REINVENT ENGINEERING

CENTRALE NANTES

Qualité Conception Modélisation

Activités et processus du génie logiciel

Plan

Quelques mythes du développement logiciel

Les acteurs du génie logiciel

Les activités du génie logiciel

Le développement - les différentes phases

Quelques mythes du développement logiciel

Pour coordonner le projet, une méthode et un bon livre de directives à suivre sont généralement suffisants pour guider les développeurs

Un équipement informatique à la pointe de la technologie suffit à garantir une production logicielle de qualité

En ajoutant des programmeurs, il est possible d'accélérer la production logicielle

A partir d'une description générale des besoins, on peut immédiatement commencer l'implantation d'une solution

Des modifications tardives peuvent être intégrées sans problème et sans surprise du moment que le logiciel est flexible

Le travail d'un développeur est juste de produire un logiciel fonctionnel

La production de documents lors du développement fait perdre beaucoup de temps

Plan

Quelques mythes du développement logiciel

Les acteurs du génie logiciel

Les activités du génie logiciel

Le développement - les différentes phases

Les acteurs du génie logiciel

Un projet de développement logiciel fait intervenir différents acteurs chacun avec son rôle

Les rôles sont :

Les **utilisateurs** ou usagers : généralement expriment le besoin initial et sont les utilisateurs finaux

La **maîtrise d'ouvrage** : entreprise ou service client

La **maîtrise d'ouvrage déléguée** : en charge du suivi du projet pour le compte du client

La **maîtrise d'œuvre** : entreprise ayant pris l'engagement par contrat de réaliser le projet

Les acteurs du génie logiciel

La **maîtrise d'œuvre déléguée** : chef de projet

- Liaison avec le client
- Évaluation des ressources
- Gestion du projet
- Décision de passage du projet d'une étape à une autre

L'**équipe projet** : experts métiers, analystes, développeurs, testeurs, intégrateurs, etc.

L'**équipe validation** : plans de test d'intégration, système et acceptation (recette avec la maîtrise d'œuvre déléguée et la maîtrise d'ouvrage)

Le **responsable qualité** : aide à la mise en œuvre de la qualité et à l'organisation du contrôle qualité

Plan

Les activités du génie logiciel

La norme ANSI/IEEE Std 1002-1987 définit trois catégories d'activités mises en œuvre par le génie logiciel :

- Ingénierie du produit (*product engineering)*
- Vérification et validation
- Gestion technique (*technical management*)

L'importance de chaque activité est variable selon le modèle de développement logiciel retenu et la nature du produit à réaliser

Les activités du génie logiciel

Définition de l'IEEE :

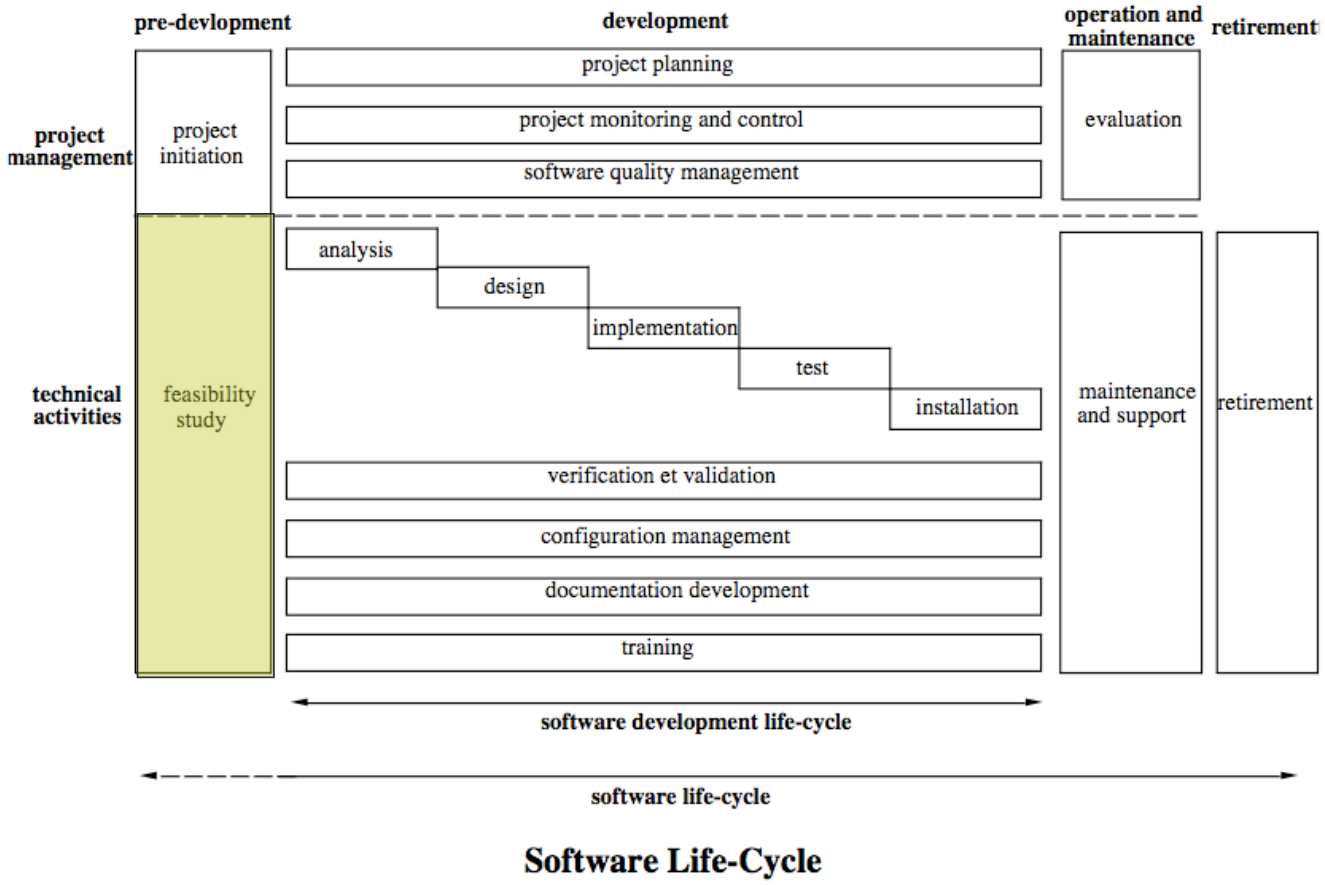
La production d'un logiciel comprend plusieurs étapes et s'étend sur une longue période de temps. Le cycle de vie d'un logiciel, c'est la période de temps qui débute au moment de la définition et du développement d'un produit logiciel et se termine lorsque le logiciel n'est plus disponible pour utilisation

Les activités du génie logiciel

La norme ISO/EIC 12207:1995 modélise le cycle de vie du logiciel :

- Avant projet (*conception exploration*)
- Développement logiciel (*software developement cycle*)
- Exploitation et maintenance (*operation and maintenance phase*)
- Retrait (*retirement phase*)

Les activités du génie logiciel [3]



Software Life-Cycle

Avant-projet - Étude préalable

Feasibility study

Objectif : Éviter de développer un logiciel non adéquat

- Pourquoi développer le logiciel?
- Comment procéder pour ce développement?
- Quels moyens faut-il mettre en œuvre ?

À faire :

- Dresser l'état des lieux de l'existant
- Analyser ses forces et faiblesses
- Identifier et affiner l'énoncé des besoins de l'utilisateur
- Étudier la faisabilité et formuler des solutions potentielles
- Planifier la transition entre l'ancien et le nouveau logiciel s'il y a lieu
- Etc.

Avant-projet : La note de cadrage

1. Introduction
2. La portée et les limites de l'étude
3. Le budget
4. L'organisation
5. Le calendrier
6. Annexes

Avant-projet : La note de cadrage

1. Introduction
    1. Rappel du contexte dans lequel l'étude a été lancée
    2. Indication des Directions et (ou) des événements déclencheurs à l'origine de l'étude
    3. Description succincte de l'étude
2. La portée et les limites de l'étude
3. Le budget
4. L'organisation
5. Le calendrier
6. Annexes

Avant-projet : La note de cadrage

1. Introduction
2. La portée et les limites de l'étude
3. Le budget
   - Gain attendu (commission, impact trésorerie ....)
     - Coûts et charges (externes ou internes)
     - Charges (MOA/MOE/utilisateurs/Organisation, assistance externe)
     - Coûts informatiques (environnements, logiciels et matériels, ...)
     - Coûts de déploiement (formation, manuels utilisateur, gestion du changement...)
   - Coûts de fonctionnement prévisibles
   - Numéro de ligne budgétaire
4. L'organisation
5. Le calendrier
6. Annexes

Avant-projet : La note de cadrage

4. L'organisation
    1. Intervenants
        - Direction chargée du projet :
        - Responsable du projet :
        - Autres acteurs :
            - Maîtres d'œuvre
            - Maîtres d'ouvrage
            - Utilisateurs
            - Organisation
            - Assistance extérieure
    2. La démarche retenue pour l'avant projet
        - Actions à mener
        - Résultats et livrables attendus (décisions, documents ...)
    3. Organisation spécifique de l'avant-projet
        - Comités : rôle, composition, compétence, périodicité
        - Documentation associée (localisation, codification du projet, organisation des répertoires)

Avant-projet : La note de cadrage

1. Introduction
2. La portée et les limites de l'étude
3. Le budget
4. L'organisation
5. Le calendrier
    1. Dates clefs
    2. Phasage du projet
    3. Planning possible
6. Annexes

Avant-projet : La note de cadrage

1. Introduction
2. La portée et les limites de l'étude
3. Le budget
4. L'organisation
5. Le calendrier
6. Annexes
   - Etudes de marchés, etc ...

