

Systems Architecting

Abdellatif MEGNOUNIF

Chap. 2

Le Paradigme de l'Architecting

COURS 2 Jeudi 20.10.2011

© **Abdellatif MEGNOUNIF** FT-Tlemcen



1. Introduction

Architecting, the planning and building of structures, is as old as human societies – and as modern as the exploration of the solar system

Paradigme: Modèle théorique de pensée qui oriente la recherche et la réflexion scientifiques

1. Introduction (suite)

Architecting, concept très ancien (Egyptiens, Grecs, les romains...)
(architecting classique).

Différence: Complexité et la capabilité technologique.

Actuellement il ya des systèmes **nouveaux** (non connus que très récemment) exemple : systèmes de très grande qualité, temps réel (real-time) , en boucle fermée, interactif, reconfigurable, software-intensif...

Nouveaux domaines (PC, réseaux intersatellites, services de santé, systèmes de control et de commandes...)

Conduit à des concepts et des outils **d'architecting nouveaux**

Nouveau formalisme, nouvelle profession « **Systems Architecting** »

Combinaison des principes et concepts de
systèmes et architecting (architecture)

1. Introduction (suite)

- ❑ Développer plus après la guerre froide (mi 1980) .
- ❑ Actuellement, le pb de la globalisation (transport, communication, compétition...) il faut sortir du lot.
- ❑ Technologiquement et politiquement : soit unique.
- ❑ Temps de restructuration, d'invention, d'architecting des procès et produits nouveaux.
- ❑ Nouvelles pensées comment les systèmes sont créés et construits.
- ❑ Depuis longtemps, l'architecture était la même partout.
- ❑ Les voitures, les avions, les navettes spatiales ont les mêmes formes (qu'elle était alors le besoin de l'architecture? Rien)
- ❑ Compétition globale actuelle pousse au changement.
- ❑ Systèmes avec des architectures différentes sont attirés par les marchés.

1. Introduction (suite)

D'autres questions peuvent se poser:

- ❑ Comment des architectures de logiciel peuvent suivre l'évolution rapide des technologies?**
- ❑ Jusqu'où (en détails) peut aller un architecte de systèmes, au sous/ sous systèmes?**
- ❑ Quelle est la distinction entre « systems architecting » et « systems engineering »?**
- ❑ ...**

2. Différence entre architecting et engineering

- ❑ Pas encore tranché entre ingénieur civil et architecte civil (approche classique). Mais:
- ❑ **Engineering**: traite des éléments **mesurables**, en utilisant des outils analytiques dérivées des mathématiques et des sciences dures. C'est un **processus déductif**
- ❑ **Architecting**: traite des éléments **non mesurables** utilisant des outils non quantitatifs et des guides basés sur les leçons pratiques apprises. C'est un **processus inductif**
- ❑ Plus en détails:
 - Engineering est concerné par les coûts **quantifiables**. L'architecting par la valeur **qualitative**
 - Engineering cherche l'**optimisation** technique. L'architecting la **satisfaction** du client.
 - Engineering c'est plus une **science**. L'architecting plus un **art**.

2. Différence entre architecting et engineering (suite)

Caractéristiques	Architecting	A & E Rôle intermédiaire proposé	Engineering
Situation/Objectifs	Mal-Structuré	Contrainte	Compris
	Satisfaction	Conformité	Optimisation
Méthodes	Heuristiques	↔	Equations
	Synthèse	↔	Analyse
	Art & Science	Art & Science	Science & Art
Interfaces	« misfit » inadapté	Critique	Complétude
Intégrité du système maintenue	« esprit seul »	Objectifs clairs	Processus et méthodologie disciplinés
Management	Travail pour le client	Travail avec le client	Travail pour le constructeur
	Conceptualisation et certification	Cascade complète	Rencontre les exigences du projet
	Confidentialité	Conflit d'intérêt	Profit Vs Cout

❑ **Pb mal structuré: Pb où l'énoncé du Pb dépend de l'énoncé de la solution**

2. Différence entre architecting et engineering (suite)

- ❑ Architecte travaille avec des situations **mal structurées** i.e situations où les objectifs et les moyens ne sont bien connus de façon certaine. (Exigences du système ne sont pas bien présentées)
- ❑ En contraste, l'approche classique engineering cherche directement une solution optimale pour une description claire des objectifs.
- ❑ Mal structuré : Objectif ne peut pas être optimisation
- ❑ Bonne architecture et bon engineering sont tous les deux produits de l'art et de la science et la combinaison de l'heuristique et de l'analyse.
- ❑ Mais, pendant l'architecting, se pencher plutôt vers l'art et l'heuristique.
- ❑ En architecting, il faut juste identifier les interfaces qui ne fonctionnent pas et les éliminer.

3. Définitions

- ❑ **Architecture: (Larousse)** Art de concevoir et construire un bâtiment selon des partis techniques et des règles techniques déterminés.
- ❑ **Architecte (classique):** Professionnel qui conçoit le parti, la réalisation ainsi qu'éventuellement la décoration d'un édifice, d'un bâtiment, et qui en contrôle l'exécution.
- ❑ **Architecture du système:**
La structure du système fondamental et unificateur définie en termes d'éléments du système, les interfaces, les processus, les contraintes et les comportements (INCOSE GTAS).

