

Génie logiciel

Notes du cours de 16/12 P n° 2

L3 Informatique appliquée 2022-2023

MABROUK Fayez

1^{er} janvier 2023

1 Documentation

1.1 Introduction

- * Un bon logiciel doit s'accompagner d'une bonne documentation
- * Définition : l'ensemble des documents écrits et du matériel (par exemple, figures, diagrammes) qui accompagne les logiciels informatiques
- * Partie essentielle de l'ingénierie logicielle
- * Objectif principal de la documentation : communication durable avec :
 - * Les parties prenantes
 - * les équipes de développement
 - * les utilisateurs

1.2 Types de documents - Contractuels

- * Cahier des charges (Specification) : décrit les exigences et les contraintes du projet.
- * Contrat
- * Convention : document contractuel suivant une forme convenue
- * Réception : formalisation de la fin d'un projet

1.3 Types de documents - Documents d'appui

- * Présentation du logiciel : présentation générale du logiciel, sans détails.
- * Manuel d'utilisation : décrit les fonctionnalités du logiciel et la manière de les utiliser.
- * Manuel d'installation : décrit les procédures d'installation du manuel.
- * Manuel d'exploitation : procédures de dépannage, de maintenance et de mises à jour

1.4 Types de documents - Gestion de projet

- * Plan de développement : définit les procédures pour réaliser le projet
- * Conception : décrit le résultat de la phase de conception.
- * Rapports de projet : rapports intermédiaires pour documenter le processus de développement.
- * Feedback : vers la fin d'un projet, qu'avons-nous appris ? Qu'est-ce qui peut être réutilisé ?

1.5 Types de documents - Qualité

- * Plan d'assurance qualité : s'il est suivi, il garantit que le produit fini répond à tous les critères de qualité prédéfinis
- * Plan d'audit interne : définit la fréquence et les procédures des audits internes.
- * Plan d'évaluation : à la fin de chaque phase, vérifier que les objectifs initiaux sont atteints.
- * Plan de test : à la fin du projet, valider que les objectifs initiaux sont atteints.

1.6 Rédaction d'un document - Structure générale

- * Identification du document :
 - * Nom du document
 - * Nom du projet
 - * Nom des auteurs / institutions
 - * Date du document
 - * Version du document
- * Qualité du document :
 - * Informations sur la vérification
 - * Informations sur la validation
 - * Informations sur la confidentialité
 - * Mots clés
- * Plan d'ensemble
- * Liste des figures
- * Liste des tableaux
- * Introduction :
 - * Contexte
 - * Objectifs du document
 - * Résumé
 - * Rappel sur les notions nécessaires
 - * Comment lire ce document : guide de lecture pour chaque catégorie de lecteur
- * Le contenu spécifique du document
- * Lexique
- * Références
- * Index

1.7 Cahier des charges

Documentation

Cahier des charges (specification)

- **Definition:** Le cahier des charges est un document rassemblant les obligations et les éléments nécessaires pour définir un besoin et les principales contraintes à respecter pour le satisfaire. (AFITEP)
- **Objective:** allow the client to define its needs and choose the appropriate party to conduct the project
- **European norm:** NF EN 16271 (European norm), 02/2013
- Previously: AFNOR X50-151

Documentation

Cahier des charges – AFNOR X50-151

- Présentation générale du problème:
 - Projet (finalités, volet financier)
 - Contexte:
 - Positionnement par rapport aux autres projets de l'entreprise
 - Etudes préalables
 - Suites prévues
 - Prestations demandées
 - Parties concernées par le déroulement du projet et ses résultats (demandeurs, utilisateurs)
 - Caractère confidentiel du projet
- Enoncé du besoin (quel finalité pour l'utilisateur final)
- Environnement du produit:
 - Liste des éléments de l'environnement du produit (personnes, équipement, matières) et des contraintes
 - Caractéristiques pour chaque élément de l'environnement

