

IFT 2255 - Genie Logiciel

Processus de développement

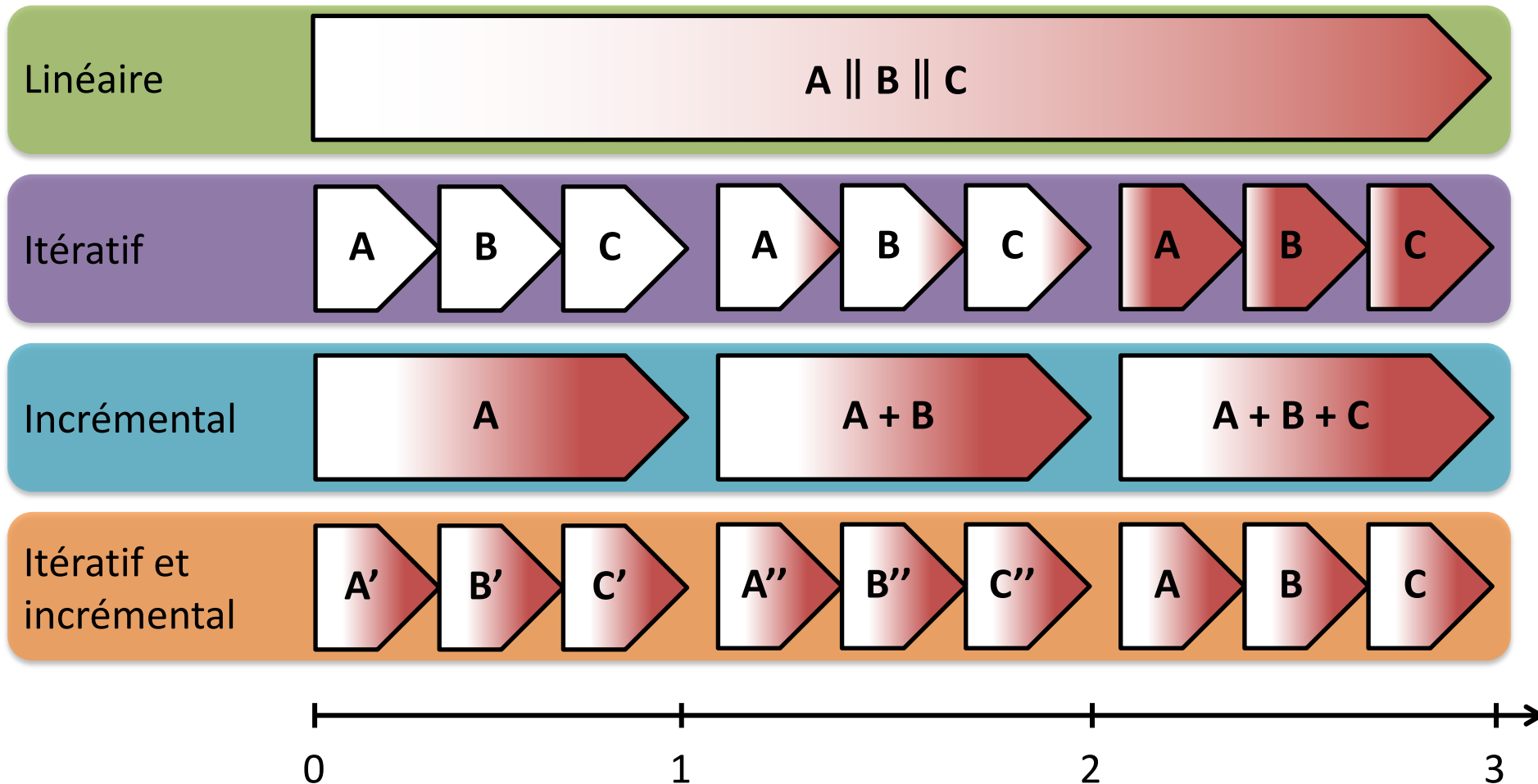
Processus

La manière selon laquelle une organisation produit un logiciel

- Méthodologie
- Modèle de développement du logiciel
- Techniques et outils à utiliser
- Rôles et individus nécessaires

