

Étude de cas : **Cycle en V – Développement d’un système d’automatisation de maison connectée**

| | |
|--|---|
| Phase de conception | 1 |
| 1. Document de spécification des exigences | 1 |
| 2. Plan de conception architecturale | 3 |
| 3. Plan de conception détaillé | 3 |
| Phase de mise en œuvre | 4 |
| Phase de validation | 4 |
| 1. Tests unitaires | 4 |
| 2. Tests d’intégration | 5 |
| 3. Tests du système | 5 |
| 4. Tests de validation | 6 |
| 5. Recette | 6 |

Phase de conception

1 – Document de spécification des exigences

La première étape de la phase de conception du projet consiste à regrouper et à **documenter les besoins du client, ses contraintes.**

Dans sa globalité, cette étape implique les éléments suivants :

- Les informations auprès des parties prenantes
- Les cas d’utilisation
- Les fonctionnalités attendues
- Les contraintes techniques
- Les critères de succès du projet
- Les performances requises

Le **document de spécifications des exigences** ou **SRS** (software requirement specification document) se compose généralement de quatre parties :

- L’introduction est l’aperçu global du projet. Elle définit notamment la portée du produit, sa valeur, son utilisation prévue, le public cible et d’autres éléments tels que la table des matières, les acronymes...

La Plateforme Formation

Société par actions simplifiée

Immatriculée au RCS de Marseille sous le numéro 883 780 496

Dont le siège social est situé à Marseille (13007) – 14 Traverse Canoubier

Tél : 04 84 89 43 69 • email : contact@laplateforme.io • www.laplateforme.io

- Les exigences systèmes et fonctionnelles qui se composent des fonctions et des fonctionnalités du produit. Le nombre d'exigences varie selon la taille du projet et peut se compter en milliers. Les plus courantes concernent la logique de traitement des données, les processus systèmes, le traitement des transactions, les exigences de performances, de sécurité...
- Les exigences d'interface externe. Elles garantissent la communication du système avec les éléments externes tels que les interfaces utilisateurs, matérielles, logicielles et de communication.
- Les exigences non fonctionnelles ou NFR regroupent les normes et manières dont le système doit mettre en œuvre les processus définis préalablement. Elles concernent des points comme la sécurité, la capacité, la maintenabilité, l'évolutivité, l'utilisabilité...

Le livrable attendu après cette étape est le **document de spécification des exigences**, mais aussi le **cahier des charges**.

Pour favoriser la mise en place de ces deux documents, il est possible de mettre en place une étude de faisabilité. Cette dernière a pour objectif de déterminer si le projet doit avancer, être modifié ou abandonné.

Pour ce faire, elle utilise les cinq cadres d'analyse pour considérer ses 4P (plan, processus, personnes, pouvoir), ses points de vulnérabilité, ses contraintes (calendrier, coûts, normes de qualité) et ses risques. Les cinq cadres sont :

- Cadre de définition
- Cadre des risques contextuels
- Cadre de potentialité
- Cadre paramétrique
- Cadre des stratégies dominantes

Pour déterminer les cinq domaines de faisabilité d'un projet, on peut utiliser l'acronyme TELOS :

T - Technique - Le projet est-il techniquement possible ?

E - Économique - Le projet peut-il être financé ? Cela augmentera-t-il les bénéfices ?

L - Juridique - Le projet est-il légal ?

O - Opérationnel - Comment les opérations actuelles tiendront-elles le changement ?

S - Planification - Le projet peut-il être réalisé à temps ?

Cette étape reprend les phases d'analyse et de spécification du modèle de cycle en V. Les **tests unitaires du système et d'acceptation** sont mis en place à ce moment.

2 – Plan de conception architecturale

Lorsque les exigences ont été définies, la seconde étape concerne la conception. On va délimiter les **principaux composants du système**, son **architecture**, et établir les **interactions et interfaces**. On parle alors de création de design pour l'ensemble du système. Il est possible d'utiliser un organigramme pour les tâches et les responsabilités afin de faciliter et d'optimiser ce processus.

L'organigramme des tâches, en anglais « Work Breakdown Structure » (WBS) ou Organigramme Technique de Projet (OTP), est un outil qui permet d'établir un plan de projet, hiérarchisé sous forme visuelle.

On intègre en général : une description de la tâche, un responsable, un budget, une date et un statut.

