

The Mythical Man-Month

~

Chapter 7 : Why Did the Tower of Babel Fails ?
Chapter 8 : Calling the shot

Dubuc Xavier



1^{er} décembre 2010

Plan

- 1 Chapter 7 ~ Why Did the Tower of Babel Fails ?
 - La communication dans les grands projets de programmation
 - Le "Project Workbook"
 - L'organisation dans les grands projets de programmation
- 2 Chapter 8 ~ Calling the shot
 - Les données de Portman
 - Les données de Aron
 - Les données de Harr
 - Les données de l'OS/360
 - Les données de Corbató
- 3 Analyse et critique
- 4 Bibliographie

Introduction

Ce chapitre traite des essentiels que doit posséder un projet afin qu'il soit mené à son terme et ce, dans les meilleurs délais et avec la meilleure qualité. Les 2 éléments les plus importants à gérer dans un projet sont :

Introduction

Ce chapitre traite des essentiels que doit posséder un projet afin qu'il soit mené à son terme et ce, dans les meilleurs délais et avec la meilleure qualité. Les 2 éléments les plus importants à gérer dans un projet sont :

- **l'organisation**

Introduction

Ce chapitre traite des essentiels que doit posséder un projet afin qu'il soit mené à son terme et ce, dans les meilleurs délais et avec la meilleure qualité. Les 2 éléments les plus importants à gérer dans un projet sont :

- **l'organisation**
- **la communication**

Conséquence d'un manque de communication

L'histoire nous apprend qu'un manque de communication peut être fatal à l'accomplissement d'un projet, c'est ce qui a eu lieu lors de la construction de la tour de Babel.

Exemple de la tour de Babel



FIGURE: La tour de Babel

Exemple de la tour de Babel : contexte

Si on en croit la Génèse, le Seigneur, prenant peur de ce que pouvaient faire les hommes, est intervenu pour que la langue unique ne soit plus d'actualité et ainsi compromettre la **communication** entre les hommes ce qui a mené à un manque d'**organisation**.

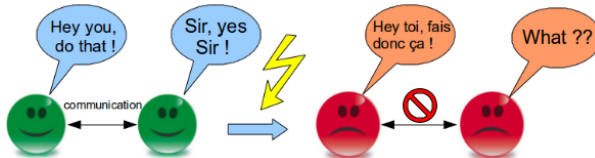


FIGURE: Problème de communication dû à la langue

Exemple de la tour de Babel : enseignements

Le manque de **communication** mène à :

Exemple de la tour de Babel : enseignements

Le manque de **communication** mène à :

- des disputes,

Exemple de la tour de Babel : enseignements

Le manque de **communication** mène à :

- des disputes,
- de la jalousie,

Exemple de la tour de Babel : enseignements

Le manque de **communication** mène à :

- des disputes,
- de la jalousie,
- une isolation des gens (ils évitent les querelles)

Exemple de la tour de Babel : enseignements

Le manque de **communication** mène à :

- des disputes,
- de la jalousie,
- une isolation des gens (ils évitent les querelles)

En conséquence, l'**organisation** s'en trouve affectée car ils ne peuvent se coordonner ce qui a eu pour effet, dans le cas de la tour de Babel, de stopper le projet.

Plan

- 1 Chapter 7 ~ Why Did the Tower of Babel Fails ?
 - La communication dans les grands projets de programmation
 - Le "Project Workbook"
 - L'organisation dans les grands projets de programmation
- 2 Chapter 8 ~ Calling the shot
 - Les données de Portman
 - Les données de Aron
 - Les données de Harr
 - Les données de l'OS/360
 - Les données de Corbató
- 3 Analyse et critique
- 4 Bibliographie

Introduction

De nos jours,

- les problèmes de calendrier,
- les incompatibilités fonctionnelles,
- les bugs de système,

surgissent à cause du fait qu'une partie du projet ne sait pas ce que font les autres parties.

