

Analyse et Conception de Logiciels

Sujets abordés

- développement logiciel : les concepts, les méthodes
- les activités dans le développement logiciel
- modélisation avec UML
- techniques de test

<http://arche.univ-lorraine.fr/course/view.php?id=71642>

Sources

- Software Engineering, 9th ed. - Ian Sommerville
- Object-Oriented Software Engineering: Practical Software Development using UML and Java - Timothy C. Lethbridge, Robert Laganière
- UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd ed. - Martin Fowler
- UML 2 - de l'apprentissage à la pratique - Laurent Audibert, Ellipses, 2009
- UML 2 par la pratique - Pascal Roques, Eyrolles, 2009
- plus de références sur la page du cours

- | -

Introduction au processus de développement logiciel

Comment produire
efficacement de *bons*
logiciels?

Le génie logiciel (software engineering)

*Software engineering is concerned with
cost-effective software development.*

I.Sommerville

Le génie logiciel (software engineering)

*Le processus visant la **résolution de problèmes** posés par un **client** par le développement systématique et l'évolution de systèmes logiciels de grande taille et de haute qualité en **respectant les contraintes** de coûts, de temps, et autres.*

T. Lethbridge, R. Laganière

F.A.Q.

- Qu'est-ce qu'un logiciel (software) ?
 - Un programme informatique et la documentation associée.
- Quels sont les attributs d'un bon logiciel ?
 - Un bon logiciel doit fournir à l'utilisateur les fonctionnalités et performances demandées et doit être maintenable, fiable et utilisable.
- Qu'est-ce que l'ingénierie logicielle (software engineering) ?
 - Une discipline concernée par tous les aspects de la production de logiciels.

F.A.Q.

- Quelles sont les activités principales dans l'ingénierie logicielle ?
 - spécification, développement, validation et évolution (des logiciels)
- Quels sont les principaux défis du génie logiciel?
 - diversité croissante, délais de livraison réduits, développement de logiciels de confiance

F.A.Q.

- Quels sont les coûts d'ingénierie logicielle ?
 - Environ 60% des coûts sont les coûts de développement, 40% des coûts liés aux test.
 - Pour les logiciels spécifiques, les coûts d'évolution dépassent souvent les coûts de développement.
- Quelles sont les meilleures techniques et méthodes de génie logiciel ?
 - différentes techniques sont appropriés pour différents types de système

Attributs des logiciels

- **Efficacité** : Aucun gaspillage de ressources (mémoire, temps de calcul, ...).
- **Fiabilité et sécurité** : Les pannes ne doivent pas engendrer des dommages au système. Les intrus ne doivent pas pouvoir accéder et endommager le système.
- **Maintenabilité** : Le logiciel doit pouvoir évoluer pour correspondre aux demandes de changement.
- **Acceptabilité** : Apprentissage aisé, facilité d'utilisation, compatibilité avec les autres systèmes utilisés par les clients.

Les parties impliquées

- **Utilisateurs**

Ceux qui se servent du logiciel

- **Clients**

Ceux qui paient pour le logiciel

- **Développeurs**

Ceux qui conçoivent le logiciel

- **Managers**

Ceux qui supervisent la production du logiciel

Tous ces rôles peuvent être remplis par la même personne

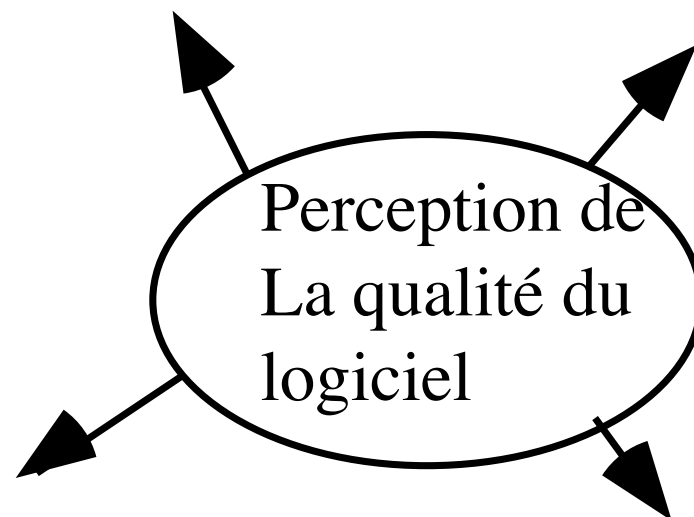
La qualité du logiciel...