Les **tests d'intégration** sont également créés durant cette étape. On parle de conception de haut niveau.

3 – Plan de conception détaillé

Étape de conception de bas niveau, qui consiste à détailler la façon dont sera mise en œuvre la logique fonctionnelle codée du produit final.

Elle se concentre sur la spécification de chaque composant du système :

- Définition précise des algorithmes
- Des structures de données
- Des interfaces internes et externes
- Des interactions entre les différents modules

Cette phase repose entièrement sur la conception préliminaire ou architecturale, et se penche sur les aspects spécifiques et techniques de chaque élément du projet.

Elle nécessite souvent la collaboration entre les concepteurs, les développeurs et les experts métiers, afin de s'assurer que toutes les spécifications sont correctement définies.

Phase de mise en œuvre

Il s'agit de l'étape de jonction entre les deux phases de conception et de validation. Durant la phase de mise en œuvre, la totalité des documents issus de la conception sont traduits en code source en veillant à respecter les meilleures pratiques et normes de développement.

Différents outils et méthodes peuvent être utilisés pour la finalisation du projet.

Ce n'est que lorsque cette étape est terminée que l'on commence la phase de test, ou validation.

Phase de validation

1 - Tests unitaires

La phase de tests unitaires des composants permet de vérifier si les caractéristiques mises en place correspondent aux spécifications et exigences.

Ils consistent à vérifier chaque module ou composant du logiciel de manière individuelle, en isolant et en testant des parties spécifiques du code.

Il est extrêmement important que les composants soient contrôlés de manière isolée.

Ils permettent notamment de :

- Trouver les erreurs rapidement. Ceci permet de définir précisément l'interface du module à développer. Les tests sont exécutés durant tout le développement, permettant de visualiser si le code fraîchement écrit correspond au besoin.
- Sécuriser la maintenance. Ces tests unitaires signalent les éventuelles régressions. En effet, certains tests peuvent échouer à la suite d'une modification, il faut donc soit réécrire le test pour le faire correspondre aux nouvelles attentes, soit corriger l'erreur se situant dans le code.
- Documenter le code. Les tests unitaires peuvent servir de complément à l'API, il est très utile de lire les tests pour comprendre comment s'utilise une méthode.

Ils fonctionnent en général selon 4 phases :

- Initialisation
- Exercice
- Vérification
- Désactivation

2 – Tests d'intégration

Les tests d'intégration sont prévus pour vérifier que le système fonctionne correctement sur toutes les intégrations tierces et que les différents composants interagissent comme voulu.

Ils permettent également de vérifier l'aspect fonctionnel, les performances et la fiabilité du logiciel.

Le test d'intégration s'organise généralement en 3 parties :

1. Introduction et organisation
Présentation du document
organisation et procédures à suivre
2. Stratégie
identifier le produit fini du test d'intégration
identifier les spécifications du test d'intégration qui doivent être produites, mettre en évidence l'intérêt de chaque test
définir l'ordre dans lequel les tests doivent être effectués
3. Contenu et spécification
identifier la configuration matérielle requise pour le test
lister les outils et logiciels de test
identifier les assemblages précédemment testés nécessaires au test.

Il existe plusieurs méthodes pour les tests unitaires, les plus courantes étant Top-down, Bottom-up, Sandwich et Big-bang.

3 – Tests du système

Dans cette étape, l'ensemble du logiciel est testé. Toutes les fonctionnalités discutées sont testées, et l'on vérifie en même temps si les fonctionnalités fonctionnent exactement avec les entrées et les sorties définies lors de la phase de conception du système.

Voici un exemple des différents types de tests envisagés lors de tests système :

- test d'utilisabilité
- test de performance
- test de compatibilité
- gestion des exceptions
- test de volume

- test de sécurité
- test d'évolutivité

4 – Tests de validation

C'est le moment de la réalisation des tests de validation, créés pendant l'étape de conception du système. Ils se composent plus particulièrement de tests de performance et de régression.

5 – Recette

Dernière étape du cycle en V pour la gestion de projet informatique. C'est la mise en œuvre de tous les tests créés lors du stade initial de définition des exigences. La réalisation de ces tests se fait dans un environnement réel, avec des données réelles. Ceci a pour but de vérifier que le produit est prêt à être livré au client. On parle également de tests d'acceptation.

La procédure de recette se déroule en deux étapes principales :

- les tests système
- les tests d'acceptation utilisateur