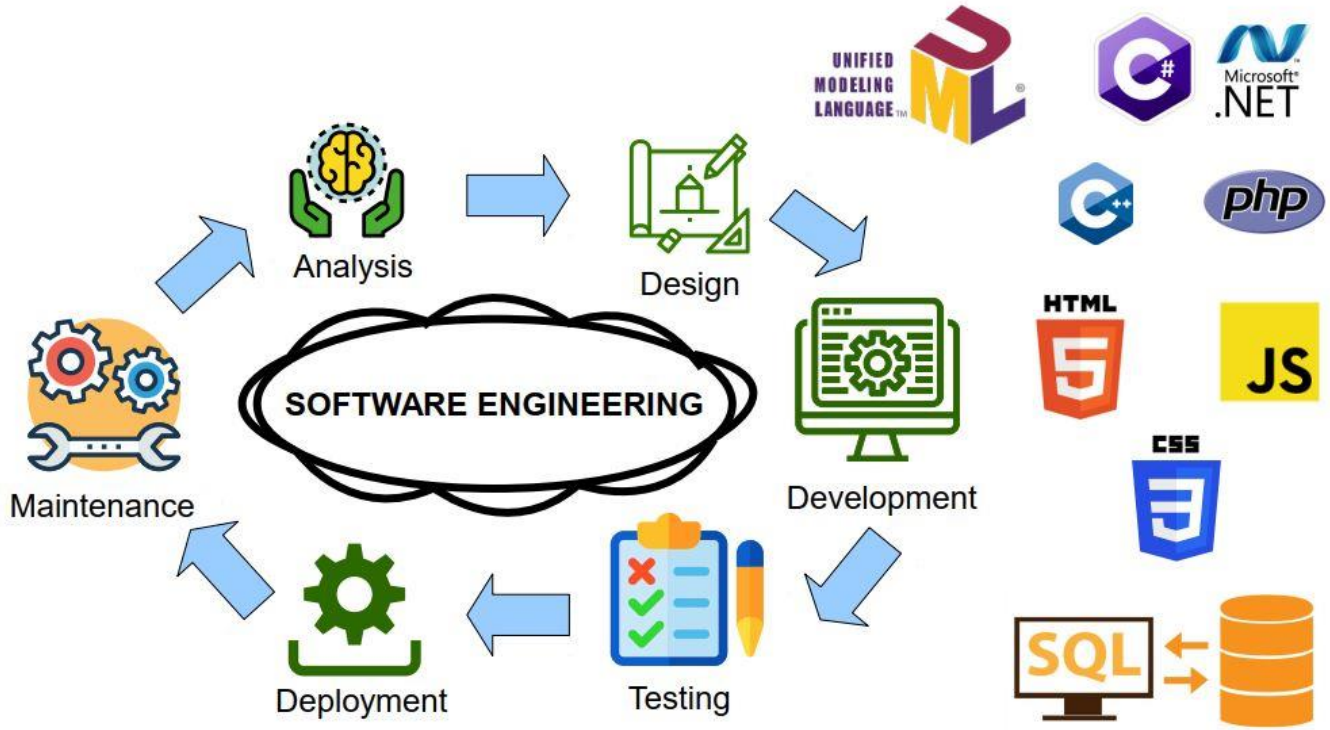
Introduction au Génie Logiciel



Dr El Hadji Bassirou TOURE

Ecole Supérieure Polytechnique (ESP)

2020 - 2021

Le processus logiciel

- Un ensemble structuré d'activités essentielles au développement de logiciels.
- Il existe plusieurs processus logiciels différents, mais tous comportent :
  - Spécification : définition de ce que le système est censé faire;
  - Conception et implémentation : définition de l'organisation du système et l'implémentation du système.
  - Validation : vérification que le système fait ce que le client veut.
  - Evolution : changement du système en réponse aux besoins du client.
- Un modèle de processus logiciel est une représentation abstraite d'un processus. Il présente une description d'un processus suivant une perspective particulière.

