

GL - 2



2.1 Le Génie Logiciel

Lydie du Bousquet

Lydie.du-bousquet@imag.fr

En collaboration avec J.-M. Favre, I. Parissis, Ph. Lalanda



Rappels

- La production logicielle est une activité complexe de façon inhérente
 - Brooks : « No silver bullet ! »
- C'est un métier d'ingénieur qui requiert des principes, des méthodes, des techniques et des outils
- Le Génie Logiciel doit prendre en compte les spécificités du logiciel pour atteindre des qualités telles que
 - La correction, la performance, la disponibilité, l'utilisabilité, ...



Plan

- **Contexte**
- Le Génie Logiciel
- Acteurs
- Activités logicielles
- Conclusion



Intérêt du génie logiciel

- Les principes et techniques de GL s'appliquent mieux aux projets de grande taille
 - Regroupant plusieurs personnes
 - Devant fournir plusieurs versions (adaptée de Parnas, 77)
 - Généralement de longue durée
- Ceci met en évidence les différences entre
 - La programmation une activité personnelle
 - Le génie logiciel une activité d'équipe au sein d'un projet



Définition d'un projet



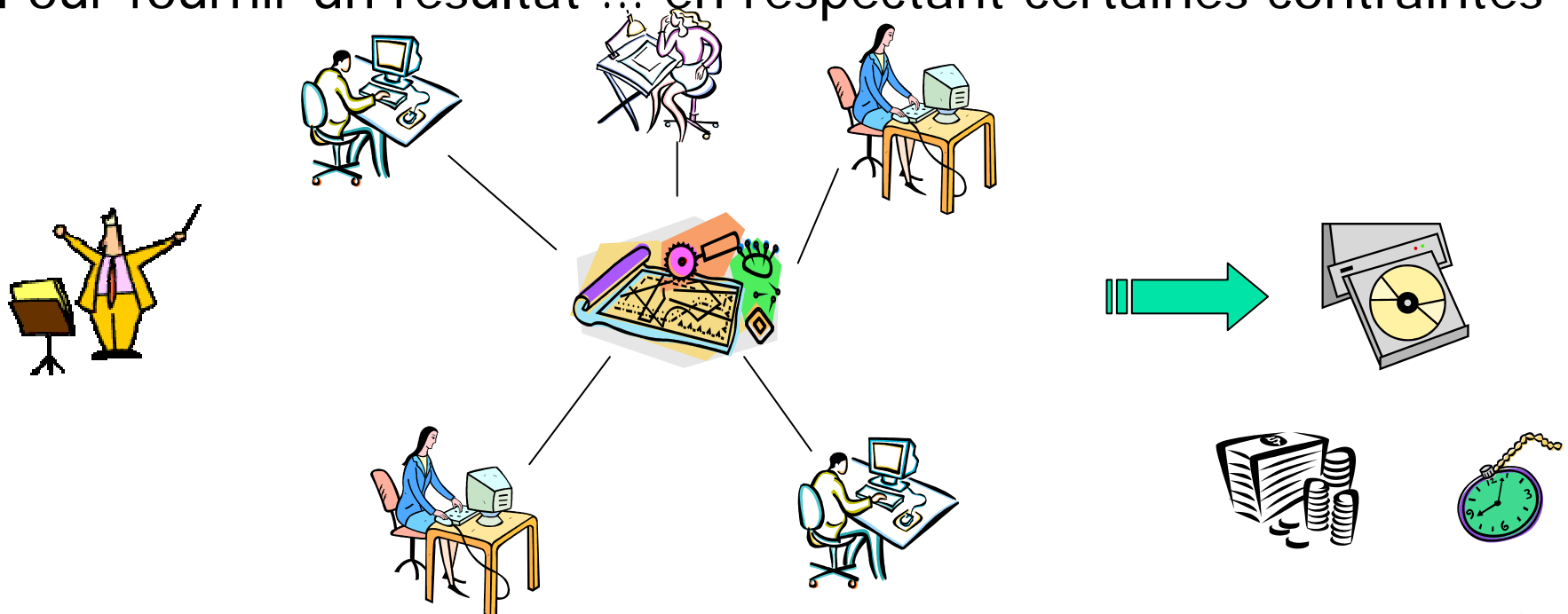
Budget

Qualité

Durée

Contexte projet

De nombreuses personnes n'ayant pas les mêmes objectifs
À coordonner, qui doivent communiquer
Pour fournir un résultat ... en respectant certaines contraintes



Des projets variés

- Logiciel commercialisé par une société
 - Dirigé par le marketing, les commerçants, les dirigeants
 - Relations souvent informelles
 - Liens et cahier des charges pas toujours très formalisés
 - Beaucoup de réactivité



Marketing,
Commerce,
Direction



Équipes de développement

Des projets variés - suite

- Logiciel construit sur demande
 - Spécifié et payé par un client
 - Réalisé par une entreprise informatique
 - Relations très formelles régies par la loi
 - Importance du contrat (avec spécifications, pénalités, ...)



Client

Maîtrise d'ouvrage



Manager

Maîtrise d'oeuvre



Équipes de développement

Des projets variés - suite

- Projet système
 - L'informatique est une composante
 - Les exigences sont déterminées au niveau système
 - Relations souvent assez formelles



Équipes de développement



Responsable de lot informatique



Manager



Responsable de lot matériel



Apport du GL

- En fonction de leur nature, les projets ont des besoins très différents
 - Logiciel commercial : lien avec les clients
 - Logiciel exploratoire : souplesse
 - Logiciel gros et complexe : spécifications stables
- Le travail d'un ingénieur en informatique est de choisir la bonne approche de GL en fonction du projet
 - Une bonne approche aidera à bien livrer le produit final
 - Le produit souhaité et le contexte déterminent la bonne approche



Plan

- Notion de projet
- **Le Génie Logiciel**
- Acteurs
- Activités logicielles
- Conclusion