Modèles de développement

- Projet de 3 mois avec 3 composants (A, B, C) à livrer



Résumé

- **Linéaire** (Cascade, V)
 - Petits projets, cadre bien défini, peu de risques, pas de changements de besoins en phase de tests, tout est déployé d'un coup
- **Itératif** (Spirale)
 - Exigences pas encore clairs, risque élevé de changements dans les exigences, prototype initial nécessaire, chaque itération améliore et converge vers le produit final suite au retour de la précédente
- **Incrémental** (Phases)
 - Livraison des composants par ordre de priorité pour réduire risques et augmenter la qualité tout en incorporant les changements tout au long du projet, risques de conception et techniques éliminés tôt, pour de gros projet car les tests commence tôt
- **Itératif et incrémental** (Unifié, agile)
 - Chaque composant est développé dès le 1^{er} incrément, elles sont améliorées dans les incréments suivants, peut rapidement intégrer les changements et changer de priorités

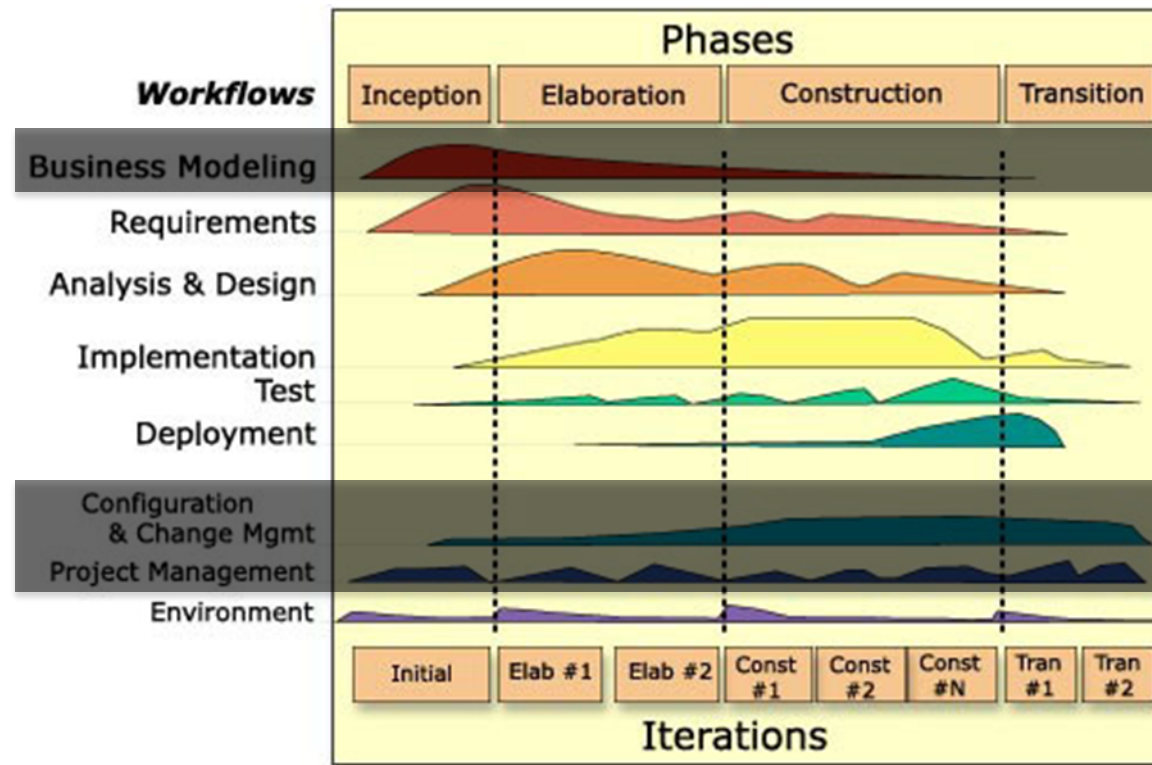
QUESTION

Quelles propriétés du processus nous aideraient à choisir le bon modèle de développement logiciel pour un projet ?

- Si le processus est dirigé par le **retour du client**
- Si la livraison doit se faire d'**un coup** ou en **phases**
- Si beaucoup de **changements** sont anticipés
- Si on est prêt à investir dans la **qualité**

Processus unifié

Modèle adopté dans ce cours



Processus unifié

- Phases
 - **Création:** cadre, cas d'utilisation
 - **Élaboration:** conception de l'architecture initiale, estimés de coût & ressource
 - **Construction:** développe les composants, livraisons, critère d'acceptation
 - **Transition:** déploiement
- Flux de travail
 - **Exigences:** analyse du domaine de l'appli & création des artefacts des exigences
 - **Analyse & conception:** création de la solution & des artefacts de conception
 - **Implémentation:** création du code
 - **Test:** évaluation du processus et des artefacts
 - **Déploiement:** transition du système à l'utilisateur
 - **Environnement:** maintenance (communication & gestion des configurations)