Description de processus logiciels

- Lorsque l'on décrit ou discute des processus, on parle souvent des activités dans ces processus et de leur ordre.
- Ces descriptions peuvent également concernées :
  - Produits : les résultats d'une activité de processus.
  - Rôles : les responsabilités des personnes travaillant dans le processus.
  - Pré et post-conditions : assertions qui sont vraies avant et après qu'une activité de processus ait été *adoptée* ou qu'un produit ait été produit.

Processus planifiés et processus agiles

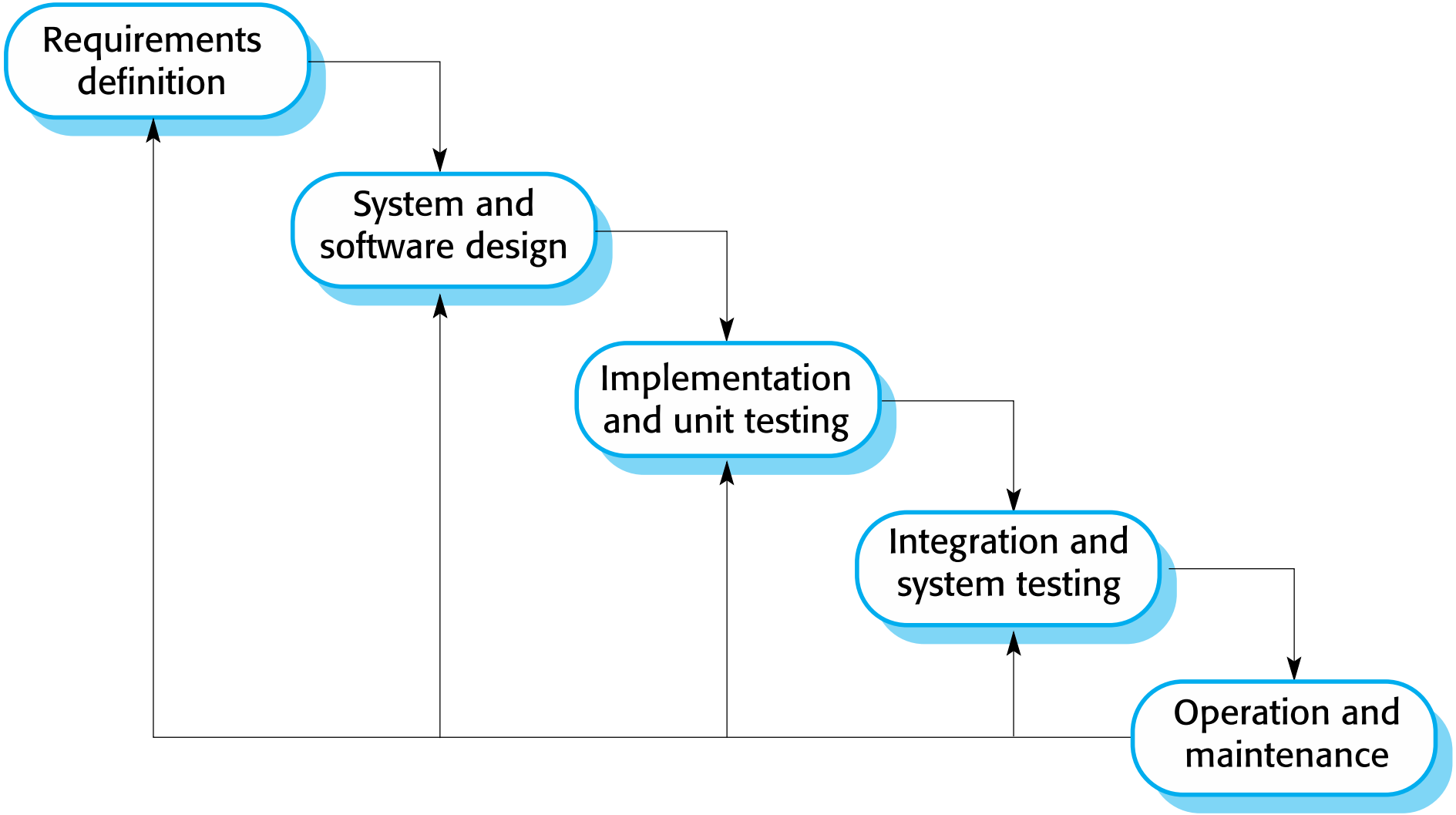
- Les processus planifiés sont des processus où toutes les activités sont planifiées à l'avance et les avancées sont mesurées par rapport à ce plan.
- Dans les processus agiles, la planification est incrémentale et il est facile de modifier le processus pour représenter les changements des exigences du client.
- Dans la pratique, la plupart des processus incluent à la fois des approches planifiées et des approches agiles.
- Il n'y a pas de bon ou mauvais processus logiciel.