3. Définitions (suite)

❑ Architecture du système: (suite)

- L'organisation fondamentale d'un système, incarnée dans ses composantes, leurs relations les uns aux autres et l'environnement, et les principes régissant sa conception et son évolution (ANSI / IEEE 1471)
- – La structure des composants, de leurs relations, et les principes et lignes directrices régissant leur conception et leur évolution dans le temps (DoD C4ISR cadre, Perry-Garlan, IEEE 610.12 PAS)
- – La structure organisationnelle d'un système ou un composant (IEEE 610.12)
- – Un ensemble d'informations qui définit une valeur de systèmes, le coût et risque suffisamment pour les besoins du commanditaire des systèmes (règle Maier du pouce)

3. Définitions (suite)

❑ **Architecting du système:** (se réfère au processus)

1) Architecting des systèmes est une partie de l'ingénierie des systèmes plus concernés par la détermination de l'objectif, le concept la formulation et la structuration et la certification pour l'utilisation

- **Objectif:** structurer et certification pour l'utilisation.

Pourquoi quelqu'un voudrait le système?

- **Concept et structure:** L'ensemble des caractéristiques structurelles, dans tout points de vue

- **Certification:** Détermination de l'aptitude pour une utilisation

2) Les activités de définir, maintenir, améliorer et certification de la bonne mise en œuvre de l'architecture du système

Architecting est un mot **inventé** qui décrit comment les architectures sont créées comme « **engineering** » qui décrit comment les moteurs (engines) et autres artefacts sont créés.

3. Définitions (suite) **Architecte du système**

- ☐ **La personne, l'équipe ou l'organisation responsable de l'architecture des systèmes.**
- ☐ **l'architecte du système est le concepteur de systèmes de haut niveau à mettre en œuvre. L'architecte de systèmes établit la structure de base du système, définissant les caractéristiques essentielles de conception de base et les éléments qui constituent le cadre de tout ce qui suit, et sont plus difficiles à changer plus tard.**
- ☐ **L'architecte de systèmes d'ingénierie fournit le point de vue de la vision des utilisateurs pour ce que le système doit être et doit faire, et les chemins le long desquels il doit être capable d'évoluer, et s'efforce de maintenir l'intégrité de cette vision car elle évolue lors de la conception détaillée et la mise en œuvre.**

- ☐ Interface avec l'utilisateur (s) et commanditaire (s) et toutes les autres parties prenantes afin de déterminer leur besoins (évolutifs).
- ☐ Génération du plus haut niveau des exigences du système, basé sur les besoins des utilisateurs et d'autres contraintes telles que le coût et le calendrier.
- ☐ Veiller à ce que cet ensemble d'exigences de haut niveau est cohérent, complet, correcte, et défini opérationnellement.
- ☐ Faire des analyses coût-bénéfice afin de déterminer si les exigences sont mieux satisfaites par des manuels, des logiciels ou des fonctions matérielles; avec une utilisation maximale des produits existants sur le marché ou des composants déjà développés.
- ☐ Développer des algorithmes de partitionnement (et d'autres processus) pour allouer toutes les exigences actuelles et prévisibles en partitions distinctes telles qu'un minimum de communication est nécessaire entre les partitions, et entre l'utilisateur et le système.



- ☐ Décomposer le système en (couches successives de) sous-systèmes et composants dont chacun peut être manipulé par un seul ingénieur ou une équipe d'ingénieurs ou d'architecte subordonné.
- ☐ Interface avec les ingénieurs de conception et de mise en œuvre, ou des architectes subordonnés, de sorte que tous les problèmes posés lors de la conception ou la mise en œuvre peuvent être résolus en conformité avec les concepts fondamentaux d'architecture, et les besoins des utilisateurs et les contraintes.
- ☐ Veiller à ce qu'un maximum d'architecture robuste est développée.
- ☐ Génération d'un ensemble d'exigences de test d'acceptation, en collaboration avec les concepteurs, les ingénieurs de test, et l'utilisateur, qui déterminent que toutes les exigences de haut niveau ont été respectées, en particulier pour l'interface homme-machine.

- ❑ **Générer des produits tels que des croquis, des modèles, un guide des premiers utilisateurs, et des prototypes pour laisser l'utilisateur et les ingénieurs constamment à jour et en accord sur le système à être fourni pendant son évolution.**
- ❑ **S'assurer que tous les produits architecturaux et des produits avec l'apport d'architecture sont maintenus en l'état le plus actuel et n'a jamais permis de devenir obsolète.**

3. Critères pour un architecting des systèmes efficace et mature

- **Besoin des managers et des clients des critères pour juger l'efficacité et la maturité de l'architecting des systèmes :**
 - **La reconnaissance des clients et autres du besoin pour architecturer des systèmes complexe.**
 - **L'existence de méthodes architecturales, standards, et organisations (i.e c'est une discipline)**
 - **Différence entre les valeurs de jugement et les décisions techniques entre client, architecte et constructeur.**
 - **Reconnaitre que l'architecture est un art et une science (utilisation de techniques analytiques et non analytiques)**

4. Méthodes de l'architecting classique

Quatre théories traditionnelles

1. Normative (basée sur la solution. « Solution-based »):

- Les règles dures sont fournies (de quelque part), le succès est défini en suivant les règles.
- Méthode analytique, déductive, basée sur l'expérience, facile... (science de l'ingénieur: meilleure méthode normative)
«Si cela ne ressemble pas à ce que nous faisons maintenant, il doit être faux».
- Ex: Codes de construction et les standards de communication.

2. Rationnelle: (Basée sur la méthode « Method-based)

- Solutions obtenues à partir des objectifs. Solveurs de systèmes généraux, optimisation et technique formelles
- Ex: analyse des systèmes et engineering

4. Méthodes de l'architecting classique (suite)

3. Participative (basée sur les intervenants « Stakeholder-based »)

- Solution obtenue par un processus de groupe. Conception par consensus du groupe.
- Reconnaît la complexité créée par des intervenants multiples. L'objectif c'est le consensus.
- Ex: Engineering concurrente et brainstorming.

