

**Plan type d'un rapport de stage ou de Travail Pratique Transversal : TPT
(pas plus de 40 pages hors annexes)**

Vous trouverez ci-dessous le plan type du TPT ou du stage en entreprise, pour un projet de type développement. Ce plan doit être suivi. Les adaptations peuvent être apportées si cela se justifie par la nature de votre projet (projet autre que du développement). Vous devez valider ces adaptations avec votre responsable pédagogique Université.

Page de garde

«Master 2 MBDS et/ou Master2 MIAGE MBDS de l'Université de Nice Sophia Antipolis (UNS) (logos MBDS et UNS, logo école partenaire si délocalisé, logo de l'entreprise d'accueil s'il y a)

« Stage en entreprise »

Titre du stage

Dates de début et de fin du stage

Nom de l'étudiant

Nom du tuteur à l'Université

Nom du tuteur en entreprise (si pas TPT)

Membres du jury

Résumé (en français, une demi-page)

Un résumé doit présenter :

1. le but et la nature du travail
2. l'envergure du travail
3. les méthodes utilisées
4. les principaux résultats
5. les conclusions
6. des mots clés

Abstract (en anglais)

Liste des figures

Liste des acronymes

Plan du document

Il doit être généré automatiquement

1. Introduction générale (1 page)

Accroche avec un fait marquant, une stat, un slogan pub (bonus pour attirer l'attention dès le départ)

- > Votre plan de formation ou de carrière (si vous le connaissez déjà)
- > Les raisons qui vous ont poussé à choisir ce stage et cette entreprise / secteur pro
- > Vos objectifs /attentes vis à vis de ce stage
- > Une présentation très rapide de l'entreprise et son secteur
- > Un bref descriptif de vos missions pendant le stage
- > La problématique de votre rapport

> L'annonce du plan de votre rapport de stage (présenter le contenu de chaque chapitre sur une ligne)

2. Présentation du stage

2.1 Présentation de l'entreprise

Présentation sommaire de l'entreprise

2.2 Présentation du sujet et objectifs du projet (problème traité)

Justification du caractère innovant du sujet

Présentation générale du contexte métier (si applicable)

Mise en avant de la criticité éventuelle du projet, des principaux enjeux et principaux risques.

3. Etat de l'art sur le sujet traité

Publications sur le sujet traité (renvois à des références de la bibliographie)

Produits et outils existants traitant du sujet

4. Etude de l'existant et solution envisagée

4.1 Etude de l'existant

4.1.1 Description externe du système logiciel existant (vision utilisateur)

Faire une description haut niveau des exigences déjà réalisées dans l'existant

4.1.2 Description interne du système logiciel existant (vision développeur/conception)

Faire une description haut niveau de la plate-forme technique existante et du logiciel déjà développé (les packages de code source ainsi que les tables existantes, les composants et leur déploiement)

4.2. Critique de l'existant

La critique doit être positive et négative

Elle peut se faire dans le chapitre 4.1 "Etude de l'existant"

4.3. Solutions envisagées

Décrire les solutions envisagées et justifier la solution choisie

4.4 Objectifs principaux et livrables

5. Démarche projet

(Attention !!! la démarche et les outils associés doivent être mis en place dès le début du stage)

5.1 Principes de la démarche projet

5.1.1 Activités d'ingénierie logicielle

Indiquer de façon synthétique (1/2 page max) les activités d'ingénierie logicielle (exigences, conception, codage, différents niveaux de test ...) applicables au projet.

Le cas échéant, préciser la méthode de référence (Unified Process, XP ...). En particulier, préciser la stratégie de test.

Parmi les activités d'ingénierie, préciser celles que vous réalisez.

5.1.2 Méthode de gestion de projet utilisée

Méthode « en cascade » ou « itérative ». Le cas échéant, indiquer la méthode de référence, (Scrum, UP, ...), sans la décrire en détails. Précisez votre rôle vis-à-vis de la gestion de projet.

Présenter la démarche de communication: réunions, compte rendus ...

5.1.3 Rôles et responsabilités

Décrire les parties prenantes (client, équipe marketing ...) et l'équipe projet.
Préciser en particulier votre rôle et vos relations avec les autres acteurs.

5.1.4 Outils

Décrire les outils choisis pour mettre en œuvre la démarche (IDE, outil de gestion de configuration, modelleur UML ...). Préciser votre rôle dans le choix des outils.

5.1.5 Gestion de la configuration

Pour la gestion de configuration, expliquer l'organisation des livrables du projet ainsi que les règles de nommage des fichiers (sauf pour les livrables d'implémentation, qui sont présentés dans le chapitre 7).