Documentation

Cahier des charges – AFNOR X50-151

- Expression fonctionnelle du besoin:
 - Fonctions de service et de contrainte
 - Fonctions de service principales
 - Fonctions de service complémentaires
 - Contraintes
 - Critères d'appréciation
 - Niveaux des critères d'appréciation

Documentation

Cahier des charges – AFNOR X50-151

- Cadre de réponse
 - Pour chaque fonction:
 - Solution proposée
 - Niveau atteint pour chaque critère d'appréciation de cette fonction et modalités de contrôle
 - Part du prix attribué à chaque fonction
 - Pour l'ensemble du produit:
 - Prix de la réalisation de la version de base
 - Options et variantes proposées non retenues au cahier des charges
 - Mesures prises pour respecter les contraintes et leurs conséquences économiques
 - Outils d'installation, de maintenance à prévoir
 - Décomposition en modules
 - Prévisions de fiabilité
 - Perspectives d'évolution technologique

1.8 Plan de développement

- * Définition : document qui présente la stratégie de développement
- * Objectif : fournir une référence sur la planification et l'organisation de l'aménagement du territoire. développement du logiciel

1.9 Plan de développement - sections

- * Introduction :
 - * Objectifs du développement
 - * Méthodologie
 - * Documents connexes
- * Organisation :
 - * Description des tâches
 - * Description du personnel
- * Planification :
 - * Cycle de vie du développement logiciel
 - * Chronologie
 - * Méthodes et choix technologiques
- * Documentation :
 - * Quel document va soutenir le développement
 - * Quelle norme sera suivie
 - * Quels outils de gestion seront utilisés
 - * Qualité : quelle norme de qualité sera utilisée, comment allez-vous évaluer la qualité ?

1.10 Conception

- * Définition : document qui présente les résultats de la phase de conception, avec une description complète et détaillée des composants du logiciel.
- * Objectif : référence pour les développeurs et la direction.

1.11 Conception - sections

Documentation

Conception - sections

- Introduction:
 - Objectives of the development
 - Methodology
 - Related documents
- Interfaces between the software's tasks
- Detailed description:
 - For each module:
 - objectives
 - relation with other modules
 - types definition
 - description of the functions and variables that are visible to users of the module
 - For each class:
 - Complete description of the class
 - Relations to other classes
 - Attributes and methods

1.12 Plan de test

- * Définition : décrit la procédure de test complète permettant de vérifier le logiciel dans son intégralité et chaque composant individuellement
- * Objectif : décrire les tests unitaires (pour tester chaque composant) et les tests d'intégration (pour tester le logiciel complet) avant la phase d'implémentation.

Documentation

Test plan - sections

- Introduction:
 - Objectives of the development
 - Methodology
 - Related documents
- Description of the unit tests: for each class/module, for each function/method:
 - Description of the test: how to run, what is covered
 - for each test: which data, what result is expected, how to validate
- Description of the integration tests: for each test:
 - Description of the test: from the external point of view
 - Description of the objective: what does it cover?
 - Testing procedure: which data, what result is expected, how to validate
- Scenarios: propose a set of tests to validate a set of functions, and how do they relate to each others

1.13 Acceptance plan (cahier de recette)

- * Définition : décrit les différents aspects de la livraison de logiciels, notamment les tests d'acceptation
- * Définition (tests d'acceptation) : Test formel par rapport aux besoins des utilisateurs, des utilisateurs, des exigences et des processus d'affaires, afin de déterminer si un système satisfait aux critères d'acceptation et pour permettre à l'utilisateur, aux clients ou à une autre entité autorisée de déterminer s'il faut accepter le système.
- * Objectif : définir contractuellement la manière de valider que le logiciel répond aux besoins et exigences prédéfinis.