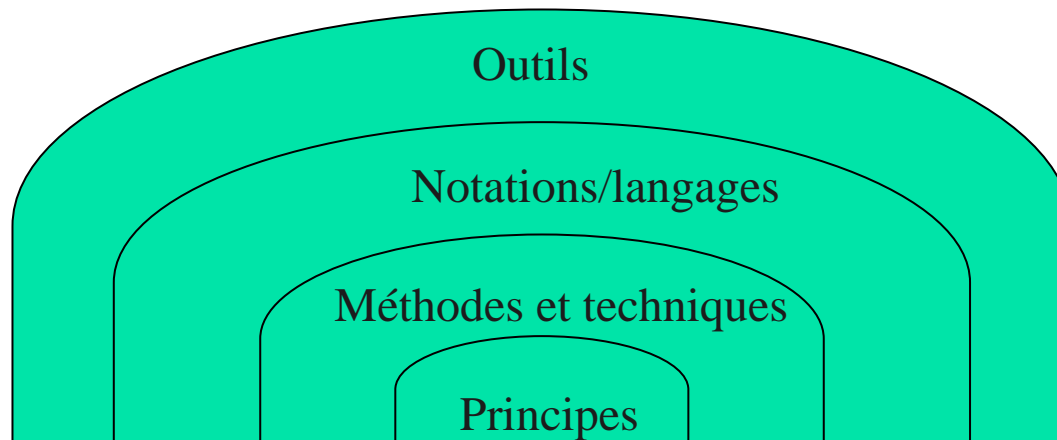


Définition du génie logiciel

- Le génie logiciel est une démarche d'ingénierie qui traite tous les aspects de la production de logiciels
 - du cahier des charges jusqu'aux activités de maintenance
 - dépasse le cadre purement technique
- Le GL vise à assurer la production de logiciels en respectant les aspects économiques
 - <Coût, échéance, qualité>

Principes et techniques

- Le GL repose sur un ensemble de principes
 - mis en œuvre par des méthodes, des techniques et des outils



Rational Rose

UML

Objet

**Encapsulation,
Masquage, ...**

D'après Ghezzi, 93

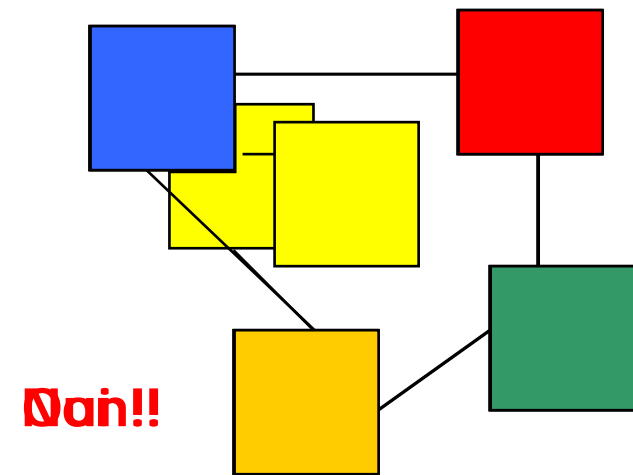


Principe : la rigueur

- Ingénieur = rigueur + précision
- Les activités logicielles doivent être réalisées rigoureusement
 - Suivi de processus adaptés
 - Utilisation correcte des techniques adaptées
 - Fourniture des livrables prévus (documents, modèles, code)
 - Validation de toutes les livraisons
 - Attitude professionnelle au sein d'une équipe, etc.
- Deux remarques
 - La rigueur ne tue pas la créativité
 - Rigueur n'est pas égal à formel (=précision)

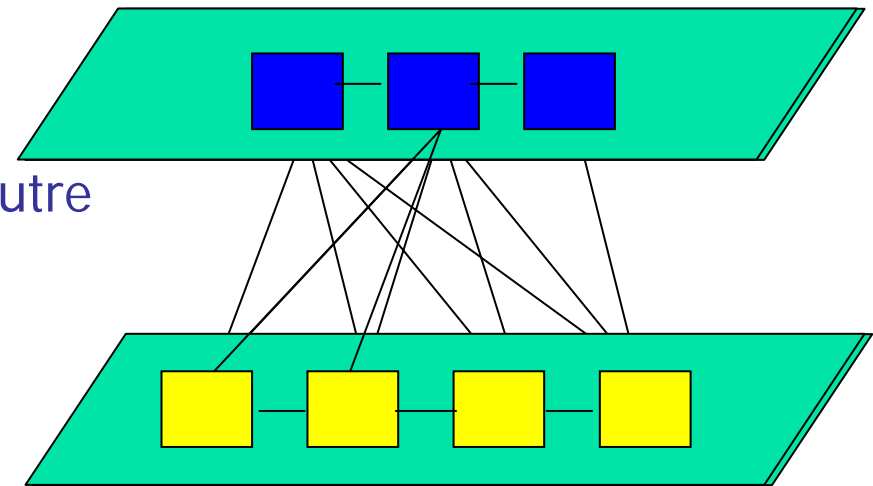
Principe : la modularité

- Le principe est de remplacer le problème initial par des modules de moindre complexité
 - chaque module traite une partie du problème
 - ils sont compréhensibles, homogènes, indépendants
 - les modules sont reliés entre eux
- On recherche
 - Faible couplage
 - Forte cohésion



Principe : l'abstraction

- Organiser les informations (ou les modules) suivant différents niveaux d'organisation
 - Définition de niveaux de généralisation
 - A un niveau donné, on ne considère que les informations ayant le même « niveau sémantique »
 - Un niveau doit être compréhensible, homogène, complet
- On recherche
 - Des niveaux bien découplés
 - Le passage aisé d'un niveau à l'autre



