

Documentation

1-Définition De La Documentation :

La documentation logicielle est un texte écrit qui accompagne le logiciel informatique. Elle explique comment le logiciel fonctionne, et/ou comment on doit l'employer. Le terme peut avoir des significations différentes pour des personnes de différents profils.

La Documentation constitue une partie importante de l'ingénierie logicielle, qui est trop souvent négligée

2-Types De Documentation :

- Documentation sur l'architecture et la conception :

La documentation sur l'architecture est un type spécial de documents de conception. D'une certaine façon, les documents sur l'architecture sont les troisièmes dérivés du code (les documents de conception étant les seconds dérivés, et documents sur le code étant le premier). Il y a très peu de chose dans les documents sur l'architecture qui soit spécifique au code lui-même. Ces documents ne décrivent pas comment programmer une fonction (routine) particulière, ou même pourquoi cette fonction particulière existe sous cette forme, mais expose les exigences générales qui motivent l'existence d'une telle fonction.

Un bon document d'architecture est court sur les détails mais dense sur l'explication. Il peut suggérer des approches pour des conceptions de plus bas niveau, mais laisse les études d'exploration effectives à d'autres documents.

Une autre sorte de documents de conception est le document de comparaison. Cela peut prendre souvent la forme d'un *livre blanc*. Il se concentre sur un aspect spécifique du système et suggère des approches alternatives. Cela peut se situer au niveau de l'interface utilisateur, du code, de la conception, ou même au niveau de l'architecture.

Il soulignera la situation du "SI" (système d'information), décrira une ou plusieurs alternatives en présentant leurs avantages et inconvénients. Une bonne étude comparative est lourde en recherche, exprime ses idées clairement (sans se reposer massivement sur un jargon abscons pour aveugler le lecteur), et surtout est impartiale. Il doit expliquer honnêtement et clairement les coûts de toute solution en regard de ce qu'elle apporte de mieux. L'objectif d'une étude comparative est de discerner la meilleure solution, plutôt que de pousser à un point de vue particulier. Il est parfaitement acceptable de ne pas établir une conclusion, ou de conclure qu'aucune des alternatives n'offre d'avantage substantiel par rapport à la situation actuelle pour justifier un changement. Elle doit être conçue comme une initiative scientifique, pas comme une technique marketing.

- Documentation technique :

La documentation associée au logiciel d'exploitation, qui fournit à l'exploitant les instructions d'utilisation du matériel informatique (ordinateur) .

La documentation associée aux logiciels d'application (grandes fonctions de l'entreprise : finances, achats, logistique,...), qui indique comment utiliser le logiciel.

La plupart des programmeurs emploient l'expression " documentation logicielle " dans le cas d'une documentation sur les *logiciels d'application*. Lorsqu'ils développent du logiciel, le code source est

insuffisant à lui seul. Il doit y avoir du texte qui l'accompagne pour décrire les différents aspects du fonctionnement attendu.

Cette documentation est habituellement incluse dans le code source lui-même de telle sorte qu'elle soit facilement accessible à quiconque qui serait amené à le traverser.

Ce document écrit peut être hautement technique, et il est principalement utilisé pour définir et expliquer les interfaces de programmation (APIs), les structures de données et les algorithmes. Par exemple, on peut utiliser cette documentation pour expliquer que la variable `m_name` se réfère au premier et au dernier nom d'une personne. Il est important pour les documents sur le code d'être précis, mais pas non plus verbeux à un point tel qu'il serait difficile de le maintenir.

Souvent, des outils, les générateurs de documentation comme Doxygen ou javadoc peuvent être utilisés pour générer automatiquement le document sur le code ; ils extraient le commentaire du code source et créent des manuels de référence sous des formes comme le texte ou des fichiers HTML. Les documents sur le code sont souvent organisés dans le style d'un guide de référence, ce qui permet à un programmeur de localiser rapidement une fonction ou une classe quelconque.

Beaucoup de programmeurs aiment réellement l'idée de la documentation générée automatiquement pour diverses raisons. Par exemple, parce qu'elle est extraite du code source lui-même (par exemple, à travers les commentaires).

Le programmeur peut l'écrire en se référant à son code, et peut utiliser les mêmes outils que ceux qu'il a utilisés pour développer le code source, pour faire la documentation. Cela rend beaucoup plus facile la mise à jour de la documentation.

