

**Q2/ Quand le terme « Génie Logiciel » a été employé pour la première fois et dans quel contexte ?**

- Le terme « Génie Logiciel » a été employé pour la première fois par à la conférence de L'OTAN concernant la *crise du logiciel* en 1968 bien que la notion elle-même est due à la conceptrice du système embarqué du programme Apollo; l'informaticienne et mathématicienne Margaret Hamilton.

« *Her Code Got Humans on the Moon—And Invented Software Itself* », Robert McMillan.

**Q3 /**

- **Logiciel :**
  - Ensemble composé d'un ou plusieurs programmes, ainsi que les fichiers nécessaires pour les rendre opérationnels. *Wikipédia*
  - Ensemble des programmes, procédés et règles, et éventuellement de la documentation, relatifs au fonctionnement d'un ensemble de traitement de données. *Dictionnaire Larousse*
- **Progiciel :** Programme destiné à un même type d'applications et conçu pour différents utilisateurs. (générique vs spécifiques).
- **G.L :**
  - Est un domaine des sciences de l'ingénierie dont l'objet d'étude est la conception, la fabrication, et la maintenance des systèmes informatique complexe.
  - Est l'ensemble des activités de conception et de mise en œuvre des produits et des procédures tendant à rationaliser la production du logiciel et son suivi.  
Source : JO 19 février 84 France
- **AGL (CASE) :** Atelier de génie logiciel, CASE (*Computer Aided Software Engineering*),
  - désigne un ensemble de programmes permettant eux-mêmes de produire des programmes. Les outils AGL couvrent un champ au-delà des environnements de développement intégrés où certains AGL peuvent aller jusqu'à la génération de code et éventuellement faciliter la génération de jeux de test comme ils permettent de documenter automatiquement un programme.
    - Ensemble cohérent d'outils formant un environnement d'aide à la conception, au développement et à la mise au point de logiciels d'application spécialisés. Dans un

AGL on peut trouver des dictionnaires de données, des outils permettant de réaliser des diagrammes, pour faciliter la phase d'analyse et de conception des applications.

Puis des générateurs de code ainsi que des aides à la mise au point (débugueurs) viendront accélérer la production et la finalisation de l'application.

*journaldunet*

- **UML** : Langage de Modélisation Unifié.
- **RUP** (Rational Unified Process) : Méthode générique, itérative et incrémentale basé sur UML.
- **XP** (eXtrem Programming) : Méthode agile plus particulièrement orientée sur l'aspect réalisation d'une application. Elle implique au maximum le client et permettent une grande réactivité à ses demandes. Elle repose sur des valeurs fondamentales : la communication, simplicité, Feedback, ... et sur un ensemble de pratiques (conception simple, intégration continue, petite livraison, client sur site, ...etc.).

#### **Q7/ Comment peut-on améliorer les facteurs de qualité suivants :**

##### **Correction (utilité)**

- Se pencher rigoureusement sur l'analyse des besoins.
- Améliorer la communication (langage commun, démarche participative).
- Travailler avec rigueur.

##### **Efficacité**

- Logiciels plus simples
- Veiller à la complexité des algorithmes
- Machines plus performantes

##### **Facilité d'utilisation**

- Analyse du mode opératoire des utilisateurs
- IHM : Adapter l'ergonomie des logiciels aux utilisateurs

#### **Q9/ Types de maintenance**

- **Maintenance corrective** : correction des erreurs 20 % / 17%  
Critère important de la qualité qui corrige les anomalies ou erreurs mises à jour par le client et non pas lors des tests de vérification et de validation.

- **Maintenance adaptative** : Adaptation : 25% / 18%  
Ajout de petites fonctionnalités qui ne modifie pas l'architecture. Ex : Mise à l'euro ; passage de données par fichiers, changement de SGBD, de machine, de taux de TVA, an 2000,
- **Maintenance perfective (évolutive)** : 55% / 65% augmentation des fonctionnalités au fur et à mesure des besoins. Elle consiste à maintenir les fonctionnalités antérieures tout en ajoutant des nouvelles fonctionnalités qui modifient profondément l'architecture (donne lieu à de nouvelles versions). Ex : changement de OS ; changement de SGBD...

**Q10/ Donnez les principaux modèles d'un C.V.L ensuite citez les avantages et les inconvénients de chacun.**

Modèle	Caractéristiques	Avantage	Inconvénient
<p><b>En cascade</b> (sans ou avec retour arrière).</p> <p><b>En V</b></p>	Phases successives	<ul style="list-style-type: none"> <li>• Simple à mettre en place</li> <li>• Permet de valider un budget au début du projet.</li> <li>• Force la documentation : une phase ne peut se terminer avant qu'un document soit validé.</li> <li>• Le test est inhérent (nécessaire et inséparable) à chaque phase</li> <li>• Les progrès sont tangibles (pour l'équipe de développement).</li> <li>• Bien adapté lorsque les besoins sont clairement identifiés et stables.</li> </ul>	<ul style="list-style-type: none"> <li>• Ne prend pas en compte le changement de besoins ou de spécification.</li> </ul>

❖ **NB** : Les autres questions sont des questions directe où vous trouverez facilement leurs réponses dans le cours.