénierie informatique, dont les méthodes dites « agiles »<sup>01</sup>, ou encore le déploiement continu et les infrastructures à la demande (Morris, 2016). Quelles incidences ces bouleversements des technologies du Web ont-ils sur les métiers d'autres secteurs, et en particulier celui de l'édition ? Quelles manifestations et quels effets vont avoir les boucles de rétroaction sur la dynamique de nos systèmes (Meadows, 2008) ?

L'observation du domaine de l'édition de livres (numériques ou imprimés) permet de constater un fonctionnement *a priori* opposé aux méthodes dites « agiles ». Les rôles sont spécialisés et séquentiellement répartis entre auteur, relecteur, correcteur, graphiste, imprimeur, communicant, marketing et éditeur — ce dernier orchestre l'ensemble. L'écriture s'effectue avec un logiciel de traitement de texte et la collaboration avec un éditeur se pratique par échange de fichiers avec

01. Les méthodes agiles sont un ensemble de pratiques centrées autour de l'amélioration continue d'un produit ou d'un service. Elles se développent dans les années 1970 et se popularisent à partir de 2001, au moment de la publication du Manifeste Agile (Beck et al., 2001).

02. LaTeX est un langage de balisage et un système de publication créé dans les années 1980 par Leslie Lamport et Donald Knuth, notamment pour répondre à la double contrainte de la mise en forme de formules mathématiques et de la mise en page de qualité. XML est un langage de balisage extensible permettant de structurer un document et de le transposer dans d'autres formats.

03. Ce travail a déjà donné lieu à publication et communication (Fauché et Parisot, 2017; Fauché, 2017).

04. Le code source, son historique et les procédés de transformation sont consultables sur <https://gitlab.com/antoineint/>

05. PDF : Portable document format, est un langage de description de mise en page développé par la société Adobe en 1992. Ce format est largement utilisé pour l'échange de documents, offrant la possibilité de conserver la mise en forme. Le format PDF est une norme ISO depuis 2008.

06. Git est un système de gestion de versions créé en 2005 par Linus Torvalds.

07. EPUB (également orthographié Epub ou ePub) est l'acronyme de Electronic PUBLication. Ce format standardisé pour les livres numériques a été publié initialement en 2005. Il évolue en 2007 avec une structuration basée sur XML (Epub 2.0) puis en 2011 (EPUB 3) avec une structuration basée sur HTML qui devient une norme ISO en 2014 (EPUB 3.1 ISO/IEC TS 30135-1 :2014).

un service de partage de documents ou par messagerie électronique. La mise en forme graphique pour l'impression, et interactive le cas échéant, est réalisée avec un logiciel dédié de publication assistée par ordinateur.

Les publications académiques ont été pionnières en favorisant depuis plusieurs années des logiciels sémantiques éprouvés mais plus difficiles d'accès, basés sur LaTeX ou XML <sup>02</sup> (Muller, 2016).

Dans le domaine du logiciel, la publication d'un support d'installation a longtemps ressemblé à la publication d'un livre imprimé mais une transition est amorcée, par exemple par l'éditeur de logiciel Adobe, qui a abandonné le modèle de publication annuelle sur CD-ROM pour celui de mise à jour continue sur le *cloud*.

Pour un ouvrage technique à paraître chez l'éditeur Eyrolles, nous avons engagé un processus d'écriture reposant sur un système modulaire <sup>03</sup>. Cette démarche a été l'occasion de plusieurs bouleversements des processus éditoriaux à travers des étapes d'écriture publique, des suggestions et des retours possibles au fil de l'eau, des contenus statiques rendus interactifs, mais aussi l'intégration de l'éditeur et des relecteurs à toutes les étapes de ce procédé. Cette méthode et ces procédés techniques s'appliquent à l'écriture du présent article <sup>04</sup>.

Fort de l'observation des fonctionnements dans le secteur de l'édition et à partir de nos propres expérimentations, cet article se propose d'étudier la possibilité d'une écriture dans l'esprit du développement Web, qui n'aurait pas recours à un logiciel de traitement de texte ni à aucun logiciel de mise en forme dédié, ce jusqu'à la remise du manuscrit dans le format souhaité par l'éditeur. Ce retour d'expérience entend susciter des interrogations, encourager des approches de publication alternatives et partager des solutions techniques peu connues. Il s'agit d'étudier comment la transposition des techniques de conception itérative et de publication web à la réalisation d'un ouvrage imprimé fait muter les processus de conception et d'écriture, notamment à travers une approche centrée utilisateur.