4. Heuristique (Leçons apprises).

- Basée sur le bon sens qui provient de l'expérience.
- Ex: Simplifier, simplifier, simplifier.

5. Le paradigme de l'architecture classique

- ❑ Le plus ancien de l'architecting classique (processus de création d'architecture) sont les pyramides (Egypte 4000 ans).
- ❑ C'est un système complexe.
- ❑ Ce ne sont pas de simples places pour enterrement. Mais traduisent une puissance politique et religieuse et des accomplissements en engineering impressionnantes.
- ❑ Les inter relations entre éléments étaient complexes de façon à ce qu'il n'y avait pas de moyens.
- ❑ Avec le manque de moyens, il y avait une architecture classique simple pour de simples travaux.
- ❑ Aujourd'hui la technologie est très avancée en construction de bateaux (par ex) création du génie maritime et de l'architecture navale.
- ❑ Avancés rapides en aérodynamique, chimie, matériaux, énergie électrique, communications, traitement de l'information et software conduisant à une complexité des systèmes.

5. Le paradigme de l'architecture classique (suite)

- ❑ **La complexité:** Le nombre d'inter relations entre éléments devient important et croît plus rapidement que le nombre d'éléments eux-mêmes.
- ❑ Un grand système ou bien un système coûteux n'est pas forcément un système complexe.
- ❑ Complexité importante = difficulté importante.
- ❑ Qualitativement, des techniques de résolution de pbs différentes sont exigées à des niveaux très élevés de complexité que ceux à des niveaux bas.
- ❑ A des niveaux élevés, méthodes d'architecting, les heuristiques basées sur l'expérience, abstraction et la modélisation intégrée doivent être utilisées.
- ❑ L'idée de base de ces techniques tourne autour de « simplifier la solution du problème en se concentrant sur l'essentiel »

5. Le paradigme de l'architecture classique (suite)

- ❑ **Consolider et simplifier les objectifs**
- ❑ **Rester avec les directives**
- ❑ **Ecarter tout ce qui n'est pas essentiel**
- ❑ **Modéliser (abstraction) le système au niveau le plus élevé et progressivement réduire le niveau d'abstraction.**
- ❑ **En finalité, il faut **simplifier**.**
- ❑ **La simplicité s'applique à travers **tout le processus** de développement du système et non pas dans la phase conceptuelle.**

6. Les fondations de l'architecting des systèmes modernes

Bien que les systèmes sont différents, les fondations de l'architecting des systèmes modernes sont les mêmes dans plusieurs disciplines techniques:

- 1. Approche systémique**
- 2. Orientation objectif**
- 3. Méthodologie de modélisation**
- 4. Ultra qualité**
- 5. Certification**
- 6. Perspicacité (insight) (clairvoyance, sagacité)**

6. Les fondations de l'architecting des systèmes modernes (suite)

i) Approche Systémique

- ❑ Approche qui se concentre sur le système comme un tout, particulièrement lorsqu'on fait des jugements de valeur (exigences) et des décisions de conception (qu'est ce qui est faisable).
- ❑ Grace à l'inter relation entre éléments il ya des fonctions du système, ce qui détermine le rôle et les principales responsabilités de l'architecte du système.
- ❑ L'architecture doit être fondée sur les objectifs du propriétaire/utilisateur/client. Ce n'est pas uniquement la structure des composantes.
- ❑ L'architecte doit s'intéresser aux connections et interfaces parce que:
 - i. Permet de distinguer un système de ses composantes
 - ii. Leur addition produit des fonctions niveau système uniques.
 - iii. Spécialistes se concentrent plus au noyau et non pas à la périphérie des sous systèmes.



6. Les fondations de l'architecting des systèmes modernes (suite)

ii) Orientation objectif (purpose orientation)

- ❑ L'architecting des systèmes est un processus conduit généralement par les propriétaires. (ils veulent qlq chose)
- ❑ Pour le succès d'un système, il doit satisfaire un objectif utile à un cout abordable pour une période de temps acceptable.
- ❑ Les 03 critères sont des prérogatives et responsabilités du propriétaire.
- ❑ D'où l'implication du propriétaire dans toutes les phases d'acquisition du système.
- ❑ Des 03 critères, l'orientation objectif reste la plus dominante. S'il n'est pas satisfait, les 02 autres ne le sont pas.
- ❑ L'architecting commence avec (et est responsable du maintien) de l'intégrité de l'utilité et de l'objectif du système.

6. Les fondations de l'architecting des systèmes modernes (suite)

iii) Méthodologie de modélisation

- ❑ **Modélisation** est la création d'abstractions ou représentations du système pour prédire et analyser la performance, les couts, les délais et les risques et pour fournir des directives pour la recherche, le développement, la conception, la fabrication et le management des systèmes.
- ❑ Modélisation est la pièce maitresse de l'architecting du système, c'est un mécanisme
 - ✓ de communication aux propriétaires et constructeurs,
 - ✓ de management de conception avec les ingénieurs et designers,
 - ✓ de maintien de l'intégrité du système avec le management de projet,
 - ✓ et d'apprentissage pour l'architecte lui-même.

Exemples: maquettes, prototypage rapide, modèle informatique d'un réseau de communication, modèle mental d'un utilisateur...