Modèles de processus logiciel

Modèles de processus logiciels

- Le modèle en cascade
  - Les phases de spécification et de développement sont distinctes et séparées. C'est un modèle planifié.
- Développement incrémental
  - La spécification, le développement et la validation ne sont pas dissociés. C'est un modèle pouvant être planifié ou agile.
- Intégration et configuration
  - Le système est assemblé à partir de composants existants configurables. C'est un modèle pouvant être planifié ou agile.
- En pratique, les grands systèmes sont développés en utilisant un process incorporant des éléments de tous ces modèles.

Le modèle en cascade



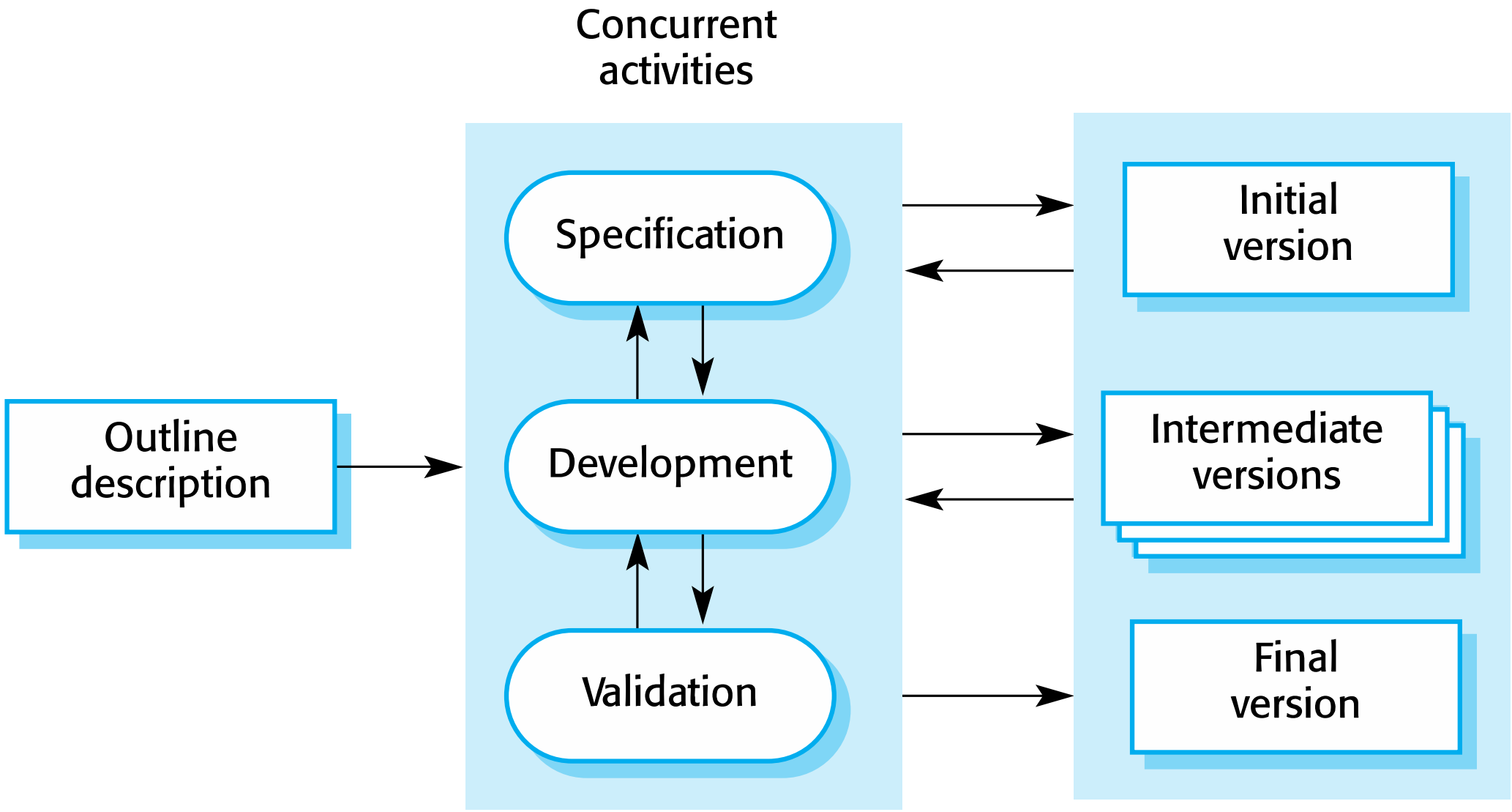
Les phases du modèle en cascade

- Il y des phases identifiées et séparées dans ce type de modèle :
  - Définition et analyse des exigences (Requirements definition)
  - Conception de systèmes logiciels (System and software design)
  - Implémentation et test unitaire (Implementation and unit testing)
  - Intégration et test du système (Integration and system testing)
  - Mise en opération et maintenance (Operation and maintenance)
- Le principal désavantage du modèle en cascade est la difficulté de prendre en compte les changements après que le processus ait déjà été enclenché. En principe, une phase doit être achevée avant que la suivante ne débute.

Problèmes du modèle en cascade

- Le partitionnement rigide du projet en différentes phases rend difficile la prise en compte des changements des exigences de l'utilisateur.
  - Donc ce modèle est uniquement préconisé lorsque les exigences sont bien comprises et les changements negligeables durant le processus de conception.
  - Peu de systèmes ont des exigences fixes et non changeantes.
- Ce type de modèle est généralement utilisé dans les grands projets logiciels où un système est développé sur plusieurs sites.
  - Pour ce type de systèmes, la nature planifiée du modèle aide à la coordination du travail.

Développement incrémental



Avantages du développement incrémental

