

# Guide d'aide à la conception du dossier à rendre : Vos projets communs

---



## Table des matières

Objectifs du projet commun .....	3
 <b>1. Conseils de rédaction.....</b>	<b>4</b>
L'orthographe et la grammaire .....	4
La relecture.....	4
Le plagiat.....	4
Le style .....	4
Les figures.....	4
Les références – La bibliographie .....	5
La page de garde et sommaire .....	5
 <b>2. Partie Analyse (environ 12 pages).....</b>	<b>6</b>
Analyse du terrain.....	6
Conduite de projet.....	6
La qualité attendue.....	7
Le système : vue fonctionnelle .....	7
Le système : vue statique .....	7
Le système : vue dynamique .....	7
Vos choix d'implémentation.....	7
Justifications des choix techniques.....	7
 <b>3. Partie architecture (environ 5 pages) .....</b>	<b>8</b>
Schéma .....	8
Questionnaire .....	8
 <b>4. L'évaluation croisé du code.....</b>	<b>9</b>
 <b>5. Partie bilan (environ 10 pages) .....</b>	<b>10</b>
Retour sur le travail réalisé.....	11
Retour réflexif sur l'action .....	11

## Objectifs du projet commun

Le projet commun est une possibilité qui vous est offerte afin de réutiliser les enseignements et de développer vos compétences naissantes au sein d'un projet que vous vous serez approprié : conduite de projet, architecture, intégration des utilisateurs, sens de l'organisation et co-construction...montrez-nous ce que vous savez faire.

Le document attendu est un travail de groupe soigné aussi bien sur le fond que sur la forme entre 30 et 50 pages.

Dans ce document vous y trouverez des suggestions et des éléments de guidage...mais il ne remplacera ni votre sens de l'analyse, ni vos camarades, ni vos enseignants.

Votre projet va se construire durant et autour de 4 étapes principales :

- 1 Partie analyse
  - 2 Partie architecture
  - 3 l'évaluation du code
  - 3 Partie bilan
- Nous allons les détailler.



## 1. Conseils de rédaction

### L'orthographe et la grammaire

L'orthographe et la grammaire, tout comme le choix de vos tenues vestimentaires, sont le reflet de votre crédibilité et du professionnalisme de votre posture et votre travail. Quel que soit le document que vous rédigez (lettre de motivation, lettre à votre employeur, propriétaire, cahier des charges, documentation technique, rapport de stage etc.), vous devez absolument le relire et corriger les fautes d'orthographe et de grammaire.

Un document comportant des fautes est inacceptable, donne une image très négative de celui qui l'a rédigé et peut jouer un rôle déterminant notamment dans votre processus de recherche d'emploi.

Tout traitement de texte dispose d'un correcteur d'orthographe. Il existe d'autres outils plus puissants, mais payants ([http://www.druide.com/a\\_description.html](http://www.druide.com/a_description.html))

### La relecture

La relecture participe au repérage et la correction : des fautes, coquilles, des erreurs de syntaxes, de la mise en page... Faire relire votre document, par une tierce personne vous donnera des éclairages nouveaux sur votre travail.

Pour les synonymes : <http://www.cnrtl.fr/definition/>

Pour la conjugaison : <http://www.leconjugueur.com/>

### Le plagiat

En parcourant les sites internet à la recherche d'informations utiles pour concevoir votre dossier, vous tomberez parfois sur une phrase ou un paragraphe (comme une définition par exemple) que vous souhaitez intégrer à votre travail : n'oubliez pas d'indiquer clairement l'auteur de la phrase en utilisant les guillemets. Comme ceci par exemple : « *La théorie sans la pratique est inutile, la pratique sans la théorie est aveugle* ». Kant. Vous pouvez aussi expliquer la même idée avec vos propres mots, dans ce cas vous omettez les guillemets et vous donnez la référence de la source dont vous vous êtes inspirés.

Copier-coller des paragraphes entiers d'un autre document sans donner aucune source et en laissant croire qu'il vient de vous s'appelle du plagiat et peut être source de lourdes sanctions.

### Le style

Vous êtes maintenant en L3 et on vous demande de faire un réel effort sur la forme. Tout comme cet effort vous sera demandé durant toute votre professionnelle : il est essentiel de différencier le style « oral » du style « écrit ». Vous ne pouvez pas rédiger un document de la même manière dont vous vous exprimez à l'oral. D'autre part, n'utilisez que des mots que vous comprenez ou dont vous avez vérifié le sens !

