

[2015][AA1] Consignes

*Consignes pour le bilan architecture de juin
2013*

Description du document

Titre	[2015][AA1] Consignes
Date	17/05/2013
Auteur	Julien FREROT
Responsable	Flavien ASTRAUD
E-Mail	eip-tech@labeip.epitech.eu
Sujet	Consignes pour le bilan architecture de juin 2013
Mots clés	
Version du modèle	1.0

Tableau des révisions

Date	Auteur	Section(s)	Commentaire
17/05/2013	Julien FREROT	Toutes	Première version

Table des matières

1	Objectifs du bilan.....	1
2	Formalités	1
3	Se préparer	1
4	Plan basique.....	2
5	Barème.....	3
6	Remarques.....	4

1 Objectifs du bilan

Ce bilan 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 bilan 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.

2 Formalités

- Le groupe doit avoir déposé au plus tard le **Vendredi 21 juin 2013 à 23h42** le document dans le répertoire de SVN dossier /rendu.
- Le document doit être nommé **2015_AA1_FR_<Groupe>.pdf** et rendu au format document portable (PDF), <Groupe> étant le nom de votre groupe.

3 Se préparer

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 privilégiez 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 privilégiez 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 privilégiez UML pour ces schémas (Déploiement, classes)
- Faire apparaître 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 privilégiez 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.
- Proposez un plan clair et structuré, la liste des points ci-dessus n'est pas forcément pertinente et ordonnée.

Note : Vous privilégiez 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 séquence
- 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 mettez ici une table des définitions et acronymes, anglicismes utilisés

1.4 Références

Vous mettez 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

4.2 Use Cases détaillés

Vous présenterez les UC détaillés dans ce paragraphe

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 connaître 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 systèmes 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, de sécurité ou de confidentialité, par exemple, ils devraient être clairement définis.

5 Barème

Question	/20
Format du document	X / 3
<ul style="list-style-type: none"> - Grammaire Orthographe - Mise en forme (en-têtes, pieds de page, styles) - Rappel du fonctionnel de l'application - Glossaire, Annexes 	
Choix des technologies (avantages / inconvénients, limites, comparatifs, ...)	X / 2

Question	/20
Diagrammes	X / 13
<ul style="list-style-type: none"> - 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	X / 2
Malus	X / -20

6 Remarques

- Pour nous contacter : Ticket <http://eip.epitech.eu/tickets> : [EIP][2015][AA1] <Sujet>, tout ticket ne portant pas cette mention en titre sera ignoré
- Aucun retard de rendu ne sera toléré. Tout retard sur ce document entrainera un échec au module pour ce semestre.