Avant-projet - document de définition des besoins

Le document de définition des besoins porte sur :

- Les buts du projet
- Les besoins
- La portée du projet
- Les limites du projet

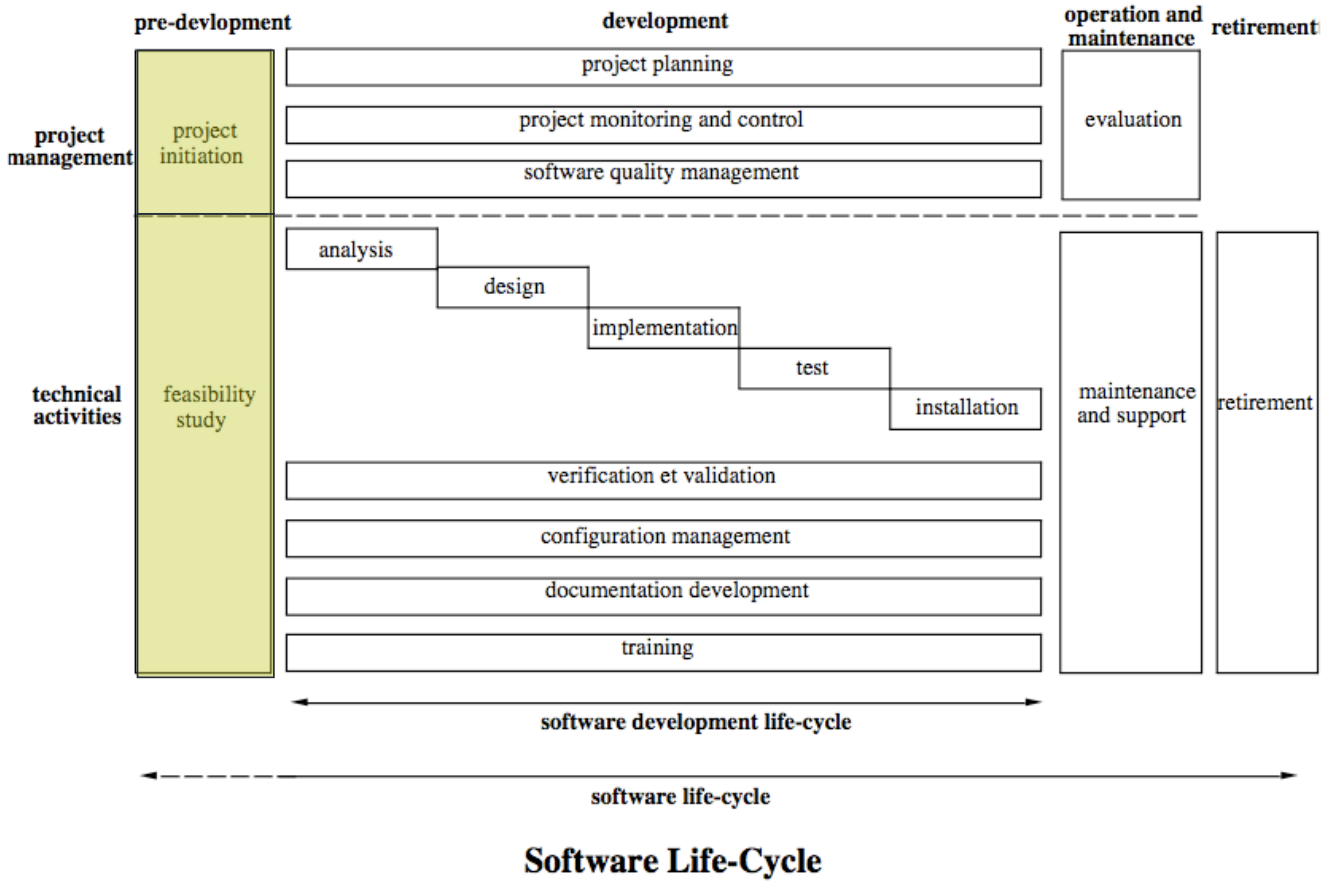
Le document de définition des besoins énonce :

- Les spécifications techniques
- Les spécifications administratives
- Les spécifications financières

Avant-projet - document de définition des besoins

« Un ensemble de propriétés ou de contraintes, décrites de façon précise, qu'un système logiciel doit satisfaire » [Yeh et Zave, 1980]

Les activités du génie logiciel [3]



Software Life-Cycle

Avant-projet - Initialisation du projet

Project initiation

Les tâches à réaliser sont :

- Représenter les activités à entreprendre au sein d'un modèle
- Prévoir les ressources nécessaires
- Mettre en place des environnements de développement et de support
- Identifier des procédures et des normes spécifiques au projet, ainsi que les mesures à appliquer pour les contrôler
- Planifier la gestion de projet

Avant-projet - le document avant-projet

1. Introduction
2. Analyse des solutions
3. Solution retenue
4. Le budget
5. L'organisation
6. Le phasage et la planification
7. Annexes

Avant-projet - le document avant-projet

1. Introduction
   - Rappel de la définition du projet
   - Rappel des dates clefs
   - Rappel de la démarche
2. Analyse des solutions
3. Solution retenue
4. Le budget
5. L'organisation
6. Le phasage et la planification
7. Annexes

Avant-projet - le document avant-projet

1. Introduction
2. Analyse des solutions
   - Description des différentes solutions possibles (ou référence au « Dossier de choix de la solution » élaboré au préalable)
   - Proposer toutefois une synthèse de la démarche utilisée (présentation des solutions, mode de notations, comparatif, synthèse et recommandation ...)
3. Solution retenue
4. Le budget
5. L'organisation
6. Le phasage et la planification
7. Annexes

Avant-projet - le document avant-projet

1. Introduction
2. Analyse des solutions
3. Solution retenue
   - Description de la solution :
     - Description fonctionnelle
     - Description technique
     - Niveau de qualité attendu
     - Impact de la solution sur l'organisation
   - Faisabilité :
     - Avantages
     - Inconvénients
   - Conditions de mise en oeuvre :
     - Moyens humains et matériel nécessaire
     - Besoins formation, documentation, communication
     - Sous-projet
     - Déploiement
   - Reprise de l'existant :
     - Principes et scénario général de reprise
4. Le budget
5. L'organisation
6. Le phasage et la planification
7. Annexes

Avant-projet - le document avant-projet

1. Introduction
2. Analyse des solutions
3. Solution retenue
4. Le budget
   - Gain attendu (commission, impact trésorerie ....)
   - Coûts et charges (externes ou internes) :
     - Charges (MOA/MOE/utilisateurs/Organisation, assistance externe)
     - Coûts informatiques (environnements, logiciels et matériels, ...)
     - Coûts de déploiement (formation, manuels utilisateur, ...)
     - Coûts de fonctionnement prévisibles
   - Numéro de ligne budgétaire
5. L'organisation
6. Le phasage et la planification
7. Annexes