### Les figures

Lorsque vous choisissez d'utiliser une figure dans un document, il faut référencer sa source. D'autre part, il faut décrire, expliquer la figure voire la critiquer. Une figure sans explication ni légende n'a aucune utilité.

## Les références – La bibliographie

Lorsque l'on utilise un document comme source pour rédiger notre propre document il est IMPERATIF de référencer cette source à l'endroit précis où on la cite. Par exemple ainsi :

« En ingénierie, une méthode d'analyse et de conception est un procédé qui a pour objectif de permettre de formaliser les étapes préliminaires du développement d'un système afin de rendre ce développement plus fidèle aux besoins du client » (Wikipédia, 2011).

En dernière page de votre dossier veillez à référencer précisément toutes les sources citées dans votre document dans le cas où le lecteur souhaiterait lire une source (rubrique « Références » ou « Bibliographie »).

Par exemple : Wikipédia, Méthodes d'analyse et de conception, [http://fr.wikipedia.org/wiki/M%C3%A9thodes\\_d'analyse\\_et\\_de\\_conception](http://fr.wikipedia.org/wiki/M%C3%A9thodes_d'analyse_et_de_conception), consulté en mai 2011.

Pour les références bibliographiques : [http://www.chicagomanualofstyle.org/tools\\_citationguide.html](http://www.chicagomanualofstyle.org/tools_citationguide.html)

## La page de garde et le sommaire

Soyez créatif et cohérent ! Doivent absolument être clairement visible : le nom du projet, les noms et prénoms des membres du groupes, votre année, votre formation et votre année universitaire. Un sommaire automatique et la numérotation des pages sont fortement recommandés. Vous êtes également invité à écrire un bref résumé (environ 15 lignes) de votre projet dans les premières pages de votre dossier.

## 2. Partie Analyse (environ 12 pages)

Comme son nom l'indique, avant de partir la tête baissée dans le codage d'une application, d'un site internet ou même juste d'une interface, il faut vous poser les bonnes questions, réfléchir, objectiver, choisir un fil rouge... Nous souhaitons voir comment vous analysez, qu'est-ce que vous analysez et pour quoi faire ? A quoi cela va vous servir dans l'élaboration de votre outil informatique ?

Pour vous y aider voici un plan suggéré :

1. Analyse du terrain
2. La conduite de projet
3. La qualité attendue
4. Le système : vue fonctionnelle
5. Le système : vue statique
6. Le système : vue dynamique
7. Vos choix d'implémentation

### Analyse du terrain

Avant l'existence de votre outil comment faisait vos acteurs ? Qu'elle était la réalité des choses ?

(Par exemple : si vous souhaitez créer une application pour aider les utilisateurs d'aquarium à entretenir leur aquarium, comment ces personnes font – elles actuellement, sans votre application, à entretenir leur aquarium ?) C'est en vous appuyant sur l'existant que vous démontrerez la valeur ajoutée de votre création.

Pour vous y aider voici quelques pistes à explorer :

- quel est le public concerné ?
- quel est son enjeu ?
- quelles difficultés rencontrent-ils ?
- existe-t-il des alternatives à leurs difficultés ?
- à quelle problématique répond votre création ?
- comment pouvez-vous la tester ?

Cette phase est déterminante pour la suite : elle sert à justifier solidement la création de votre outil. Pour vous y aider tous les moyens sont acceptés : articles scientifiques, étude de terrain, photos, témoignages etc. C'est à vous de construire la justification de l'outil que vous allez concevoir.

### Conduite de projet

Vous présenterez ici comment vous avez mené votre projet : gestion du temps, planification, répartition des tâches, vos phases de conception, implication des utilisateurs, l'utilisation de plateformes collaboratives ou logiciel de gestion de version. L'implication et rôle de votre commanditaire (ou client) si vous en avez un, ainsi que de vos utilisateurs au sein de votre projet (ex : si vous faites une application pour aquarium vos acteurs seront les aquariophiles). Voici quelques questions pour vous aiguiller durant votre projet :

- Où en suis-je ? Où en sommes-nous ?
- Qu'est-il opportun de faire ? Pour qui ?
- Sur qui puis-je m'appuyer ?
- Finalement, quel est le problème principal ?

Si vous travaillez selon une méthode Agile comme Scrum, il faudra expliquer les bonnes pratiques adoptées et votre « implémentation » de Scrum dans le cadre de votre projet (M1).

