420-5A5
Développement d'applications de bases de données

Méthodologies

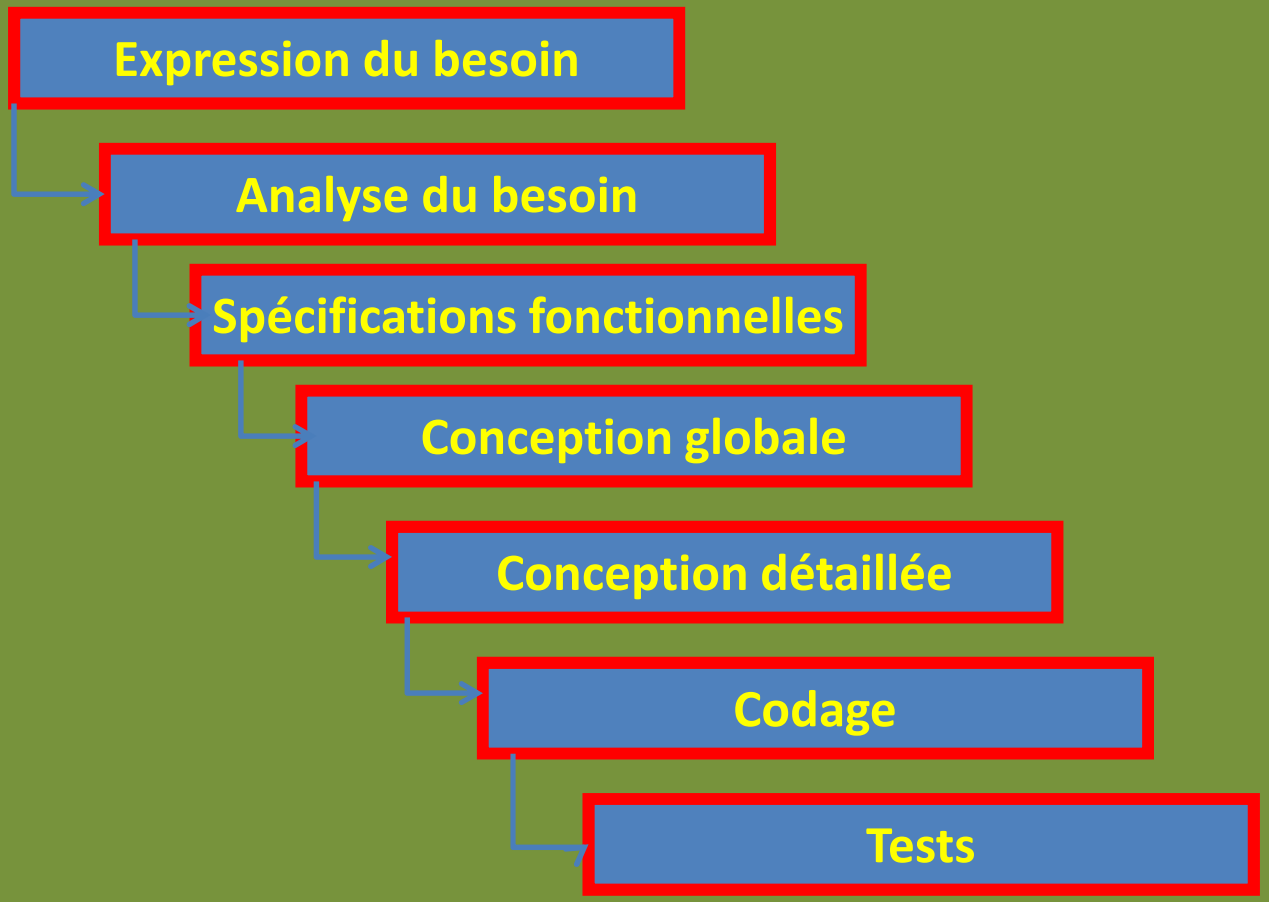
Rôles dans le développement de projets

- Chef de projet
- Analystes spécialisés (graphiste, hardware, sécurité, bases de données, réseau, etc)
- Concepteurs
  - Architecte
  - Expert dans le domaine
  - Ergonomiste
  - Programmeur d'interface utilisateur
  - Programmeur senior et junior
- Testeurs
- Documentalistes
- Gestionnaire de produit (marketing)
- Support technique

