



01

Chapitre

Introduction au génie logiciel

112AC1 : Génie logiciel et Conception orientée objet

Régis Clouard, ENSICAEN - GREYC

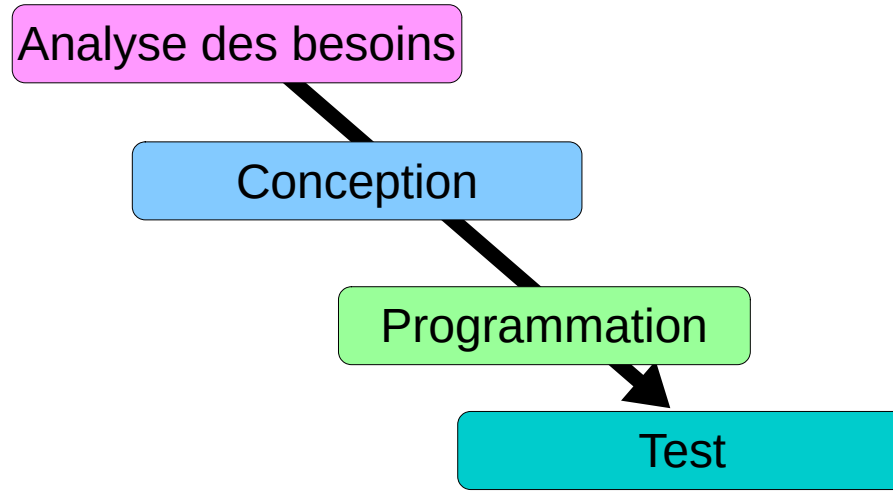
« Si les ouvriers construisaient les bâtiments
comme les développeurs écrivent leurs programmes,
le premier pic-vert venu aurait détruit toute civilisation. »

Gerald Weinberg

Cause d'échec n°3 :

Définir les besoins au début

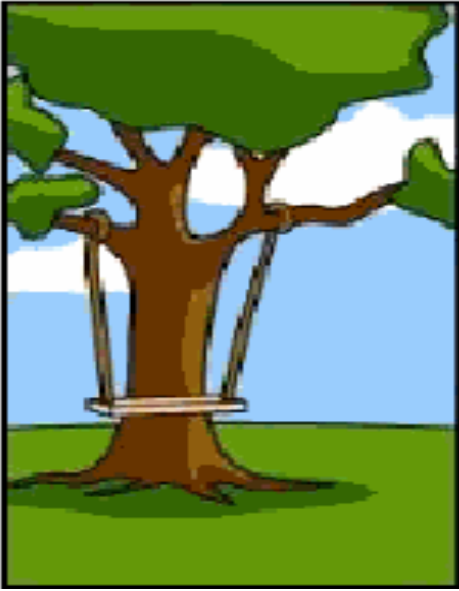
61



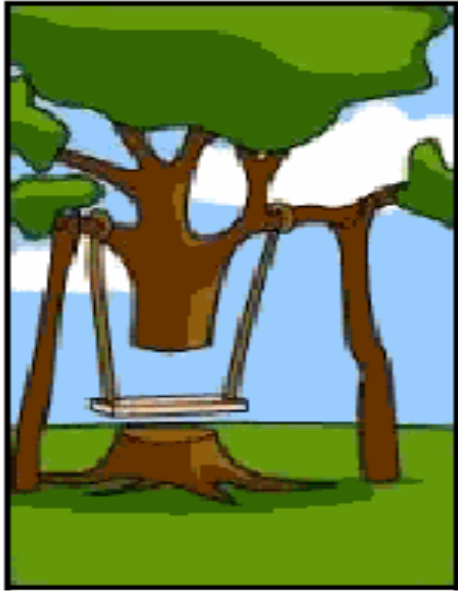
- Les clients ne savent pas identifier exactement ce qu'ils veulent
- Les clients ne savent pas exprimer clairement les besoins identifiés
- Les clients changent leurs besoins au cours du projet