6. Les fondations de l'architecting des systèmes modernes (suite)

iii) Méthodologie de modélisation (suite)

- ❑ **Modélisation ce n'est pas un processus conceptuel uniquement, mais existe à travers le cycle de vie d'un système.**
- ❑ **Il ya des modèles de conception, des modèles d'engineering, modèles pour simulation, prototypes et tests du système, modèles de démonstration, modèles opérationnels et modèles mentaux par l'utilisateur comment le système réagit.**
- ❑ **Modèles sont créés par plusieurs participants, pas uniquement par architectes. Ces modèles doivent être consistants avec les impératives du système, i.e avec le modèle du système de l'architecte.**
- ❑ **Un modèle qui évolue, devenant plus concret et spécifique lorsque le système se construira.**
- ❑ **La modélisation est une activité progressive, à usage multiple, évolutive et devenant moins abstrait et plus concrète lorsque le système est construit et utilisé.**

6. Les fondations de l'architecting des systèmes modernes (suite)

iv) Implémentation de l'ultra qualité

- ❑ Ultra qualité est définie comme le niveau de la qualité tellement demandée de façon à ce que c'est non pratique de mesurer les défauts, plus ou moins certifier le système avant utilisation.
- ❑ C'est un cas limite de pousser la qualité à l'extrême, un état au-delà des limites de qualité acceptables (AQL: acceptable quality limits) et le control de qualité statistique.
- ❑ Exige une approche « Zéro défaut » non seulement en fabrication, mais en conception, en ingénierie, assemblage, test, opération, maintenance, adaptation et mise hors service (cycle de vie).
- ❑ Ultra qualité est la reconnaissance que plus il ya de composants dans le système plus de fiabilité chaque composant doit etre au point où, au niveau élément les défauts deviennent non pratiques de mesurer en temps et ressources disponibles. La fiabilité de tout le système est atteinte.
- ❑ Les conséquences de « zéro défaut » en construction ou fabrication : Pas besoin d'une supportabilité importante (réduction de 30% en cout et temps)



6. Les fondations de l'architecting des systèmes modernes (suite)

iv) Implémentation de l'ultra qualité (suite)

- ❑ L'idée des techniques d'ultra qualité des années 1990 où chaque participant dans la séquence d'acquisition du système est en meme temps acheteur et fournisseur.
- ❑ L'application originale (idée japonaise) est que chaque travailleur dans une ligne de production est un acheteur du travailleur précédent et en meme temps fournisseur au suivant..
- ❑ Chacun exige une grande qualité. Refus d'acheter un article en défaut et d'en délivrer un.
- ❑ Le concept fournisseur-acheteur génère un programme de qualité auto –applicable avec une inspection interne (built-in).
- ❑ Concept applicable aussi à travers l'acquisition de systèmes. De l'architecte à l'ingénieur au concepteur au producteur au vendeur et enfin à l'utilisateur.
- ❑ L'ultra qualité est assurée par la combinaison des méthodes analytiques et heuristiques. La partie analytique est représentée par une analyse à la rupture (failure analysis) détaillée.

6. Les fondations de l'architecting des systèmes modernes (suite)

v) Certification

- ☐ Certification est une déclaration officielle de l'architecte au propriétaire ou utilisateur que le système, construit, vérifie le critère pour l'acceptation du propriétaire et pour le paiement du producteur.
- ☐ Le système est prêt pour utilisation (pour remplir ses missions).
- ☐ Certification c'est la « note de l'examen final » de l'évaluation et test du système.
- ☐ Pour être accepter, il faut qu'il soit bien supporté, objectif et juste pour le propriétaire et constructeur.

6. Les fondations de l'architecting des systèmes modernes (suite)

vi) Clairvoyances (insights) et heuristiques

- ❑ Clairvoyance, où la possibilité de structurer une situation complexe pour la comprendre mieux, est guidée par des leçons apprises des expériences et observations du passé et des autres.
- ❑ Ayant plusieurs leçons, leur sens peut être codifié en expressions simples appelées « heuristiques » (mot grec pour guide).
- ❑ Heuristiques sont complémentaires à l'analytique.
- ❑ Elles ressemblent aux principes dans d'autres arts. (ex: l'importance de la balance et de proportion dans une composition musicale, tableau...).
- ❑ Architecting exige clairvoyance et inspiration et aussi des compétences extraordinaires pour atteindre des niveaux élevés de réalisation.
- ❑ Les meilleurs architectes sont des artistes dans ce qu'ils font.
- ❑ Plusieurs ingénieurs et architectes sont des musiciens

6. Les fondations de l'architecting des systèmes modernes (suite)

vi) Clairvoyances (insights) et heuristiques (suite)