Exemple

La partie s'occupant de la gestion des coûts ne sait pas forcément ce que fait la partie s'occupant de la production.

Pour éviter tous ces désagréments, les différentes parties d'un projet doivent avoir une **bonne communication** et ce, de toutes les manières possibles :

Pour éviter tous ces désagréments, les différentes parties d'un projet doivent avoir une **bonne communication** et ce, de toutes les manières possibles :

- **de manière informelle** : *via un bon service téléphonique et une définition claire des dépendances entre groupes,*

Pour éviter tous ces désagréments, les différentes parties d'un projet doivent avoir une **bonne communication** et ce, de toutes les manières possibles :

- **de manière informelle** : *via un bon service téléphonique et une définition claire des dépendances entre groupes,*
- **via des réunions** : *(précieuses) réunions de projet régulières où les différentes équipes présentent des briefings techniques,*

Pour éviter tous ces désagréments, les différentes parties d'un projet doivent avoir une **bonne communication** et ce, de toutes les manières possibles :

- **de manière informelle** : *via un bon service téléphonique et une définition claire des dépendances entre groupes,*
- **via des réunions** : *(précieuses) réunions de projet régulières où les différentes équipes présentent des briefings techniques,*
- *un utilisant un "Project Workbook" (littéralement "classeur de projet") que l'on initialise dès le début du projet.*

Plan

- 1 Chapter 7 ~ Why Did the Tower of Babel Fails ?
 - La communication dans les grands projets de programmation
 - **Le "Project Workbook"**
 - L'organisation dans les grands projets de programmation
- 2 Chapter 8 ~ Calling the shot
 - Les données de Portman
 - Les données de Aron
 - Les données de Harr
 - Les données de l'OS/360
 - Les données de Corbató
- 3 Analyse et critique
- 4 Bibliographie

Project Workbook : Définition

Ce *project workbook* (**PW**), est un document à part régissant la structure des documents que le projet va fournir quoi qu'il arrive. Chacun de ces documents doit faire partie de cette structure, cela inclut :

Project Workbook : Définition

Ce *project workbook* (**PW**), est un document à part régissant la structure des documents que le projet va fournir quoi qu'il arrive. Chacun de ces documents doit faire partie de cette structure, cela inclut :

- les objectifs,

Project Workbook : Définition

Ce *project workbook* (**PW**), est un document à part régissant la structure des documents que le projet va fournir quoi qu'il arrive. Chacun de ces documents doit faire partie de cette structure, cela inclut :

- les objectifs,
- les spécifications externes,

Project Workbook : Définition

Ce *project workbook* (**PW**), est un document à part régissant la structure des documents que le projet va fournir quoi qu'il arrive. Chacun de ces documents doit faire partie de cette structure, cela inclut :

- les objectifs,
- les spécifications externes,
- les spécifications internes,

Project Workbook : Définition

Ce *project workbook* (**PW**), est un document à part régissant la structure des documents que le projet va fournir quoi qu'il arrive. Chacun de ces documents doit faire partie de cette structure, cela inclut :

- les objectifs,
- les spécifications externes,
- les spécifications internes,
- les spécifications de l'interface,

Project Workbook : Définition

Ce *project workbook* (**PW**), est un document à part régissant la structure des documents que le projet va fournir quoi qu'il arrive. Chacun de ces documents doit faire partie de cette structure, cela inclut :

- les objectifs,
- les spécifications externes,
- les spécifications internes,
- les spécifications de l'interface,
- les standards techniques,

Project Workbook : Définition

Ce *project workbook* (**PW**), est un document à part régissant la structure des documents que le projet va fournir quoi qu'il arrive. Chacun de ces documents doit faire partie de cette structure, cela inclut :

- les objectifs,
- les spécifications externes,
- les spécifications internes,
- les spécifications de l'interface,
- les standards techniques,
- les mémos administratifs.

Project Workbook : Utilités

- 1 Si on examine la généalogie d'un tel document, on peut non seulement **retracer les idées** mais aussi une grande partie des phrases du premier mémo qui propose le produit ou explique le premier design.