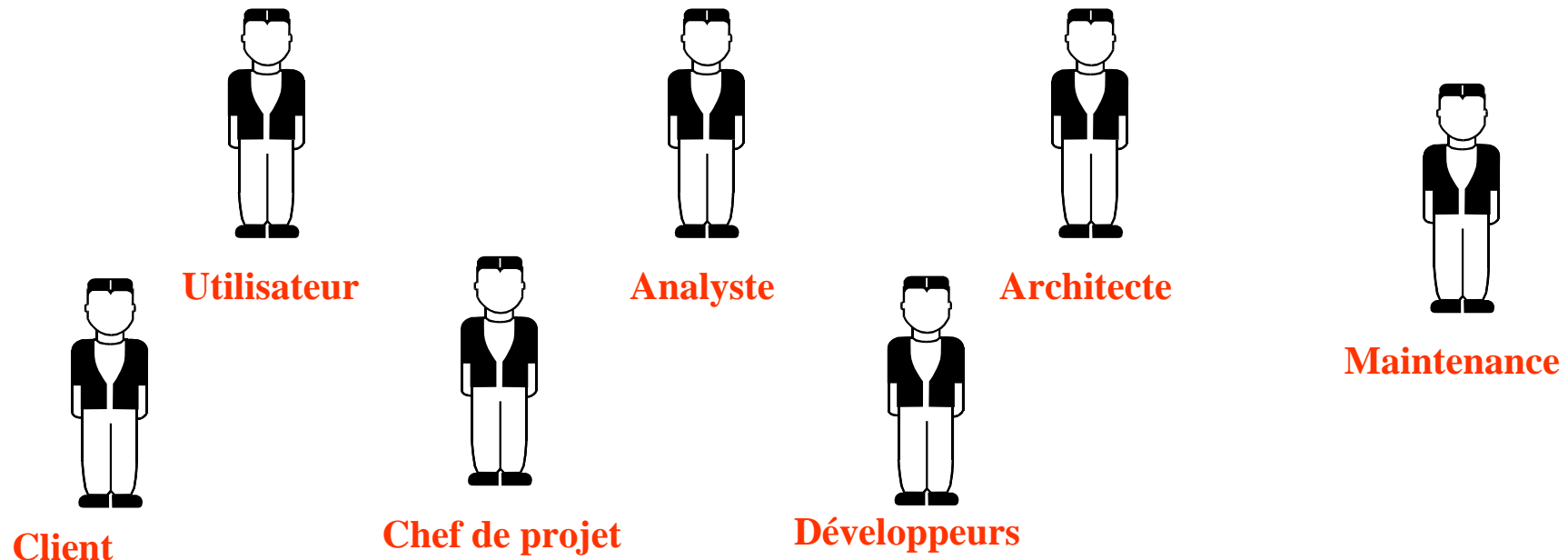


Principe : séparation des préoccupations

- Le principe est de se concentrer sur un seul aspect du problème à la fois et le traiter de façon indépendante
- Exemples
 - Séparation des rôles des différents acteurs
 - Séparation des phases de développement
 - Se concentrer uniquement sur la sécurité
- Il faut choisir des aspects suffisamment indépendants
 - Les aspects trop liés doivent être traités ensemble
- On peut parfois automatiser la réunion des aspects séparés (exemple AOP)

Séparation des préoccupations - exemple

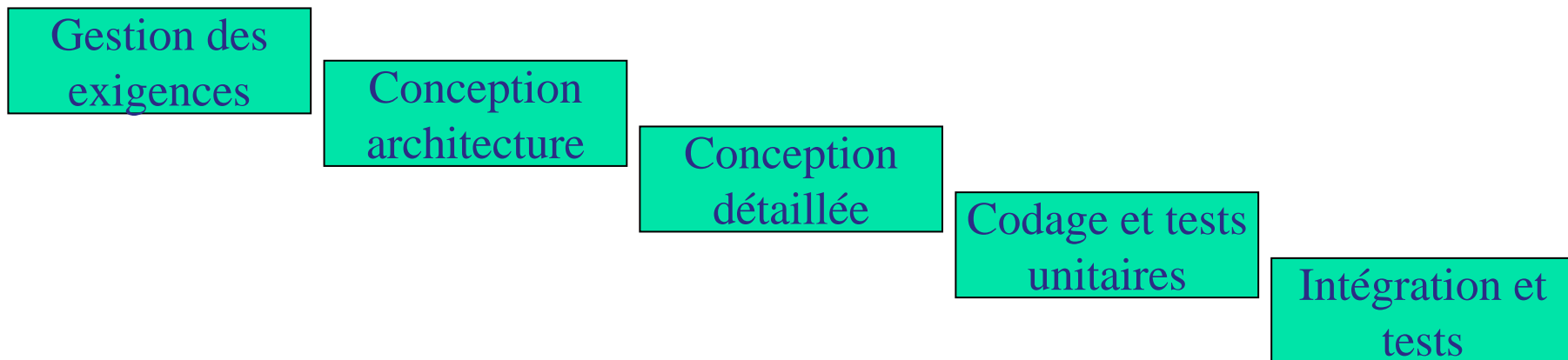
- Identification et définition de rôles différents au sein d'un développement logiciel
 - Chaque acteur a une préoccupation différente





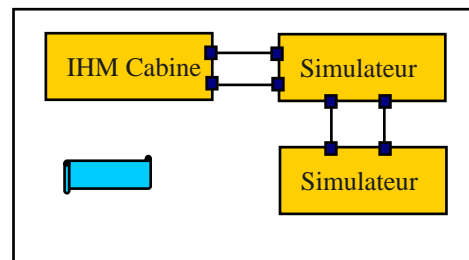
Séparation des préoccupations - exemple

- Identification et définition d'activités différentes lors d'un développement logiciel (**séparation temporelle**)
 - Chaque activité a une focalisation spécifique
 - Permet, entre autre, la séparation entre spécification et implem.

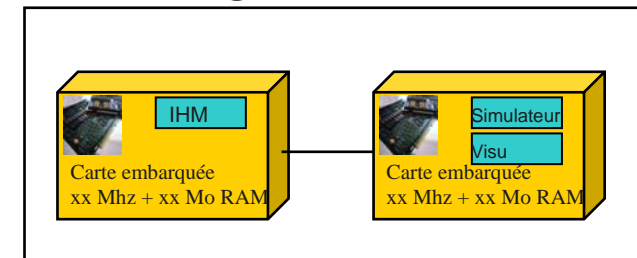


Séparation des préoccupations - exemple

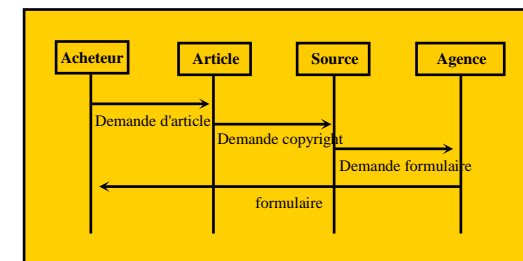
- Identification et construction de vues complémentaires de la structure du logiciel
 - Vue logique
 - Vue dynamique
 - Vue physique



Vue logique



Vue physique



Vue dynamique



Remarque

- Ces principes reposent sur la décomposition des problèmes en sous problèmes plus petits (et moins complexes)
 - « diviser pour régner »
 - Idée simple
- Le problème est ensuite de recomposer
 - C'est le défi majeur du GL (moins simple !)