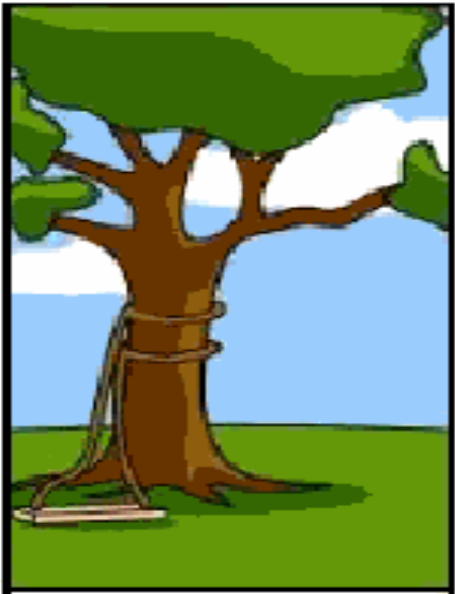
Ce que le client explique



Ce que le chef de projet comprend



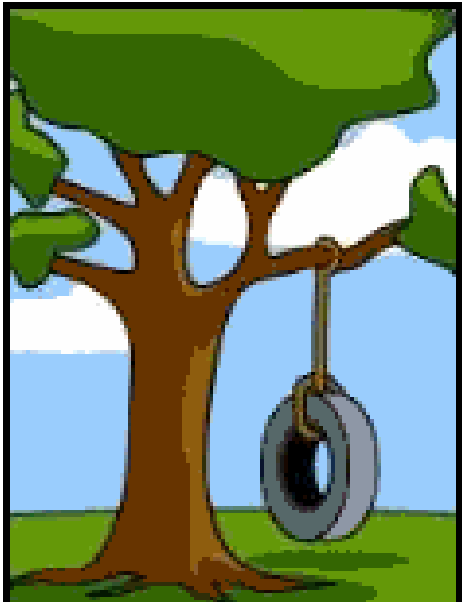
Ce que les analystes modélisent



Ce que les développeurs programment



Ce que les commerciaux vendent

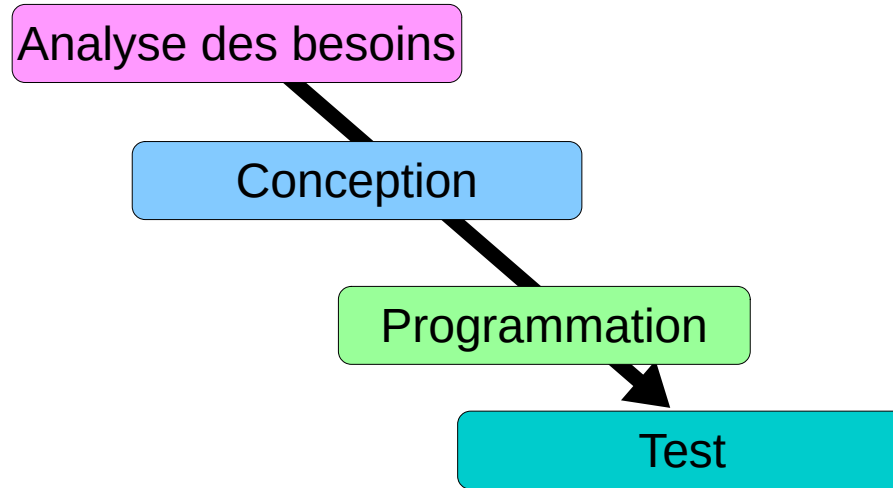


Ce que le client a réellement besoin

**Le produit final est le résultat
d'une séquence d'incompréhensions**

Cause d'échec n°4 : Réaliser les tests à la fin

65



- Les tests sont généralement la variable d'ajustement du temps
 - Les tests sont faits selon le temps restant (ou pas)
- Pourtant, il est impossible de garantir des logiciels sans défaut
 - Estimation : 1 à 10 bugs / KLOC
 - ▶ Windows 10 (80 MLOC) → **80 000 bugs !**

Cause d'échec n°5

Raisonner au niveau procédure

66

- Le paradigme procédurale est inadapté à la conception de programmes de très grande taille
 - Programmer au niveau procédural est une activité complexe qui revient à construire une voiture au niveau moléculaire ; le niveau atomique correspondant à l'assembleur

Constat préliminaire

- La logiciel se prête mal aux méthodes du génie civil
 - Suivre un plan élaboré au début du projet
 - Estimer la durée des étapes d'un plan
 - Définir les besoins au début
 - Réaliser les tests à la fin
- Le logiciel n'est pas le matériel : il est mou
 - On peut construire une tour de 200 m en commençant par un bungalow
 - On peut construire un bungalow en commençant par la salle de bain
 - On peut construire une salle de bain en commençant par le papier peint
- Il doit donc être possible de repenser le génie logiciel en s'émancipant du génie civil

Plan du chapitre

1

Pourquoi
un cours sur le
génie logiciel ?