Project Workbook : Utilités

- 1 Si on examine la généalogie d'un tel document, on peut non seulement **retracer les idées** mais aussi une grande partie des phrases du premier mémo qui propose le produit ou explique le premier design.
- 2 Il permet le **contrôle de la distribution de l'information**.
L'idée n'est pas de restreindre l'information mais d'assurer que l'information pertinente parvienne à toutes les personnes qui en ont besoin.

Project Workbook : En pratique

Le problème du **PW** n'est **pas linéaire** avec la taille du projet, ainsi, plus le projet rassemble de gens, plus le **PW** est grand (en taille) et plus il est **nécessaire** pour la bonne organisation du projet.

⇒ **Importance de trouver un bon moyen de le tenir à jour.**

Project Workbook : Cas de l'OS/360

Ils disposaient d'un système d'édition de texte dirigé par un ordinateur pour éditer un **PW** à feuilles mobiles (ils imprimaient les modifications).

Project Workbook : Cas de l'OS/360

Ils disposaient d'un système d'édition de texte dirigé par un ordinateur pour éditer un **PW** à feuilles mobiles (ils imprimaient les modifications).

- au début du projet, ils en étaient totalement contents,

Project Workbook : Cas de l'OS/360

Ils disposaient d'un système d'édition de texte dirigé par un ordinateur pour éditer un **PW** à feuilles mobiles (ils imprimaient les modifications).

- au début du projet, ils en étaient totalement contents,
- après **6 mois** consacrés au projet, les premiers problèmes sont apparus, problèmes liés à la **taille du PW** :

Project Workbook : Cas de l'OS/360

Ils disposaient d'un système d'édition de texte dirigé par un ordinateur pour éditer un **PW** à feuilles mobiles (ils imprimaient les modifications).

- au début du projet, ils en étaient totalement contents,
- après **6 mois** consacrés au projet, les premiers problèmes sont apparus, problèmes liés à la **taille du PW** :
 - une copie de ce **PW** faisait **1,524m de large**, les **misés à jour 5cm** (*environ 150 pages*)

Project Workbook : Cas de l'OS/360

Ils disposaient d'un système d'édition de texte dirigé par un ordinateur pour éditer un **PW** à feuilles mobiles (ils imprimaient les modifications).

- au début du projet, ils en étaient totalement contents,
- après **6 mois** consacrés au projet, les premiers problèmes sont apparus, problèmes liés à la **taille du PW** :
 - une copie de ce **PW** faisait **1,524m de large**, les **misés à jour 5cm** (*environ 150 pages*)
 - possédant 100 bureaux, il y avait donc 100 copies de ce **PW**, en les empilant, la pile dépassait le toit du *Manhattan's Time-Life Building*.

⇒ **l'entretien du PW prenait un temps conséquent !**

Project Workbook : Cas de l'OS/360



FIGURE: Manhattan's Time-Life Building

Project Workbook : Cas de l'OS/360 ~ Solution

Solution employée : passage aux **microfiches**,

- ⇒ 1 million de dollars sauvés,
- ⇒ le **PW** a diminué de $0.0849m^3$ à $0.00429m^3$ (*division par 2 du volume !*) et les mises à jour apparaissaient par morceau de centaines de pages, réduisant par 100 le problème d'insertion.

Néanmoins, ces microfiches ont leurs **inconvénients**, elles ne peuvent pas, par exemple, être marquées, surlignées, ... il est donc moins évident de se rendre compte des modifications apportées à un document.

Project Workbook : Les années 90

Avec la technologie de systèmes qui arrivent, la meilleure technique, selon le livre, est de garder le **PW** sur des fichiers à accès direct, marqué par des barres de changements et des dates de révision.