Plan

- Notion de projet
- Le Génie Logiciel
- **Acteurs**
- Activités logicielles
- Conclusion




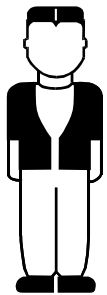
Acteurs

- Différents acteurs interviennent
 - Utilisateurs
 - Clients
 - Manager/ingénieur d'affaires
 - Chef de projet
 - Architecte
 - Analyste
 - Développeur
 - Maintenance
- Ils n'ont pas les mêmes intérêts



Utilisateur

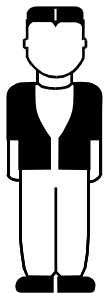
- 
- Fonctionnalités
 - Facilité d'utilisation
 - Performance
 - Sécurité
 - Robustesse





Client/Maîtrise d'ouvrage

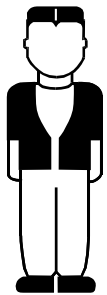
- Respect des coûts et des délais
- Garantie de fonctionnement
- Maîtrise des risques (technologiques)
- Pérennité
- Efficacité
- Coût raisonnable





Manager/Maîtrise d'oeuvre

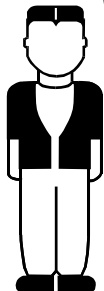
- Relation client
- Bénéfice
- Responsabilité limitée/maîtrisée





Chef de projet/Maîtrise d'oeuvre

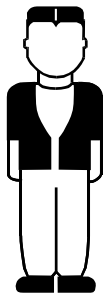
- Simplicité de l'ensemble
- Possibilité d'évaluer les progrès (incréments)
- Bonne expression des contraintes
- Technologies maîtrisées
- Structuration correspondant à ses équipes
- Réutilisation des composants internes
- Maîtrise des risques
- Maîtrise des coûts et des délais





Architecte

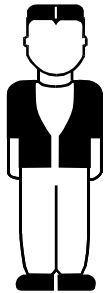
- Cohérence de l'architecture
- Simplicité
- Pérennité
- Réutilisation de l'existant
- Technologies maîtrisées





Développeur

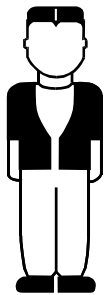
- Simplicité de l'interface de ses composants
- Technologie connue ou « d'avenir »
- Pas trop de contraintes imposées sur ses composants





Analyste

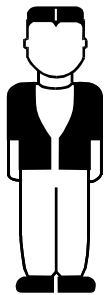
- Accès aux utilisateurs
- Clarté du domaine
- Indépendance technologique