- Le coût de la prise en compte des changements des exigences des utilisateurs est réduit.
  - La quantité d'analyse et de documentation à refaire est moins importante.
- Il est possible de recevoir les retours du client sur les tâches de développement déjà effectuées.
  - Les clients peuvent laisser une appréciation sur les démonstrations du logiciel et mesurer l'état d'avancement dans l'implémentation du système.
- Une livraison ainsi qu'un déploiement plus rapide du logiciel.
  - Les clients sont capable d'utiliser et de bénéficier du logiciel plus tôt que pour les modèles en cascade.

Problèmes du développement incrémental

- Le processus n'est pas visible.
  - Les gérants ont besoin de délivrables réguliers pour mesurer les avancées.
- La structure des systèmes tendent à se dégrader lorsque de nouveaux incréments sont ajoutés.
  - Outre le temps et l'argent dépensés pour l'amélioration du système, les changements réguliers tendent à corrompre la structure du système.

Intégration et configuration

- Se base sur la réutilisation logicielle où des systèmes sont integrés à partir de composants ou d'autres applications.
- Les éléments réutilisés peuvent être configurés pour adapter leur comportement ainsi que leurs fonctionnalités aux besoins de l'utilisateur.
- La réutilisation est maintenant devenue une approche standard de beaucoup de systèmes applicatifs.

Types de logiciels réutilisables

- Applications autonomes qui sont configurées pour être utilisées dans un environnement particulier.
- Collection d'objets développés sous forme de paquetage et à intégrer avec un cadre de composants tels que .NET ou J2EE.
- Services web développés selon les standards de service et qui sont disponibles pour être invoqués à distance.

Avantages et inconvénients

- Avantages
  - Réduction des coûts et des risques.
  - Livraison et déploiement plus rapides.
- Inconvénients
  - Inévitable compromission des exigences.
  - Manque de contrôle sur l'évolution des éléments de systèmes réutilisés.