- Chaque utilisateur le consultait depuis un terminal d'affichage (*les machines à écrire sont trop lentes*),
- un résumé des changements, préparé chaque jour, serait stocké de manière **LIFO** dans un point d'accès fixé,

Project Workbook : Les années 90

Les **avantages** sont notables, par exemple :

- un programmeur devrait normalement lire ces résumés chaque jour, mais admettons qu'il soit malade 2-3 jours, à son retour il n'aurait qu'à lire ces résumés pour se tenir au courant des changements qui ont eu lieu en son absence.
- **de plus**, il peut s'interrompre pendant sa lecture pour consulter le texte dans le **PW** qui est concerné par les modifications reprises dans le résumé.

Plan

- 1 Chapter 7 ~ Why Did the Tower of Babel Fails ?
 - La communication dans les grands projets de programmation
 - Le "Project Workbook"
 - **L'organisation dans les grands projets de programmation**
- 2 Chapter 8 ~ Calling the shot
 - Les données de Portman
 - Les données de Aron
 - Les données de Harr
 - Les données de l'OS/360
 - Les données de Corbató
- 3 Analyse et critique
- 4 Bibliographie

Introduction

n travailleurs dans un projet

⇒ $\frac{n^2 - n}{2}$ interfaces de communication différentes,

⇒ 2^n équipes potentielles à coordonner.

Le but de l'**organisation** est de réduire le nombre de **communications** et de **coordinations** nécessaires.

Éviter les communications

Pour **éviter les communications**, il est recommandé d'appliquer les principes élémentaires de **management** tels que **les principes de Fayol**. Les plus importants en ce qui nous concerne sont :

- ① la division du travail,
- ② la spécialisation fonctionnelle.

Ces 2 principes impliquent que la structure de l'entreprise tende vers un arbre, afin de respecter un autre de **ces principes de Fayol**, l'**unicité du supérieur hiérarchique direct**.

Unicité du chef : pourquoi ?

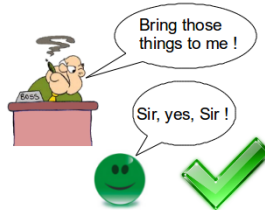


FIGURE: Principe du chef unique respecté.

Unicité du chef : pourquoi ?



FIGURE: Principe du chef unique non-respecté.

Structure

Chaque sous-arbre de la structure se doit de contenir les éléments suivants pour être fonctionnel :

Structure

Chaque sous-arbre de la structure se doit de contenir les éléments suivants pour être fonctionnel :

- 1 une mission,

Structure

Chaque sous-arbre de la structure se doit de contenir les éléments suivants pour être fonctionnel :

- ① une mission,
- ② un producteur,

Structure

Chaque sous-arbre de la structure se doit de contenir les éléments suivants pour être fonctionnel :

- ① une mission,
- ② un producteur,
- ③ un directeur technique,

Structure

Chaque sous-arbre de la structure se doit de contenir les éléments suivants pour être fonctionnel :

- ① une mission,
- ② un producteur,
- ③ un directeur technique,
- ④ un calendrier,

Structure

Chaque sous-arbre de la structure se doit de contenir les éléments suivants pour être fonctionnel :

- ① une mission,
- ② un producteur,
- ③ un directeur technique,
- ④ un calendrier,
- ⑤ une division du travail,

Structure

Chaque sous-arbre de la structure se doit de contenir les éléments suivants pour être fonctionnel :

- ① une mission,
- ② **un producteur**,
- ③ **un directeur technique**,
- ④ un calendrier,
- ⑤ **une division du travail**,
- ⑥ des définitions d'interfaces entre les parties.

Toutes ces choses sont conventionnelles et évidentes, mis à part la distinction entre le **producteur** et le **directeur technique**.

