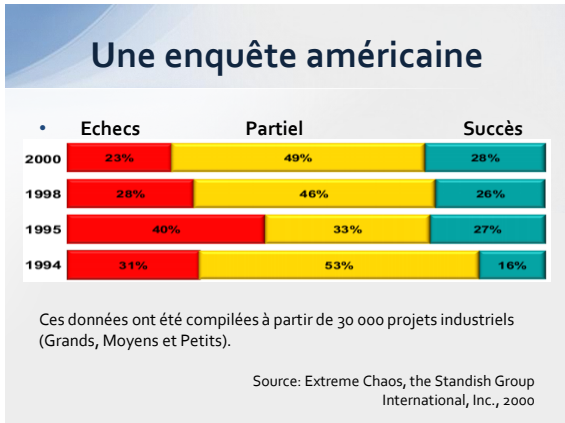
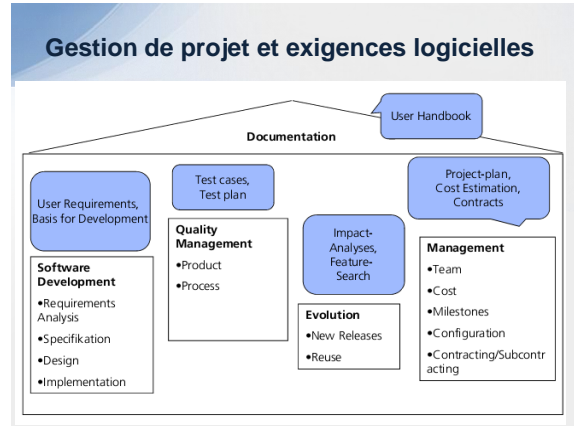


Département Informatique
Master1 SIGL
Cours Ingénierie des Besoins

Introduction



Symptômes des projets réalisés partiellement

- Ce qu'ont dit les décideurs et les utilisateurs
 - "Ce ne marche pas dans notre environnement"
 - "Le projet fut en retard et hors budget"
 - "Au final, ce n'est pas ce dont nous avions besoin"
 - "Ce ne correspond pas à nos attentes... Nous sommes mécontents"
 - "C'est trop difficile à utiliser"
 - "Ce truc est imprévisible - On découvre chaque jour de nouveaux problèmes"
- Ce qu'ont dit les informaticiens
 - "Nous ne pouvions obtenir les informations nécessaires au projet"
 - "Nous n'avons pas vraiment compris ce que nous devions faire."
 - "Nous ne savions pas si le travail des autres équipes impacteraient notre travail"

Facteurs d'échec

Les facteurs qui font que le projet soit en retard ou ne répond jamais aux exigences des décideurs et des utilisateurs (selon Standish Groupe 2006).

Une mauvaise gestion des exigences est à l'origine de la plus part des échecs !!!

Les 10 premières causes d'échec

- Pas assez d'informations de l'utilisateur **13%**
- Exigences incomplètes **12%**
- Changement exigences **11%**
- Pas assez de soutien de la direction **8%**
- Incompétence technique **7%**
- Manque de ressources **6%**
- Attentes irréalistes **6%**
- Objectifs confus **5%**
- Phasage irréaliste **4%**
- Nouvelles technologies **3%**

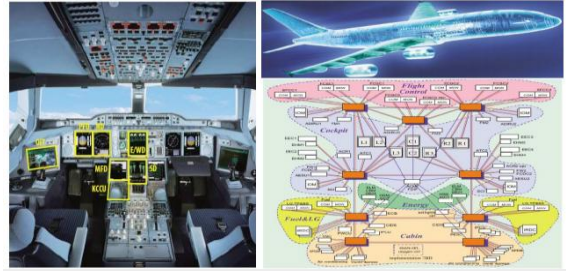
Les erreurs d'analyse sont

- **les plus coûteuses**

Erreur détectée	Coût de réparation
En Analyse	1
En Conception	5
En Codage	10
En Tests	20
En exploitation	200
- **les plus nombreuses**
 - 67% : erreurs de spécification et conception
 - 33% : erreurs de codage

Airbus A380

- Trop d'ordinateurs et trop de logiciels



Airbus A380

- Airbus 380: Environ 1 milliard (1.000.000.000) de lignes de code.
- Comparativement à Windows XP: ~40 million de lignes de code.

