[2017][AA1] Consignes

Consignes pour le bilan architecture #1

[2017][AA1] Consignes

Description du document

Titre	[2016][AA1] Consignes
Date	15/07/2015
Auteur	Julien FREROT
Responsable	Flavien ASTRAUD
E-Mail	eip-tech@labeip.epitech.eu
Sujet	Consignes pour le bilan architecture d'août 2015
Mots clés	
Version du modèle	1.0

Tableau des révisions

Date	Auteur	Section(s)	Commentaire
15/07/2015	Julien FREROT	Toutes	Première version
01/06/2014	Julien FREROT		Mise à jour 2016
15/07/2015	Céline Baraban		Mise à jour 2017

Table des matières

1 Objectifs du document

Ce document a pour but de vérifier la bonne conception de l'architecture du projet. Malgré le peu d'avancement de votre projet, nous savons que ceci reste une conception, elle permet de poser les bases et n'est pas forcément définitive. Ce bilan a pour but de vous inciter à poser sur papier votre conception plutôt que de partir dans plusieurs POC et dans des heures de code jetable.

Ce document a aussi pour but de vous familiariser avec la conception sur papier, les diagrammes, la schématisation et le transfert de vos idées par papier. Nous vous proposons ici de commencer à approcher le cycle d'ingénierie logicielle consistant à exprimer votre conception sur papier, pour la critiquer et l'implémenter par la suite. Ce document est voué à évoluer itérativement dans les premières phases d'implémentation.

L'approche de ce document, est de partir des fonctionnalités de haut niveau (requirements), pour ensuite décrire une architecture logique (par composants fonctionnels, non liés à l'infrastructure directement), expliquer les processus complexes, pour arriver sur la représentation physique de votre architecture (infrastructure et découpage logiciel). Vous y indiquerez également les configurations. Ce document sert à gérer les risques techniques de votre projet, à modéliser les parties complexes qui pourraient poser problème dans l'implémentation. Il ne s'agit pas de faire du remplissage mais de bien expliquer les points sensibles, en gardant un niveau de détail lié à la complexité du composant décrit.

2 Formalités

- Le groupe doit avoir déposé au plus tard le ***dimanche 16 août 2015 à 23h42*** le document dans le répertoire de SVN dossier /rendu.
- Le document doit être nommé ***2017_AA1_FR_<Groupe>.pdf*** et rendu au format document portable (PDF), <Groupe> étant le nom de votre groupe.

3 Se préparer

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.

Sur la page suivant la page de garde, vous ferez figurer un résumé (cf résumé).

Ensuite vous devez faire figurer sur la page suivante 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 page suivante doit figurer un sommaire à jour, directement généré depuis le contenu de votre document, idéalement sur 1 seule page, avec les titres de chapitre et les numéros de page.

Vos pages doivent être numérotées, la première page correspondant à la partie 1, les pages citées dans ce paragraphe ne faisant pas l'objet de numérotation.

3.2 Contenu du document

3.2.1 Résumé

Le résumé du document n'est pas une introduction. Il sert à synthétiser les idées principales exprimées dans le document. Le résumé s'adresse à une personne qui ne lirait pas forcément le document mais qui veut savoir quel est son contenu et s'il est destinataire du document. Le résumé peut faire de quelques lignes jusqu'à 1 page, selon la nature et le contenu du document, sans contrainte forte sur ce point. Il doit permettre de comprendre tous les points clés du document.

3.2.2 Les points essentiels du contenu

Pour vous préparer vous devez écrire un ***document rédigé*** qui intègre :

- Un diagramme global de votre projet : cela peut permettre d'expliquer comment le projet est fait et comment il fonctionne. Vous privilégierez l'UML pour ces schémas. (Use case, Activité, …), mais pas uniquement vous ferez également un ou plusieurs schémas « utilisateurs ».
- Un diagramme détaillé de votre projet : Cela permet d'expliquer en détail son fonctionnement et ses modules, ses évolutions possibles. Vous privilégierez UML pour ces schémas (Use case détaillés, Activité, déploiement, classes)
- Un diagramme de communication : une vue simpliste des interconnexions entre les différentes parties internes et externes du projet. Mettre en évidence la distinction entre les composants existants (service ou bibliothèque du domaine public, données issues d'un ancien EIP, ...) des composants devant être développés ou intégrés par le groupe actuel. Vous privilégierez UML pour ces schémas (Déploiement, classes)
- Faire apparaitre l'ensemble de la logique métier de votre projet afin d'identifier tous les blocs logiciels, composants et autre (diagramme de séquence UML)
- L'architecture de vos classes, modélisation conceptuelle de votre projet. Vous privilégierez UML pour ces schémas (Déploiement, classes, séquence)
- Modèle conceptuel ou physique de la base de données
- Le choix des technologies, les justifications de ces choix, les avantages et inconvénients de ces choix (ceux qui ont présenté des matrices de comparaisons complètes au Vision peuvent les réutiliser et les compléter, pour ceux qui ont mal compris ce qui était attendu, c'est une occasion de recommencer cet exercice)
- Proposez un plan clair et structuré, la liste des points ci-dessus n'est pas forcément pertinente et ordonnée.