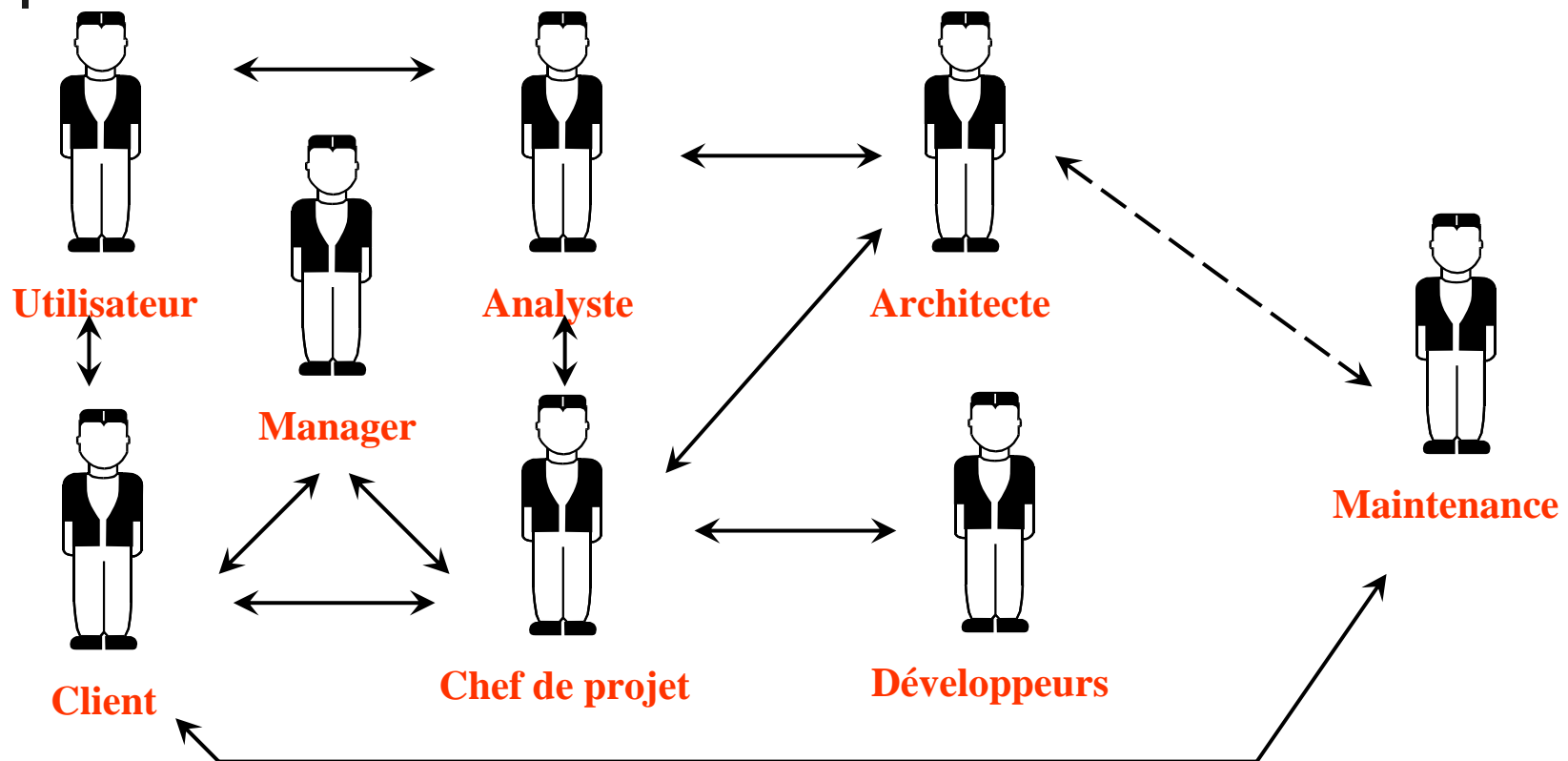


Maintenance

- Facilité de modification
- Isolation des composants
- Existence d'interfaces d'administration
- Technologies connues et pérennes



Relations



Comment organiser le travail entre ces acteurs ?



Plan

- Notion de projet
- Principes du Génie Logiciel
- Acteurs
- **Activités logicielles**
- Conclusion



Activités

- Le développement comprend un ensemble d'activités
 - La gestion des exigences
 - La spécification
 - La conception
 - L'implantation
 - La validation
 - L'intégration
 - Le déploiement
 - La maintenance
- Elles fournissent différents produits logiciels (documents, modèles, code, ...)

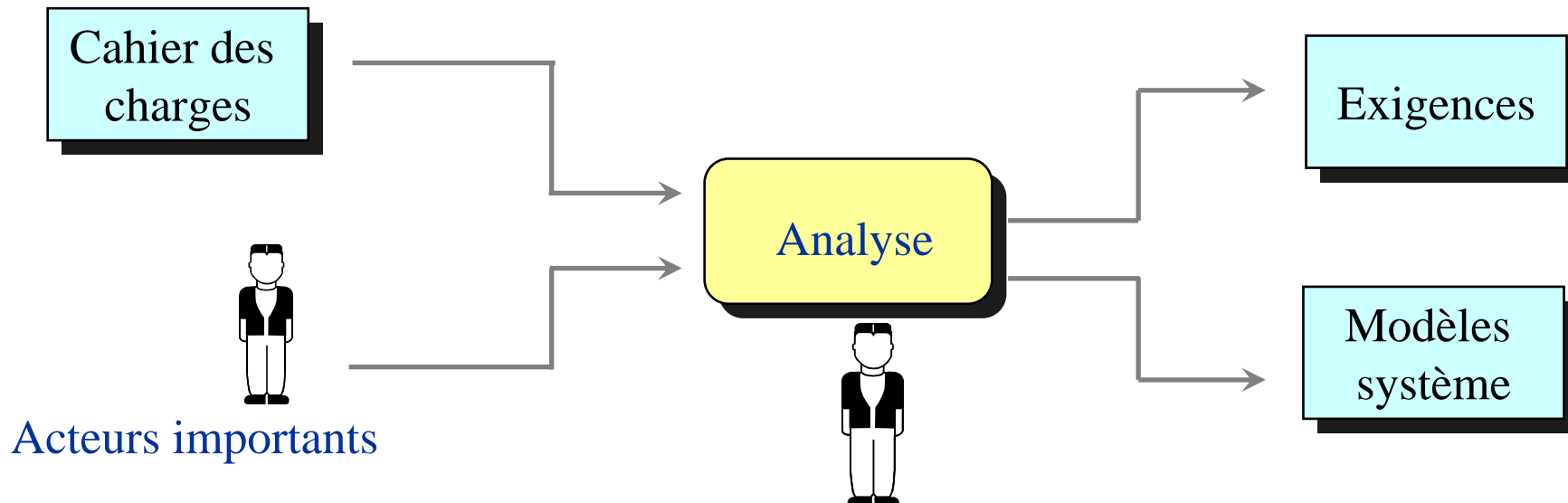


Activités permanentes

- Il existe également des activités permanentes
 - La documentation
 - La gestion de projet
 - La gestion de la qualité
 - La gestion de la sous-traitance
 - etc.

Analyse des besoins/spécification

- Objectif de cette phase :
 - identifier ce que veut le client et les contraintes
 - modéliser sous forme d'exigences et de modèles



Exemple d'exigences

Use case 1 : Emprunt d'une revue

Description

Ce « use case » montre comment un utilisateur peut imprimer un article dès lors qu'il a clairement identifié l'article qui l'intéresse et payé les droits d'auteur (copyrights).

Acteurs

L'utilisateur

L'imprimante

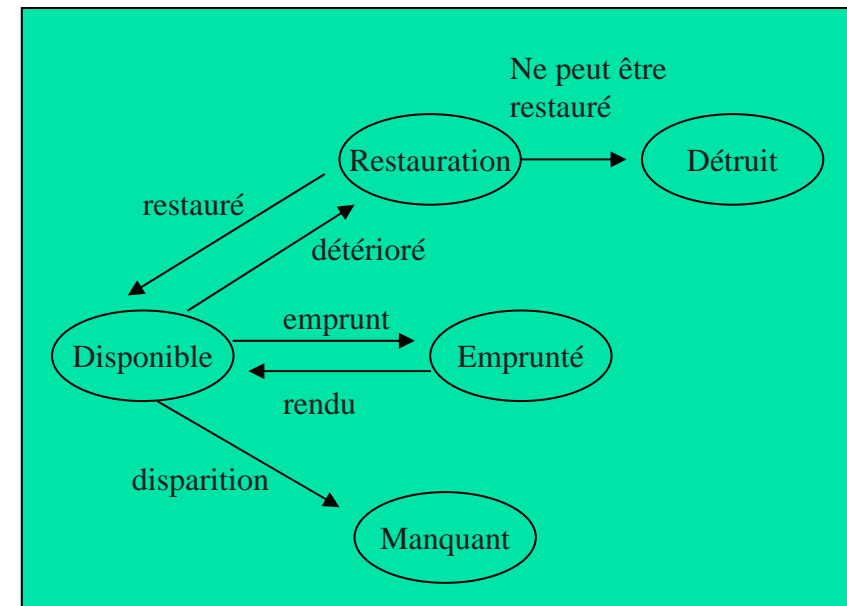
Flots d'événements

1. Accès à l'article

- 1.1 L'utilisateur spécifie le titre de l'ouvrage recherché
- 1.2 L'utilisateur valide sa recherche
- 1.3 Si le système indique « requête incomplète », retour à 1.1
- 1.4 Le système présente à l'utilisateur la liste des réponses

