



Introduction Au Genie Logiciel

PAR RONEL TCHOULAYEU

Definition

- ▶ **Genie Logiciel:** Le terme est composé de deux mots, logiciel et ingénierie.
- ▶ **Logiciel:** ensemble de codes de programmation exécutables, de bibliothèques et de documentations associées.
- ▶ **Ingénierie:** consiste à développer des produits, en utilisant des principes et des méthodes scientifiques bien définis.

Pour résumer

Le génie logiciel est une branche d'ingénierie associée au développement de produits logiciels utilisant des principes, méthodes et procédures scientifiques bien définis.

Definition



Les Etapes de la creation d'un logiciel

Types de logiciels

- ▶ **Logiciel Systeme:** collection de programmes pour fournir des services à d'autres programmes : compilateurs, pilotes, systemes d'exploitations
- ▶ **Logiciel temps reel:** utilisé pour surveiller, contrôler et analyser les événements du monde réel au fur et à mesure qu'ils se produisent: Meteo
- ▶ **Logiciel embarqué :** Ce type de logiciel est placé dans la « mémoire morte (ROM) » du produit et contrôle les différentes fonctions du produit: avion, automobile, un système de sécurité, centrales
- ▶ **Logiciels Web :** les logiciels liés aux applications Web entrent dans cette catégorie. Les exemples sont CGI, HTML, Java, Perl, DHTML, etc.

Types de logiciels

- ▶ **Logiciel d'intelligence artificielle** : ils utilisent des algorithmes non numériques pour résoudre des problèmes complexes. Les exemples sont les réseaux neuronaux artificiels, les logiciels de traitement du signal, vision par ordinateur.
- ▶ **Logiciels pour ordinateurs personnels** : Les exemples sont les traitements de texte, l'infographie, les outils multimédias et d'animation, la gestion de bases de données, les jeux informatiques, etc.
- ▶ **Logiciels métiers** : le système de surveillance des fichiers, la gestion des employés, la gestion des comptes ,Le système d'information de gestion.



Mythes sur les logiciels

- ▶ Les mythes sur les logiciels propagent de fausses croyances et une certaine confusion dans l'esprit des dirigeants, des utilisateurs et des développeurs.
- ▶ Ce qui entraîne le plus souvent des problèmes lors de la phase de développement.

Mythes sur les logiciels

Cote management:

- ▶ **Si le projet est en retard, augmenter le nombre de programmeurs peut réduire l'écart de temps.**
- ▶ L'ajout de main d'œuvre supplémentaire au projet, qui est déjà en retard sur le calendrier, retarde encore davantage le projet. □ Les nouveaux travailleurs mettent plus de temps à se renseigner sur le projet que ceux qui travaillent déjà sur le projet.
- ▶ **Si le projet est sous-traité à un tiers, la direction peut se détendre et laisser l'autre entreprise développer des logiciels pour elle.**
- ▶ L'externalisation des logiciels à un tiers n'aide pas l'organisation, qui est incompetente dans la gestion et le contrôle du projet logiciel en interne. L'organisation souffre invariablement lorsqu'elle sous-traite le projet logiciel.

Mythes sur les logiciels

Cote Utilisateur

- ▶ **Une brève exigence énoncée dans le processus initial suffit pour commencer le développement ; des exigences détaillées peuvent être ajoutées ultérieurement.**
- ▶ Commencer le développement avec des exigences incomplètes et ambiguës conduit souvent à une panne logicielle. Au lieu de cela, une description complète et formelle des exigences est essentielle avant de commencer le développement.
- ▶ **Le logiciel est flexible ; par conséquent, des modifications des exigences logicielles peuvent être ajoutées à n'importe quelle phase du processus de développement.**
- ▶ L'intégration des demandes de changement plus tôt dans le processus de développement coûte moins cher que celles qui se produisent à des étapes ultérieures.

Mythes sur les logiciels

Cote Developpeur

- ▶ **Le développement du logiciel est considéré comme terminé lorsque le code est livré.**
- ▶ 50 à 70 % de tous les efforts sont déployés après la livraison du logiciel à l'utilisateur.
- ▶ **Le succès d'un projet logiciel dépend de la qualité du produit réalisé.**
- ▶ La qualité des programmes n'est pas le seul facteur de réussite du projet ; la documentation et la configuration du logiciel jouent également un rôle crucial.
- ▶ **La qualité du logiciel ne peut être évaluée qu'après l'exécution du programme.**
- ▶ Elle peut être mesurée à n'importe quelle phase du processus de développement en appliquant un mécanisme d'assurance qualité. L'un de ces mécanismes est l'examen technique formel qui peut être utilisé efficacement à chaque phase de développement pour découvrir certaines erreurs.

Le problème logiciel : Coût, calendrier et qualité, Échelle et changement

- ▶ Cout: Les logiciels de qualité industrielle sont très chers, principalement en raison du fait que le développement de logiciels demande extrêmement de main d'œuvre.
- ▶ Calendrier: réduire le coût et la durée du cycle de développement de logiciels sont des objectifs centraux du génie logiciel

si la productivité est plus élevée, le potentiel de développement du logiciel en moins de temps s'améliore .

- ▶ Qualité: La norme internationale sur la qualité des produits logiciels [55] suggère que la qualité des logiciels comprend six attributs principaux:

Fonctionnalité, Fiabilité, Convivialité, Efficacité, Maintenabilité, Portabilité

Le problème logiciel : Coût, calendrier et qualité, Échelle et changement

- ▶ **Fonctionnalité.** La capacité de fournir des fonctions qui répondent aux besoins exprimés et implicites lorsque le logiciel est utilisé.
- ▶ **Fiabilité.** La capacité de fournir un service sans panne.
- ▶ **Utilisabilité.** La capacité d'être compris, appris et utilisé.
- ▶ **Efficacité.** La capacité à fournir des performances appropriées par rapport à la quantité de ressources utilisées.
- ▶ **Maintenabilité.** La capacité d'être modifié à des fins de correction, d'amélioration ou d'adaptation.
- ▶ **Portabilité.** La capacité de s'adapter à différents environnements spécifiés sans appliquer d'actions ou de moyens autres que ceux prévus à cet effet dans le produit.

7 Principes du génie logiciel

- ▶ (1) Gérer à l'aide d'un plan de cycle de vie par étapes.
- ▶ (2) Effectuer une validation continue.
- ▶ (3) Maintenir un contrôle discipliné des produits.
- ▶ (4) Utiliser des pratiques de programmation modernes.
- ▶ (5) Maintenir une responsabilité claire en matière de résultats.
- ▶ (6) Utiliser mieux et moins de personnes.
- ▶ (7) Maintenir un engagement à améliorer le processus.

FIN