Flux de travail (*Workflows*)

Activités en continu



Workflow des exigences

- But: déterminer ce dont le client a **besoin**
 1. Comprendre le **domaine** d'application
 - Environnement spécifique au métier où le logiciel sera exploité
 2. Construire un **modèle d'affaire**
 - Utiliser UML pour décrire le processus d'affaire du client
 - À n'importe quel moment, si le client pense que le coût n'en vaut pas la peine, on met immédiatement fin au développement
 3. Définir les **exigences** du système
 - Découvrir quelles sont les **contraintes**



Exigences

Analyse

Conception

Implémentation

Test

Déploiement

Workflow d'analyse

- But: analyser et **affiner** les exigences
- *Pourquoi ne pas le faire dans le workflow des exigences ?*
 - Exigences doivent être entièrement compréhensible par le client
 - Exprimés en langage naturelle, donc imprécises/ambigües
 - Artéfacts d'analyse doivent être assez précis et complets pour l'équipe de développement
- Document (artefacts) de **spécifications**
 - Constitue un contrat, décrit ce que le logiciel doit faire
 - Précis et non ambigü
- Spécifications doivent être complètes et correctes
 - Base pour les tests et la maintenance
 - Une fois approuvé, planification détaillée, cahier des charges, estimation du temps et du coût



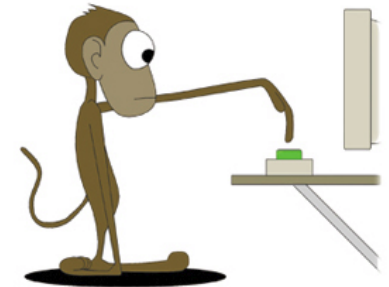
Workflow de conception

- But: **affiner** le workflow d'analyse jusqu'à ce que la matériel soit dans une forme **implémentable** par les programmeurs
- Produit la documentation du **design**
 - Comment le logiciel doit effectuer les spécifications
- **Conception architecturale**
 - Modules, composants, communication, fiabilité
- **Conception détaillée**
 - Algorithmes, structures de données, langage de programmation, réutilisation



Workflow d'implémentation

- But: **traduire** le design dans le langage de programmation choisi
- À ce moment, toutes les décisions sont déjà prises
- Grands logiciels complexes partitionnés en sous-systèmes
 - Composants et artéfacts de code
 - Stratégie de « diviser pour régner » (*divide-and-conquer*)



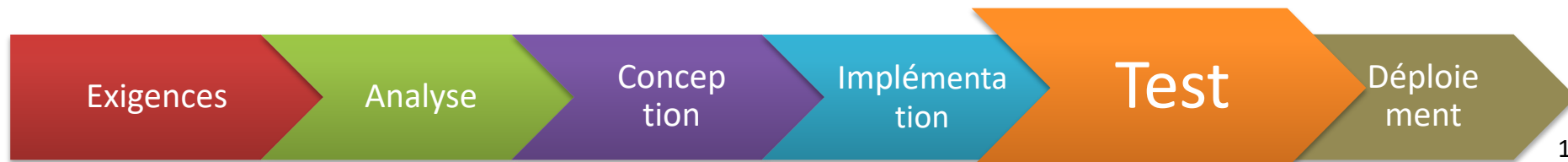
Workflow d'implémentation

Intégration

- L'implémentation de chaque artéfact à coder est assignée à un programmeur (ou à une équipe)
 - Si l'artéfact A dépend de l'artéfact B, alors le programmeur de A et celui de B doivent communiquer et être au courant de ces dépendances
- L'intégration de chaque artéfact est essentielle
 - Valide la conception
- Plusieurs livrables peuvent être nécessaire
 - Alpha release
 - Beta release
 - Release candidates (Microsoft)

