



# Lois de Golub...

*... et autres lois de l'informatique*

## **Loi 1**

Aucun grand projet informatique n'est jamais mis en place dans les délais, dans les limites du budget, avec le même personnel qu'au départ et le projet ne fait pas non plus ce qu'il était censé faire. Il est fort improbable que le vôtre soit le premier.

- **Corollaire 1** : Les bénéfices seront inférieurs aux estimations - si on a pensé à faire des estimations.
- **Corollaire 2** : Le système finalement mis en place le sera avec du retard et ne fera pas ce qu'il est censé faire.
- **Corollaire 3** : Il coûtera plus cher, mais ce sera une réussite technique : l'opération réussit presque toujours, mais malheureusement le malade meurt très souvent.

## **Loi 2**

L'un des avantages à fixer un objectif vague à un projet est que vous n'aurez pas de difficulté à estimer les dépenses correspondantes.

## **Loi 3**

L'effort nécessaire croît géométriquement avec le temps.

- **Corollaire 1** : Plus vous attendez (pour définir les objectifs), plus c'est difficile.
- **Corollaire 2** : Après l'installation, c'est trop tard.
- **Corollaire 3** : Faites-le maintenant.

## **Loi 4**

Les buts, tels que les entend celui qui en décide, seront compris différemment par chacun des autres.

- **Corollaire 1** : Si vous expliquez avec une clarté telle qu'il soit impossible que qui que ce soit ait mal compris, ce sera le cas de quelqu'un.
- **Corollaire 2** : Si vous faites quelque chose qui, vous en êtes sûr, recevra l'approbation de tous, quelqu'un n'aimera pas ça.

## **Loi 5**

Seuls les bénéfices mesurables sont réels. Or, les bénéfices immatériels ne sont pas mesurables. Donc les bénéfices immatériels ne sont pas réels.

## **Loi 6**

Toute personne qui peut travailler à temps partiel pour un projet n'a sûrement pas assez de travail en ce moment.

- **Corollaire 1** : Si son patron ne lui donne pas un travail à temps complet, vous ne devez pas le faire non plus.
- **Corollaire 2** : Si cette personne a un problème de répartition d'horaire, le travail de son patron n'en souffrira pas.

### **Loi 7**

Plus grande est la complexité technique du projet, moins vous avez besoin d'un technicien pour le diriger.

- **Corollaire 1** : Trouvez le meilleur manager possible, il trouvera le technicien.
- **Corollaire 2** : Le contraire n'est presque jamais vrai.

### **Loi 8**

Un projet mal planifié prendra trois fois plus de temps à réaliser que prévu. Un projet bien planifié prendra seulement deux fois plus de temps.

### **Loi 9 (ou Première Loi de Murphy)**

S'il y a un risque que quelque chose marche mal, ça marchera mal.

- **Corollaire** : S'il est impossible que quelque chose marche mal, ça marchera mal quand même.

### **Loi 10 (ou Deuxième Loi de Murphy)**

Quand les choses vont bien, quelque chose ira mal.

- **Corollaire 1** : Quand les choses ne peuvent pas réellement devenir pire, elles le deviendront.
- **Corollaire 2** : Quand les choses semblent aller mieux, c'est que vous avez oublié quelque chose.

### **Loi 11**

Les équipes de projet détestent les comptes-rendus hebdomadaires d'avancement des travaux parce que ceux-ci mettent trop évidemment en lumière l'absence de leur progrès.

### **Loi 12**

Les projets progressent rapidement jusqu'à 90% de leur achèvement, puis ils restent achevés à 90% pour toujours.

### **Loi 13**

Si on laisse le contenu d'un projet changer librement, le taux de changement dépassera le taux d'avancement.

### **Loi 14**

Si l'utilisateur ne croit pas au système, il créera un système parallèle. Ni l'un ni l'autre ne fonctionneront très bien.

### **Loi 15**

Les bénéfices obtenus sont fonction de l'audit a posteriori.

### **Loi 16**

Aucune loi n'est immuable.

### **Loi de Brooks**

Lorsqu'un projet de logiciel est en retard, mobiliser de la main-d'œuvre supplémentaire ne fait qu'augmenter ce retard.

### **Axiome de Cann**

Quand rien ne marche, il est recommandé de lire les instructions.

### **Troisième loi de Clarke**

Il est impossible de faire la différence entre la magie et une technologie de pointe.

### **Quatrième loi de Murphy**

Si plusieurs choses risquent d'aller mal, le plus grand dommage sera causé précisément par la chose qui ira mal.

### **Lois de non-fiabilité de Gibbs**

- Les ordinateurs ne sont pas fiables, les hommes le sont encore moins. À l'origine de chaque erreur attribuée à l'ordinateur, on trouve au moins deux erreurs humaines, dont celle qui consiste à accuser l'ordinateur.
- Tout système qui dépend de la fiabilité de l'homme n'est pas fiable.
- La seule différence entre un imbécile et un criminel qui attaquent un système réside dans le fait que l'action de l'imbécile est imprévisible et d'une plus grande envergure.
- Il existe une quantité infinie d'erreurs non décelables, alors que le nombre des erreurs décelables est, ipso facto, limité.
- Les investissements pour assurer la fiabilité augmenteront jusqu'à ce qu'ils excèdent le coût probable des erreurs... ou jusqu'à ce que quelqu'un exige que l'on fasse du bon travail.

### **Loi de Hoare des grands programmes**

À l'intérieur de chaque grand logiciel, il existe un petit logiciel qui ne demande qu'à sortir.

### **Lois de programmation des ordinateurs**

- Lorsqu'il est en exploitation, n'importe quel programme est déjà obsolète.
- N'importe quel programme coûte plus cher et demande davantage de temps.
- Si un programme est utile, il faudra le changer.
- Si un programme est inutile, il faut augmenter la documentation qui le concerne.
- N'importe quel programme croît jusqu'à occuper toute la capacité mémoire.
- La valeur d'un programme est proportionnelle au poids de ses sorties.
- La complexité d'un programme augmente jusqu'à ce qu'elle excède les capacités du programmeur chargé de le maintenir.
- Si on faisait en sorte que les programmeurs puissent écrire les programmes en français, on s'apercevrait qu'ils ne savent pas écrire le français.

### **Loi de Meskimann**

On n'a jamais le temps de bien faire les choses, mais on a toujours le temps de les refaire.

### **Loi de Sattinger**

Ça marche mieux si on branche (Complément : en dernier ressort, tournez le commutateur).

### **Loi de Weinberg**

Si les architectes construisaient les bâtiments de la même façon que les programmeurs écrivent les programmes, le premier pivert venu détruirait la civilisation. ▲

*recueilli par Jean-Luc Blary*