

[2017][DA2] Consignes

*Consignes pour le bilan documentaire de
juillet*

Description du document

Titre	[2017][DA2] Consignes
Date	15/06/2016
Auteur	Julien FREROT
Responsable	Flavien ASTRAUD
E-Mail	eip-tech@labeip.epitech.eu
Sujet	Consignes pour le bilan documentaire de juillet
Mots clés	
Version du modèle	1.0

Tableau des révisions

Date	Auteur	Section(s)	Commentaire
08/03/2012	Julien FREROT	Toutes	Première version
09/03/2012	Julien FREROT	5	Mise à jour lien vers le nouvel intra
19/02/2014	Julien FREROT	2, 3, 4	Mise à jour Tek 2015
17/02/2015	Julien FREROT	1, 2, 3	Mise à jour Tek 2016
03/04/2016	Julien FREORT	Toutes	Mise à jour Tek 2017 – refonte du rendu

Table des matières

1	Objectifs du bilan	1
1.1	Description	1
1.1.1	Documentation Utilisateur	1
1.1.2	Documentation technique	1
1.1.3	Liste des prérequis	1
1.1.4	Documentation d'installation	1
1.2	Glossaire	1
2	Formalités	2
3	Se préparer	2
3.1	Formalisme	2
3.2	Contenu du document	3
3.2.1	Résumé	3
3.2.2	Documentation utilisateur	3
3.2.3	Documentation technique	3
3.2.4	Liste des prérequis	4
3.2.5	Documentation d'installation	5
4	Barème	5
5	Remarques	5

1 Objectifs du bilan

1.1 Description

Ce bilan a pour but de vous faire écrire la deuxième version de vos documentations liées au projet. Les documents accompagnant votre projet sont les **documentations utilisateur** (manuel d'utilisation), **technique** (manuel du développeur), d'**installation** (procédure d'installation), et la **liste des prérequis** pour votre projet (matériels et logiciels). Un point clé à garder à l'esprit : ces documents, au même titre que votre code, **doivent faire l'objet d'une évolution constante et progressive** dans leur rédaction/mise à jour, et non uniquement avant un bilan.

Comme c'est une deuxième version, nous attendons que vous preniez en compte les commentaires de la correction de la première version, et que vous fassiez également preuve d'auto critique en cherchant à améliorer ces documents sur les axes qui n'ont pas été mentionnés dans la correction. La documentation doit également avoir évolué avec l'avancement de votre projet.

1.1.1 Documentation Utilisateur

Le but de la documentation utilisateur est d'avoir un guide pour que les utilisateurs de votre projet puissent s'en servir pleinement (le manuel de votre gaufrier pour vulgariser). La forme de ce document dépend du projet et du type d'utilisateur. Pour les EIP de type jeu, le petit livret contenu dans la boîte de jeu est une excellente illustration.

1.1.2 Documentation technique

Le but de la documentation technique est de maintenir un manuel technique de votre projet (encore pour vulgariser, la revue technique de votre voiture par exemple). Ce document doit décrire la conception du projet, son organisation, le moyen de contribuer, sa gestion, ses tests, et tous les éléments qu'un développeur qui rejoindrait le projet aurait besoin de connaître pour développer sur le projet. Le but est vraiment d'établir un guide du développeur qui permettrait à toute personne rejoignant l'équipe de contribuer au projet en comprenant son état, sa structure, et la façon de contribuer.

1.1.3 Liste des prérequis

Le but de la liste des prérequis est de bien comprendre quelles sont les configurations minimales dont votre projet a besoin (matériel et logiciel) pour fonctionner. Cela viendra appuyer des éventuelles demandes matérielles et logicielles que vous pouvez formuler, et permettre à l'équipe EIP de globaliser vos besoins.

1.1.4 Documentation d'installation

Le but de la documentation d'installation est de maintenir à jour la procédure pour installer votre projet (partie serveur et cliente). Ce document doit vous permettre en suivant les étapes pas-à-pas d'installer votre projet sur un environnement vierge. Ce document mis entre les mains d'un nouveau contributeur doit lui permettre d'effectuer son installation (pour tous les composants de la partie serveur et cliente), et le résultat doit être que votre projet soit fonctionnel et utilisable.

1.2 Glossaire

- F -

FAQ : Foire aux questions (ou « frequently asked questions » en anglais). C'est une liste de questions les plus communes qu'un utilisateur peut se poser et auxquels vous apportez une réponse.