Modèle en cascade

- On ne peut construire la toiture sans fondation
- Modifier le cycle en amont == impacts majeurs

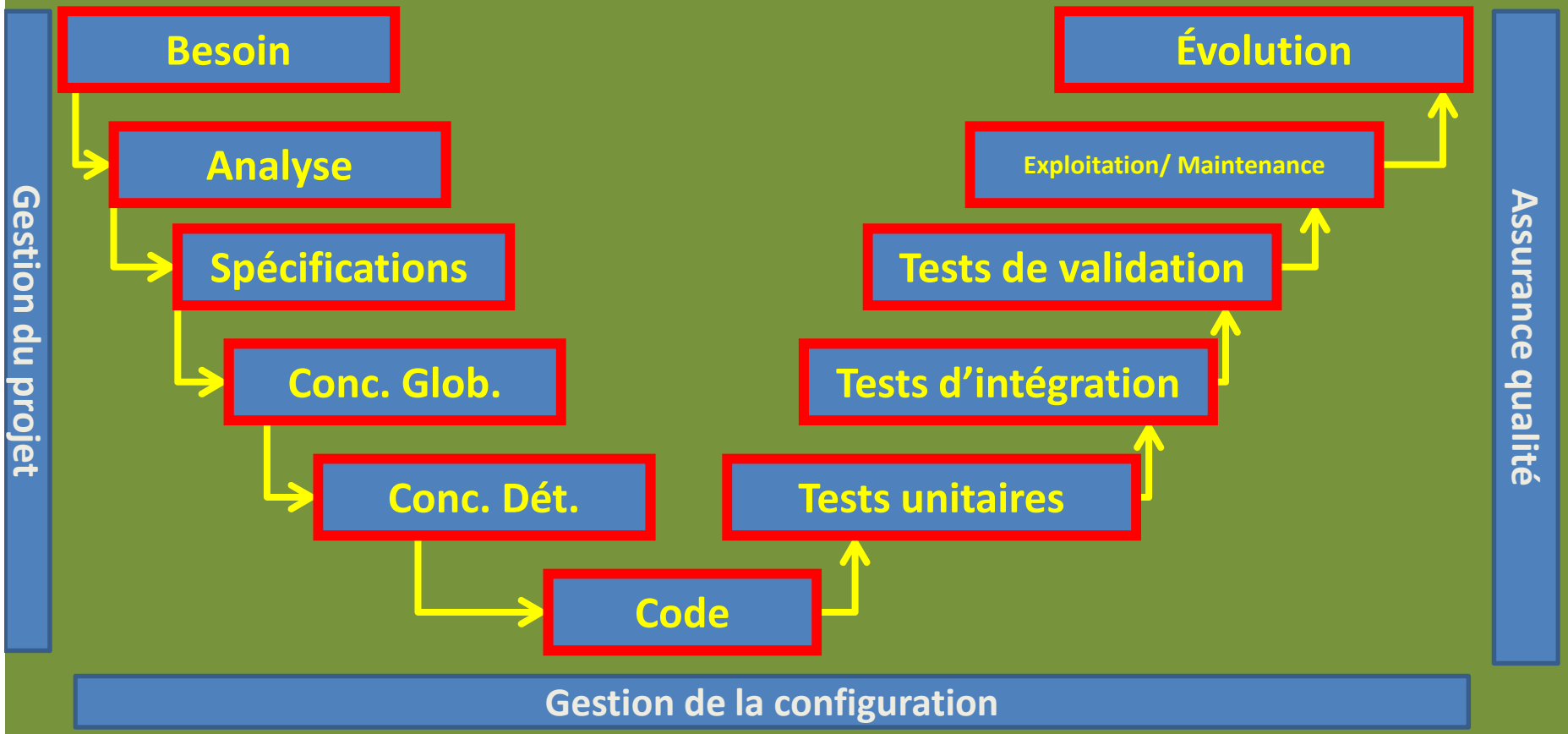
Modèle classique, en cascade, du développement logiciel



Modèle en V

- Ajouter de la réactivité à la cascade
- En préparant dans les étapes descendantes les attendus des futures étapes montantes

Modèle classique en V (intégrant l'AQL)



Critique du modèle classique

- Le modèle classique est critiqué et remis en question.
- Trop lourd
  - Prend trop de temps
  - inefficace
- Ne prend pas en considération
  - les changements très fréquents
  - Le feedback du client

2001-Manifeste Agile

2001- Manifeste Agile

- 17 représentants des méthodes légères définissent de grands concepts, simples, mais qui proposent une nouvelle façon de penser un projet de développement informatique.