## 1. — Vers une approche modulaire

### 1.1. — De la linéarité à la modularité

Les éléments qui forment les chaînes de publication des structures d'édition sont difficilement interchangeable. Remplacer un logiciel de publication assistée par ordinateur (PAO) par un autre est peu aisé. La gestion avancée des styles ou les fonctionnalités liées aux commentaires diffèrent sensiblement entre le logiciel de traitement de texte Microsoft Word et son équivalent ouvert et libre qu'est LibreOffice Writer. Cette non-interopérabilité des outils entretient une dépendance des utilisateurs et détermine une chaîne de publication séquentielle qui se décline comme suit : de Microsoft Word vers Adobe InDesign, conduisant à une sortie en PDF <sup>05</sup> ou en tout autre format destiné à l'impression. Chaque fichier est un ensemble fini qui appartient presque exclusivement à un seul logiciel (Dehut, 2018), celui-ci ayant la double fonction de visionneuse et de machine à écrire.

Au contraire, pour l'écriture de notre livre technique, nous nous sommes inspirés du développement web. Tout d'abord, le contenu source est lisible avec n'importe quel éditeur de texte. Ensuite, le contenu source est historisé avec Git <sup>06</sup> (Chacon et Straub, 2018). Ce contenu est modifiable en ligne. Enfin, les fichiers HTML, PDF et EPUB <sup>07</sup> sont générés automatiquement, dès qu'un changement est enregistré.

Toutes les personnes qui prennent part à un projet mené selon ces modalités voient le système dans son ensemble et ont la liberté d'agir à chaque étape du cycle. D'opérateur ou surveillant, l'auteur ou l'éditeur devient chef d'orchestre.

À la suite de Gilbert Simondon, il s'agit de repenser les rapports entre humains et machines, entre humains et programmes : « [L'humain] est parmi les machines qui opèrent avec lui » (Simondon, 2012, p. 11). Se positionner plus justement consiste non pas à être dedans comme avec un logiciel qui fonctionne tel une boîte noire, mais *au-dessous*, dans la création et l'ajustement d'éléments et de briques techniques, et *au-dessus*, dans l'orchestration de l'ensemble technique que forme cette chaîne de publication.

## 1.2. — Revenir aux sources : l'exemple d'AsciiDoc

Tenter d'amorcer la production d'un livre selon les modalités de publication d'un logiciel informatique a fait émerger la question des formats, une des caractéristiques fondamentales des affordances numériques. Quels formats alternatifs à ceux des logiciels de traitements de texte proposera une écriture aussi riche ? Quelles sont les incidences d'un tel choix ? Les développeurs et développeuses sont des adeptes de langages de balisage légers. Ces syntaxes placent le contenu et les règles de présentation au même niveau, sous forme textuelle, sans que cela n'entrave la lecture. Conçu pour simplifier l'écriture de fichiers HTML, Markdown<sup>08</sup> est un format populaire dans le domaine. Markdown ne possède pas l'expressivité grammaticale d'un logiciel de traitement de texte, au contraire de AsciiDoc, un langage de balisage léger moins connu vers lequel nous nous sommes tournés. Cela nous a permis de reproduire en quelques heures la structure du document dans le gabarit fourni par l'éditeur du livre.

08. Markdown est un langage de balisage léger créé au début des années 2000 par John Gruber.

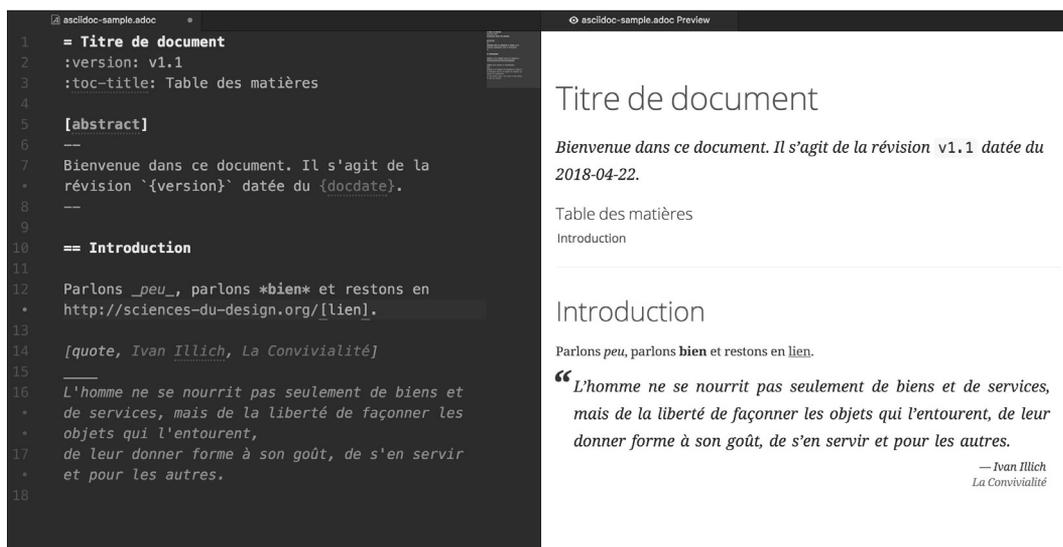


Fig. 1. — Exemple de la syntaxe AsciiDoc et de son rendu HTML : sur la partie gauche le langage de balisage léger en texte brut qui structure les contenus, sur la partie droite l'interprétation du balisage avec une attribution de mise en forme en fonction de la structuration et avec une feuille de style standard.