Activités de processus

Activités de processus

- Dans la réalité, un processus logiciel est composé de séquences non dissociées d'activités techniques, collaboratives et de gestion, avec comme principal objectif de spécifier, concevoir, implémenter et tester un logiciel.
- Les quatre activités de processus basiques que sont la spécification, le développement, la validation et l'évolution, sont organisées différemment suivant les processus de développement employés.
- Par exemple, pour un modèle en cascade, elles sont organisées en séquences alors que pour le développement incrémental, elles sont indissociées.

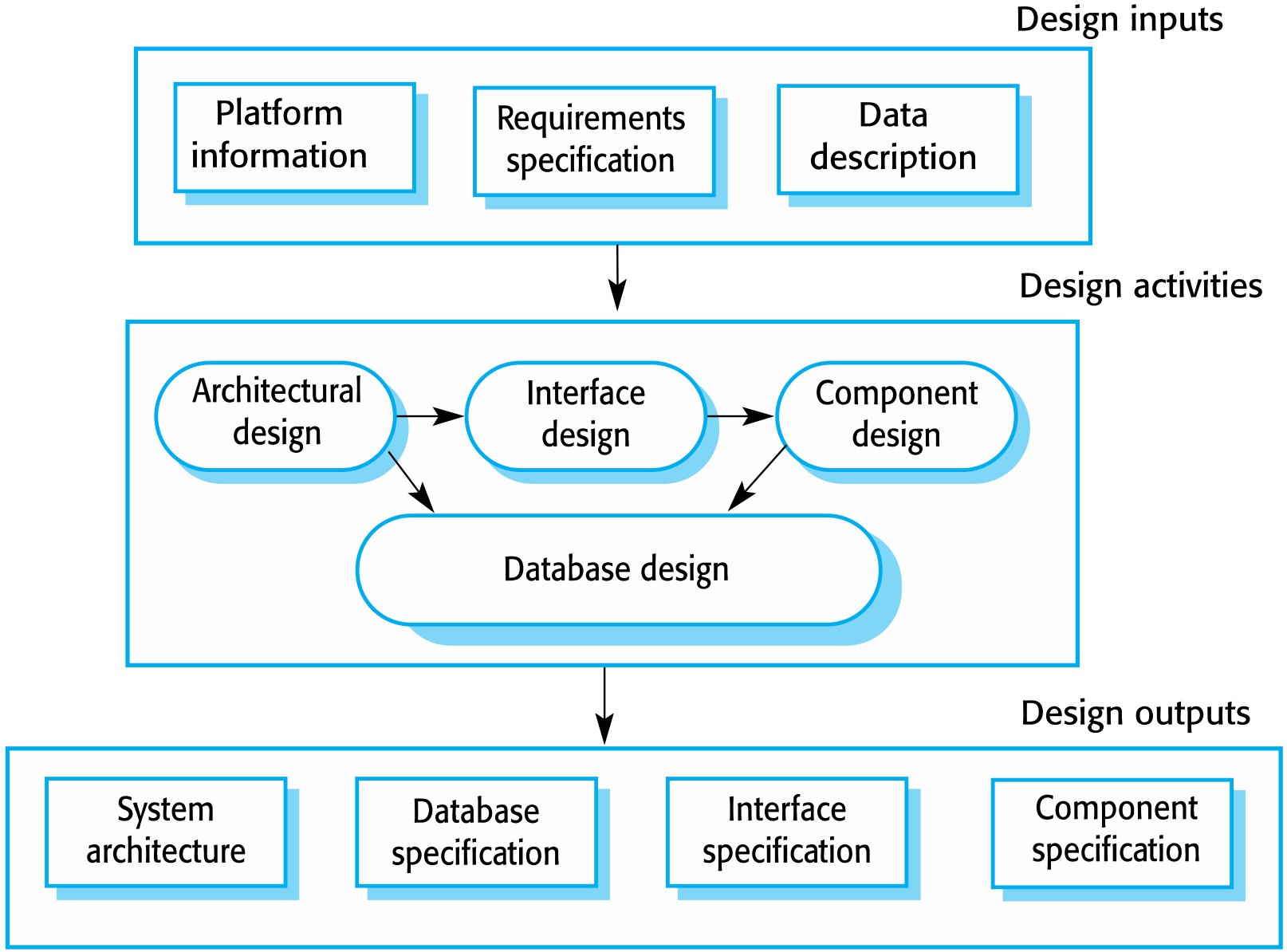
Spécification logicielle

- C'est le procédé par lequel on établit les services requis et les contraintes des opérations du système et de son développement.
  - Processus de l'ingénierie des exigences
    - Recueil et analyse des exigences
      - Ce dont les acteurs du système ont besoin ou attendent du système
    - Spécification des exigences
      - Définition des exigences en détail
    - Validation des exigences
      - Vérification de la conformité des exigences par rapport aux attentes du client.