Note : Vous privilégierez l'utilisation d'un langage standard de modélisation : UML
Diagrammes globaux :
- Cas d'utilisation
- Activité
- Déploiement (Composants)
Diagrammes détaillés :
- Les même mais en plus détaillé
- Diagramme de communications
- Diagramme de classes
- Diagramme de séquence

4 Plan basique

Ce plan est basé sur la méthodologie RUP (développement avec cycle en V et fortement itératif), vous pouvez le reprendre, l'adapter ou proposer complètement autre chose

1. Introduction

1.1 Rappel de l'EIP

Vous présentez ici ce qu'est un EIP, l'école, la formation

1.2 Contexte et périmètre du projet

Vous présentez ici votre projet remis dans son domaine et ce qu'il fait. Attention ce n'est pas une avant-vente mais une précision fonctionnelle pour comprendre le contexte du document

1.3 Définitions, Acronymes et Abréviations

Vous mettrez ici une table des définitions et acronymes, anglicismes utilisés

1.4 Références

Vous mettrez ici une table des références vers d'autres documents du projet et/ou sources documentaires qui sont pertinentes dans votre document

2. Représentation de l'architecture globale

Vous présenterez dans cette partie l'approche du document, comment est faite l'actuelle architecture et les points de vue qui vont être présentés, et pour chaque point de vue, quels sont les éléments qui vont être présentés

3. Architecture, buts et contraintes

3.1 Objectifs spécifiques ayant un impact sur l'architecture

Vous présenterez dans cette section, les objectifs que vous avez pour votre projet qui ont un impact sur l'architecture

3.2 Contraintes fonctionnelles

Vous présenterez ici les contraintes fonctionnelles qui ont un impact sur l'architecture (A vous de chercher ce qu'est une contrainte fonctionnelle)

3.3 Contraintes non fonctionnelles

Vous présenterez ici les contraintes non fonctionnelles qui ont un impact sur l'architecture (A vous de chercher ce qu'est une contrainte non fonctionnelle)

4. Vue globale du projet

4.1 Use Cases principaux

Vous présenterez ici les scénarios / fonctionnalités par un diagramme UML de use case par exemple de haut niveau

4.2 Use Cases détaillés

Vous présenterez les UC détaillés dans ce paragraphe de façon succincte pour rappeler les fonctionnalités de votre projet

5. Vue Logique de l'application

5.1 Vue globale

Vous présenterez ici le design global de la solution, les composants, découpages en paquets, modules. Vous pourrez utiliser un diagramme de composants et/ou de déploiement UML

5.2 Composants principaux

Vous décrirez ici les composants principaux utilisés, s'ils sont existants, créés, …

6. Vue Processus

Vous décrirez ici la décomposition en petites tâches de votre système pour vous permettre de valider les processus et les acteurs mis en jeu dans les différentes fonctionnalités. Vous utiliserez par exemple des diagrammes de Séquence UML et Activité

7. Vue Déploiement

Vous présenterez ici la vue physique de votre application (Serveurs, liens réseau, protocoles, communication, et décrirez les éléments physiques en jeu lors des différents processus (présentés dans le chapitre précédent). Vous pourrez utiliser des diagrammes de déploiement UML. Vous préciserez aussi la partie gestion de la configuration qui permet de connaitre les contraintes de configuration de votre environnement cible.

8. Implémentation

8.1 Vue globale

Vous présenterez ici votre architecture logicielle à proprement parler, de façon plus détaillée, modèle en couche, N-Tiers, MVC, MVVM, MVP, … Vous préciserez les limites de chaque système pour identifier les composants de chaque module (diagramme de package, diagramme de composants UML)

8.2 Couches applicatives

Vous présenterez ici une vue plus détaillée avec des diagrammes objets, diagramme de classe par couche / composant

9. Vue données

Vous présenterez, si vous en avez, une vue conceptuelle de votre modèle de données. Les captures d'écran de bases déjà implémentées sont à proscrire.

10. Taille et Performance

Vous présenterez les ordres de grandeurs de la dimension de votre application dans cette section (nombre d'utilisateurs, de requête, taille du stockage de données, taille des requêtes, temps de réponse, localisation, …)

11. Qualité

Une description de la façon dont l'architecture logicielle contribue à toutes les exigences (autres que fonctionnelles) du système: l'extensibilité, la fiabilité, la portabilité, et ainsi de suite. Si ces caractéristiques ont une importance particulière, pour les implications de sécurité ou de confidentialité, par exemple, ils devraient être clairement définis.

5 Barème

Question
Format du document - Grammaire Orthographe, mise en forme (en-têtes, pieds de page, styles) - Rappel du fonctionnel de l'application - Glossaire, Annexes
Contenu supporté par des diagrammes normalisés - Globaux / logiques de l'application - Processus / flow - Physiques de l'application - Modélisation conceptuelle de l'application (logicielle et données)
Description de la taille, performance, qualités

6 Remarques

- Pour nous contacter : Ticket http://eip.epitech.eu/tickets : [EIP][2017][AA1] <Sujet>, tout ticket ne portant pas cette mention en titre sera ignoré
- Aucun retard de rendu ne sera toléré.