Le choix du format AsciiDoc pour structurer notre ouvrage a renforcé la portabilité et l'extensibilité de notre chaîne éditoriale. La brique logicielle qui comprend la syntaxe AsciiDoc peut être remplacée par une autre. Ainsi, le contenu des documents est préservé lorsque la chaîne de publication évolue. L'extensibilité est permise par l'outillage AsciiDoc qui accueille des fonctionnalités créées et partagées par d'autres développeurs.

Les prémisses de cette chaîne de publication ont des points communs avec le développement informatique moderne : écrits consignés dans des fichiers texte (AsciiDoc), versionnés (Git), puis distribués sur une plateforme collaborative (GitHub).

### 1.3. — Pourquoi versionner plutôt que numéroter ?

Le logiciel de contrôle de versions Git (Demaree & Brown, 2017) a pour vocation première la gestion d'historique des modifications de fichiers texte. Plutôt que de numéroter chaque nouvelle version ou d'écraser les précédentes, chaque changement déclaré crée un nouveau point sur une ligne de temps. Son usage a été popularisé grâce à des plateformes collaboratives comme GitHub ou GitLab. Leur force est de rendre accessible l'historique sous forme de pages web qui sont plus simples d'accès que l'outil informatique complexe qu'est le terminal.

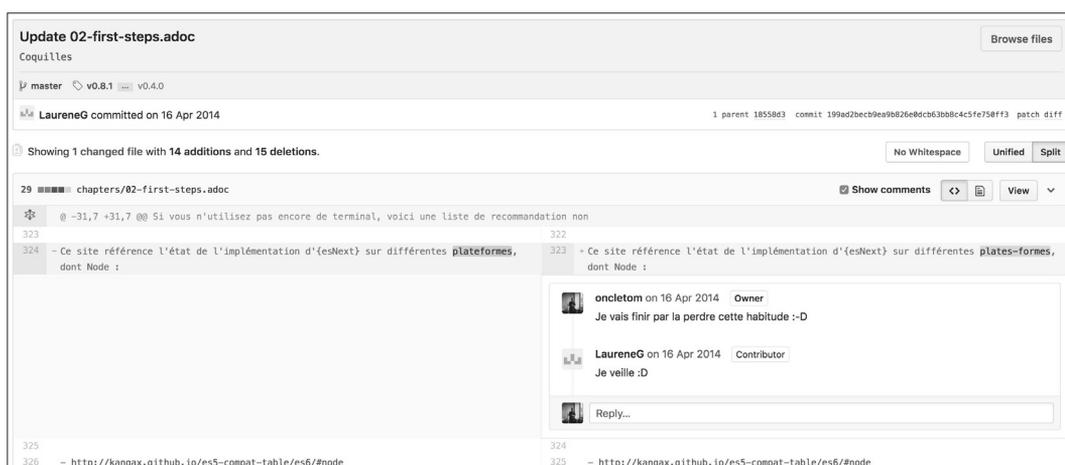


Fig. 2. — Affichage d'une modification de fichier sur la plateforme GitHub accompagnée d'une interaction entre l'auteur et son editrice.

L'historique généré et distribué par Git se partage et se synchronise entre plusieurs ordinateurs. Si deux personnes modifient le même morceau d'un fichier, Git nous alerte alors de la présence d'un conflit et invite à choisir la version du contenu à privilégier sans perdre le travail déjà effectué.

La conservation et l'organisation de versions parallèles est possible grâce à l'évolution de la ligne de temps de Git en arborescence. Cette fonctionnalité permet d'amorcer des brouillons, d'effectuer des corrections à la marge, ou encore d'expérimenter sans subir de répercussion sur la copie de travail principale. Les branches sont fusionnées dans leur intégralité ou en sélectionnant les changements les plus pertinents.

L'historique collaboratif et interactif de Git respecte la variété des rythmes de travail d'une équipe, sans interrompre, ralentir ou altérer le travail. En choisissant d'appliquer les méthodes de développement web à l'écriture d'un ouvrage technique, nous avons retrouvé des habitudes acquises dans nos pratiques de développeurs et redécouvert la convivialité de cet outillage. Ce terme de convivialité est à entendre ici au sens développé par Ivan Illich qui met en tension d'une part la prise de pouvoir de l'outil sur les capacités de choix des humains qui les utilisent, et d'autre part leur puissance émancipatrice. En effet, comme l'affirme Ivan Illich :