Client:

Résoud le problème
à un coût acceptable

Utilisateur:

Facile à apprendre,
utile et efficace

**Développeur:**

Facile à concevoir,
à maintenir, à réutiliser

Gestionnaire:

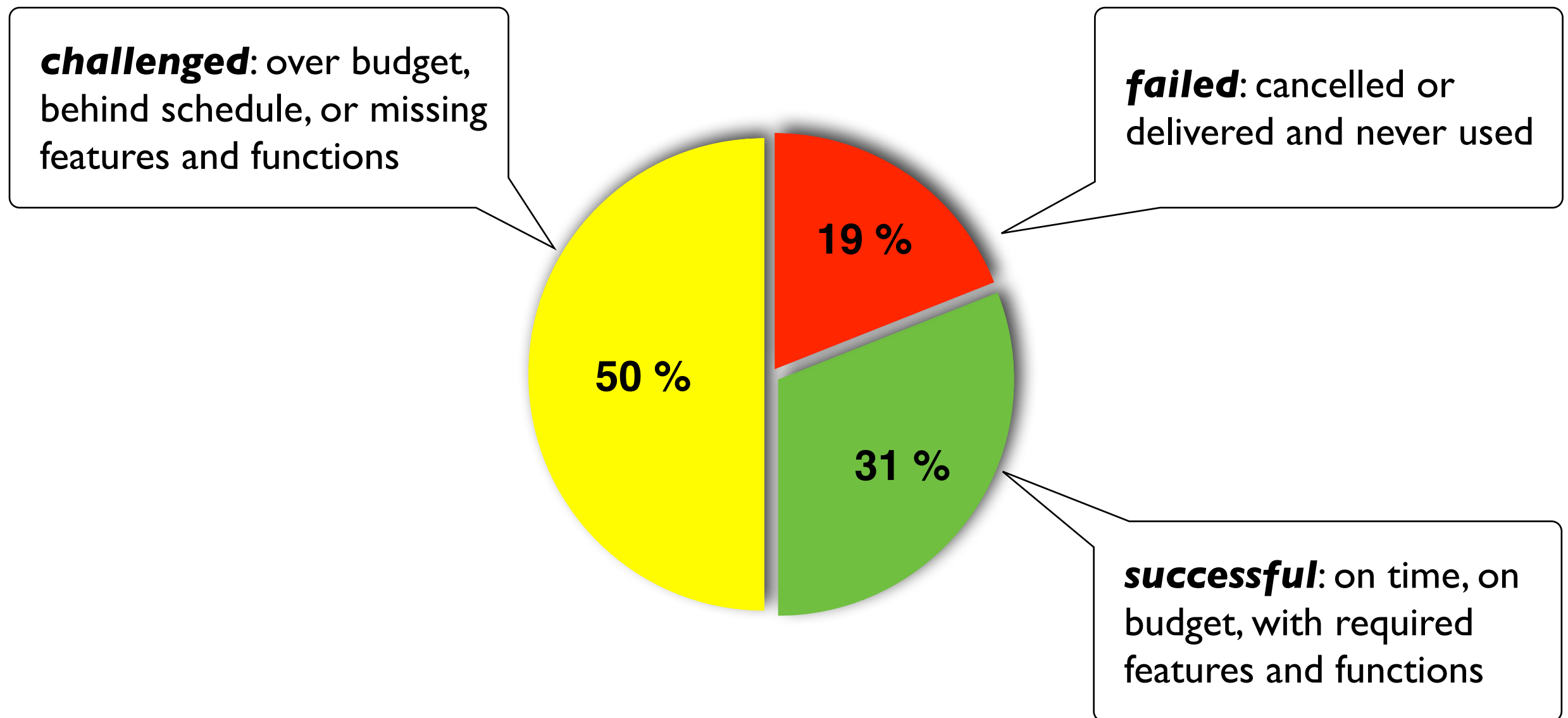
Se vend bien,
satisfait les clients,
peu coûteux à développer

Le génie logiciel

Concerne *tous les aspects du développement logiciel* de sa spécification jusqu'à la gestion de ses évolutions

- utilise des méthodes et théories appropriées pour résoudre des problèmes sous des contraintes organisationnelles et financières
- non seulement un processus de développement

Pourquoi c'est important?



*CHAOS 2020: Beyond Infinity Overview. January'2021,
QRC by Henny Portman*

Pourquoi c'est important?

- produire *efficacement* des logiciels *robustes*
- à long terme il est *moins couteux* d'utiliser des techniques et méthodes d'ingénierie logicielle

A retenir

- Le génie logiciel concerne tous les aspects liés à la production de logiciel
- Principaux attributs d'un logiciel: maintenabilité, fiabilité et sécurité, efficacité, acceptabilité
- Activités communes: spécification, développement, validation, évolution
- Les principes fondamentaux de l'ingénierie logicielle s'appliquent à tous les types de développement
- Les méthodes et outils d'ingénierie doivent être choisis en fonction des systèmes développés

Modèles du processus de développement logiciel

- **Le cycle (processus) de développement :**
un ensemble structuré d'activités nécessaires au développement d'un logiciel
 - ↳ Différents processus de développement mais comportant des activités communes
- **Modèle de processus de développement :**
une représentation abstraite d'un processus (par rapport à une certaine perspective).

Activités

- **Spécification**
 - ▶ définir le comportement du système;
- **Conception et implémentation**
 - ▶ définir l'organisation et implanter le système;
- **Validation**
 - ▶ vérifier que ça correspond à la demande;
- **Evolution**
 - ▶ modifier le système en fonction des demandes.