- G -

Gestion de configuration : C'est le nom donné aux outils de « versionning » (du code ou documentaire). Par exemple SVN est un outil de gestion de configuration tout comme GIT, ...

- U -

UML : (Unified Modeling Language) c'est un langage de modélisation universel qui permet de représenter logiquement et physiquement des processus et produits d'ingénierie. <http://uml.free.fr> est une référence simpliste qui liste les types de diagrammes et leur grammaire.

2 Formalités

Les documents seront corrigés et **il y aura un rendez-vous associé** avec le professeur qui vous a corrigé une fois les corrections publiées afin de vous aider à itérer sur les prochaines versions de ces documents. Les documents UD, TD, ID devront être au format « PDF », la liste des prérequis dans un fichier texte UTF8. Ces documents devront être rendus en français pour **le Dimanche 10 juillet 2016 à 23h42**. Ils devront être rendus sur votre SVN dans rendu : <https://labeip.epitech.eu/svn/2017/<groupe>/rendu/>

Les documents suivant devront figurer :

- 2017_UD2_<Groupe>.pdf (documentation utilisateur)
- 2017_TD2_<Groupe>.pdf (documentation technique)
- 2017_ID2_<Groupe>.pdf (documentation d'installation)
- 2017_PR2_<Groupe>.txt (liste des prérequis matériels et logiciels)

3 Se préparer

Comme tous les documents de l'EIP, ces documents doivent présenter un document construit, élaboré, avec plusieurs parties obligatoires.

3.1 Formalisme

Tout d'abord, le document doit avoir un format spécifique. Il est obligatoire de voir figurer des en-têtes et pieds de pages, une page de garde qui indique le titre du document, le nom du groupe et éventuellement un visuel du groupe et de l'EIP/Epitech.

Ensuite, vous devez faire figurer sur la deuxième page un résumé du document (cf [Résumé](#)), optionnel pour la documentation utilisateur (à vous de juger de la pertinence).

Sur la troisième page, vous devez faire figurer un cartouche du document (propriétés, métadonnées du document), et un tableau des révisions, permettant de tracer l'historique de votre document à la façon d'un SVN (date, version, auteur, sections modifiées, commentaires).

Enfin sur la 4^{ème} page doit figurer un sommaire à jour, directement généré depuis le contenu de votre document, tenant sur 1 page si possible (en réduisant la profondeur utilisée par exemple).

Vos pages doivent être numérotées, la page numéro 1 correspondant à la première page de contenu (après le sommaire), les pages citées dans ce paragraphe ne faisant pas l'objet de numérotation.

3.2 Contenu du document

3.2.1 Résumé

Le résumé est un paragraphe (généralement de 200 à 300 mots, mais cela dépend de la taille du document) que le lecteur trouvera au début du document. Ce paragraphe sert à synthétiser les informations importantes du document, ainsi que les arguments principaux. On trouvera aussi les résultats et la conclusion du document.

Le résumé sert aux lecteurs qui souhaitent prendre rapidement connaissance du contenu de l'article :

- il pourra décider si l'article l'intéresse ou non,
- il pourra aller directement à la section qui l'intéresse.

Le résumé doit être bref et précis. Utilisez des phrases courtes et évitez les énumérations et graphiques.

Le résumé contient :

- le contexte de départ du document,
- les idées principales (les choix et arguments)
- la conclusion.

3.2.2 Documentation utilisateur

Vous devez fournir une première version de la documentation utilisateur. Celle-ci doit correspondre à un **manuel utilisateur**, elle doit permettre à un utilisateur de votre projet d'installer la partie cliente si c'est nécessaire et de prendre en main votre projet avec toutes ces fonctionnalités. Elle doit être claire et orientée non technique, tel qu'un manuel de n'importe quel appareil que vous utilisez chez vous. Pour les EIP jeu vidéo, il est fortement recommandé de faire une documentation utilisateur type **livret de jeu** qui est beaucoup plus adaptée.

Elle peut contenir par exemple les points suivants (mais non exhaustif et non applicable à tous les projets):

- Un schéma fonctionnel très haut niveau (commercial) de votre projet
- Installation / désinstallation des parties clientes (s'il y en a)
- Liste des fonctionnalités triées et organisées par thème. Il faudra appuyer avec des captures d'écran ainsi qu'une numérotation de chaque zone d'interaction pour la référencer et la décrire (cf manuel gaufrier et la liste des boutons/diodes)
- Pour les applications mobiles, une capture de chaque OS mobile est mieux car il y a des différences, mais ne répétez pas 3 fois la description fonctionnelle (à priori seule l'interface peut changer un peu).
- Une FAQ
- Lien vers le site vitrine / d'autres documentations

Ce document est destiné aux utilisateurs : il doit être clair et léger, agrémenté de **captures d'écran**.