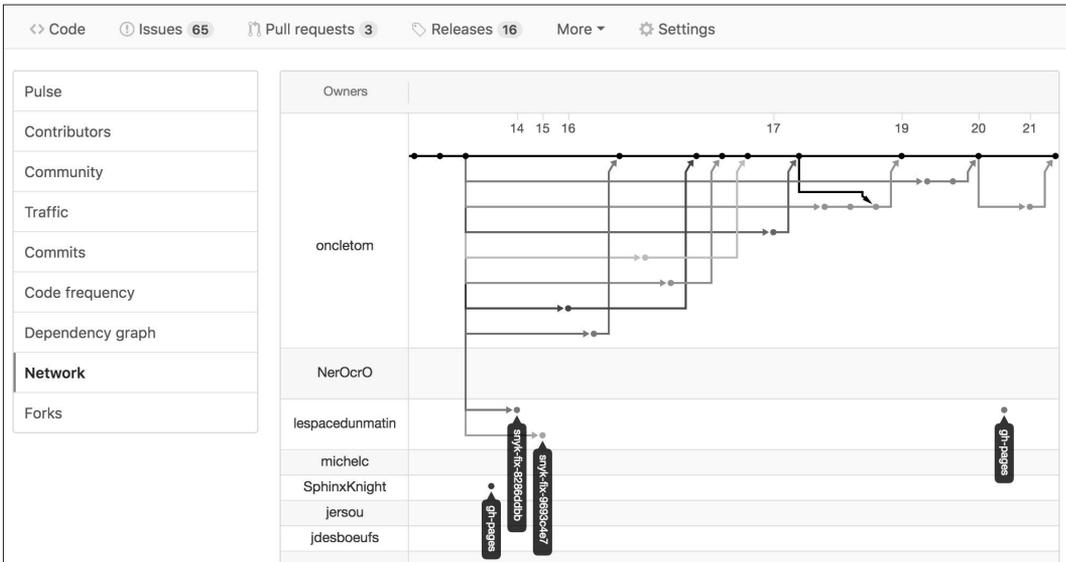


Fig. 3. — Visualisation d'une ligne de temps d'un livre versionné avec Git : chaque point de fusion d'une version parallèle matérialise une boucle de rétroaction.

« Pour autant que je maîtrise l'outil, je charge le monde de mon sens ; pour autant que l'outil me domine, sa structure me façonne et forme la représentation que j'ai de moi-même. L'outil convivial est celui qui me laisse la plus grande latitude et le plus grand pouvoir de modifier le monde au gré de mon intention. » (Illich, 1973, p. 43)

Nous éprouvons la puissance émancipatrice de l'outil en concevant, façonnant et utilisant un outil sur mesure – l'équivalent d'un traitement de texte, combinant AsciiDoc et Git, respectivement, pour le formatage des contenus et leur versionnement. Comment le mécanisme d'interprétation et de transformation de l'outillage AsciiDoc va-t-il permettre ou empêcher de se conformer au gabarit de l'éditeur ?

#### 1.4. — Interpréter, transformer, automatiser

L'outillage AsciiDoc repose sur un fichier plat qui est lu caractère par caractère afin de créer une représentation de la structure du document sous forme d'arbre sémantique. Celui-ci est ensuite transposé dans une autre structure, celle du format de destination.

L'arbre sémantique d'Asciidoc se modifie grâce à un mécanisme d'extensions. Nous sommes en mesure de le parcourir pour vérifier l'orthographe, appliquer les règles de microtypographie française mais aussi ajouter de nouveaux formats d'exportation ou transformer des exemples de code en bacs à sable interactifs.

Nous hébergeons sur les plateformes collaboratives GitHub ou GitLab à la fois les sources de notre texte, le gabarit de présentation et les scripts d'automatisation qui reproduisent la « recette » de transformation pour chaque changement. Le système est entièrement décrit avec des fichiers texte, comme posé à plat devant nos yeux, au même titre que les mouvements inhérents à son fonctionnement.

#### 1.5. — Appréhender un changement

S'il est plus facile pour un développeur ou une développeuse d'écrire un livre comme un logiciel informatique, qu'en est-il du chemin à parcourir pour les acteurs et actrices d'une chaîne de publication ? Nos expérimentations ont révélé

```

104 lines (103 sloc) | 3.19 KB
Copy Raw Blame History
67     "level": 0,
68     "blocks": [
69       {
70         "attributes": {
71           },
72         "context": "paragraph",
73         "nodeName": "paragraph",
74         "roles": [
75           ],
76         "level": 0,
77         "lines": [
78           "Lorem ipsum dolor sit amet, *consectetur* adipiscing elit. Donec _aliquam_ tempus quam et sollicitudin."
79         ],
80         "source": "Lorem ipsum dolor sit amet, *consectetur* adipiscing elit. Donec _aliquam_ tempus quam et sollicitudin."
81       },
82       {
83         "attributes": {
84           },
85         "context": "paragraph",
86         "nodeName": "paragraph",
87         "roles": [
88           ],
89         "level": 0,
90         "lines": [
91           "Ut commodo _lorem_ nec lacinia finibus.",
92           "Etiam *eu* ornare lectus.",
93           "In molestie `ut` augue a ultricies."
94         ],
95         "source": "Ut commodo _lorem_ nec lacinia finibus.\nEtiam *eu* ornare lectus.\nIn molestie `ut` augue a ultricies."
96       }