Bien sûr, l'inconvénient est que seuls les programmeurs peuvent éditer cette sorte de documentation, et c'est d'eux que dépend la mise à jour des sorties (par exemple, en exécutant un `crontab` pour mettre à jour les documents la nuit). Certains pourraient caractériser cela comme un avantage plutôt que comme un inconvénient.

Donald Knuth a insisté sur le fait que la documentation peut être un processus très difficile de réflexion après coup et a recommandé la programmation lettrée, qui consiste à écrire la documentation en même temps et en un même lieu que le code source et à l'extraire par des moyens automatiques.

- Documentation utilisateur :

À la différence de la documentation sur le code, les documents utilisateurs sont généralement assez éloignés du code source du programme, et décrivent simplement comment il est employé.

Dans le cas d'une bibliothèque logicielle, les documents sur le code et les documents utilisateurs pourraient effectivement être couplés et cela vaut la peine de les regrouper, mais cela n'est pas toujours valable pour les applications en général.

La machine Lisp a suivi la tradition selon laquelle chaque élément de code avait un champ de documentation attaché. En relation avec les fortes capacités de recherche (basées sur une commande appropriée assimilée à Unix), et des sources en ligne, les utilisateurs de Lisp pouvaient consulter la documentation et reprendre la fonction associée directement dans leur propre code. Ce niveau de facilité est inconnu de systèmes présumés plus modernes.

Typiquement, la documentation utilisateurs décrit chaque caractéristique du programme, et les différentes étapes nécessaires pour l'appeler. Un bon document utilisateur peut aussi aller jusqu'à

fournir une assistance minutieuse en ligne. Il est très important que les documents utilisateurs ne soient pas confus, et qu'ils soient à jour. Les documents utilisateurs n'ont pas besoin d'être structurés d'une façon particulière, mais il est très important qu'ils aient un index précis. La cohérence et la simplicité sont aussi deux qualités très précieuses. On considère que la documentation utilisateur constitue un contrat qui spécifie ce que le logiciel doit faire. Voir aussi Technical Writing.

Il y a trois grandes manières d'organiser la documentation utilisateur :

➤ **Tutoriel**

On considère qu'une approche par tutoriel est la plus utile pour un nouvel utilisateur. Dans cette méthode l'utilisateur est guidé à chaque étape d'accomplissement des tâches particulières.

➤ **Thématique**

Pour un utilisateur intermédiaire, on emploie généralement une approche thématique, dans laquelle les chapitres ou sections se concentrent sur un domaine d'intérêt particulier.

➤ **Liste**

Le type final de principe d'organisation est celui dans lequel les commandes ou les tâches sont simplement listées par ordre alphabétique, souvent via des indices croisés. Cette dernière approche est d'un intérêt très élevé pour des utilisateurs avancés qui connaissent exactement quelle sorte d'information ils recherchent. Un grief universellement exprimé par les utilisateurs au sujet de la documentation logicielle est qu'elle n'adopte que l'une de ces trois approches à l'exclusion des deux autres.

Dans le cas des micro-ordinateurs, il est fréquent de limiter la fourniture de documentation logicielle à l'aide en ligne, qui se limite à des informations de référence sur les commandes ou les lignes de menu. Le travail d'enseignement de nouveaux utilisateurs ou d'aide à des utilisateurs plus expérimentés à tirer le meilleur parti d'un programme est laissé à des publicateurs privés, à qui le développeur du logiciel donne une assistance significative.

Documentation marketing :

Pour beaucoup d'applications, il est nécessaire de disposer de matériaux promotionnels pour inciter des observateurs occasionnels à passer plus de temps à s'intéresser au produit.

Cette forme de documentation a trois objectifs :

- ❖ Inciter les utilisateurs potentiels à s'intéresser au produit et installer le désir de s'impliquer davantage dans le produit.
- ❖ Informer l'utilisateur sur ce que fait exactement le produit, de telle sorte que leurs attentes soient en phase avec ce qu'ils vont recevoir.
- ❖ Expliquer la position de ce produit par rapport à d'autres alternatives.
- ❖ Une bonne technique de marketing est de fournir des phrases d'accroche claires et mémorisables qui illustrent le point que l'on souhaite transmettre, et aussi mettre l'accent sur l'interopérabilité du programme avec d'autres produits.