Cycle de développement

- Activités (avec sous-activités) et leur ordonnancement
- Produits (résultant des activités)
 - ▶ **ex:** modèle de l'architecture
- Rôles (et responsabilités)
 - ▶ **ex:** manager projet, programmeur, ...
- Pre- et post-conditions
 - ▶ **ex:** tous les besoins validés avant la conception, modèles UML examinés après la conception, ...

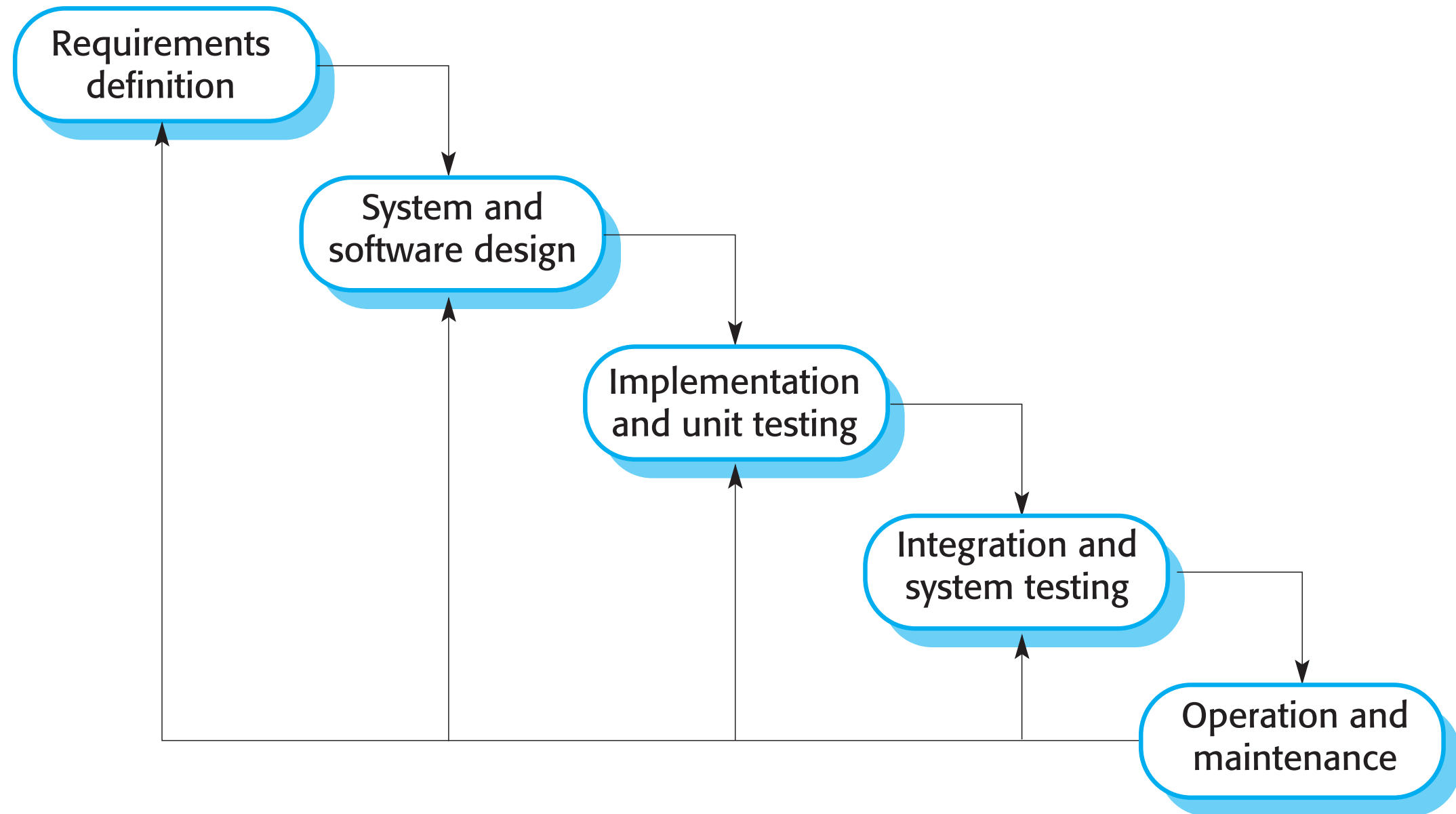
Plan-driven vs. Agile

- **Plan-driven** : les activités sont planifiées en avance et le progrès est évalué par rapport au plan.
- **Agile** : le planning est incrémental et plus facilement adaptable aux changements (demandés par les clients).

Modèles

- **Cascade** : phases distinctes de spécification et développement
- **Développement incrémental** : la spécification, le développement et la validation se superposent.
- **Modèle orienté réutilisation** : assemblage à partir de composants existants.