Rôles du producteur

- Rassembler l'équipe

Rôles du producteur

- Rassembler l'équipe
- Diviser le travail

Rôles du producteur

- Rassembler l'équipe
- Diviser le travail
- Établir le calendrier et s'assurer qu'il est respecté

Rôles du producteur

- Rassembler l'équipe
- Diviser le travail
- Établir le calendrier et s'assurer qu'il est respecté
- Acquérir les ressources nécessaires

Rôles du producteur

- Rassembler l'équipe
- Diviser le travail
- Établir le calendrier et s'assurer qu'il est respecté
- Acquérir les ressources nécessaires
- Établir le modèle de communication et de rapports à l'intérieur de l'équipe

Rôles du producteur

- Rassembler l'équipe
- Diviser le travail
- Établir le calendrier et s'assurer qu'il est respecté
- Acquérir les ressources nécessaires
- Établir le modèle de communication et de rapports à l'intérieur de l'équipe

Cela signifie qu'une majeure partie de son rôle consiste en **communications en dehors de son équipe** (*envers les instances supérieures ou avec des collègues du même niveau que lui*).

Rôles du directeur technique

- Concevoir le design

Rôles du directeur technique

- Concevoir le design
- Identifier les sous-parties

Rôles du directeur technique

- Concevoir le design
- Identifier les sous-parties
- Spécifier à quoi le projet va ressembler vu de l'extérieur

Rôles du directeur technique

- Concevoir le design
- Identifier les sous-parties
- Spécifier à quoi le projet va ressembler vu de l'extérieur
- Esquisser la structure interne

Rôles du directeur technique

- Concevoir le design
- Identifier les sous-parties
- Spécifier à quoi le projet va ressembler vu de l'extérieur
- Esquisser la structure interne
- Fournir l'unité et l'intégrité conceptuelle à tout le design

Rôles du directeur technique

- Concevoir le design
- Identifier les sous-parties
- Spécifier à quoi le projet va ressembler vu de l'extérieur
- Esquisser la structure interne
- Fournir l'unité et l'intégrité conceptuelle à tout le design
- Servir de limite à la complexité du système

Rôles du directeur technique

- Concevoir le design
- Identifier les sous-parties
- Spécifier à quoi le projet va ressembler vu de l'extérieur
- Esquisser la structure interne
- Fournir l'unité et l'intégrité conceptuelle à tout le design
- Servir de limite à la complexité du système
- Inventer des solutions à d'éventuels problèmes techniques qui surgissent (ou changer le design en conséquence)

Rôles du directeur technique

- Concevoir le design
- Identifier les sous-parties
- Spécifier à quoi le projet va ressembler vu de l'extérieur
- Esquisser la structure interne
- Fournir l'unité et l'intégrité conceptuelle à tout le design
- Servir de limite à la complexité du système
- Inventer des solutions à d'éventuels problèmes techniques qui surgissent (ou changer le design en conséquence)

Ses **communications sont principalement "intra-équipe"** et son travail est presque entièrement technique.

Relation producteur-directeur technique

Après avoir bien différencier ces 2 fonctions, il est clair que les 2 rôles demandent des talents différents, il y a dès lors 3 situation possibles :

Relation producteur-directeur technique

Après avoir bien différencier ces 2 fonctions, il est clair que les 2 rôles demandent des talents différents, il y a dès lors 3 situation possibles :

- ❶ le producteur et le directeur technique sont une seule et même personne,
- ❷ le producteur est le boss et le directeur technique son bras droit,
- ❸ le directeur technique est le boss et le producteur son bras droit,

Le producteur et le directeur technique sont une seule et même personne

Cette situation est réalisable sur de **très petites équipes** (*typiquement de 3 à 6 programmeurs*) mais au delà, 2 problèmes se posent :

Le producteur et le directeur technique sont une seule et même personne

Cette situation est réalisable sur de **très petites équipes** (*typiquement de 3 à 6 programmeurs*) mais au delà, 2 problèmes se posent :

- ❶ la personne la meilleure en **management** et en **technique** est rarement trouvée,

Le producteur et le directeur technique sont une seule et même personne

Cette situation est réalisable sur de **très petites équipes** (*typiquement de 3 à 6 programmeurs*) mais au delà, 2 problèmes se posent :

- ❶ la personne la meilleure en **management** et en **technique** est rarement trouvée,
- ❷ il est **difficile** pour le producteur de déléguer assez de ses fonctions pour s'occuper de son rôle de directeur technique et, dans l'autre sens, c'est **impossible** pour le directeur technique de déléguer sans compromettre l'intégrité conceptuelle du projet.

