

Rapport de Stage - Journalier

Stagiaire :	Stagiaire Marie
Encadreur :	Encadreur Jean
Date de soumission :	02/09/2025

1. Activités Réalisées

Choix du processus unifié pour notre projet

Après l'analyse comparative ci-dessus, il est manifeste qu'il existe une diversité de méthodes de développement, chacune avec ses forces spécifiques. Pour notre projet de Gestion des Stagiaires, il est primordial de sélectionner une méthode qui soit à la fois efficace, optimale et alignée avec nos choix technologiques. Ayant opté pour la modélisation à l'aide du langage UML, il est naturel de choisir une méthode qui s'harmonise avec cette approche.

Ainsi, il est apparu nécessaire d'adapter le cadre général du Processus Unifié au contexte de notre projet. Après mûre réflexion, nous allons retenir le processus 2TUP (Two Tracks Unified Process).

Le processus 2TUP, construit autour du langage UML, définit une séquence d'étapes ordonnées qui contribueront à l'évolution de notre système. Son objectif est de produire des logiciels de qualité qui répondront aux besoins des utilisateurs dans des délais et des coûts prévisibles.

Les caractéristiques clés du processus 2TUP pour notre projet :

- **Itératif et incrémental** : Pour modéliser notre système, nous allons privilégier une approche en plusieurs cycles, affinant notre analyse et notre implémentation étape par étape. Cela nous permettra de mieux appréhender les incertitudes et les complexités inhérentes à un système de gestion avec différentes interactions utilisateur (connexion, gestion des messages, rapports, tâches, etc.).
- **Piloté par les risques** : Les causes potentielles d'échec de notre logiciel seront identifiées et traitées en priorité. Par exemple, la sécurité (hachage des mots de passe), la gestion des accès par rôle et la robustesse des interactions AJAX feront partie de nos préoccupations majeures dès le début du développement.
- **Conduit par les cas d'utilisations** : Le processus sera entièrement orienté par les besoins des utilisateurs que nous allons identifier pour le Stagiaire, l'Encadreur et l'Administrateur. Chaque fonctionnalité développée, du tableau de bord à la gestion des présences, découlera directement d'un cas d'utilisation défini (comme le montrera notre diagramme de cas d'utilisation).
- **Centré sur l'architecture du logiciel** : Une architecture adaptée constituera la pierre angulaire du succès de notre développement. Elle guidera nos choix stratégiques, notamment en ce qui concerne la conception de la base de données (MySQL), la structuration des classes PHP et la mise en œuvre de la logique métier. Cela déterminera une grande partie des qualités de notre logiciel, telles que sa maintenabilité, sa performance et son évolutivité.

Le cycle en Y du processus 2TUP appliqué à notre projet :

La branche gauche (fonctionnelle)

Cette branche sera dédiée à la compréhension des besoins du métier et des utilisateurs.

- **La capture des besoins fonctionnels** : Nous allons identifier et détailler les exigences du système en nous focalisant sur les actions des utilisateurs. Cela inclura des fonctionnalités telles que la connexion sécurisée, la consultation des messages, la création et la gestion des rapports, l'affectation et le suivi des tâches, la gestion des présences, l'évaluation des stagiaires et l'administration complète du système par l'administrateur. Cette phase permettra de qualifier au plus tôt le risque de produire un système inadapté.
- **L'analyse** : Cette étape consistera à étudier précisément les spécifications fonctionnelles afin d'obtenir une compréhension claire de ce que le système devra réaliser en termes de processus métier, indépendamment de toute technologie spécifique. Nous allons formaliser les interactions et les flux de données attendus pour chaque cas d'utilisation.

La branche droite (technique)

Cette branche se concentrera sur les aspects techniques et l'infrastructure du système.