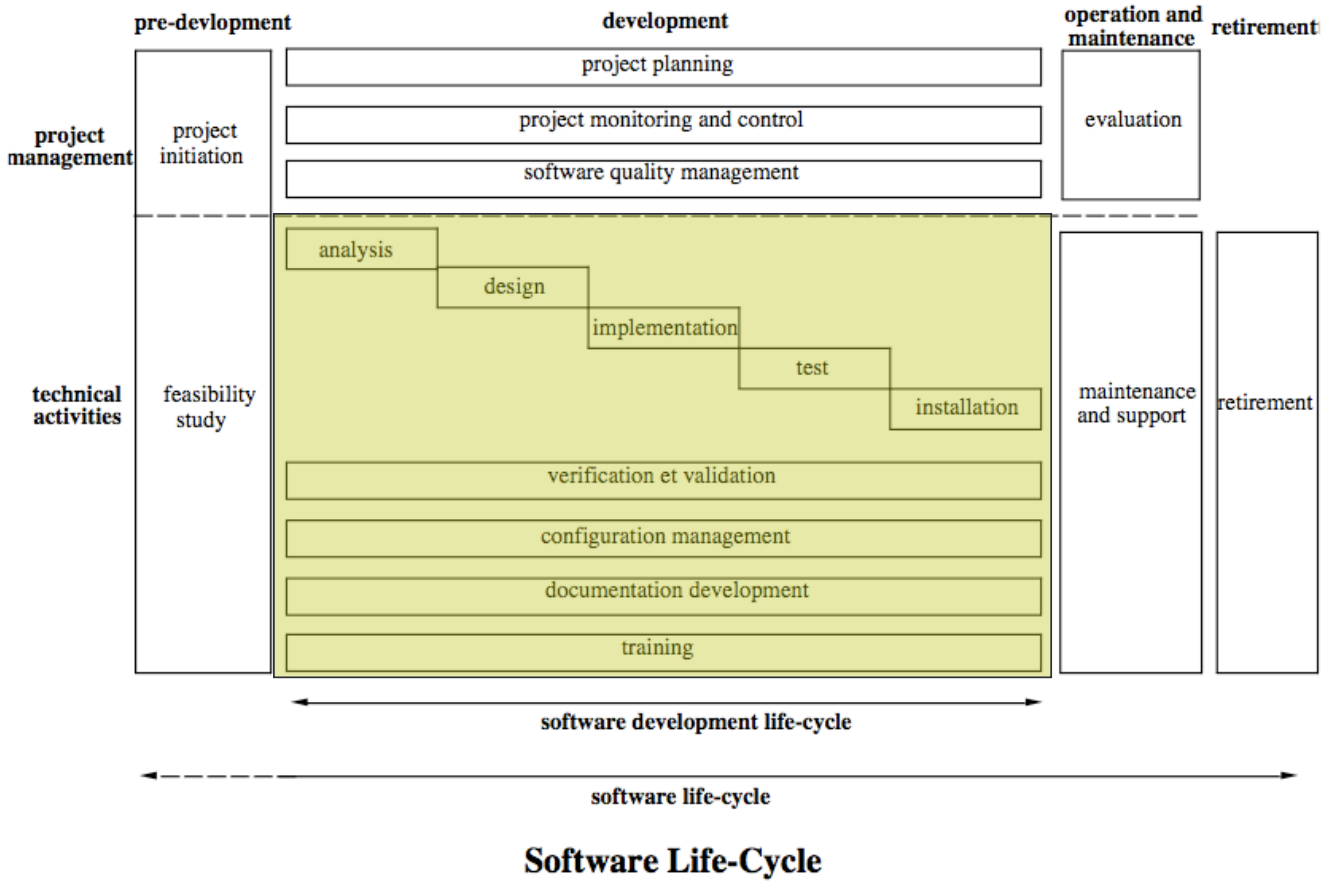
Avant-projet - le document avant-projet

5. L'organisation
   1. Les intervenants :
      - Direction chargée du projet
      - Responsable du projet, (cf. lettre de mission)
      - Autres acteurs (désignation des équipes, rôles et responsabilités) :
        - Maîtres d'œuvre
        - Maîtres d'ouvrage
        - Utilisateurs
        - Organisation
        - Assistance extérieure
      - L'organigramme du projet
   2. Les instances de fonctionnement du projet :
      - Comité de pilotage :
        - Composition
        - Rôle
        - Compétence
        - Périodicité
      - Comité de suivi :
        - Composition
        - Rôle
        - Compétence
        - Périodicité
   3. Méthode de rapport choisie (*reporting*)
   4. Documentation associée (localisation, codification du projet, organisation des répertoires ...)

Avant-projet - le document avant-projet

1. Introduction
2. Analyse des solutions
3. Solution retenue
4. Le budget
5. L'organisation
6. Le phasage et la planification
   - Phasage et étapes du projet
   - Référentiel des livrables attendus à chaque étape
   - Planification du projet et méthode de suivi (en délai, en charge, en qualité ...)
7. Annexes

Les activités du génie logiciel [3]



Software Life-Cycle

Le développement du logiciel - aspects techniques

Le cycle de développement du logiciel présente une grande variété d'activités exécutées successivement et/ou en parallèle

Les activités transversales qui accompagnent le développement logiciel sont :

- La gestion de projet
- La vérification et la validation
- La production de documentation
- La gestion de configuration
- La formation

Aspects techniques - La vérification et la validation

Ensemble des activités techniques déployées pour maîtriser la qualité du produit

La validation d'un produit (intermédiaire ou finale) revient à s'assurer que le produit correspond aux attentes

- Les critères de validation appliqués sont externes au produit lui-même :
  - Conformité au cahier des charges
  - Satisfaction des utilisateurs

Aspects techniques - La vérification et la validation

La vérification d'un produit consiste à s'assurer que le produit est bien construit, *ie* que sa construction satisfait les exigences de qualité spécifiées

Aspects techniques - La vérification et la validation

Les actions à entreprendre sont :

Inspection ou revue

Audit

Test

Aspects techniques - La vérification et la validation

L'exécution du programme qualité comporte les actions suivantes :

Déterminer les produits à évaluer et les procédures d'évaluation

Collecter et analyser les informations

Planifier l'effort de test

Développer des spécifications et procédures de test

Faire les vérifications et validations, exécuter les tests

➔ **Assurance qualité du logiciel** (*Software Quality Assurance*)

Aspects techniques - La gestion de configuration

➔ maîtriser l'évolution des éléments pendant le cycle de vie

Elle comprend :

Identifier et définir les éléments de configuration et toutes leurs relations

Archiver les éléments de configuration (états initiaux et successifs)

Contrôler les modifications des éléments de configuration, ce qui inclut pour chaque modification :

- Autoriser le changement
- Vérifier que le nouvel état est complet et cohérent
- Enregistrer le nouvel état
- Annoncer la modification
- Mettre à disposition le nouvel état

Aspects techniques - La gestion de configuration

Différents types d'éléments peuvent être pris en compte par la gestion de configuration :

Les documents de gestion du projet

Les documents techniques de réalisation

Les manuels d'utilisation et d'exploitation

Les programmes sources et les moyens permettant de produire et de reproduire le programme machine

Les jeux de données de test (avec leurs résultats)

Les procédures et scénarios de test

Aspects techniques - La documentation

C'est un élément essentiel dans le développement d'un logiciel :

Fait partie du produit à réaliser

Matérialise l'avancement des travaux

Garantit la continuité du projet à travers la traçabilité des documents produits d'une phase à l'autre

Constitue le support de communication entre les différents intervenants du projet

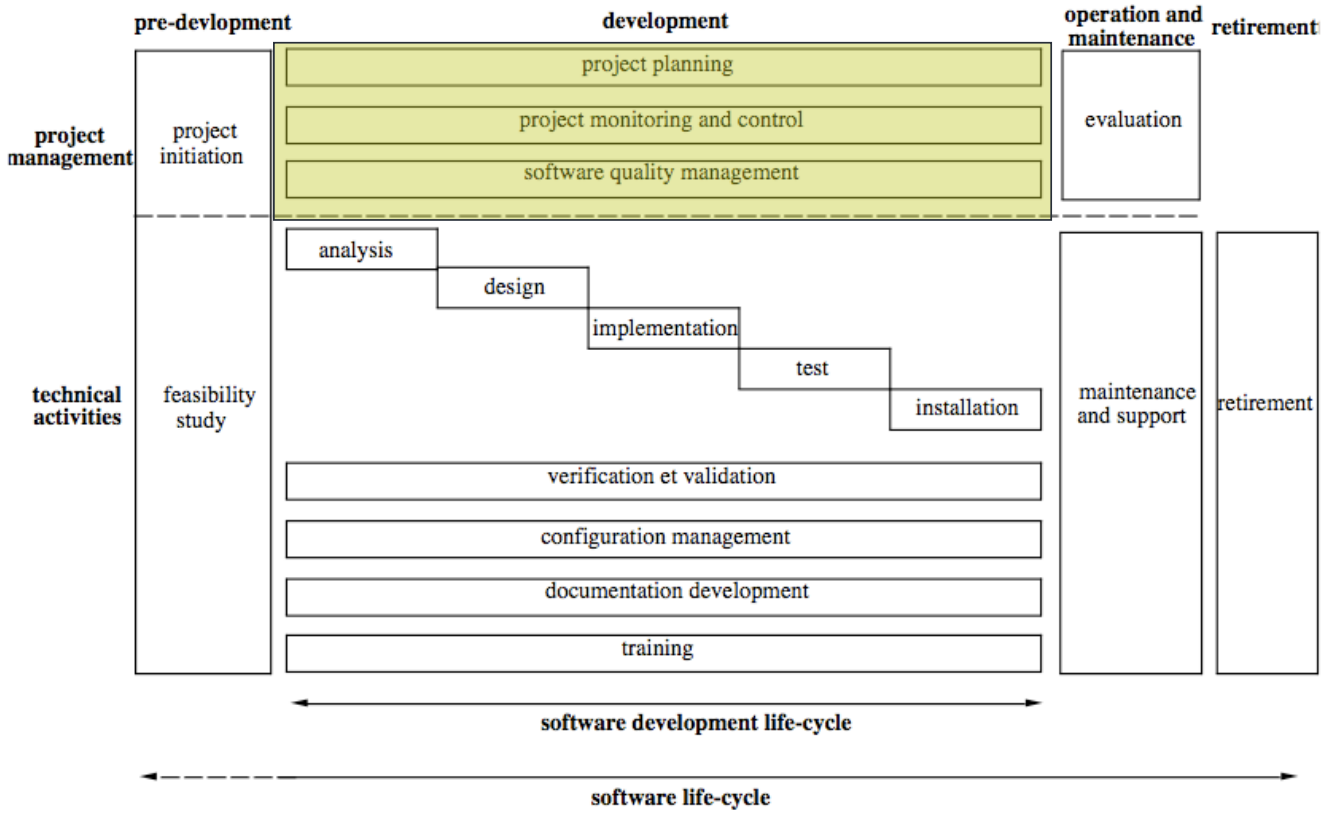
On distingue trois types de documents :