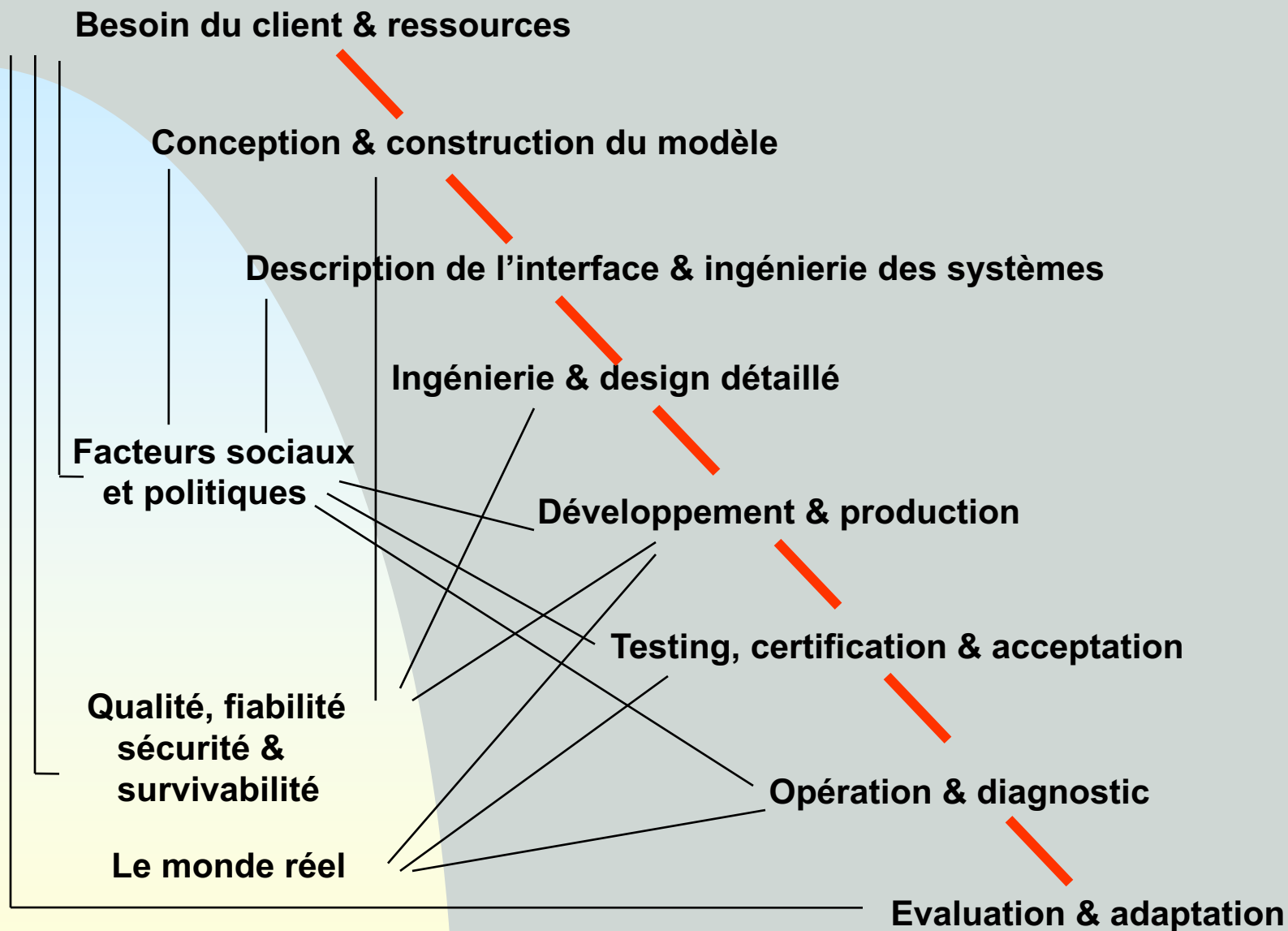
- ❑ Il ya de la sagesse en général.
 - Le succès arrive de la sagesse
 - La sagesse arrive avec l'expérience
 - L'expérience vient des erreurs.
- ❑ Commettre des erreurs pour le succès, chose difficilement acceptée par les clients.
- ❑ Ce sont les erreurs des autres qui nous intéressent.
- ❑ Il faut toute une éducation (organiser, documenter, codifier et enseigner) pour profiter de ces leçons afin de ne pas répéter les mêmes erreurs