```

Fig. 4. — Visualisation de l'arbre sémantique d'un document AsciiDoc : chaque ligne du texte source est contextualisée par un ensemble d'attributs qui vise à produire le résultat attendu, dans le format souhaité.

qu'un des gains majeurs est le suivi en temps réel. L'augmentation de la qualité des échanges est corrélée à la diminution de leur volume. Les e-mails ont donc été réservés au partage d'idées et de questionnements, tandis que les notifications ont fluidifié les échanges techniques et que l'historique des versions pouvait être consulté en ligne. D'autre part, l'organisation des relectures en sessions a révélé, outre l'amélioration des contenus, un enthousiasme des lecteurs et lectrices vis-à-vis de la plateforme elle-même.

Du modèle d'organisation séquentielle vers celui de système dynamique, une mutation opère où travaille la notion d'objet concret telle que la développe Gilbert Simondon (2012, p. 41) : l'environnement réunissant des acteurs spécialisés mais isolés les uns des autres laisse place à l'écosystème modulaire dont les éléments rétroagissent entre eux. Ce mode d'organisation permet une souplesse d'adaptation face aux événements extérieurs et aux logiques financières. Ce n'est donc pas seulement d'un changement d'outils dont il est question, mais bel et bien d'un changement de culture.

## 2. — Vers la naissance d'un nouvel écosystème

### 2.1. — Réduction de la distance : publier vite, publier souvent

L'un des apports déterminants du numérique pour la publication est la génération d'artéfacts en amont des étapes finales d'édition. Nous disposons de prévisualisations ou d'aperçus imprimables du livre dans différents formats (PDF,

HTML, EPUB) à des étapes intermédiaires du processus. Les chaînes de publication classiques ne permettent qu'une version PDF précédant les traditionnelles épreuves. La distance entre l'auteur, les acteurs du projet et l'ouvrage final est donc singulièrement réduite.

Dans le monde du développement informatique, le déploiement continu joue ce rôle. Un programme, une application ou un site web peuvent être testés et utilisés bien avant leur version finale – si tant est qu'il puisse y avoir une version dite « finale ». Le déploiement continu repose sur l'absence d'opération manuelle et sur le maintien de l'intégrité de la chaîne durant le processus. Il s'agit à la fois de gérer le flux des modifications proposées par les différentes personnes travaillant sur un même projet, puis de tester et de valider ces propositions sans mettre en péril l'existant.

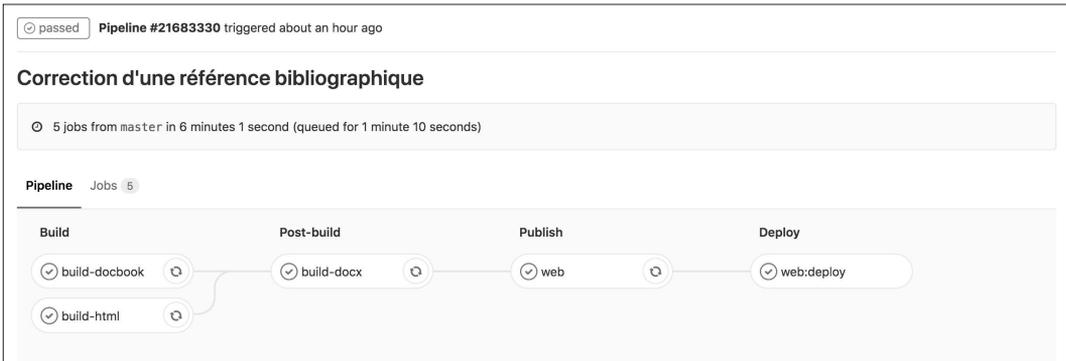


Fig. 5.. —Génération automatisée d'épreuves Word, DocBook et HTML depuis une source AsciiDoc: visualisation des différentes phases de génération de formats spécifiques sur la plateforme GitLab.

La maxime empruntée au développement informatique *Release Early, Release Often*<sup>09</sup> (Raymond & Young, 2001) est un appel à privilégier des formes et des formats intermédiaires pour une meilleure compréhension du projet. Cette matérialisation de l'amélioration continue est transposable au design itératif appliqué aux contenus, au système de publication et au processus d'écriture.

09. Publier vite, publier souvent.

## 2.2. — Charge cognitive et appropriation

Dans les étapes d'écriture de notre ouvrage, nous constatons que la lourdeur imputable à l'étalement d'un projet dans le temps est amoindrie grâce au processus de publication en continu. Nous consultons le contenu dans sa mise en forme visuelle tout au long de l'écriture, au lieu de le découvrir seulement dans la dernière ligne droite du projet.