Expliquer l'organisation des sauvegardes/versions : des sources et des documents du projet etc.

5.2 Contraintes et risques sur le projet

Présenter les contraintes et les risques du projet.

Les risques peuvent être présentés dans un tableau. Par exemple:

N° du risque	Libellé du risque	Priorité	Importance	Facteur Contribuant	Solution Proposée	STATUS (en cours, réalisé : solution choisie, ...)
1	Ne pas pouvoir réaliser le prototype avec le produit Prévu			1.1 La date de livraison du produit Jasmine ii v2 est décalée	1.1 Utiliser la version 1 de Jasmine	en cours
				1.2 ...	1.2	
				1.3 ...	1.3	
2	Ne pas pouvoir réaliser l'exigence fonctionnelle X de cryptage			1.2 Manque de connaissance sur la cryptographie	1.1 autoformation pendant 20 jours	Réalisé : 1.2
					1.2 Acheter une formation chez CISCO Paris 2500 Euros pour 4 jours)	Vu l'urgence et la sensibilité
					Prix catalogue	
					1.3 Embaucher un stagiaire d'une école de sécurité	
					(au plus tard le 15 Avril)	

Un plan de risque doit être exécutable. Les risques, les facteurs contributions et les solutions doivent être précis.

Etre Malade n'est pas risque. C'est facteur contribuant.

Un risque c'est de ne pas pouvoir atteindre un de vos objectifs (une ou plusieurs de vos exigences fonctionnelles ou non fonctionnelles, ...).

Une solution proposée peut consister à faire quelque chose dans une des premières itérations pour lever le risque.

5.3 Démarche projet mise en œuvre

Il s'agit de présenter ici le découpage du projet en étapes.

Dans le cas d'une méthode de gestion de projet itérative (Scrum, par exemple), présenter le découpage en sprints/itérations.

Pour chaque sprint/itération, présenter son objectif en le justifiant en fonction des priorités, des contraintes et des risques. Définir les tâches à réaliser pour atteindre l'objectif. La nature de chaque tâche doit être cohérente avec les activités d'ingénierie

logicielle présentées au chapitre 5.1. Dans le cas de Scrum, les tâches sont reliées à des stories.

Dans le cas d'une gestion de projet en cascade (i.e. non itérative), une étape est généralement la réalisation complète d'une activité d'ingénierie (par exemple "réaliser la spécification des exigences", "réaliser la conception").

5.4 Planification

Présenter un macro planning du projet. Ce planning doit être cohérent avec la démarche projet présentée au chapitre précédent. Mettre en avant les tâches sous votre responsabilité. Préciser également votre rôle dans l'élaboration de ce planning. S'il y a des écarts significatifs entre le planning initial et le planning réalisé, présenter les 2 plannings en expliquant les écarts.

Un diagramme de GANTT peut être utilisé. Mais dans ce cas, veuillez à rester à un niveau macro, c'est-à-dire facilement lisible sur une page. Un diagramme de GANTT détaillé peut être présenté en annexe.

5.5 Budget du projet

Le coût de projet au moins par rapport à vous. Locaux, machines, salaires, ...

INFO POUR LES CHAPITRES 6, 7 ET 8.

CES CHAPITRES PRESENTENT LE RESULTAT DES DIFFERENTES ACTIVITES D'INGENIERIE LOGICIELLE REALISEES (PRECISER CONCRETEMENT CE QUE VOUS AVEZ FAIT ET AUSSI LES RESULTATS QUE VOUS AVEZ SIMPLEMENT UTILISES).

LES ACTIVITE PRESENTEES DOIVENT ETRE EN COHERENCE AVEC CELLES QUI ONT ETE PRESENTEES AU CHAPITRE 5. LES TITRES DES CHAPITRES PEUVENT ETRE ADAPTES EN CONSEQUENCE. TOUTEFOIS ILS DOIVENT RESPECTER LA SEPARATION ENTRE LA VISION EXTERNE ET LA VISION INTERNE DU LOGICIEL.

DANS LE CAS D'UN PROCESSUS ITERATIF, CES ACTIVITES SONT REALISEES DE FACON DISCONTINUE, UNE PARTIE A CHAQUE ITERATION/SPRINT. CEPENDANT, IL CONVIENT DE PRESENTER ICI LE RESULTAT GLOBAL.

6. Exigences réalisées dans le projet (vision externe/utilisateur)

Dans le cas d'un existant, décrire ici uniquement les nouvelles exigences réalisées au sein du projet.

S'inspirer de l'étude de cas du cours démarche de conception.