Méthodes itératives

- On reprend les étapes de la cascade mais de multiples fois

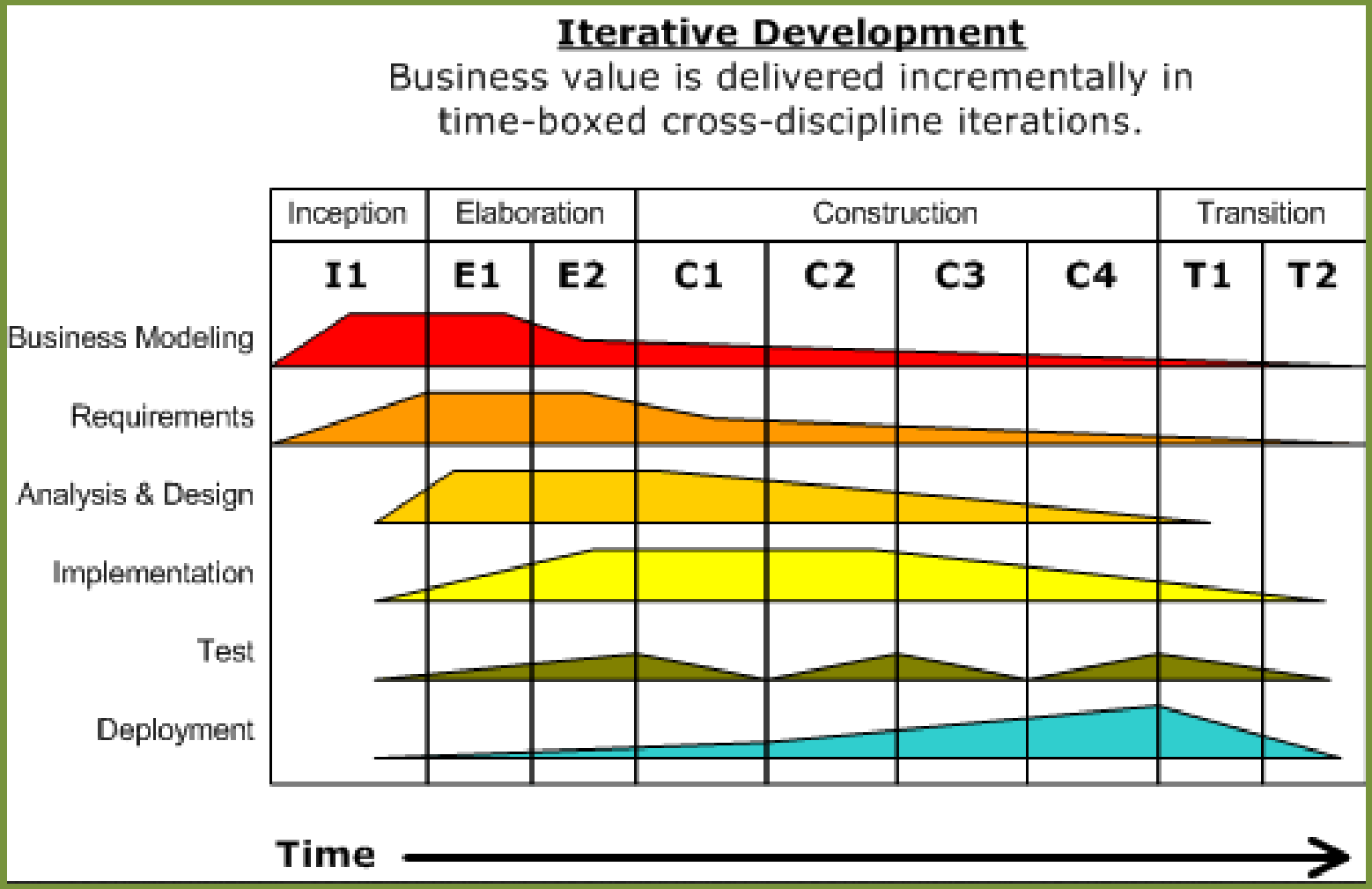
Principes Agile

- Satisfaction du client en livrant continuellement des logiciels utiles
- Release fréquents
- Le progrès se mesure aux logiciels “qui marchent”
- Les exigences tardives sont bienvenues
- Coopération intime et quotidienne entre les gens d’affaires et les développeurs
- La discussion en face à face est la meilleure communication
- Les projets se construisent autour d’individus motivés qui doivent être considérés dignes de confiance
- Attention continuelle à l’excellence technique et au bon design
- Simplicité
- Équipes autonomes qui s’auto-organisent
- Adaptation régulière aux circonstances changeantes

Principes Agiles résumés en quatre oppositions

- Individus et interactions contre processus et outils
- Logiciel qui fonctionne contre documentation exhaustive
- Collaboration du client contre négociation de contrat
- Réponse au changement contre suivi d'un plan prédéfini

Disciplines et Phases



Xtrem Programming (4 valeurs)

- **La communication**
- **La simplicité**
- **Le *feed-back***
- **Le courage**

X-Trem Programming

1- Planification et exigences

- Évaluation de la valeur commerciale de chacune des options du logiciel
- Chaque options majeures est réécrite sous la forme d'une « histoire »
- Programmeurs fournissent des estimés de temps requis
- Le client choisi les options selon les estimés et la valeur commerciale

X-Trem Programming

- 2- Petits releases incrémentaux
- 3- Métaphore système
- 4- Conception simple
  - le changement viendra; ne pas investir trop de temps à concevoir
- 5- Tester continuellement
  - Tests unitaires avant d'écrire le code
- 6- Refactorisation
  - Nettoyer le code

X-Trem Programming -3

- 7- Programmer à quatre mains/ deux cerveaux / un clavier
- 8- Propriété collective du code
- 9- Intégration continuelle