### **La qualité attendue**

Vous présenterez les qualités attendues du futur système en termes d'implémentation, maintenance, évolutivité, interopérabilité etc. (INF2 et INF6 en L3, INF1 et INF3 en M1).

### **Le système : vue fonctionnelle**

Vous décrierez ici les futures fonctionnalités du système sous forme de diagramme(s) de cas d'utilisation et de scénarios en langue naturelle respectant le patron donné en L3 ISI1 (scénario nominal, alternatifs, d'erreur). Chaque diagramme comporte un titre et une brève description textuelle introductive expliquant ce que spécifie le schéma.

### **Le système : vue statique**

Un ou des diagrammes de classes seront présentés pour montrer l'information manipulée par le système. En cas d'utilisation d'une architecture MVC, vous distinguerez les classes modèles, contrôleurs et vues. Différents diagrammes de classes pourront être utilisés afin d'en faciliter la lecture et la compréhension. Des diagrammes de paquetages pourront montrer les dépendances entre les diagrammes de classes. Les classes comprendront les attributs ainsi que les méthodes utilisées dans les diagrammes de séquences. Un ou des diagrammes de composants pourront présenter les composants, leurs interfaces et les communications entre les composants si un système à base de composants est envisagé.

### **Le système : vue dynamique**

Vous présenterez les diagrammes de séquences correspondant aux cas d'utilisation définis (1 diagramme de séquence par cas d'utilisation). Vous pourrez également présenter des diagrammes de séquences systèmes. Les objets, méthodes, associations définis dans les diagrammes de séquences devront être cohérents avec les diagrammes de classes présentés.

### **Vos choix d'implémentation**

Vous présenterez et justifierez les langages de programmation choisis (pourquoi, avantages, inconvénients, comment vous allez contourner les difficultés etc.), la façon de gérer vos données (base de données relationnelle, objet, etc.). Vous présenterez en annexe le schéma de votre base de données relationnelle en 3ème forme normale (E/R ou schéma relationnel) le cas échéant. Vous présenterez une première version de votre architecture sous la forme d'un diagramme de déploiement UML. Vous expliquerez comment l'architecture choisie (le modèle) permettra d'assurer les critères de qualité énoncés dans la section « Qualité attendue ».

### **Justifications des choix techniques**

Vous montrerez vos choix techniques avec leurs justifications (les schémas UML, l'architecture, implémentation, IHM, base de données, langage).

### 3. Partie architecture (environ 5 pages)

#### Schéma

Dans cette partie nous vous demandons dans un premier temps de donner le schéma de votre architecture de la manière la plus complète possible. Vous veillerez à modéliser tous les éléments de votre architecture, de les nommer de manière complète et de définir les noms des liens entre les différents éléments. Un diagramme de classes orienté MVC de votre architecture doit être inclus dans votre dossier.

Si vous utilisez un ou plusieurs frameworks pour le développement, le diagramme d'architecture devra expliquer les composants que vous devez développer, ceux que vous récupérez directement du framework et les relations entre les composants.

L'utilisation d'un diagramme de déploiement UML est fortement recommandée pour les M1 (voir la norme UML <http://www.omg.org/spec/UML/2.4.1/>).

#### Questionnaire

Il s'agit de répondre au questionnaire ci-dessous et de justifier vos réponses par rapport à votre architecture. Votre architecture devra être conçue de manière à répondre à tous les critères sous-jacents aux questions.

N°	Question	Réponse
1	Votre architecture vous permettrait-elle de supporter facilement une autre langue utilisateur (français et anglais par exemple) ?	Oui <input type="checkbox"/> Non <input type="checkbox"/> Ne s'applique pas <input type="checkbox"/> Justifiez votre réponse.
2 (web)	Evaluez la difficulté du passage de votre application web à du mobile.	Très facile <input type="checkbox"/> Facile <input type="checkbox"/> Pas facile <input type="checkbox"/> Pas facile du tout <input type="checkbox"/> Justifiez votre réponse.
3	La modularité est un critère important de qualité des architectures (Un composant ou classe n'a qu'un seul objectif). Les composants de votre architecture respectent-ils ce principe de modularité ?	Oui <input type="checkbox"/> Non <input type="checkbox"/> Ne s'applique pas <input type="checkbox"/> Justifiez votre réponse.
4	Des éléments de votre architecture seraient-ils réutilisables dans un autre projet ?	Oui <input type="checkbox"/> Non <input type="checkbox"/> Ne s'applique pas <input type="checkbox"/> Justifiez votre réponse.
5	Votre architecture permettrait-elle de changer de SGBD facilement ?	Oui <input type="checkbox"/> Non <input type="checkbox"/> Ne s'applique pas <input type="checkbox"/> Justifiez votre réponse.
6	Si vous deviez changer totalement la réalisation de l'interface graphique, y-aurait-il des composants ou classes inchangé(e)s ?	Oui <input type="checkbox"/> Non <input type="checkbox"/> Ne s'applique pas <input type="checkbox"/> Justifiez votre réponse.
7	Quels seraient les effets d'un éventuel changement dans le schéma de la base de données (ajout d'une nouvelle table ou nouvel attribut, changement du nom d'une table ou d'un attribut, changement sur un type d'attribut, sur une contrainte...) dans l'application ?	Détaillez votre réponse.