Cascade (Waterfall)



Waterfall

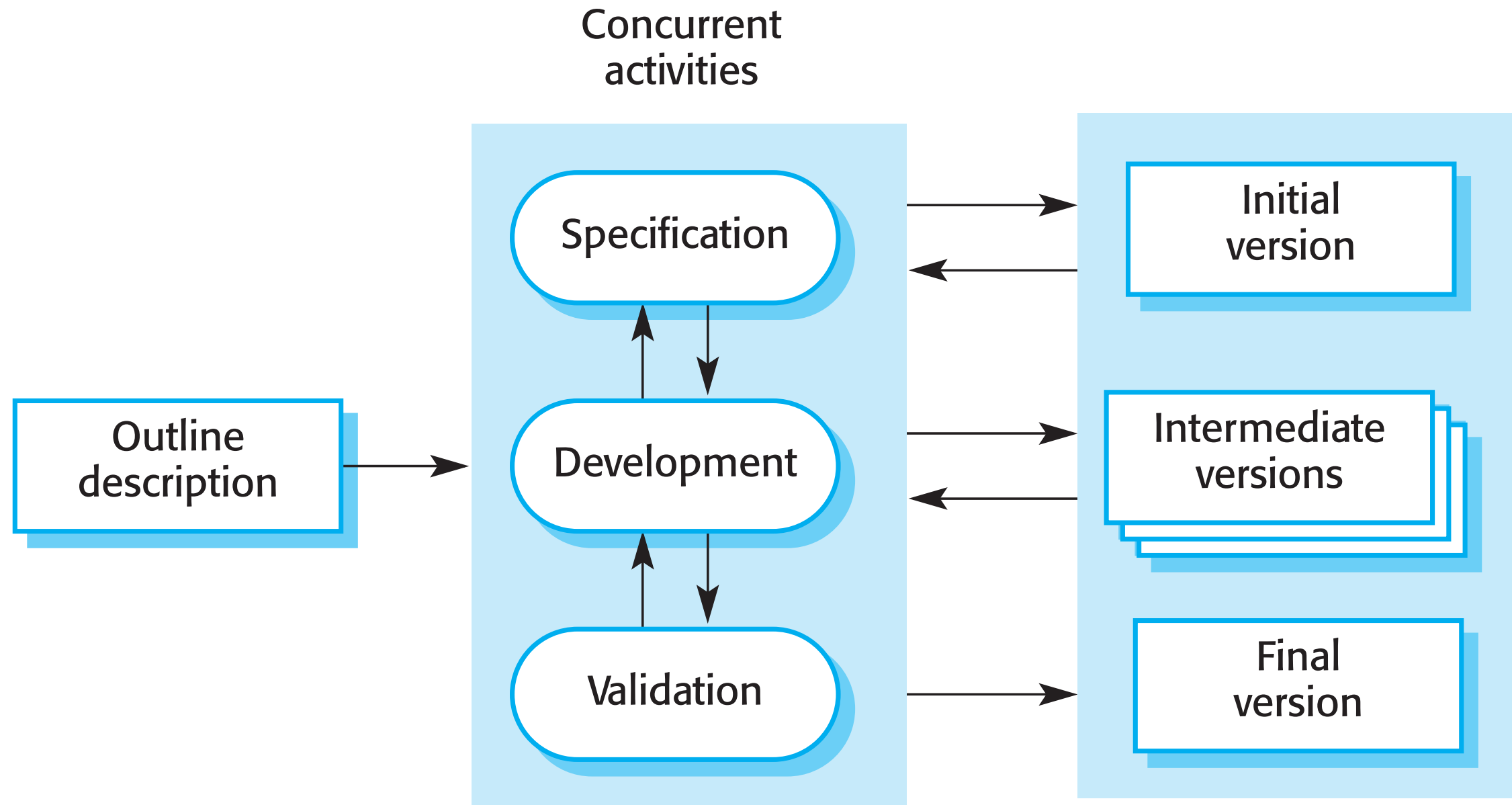
- Inconvénients -

- modifications difficiles à intégrer après le début du processus (chaque phase doit être terminée avant de passer à la suivante)
- difficile à dire si le projet est en bonne voie (décalages dans le planning difficiles à anticiper)

Waterfall

- **Découpage strict des phases**
 - ▶ adapté quand les besoins sont claires et relativement stables
- **Utilisé pour coordonner le développement**
 - ▶ adapté quand le système est développé sur plusieurs sites

Développement incrémental



Développement incrémental

- Avantages -

- Le coût pour l'intégration des modifications (demandées par le client) est réduit
 - ▶ moins d'analyse et documentation à refaire
- Plus facile à obtenir du feedback client par rapport au travail déjà réalisé.
 - ▶ les clients peuvent commenter les demos
- Livraison et déploiement chez le client plus rapides
 - ▶ les clients peuvent bénéficier du produit plus rapidement

Livraison incrémentale

- Avantages -

- Des fonctionnalités disponibles plus tôt
- Incréments utilisés comme prototype pour la découverte des besoins
- Fonctionnalités prioritaires testées davantage
- Risque réduit d'échec global du projet

Développement incrémental

- Inconvénients -

- Le processus n'est pas visible (pour les managers)
 - ▶ la production de documents pour le nombreuses versions (delivrables) n'est pas rentable
- La structure du système a tendance à se dégrader avec l'ajout de nouveaux incréments
 - ▶ l'intégration de nouvelles fonctionnalités devient de plus en plus difficile