6.1 Exigences fonctionnelles – Cas d'utilisation

Choisir 1 ou 2 cas d'utilisation représentatifs. Mettre en annexe la description des autres cas d'utilisation. Expliquer la valeur de ces cas d'utilisation pour l'utilisateur.

6.1.X Cas d'utilisation X

Description textuelle détaillée (voir votre cours)

Scénario(s) : Diagrammes de séquence (niveau système logiciel boîte noire, interactions entre le système et les acteurs extérieurs)

6.2 Exigences non fonctionnelles transverses

Utilisabilité (charte graphique, ergonomie générale)

Performances (temps de réponses, volumétrie)

Robustesse (tolérances aux pannes, disponibilité)

Sécurité

Supportabilité (maintenabilité, évolutivité)

6.3 Interfaces détaillées

6.3.1 IHM

CU d'utilisation X

écran 1

écran 2

6.3.2 Interfaces avec d'autres systèmes

7. Conception du système logiciel réalisée dans le projet (vision interne/développeur)

Dans le cas d'un existant, faire référence à l'existant. Bien faire ressortir ce qui change ou ne change pas par rapport à l'existant. **Justifier les choix.** Préciser votre rôle dans ces choix. Dans le cas d'un existant, faire ressortir ce qui change par rapport à l'existant (si aucun changement sur un aspect, faire référence au chapitre décrivant l'existant). (voir l'étude de cas du cours démarche de conception).

7.1 Plate-forme technique

Architecture matérielle et framework logiciel

7.2 Conception du logiciel développé

7.2.1 Conception du code source

Expliquer les choix structurants sur concernant la structuration en unités de code (par exemple en Java, la structuration en classes et packages) et en tables.

Expliquer dans quelle mesure cette structuration applique les bonnes pratiques de conception.

Expliquer les règles de nommage.

7.2.1.1 Le code source – vue statique

Architecture du code (structuration en packages)

Pour chaque package: structuration en classes/fichiers de code source – des exemples d'implémentation de code source peuvent être présentés (ou bien mis en annexe et référencés ici).

Décrire la structure de la BDD: description des tables dans la base la base de données. Justifier vos choix

7.2.1.2 Réalisation des cas d'utilisation

Choisir un cas d'utilisation significatif (1 de ceux présentés dans le chapitre précédent) et décrire le scénario de cas d'utilisation dans une vision interne/développeur (diagramme de séquence système logiciel boîte blanche)

7.2.2 Les composants et leur déploiement

Par exemple, en JEE, la structuration en fichiers .war, ejb-jar ...

Préciser les règles de nommage.

8. Tests du système logiciel

En cohérence avec la stratégie de test décrite au chapitre 3, décrire les tests réalisés et leurs résultats. Préciser votre rôle dans la définition et l'exécution des tests.

9. Conclusion générale

Principaux apports du stage pour vous, pour l'entreprise, pour l'école

Précisions sur le travail réalisé (nombre de programmes réalisés, nombre de classes implantées, ..., nombre de lignes de code, ...)

Précisions sur la situation finale de l'application réalisée (Le statu des livrables en production, ..., en test, fini à X %)

Principales difficultés rencontrées et solutions apportées

Perspectives ou encore ce qui reste à faire ou encore les extensions futures

10. Références et Bibliographie

- [1] Auteur1, auteur 2, ..., Titre_du_livre, Edition, date_parution, lien s'il ya
...
- [2] Auteur1, auteur 2, ..., Titre_article, Revue de parution ou site Web,
date_parution, lien s'il ya

**11. Annexes (à intégrer absolument si ce n'est pas déjà fait dans le rapport principal.
Elles peuvent être dans des fichiers séparés)**

- Annexe 1 : Exigences détaillées (correspond à peu près au cahier des charges)
- Annexe 2 : Modèle conceptuel de données (diagramme de classes)
- Annexe 3 : Dossier technique (Code réalisé y compris sa documentation)
- Annexe 4 : Présentation des outils de développement utilisés
- ...

NOTES IMPORTANTES :

- a) Les 40 pages maximum du rapport n'intègrent pas les annexes
- b) Dans le cas du TPT, certains enseignants, pour leur cours (Java avancée, HTML5, Androïd, NFC, Iphone, Conception avancée, Gestion de projets, administration/tuning, Bigdata/Data analytics) donneront une note projet (en plus de l'examen) à partir de ce rapport. Vous devez donc donner un maximum d'informations dans le rapport et ses annexes.
- c) Les éléments d'organisation du projet sont des objets vivants. Ils doivent évoluer avec le projet. Ils doivent être mis à jour régulièrement. Ils doivent être utilisables.