## 4. L'évaluation croisé du code

Un document imprimé par groupe, à rendre le jour de la soutenance.

Dans ce document, nous vous demandons d'évaluer le code de l'application réalisée par un autre groupe d'étudiants de votre promotion. Il s'agit de répondre à chaque question en justifiant le mieux possible vos réponses et de manière complète. Vous justifierez toutes vos réponses de manière précise, vous pouvez faire du copier-coller du code si nécessaire Vous donnerez une note finale au groupe, et vous-même serez notés en tant qu'évaluateurs.

### Soutenance

Vous assisterez à la soutenance du groupe évalué et poserez des questions par rapport à l'évaluation réalisée.

<b>Groupe évalué :</b>		
<b>Groupe évaluateur :</b>		
<b>Nom du projet :</b>		
<b>N°</b>	<b>Question</b>	<b>Réponse</b>
<b>Bonnes pratiques d'architecture</b>		
1.	L'application permet-elle de supporter facilement une autre langue utilisateur (français et anglais par exemple) ?	Oui <input type="checkbox"/> Non <input type="checkbox"/> Ne s'applique pas <input type="checkbox"/> Justifiez votre réponse.
2.	Evaluez la difficulté du passage de l'application web à du mobile.	Très facile <input type="checkbox"/> Facile <input type="checkbox"/> Pas facile <input type="checkbox"/> Pas facile du tout <input type="checkbox"/> Justifiez votre réponse.
3.	La modularité est un critère important de qualité des architectures (Un composant ou classe n'a qu'un seul objectif). Les composants de l'application évaluée respectent-ils ce principe de modularité ?	Oui <input type="checkbox"/> Non <input type="checkbox"/> Ne s'applique pas <input type="checkbox"/> Justifiez votre réponse.
4.	Des éléments de l'application seraient-ils réutilisables dans un autre projet ?	Oui <input type="checkbox"/> Non <input type="checkbox"/> Ne s'applique pas <input type="checkbox"/> Justifiez votre réponse.
5.	L'architecture de l'application permet-elle de changer de SGBD facilement ?	Oui <input type="checkbox"/> Non <input type="checkbox"/> Ne s'applique pas <input type="checkbox"/> Justifiez votre réponse.
6.	Si l'on devait changer totalement la réalisation de l'interface graphique, y-aurait-il des composants ou classes inchangé(e)s ?	Oui <input type="checkbox"/> Non <input type="checkbox"/> Ne s'applique pas <input type="checkbox"/> Justifiez votre réponse.
7.	Quels seraient les effets d'un éventuel changement dans le schéma de la base de données (ajout d'une nouvelle table ou nouvel attribut, changement du nom de la table table ou d'un attribut, changement sur un type d'attribut, sur une contrainte...) dans l'application ?	Détaillez votre réponse.
<b>Bonnes pratiques de programmation</b>		
8.	Le code :	Compile <input type="checkbox"/> S'exécute <input type="checkbox"/>
9.	Le code comporte-t-il des tests unitaires ? (JUnit pour Java, PHPUnit pour PHP etc.)	Oui <input type="checkbox"/> Non <input type="checkbox"/> Ne s'applique pas <input type="checkbox"/> Justifiez votre réponse.
10.	Une norme de codage est-elle utilisée ? (PSR pour PHP, etc.)	Oui <input type="checkbox"/> Non <input type="checkbox"/> Ne s'applique pas <input type="checkbox"/>