Développement incrémental

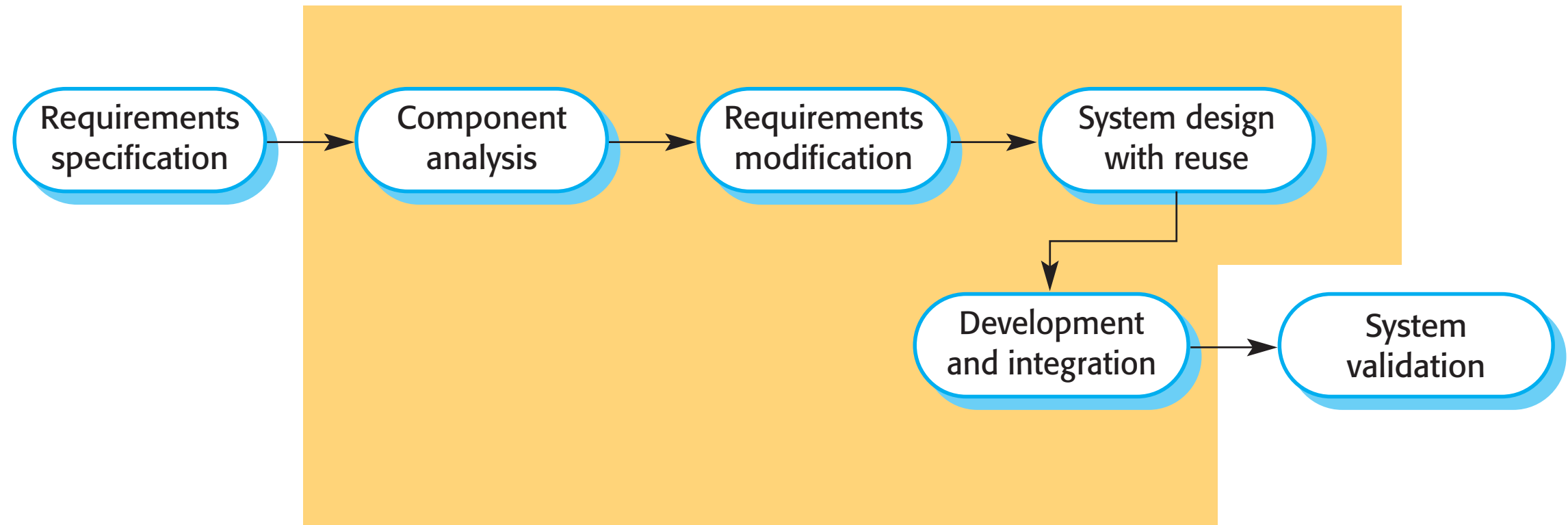
- Problèmes -

- Problèmes liés au management
- Problèmes liés aux contrats
- Problèmes liés à la validation
- Problèmes de maintenance
- Problèmes liés aux technologies

Modèle orienté réutilisation

- Basé sur le développement de systèmes à partir de composants existants - COTS (Commercial-off-the-shelf).
- Approche standard pour beaucoup de systèmes d'information

Modèle orienté réutilisation



Modèle orienté réutilisation

- Services Web (invocables à distance)
- Collections d'objets (package)
- Software stand-alone (COTS) configurable pour un environnement particulier

Les activités du processus de développement logiciel

Activités

- **Analyse des besoins** ➡ développement d'une spécification des besoins (cahier de charges)
- **Conception et implémentation** ➡ transformation du cahier de charges en un logiciel exécutable
- **Validation & vérification** ➡ vérification de la conformité avec la spécification et les besoins
- **Evolution** ➡ adapter le logiciel pour prendre en compte de nouveaux besoins