Ceci donne une idée de la taille du défis auquel font face les ingénieurs logiciel !!!

La sonde sur Mars

- En 1999 le «Mars Climate Orbiter» **disparaît** alors qu'il débute son orbite autour de Mars.
- Coût: environ 125 millions de dollars US.
- Problème causé par une erreur de transfert d'information entre une équipe au Colorado et une en Californie.
- Une équipe utilisait le système de mesure **anglais** (pouces, pieds, livres...) alors que l'autre utilisait le système **métrique** pour une fonction clé de l'appareil...

Il reste toujours vrai que...

- "The hardest single part of building a software system is deciding precisely what to build." (Brooks, 1987)
- "Coût d'une correction tardive = 200 fois coût d'une correction initiale" (B. Boem, 1981)

Qu'est-ce que c'est qu'un besoin?

- (1) A condition or capability needed by a user to solve a problem or achieve an objective;
- (2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents;
- (3) A documented representation of such a condition or capability.

IEEE Standards Collection: Software Engineering

Qu'est-ce que c'est qu'un besoin?

- Les besoins (exigences ou requis) décrivent la raison d'être d'un système.
- Les besoins expriment les idées qui doivent être incarnées dans un système ou une application en développement.
- La définition varie, mais reste généralement autour de ces lignes:

A capability that the system must deliver or a condition that it must be satisfied in order to address a need of a stakeholder. [Larman, 2002]

Ingénierie des besoins (Requirements Engineering)

Une approche systématique de spécification et de gestion des besoins pour les buts suivants:

1. Savoir les besoins pertinents, accomplir un consensus des parties prenantes au sujet de ces besoins, les documenter selon les standards en vigueur, et les gérer systématiquement.
2. Comprendre et documenter les attentes et souhaits des parties prenantes.
3. Spécifier et gérer les besoins pour minimiser le risque de livraison d'un système qui ne satisfait pas les attentes et souhaits des parties prenantes.

Qu'est ce qu' une partie prenante (Stakeholder)?

- Clients / investisseur
- Acheteurs
- Utilisateur final
- Experts du domaine
- Les fournisseurs de contenu
- Développeurs, Ingénieurs logiciel, gestionnaires de projets, ...
- Inspecteurs (reviewer)
- Experts d'un autre système en liaison avec le projet
- Tout autre personne qui apporte une valeur ajoutée au futur système.

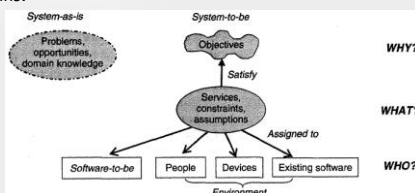
Système existant et système futur

- Système existant (system-as-is): avant l'informatisation.
- Système futur (system-to-be): après l'informatisation.
- Système existant a ses problèmes, limites, défauts.
- Système futur doit résoudre ces problèmes en utilisant les technologies adéquates avec la coopération de son environnement (composants physiques et organisationnels)

Dimensions de l'Ingénierie des Besoins

Le problème d'informatisation d'un système existant est structuré en trois dimensions:

- **POURQUOI** on a besoin d'un système futur
- Quels (**QUOI**) besoins doivent être accomplis par ce système
- **QUI**, dans ce système, doit participer à l'accomplissement de ces besoins.



Dimension POURQUOI (why)

- Expliciter les objectifs à atteindre.
- Acquérir une connaissance du domaine: concepts, lois, procédures, terminologie, etc.
- Evaluer les alternatives: les pour et les contre.
- Evaluer les opportunités technologiques.
- Maîtriser les conflits.

Dimension QUOI (what)

- Les services (fonctions) que le système futur doit fournir pour satisfaire les objectifs identifiés dans la dimension POURQUOI.
- Ces services sont basés sur des hypothèses et suppositions.
- Ils doivent satisfaire des contraintes liées à la performance, la sécurité, l'exploitation, l'interopérabilité et le coût.

Dimension QUI (who)

- Affectation des responsabilités pour achever les objectifs, services et contraintes, parmi les composants du système futur: humains, appareils et software.
- Affectations souvent critiques.
- Evaluer les affectations alternatives
- Elaborer la limite entre logiciel et environnement.