		Justifiez votre réponse.
11.	Des API ont-elles été correctement utilisées afin d'optimiser le développement ?	Oui <input type="checkbox"/> Non <input type="checkbox"/> Ne s'applique pas <input type="checkbox"/> Justifiez votre réponse.
12.	En utilisant l'outil SonarQube ( <a href="http://www.sonarqube.org/">http://www.sonarqube.org/</a> ) donnez les métriques suivantes et évaluez les résultats.  Ajouter une capture d'écran des résultats donnés par l'outil.	<ul style="list-style-type: none"> <li>- Nombre de lignes de code</li> <li>- Documentation</li> <li>- Duplications</li> <li>- Complexité</li> <li>- Problèmes</li> <li>- Couverture tests unitaires</li> <li>- Complétez avec les indicateurs pour détecter les « 7 péchés capitaux du développeur »<sup>1</sup></li> </ul>
13.	Robustesse du code : des exceptions sont-elles correctement spécifiées pour éviter une mauvaise utilisation des méthodes ?	Partout <input type="checkbox"/> A quelques endroits <input type="checkbox"/> Pas du tout <input type="checkbox"/> Ne s'applique pas <input type="checkbox"/> Justifiez votre réponse.
14.	Le code rendu remplit-il les qualités énoncées dans le dossier d'analyse (modularité, réutilisation, usage MVC ou architecture 3-tiers, patrons de conception (design patterns)...) ?	Oui <input type="checkbox"/> Non <input type="checkbox"/> Justifiez votre réponse.
<b>Bonnes pratiques de documentation</b>		
15.	Le code est-il commenté de manière complète et cohérente et utile ?	Oui <input type="checkbox"/> Non <input type="checkbox"/> Justifiez votre réponse.
16.	Y-a-t-il une java/PHP/... doc ?	Oui <input type="checkbox"/> Non <input type="checkbox"/> Ne s'applique pas <input type="checkbox"/> Justifiez votre réponse.
17.	La documentation technique fournie est-elle compréhensible, utile et complète ?	Oui <input type="checkbox"/> Non <input type="checkbox"/> Justifiez votre réponse.
18.	La documentation technique et le code fournis sont-ils cohérents ?	Oui <input type="checkbox"/> Non <input type="checkbox"/> Justifiez votre réponse.
19.	Les modèles UML fournis dans la documentation correspondent-ils au code fourni ?	Oui <input type="checkbox"/> Non <input type="checkbox"/> Justifiez votre réponse.
20.	Les modèles UML fournis permettent-ils de bien comprendre l'application ?	Oui <input type="checkbox"/> Non <input type="checkbox"/> Justifiez votre réponse.
21.	Note globale	.... / 20 Justifiez votre réponse.

<sup>1</sup> <http://docs.codehaus.org/display/SONAR/Developers%27+Seven+Deadly+Sins>

## 5. Partie Bilan (environ 10 pages)

### Retour sur le travail réalisé

Vous soulèverez les limites de votre réalisation ainsi que votre démarche en faisant preuve d'autocritique. C'est également l'occasion de présenter les axes d'améliorations envisagés en les justifiant.

### Retour réflexif sur l'action

*« La pratique réflexive est la capacité à réfléchir à l'action de sorte à s'engager dans un processus d'apprentissage continu ». Schön (1987)*

Ce retour peut être individuel. En vous appuyant sur un élément de votre vécu (votre expérience, un évènement, vos doutes, une situation particulière...) vous choisirez un point précis qui vous intéresse et qui vous a interpellé durant votre projet commun. Vous donnerez un nom à l'objet de votre réflexion.

Quelques exemples : ma méthode est-elle une bonne méthode ? Comment cerner le problème ? Qui suis-je, étudiant ou professionnel ? etc.

Vous développerez à votre convenance cette réflexion en le décrivant le plus précisément possible en nous expliquant pour quelles raisons ce point vous amène à réfléchir.

Voici quelques questions pour vous aiguiller.

- pourquoi ceci a marché ? A échoué ?
- qu'est ce qui m'a donné à penser ça ?
- comment aurais-je pu faire différemment ?
- quelles limites dans les outils proposés ?
- qu'est-ce qui m'a posé problème ?
- ai-je déjà vécue une situation similaire ?
- qu'est-ce que j'ai appris ?

/ ! à éviter : aucune justification, sujet trop vaste, descriptions générales, les avis non constructifs

Cette partie est également l'occasion pour vous d'être une force de proposition en vue d'améliorer l'expérience des projets communs pour les années à venir.

*Vous n'oublierez pas d'ajouter un moyen pour que votre jury puisse accéder en toute autonomie à votre code et à l'outil que vous avez développé.*

*Bon travail !*