|  |  |
| --- | --- |
| **BTS SIO** | **BTS SIO (SLAM)** |
|  | **Méthodologie** |
|  |  |
| **Méthodologie organisation** | |

|  |  |  |  |
| --- | --- | --- | --- |
| **date** | **révision** | | |
|  | Timothée Robert | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
| **BTS SIO** | |  | **BTS SIO (SLAM)** |
|  | | |  |

|  |  |
| --- | --- |
| Objectif |  |

Vous êtes de retour de vacances, à la fois motivé par la rentrée mais inquiets et déboussolés car cela fait 3 mois que vous n’avez pas codé, effectué de requête SQL, ouvert un IDE etc

Votre organisation et votre méthode de travail est fondamentale, pour vous permettre d’améliorer votre productivité (la quantité de travail scolaire ou professionnel utile par unité de temps) mais aussi la qualité de votre travail.

La façon dont vous organisez vos données n’est pas innocente, elle fait toute la différence.

Plutôt que de m’épuiser à vous répéter éternellement les mêmes choses, j’ai choisi ici de documenter au maximum les différents types de document à utiliser et à produire ainsi que quelques bonnes pratiques.

|  |
| --- |
| Organisation des documents : cours, TD & TP |

Il s’agit ici des documents de cours, TD et TP, donc essentiellement des documents que je vous fournis, mais aussi ceux fournis par des collègues, ceux que vous allez trouver sur Internet, obtenus auprès de camarades d’autres établissements.

L’objectif est donc ici de les *retrouver le plus facilement possible lorsque vous en aurez besoin* : révisions, préparation évaluation ou étude de cas, support pour travail en entreprise.

L’organisation des documents consiste en la façon dont on ***range*** et ***nomme*** ses documents au format Word, open office, Excel, Pdf etc

**Accès aux documents**

Les documents sont organisés dans des répertoires, accessibles sur le réseau.

Les répertoires sont nommés par matière et éventuellement professeur si une matière est enseignée par plusieurs professeurs.

Exemples

BLOC3 - Cybersécurité : un seul professeur donc on ne précise pas

BLOC2 – Trobert : ici on précise car deux professeurs partagent l’enseignement du bloc 2

**Accès à la séance**

Chaque séance est stockée dans un sous répertoire. Cela permet aux étudiants qui ont manqué une séance (ou alors à ceux qui ont manqué plus de la moitié de mes cours au 2ème semestre de l’an dernier) de rattraper plus facilement.

Le nommage de la séance commence par une date au format YYYYMMDD où YYYY représente l’année sur 4 caractères, MM le mois sur 2 caractères et DD le jour sur 2 caractères

Par exemple 20230324 correspond à la séance du 24 mars 2023.

Ce nommage permet d’assurer que l’ordre alphabétique par défaut de tri des documents ou dossiers d’un répertoire corresponde toujours à l’ordre chronologique.

Ensuite après la date j’indique le contenu de la séance dans le nom du dossier, pour aider à s’y retrouver.

**Nommage des documents**

Les documents sont contextuels à une séance donnée, c’est-à-dire que le contenu n’a de sens que si on se situe dans la séance.

En effet, ils commencent généralement par DOC et un numéro d’ordre. On peut très bien avoir un même document remis lors d’une séance ultérieure mais son numéro d’ordre change.

|  |
| --- |
| Organisation des programmes |

## Introduction

Il est nécessaire de distinguer les programmes et exemples de code que je vous fournis pour les différentes séances et apprentissage et les programmes que vous allez utiliser pour vous

Afin que tout le monde puisse suivre, je redonne en début de chaque séance l’ensemble des programmes nécessaires pour pouvoir faire le TP et finaliser la séance.

En revanche, les programmes que vous reprenez sous forme de tutorial ou que vous effectuez doivent tous être sur GIT et régulièrement « poussés » sur Gitlab ou Github dans des projets GIT.

C’est à la fois fondamental pour votre compréhension et apprentissage mais également comme compétence professionnelle de base à part entière.

## Programmes pour cours et TP

Les programmes que je fournis en début de séance sont organisés dans des répertoires et parfois numérotés.

Ils ne servent que de point de départ.

Quand vous les reprenez, il est fondamental que vous les mettiez dans un projet GIT (cf 3.3) et que vous les commentiez

## Programmes pour Atelier de Professionnalisation, projets et développement

Tous les programmes doivent être placés sous Git[[1]](#footnote-1), qui est le logiciel de référence dans le monde du développement informatique.