Conception et implémentation logicielles

- Le procédé par lequel les spécifications du système sont converties en un système exécutable.
- Conception logicielle
  - Conception d'un modèle du système qui réalise les spécifications.
- Implémentation
  - Traduction de ce modèle en un programme exécutable.
- Les activités de conception et d'implémentation sont assez proches et ne sont pas souvent dissociées.

Modèle générique de conception (design)



Activités de conception (Design activities)

- Conception architecturale, on identifie la structure générale du système, ses principaux composants (sous-systèmes, modules), leurs relations et comment ils sont distribués.
- Conception de bases de données, on définit la structure des données et comment ces données seront représentées.
- Conception d'interfaces, on définit les interfaces entre les composants.
- Conception de composants et sélection, on recherche les composants réutilisables. S'ils ne sont pas disponibles, on conçoit de nouveaux composants.

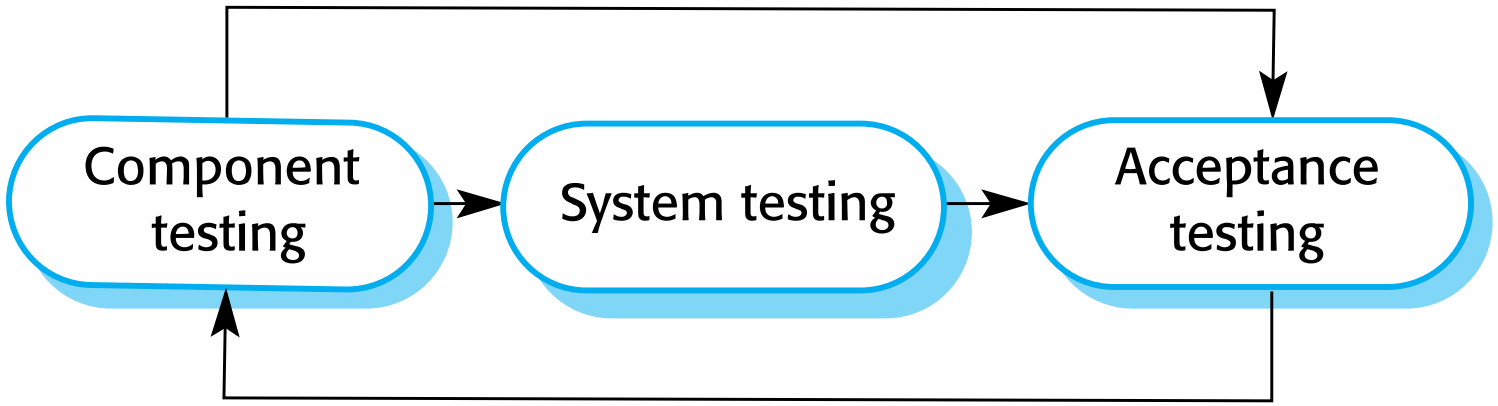
Implémentation du système

- Le logiciel est implémenté soit en développant un nouveau programme ou en configurant une application existante.
- Le débogage est l'activité où l'on recherche les défauts des programmes, et où on corrige ces défauts.

Validation logicielle

- La vérification et la validation (V&V) sont destinées à prouver que le système est conforme à ses spécifications et répond aux exigences du client.
- Impliquent la vérification et la révision des processus ainsi que le test du système.
- Le test du système signifie l'exécution du système avec des cas de tests dérivés des spécifications des données réelles à traiter par le système.
- Le test est l'activité la plus usuelle de V&V.