➡ Organisées en fonction du modèle

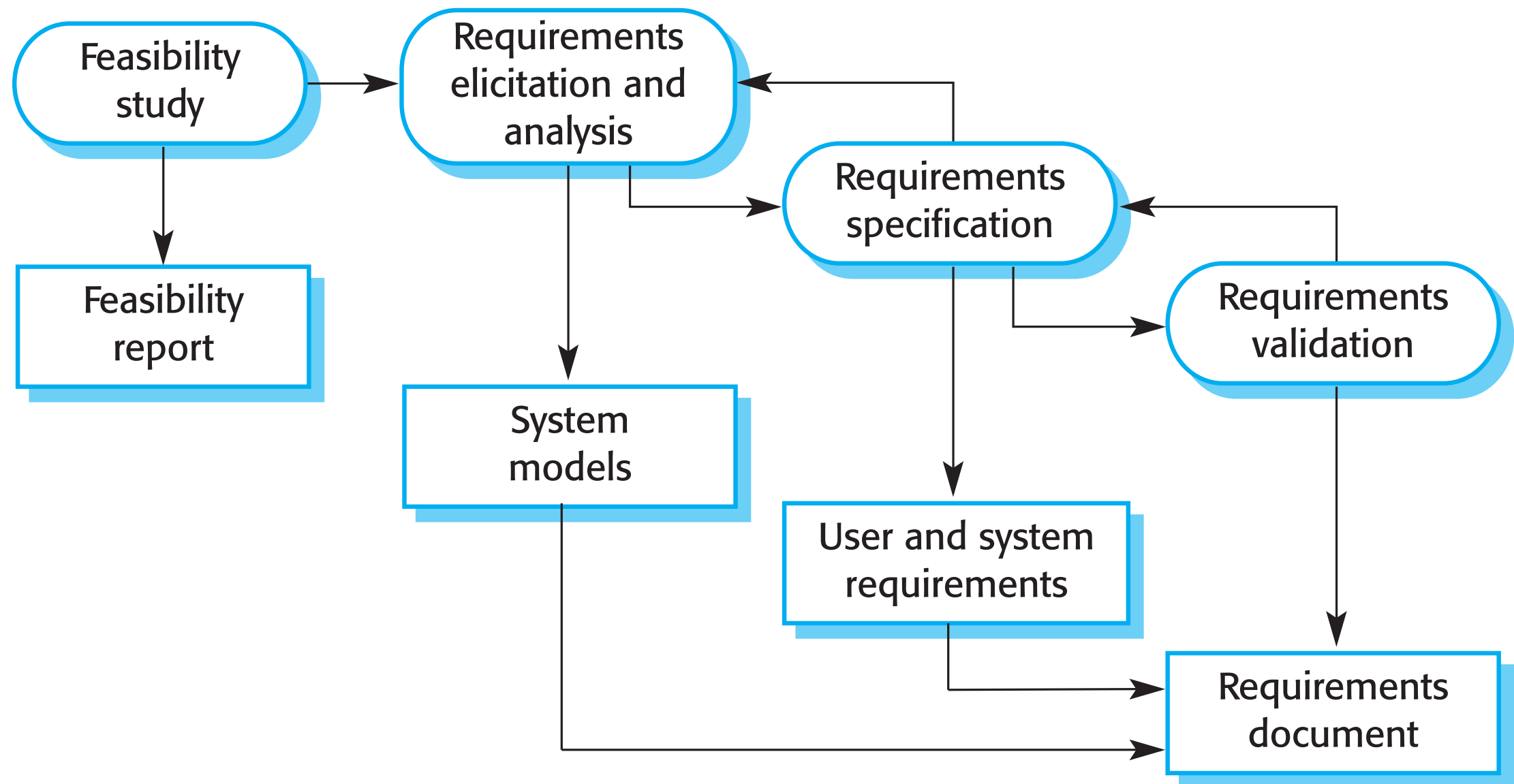
Waterfall ➡ séquentielles

Incrémental ➡ entrelacées

Analyse des besoins

- Etablir les services demandés et les contraintes de fonctionnement et de développement
- Activités de l'analyse des besoins
 - étude de faisabilité
 - découverte et analyse des besoins
 - spécification des besoins
 - validation des besoins