- **La capture des besoins techniques** : Nous allons recenser toutes les contraintes techniques et les choix dimensionnant la conception de notre système. Cela inclura le choix de PHP pour le back-end, MySQL pour la base de données, JavaScript pour les interactions front-end, et l'intégration

2. Difficultés Rencontrées

Choix du processus unifié pour notre projet

Après l'analyse comparative ci-dessus, il est manifeste qu'il existe une diversité de méthodes de développement, chacune avec ses forces spécifiques. Pour notre projet de Gestion des Stagiaires, il est primordial de sélectionner une méthode qui soit à la fois efficace, optimale et alignée avec nos choix technologiques. Ayant opté pour la modélisation à l'aide du langage UML, il est naturel de choisir une méthode qui s'harmonise avec cette approche.

Ainsi, il est apparu nécessaire d'adapter le cadre général du Processus Unifié au contexte de notre projet. Après mûre réflexion, nous allons retenir le processus 2TUP (Two Tracks Unified Process).

Le processus 2TUP, construit autour du langage UML, définit une séquence d'étapes ordonnées qui contribueront à l'évolution de notre système. Son objectif est de produire des logiciels de qualité qui répondront aux besoins des utilisateurs dans des délais et des coûts prévisibles.

Les caractéristiques clés du processus 2TUP pour notre projet :

- **Itératif et incrémental** : Pour modéliser notre système, nous allons privilégier une approche en plusieurs cycles, affinant notre analyse et notre implémentation étape par étape. Cela nous permettra de mieux appréhender les incertitudes et les complexités inhérentes à un système de gestion avec différentes interactions utilisateur (connexion, gestion des messages, rapports, tâches, etc.).
- **Piloté par les risques** : Les causes potentielles d'échec de notre logiciel seront identifiées et traitées en priorité. Par exemple, la sécurité (hachage des mots de passe), la gestion des accès par rôle et la robustesse des interactions AJAX feront partie de nos préoccupations majeures dès le début du développement.
- **Conduit par les cas d'utilisations** : Le processus sera entièrement orienté par les besoins des utilisateurs que nous allons identifier pour le Stagiaire, l'Encadreur et l'Administrateur. Chaque fonctionnalité développée, du tableau de bord à la gestion des présences, découlera directement d'un cas d'utilisation défini (comme le montrera notre diagramme de cas d'utilisation).
- **Centré sur l'architecture du logiciel** : Une architecture adaptée constituera la pierre angulaire du succès de notre développement. Elle guidera nos choix stratégiques, notamment en ce qui concerne la conception de la base de données (MySQL), la structuration des classes PHP et la mise en œuvre de la logique métier. Cela déterminera une grande partie des qualités de notre logiciel, telles que sa maintenabilité, sa performance et son évolutivité.

Le cycle en Y du processus 2TUP appliqué à notre projet :

La branche gauche (fonctionnelle)

Cette branche sera dédiée à la compréhension des besoins du métier et des utilisateurs.

- **La capture des besoins fonctionnels** : Nous allons identifier et détailler les exigences du système en nous focalisant sur les actions des utilisateurs. Cela inclura des fonctionnalités telles que la connexion sécurisée, la consultation des messages, la création et la gestion des rapports, l'affectation et le suivi des tâches, la gestion des présences, l'évaluation des stagiaires et l'administration complète du système par l'administrateur. Cette phase permettra de qualifier au plus tôt le risque de produire un système inadapté.
- **L'analyse** : Cette étape consistera à étudier précisément les spécifications fonctionnelles afin d'obtenir une compréhension claire de ce que le système devra réaliser en termes de processus métier, indépendamment de toute technologie spécifique. Nous allons formaliser les interactions et les flux de données attendus pour chaque cas d'utilisation.

La branche droite (technique)

Cette branche se concentrera sur les aspects techniques et l'infrastructure du système.