Phases du test



Phases du test

- Test de composant
  - Des composants individuels sont testés indépendemment
  - Les composants peuvent être des fonctions ou des objets ou un ensemble cohérent de ces entités.
- Test du système
  - Tester le système comme un ensemble.
- Test du client
  - Tester avec les données du client pour s'assurer que le système répond aux besoins du client.

Evolution logicielle

- De façon inhérente, un logiciel est flexible et peut changer.
- Etant donné que les exigences changent, pour s'adapter aux changements du business métier, le logiciel supportant ce business doit aussi changer et évoluer.
- La démarcation entre les phases de développement et d'évolution est de moins en moins flagrante parce que de moins en moins de systèmes sont complètement nouveaux.

Prise en compte des changements

Prise en compte des changements

- Les changements sont inévitables dans les grands projets logiciels.
  - Les changements métier entraînent de nouvelles exigences
  - Les nouvelles technologies offrent de nouvelles possiblités visant l'amélioration des implémentations
  - Le changement des plateformes implique la modification des applications.
- Un changement implique des reprises (Ex : nouvelle analyse des exigences).
- Ce qui fait que les coûts relatifs aux changements incluent à la fois ceux liés aux reprises ainsi que les coûts d'implémentation de nouvelles fonctionnalités.

Réduction des coûts liés aux reprises

- Anticipation des changements : Le processus logiciel inclue des activités pouvant anticiper de possibles modifications avant que des reprises importantes ne soient nécessaires.
  - Par exemple, un prototype du système peut être développé pour montrer quelques caractéristiques clés du système aux clients.
- Tolérance aux changements : Le processus est conçu de sorte que les changements puissent être intégrés à un moindre coût.
  - Ceci implique une certaine forme de développement incrémental. Les changements proposés peuvent être implémentés sous forme d'incréments pas encore développés. De ce fait, juste une petite partie du système va être altérée pour incorporer les changements.

Prise en compte des changements d'exigence

- Prototyage
  - Une version du système ou partie de celui-ci est développée rapidement de sorte à verifier les exigences du client et la faisabilité des choix conceptuels.
  - Cette approche supporte l'anticipation des changements.
- Livraison incrémentale
  - Des incréments sont livrés au client pour appréciation et expérimentation.
  - Cette approche supporte à la fois l'anticipation et la tolérance aux changements.