Documents de gestion du projet

Documents techniques de réalisation

Manuels d'utilisation et d'exploitation

Les activités du génie logiciel [3]



Software Life-Cycle

Le développement du logiciel - gestion de projet

La gestion de projet consiste en :

L'affinement et les modifications de la planification du projet

- Décomposition en tâches plus élémentaires
- Produit une structure de référence

➔ La planification de projet définit les tâches, le calendrier, les ressources, l'allocation de ces ressources aux tâches et les procédures du projet

Le pilotage et le suivi du projet

- Enregistre les faits sur l'avancement du projet
- Compare l'avancement avec la planification
- Entreprend, si nécessaire, des mesures correctives

La gestion de qualité du logiciel

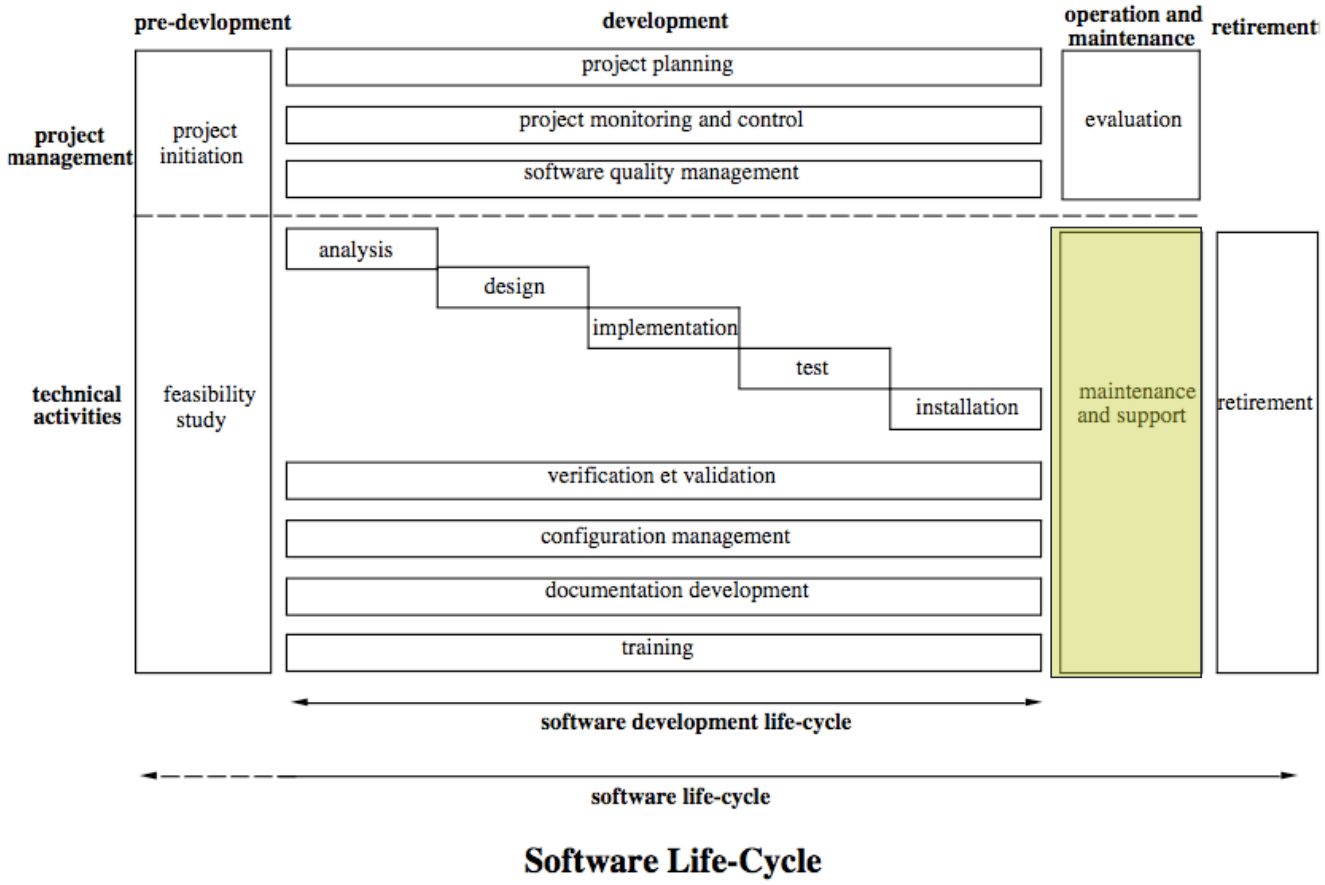
La gestion de projet - gestion de la qualité du logiciel

C'est le procédé qui pilote les activités techniques de vérification et de validation

La gestion de qualité se découpe en différentes tâches :

- Planifier le programme de qualité
- Mettre en place un système de comptes-rendus de problèmes et d'actions correctives
- Définir un programme pour mesurer la qualité
- Piloter et contrôler l'application du programme de qualité
- Recommander des améliorations pour les programmes de qualité futurs

Les activités du génie logiciel [3]



Software Life-Cycle

L’exploitation et la maintenance

Fourniture de la **recette** du logiciel

- Recevabilité
- Qualification

➔Signature du PV de recette :

- ➔ Transfert de responsabilité entre fournisseur et client

L’exploitation et la maintenance

Maintenance et le support logiciel :

Effectuer un dépannage pour des corrections mineures

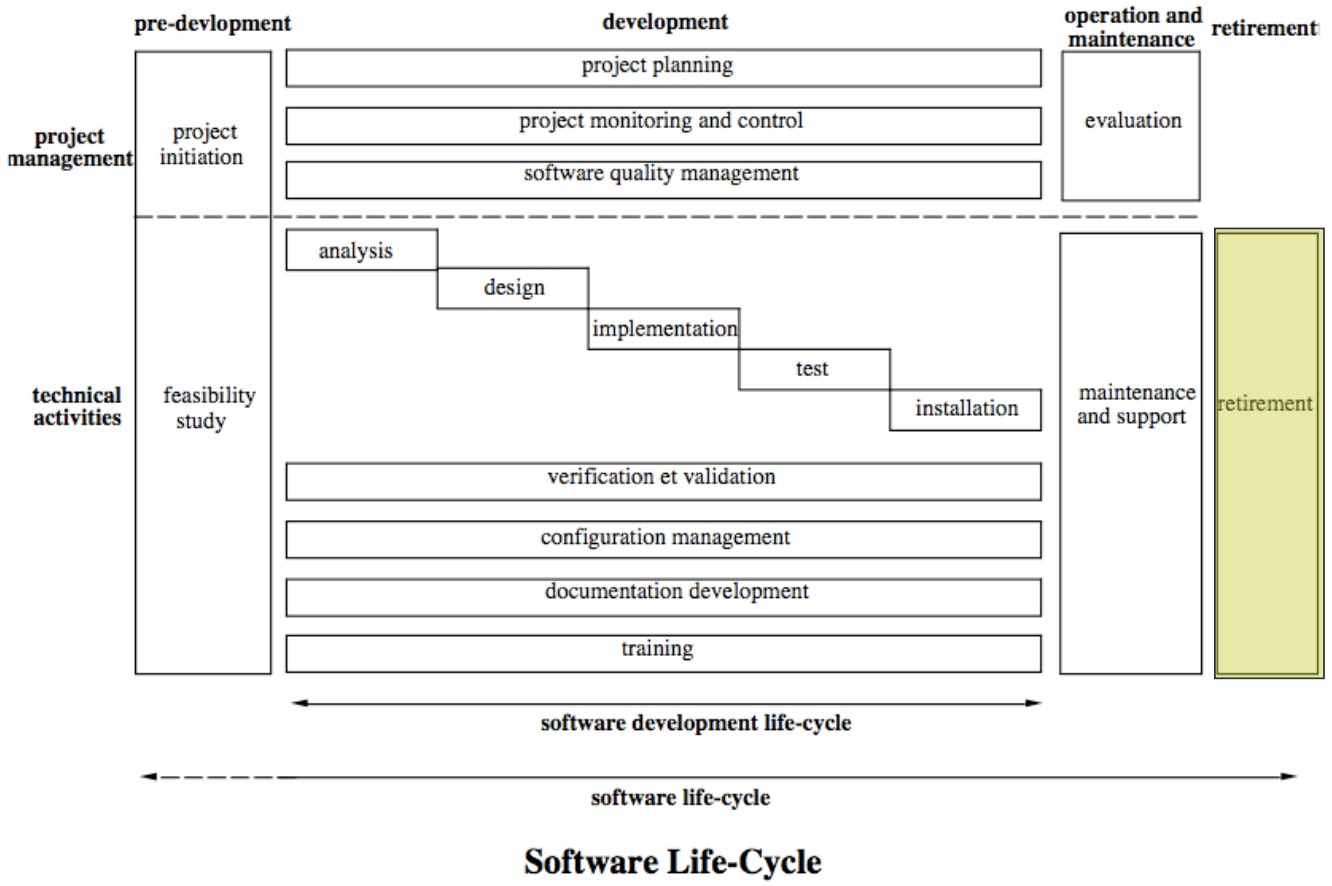
Réappliquer le cycle de développement pour des modifications plus importantes

