

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

Table des matières

Plan type d'un rapport de stage ou de Travail Pratique Transversal : TPT (pas plus de 40 pages hors annexes)	1
1. Introduction générale (1 page)	5
2. Présentation du stage	5
2.1 Présentation de l'entreprise	5
2.2 Présentation du sujet et objectifs du projet (problème traité)	5
3. Etat de l'art sur le sujet traité	5
3.1 Critères de comparaison	6
3.2 Etude de chaque solution au vu des critères	6
3.3 Tableau comparatif des solutions au vu des critères	6
4. Etude de l'existant et solution envisagée	6
4.1 Etude de l'existant	6
4.1.1 Description externe du système logiciel existant (vision utilisateur)	6
4.1.2 Description interne du système logiciel existant (vision développeur/conception)	6
4.2 Critique de l'existant	6
4.3 Solutions envisagées	6
4.4 Objectifs principaux et livrables	6
5. Démarche projet	6
5.1 Principes de la démarche projet	6
5.1.1 Activités d'ingénierie logicielle	6
5.1.2 Méthode de gestion de projet utilisée	6
5.1.3 Rôles et responsabilités	6
5.1.4 Outils	7
5.1.5 Gestion de la configuration	7
5.2 Contraintes et risques sur le projet	7
5.3 Démarche projet mise en œuvre	7
5.4 Planification	8
5.5 Budget du projet	8
6. Exigences réalisées dans le projet (vision externe/utilisateur)	8
6.1 Exigences fonctionnelles – Cas d'utilisation	8
6.1.1 Cas d'utilisation 1	8
6.1.2 Cas d'utilisation 2	9
6.1.3 Cas d'utilisation X	9
6.2 Exigences non fonctionnelles transverses	9
6.3 Interfaces détaillées	9

6.3.1	IHM.....	9
6.3.2	Interfaces avec d'autres systèmes.....	9
7.	Architecture(s) système.....	9
8.	Conception du système logiciel réalisée dans le projet (vision interne/développeur)	9
8.1	Plate-forme technique.....	10
8.2	Conception du logiciel développé.....	10
8.2.1	Conception du code source.....	10
8.2.2	Le code source – vue statique.....	10
8.2.3	Modélisation de données	10
8.2.4	Réalisation des cas d'utilisation	10
8.2.5	Les composants et leur déploiement.....	10
9.	Tests du système logiciel	10
10.	Conclusion générale.....	11
11.	Références et Bibliographie	11
12.	Annexes (à intégrer absolument si ce n'est pas déjà fait dans le rapport principal. Elles peuvent être dans des fichiers séparés)	12

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 pour chacun d'eux)

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

L'état de l'art consiste à faire une étude comparée de différentes solutions concernant un domaine d'intérêt de votre stage. Par exemple vous souhaitez créer une nouvelle monnaie électronique. Une étude des monnaies électroniques existantes s'impose. Vous souhaitez implémenter une nouvelle technologie blockchain, une étude des technologies existantes s'impose.

Pour effectuer de façon efficace l'étude, vous devez dans un premier temps élaborer des critères de comparaison.

Ensuite vous étudiez de façon exhaustive chaque solution au regard de ces critères.

Enfin vous produisez un tableau comparé des solutions y compris la vôtre.

Cela permet de se positionner, connaître la concurrence et voir si on n'a pas oublié des services.

NE PAS CONFONDRE ETAT DE L'ART ET ETUDE DE L'EXISTANT.

- 3.1 Critères de comparaison**
- 3.2 Etude de chaque solution au vu des critères**
- 3.3 Tableau comparatif des solutions au vu des critères**

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 la configuration, modèleur 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 1.2 ... 1.3 ...	1.1 Utiliser la version 1 de Jasmine 1.2 1.3	en cours
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 1.2 Acheter une formation chez CISCO Paris 2500 Euros pour 4 jours Prix catalogue 1.3 Embaucher un stagiaire d'une école de sécurité (au plus tard le 15 Avril)	Réalisé : 1.2 Vu l'urgence et la sensibilité

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) :

- Définir un backlog avec des users stories et / ou les tâches. Chaque user story ou tâche doit être décrit de façon exhaustive (voir le chapitre 6)
- Présenter le découpage en sprints/itérations dans lesquels les Users stories et les tâches sont affectés. Chaque sprint doit être clairement défini
- Utiliser les outils de pilotages d'un projet scrum (vélocité, etc.)

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, veillez 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.

Vous pouvez vous appuyer sur d'autres méthodes de planification en fonction des outils de gestion de projet que vous allez utiliser.