7. Résumé du paradigme de l'architecture

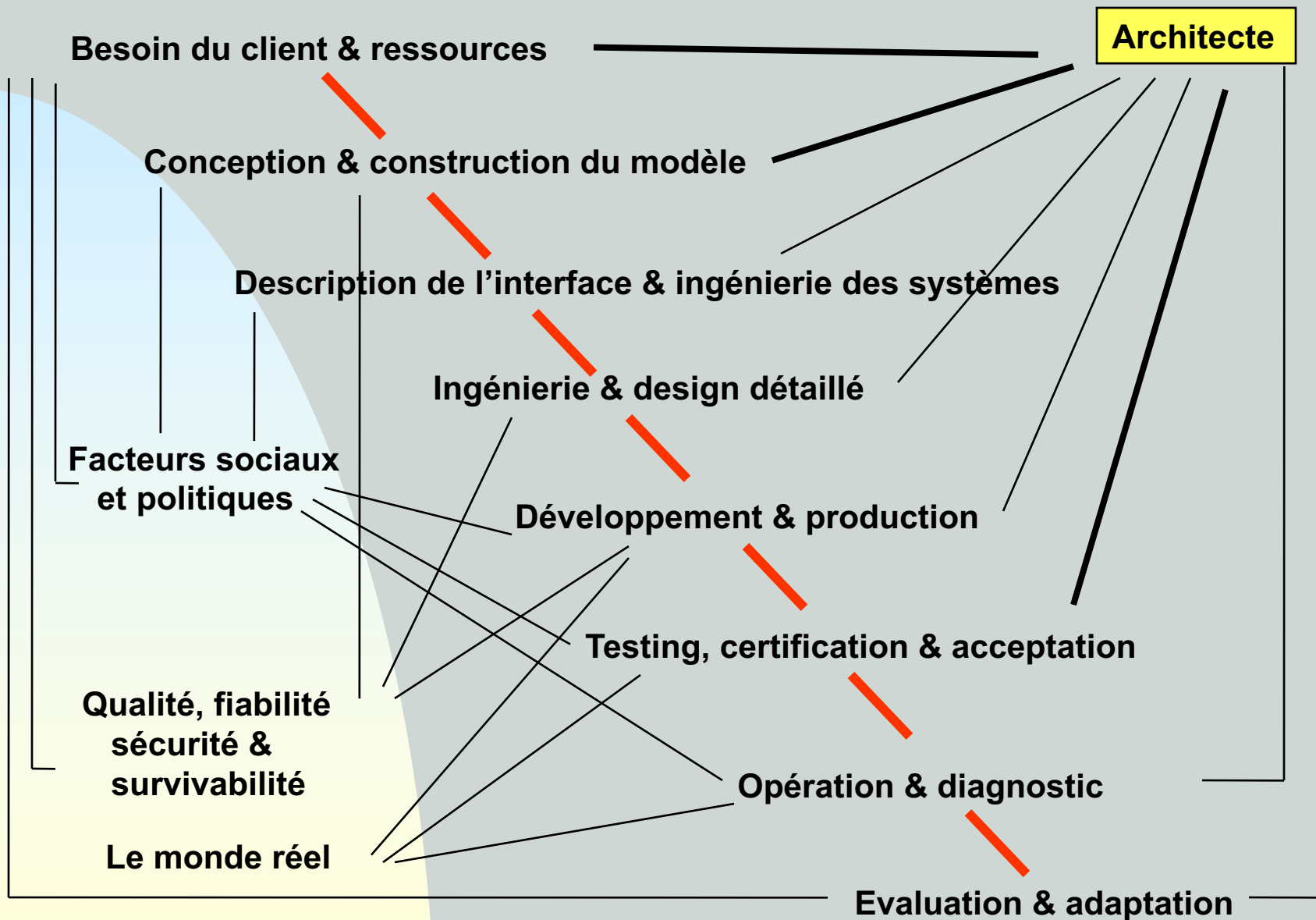
- ❑ L'architecte est principalement un agent du propriétaire pas du constructeur. Il doit agir pour le bien du propriétaire.
- ❑ L'architecte travaille conjointement avec le propriétaire et le constructeur pour définir les pbs et solutions. Les exigences du système sont des outputs pour l'architecting et non pas des inputs. Le propriétaire donne qlqs exigences à l'architecte qui doit l'aider à définir les exigences ultimes.
- ❑ Le produit de l'architecte est une représentation architecturale, un groupe de dessins abstraits du système. Dessins qui ne sont pas généralement prêts pour réaliser qlq chose. Ils doivent être détaillés plus.
- ❑ Le produit architectural n'est pas uniquement des représentations physiques. Il peut inclure le cout, la performance, l'organisation et autres éléments pour clarifier les priorités du propriétaire.
- ❑ Une architecture initiale est une vision. Une description architecturale est un groupe de modèles spécifiques.

8. Modèle en cascade d'acquisition de systèmes (rappel)

- ☐ Architecting prend place dans un contexte de processus d'acquisition . Un modèle classique est en cascades.
- ☐ Chaque processus fait partie d'un processus plus grand et chaque processus a des sous processus.
- ☐ Le processus d'architecting est partie d'une grande activité, l'acquisition de choses utiles.
- ☐ L'acquisition du système est un processus séquentiel qui inclut le design, ingénierie, fabrication, testing et opération. (figure)



Cascade Etendue



L'architecte et la Cascade Etendue

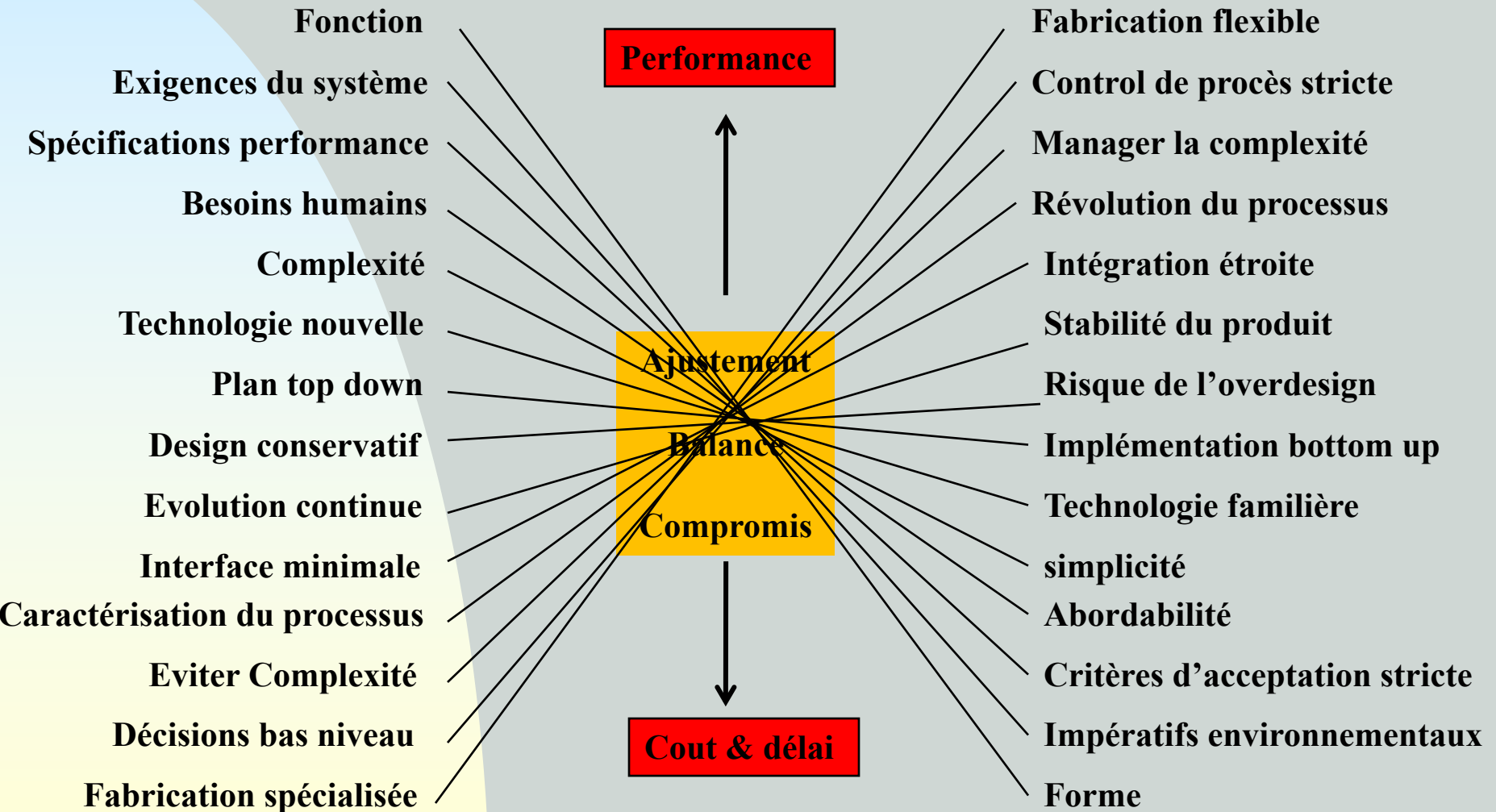
8. Modèle en cascade d'acquisition de systèmes (suite)