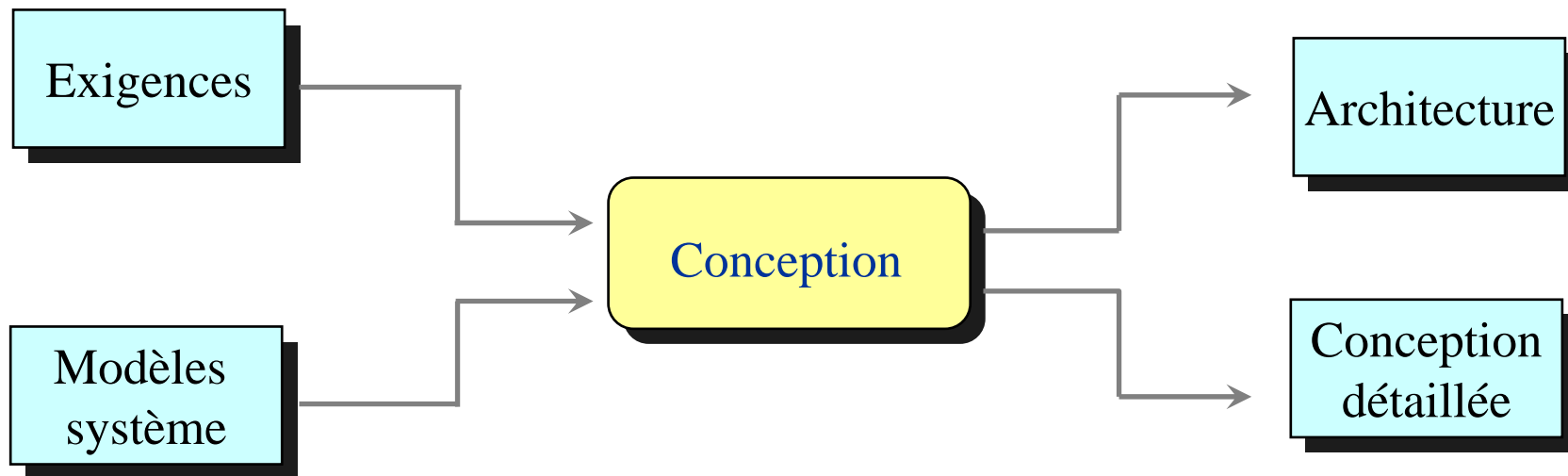
2. Impression de l'article

- 2.1 L'utilisateur sélectionne l'article qui l'intéresse
- 2.2 L'utilisateur clique sur l'icône d'impression
(on suppose ici que l'imprimante est connectée et fonctionne)
- 2.3 L'utilisateur quitte le logiciel

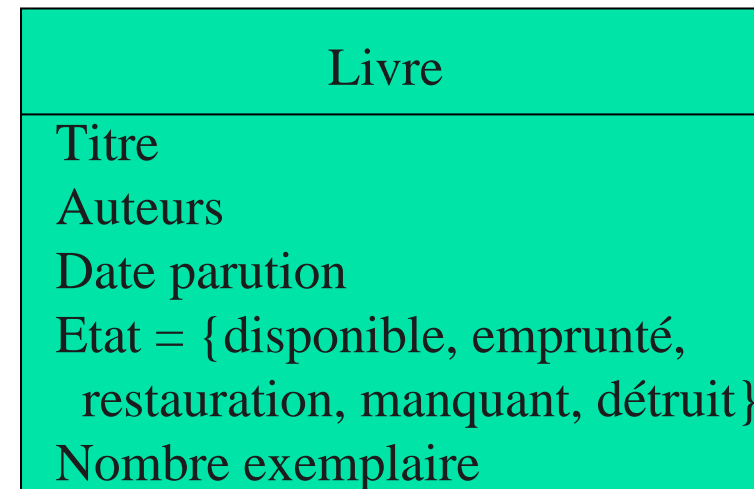
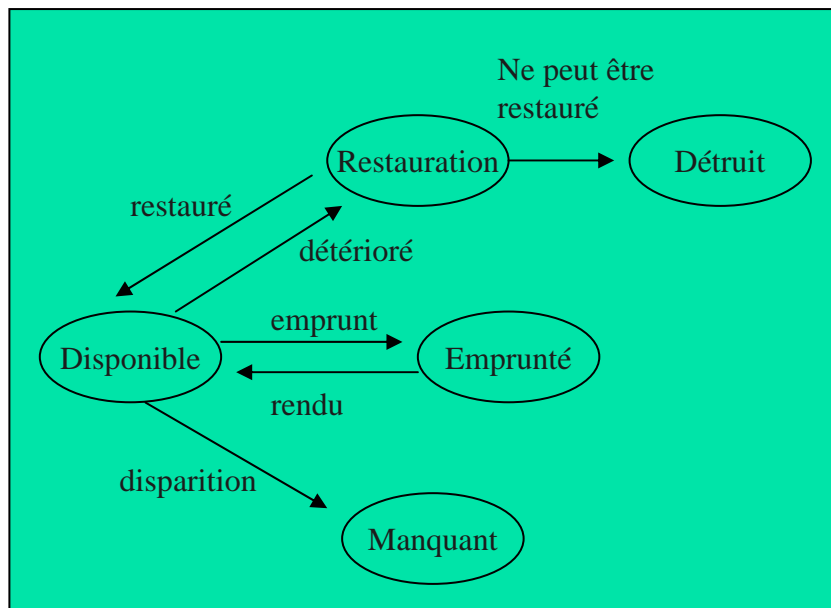
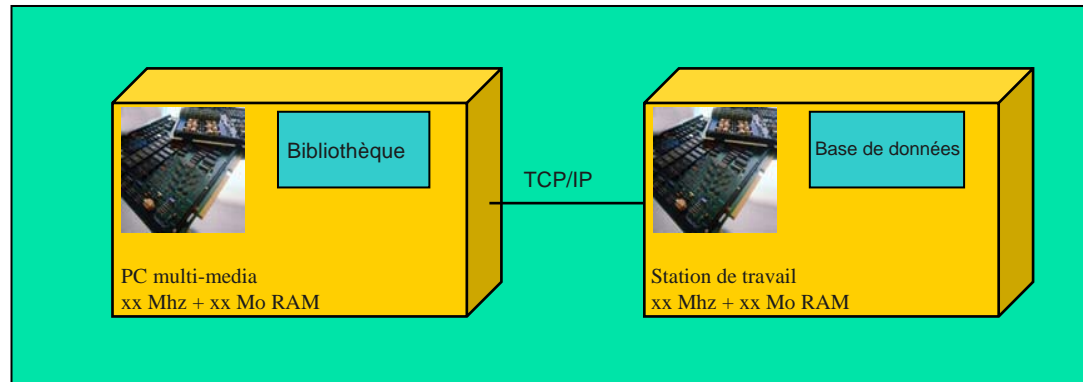