Distribuer les mises à jour

Fournir l’assistance technique et un support de consultation

Maintenir un journal des demandes d’assistance et de support

Les activités du génie logiciel [3]



Software Life-Cycle

Le retrait

La décision est prise de mettre le logiciel hors service

Les tâches suivantes sont accomplies pendant le retrait :

Avertir les utilisateurs

Effectuer une exploitation en parallèle du logiciel et de son successeur (si il existe)

Arrêter le support du logiciel

Plan

Quelques mythes du développement logiciel

Les acteurs du génie logiciel

Les activités du génie logiciel

Le développement - les différentes phases

- Analyse
- Conception
- Implémentation
- Test
- Installation

Le développement - les différentes phases

Analyse

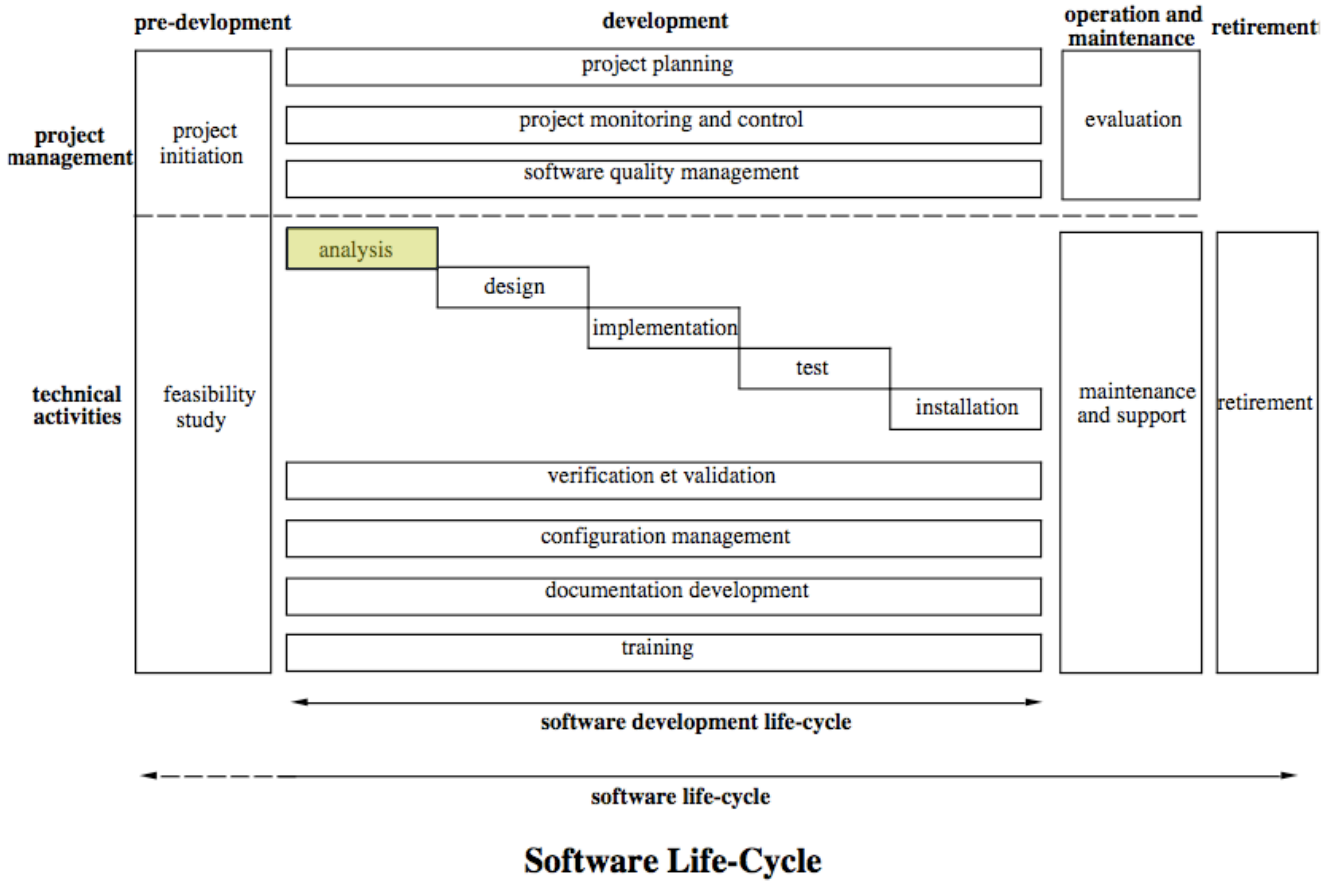
Conception

Intégration

Test

Installation

Les activités du génie logiciel [3]



Software Life-Cycle

Analyse [5]

Cette phase peut se diviser en :

Analyser les besoins des utilisateurs ou du système englobant

Spécifier le comportement du logiciel

➔ **cahier des charges du logiciel**

Analyse [5]

L'étude des besoins devra porter sur :

Le domaine d'application (métier)

L'état actuel de l'environnement (matériel et logiciel) du futur système

Le rôle du futur système

Les ressources disponibles et requises

Les contraintes d'utilisation et les performances attendues

Etc.

Les données sont fournies par des experts du domaine d'application et par les futurs utilisateurs

Analyse [5]

L'étude se fera à l'aide de méthodes des sciences cognitives :

Entretiens

Questionnaires

Observation de l'existant

Analyse de situations analogues

Instauration d'un dialogue avec les experts métier

Analyse [5]

Le résultat de l'étude est un ensemble de documents décrivant les aspects pertinents de l'environnement du futur système, son rôle et sa future utilisation

Le partage logiciel/matériel est définit en relation avec les études de faisabilité

La phase d'analyse des besoins se poursuit durant tout le processus de développement et durant tout le cycle de vie du logiciel

Analyse - spécification

Représente un modèle de ce qui est nécessaire et un énoncé du problème pris en compte

Est élaborée par une approche itérative de modélisation et d'analyse de la réalité modélisée

Est une description de ce que doit faire le logiciel (*quoi* et non *comment*)

Représente une vue externe du logiciel, tout ce qui peut être perçu sans pénétrer dans la structure interne

Analyse - spécification

Est composée de :

Description de l'environnement logiciel

Spécifications fonctionnelles qui définissent toutes les fonctions que le logiciel doit offrir

Comportement en cas d'erreurs (*ie.* dans le cas où le logiciel ne peut pas accomplir sa fonction)

Performances requises (temps de réponse, encombrement mémoire, sécurité de fonctionnement)

Analyse - spécification

Est composée de :

Interfaces utilisateurs (terminaux, présentation des écrans, disposition des impressions)

Interfaces avec d'autres logiciels

Interfaces avec le matériel

Contraintes de réalisation (environnement de développement, langage de programmation, procédures et normes,...)

Analyse - spécification - Le cahier des charges fonctionnel général [IEEE Std 830:1998]

1. Présentation du projet
    1. Objectifs et destinataires du document
    2. Portée du logiciel
    3. Définition, acronymes et abréviations
    4. Références bibliographiques
    5. Plan du document
2. Description générale
    1. Environnement du logiciel
    2. Modèle conceptuel
    3. Caractéristiques des utilisateurs
    4. Hypothèses et dépendances
    5. Priorisation des exigences
3. Spécification détaillée

Annexes

Index

Analyse - spécification détaillée [2]

Cette partie du cahier des charges doit être adaptée à la démarche de spécification et de conception retenue :

Sept approches possibles :

1. Les modes du système - entraînement, normal, urgence
2. Les catégories d'usagers - employé, patron, administrateur
3. Les objets (ou entités) - client, DAB, banque
4. Les propriétés et les services - dépôt, retrait, consultation
5. Les évènements ou stimuli - carte insérée, transaction validée
6. Les produits ou artéfacts - délivrer billet, générer reçu
7. Les fonctions - description fonctionnelle détaillée et dictionnaire de données

Ces approches peuvent être combinées

Analyse - spécification détaillée selon l'approche objet

1. Spécifications d'interfaces
2. Classes/Objets
3. Spécifications de performances
4. Contraintes de conception
5. Qualités requises par le logiciel
6. Autres contraintes

Analyse - spécification détaillée selon l'approche objet

1. Spécifications d'interfaces
    1. Interfaces utilisateur
    2. Interfaces matérielles
    3. Interfaces logicielles
    4. Interfaces de communication
2. Classes/Objets
3. Spécifications de performances
4. Contraintes de conception
5. Qualités requises par le logiciel
6. Autres contraintes