5.5 Budget du projet

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

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.

Notes : Les cas d'utilisations peuvent être remplacés par les User Stories si vous utilisez SCRUM par exemple.

6.1 Exigences fonctionnelles – Cas d'utilisation

Donner ici le tableau de toutes vos cas d'utilisations.

Choisir ensuite 3 ou 4 cas d'utilisation représentatifs que vous allez décrire dans cette section. Mettre en annexe la description des autres cas d'utilisation. Expliquer la valeur de ces cas d'utilisation pour l'utilisateur.

6.1.1 Cas d'utilisation 1

Description textuelle détaillée (voir un cours UML). Un cas doit contenir :

- Un nom
- Des préconditions
- Des postconditions
- Une description textuelle détaillée
- Des codes et messages d'erreurs.
- ...

Un cas peut être accompagné de certains diagrammes UML :

- Des diagrammes de séquence (niveau système logiciel boîte noire, interactions entre le système et les acteurs extérieurs)
- Des diagrammes d'états

- Des diagrammes d'activités
- Etc.

Un cas peut être accompagné d'écrans (voir la section 6.3)

6.1.2 Cas d'utilisation 2

...

6.1.3 Cas d'utilisation X

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)

Les exigences fonctionnelles doivent être décrites de façons concrètes et chiffrés. Par exemple si le temps de réponse d'une page web.

6.3 Interfaces détaillées

6.3.1 IHM

CU d'utilisation X

écran 1

écran 2

Faire suivre chaque écran d'une description de ce qui s'y passe.

6.3.2 Interfaces avec d'autres systèmes

Description d'API par exemple

7. Architecture(s) système

Dans ce chapitre vous devez décrire les architectures cibles de vos projets :

- L'*architecture logicielle* est une vue tournée sur l'organisation interne et le découpage en couches et modules du ou des logiciels du système informatique. Les responsabilités de chaque module et la nature et la structure des relations entre modules.
- L'*architecture technique* est une vue tournée vers les différents éléments matériels et l'infrastructure dans laquelle le système informatique s'inscrit, les liaisons physiques et logiques entre ces éléments et les informations qui y circulent³.

Il est important de décrire ces architectures et de positionner votre contribution.

8. 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).

8.1 Plate-forme technique

Architecture matérielle et framework logiciel

8.2 Conception du logiciel développé

8.2.1 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.

8.2.2 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).

8.2.3 Modélisation de données

Décrire ici le modèle conceptuel MERISE (dictionnaire de données et MCD) ou UML (modèles de classes).

Décrire ensuite le modèle logique de données.

8.2.4 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)

8.2.5 Les composants et leur déploiement

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

Préciser les règles de nommage.

9. 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.

10. Conclusion générale

La conclusion est l'occasion pour vous de faire un bilan sur votre projet. Ce bilan doit comporter les parties suivantes :

- Le bilan des résultats obtenus pour l'entreprise
- Le bilan des problèmes rencontrés et des solutions apportés
- Les perspectives du projet
- Le bilan personnel

Le bilan des résultats obtenus pour l'entreprise

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, finis à X %, etc.)

Les livrables peuvent être : des releases d'une application, des uses cases traités, des users storiess traitées, des sprints traités, des modules traités, etc.

Le bilan des problèmes rencontrés et des solutions apportés

Il s'agit ici de rapporter les principaux problèmes rencontrés et la façon dont ils ont été résolus.

Les perspectives du projet

Quelle est la suite possible à votre projet :

- Amélioration
- Nouveaux services
- Etc.

Le bilan personnel

Indiquer ici ce que ce stage et/ou projet a pu vous apporter.

11. 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

...

Les liens web seuls ne peuvent être une bibliographie. Vous devez mettre dans la bibliographie : des articles de revues, des livres, des documents en respectant les exemples ci-dessus. Si vous avez un doute, merci d'aller voir les références et bibliographies de n'importe quel article WIKIPEDIA.

12. 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) Concernant le stage, il y'aura trois soutenances (une première intermédiaire en Mars Orienté Organisation en présence du professeur de gestion de projet, un deuxième intermédiaire fin mai/début juin et la finale Début Septembre). Le professeur de gestion de projet donnera une note à lui au vu de la 1^{ère} soutenance. Vous devez donc donner un maximum d'informations dans le rapport et ses annexes.
- c) Les éléments d'organisation du projet (risques, planning, coûts, etc.) sont des objets vivants. Ils doivent évoluer avec le projet. Ils doivent être mis à jour régulièrement. Ils doivent être utilisables.