Conception

- Objectif de cette phase :
 - définir une organisation logique du code
 - modéliser la résolution du problème posé par l'analyse

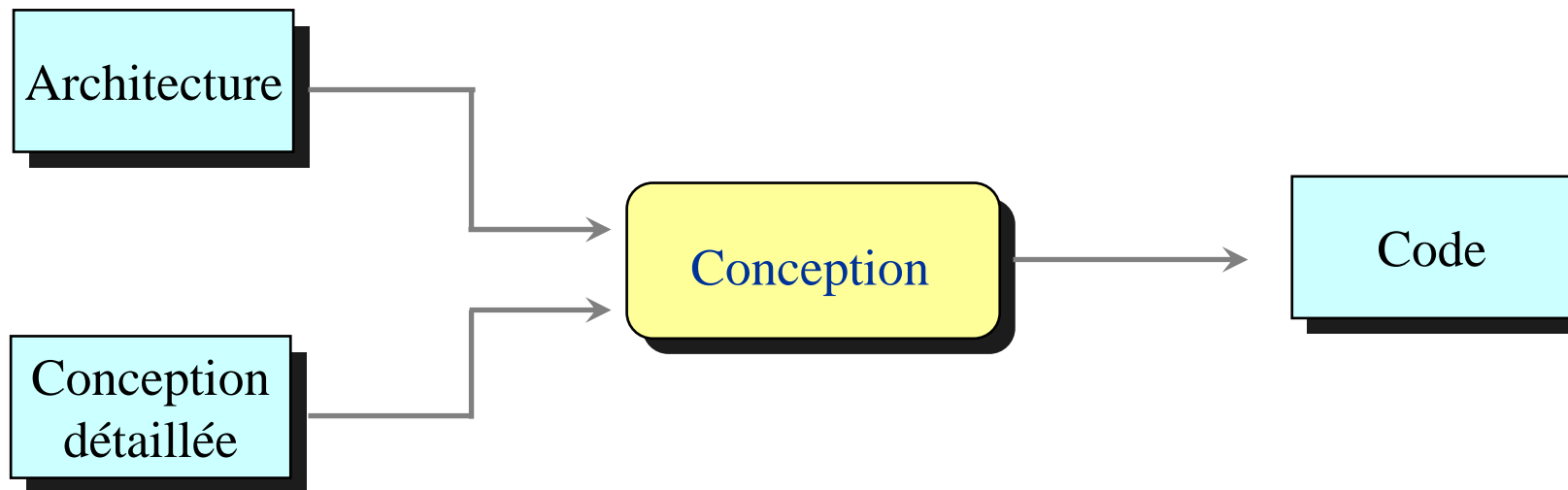


Conception – exemples de modèles



Implantation

- Objectif de cette phase :
 - production d'un exécutable réalisant la conception
 - ajout d'optimisation si nécessaire





Implantation – exemples de code

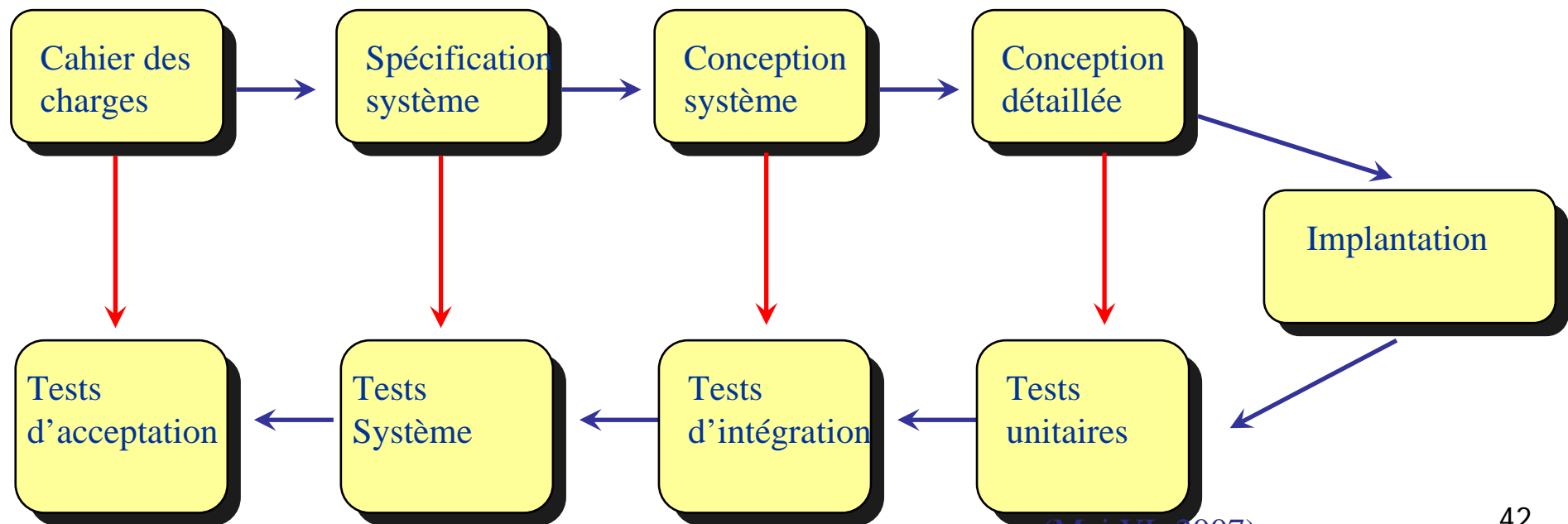
```
// exception pour lecture de fichier
FileOutputStream fos = null;
try {
    fos = new FileOutputStream(...);
    fos.write(a);}
catch (IOException e) {
    e.printStackTrace(); }
finally{
    if(fos != null) fos.close();
}
```

```
(defun mklist (obj)
  (if (listp obj) obj (list obj)))
```

```
LES AX,a                ; Valeur a
MOV DX,ES
LES CX,b                ; Valeur b
MOV BX,ES
MOV SI,AX
MOV DI,DX
MUL CX
PUSH AX
PUSH DX
    MOV AX,SI
    MUL BX
    MOV BX,AX
    MOV AX,DI
    MUL CX
    MOV CX,AX
POP DX
POP AX
ADD DX,BX
ADD DX,CX
```