Documentation

Acceptance plan - sections

- Introduction:
 - Objectives of the development
 - Methodology
 - Related documents
- Deliverables description
- Description of the necessary means and tools to validate the software
- Conformity to general specifications
- Conformity to functional specifications: description of scenarios. For each scenarios, provide the description and the testing procedure
- Conformity to interfaces specifications
- Conformity of the documentation

1.14 Manuel d'installation

- * Définition : le manuel d'installation est un document qui rassemble toutes les procédures nécessaires pour installer le logiciel dans son environnement de production.
- * Objectif : permettre à l'administrateur système d'installer et de configurer le logiciel sur le système d'information visé.

1.15 Manuel d'installation - sections

- * Installation du matériel : quel matériel doit être installé, quelle procédure sont nécessaires pour la production
- * Configuration du système : quels paramètres doivent être appliqués pour correctement configurer le système
- * Installation du logiciel : quelle est la procédure à suivre pour installer le logiciel sur le système
- * Configuration du logiciel : quels sont les paramètres à appliquer pour configurer le logiciel
- * Données : quelles sont les procédures à suivre pour configurer les données du logiciel ?
- * Autres informations : conflits éventuels avec d'autres parties du système, mode de maintenance

1.16 Manuel d'utilisation

- * Définition : le manuel d'utilisation est un document décrivant les fonctionnalités du logiciel et les informations sur la façon de les utiliser.
- * Objectif : permettre à l'utilisateur final du logiciel d'utiliser toutes les fonctionnalités du logiciel.

1.17 Manuel de l'utilisateur - sections

- * Opérations de base : comment utiliser le logiciel pour utiliser les différentes fonctions fournies par
- * Dépannage : description des exceptions/codes d'erreur, liste des problèmes éventuels et problèmes possibles et comment les résoudre

1.18 Conseils généraux sur la façon de rédiger des documents

- Be precise
- Write short paragraphs
- Use short sentences
- Choose the right format
- Use illustrations when relevant
- Use language checking tools
- Provide references when they are useful
- Use collaborative features of your editor
- Consider LaTeX (with a versioning system)

1.19 Conclusion

- Do not neglect documentation!
- A good documentation helps:
 - the planification
 - the design
 - the development
 - the delivery
 - the usage
- Pay attention to the format

2 Génie logiciel

2.1 Développement logiciel agile

- In 2001, 17 engineers published the **Manifesto for Agile Software Development**.
- Core values:
 - Individuals and interactions over processes and tools
 - Working software over comprehensive documentation
 - Customer collaboration over contract negotiation
 - Responding to change over following a plan
- Aims at lightweight methodologies

Le manifeste présente 12 principes :

- Notre priorité absolue est de satisfaire le client par la livraison rapide et continue de logiciels de qualité.
- Accueillir l'évolution des besoins, même à un stade avancé du développement. Les processus agiles exploitent le changement pour l'avantage concurrentiel du client.
- Livrer fréquemment des logiciels fonctionnels, de quelques semaines à quelques mois, avec une préférence pour les délais les plus courts.
- Les responsables commerciaux et les développeurs doivent collaborer quotidiennement tout au long du projet.
- Construisez des projets autour de personnes motivées. Donnez-leur l'environnement et le soutien dont ils ont besoin, et faites-leur confiance pour faire le travail.
- La méthode la plus efficace pour transmettre des informations à une équipe de développement et au sein de celle-ci est la conversation en face à face.
- Le logiciel fonctionnel est la principale mesure du progrès.
- Les processus agiles favorisent le développement durable. Les commanditaires, les développeurs et les utilisateurs doivent être en mesure de maintenir indéfiniment un rythme constant.
- Une attention constante à l'excellence technique et à une bonne conception renforce l'agilité.
- La simplicité - l'art de maximiser la quantité de travail non effectué - est essentielle.
- Les meilleures architectures, exigences et conceptions émergent d'équipes auto-organisées.
- À intervalles réguliers, l'équipe réfléchit à la manière de devenir plus efficace, puis adapte et ajuste son comportement en conséquence.