3.2.3 Documentation technique

Ce document est un « **guide du développeur** » qui doit permettre à un membre du projet de comprendre les différentes parties, l'organisation de l'architecture du projet, les différents modules et comment

contribuer sur ces derniers. Elle doit permettre à un développeur de comprendre les règles et conventions utilisées pour le développement et les appliquer à des nouveaux modules en développement.

La documentation peut contenir les thèmes suivants (mais non exhaustif et non applicable à tous) :

- Vulgarisation de la vue du projet via un schéma de haut niveau expliquant le fonctionnement
- Vue logique de tous les composants/modules en place dans la solution (diagramme de composants)
- Utilisation du système de gestion de configuration, stratégie, et règles
- Découpage en grands blocs avec explication de l'implémentation des grands composants via des diagrammes de séquence, de composants, objets
- Description des API et interfaces (lien vers de la documentation générée + compléments d'explication mais pas de duplication d'information par rapport à la documentation générée)
- Norme du code
- Tests et conditions de passage du dev à une release
- Stratégie de release
- Release note

Si vous avez prévu une documentation du code générée via un outil ou autre, un lien dans la documentation technique vers lequel la documentation du code est disponible **suffit pour la description de ces API, et interfaces**. Cela ne **vous dédouane cependant pas des diagrammes et autres modélisations** ainsi que des explications. Dans ce cas, il vous suffit de faire référence à votre documentation de code, et si nécessaire, d'ajouter des éléments. Il serait mieux d'avoir une description détaillée dans la documentation du code et des exemples de « code snippet » ou autre.

Ce document est une architecture détaillée de ce que vous faite et un « guide du développeur ». Ce document doit contenir la liste des bugs techniques connus et un historique des « release note » ou « patch note » ou « change log » qui permet de retracer l'évolution des bugs et leur correction. Vous pouvez (et il est même conseillé de le faire) mettre cela en relation avec un système de gestion de tickets (bug tracking, assignement de tâches comme GIT, TFS ou autre)

3.2.4 Liste des prérequis

Ce fichier texte **UTF8** doit lister les prérequis matériels et logiciels. Ils doivent permettre de définir le matériel et logiciel nécessaire à votre configuration minimum pour pouvoir installer et utiliser votre projet sur les parties clientes et serveur. Veillez à bien séparer dans votre fichier les prérequis matériels des prérequis logiciels.

Les prérequis matériels peuvent contenir par exemple :

- Nombre de CPU par serveur
- Mémoire par serveur
- Taille des disques durs
- Architecture spécifique

Les prérequis logiciels peuvent contenir par exemple :

- Version du système d'exploitation
- Librairies et leurs versions
- Outils et Framework
- ...

3.2.5 Documentation d'installation

Cette documentation doit ressembler à une **procédure d'installation** étape par étape, et a pour but de permettre à n'importe qui **d'installer votre projet** (partie serveur et clientes), et de le **rendre fonctionnel** depuis un environnement vierge.

Ce document peut contenir par exemple :

- Liste des technologies utilisées et leur référence
- Configuration/Installation des environnements de production (partie serveur)
- Configuration/Installation des parties clientes
- Configuration/Installation des environnements de développement
- Procédure pas-à-pas d'installation
- Commandes à exécuter à copier/coller (exemple guide d'installation Gentoo)

4 Barème

Barème
Format du document
- Page de garde, cartouche, en-têtes / pieds de page, Mise en forme
- Grammaire / Orthographe FR
Contenu de la documentation utilisateur (UD)
Contenu de la documentation technique (TD)
Contenu de la procédure d'installation et la possibilité de l'exécuter (ID)
Liste des prérequis matériels et logiciels (PM, PL)
Prise en compte des remarques DA1

Tout retard de rendu entraînera un **impact sur l'attribution de crédits**.

5 Remarques

Pour nous contacter :

- Ticket <http://eip.epitech.eu/#/tickets>
- Sujet : [2017][Groupe][DA2] <Sujet>

Toutes les demandes par e-mail ou ne respectant pas ce formalisme de tickets **seront clôturées** sans réponse.