Dans le domaine de l'édition technique, la relecture d'un premier chapitre peut commencer alors que l'écriture de la partie suivante a débuté. Les retours recueillis alimentent une logique d'amélioration continue. Les acteurs et actrices du projet s'ajustent au rythme de l'autre (Ilich, 2014). La mise en page peut ainsi se préparer dès le deuxième chapitre, participer à la boucle de retours et influencer la phase d'écriture. La prise en compte et la compréhension des erreurs s'appréhendent rapidement et s'ajustent en cours d'écriture.

Nous sommes conscients qu'un fonctionnement par itérations ne s'applique pas à tout projet éditorial et que cette écriture programmatique convient à des domaines spécifiques, parmi lesquels se trouvent les livres techniques. Si la structuration et la progression d'une réflexion ne se limitent pas à la succession de modifications dans un texte, cette dimension peut toutefois être bénéfique,

y compris pour la rédaction d'ouvrages académiques. Aussi, une activité d'édition dépasse largement le cadre de production d'un livre et la dimension de flux (Tangaro, 2017) concerne la gestion d'un texte, non celle d'une pensée ou d'un mouvement.

### 2.3. — Réduire l'effort des acteurs d'un projet

Nous avons évoqué le format de document AsciiDoc comme étant une syntaxe à balisage léger. Ce format a été conçu comme une représentation textuelle d'un autre format, DocBook <sup>10</sup>. Il s'agit d'une syntaxe XML inventée en 1991 pour mieux gérer la documentation technique des systèmes d'exploitation UNIX <sup>11</sup>. L'éditeur américain O'Reilly <sup>12</sup> a participé à l'élaboration du format DocBook, ce n'est donc pas un hasard si ce format, celui d'AsciiDoc et celui de Git sont les pierres angulaires de sa chaîne de publication automatisée Atlas <sup>13</sup>. Cette infrastructure leur procure la capacité de publier aussi vite et aussi souvent que nécessaire, tout en allégeant la charge de travail, faite de tâches ingrates et répétitives. Ce temps et cette énergie reconquises offrent l'opportunité de comprendre le système dans son ensemble, de gagner en autonomie et en qualité.

L'effort nécessaire pour s'emparer d'un nouveau format d'export, ou pour le remplacement d'un élément de la chaîne de publication, se réduit peu à peu. Il est ainsi possible d'apprendre à façonner le contenu par itération en fonction de l'énergie collective au lieu de subir des bouleversements importants. C'est un point essentiel développé par Ivan Illich dans *La convivialité*, où il développe les termes de *façonnage* et d'*énergie* : « La ressource la mieux répartie dans le monde : l'énergie personnelle que contrôle la personne. » (Illich, 2014). Il s'agit de prendre en compte les humains qui travaillent sur une chaîne d'édition, leur vitalité individuelle et collective en tant qu'éléments du design du système de publication.

Le principe d'amélioration continue tournée vers le sens du projet se retrouve dans les décisions concertées, priorisées et consenties des méthodes agiles. Cette réduction d'effort offre la perspective d'une création de nouvelles dispositions de distribution des ouvrages, centrées sur le contenu et dérivables à volonté sur une grande variété de supports. Le livre devient alors un des artéfacts de destination, au lieu d'être le passage obligé avant une adaptation web ou sous forme de livre numérique.

### 2.4 — La possibilité d'un changement de paradigme

Utiliser les méthodes et les outils exposés ici est à la portée de toute personne ayant une culture et une pratique numérique : connaissance générale du Web et de son écosystème, notions en HTML, compréhension de la distinction structure et mise en forme. Notre expérience conforte l'hypothèse selon laquelle la difficulté d'appréhension d'un changement est culturelle avant d'être technique. Plus que des manques en compétence techniques, c'est vraisemblablement un ensemble d'habitudes et de culture qui freinerait une personne travaillant dans l'édition à adopter les principes présentés dans cet article : comment celle-ci pourrait-elle développer une appétence pour les langages de balisage léger, les systèmes de gestion de version ou le Web ? C'est-à-dire des modèles et outils répondant à un paradigme tout à fait différent de celui dont elle aurait suivi l'usage jusqu'alors. Deux préalables sont essentiels : le contexte et les interfaces.

Getty Publications, la structure d'édition du musée d'art The Getty, est un cas original dans le domaine de l'édition. Constatant une nécessité, l'équipe du département numérique a initié une transformation culturelle et technique (Gardner, 2017). La transition a démarré avec une petite équipe qui partageait cette culture pour ensuite être diffusée dans l'ensemble de la maison d'édition (Evan, 2016).

Ce changement repose sur une étape d'acceptation conduisant à transformer les processus, les habitudes et les outils. L'usage d'un logiciel comme InDesign n'est pas un passage obligé. Il repose également sur un changement de perception du statut du livre, considéré dès lors comme un des artéfacts de

10. <https://docbook.org/>

11. UNIX est un système d'exploitation créé par Kenneth Thompson en 1969.

12. <https://www.oreilly.com/>

13. <https://atlas.oreilly.com/>

publication plutôt qu'une finalité absolue. Il repose enfin sur une mutation des modalités de publication qui devient incrémentale sur le Web ou en e-book avant d'être promue à des publications papier.