Analyse - spécification détaillée selon l'approche objet

1. Spécifications d'interfaces
2. Classes/Objets
   1. Classe/Objet 1
      - Attributs (directs et hérités)
        - Attribut 1
        - ....
        - Attribut n
      - Fonctions (services, méthodes directs et hérités)
        - Spécification fonctionnelle 1
        - ....
        - Spécification fonctionnelle m
      - Collaborations (messages reçus et/ou envoyés)
   2. Classe/Objet 2
   3. ....
   4. Classe/Objet p
3. Spécifications de performances
4. Contraintes de conception
5. Qualités requises par le logiciel
6. Autres contraintes

Analyse - spécification détaillée selon l'approche objet

1. Spécifications d'interfaces
2. Classes/Objets
3. Spécifications de performances
4. Contraintes de conception

   Conformité aux normes, limitation matérielle, etc.
5. Qualités requises par le logiciel
6. Autres contraintes

Analyse - spécification détaillée selon l'approche objet

1. Spécifications d'interfaces
2. Classes/Objets
3. Spécifications de performances
4. Contraintes de conception
5. Qualités requises par le logiciel
   Sécurité, portabilité, etc.
6. Autres contraintes

Analyse - spécification détaillée selon l'approche objet

1. Spécifications d'interfaces
2. Classes/Objets
3. Spécifications de performances
4. Contraintes de conception
5. Qualités requises par le logiciel
6. Autres contraintes

   Base de données, exploitation, migration, etc.

Analyse - spécification détaillée selon l'approche fonctionnelle

1. Architecture fonctionnelle
2. Les fonctions

Analyse - spécification détaillée selon l'approche fonctionnelle

1. Architecture fonctionnelle
   - Représenter l'intégration de l'application dans le système existant
   - Exprimer l'ensemble des interactions du projet avec le système existant
   - Identifier les flux externes de façon exhaustive

Analyse - spécification détaillée selon l'approche fonctionnelle

2. Les fonctions
   - Modèle détaillé des fonctions
     - Identifier et lister la totalité des fonctions attendues
     - Décrire les enchaînements de fonctions
     - Représenter le modèle détaillé des fonctions
   - Description détaillée de l'ensemble des fonctions

   Décrire de façon détaillée chaque fonction :
     - définition et objectif
     - description fonctionnelle :
       - traitements
       - écrans
       - enchaînement d'écrans
       - règles de gestion
       - contrôles sur les données
   - Lister par fonctions les entrées/sorties

Analyse - cahier des charges

La norme IEEE 830:1998 définit un bon cahier des charges comme :

Non ambigu - élaboration d'un glossaire

Complet - prévision des situations exceptionnelles

Vérifiable - les fonctionnalités doivent correspondre à leur spécification

Consistant - non contradiction des spécifications

Modifiable - reports aisé des modifications tardives

Traçable - références croisées entre les versions successives

Utilisable durant la maintenance - tentative de prévision des évolutions

Analyse - objectifs d'un cahier des charges [7]

[Heninger, 80]

Spécifier uniquement les comportements externes du logiciel, sans faire référence à une implémentation

Spécifier les contraintes d'implémentation (système d'exploitation, interfaces avec l'utilisateur et avec le système, etc.)

Être facilement modifiable

Servir de document de référence

Contenir des précisions sur l'évolution future du logiciel (modifications prévisibles, fonctions à ajouter ou supprimer)

Définir les réponses acceptables à des événements non désirés

Analyse - exemple de cahier des charges

Voici quelques spécifications tirées du cahier des charges pour l'environnement de programmation de ADA (cahier des charges appelé STONEMAN) :

4.C.1 Une interface virtuelle, indépendante de toute machine hôte, doit être fournie pour communiquer avec l'APSE (*Ada Programming Support Environment*)

4.C.2 L'interface virtuelle doit être basée sur des concepts simples, évidents à comprendre et à utiliser, et peu nombreux

…

4.C.8 Toutes les communications entre l'utilisateur et l'APSE doivent pouvoir s'exprimer en utilisant le jeu de caractère standard de Ada

Analyse - exemple de cahier des charges

Ces spécifications appellent les remarques suivantes [Sommerville, 82]:

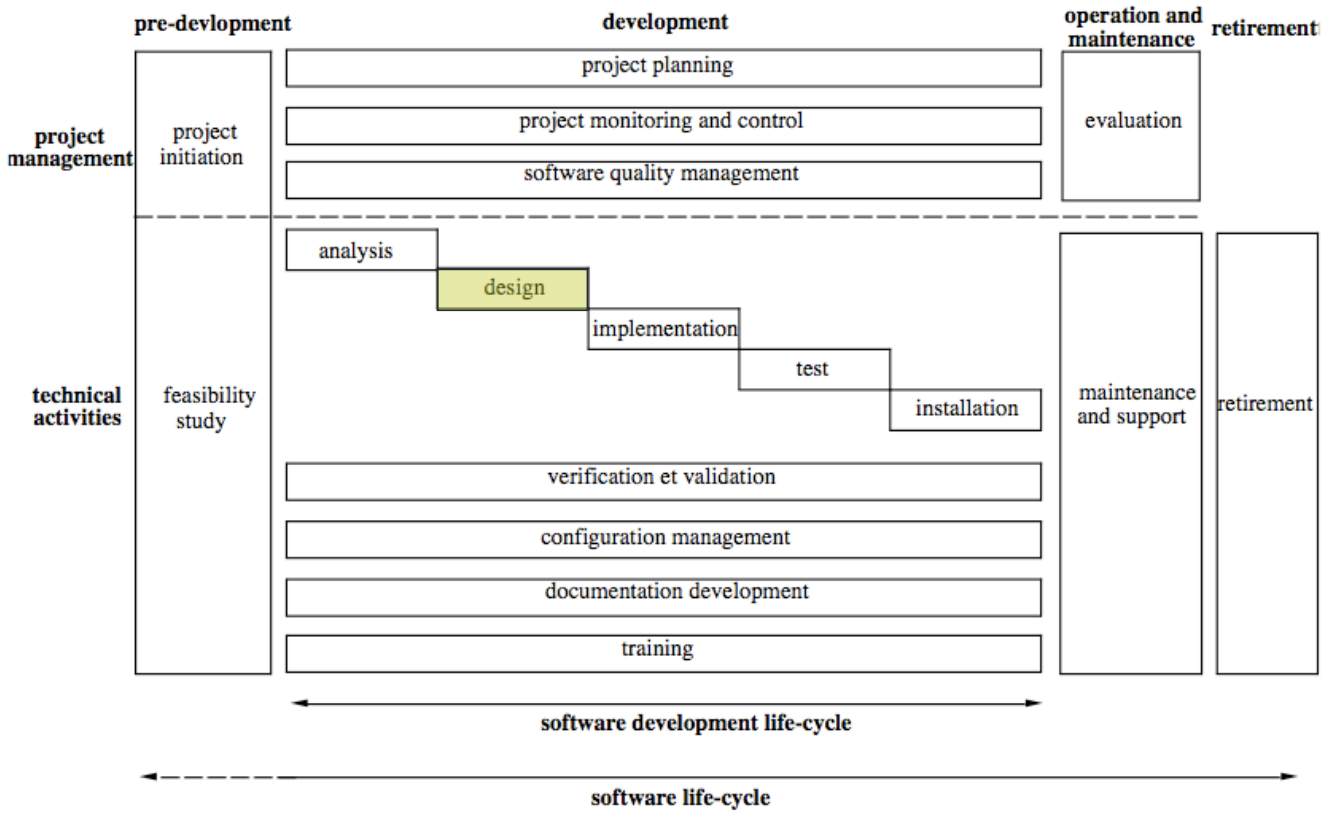
La spécification 4.C.1 est une spécification fonctionnelle. Elle décrit de manière imprécise une fonction (ici une interface virtuelle) qui doit être fournie par le logiciel

La spécification 4.C.8 est une spécification non fonctionnelle

La spécification 4.C.2 est une spécification non vérifiable, difficile à classer, qui n'apporte pas beaucoup d'information

Les spécifications fonctionnelles et non-fonctionnelles ne sont pas clairement séparées

Les activités du génie logiciel [3]



Software Life-Cycle

Conception

Divisée en deux phases successives :

Conception générale, conception globale, conception préliminaire ou conception architecturale

Conception détaillée

Conception

Méthodes de conception :

Conception fonctionnelle descendante

Conception orienté objet

Conception dirigée par les données

Conception générale

La phase de conception générale :

Ébaucher plusieurs variantes de solutions et choisir celle qui offre le meilleur rapport entre coût et avantages

Figer la solution retenue, la décrire et la détailler

L'architecture de la solution est décrite :

Organisation en entité (ou modules)

Interfaces de chaque entité

Interaction entre les entités

Tous les éléments du document de spécification doivent être pris en compte lors du processus de structuration de la solution