Validation

- Objectif de cette phase :
 - tester/valider les différents artefacts logiciels
 - activité transverse/permanente



(Maj YL 2007)

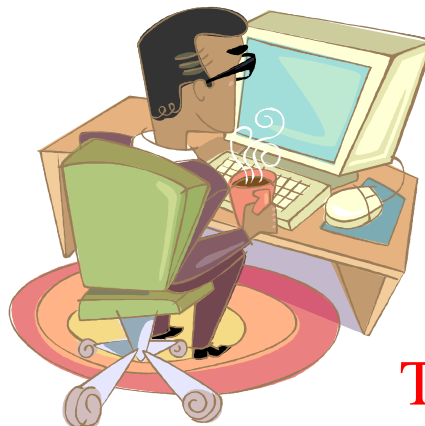
Validation – exemples de techniques



Audit



Revue/inspection



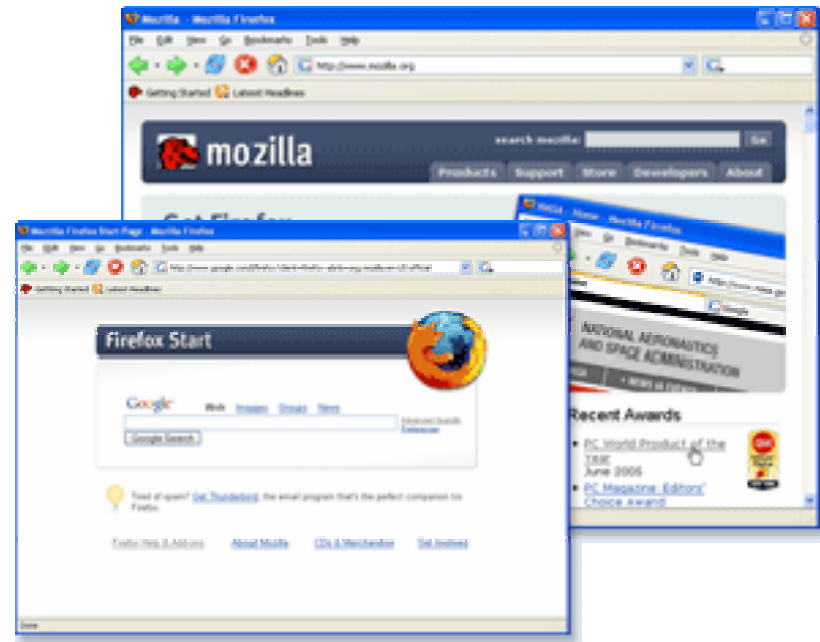
Test



Intégration et déploiement

- Combiner les modules et vérifier le produit dans son ensemble
 - essentiellement un problème de test
- Le déploiement correspond à l'installation du produit fini chez le client
 - Importance du web aujourd'hui
 - De plus en plus réalisé par les clients ou les utilisateurs
- Acteurs : programmeurs, ingénieurs du client, utilisateurs

Intégration et déploiement – exemples





Maintenance

- Il s'agit d'apporter des modifications à un logiciel existant
 - C'est la phase la plus coûteuse (70% du coût total)
- Types
 - Maintenance corrective
 - Maintenance prédictive
 - Maintenance évolutive



Problème : organisation

- Les activités s'enchaînent plus ou moins bien ...
 - on ne dispose pas de toutes les informations au bon moment
 - les besoins et les contraintes évoluent
 - les décisions à prendre sont parfois trop complexes (stratégie « do and see »)
 - les bonnes personnes ne sont pas toujours disponibles
- Définition de cycle de vie
 - Itératifs aujourd'hui



Problème : communication

- La communication entre les différents acteurs et les différentes phases est complexe
- Documents en langue naturelle (français/anglais)
 - Peu de possibilités de traitement automatique
- Utilisation de modèles, mais
 - pas de formalisme général (pour toutes les activités et tous les domaines)
 - peu de traçabilité entre les modèles
 - complétude et vérification des modèles ?



Plan

- Notion de projet
- Le Génie Logiciel
- Acteurs
- Activités logicielles
- **Conclusion**



Synthèse

- Le génie logiciel est une démarche d'ingénierie et poursuit les objectifs suivants :
 - prédictible
 - reproductible
 - évaluable
- Le GL repose sur deux grands principes
 - La rigueur
 - La décomposition des problèmes



Synthèse – suite

- Premiers apports du GL
 - définition de différents acteurs
 - définition de différentes activités
- Premiers problèmes
 - Enchaînement des ces activités
 - Communication entre ces acteurs (et ces activités)

Conclusion