## 2.5. — Le design responsif du livre

Au cours des années 2010, le *responsive design* a révolutionné la façon de produire des sites web (Marcotte & Keith, 2017) en favorisant l'émergence d'interfaces adaptatives plutôt que les interfaces statiques, en usage jusqu'alors. Le livre connaît un bouleversement similaire avec le format EPUB<sup>14</sup> : le texte peut être redimensionné, les paramètres typographiques se modifient en cours de lecture, l'éditeur perd la maîtrise absolue du rendu graphique.

Plutôt que de viser la parfaite correspondance entre conception graphique et matérialisation imprimée, dont le format PDF était l'intermédiaire, la priorité du format EPUB est la résilience : le texte reste lisible et reconfigurable quelle que soit la situation. Le contexte de lecture n'est plus seulement un objet aux contraintes fixes, il devient multiple – un flux qui s'adapte aux dispositifs de lecture (Tangaro, 2017).

Parmi les outils disponibles pour produire ces objets numériques, les premières solutions artisanales ont mis à mal les logiciels classiques. Un temps d'adaptation a été nécessaire pour comprendre les bouleversements, les nouvelles pratiques, les nouvelles attentes. Les enjeux se sont cristallisés sur un aspect spécifique du processus, à savoir la génération d'un fichier PDF imprimable possédant une puissance fonctionnelle comparable à celle du logiciel InDesign alors majoritairement utilisé. Actuellement, cette fonction est assurée par des solutions propriétaires et fermées comme PrinceXML (Cramer, 2017), mais une communauté se constitue afin d'apporter la richesse et l'ouverture des technologies et méthodes du Web à cette étape de mise en forme.

## 2.6. — Mettre à jour un livre, maintenir une chaîne

Travailler avec une même source pour l'ensemble du processus d'écriture, d'édition et de mise en forme graphique apporte une grande fluidité pour chacune de ces étapes. Mais que se passe-t-il si un des outils n'est plus mis à jour ? Quel sera l'impact sur la chaîne ? L'intérêt de la modularité est de pouvoir changer un élément et non la chaîne entière. L'écosystème à accès ouvert (*open source*) GNU/Linux repose sur ce principe (Stallman, 1985).

L'ouverture du code source des outils permet, au-delà d'une simple utilisation, de les forger<sup>15</sup> et de contribuer à leur développement. Ces interactions sont vectrices de transferts de compétences. Les étapes de développement, d'adaptation et d'implémentation sont motivées et mises en œuvre pour des projets concrets, permettant à chacun de constater en temps réel les résultats de son effort. L'auteur, l'éditeur, le lecteur, le designer graphique et les autres acteurs du projet éditorial ne sont plus seulement des consommateurs d'outils mais les acteurs d'un système qu'ils modèlent.

Les questions de compétences et de dépendances se déplacent vers la mise à disposition d'interfaces intermédiaires pour faciliter la compréhension et la manipulation du système de publication. La chaîne ne repose plus sur un logiciel spécifique, mais sur un assemblage de programmes, qui demeurent remplaçables tout en étant soumis aux compétences des administrateurs du système.

Les formats des fichiers et les logiciels sont ouverts, au même titre que les processus d'interaction – consulter, interroger, proposer, réaliser, faire réaliser, valider, etc. Dans ces conditions, il est difficile d'anticiper les mutations à venir de la chaîne d'édition. Mais quelle heureuse perspective que celle de pouvoir voir évoluer et façonner les outils qui permettent de concevoir et de produire des livres, quelle que soit leur forme.

14. Voir l'initiative « Lebook redistribuable » de Jiminy Panoz pour vulgariser ce principe (<https://jaypanoz.github.io/reflow/>).

15. Dans le domaine de l'informatique, une forge constitue un système de développement collaboratif permettant d'éditer un logiciel.

## Conclusion

Les éléments mobilisés pour l'écriture de notre ouvrage et analysés dans cet article ont en commun de réduire les écarts entre effort et résultat. D'une part, le format d'écriture à balisage léger est plus proche du système de fichier d'un ordinateur, ce qui permet de supprimer la barrière logicielle. D'autre part, l'organisation des actrices et acteurs se constitue en réseau plutôt qu'en séquence, ce qui dé-hiérarchise et facilite les échanges. Enfin, le contenu redevient central en dérivant tous nos artéfacts depuis un même fichier source, grâce à une chaîne intégrée, modulaire et extensible.