Le résultat de cette démarche est un document de conception générale

Conception détaillée

La conception détaillée affine la conception générale par décompositions successives des entités en briques plus élémentaires pour arriver au niveau des composants logiciels (briques de base - faciles à implémenter et à tester)

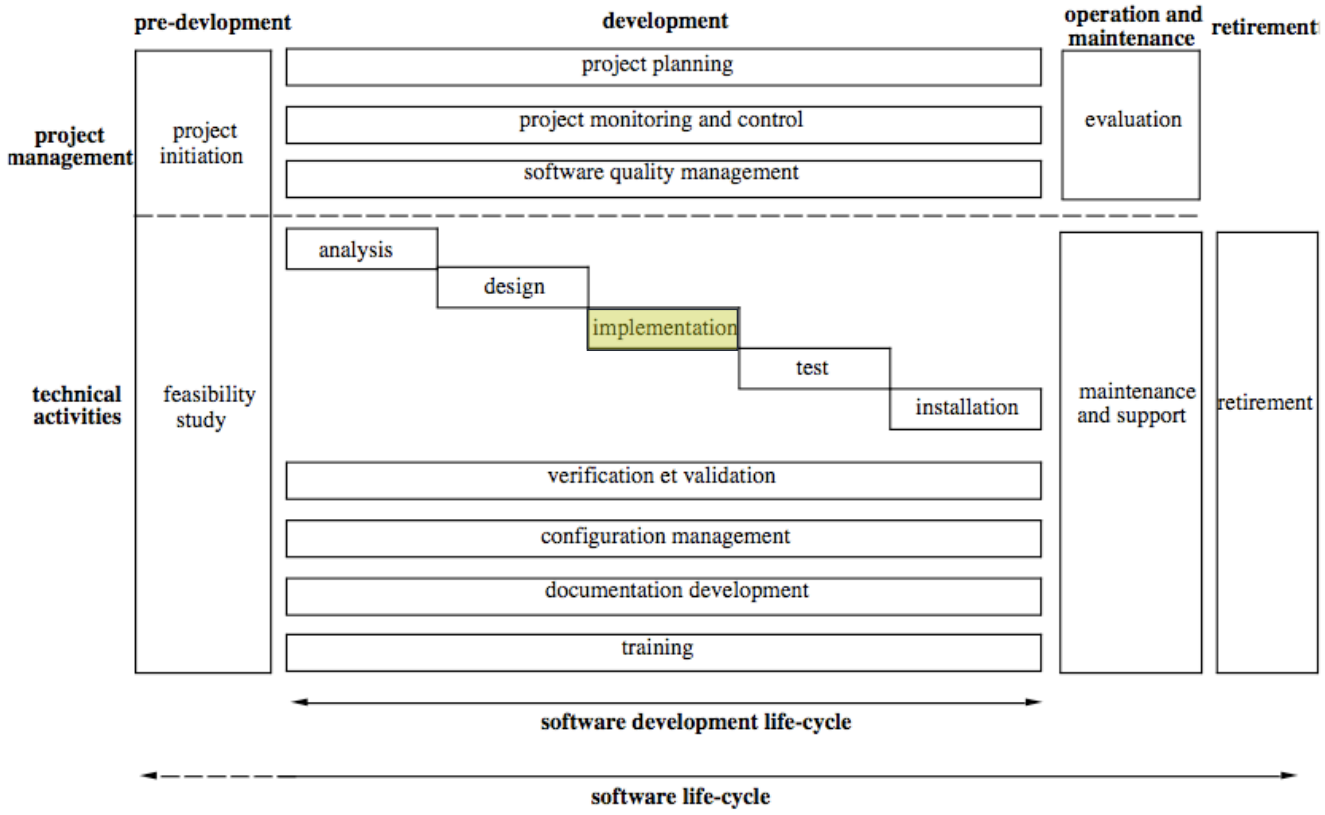
Ce niveau dépend beaucoup du langage de programmation retenu pour l'implémentation

Chaque composant logiciel doit être décrit en détail :

- Interface
- Algorithmes utilisés
- Traitements des erreurs
- Performances
- Etc

L'ensemble de ces descriptions constitue le document de conception détaillé

Les activités du génie logiciel [3]



Software Life-Cycle

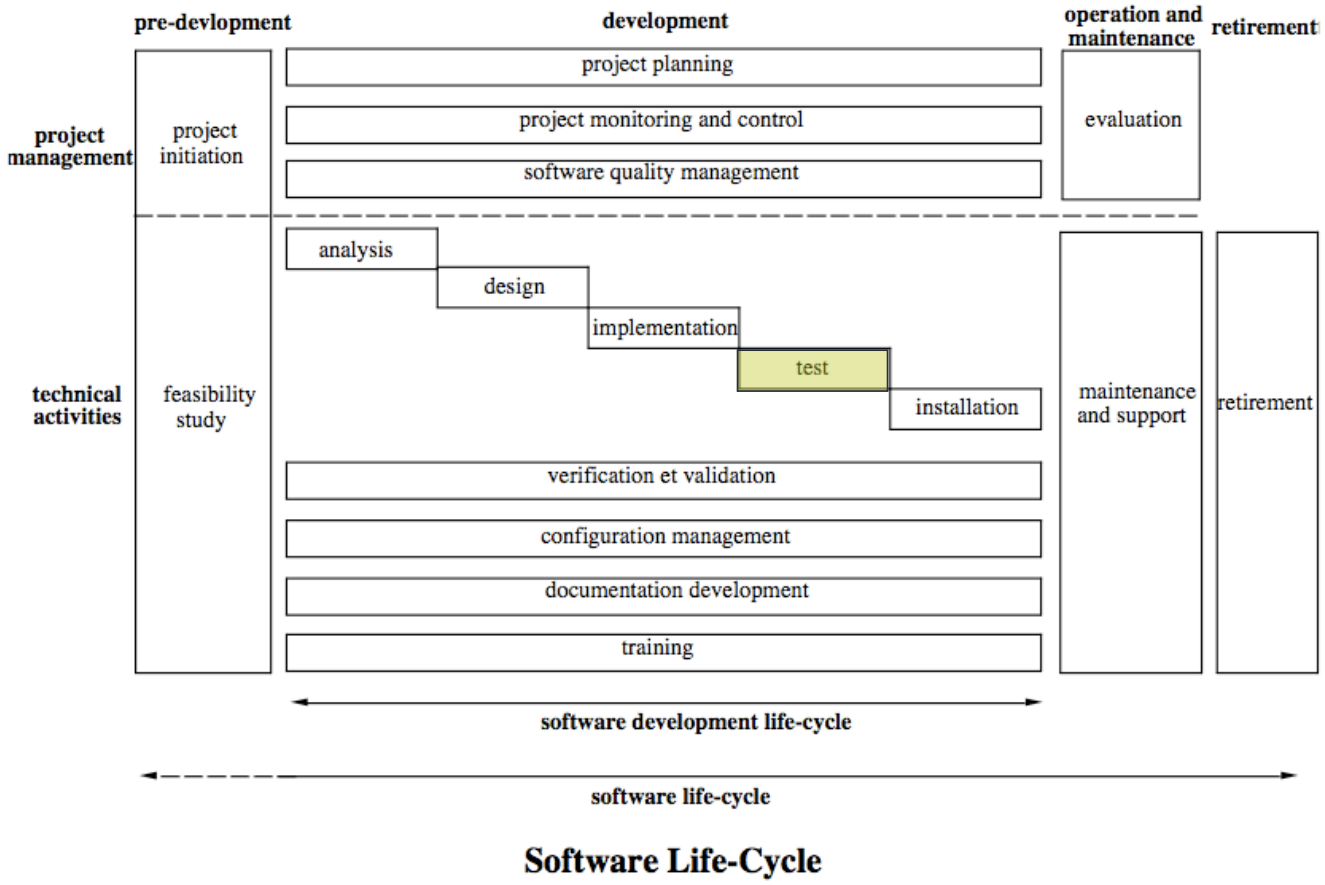
Implémentation

Traduction de la conception détaillée dans un langage de programmation

Son résultat comprend :

- Les sources et objets compilés
- Les comptes-rendus des résultats de compilation
- Les références croisées internes et externes des composants
- La liste des commandes de production du logiciel (ex : options de compilation)
- La description de l'environnement de production

Les activités du génie logiciel [3]



Software Life-Cycle

Tests

Les composants du logiciel sont évalués et intégrés

Le logiciel est évalué pour déterminer si il correspond à la spécification élaborée lors de l'analyse

Les phases de test sont :

- Tests unitaires
- Tests d'intégration
- Tests de réception (pendant la phase d'installation)

Tests unitaires

Évaluation individuelle de chaque composant pour s'assurer qu'il est conforme à la conception détaillée

Un test unitaire comprend :
L'élaboration pour chaque composant d'un jeu de données de tests
L'exécution du composant avec ce jeu
La comparaison des résultats obtenus aux résultats attendus
La consignation de ces résultats dans le document des tests unitaires

En cas d'erreurs :
Le composant est renvoyé à son auteur
L'auteur cherche la cause de l'erreur et la corrige
Le test unitaire est repris