2

Première génération
du génie logiciel

3

Bilan : échec

4

Le génie logiciel
aujourd'hui

Révolution du génie logiciel (2001)

- Nouvelle définition du génie logiciel : **Agilité**

Le génie logiciel est l'art et l'ensemble des moyens techniques, industriels et humains qu'il faut réunir pour construire, distribuer et maintenir des logiciels de qualité

- Ajoute une dimension **artisanale** au développement (cf *software craftsmanship*) en prônant qu'il ne suffit pas qu'un logiciel soit fonctionnel, mais il faut qu'il soit **bien conçu**
- Le développeur travaille comme un artisan, guidé par son talent, son expérience et ses connaissances théoriques puisées dans le **partage** et la **formation**
- Le développeur s'appuie sur l'expérience des développeurs expérimentés. Il existe aujourd'hui près de 60 ans d'expérience en génie logiciel dont il est nécessaire de s'inspirer

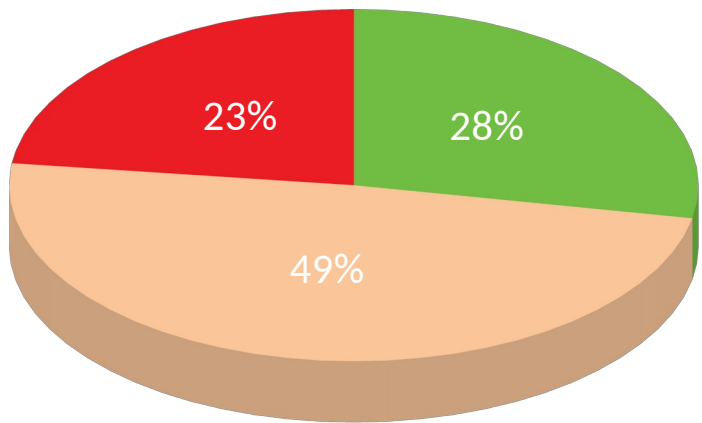
Le génie logiciel aujourd'hui

70

- Nouvelle méthode de gestion de projet
 - **Cycle itératif et incrémental**
- Nouveau paradigme de conception logicielle
 - **Conception orientée objet**
- Nouveau formalisme de modélisation
 - **UML**
- Ne reste qu'un seul métier : développeur logiciel
- Ces points sont développés dans la suite du cours

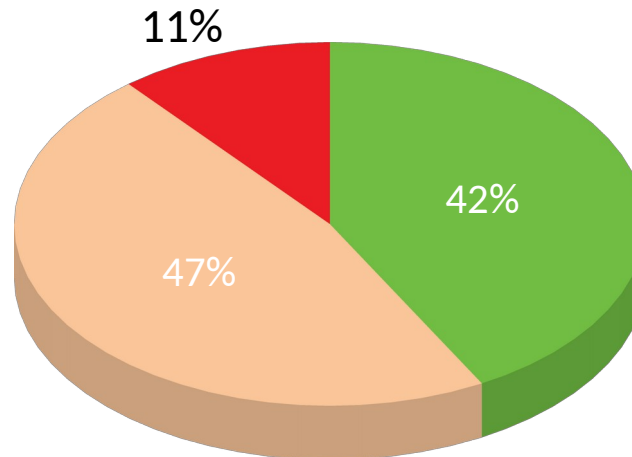
Effet sur la réussite des projets

Rappel 2000

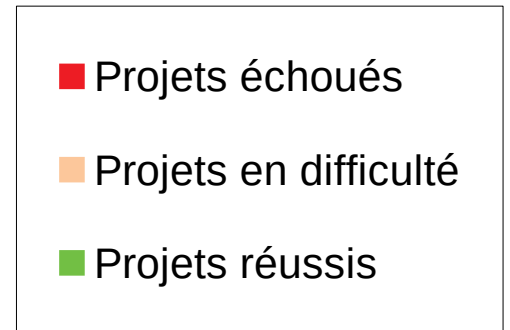


(35.000 projets étudiés)

2020



(10.000 projets étudiés)



Source : the Standish Group