Implémentation

MASTER I
AGITEL FORMATION

FORMATEUR : THIERRY MPOULI

Implémentation

- Chacun des modules décrit dans le document de spécification détaillé doit être réalisé. Cela comprend la petite activité de codage ou de programmation qui constitue le cœur et l'âme du processus de développement du logiciel. Il est malheureux que cette petite activité soit quelquefois l'unique partie du génie logiciel qui soit enseignée (ou étudiée), puisque c'est également la seule partie du génie logiciel qu'un autodidacte peut réellement appréhender.
- On peut considérer qu'un module a été réalisé quand il a été créé, testé et utilisé avec succès par un autre module (ou par un processus de test au niveau système). La création d'un module se fait dans le cadre du cycle classique édition-compilation-répétition. Le test des modules comprend les tests au niveau unitaire les tests de non-régression définis lors de la conception détaillée, ainsi que les tests de performances et de charge et l'analyse de couverture du code.

Implémentation

- La méthodologie
- La lisibilité des programmes
- La portabilité des programmes
- Les outils
- Les langages

Méthodologie

- Passage de l’analyse à la programmation
- Méthodologie ascendante
- Méthodologie descendante

Lisibilité des programmes

- Dépend du langage et du style d'écriture
- Choix des noms
- Mise en page des programmes
- Bonnes structures de contrôle (boucle, condition, …)

Portabilité des programmes

- Compilation sur la machine cible
- Dépendance due à l'architecture machine
- Dépendance due au système d’exploitation

Les outils

- Outils de préparation des programmes
  - Éditeurs
  - Outils de traduction
  - Compilateurs : bons et moins bons
- Outils d'analyses
  - Références croisées
  - Mise en forme du source
  - Liste de partie de programme
- Outils de gestion
  - Traces du développement
  - Suivi de la cohérence des versions (SCCS, MAKE, RCS, CVS Concurrent Versions System)

Les environnements de programmation

- Logiciels de communication entre machine développement et machine cible
- Simulateurs de machine cible
- Outils de tests et mise au point (bancs de test, analyseurs de programmes)
- Traitements de texte
- Outils de spécifications fonctionnelles
- Outils de graphiques de description
- Outils de gestion de projets : génération de rapports d'avancement du projet

Les langages

- Les langages d'assemblage : processeur
- Les langages de réalisation de systèmes ©
- Les langages statiques de haut niveau (COBOL, Visual Basic)
- Les langages de haut niveau à structure de blocs (Ada)
- Les langages dynamiques de haut niveau (Prolog)
- Les langages objets (C++, Java)
- Les langages spécialisés (Perl, SQL, HTML, XHTML, PHP)

Intégration

- Quand tous les modules sont terminés, l'intégration, au niveau du système, peut être réalisée. C'est là que tous les modules sont réunis en un seul ensemble de code source, compilés et liés pour former un paquetage qui constitue le système. L'intégration peut être réalisée de façon incrémentale, en parallèle avec la réalisation de différents modules, mais on ne peut pas décider de manière autoritaire que « c'est fini » tant que tous les modules ne sont pas effectivement terminés.
- L'intégration comprend le développement de tests au niveau du système. Si le paquetage réalisé est capable de s'installer lui-même (ce qui signifie simplement le décompactage d'une archive ou la copie de fichiers d'un CD-ROM) il doit alors exister un moyen de le faire automatiquement, soit sur des systèmes spécialisés, soit dans des environnements de simulation.
- Parfois, dans le cas des logiciels personnalisés, le paquetage est constitué du simple exécutable résultant de la compilation de l'ensemble des modules, et, dans ce cas, il n'y a pas d'outil d'installation ; les tests seront effectués tels quels.
- Le système ayant été installé (s'il doit l'être), le processus de tests au niveau système doit pouvoir lancer toutes les commandes publiques et appeler tous les points d'entrée publics, en utilisant toutes les combinaisons raisonnables d'arguments. Si le système doit pouvoir créer une sorte de base de données, la procédure automatisée de test au niveau système doit en créer une et utiliser des outils extérieurs (écrits séparément) pour vérifier l'intégrité de la base de données. Les tests unitaires peuvent être utilisés pour répondre à certains de ces besoins, et tous les tests unitaires doivent être exécutés en séquence pendant le processus d'intégration, de construction et de réalisation du paquetage.

Maintenance et assistance

- Les défauts du logiciel rencontrés soit pendant la phase de test *in situ* soit après sa diffusion doivent être enregistrés dans un système de suivi. Il faudra affecter un ingénieur logiciel pour la prise en charge de ces défauts, qui proposera de modifier soit la documentation du système, soit la définition d'un module ou la réalisation de ce module. Ces modifications devront entraîner l'ajout de tests unitaires ou au niveau système, sous forme de tests de non-régression pour mettre en évidence le défaut et montrer qu'il a bien été corrigé (et pour éviter de le voir réapparaître plus tard).
- La maintenance est une entreprise commune aux ingénieurs et au service client. La liste des bogues, la description de bogues particuliers, le nombre maximum de défauts critiques dans une version diffusée du logiciel,... constituent les pièces maîtresses de cette édifice.

Maintenance

- Deux techniques de maintenance
  - Correction des erreurs du système
  - Demande d'évolution (modification de l'environnement technique, nouvelle fonctionnalité)
- Facteurs de qualité essentiels
  - Corrections : robustesse
  - Evolutions : modifiabilité, portabilité
- Etape longue, critique et coûteuse
  - 80 % de l'effort de certaines entreprises (Pb de pratiques ?)