Workflow de test

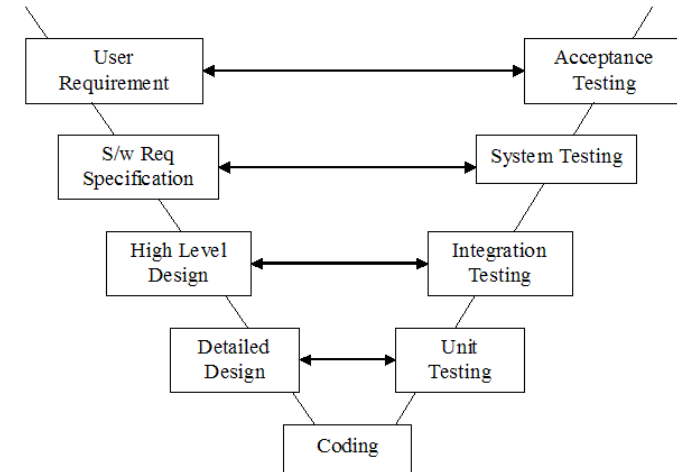
- But: **Évaluer** les artefacts produits par les autres workflows
- Tester est la responsabilité de:
 - **Chaque développeur** et mainteneur
 - L'équipe d'**assurance qualité**
- Tous les artefacts doivent être **traçables**
 - Pouvoir retracer chaque module, classe, méthode à un artefact de conception, qu'on peut retracer à un artefact d'analyse, qu'on peut retracer à un artefact des exigences
 - Essentiel pour la phase de test



Workflow de test

Test à chaque phase

- Exigences
 - Tous les artéfacts du logiciel produits doivent **provenir** des exigences
 - Client révisé les exigences
- Analyse
 - Spécifications **révisées conjointement** avec un expert du client et l'équipe d'analyse
- Conception
 - **Révisé** par les développeurs et l'équipe d'assurance qualité
- Implémentation
 - **Test unitaire**: chaque composant implémentée doit être testée dès que complétée
 - **Test d'intégration**: après chaque itération, combiner les composants et tester
 - **Test système**: tester le système au complet
 - **Test d'acceptation**: par le client à chaque livraison



Workflow de déploiement

- **Plusieurs** déploiements durant le développement
 - À la fin de chaque phase
 - Déploiement en continu
- Préparation des **environnements**
 - Environnement de développement
 - Environnement de test
 - Environnement de production
- **Installation** du logiciel
- **Livraison** de tous les artefacts de tous les workflows au client

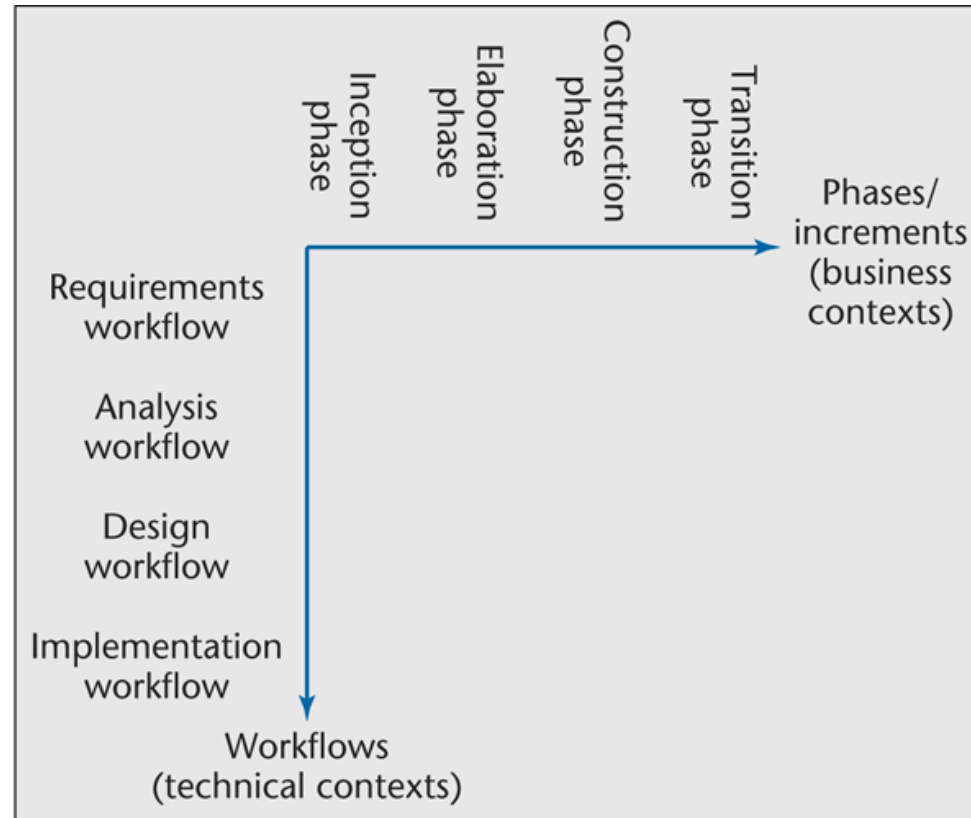


Maintenance post-déploiement

- Après le déploiement de la première version du logiciel effectuée
- Préoccupations de maintenance doivent avoir été pensées assez tôt (**pendant la conception et l'implémentation**)
- La phase la plus **longue** et la plus **dispendieuse**
- Problèmes causés surtout par manque de documentation
 - Auto-documentation (encore) hasardeuse
- Après chaque modification du logiciel, tous (ou presque) les tests doivent être repris
 - **Test de régression**