- ☐ Relation fonctionnelle entre l'architecte avec un processus plus grand (figure)
- ☐ Managérialement, l'architecte peut être membre de l'organisation du propriétaire, du constructeur ou bien un partenaire d'architecting indépendant où les perceptions des conflits d'intérêt sont à éviter à n'importe quel prix.
- ☐ Dans la relation, les décisions fortes (traits foncés) sont avec besoin du client & ressources, conception & construction du modèle et testing, certification & acceptation.
- ☐ Les relations indiquent les canaux de communication à 02 sens et permettent de déduire les tensions entre les éléments connectés, causées par différents impératifs, besoins et perceptions.

8. Modèle en cascade d'acquisition de systèmes (suite)

- ☐ Il ya certains facteurs techniques compétitifs (figure)
- ☐ Les facteurs en position opposé étirent exactement dans des directions opposées. Ex: évolution continue tire contre stabilité du produit .
- ☐ Une balance typique est celle architecturalement stable, créant la ligne du produit.
- ☐ Décisions à bas niveau tire contre le contrôle du processus stricte qui peut être soulagé par un partitionnement, une agrégation et une surveillance architectural du système.
- ☐ La plupart de ces trades-offs peuvent etre exprimés en termes analytiques.

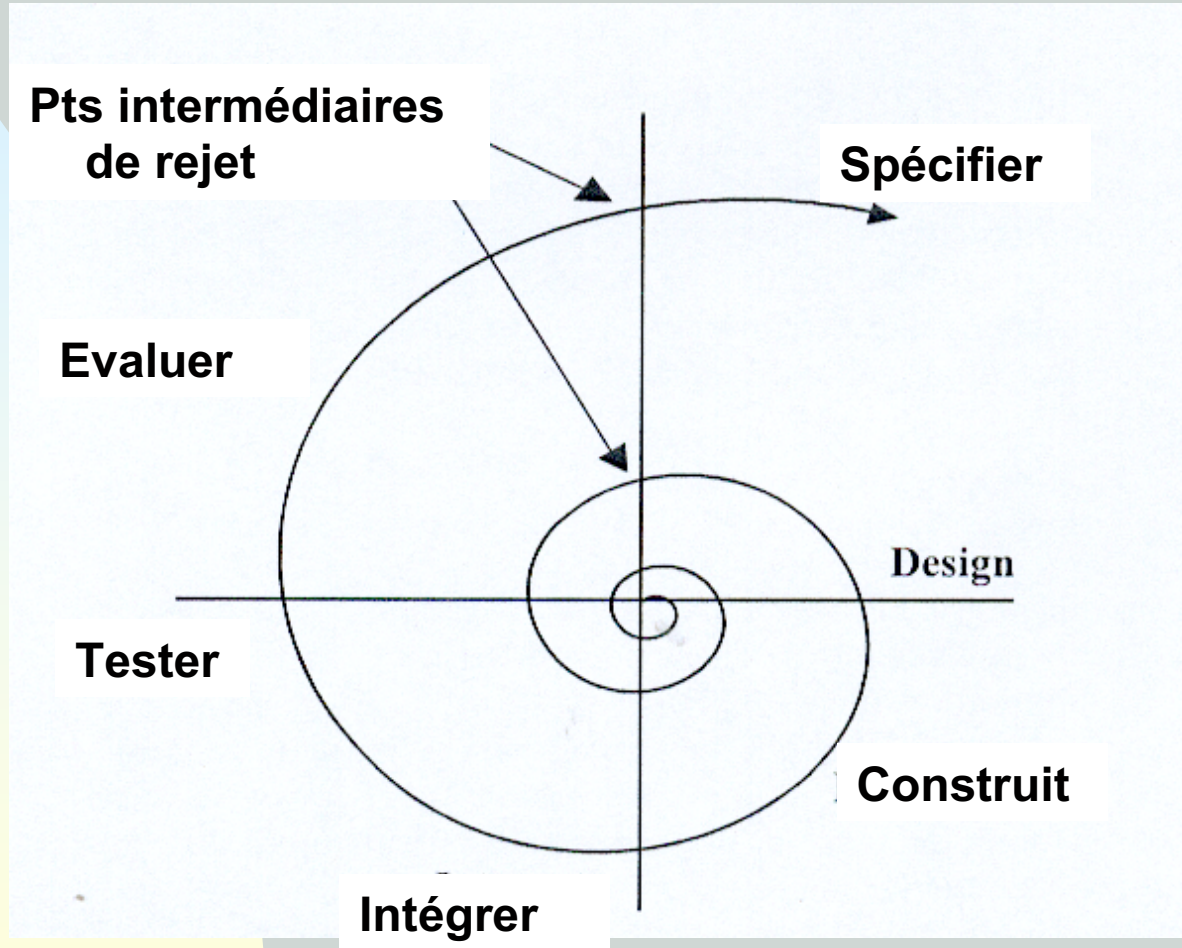
Tensions en architecting des systèmes



9. Assemblage collaboratif, en incréments et en spirale

- ❑ Plusieurs systèmes (spécialement projets software-intensif) ne s'adaptent pas bien au processus séquentiel mais plutôt à des processus hautement itératifs comme la spirale.
- ❑ Il ya une forte incitation au changement du software par itération en réponse à l'expérience de l'utilisateur.
- ❑ s'il ya de nouvelles données (désirs) du marché ou de l'environnement opérationnel, elles sont immédiatement injectées dans le produit.
- ❑ Une des premières formalisations du développement itératif est le modèle en spirale.
- ❑ Le modèle envisage le développement itératif comme une séquence répétée des phases.
- ❑ Au lieu de traverser la séquence d'analyse, modélisation, développement, intégration et le test une seule fois, le software peut retourner autour et autour de chaque phase. Les résultats de chacun sont les données des suivants. (figure)

9. Assemblage collaboratif, en incréments et en spirale (suite)



9. Assemblage collaboratif, en incréments et en spirale (suite)

- ☐ Le modèle est supposé livrer une version stable du produit, à la fin du cycle en spirale.
- ☐ Plusieurs cycles sont utilisés pour le control du risque.
- ☐ L'approche nominale est de poser un certain nbr de cycles (cible) au début du développement et diviser le temps disponible par ce nbr.
- ☐ L'objectif de chaque cycle est de résoudre la partie restante présentant le plus de risque.
- ☐ Ex: si on juge l'acceptation de l'utilisateur, le plus risqué au début du projet, la 1^{ère} spirale doit etre concentrée sur les parties du système qui produit les plus grands éléments de l'expérience de l'utilisateur.