Le producteur est le boss et le directeur technique son bras droit

- Difficulté d'établir l'autorité du directeur technique à prendre des décisions sans impact tel sur son temps qu'il serait placé dans la chaîne de commande du management.
- Les 2 hommes doivent avoir la même vue sur la philosophie technique fondamentale.
- Ils doivent discuter sur les questions techniques de manière privée avant qu'elles ne deviennent opportunes.

Cette structure peut être organisée pour être efficace mais est, hélas, rarement essayée en pratique.

Le directeur technique est le boss et le producteur son bras droit

Le producteur se doit de proclamer l'autorité suprême du directeur technique tout en restant ultra disponible et au petit soin pour lui de manière à ce que ce dernier n'ait à s'occuper que de l'ingénierie.

Cet arrangement peut également être organisé de manière à fonctionner de manière effective, mais Brooks suspecte que cet arrangement est mieux adapté pour les petites équipes.

Structure idéale

Selon [Brooks](#), la structure dans laquelle le **directeur technique** est le boss est la plus fiable pour les plus gros sous-arbres d'un projet vraiment gros.



FIGURE: Structure idéale

Conclusion du chapitre

- **La tour de Babel** était peut-être le premier fiasco de l'ingénierie mais ce n'était clairement **pas le dernier**.
- La **communication** et ses conséquences ainsi que l'**organisation** sont **critiques** pour la réussite d'un projet.
- Les *techniques de communication* et d'*organisation* demandent du manager **plus de pensées** et **autant de compétences d'expérience** que la **technologie du logiciel lui-même**.

Plan

- 1 Chapter 7 ~ Why Did the Tower of Babel Fails ?
 - La communication dans les grands projets de programmation
 - Le "Project Workbook"
 - L'organisation dans les grands projets de programmation
- 2 Chapter 8 ~ Calling the shot
 - Les données de Portman
 - Les données de Aron
 - Les données de Harr
 - Les données de l'OS/360
 - Les données de Corbató
- 3 Analyse et critique
- 4 Bibliographie

Entrée en matière



"La pratique est le meilleur des instructeurs"

Publilius Syrus. (poète latin)

"L'expérience est cher enseignant, mais les fous n'apprendront à personne."

Benjamin Franklin.



Fil directeur

Comment peut-on estimer

Fil directeur

Comment peut-on estimer

- le **temps** que va prendre un projet ?

Fil directeur

Comment peut-on estimer

- le **temps** que va prendre un projet ?
- les **efforts nécessaires** à sa réussite ?

Fil directeur

Comment peut-on estimer

- le **temps** que va prendre un projet ?
- les **efforts nécessaires** à sa réussite ?

Cette partie se base sur des **résultats expérimentaux** publiés par différents managers de projet afin de répondre au mieux à ces questions.

Préambule

Il faut savoir qu'il **ne suffit pas d'estimer la portion de code**
mais qu'il faut également tenir compte :

Préambule

Il faut savoir qu'il **ne suffit pas d'estimer la portion de code** mais qu'il faut également tenir compte :

- du planning,
- de la documentation,
- des tests (des composants et du système),
- de l'intégration du système
- des temps de formation.

Préambule

Il faut également considérer le fait que les données trouvées sur un **petit projet** ne seront pas forcément applicables aux **grands projets**, et inversement.

Des études ont montré que l'effort à fournir augmente comme une puissance de la taille du projet et qu'il pouvait être estimé comme suit :

$$\text{effort} = (\text{constante}) \times (\text{nombre d'instructions})^{1.5}$$

Introduction aux données

Cette partie présente à présent différentes données recueillies par des managers lors de leurs expériences professionnelles, ces données concernent toutes la **productivité** d'un programmeur.