Phases

- Workflow: contexte technique d'une étape
- Phase: contexte d'affaire d'une étape



Phase de création

- But: déterminer les nécessités et la faisabilité du logiciel à développer
1. Comprendre le domaine
 2. Construire un modèle d'affaire
 3. Délimiter le cadre du projet
 4. Débuter le dossier d'analyse commercial (*business case*)

Phase de création

Business case

- Est ce que le logiciel est rentable ?
- Combien de temps avant d'obtenir son rendement sur investissement (ROI) ?
- Si le produit va être vendu sur le marché, y a-t-il eu une étude de marché ?
- Est-il possible de livrer le logiciel proposé à temps ?
- Quel sera l'impact s'il est livré plus tard ?
- L'équipe de développement a-t-elle l'expérience nécessaire ?
- A-t-on besoin de nouveau matériel pour le logiciel ?
- D'autres outils logiciels seront-ils nécessaires ?
- Ont-ils la/les fonctionnalité(s) nécessaire(s) ?

Phase de création

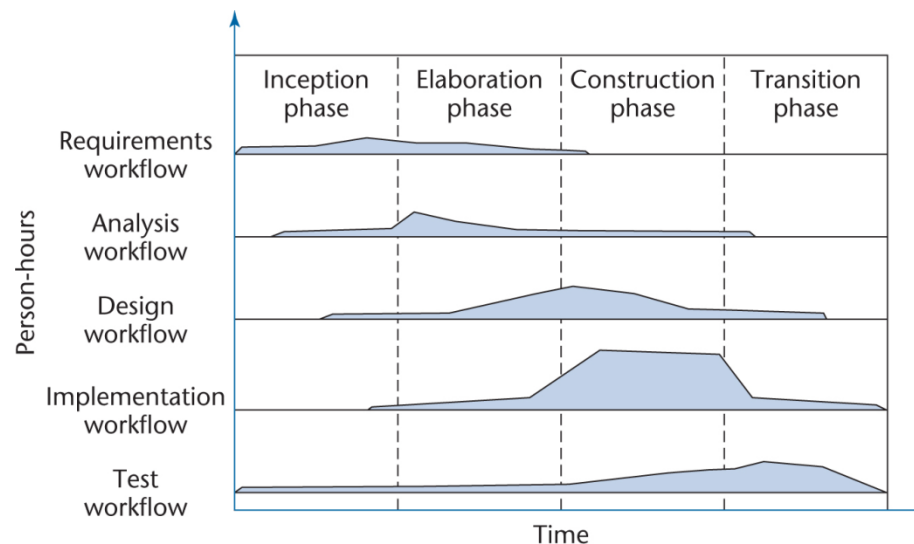
Gestion du risque

- Risques **techniques**
 - Compétence, acquisition de matériel et autres logiciels
- Risque de ne pas avoir les **bonnes exigences**
 - Effectuer le workflow des exigences correctement et incorporer les retours du client
- Risque de ne pas aboutir à la **bonne architecture**
 - L'architecture n'est pas assez solide pour des additions futures
- **Trier** les risques par probabilité de réalisation
- Toutes les questions doivent être **répondues** à la fin de la phase de création

Phase de création

Tâches

- Planification et cueillette des exigences
- Début de conception architecturale
- Pas de programmation
 - Prototypage de preuves de concept très utiles pour évaluer la faisabilité
- Test commencent sur les exigences



Phase de création

Livrables

- Version initiale du modèle du domaine, modèle d'affaire et business case
- Version initiale des artefacts d'exigence
- Début des artefacts d'analyse et d'architecture
- Liste initiale des risques
- Estimés vagues
- Plan pour la phase d'élaboration