- ☐ Ou bien si le plus risqué élément est le performance technique, le produit du cycle initial doit se concentrer sur la faisabilité technique.
- ☐

9. Assemblage collaboratif, en incréments et en spirale (suite)

- ☐ Plusieurs produits sont livrés de plus en plus. Le cas des hardwares, on l'achète la 1^{ère} fois et au fur et à mesure on reçoit des softwares de mise à jour qui améliorent la performance et la fonctionnalité du système.
- ☐ Ce qui modifie le processus de développement en spirale (qui a une fin définie) à un processus incrémental (pas de fin définie)
- ☐ Le modèle tend vers des cercles (au lieu de spirale) représentant le produit stable à livrer.
- ☐ Après avoir atteint un cercle, un incrément est livré et le processus continue.

9. Assemblage collaboratif, en incréments et en spirale (suite)

- ☐ **Pb d'assemblage.**
- ☐ **Actuellement, plusieurs systèmes sont en assemblage continue, et le processus d'assemblage n'est pas contrôlé directement. Ex: le système INTERNET.**
- ☐ **Le contrôle du développement et du déploiement est fondamentalement collaboratif.**
- ☐ **Organisations choisissent quelle version du produit à utiliser et quand.**
- ☐ **Personne ne peut contrôler l'évolution de l'élément.**
- ☐ **Ce sont des systèmes collaboratifs où le processus du développement est un assemblage collaboratif.**

10. Résumé et Conclusions

- ☐ Architecting créent et construit des structures (structuring).
- ☐ Systems architecting créent et construit des systèmes. Elle s'efforce pour ajustement, balance et compromis entre les tensions des besoins et ressources du propriétaire, technologie et les intérêts des différents intervenants.
- ☐ Architecting est un art et une science en même temps (Synthèse et analyse, induction et déduction, conceptualisation et certification) utilisant des directives de sa partie art et des méthodes de sa partie science.
- ☐ Comme processus, elle se distingue de l'ingénierie des systèmes dans sa grande utilisation des heuristiques, moins d'utilisation d'analytique, liens étroits avec le client.
- ☐ Les fondations de l'architecting des systèmes sont: approche systémique, orientation objectif, méthodologie de modélisation, ultra qualité, certification et clairvoyances.

10. Résumé et Conclusions (suite)

- ☐ Pour éviter perceptions des conflits d'intérêt, les architectes doivent éviter les jugements de valeur, éviter les conflits d'intérêt perçus et garder une bonne relation avec le management du projet.
- ☐ un grand architecte doit être aussi bon qu'un ingénieur et plus créateur qu'un artiste sinon le travail sera incomplet.
- ☐ Ça vient généralement avec l'apprentissage des leçons des autres.
- ☐ Le rôle de l'architecting des systèmes dans le processus d'acquisition des systèmes dépend des phases du processus. Elle est forte pendant la conceptualisation et la certification, mais jamais absente.
- ☐ En l'oubliant à n'importe quel point, et avec n'importe quelle partie du processus d'acquisition conduit généralement à des erreurs d'omission à ce point et aux connectés à lui.

Systems Architecting

Abdellatif MEGNOUNIF

Semaine Prochaine

Les Heuristiques

Merci. Fin du chapitre 2