Tests d'intégration

L'intégration consiste à assembler l'ensemble des briques logicielles élémentaires, cet assemblage est testé

On distingue :

La phase d'assemblage

L'exécution des tests pour chaque composant assemblé et l'enregistrement des résultats

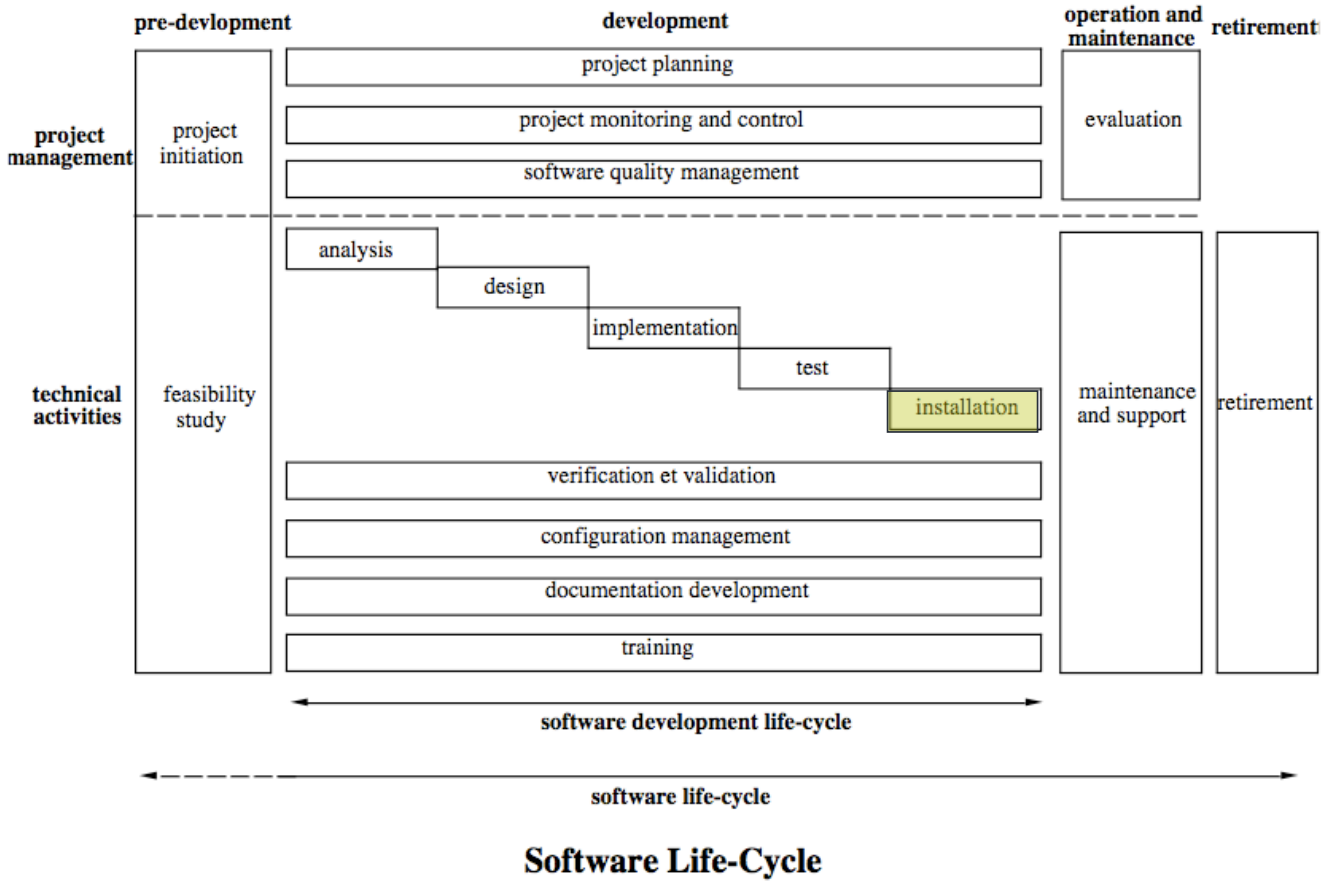
La comparaison des résultats obtenus aux résultats attendus

L'engagement de la procédure de modification si le composant n'est pas conforme

La rédaction des comptes-rendus du test d'intégration si le composant est conforme

L'archivage des sources, objets compilés, jeux de tests et de leurs résultats

Les activités du génie logiciel [3]



Software Life-Cycle

Installation

Planifier l'installation

Distribuer le logiciel

Installer le logiciel dans son environnement d'exploitation

Vérifier le logiciel dans son environnement opérationnel

Mettre hors service tout logiciel existant remplacé

Mettre en place les mises à jour

Documentation [6]

	Avant-projet	Analyse	Conception générale	Conception détaillée	Implémentation	Tests unitaires	Intégration et test d'intégration	Installation et test de réception
Cahier des charges du projet		→						
Spécification			→					→
Conception générale				→			→	
Conception détaillée					→	→		
Listage						→		
Tests unitaires					?			
Test d'intégration								
Test de réception			?					
Manuel d'utilisation et d'exploitation			→ ?					→ ?

Document élaboré entièrement pendant la phase

Document partiellement élaboré pendant la phase

→ Document en entrée de phase

? Document éventuellement complété pendant la phase

Documentation - ex. du manuel utilisateur [9]

1. **Page de titre**
   - (nom, version et date du document, logiciel concerné, organisation réalisatrice)
2. **Restriction**
   - (utilisation, copie, marques déposées)
3. **Garanties et obligations contractuelles**
   - (engagements respectifs de l'organisation et des utilisateurs)
4. **Table des matières**
5. **Table des illustrations**

Documentation - ex. du manuel utilisateur [9]

6. Introduction

1. Généralités: Description des utilisateurs, raisons d'être du document et du logiciel, objectifs principaux du logiciel
2. Identification du logiciel : (nom du logiciel, numéro de version; langue d'utilisation; auteur ou éditeur du logiciel; marque et modèle du matériel ainsi que nom et version du système d'exploitation si plusieurs variantes du logiciel sont disponibles)
3. Environnement du logiciel: (préciser l'environnement d'utilisation du logiciel: environnement matériel, configurations matérielles minimales et conseillées, environnement logiciel, y compris système d'exploitation)
4. Installation du logiciel: (décrire toutes les manipulations nécessaires pour rendre le logiciel opérationnel: connexion de périphériques, chargement de fichiers, modification de paramètres, etc..)
5. Utilisation du manuel: (objectifs, contenus fixe et relatifs de chaque section, conseils d'utilisation)
6. Documents complémentaires: (autres documents aidant l'utilisation du document présent)

Documentation - ex. du manuel utilisateur [9]

7. Guide de l'utilisateur

   1. Généralités: (Présentation générale du domaine fonctionnel dans lequel s'intègre le logiciel et des objectifs qu'il rempli)
   2. Fonctions du logiciel (Présentation synthétique des fonctions principales réalisées par le logiciel et des données utilisées)
   3. Architecture générale de l'interface (Présentation des différentes fenêtres et de leurs organisations)
   4. Exemples d'utilisation: (montrer à l'aide d'exemples simples comment l'utilisateur peut mettre en œuvre les principales fonctions offertes par le logiciel, principes d'interactions)

Documentation - ex. du manuel utilisateur [9]

8. Manuel de référence: (décrire les formats et fonctions des commandes et préciser leurs limites d'utilisation; énumérer les messages affichés en réponse aux différente commandes, préciser leurs significations et décrire les mesures à prendre en réponse à ces messages)

8. Message d'erreur et problèmes connus: (décrire tous les contextes d'erreurs ainsi que les messages qui leur sont liés; indiquer les opérations de reprise pour chaque erreur)

9. Appendice: (informations sur les données fréquemment utilisées, codifications, interactions complexes entre les tâches, limitations de calcul, description des structures de données ou de fichier, fichiers d'exemples, de programmes ou de documentation)

10. Bibliographie
11. Glossaire
12. Index
13. Conception de l'interface

Bibliographie

1. Le génie Logiciel, *Jacques Printz*, Que sais-je?, presses universitaires de France, 1995.
2. Introduction au génie Logiciel, *Guillaume Raschia*, polytech' Nantes, 2004.
3. Software Engineering, Why? What, *Alfred Strohmeier*, École Polytechnique Fédérale de Lausanne, 2000.
4. Guide to the Software Engineering Body of Knowledge, *IEEE Computer Society Professionnal Practices Committee*, 2004.
5. Précis de génie logiciel, *M-C Gaudel et al.*, Masson, 1996.
6. Génie Logiciel : principes, méthodes et techniques, A. Strohmeier et D. Buchs, Presses Polytechniques et universitaires romanes, 1996.
7. Le génie logiciel et ses applications, Ian Sommerville, InterEditions, 1988.
8. *Processus d'ingénierie du logiciel - Méthodes et Qualité, Claude Pinet, Pearson Education, 2002.*
9. Manuel de l'utilisateur, *Didier Buchs*, École Polytechnique Fédérale de Lausanne, 1999.