

Comment concevoir une solution informatique

(voir l'exemple «Comment concevoir une solution informatique – Exemple»)

1. Bien **analyser** et **comprendre** le problème (pas la solution !)
2. Réaliser **à la main sur papier**, les étapes nécessaires à trouver le résultat demandé
3. **Analyser** les étapes réalisées à la main
4. Décrire en **mots-clés** une **démarche** à appliquer à n'importe quel cas pour atteindre le résultat demandé
5. Concevoir une solution sous forme d'**algorithme** (**structogramme GNS** ou **JSP**, **pseudo-code**, etc.)
6. On peut **affiner** ensuite le structogramme pour tenir compte des **spécificités du langage** utilisé pour le codage qui suivra
7. **Tester** le fonctionnement de l'algorithme (structogramme) **sur papier** en passant par tous les chemins possibles

8. Traduire cet algorithme sous forme d'un **programme** en un langage de programmation adéquat (**commenter** le programme pour qu'il soit compréhensible)
 9. Faire **traduire** le programme du langage évolué au langage compréhensible par le processeur (compiler)
 10. Faire **exécuter** ce programme par le processeur (débuguer)
 11. **Tester** le programme en remplissant un **protocole**
- Attention à **ne pas faire plus** que ce qui est demandé dans le cahier des charges. Les fonctions superflues ont pris du temps de développement, donc du travail, donc un certain coût.
Si un client demande un logiciel (ou un appareil) qui doit comporter 5 fonctions et que vous lui en développez un qui en comporte 10, même si elles sont utiles, il va vous dire que la moitié de votre travail n'était pas demandé et va vous payer la moitié !
Si vous avez de bonnes idées d'amélioration, **proposez-les** au client et **faites modifier le cahier des charges** avant de les implémenter