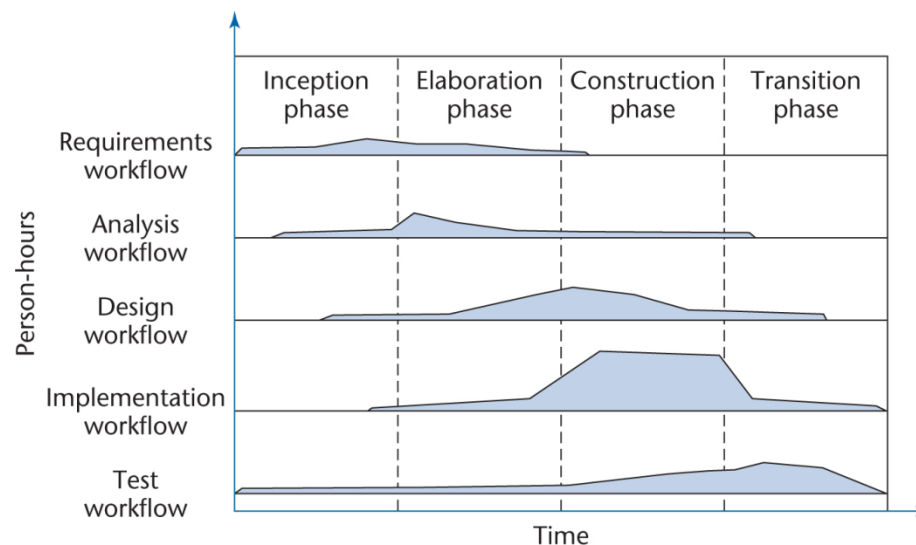
Phase d'élaboration

- But: affiner et élaborer ce qui a été fait durant la phase de création
- Affiner l'architecture
- Surveiller les risques et affiner leurs priorités
- Affiner le business case
- Produire un plan de gestion de projet

Phase d'élaboration

Tâches

- Compléter le workflow des exigences
- Se concentrer presque entièrement sur le **workflow d'analyse**
- Commencer la conception architecturale
- Mettre en place l'environnement de développement et de test



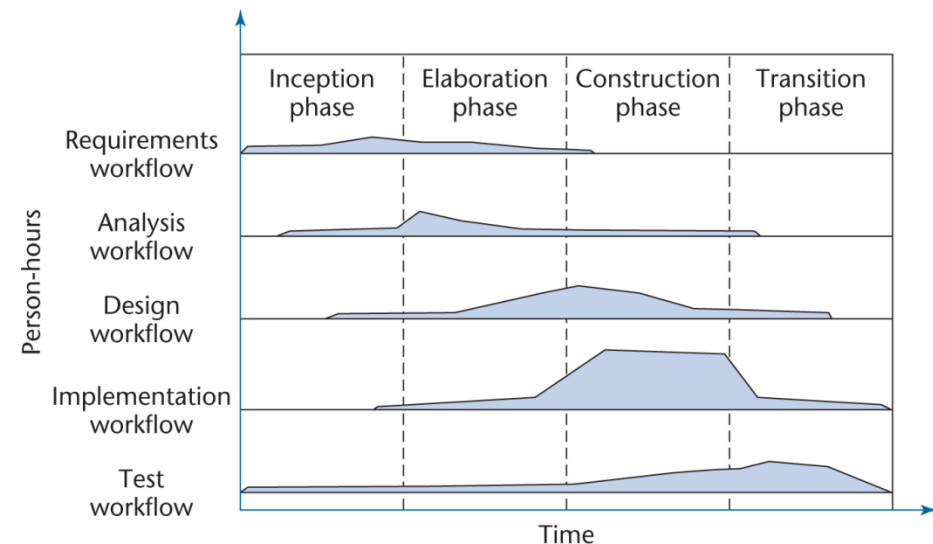
Phase d'élaboration

Livrables

- Version complète du modèle du domaine, modèle d'affaire et business case
- Version complète des artéfacts d'exigences et d'analyse (cahier des charges, use cases)
- Mise à jour des artéfacts d'architecture
- Résolution des risques les plus élevés
- Plan de gestion de projet et cahier des charges

Phase de construction

- But: produire une première version du logiciel, fonctionnelle et de qualité
- Tâches: emphase surtout sur
 - L'implémentation itérative
 - Les tests
- Livrables
 - Manuel utilisateur
 - Tous les artéfacts en version beta release
 - Architecture complétée
 - Mise à jour des autres documents
 - Plan de gestion de projet, business case



Phase de transition

- S'assurer que toutes les exigences du clients sont satisfaites
 - Correction des défauts du logiciel
 - Découverte de risques non-identifiés antérieurement
- Implantation du système dans l'environnement de production
 - Déploiement et installation
- Livrables: les versions finales de tous les artéfacts