- 10- Semaine de 40 heures
- 11- Client sur le site
- 12- Standard de programmation

Spécifications des exigences

- « User story » écrites par le client
- Description courte et simple d'une fonction/option/possibilité
- En tant que <*type d'utilisateur*> je veux <*un objectif désiré*> ainsi je pourrai < *justification*>
- Beaucoup moins détaillée qu'un document de spécifications
- Sert à estimer le coût de l'histoire à développer

User stories: exemples

- En tant que nouveau membre je veux m'inscrire pour pouvoir accéder aux possibilités du logiciel
- En tant que membre inscrit, je veux me logguer pour voir mes statistiques
- En tant qu'administrateur je veux éliminer un membre ne respectant pas les règles

Backlog

- Liste des fonctionnalités, tâches, user stories nécessaires et suffisantes
- Pas de forme imposée
- Atomicité des éléments

Tableau des tâches

Exemples

- https://agiletools.wordpress.com/2007/11/24/task-boards-telling-a-compelling-agile-story/

Élection du contenu d’une itération

- Les développeurs détermines la valeurs de chaque item du backlog
- Le client décide du contenu du cycle

Méthode d’estimation

- Poker planning

Vélocité

- A la fin d'une itération, on additionne les estimations associées aux tâches/user stories développés
- Donne des estimations pour les prochaines itérations
- Précision augmente avec le temps

Réunions fréquentes

- Qu’as-tu terminé depuis la dernière réunion?
- Qu’auras-tu terminé d’ici la prochaine réunions?
- Quels obstacles te ralentissent/bloquent en ce moment?

Proverbes Agiles

- Trying to catch up is the fastest way to get further behind.
- Your test suite is more valuable than your code.
- Don't theorize, try it.
- Everything you did today can be done over tomorrow in half an hour and be better.
- If it isn't fun you're doing something wrong.
- http://www.agile-process.org/proverbs.html

X-trem Programming conclusion:
7 puisque