Notre expérience d'écriture nous a conduits à faire le constat d'une cristallisation des enjeux relatifs à la manière dont les livres sont produits, depuis l'outillage jusqu'aux modalités de collaboration. La transition numérique du développement web peut bénéficier aux chaînes de publication. En effet, des connaissances en HTML, CSS et JavaScript suffiraient à programmer l'entièreté de la chaîne de publication, du site web *responsive*, voire paramétrique, jusqu'au rendu graphique sur papier. La mise en place d'une telle chaîne de publication nécessite que les personnes impliquées soient expérimentées à la fois avec les systèmes logiciels et avec les systèmes de publication, tout en étant désireuses d'accompagner et de participer aux transformations systémiques plutôt que de se contenter d'utiliser un outil figé sans le mettre en question.

Le rapprochement des chaînes de publication de livres numériques et imprimés avec les méthodes et techniques de développement web ouvre la voie à une effervescence créative, tant du point de vue de l'expérience de lecture que de celui des nouveaux modèles de distribution. Afin de déployer le livre sous des formes et des dispositions diverses, il est nécessaire de transformer les modes de conception, de fabrication, de production et de distribution. De tels systèmes sont autant de propositions de réponse au concept d'hybridation du livre (Ludovico & Cramer, 2016) et autant de tentatives de faire exister un modèle pour la publication avec le numérique qui reposerait sur un cheminement itératif, sur des processus modulaires et sur des équipes multidisciplinaires.

## RÉFÉRENCES

- BECK, K. ET AL. (2001). Manifeste agile. Repéré à l'adresse <http://agilemanifesto.org/iso/fr/manifesto.html>
- CHACON, S., ET STRAUB, B. (2018). Pro Git Book. Repéré à l'adresse <https://git-scm.com/book/fr/v2/>
- CRAMER, D. (2017). Beyond XML : Making Books with HTML. XML.com. Repéré à l'adresse <https://www.xml.com/articles/2017/02/20/beyond-xml-making-books-html/>
- DEHUT, J. (2018). En finir avec Word! Pour une analyse des enjeux relatifs aux traitements de texte et à leur utilisation. Repéré à l'adresse <https://eriac.hypotheses.org/80>
- DEMAREE, D., & BROWN, M. (2017). *Git par la pratique*. Paris : Eyrolles.
- EVAN LAN, R. (2016). An Editor's View of Digital Publishing. Repéré à l'adresse <http://blogs.getty.edu/iris/an-editors-view-of-digital-publishing/>
- FAUCHIÉ, A. (2017). Écrire un livre en 2017. Repéré à l'adresse <https://www.quaternum.net/2017/03/07/crire-un-livre-en-2017/>
- FAUCHIÉ, A. (2017). Une chaîne de publication inspirée du web. Repéré à l'adresse <https://www.quaternum.net/2017/03/13/une-chaîne-de-publication-inspiree-du-web/>
- FAUCHIÉ, A., ET PARISOT, T. (2017). *README.book*. Communication présentée aux journées professionnelles Codeurs en Seine, Rouen.
- FAUCHIÉ, A ET PARISOT, T. (2018). *README.book*. Dépôt Git à l'adresse <https://gitlab.com/antoine1/readme.book>
- GARDNER, E. (2017). Publier des livres avec un générateur de site statique. Repéré à l'adresse <https://jamstatic.fr/2017/01/23/produire-des-livres-avec-le-statique/>
- ILLICH, I. (1973). *La convivialité*. Paris : Éditions du Seuil.
- LUDOVICO, A., ET CRAMER, F. (2016). *Post-digital print : la mutation de l'édition depuis 1894*. Paris : B42.
- MARCOTTE, E., ET KEITH, J. (2017). *Responsive Web design*. Paris, France : Eyrolles.
- MEADOWS, D. H. (2008). *Thinking in Systems: A Primer*. Londres : Earthscan Publications.
- MORRIS, K. (2016). Infrastructure as Code. Repéré à l'adresse <http://shop.oreilly.com/product/0636920039297.do>
- MULLER, C. (2016). Édition XML-TEI, écriture et document numérique : retour sur la Biennale du numérique 2015. Repéré à l'adresse <http://www.enssib.fr/recherche/enssiblab/les-billets-denssiblab/biennale-du-numerique-edition-numerique-encodage-xml-tei>
- PARISOT, T. (2018). *Node.js. Apprendre par la pratique*. Paris : Eyrolles.
- RAYMOND, E. S., ET YOUNG, R. M. (2001). *The cathedral and the bazaar : musings on linux and open source by an accidental revolutionary*. Sebastopol : O'Reilly.
- SIMONDON, G. (2012). *Du mode d'existence des objets techniques*. Paris : Aubier.
- STALLMAN, R. (1985). Le manifeste GNU. Repéré à l'adresse <https://www.gnu.org/gnu/manifesto.fr.html>
- TAILLANDIER, F. (2016). La mouvance statique. Repéré à l'adresse <https://frank.taillandier.me/2016/03/08/les-gestionnaires-de-contenu-statique>
- TANGARO, B. (2017). De la page au flux : la conception du livre numérique. Repéré à l'adresse <http://dlis.hypotheses.org/1255>