Reportez vous à la documentation spécifique des commandes et outils Git.

Pour rappel, les étapes sont les suivantes :

* Initialisation d’un répertoire Git (git init)
* Ajout de fichiers (programmes ou documents) au suivi git (git add)
* Commit : crée une version du développement en cours avec les fichiers qui ont été ajoutés au suivi (git commit)
* Branche : crée une nouvelle branche de git, pour corriger un bug, développer tester une nouvelle fonctionnalité, un nouveau module (git branch)
* Création d’un projet (vide) sur Github ou Gitlab
* Copie du développement local vers le projet hébergé (git push après avoir défini le remote)

Il y a d’innombrables commandes ensuite qui permettent de récupérer un projet, le cloner, de chercher dans l’historique, de comparer des fichiers, de comparer des Commits, de passer d’une branche à l’autre, de merger des branches etc etc

Ce n’est pas simple d’utiliser Git, il y a un coût d’entrée, mais c’est indispensable et ensuite un outil de productivité extraordinaire.

|  |
| --- |
| Organisation des documents : spécifications, doc technique et document utilisateur |

## Spécifications

Les spécifications sont souvent données à des formats bureautique tel que Word, PowerPoint, PDF etc

Il s’agit de documents précisant les désiderata et les souhaits des métiers, c’est-à-dire des utilisateurs finaux. Ces souhaits sont traduits dans un langage compréhensible par des développeurs par des maitrises d’ouvrage, qui vont expliquer et détailler les règles de gestion aux développeurs, qui ne sont pas censés avoir des connaissances métier approfondies.

Ces documents peuvent également contenir des éléments de design lorsque le développement à faire comporte une ou plusieurs interfaces, soit des clients lourd (WinUI pour Windows, JavaFx pour Java, SDK Android pour Android …) soit des clients léger (interfaces Web).

## Documentation technique

### Commentaire dans le code

Ce 1er niveau de documentation technique est obligatoire.

C’est à la fois le plus utile, car indispensable pour vous relire ou pour partager le code avec d’autres développeurs, et le plus productif car grâce à des bibliothèques comme PHPDoc (<https://docs.phpdoc.org/3.0/guide/references/phpdoc/basic-syntax.html> ) ou JavaDoc, la documentation va être directement générée à partir du code, aux formats HTML, PDF …

### Format Ascii Doc et Markdown

Ce 2ème niveau de documentation est fortement recommandé. Plus chronophage, il est cependant productif et facile à maintenir car le balisage des documents est léger et la sortie accompagne le code, c’est-à-dire qu’elle peut être Gitté (ajoutée au projet et commité).

J’ai une préférence pour AsciiDoc (<https://docs.asciidoctor.org/asciidoc/latest/syntax-quick-reference/> ) par rapport à Markdown.

### Documentation d’exploitation

Lorsque un projet est mis en production, il s’agit ensuite de le maintenir et de documenter les étapes à effectuer pour publier un correctif ou une mise à jour. On parle d’intégration continue et de DevOps, c’est-à-dire un ensemble de tâches qui implique à la fois les équipes de développement applicatif et les équipes de maintenance système (en gros les SLAM côté Dev et les SISR côté Prod).

Cette documentation est préférable dans un format simple comme AsciiDoc car elle doit souvent évoluer lorsque l’application développée est en mode Agile.

### Documentation Word ou autre

Ces documentations sont optionnelles. Non pas qu’elles soient inutiles, bien au contraire, mais elles sont chronophages et difficiles à faire évoluer, c’est-à-dire peu évolutives.

Elles peuvent être très intéressantes pour apprendre, par exemple des tutoriels Youtube accompagné de PDF ou de pages Web HTML5 mais prennent beaucoup de temps à réaliser pour vous.

Donc je vous préconise de vous concentrer dans un premier temps sur les commentaires et l’utilisation d’AsciiDoc.

## Documentation utilisateur

La documentation utilisateur est essentielle au moment du déploiement. Il peut s’agir de tutoriel, de vidéo, d’aide en ligne, de documents Word et PDF. Lors d’une première version la documentation est multiple et détaillée, lors des mises à jour seules certaines nouveautés ou correctifs sont explicités.

Nous n’accorderons que très peu de place à la documentation utilisateur dans nos projets, pour nous concentrer sur les commentaires et la doc technique.

1. Cf l’histoire de Git qui est passionnante, liée à celle de Linux avec le même créateur, un des plus grands génies de l’informatique, Linus Torvalds [↑](#footnote-ref-1)