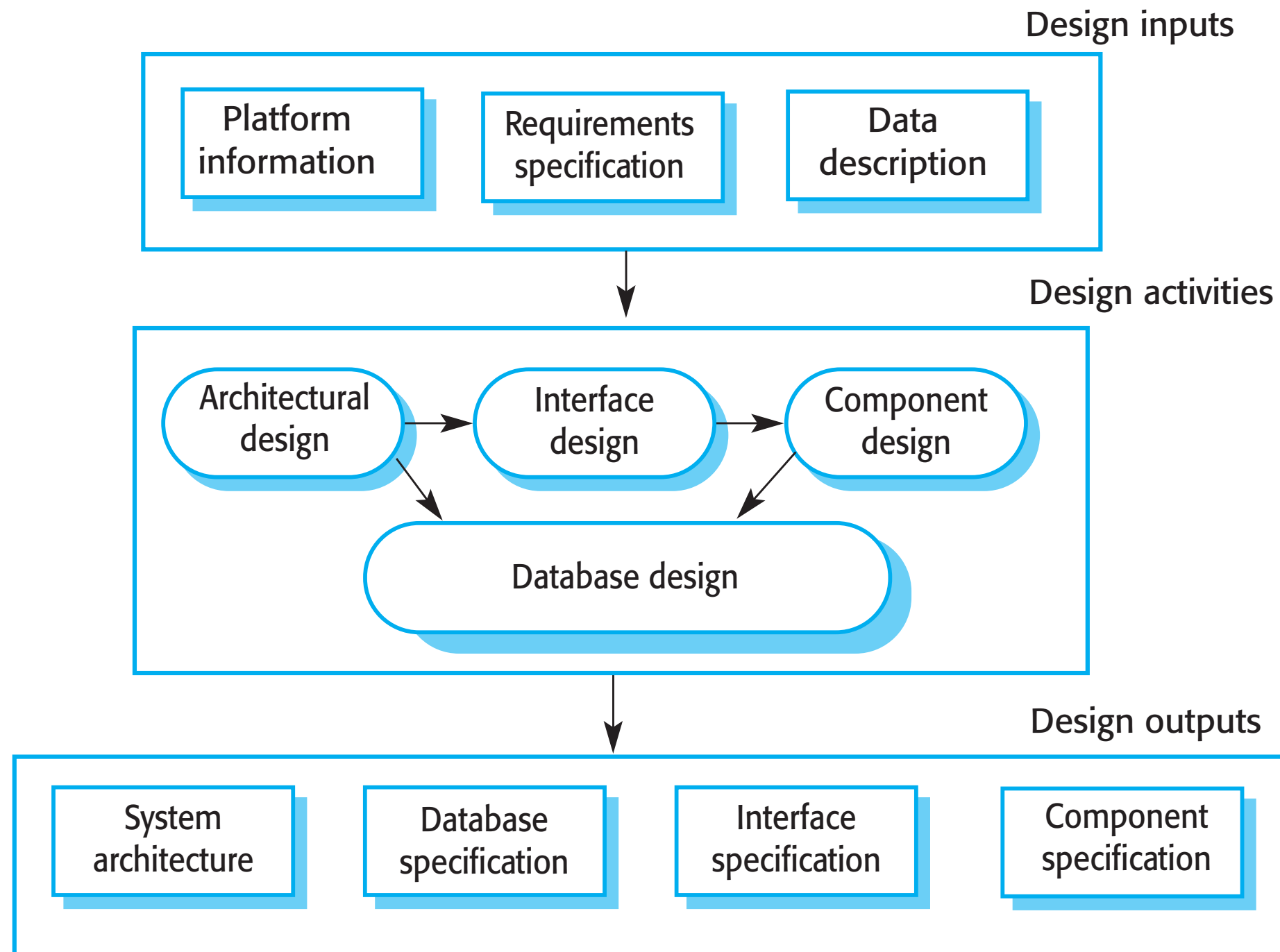
Analyse des besoins



Conception et implémentation

- Transformation de la spécification en un système exécutable
- Conception
 - ▶ définir une structure logicielle qui correspond à la spécification
- Implémentation
 - ▶ traduire la structure en un programme

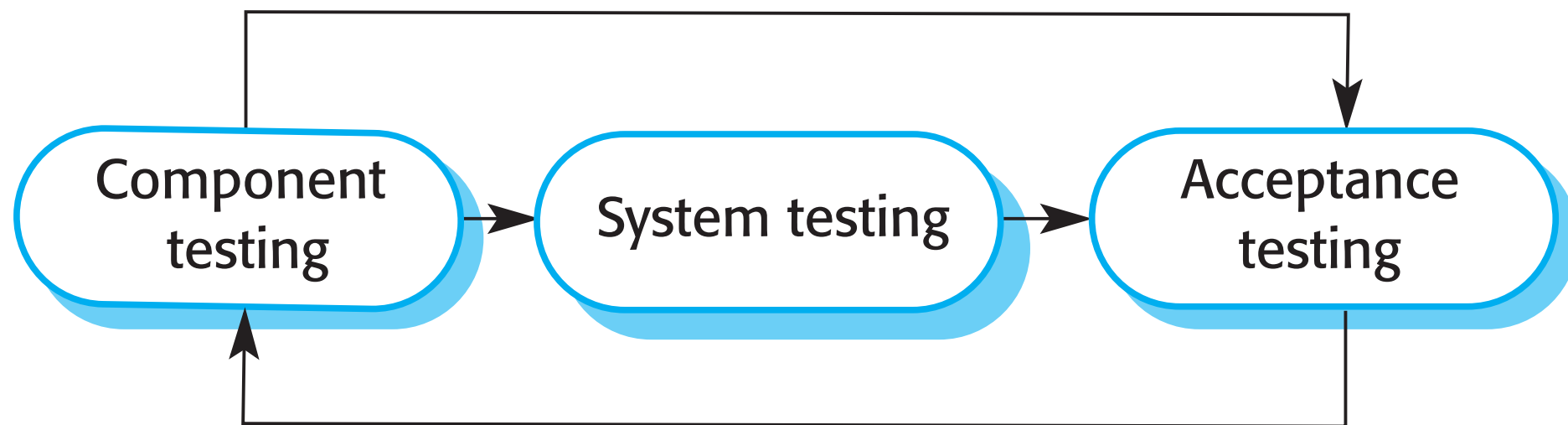
Conception et implémentation



Validation et vérification

- Montrer que le système est **conforme à la spécification** et **satisfait les besoins** du client
- **Test** - exécution avec des cas de test extraits de la spécification (des données réelles traitées)
- **Checking** - inspection et review