Plan

- 1 Chapter 7 ~ Why Did the Tower of Babel Fails ?
 - La communication dans les grands projets de programmation
 - Le "Project Workbook"
 - L'organisation dans les grands projets de programmation
- 2 Chapter 8 ~ Calling the shot
 - **Les données de Portman**
 - Les données de Aron
 - Les données de Harr
 - Les données de l'OS/360
 - Les données de Corbató
- 3 Analyse et critique
- 4 Bibliographie

Portman

Charles Portman, *manager de ICL's Software Division, Computer Equipment Organization (Northwest) à Manchester* ; a découvert que **seulement la moitié** du temps que possède un programmeur est employé pour programmer et debugger.

Le reste du temps est occupé par :

- des **réunions**,
- de la **paperasse**,
- des éventuelles **maladies**,
- ...

Portman

⇒ Les estimations font une **hypothèse irréaliste** à propos du nombre d'heures de technique par personne/an.

Plan

- 1 Chapter 7 ~ Why Did the Tower of Babel Fails ?
 - La communication dans les grands projets de programmation
 - Le "Project Workbook"
 - L'organisation dans les grands projets de programmation
- 2 Chapter 8 ~ Calling the shot
 - Les données de Portman
 - **Les données de Aron**
 - Les données de Harr
 - Les données de l'OS/360
 - Les données de Corbató
- 3 Analyse et critique
- 4 Bibliographie

Aron

Joel Aron, *manager de Systems Technology chez IBM à Gaithersburg dans le Maryland*, a étudié la productivité d'un programmeur lorsqu'il travaille sur 9 grands projets.
(*grand = plus de 25 programmeurs et plus de 30 000 instructions livrables*)

Il en a dégagé la loi suivante :

Nombre d'interactions entre programmeurs ↗
⇒ **Productivité** ↘

Plan

- 1 Chapter 7 ~ Why Did the Tower of Babel Fails ?
 - La communication dans les grands projets de programmation
 - Le "Project Workbook"
 - L'organisation dans les grands projets de programmation
- 2 Chapter 8 ~ Calling the shot
 - Les données de Portman
 - Les données de Aron
 - **Les données de Harr**
 - Les données de l'OS/360
 - Les données de Corbató
- 3 Analyse et critique
- 4 Bibliographie

Harr

John Harr, *manager de programmation pour Bell Telephone Laboratories' Electronic Switching System* a réuni des données concernant **4 programmes de taille similaire** différents uniquement au niveau :

- de la taille des groupes de travail,
- du temps pris,
- du nombre de modules.

⇒ On constate dès lors que ces programmes donnent des productivités complètement différentes, cela étant probablement dû à la complexité des problèmes mais on ne peut que faire des suppositions.

Plan

- 1 Chapter 7 ~ Why Did the Tower of Babel Fails ?
 - La communication dans les grands projets de programmation
 - Le "Project Workbook"
 - L'organisation dans les grands projets de programmation
- 2 Chapter 8 ~ Calling the shot
 - Les données de Portman
 - Les données de Aron
 - Les données de Harr
 - **Les données de l'OS/360**
 - Les données de Corbató
- 3 Analyse et critique
- 4 Bibliographie

OS/360

Ces données confirment la conclusion précédente, c'est-à-dire que la productivité est fortement liée à la complexité et à la difficulté de la tâche à effectuer.

Plan

- 1 Chapter 7 ~ Why Did the Tower of Babel Fails ?
 - La communication dans les grands projets de programmation
 - Le "Project Workbook"
 - L'organisation dans les grands projets de programmation
- 2 Chapter 8 ~ Calling the shot
 - Les données de Portman
 - Les données de Aron
 - Les données de Harr
 - Les données de l'OS/360
 - **Les données de Corbató**
- 3 Analyse et critique
- 4 Bibliographie

Corbató

Corbató du *MIT's Project MAC* a recueilli des données sur le système d'exploitation MULTICS (entre 1 million et 2 millions de mots dans le langage de programmation PL/1 (ancêtre du REXX)), ces données suggèrent 2 conclusions importantes :

- 1 La productivité semble être constante en terme de statement élémentaire (mots/lignes de code),
- 2 La productivité peut être **multipliée par 5** lorsqu'un langage de programmation de haut niveau approprié est utilisé!