- **La capture des besoins techniques** : Nous allons recenser toutes les contraintes techniques et les choix dimensionnant la conception de notre système. Cela inclura le choix de PHP pour le back-end, MySQL pour la base de données, JavaScript pour les interactions front-end, et l'intégration

3. Solutions Apportées

Choix du processus unifié pour notre projet

Après l'analyse comparative ci-dessus, il est manifeste qu'il existe une diversité de méthodes de développement, chacune avec ses forces spécifiques. Pour notre projet de Gestion des Stagiaires, il est primordial de sélectionner une méthode qui soit à la fois efficace, optimale et alignée avec nos choix technologiques. Ayant opté pour la modélisation à l'aide du langage UML, il est naturel de choisir une méthode qui s'harmonise avec cette approche.

Ainsi, il est apparu nécessaire d'adapter le cadre général du Processus Unifié au contexte de notre projet. Après mûre réflexion, nous allons retenir le processus 2TUP (Two Tracks Unified Process).

Le processus 2TUP, construit autour du langage UML, définit une séquence d'étapes ordonnées qui contribueront à l'évolution de notre système. Son objectif est de produire des logiciels de qualité qui répondront aux besoins des utilisateurs dans des délais et des coûts prévisibles.

Les caractéristiques clés du processus 2TUP pour notre projet :

- **Itératif et incrémental** : Pour modéliser notre système, nous allons privilégier une approche en plusieurs cycles, affinant notre analyse et notre implémentation étape par étape. Cela nous permettra de mieux appréhender les incertitudes et les complexités inhérentes à un système de gestion avec différentes interactions utilisateur (connexion, gestion des messages, rapports, tâches, etc.).
- **Piloté par les risques** : Les causes potentielles d'échec de notre logiciel seront identifiées et traitées en priorité. Par exemple, la sécurité (hachage des mots de passe), la gestion des accès par rôle et la robustesse des interactions AJAX feront partie de nos préoccupations majeures dès le début du développement.
- **Conduit par les cas d'utilisations** : Le processus sera entièrement orienté par les besoins des utilisateurs que nous allons identifier pour le Stagiaire, l'Encadreur et l'Administrateur. Chaque fonctionnalité développée, du tableau de bord à la gestion des présences, découlera directement d'un cas d'utilisation défini (comme le montrera notre diagramme de cas d'utilisation).
- **Centré sur l'architecture du logiciel** : Une architecture adaptée constituera la pierre angulaire du succès de notre développement. Elle guidera nos choix stratégiques, notamment en ce qui concerne la conception de la base de données (MySQL), la structuration des classes PHP et la mise en œuvre de la logique métier. Cela déterminera une grande partie des qualités de notre logiciel, telles que sa maintenabilité, sa performance et son évolutivité.

Le cycle en Y du processus 2TUP appliqué à notre projet :

La branche gauche (fonctionnelle)

Cette branche sera dédiée à la compréhension des besoins du métier et des utilisateurs.

- **La capture des besoins fonctionnels** : Nous allons identifier et détailler les exigences du système en nous focalisant sur les actions des utilisateurs. Cela inclura des fonctionnalités telles que la connexion sécurisée, la consultation des messages, la création et la gestion des rapports, l'affectation et le suivi des tâches, la gestion des présences, l'évaluation des stagiaires et l'administration complète du système par l'administrateur. Cette phase permettra de qualifier au plus tôt le risque de produire un système inadapté.
- **L'analyse** : Cette étape consistera à étudier précisément les spécifications fonctionnelles afin d'obtenir une compréhension claire de ce que le système devra réaliser en termes de processus métier, indépendamment de toute technologie spécifique. Nous allons formaliser les interactions et les flux de données attendus pour chaque cas d'utilisation.

La branche droite (technique)

Cette branche se concentrera sur les aspects techniques et l'infrastructure du système.

- **La capture des besoins techniques** : Nous allons recenser toutes les contraintes techniques et les choix dimensionnant la conception de notre système. Cela inclura le choix de PHP pour le back-end, MySQL pour la base de données, JavaScript pour les interactions front-end, et l'intégration