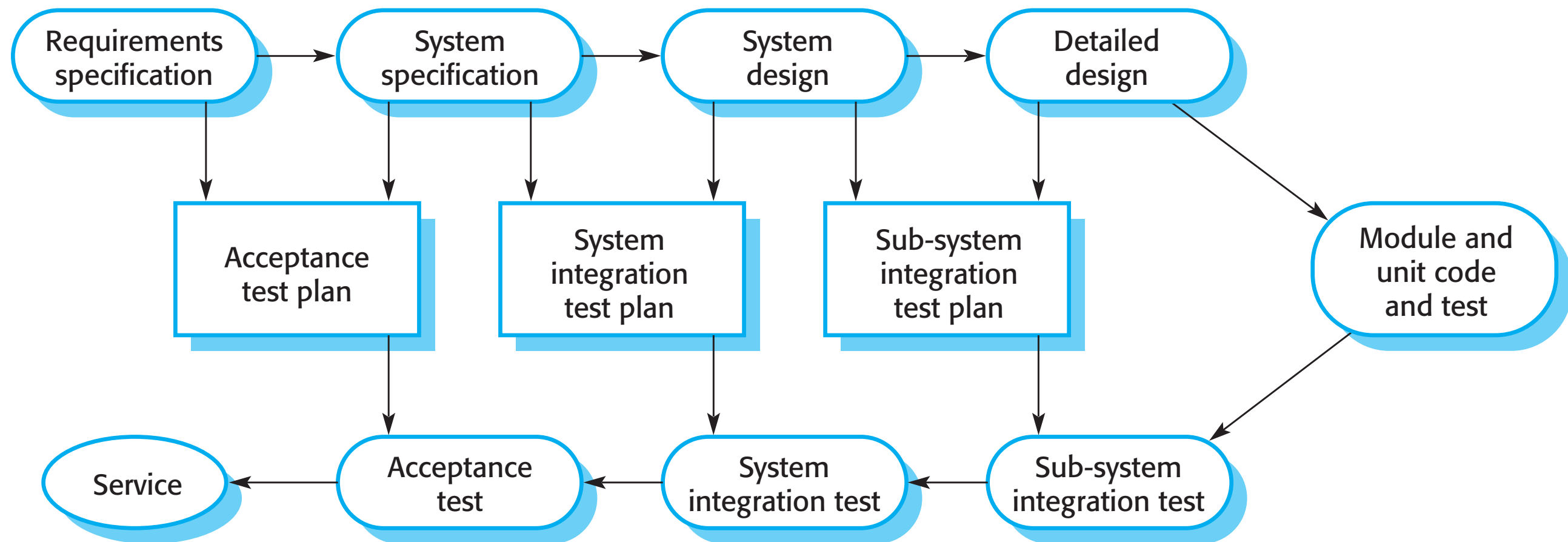
Testing



Testing

- **Component testing**
 - ▶ tests indépendants des composants
- **System testing**
 - ▶ tests du système (et des sous-systèmes)
- **Acceptance testing**
 - ▶ tests avec données client

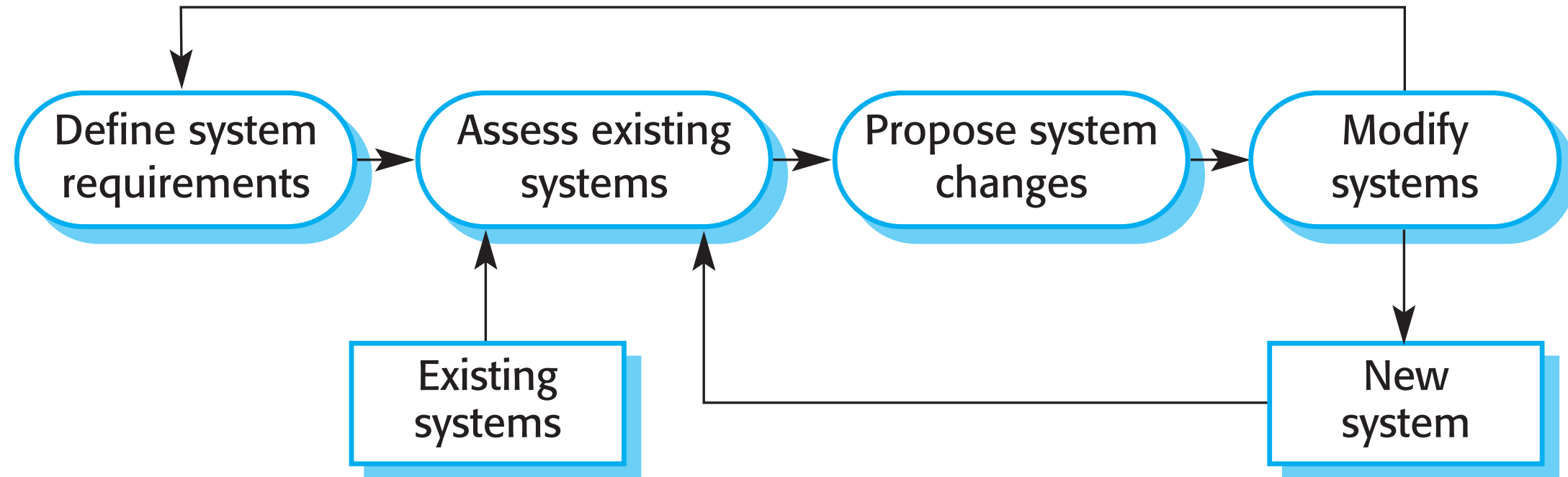
Testing dans un modèle en V



Evolution

- Le logiciel est naturellement flexible
- Les besoins du domaine peuvent changer et les logiciels correspondants doivent évoluer en conséquence
- La limite entre développement et maintenance est de plus en plus difficile à déterminer

Evolution



Questions ?

Quizz



1

Allez sur wooclap.com

2

Entrez le code d'événement dans le bandeau supérieur

Code d'événement
LQOGPC



Activer les réponses par SMS