(Les données précédentes concernaient toutes la programmation en langage assembleur)

Conclusion

- 1 Il faut **estimer de manière prudente** le temps pris par un programmeur pour finir son travail, ce temps devant comprendre les réunions, les éventuelles maladies, ...

Conclusion

- 1 Il faut **estimer de manière prudente** le temps pris par un programmeur pour finir son travail, ce temps devant comprendre les réunions, les éventuelles maladies, ...
- 2 La productivité (et donc l'effort fourni et le temps pris pour le fournir) du programmeur dépend fortement de la **complexité et de la difficulté du problème** qui lui est soumis.

Conclusion

- 1 Il faut **estimer de manière prudente** le temps pris par un programmeur pour finir son travail, ce temps devant comprendre les réunions, les éventuelles maladies, ...
- 2 La productivité (et donc l'effort fourni et le temps pris pour le fournir) du programmeur dépend fortement de la **complexité et de la difficulté du problème** qui lui est soumis.
- 3 L'utilisation d'un langage de programmation de haut niveau **multiplie par 5** la productivité.

Plan

- 1 Chapter 7 ~ Why Did the Tower of Babel Fails ?
 - La communication dans les grands projets de programmation
 - Le "Project Workbook"
 - L'organisation dans les grands projets de programmation
- 2 Chapter 8 ~ Calling the shot
 - Les données de Portman
 - Les données de Aron
 - Les données de Harr
 - Les données de l'OS/360
 - Les données de Corbató
- 3 **Analyse et critique**
- 4 **Bibliographie**

Points forts

- Mise en avant l'importance du management.
- Structure irréfutable.
- Approfondissement de chaque terme utilisé.

Points faibles

- Anglais assez soutenu.
- Résultats concernant d'anciens projets, pas très représentatifs à nos yeux.
- Utilisation de la théorie en pratique pas d'actualité (microfiches, terminaux d'affichage, ...).
- Utilisation de la première personne pour la narration (objectivité).

Apprentissage

Peu de choses non connues, en effet, la plupart du temps l'auteur parle de principes qui sont la base du management. Ces mêmes principes que nous avons vu dans le cours du même nom l'année passée.

Le livre n'étant pas très récent, il parle à peine des techniques disponibles de nos jours pour gérer un **Project Workbook** de manière simple.

Project Workbook de nos jours

Il existe de nombreuses solutions informatiques,

- pour les projets open-source, les [forges](#) sont recommandées,
- pour les autres projets, il existe des logiciels tels que [Microsoft Project](#) qui permettent de gérer un projet de A à Z de manière simple.

Plan

- 1 Chapter 7 ~ Why Did the Tower of Babel Fails ?
 - La communication dans les grands projets de programmation
 - Le "Project Workbook"
 - L'organisation dans les grands projets de programmation
- 2 Chapter 8 ~ Calling the shot
 - Les données de Portman
 - Les données de Aron
 - Les données de Harr
 - Les données de l'OS/360
 - Les données de Corbató
- 3 Analyse et critique
- 4 **Bibliographie**

Bibliographie

- Brooks, F.P : *The Mythical Man-Month : Essays on Software Engineering*. 20th anniversary edn. Addison-Wesley (1995)
- Forge : [http ://fr.wikipedia.org/wiki/Forge_\(informatique\)](http://fr.wikipedia.org/wiki/Forge_(informatique))
- Microsoft Project :
[http ://www.microsoft.com/france/office/project/](http://www.microsoft.com/france/office/project/)