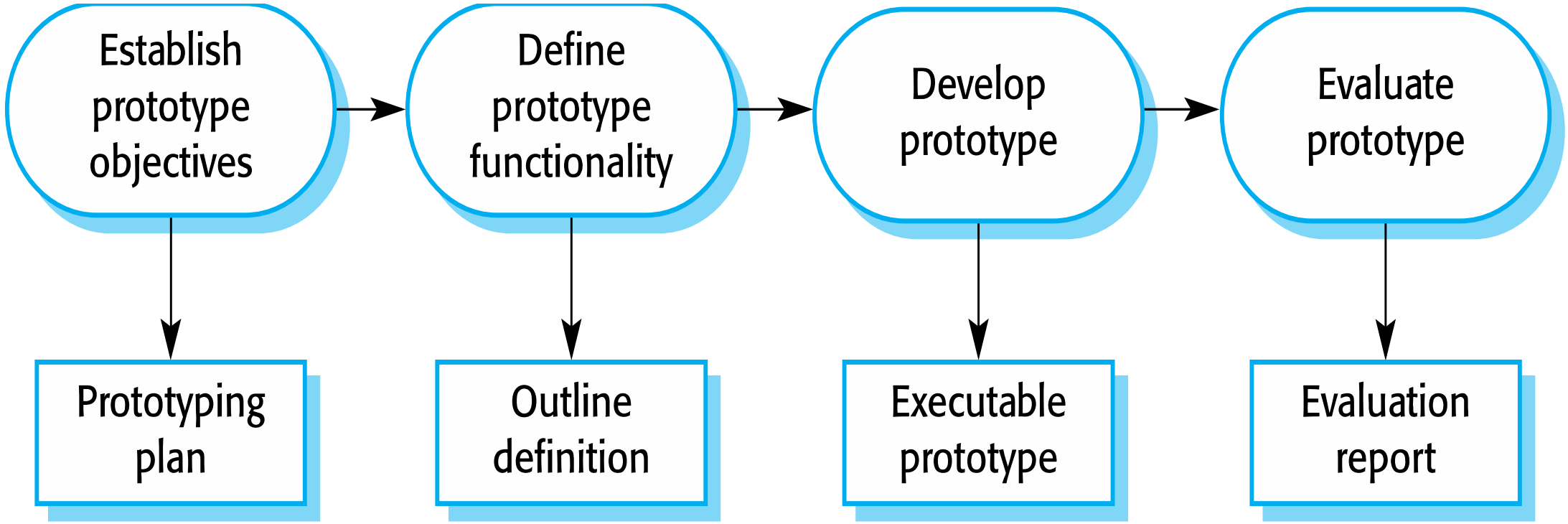
Prototypage

- Un prototype est une version initiale d'un système utilisé pour démontrer des concepts et tester les choix conceptuels.
- Un prototype peut être utilisé dans :
  - Les processus d'ingénierie des exigences pour aider au recueil, analyse, spécification et validation des exigences.
  - Les processus conceptuels pour explorer les choix et développer une interface utilisateur.
  - Les processus de test pour exécuter des tests.

Bénéfices du prototypage

- Améliore l'utilisabilité du système.
- Plus proche des besoins de l'utilisateur.
- Améliore la qualité de la conception.
- Améliore la maintenabilité.
- Réduit des efforts de développement.

Processus de développement de prototype



Développement de prototype

- Utilise les langages et outils permettant le prototypage rapide
- Peut impliquer un abandon de fonctionnalités
  - Se concentrer sur les aspects du produit les moins bien maitrisés.
  - N'inclut pas la vérification et la découverte des erreurs.
  - Se concentre sur les exigences fonctionnelles
  - au détriment de celles non fonctionnelles (la fiablité, la sécurité, etc).

L'abandon de prototype

- Les prototypes doivent être laissés de côté après le développement car ils ne peuvent pas être un bon socle pour la production de systèmes :
  - Il peut s'avérer impossible d'ajuster le système pour satisfaire les exigences non fonctionnelles.
  - Généralement les prototypes ne sont pas documentés.
  - La structure du prototype a tendance à se dégrader à travers les changements rapides et fréquents.
  - Le prototype ne satisfait généralement pas les standards de qualité.

Livraison incrémentale

- A lieu de livrer le système en une seule livraison, le développement et la livraison sont décomposés en incréments de sorte que chaque incrément soit une livraison d'une partie des fonctionnalités attendues.
- Les exigences de l'utilisateur sont priorisées et celles avec la plus haute priorité sont incluses dans les premiers incréments.
- Une fois que le développement d'un incrément a démarré, ses exigences sont bloquées. Toutefois, les exigences des incréments suivants peuvent quant à elles continuer à évoluer.

Développement et livraison d'incréments

- Développement incrémental
  - Développer les incréments du système et évaluer chacun d'eux avant de procéder au développement du prochain incrément.
  - Il s'agit de l'approche normale utilisée dans les méthodes agiles.
  - L'évaluation est effectuée par le client/l'utilisateur.
- Livraison incrémentale
  - Déployer un incrément à utiliser par les utilisateurs finaux.
  - Evaluation plus réaliste sur l'utilisation pratique du logiciel.
  - Difficile à implementer pour les systèmes de remplacement car les incréments ont moins de fonctionnalité que les systèmes à remplacer.

Avantages de la livraison d'incréments

- Les premiers incréments servent de prototype et aide au recueil des exigences pour les incrément suivants.
- Moins de risques d'echec pour le projet.
- Les parties du système avec une plus grande priorité font objet de la plupart des tests.

Inconvénients de la livraison d'incréments

- La plupart des systèmes nécessitent un ensemble d'outils qui sont utilisés par différentes parties du système.
  - Etant donné qu'une exigence n'est entièrement définie qu'après un incrément ait été implémenté, il peut être difficile d'identifier les outils communs dont auront besoin tous les incréments.
- L'essence des processus itératifs est que les spécifications sont définies en conjonction avec le logiciel.
  - Cela peut être un problème si la spécification complète du système est par exemple une nécessité.