

Systems Architecting

Abdellatif MEGNOUNIF

Chap. 4

Partie II: Nouveaux domaines, nouvelles idées

**Systèmes avec architecture de
constructeur**

COURS 4 Jeudi 24.11.2011

© **Abdellatif MEGNOUNIF** FT-Tlemcen



1. Introduction

- ☐ Le paradigme de l'architecting classique n'est pas la seule méthode pour créer et construire des systèmes complexes, ni même le seul régime où les architectes et l'architecting est important.
- ☐ Une autre approche différente existe: « the form first »
- ☐ Ça commence avec une architecture conçue dans l'esprit du constructeur au lieu d'un ensemble de buts acceptés par le client.
- ☐ Ces architectes sont généralement des membres du staff technique de la compagnie.
- ☐ Leurs clients sont la compagnie elle-même, bien qu'il faut atteindre une base de clientèle dans le marché.

1. Introduction (suite)

1.1 Développement incrémental pour un client existant

- ❑ La plupart des architectures initiées par les constructeurs ne sont que des variations d'architectures existantes (ex: avions, PC, voitures intelligentes, logiciels...)
- ❑ L'architecture originale a été prouvée par utilisation d'être bonne, les variations et les extensions sont à risque très bas.
- ❑ Une réutilisation importante de modules existants peut se faire, du fait que les hypothèses de design, les fonctions du système et les interfaces restent largement inchangées.
- ❑ les responsabilités de l'architecte restent les memes avec un point en plus important: **l'identification des droits des caractéristiques architecturales sont jugées essentielles à maintenir l'avantage concurrentiel sur le marché.**
- ❑ La question posée sera « **who owns quoi** » . Problème pour le constructeur et le client.

1. Introduction (suite)

1.1 Développement incrémental pour un client existant

- ❑ Aussi avec cette idée on passe de l'approche fonction-à-la forme (orientation objectif) à l'approche forme-à-la fonction (orientation forme).
- ❑ A l'inverse du paradigme classique, dans l'architecting « form-first » le client juge la valeur du produit après (au lieu d'avant) que le produit soit développé et produit, (moins de « test-driven »).
- ❑ Le risque résultant a engendré plusieurs stratégies de réduction de risques.
- ❑ Le plus simple est de présenter, tôt, une démonstration prototype aux clients avec les risques associés aux rejets prématurés.
- ❑ Ex: la stratégie la plus récente est la méthode « open source » pour le design des logiciels.

1. Introduction (suite)

1.2 Nouveaux produits, nouveaux marchés.

- ❑ Il peut y avoir des risques pour les systèmes « first-form » qui créent des changements qualitatifs majeurs au comportement niveau-système. Changement dans le genre et pas du degré.
- ❑ Ce type de systèmes exige généralement de nouveaux lancements dans le design, développement et utilisation. (à recommencer).
- ❑ Ils apparaissent souvent quand il ya de nouvelles technologies disponibles ex: moteurs jet, nouveaux matériaux, microprocesseurs, lasers, architectures software et machines intelligentes.
- ❑ Bien qu'avec les nouvelles technologies il ya rarement de surprises, le plus important **risque** reste **le timing**.
- ❑ Bien que la forme est faisable, introduire un nouveau produit un peu tôt ou bien un peu tard peut être négatif.

1. Introduction (suite)

1.2 Nouveaux produits, nouveaux marchés. (suite)

- ❑ **Ex: La compagnie d'aviation Douglas était très en retard dans l'utilisation du turbo (jet aircraft) devenu très en retard par rapport à Boeing.**
- ❑ **Un nbr important de petites compagnies, voulant utiliser la technologie très tôt, étaient incapables de s'adapter.**
- ❑ **Des systèmes de défense high tech, souvent du à l'utilisation d'une nouvelle technologie prématurément, subissent des couts et délais supplémentaires importants.**

2. Substitutions Technologiques pour les systèmes existants.

- ❑ Un autre risque est que les changements dans l'architecture (technology-driven architectures) ne concerne pas uniquement (et facilement) le remplacement de composantes d'une ancienne technologie une à la fois. C'est plus que ça.
- ❑ L'expérience a montré que sans des changements plus larges dans le système et son management, les initiatives de « technology-driven » rencontrent rarement les attentes voulues et peuvent causer des surcouts pour moins de valeur.
- ❑ Ex: remplacements directs de travailleurs par des machines, de tubes cathodiques par des transistors, d'analystes expérimentés par un système d'information de management automatisé...
- ❑ Il ne peut y avoir de succès que si ces changements sont faits en concertation avec d'autres changements planifiés.

2. Substitutions Technologiques pour les systèmes existants. (suite)

- ❑ **Ex: Automobiles.** Le changement le plus récent est l'introduction d'électronique ultra qualité et de software entre le conducteur et les sous systèmes mécaniques.
- ❑ Cette évolution permet au conducteur de ne pas être en contact (physique) et en control du sous système mécanique.
- ❑ Ça a changé considérablement les caractéristiques globales du système comme la consommation du fuel, style aérodynamique, performance de conduite, sécurité et aussi le design d'élément non prévus comme les moteurs, les transmissions, les pneus, les sièges, les tableaux de bord...
- ❑ Un changement qui permet à l'industrie d'automobile d'arriver à 93% d'équipement de voitures automatisé (fin du siècle).
- ❑ Ceci permet de développer des autoroutes qui doivent permettent une vitesse de circulation plus importante. Chose qui doit arranger beaucoup de clients.



2. Substitutions Technologiques pour les systèmes existants. (suite)

- ❑ **Autre Ex: en communications, les satellites, la fibre optique high-speed, e-mail, le World Wide Web (WWW) ont été combiné pour un accès facile à une communauté globale mais avec un pb croissant concernant l'intimité et la sécurité.**
- ❑ **Les innovations dans les communications seules étaient incapables de faire cette révolution. C'est leur interaction avec les changements dans les procédures de commerce et des habitudes des personnes qui a conduit à la révolution.**
- ❑ **Dans ces exemples, il n'ya pas que le changement des éléments internes. Mais beaucoup d'éléments externes sont touchés. Le management de la fabrication, financement équitable, régulations gouvernementales et la minimisation de l'impact environnemental.**
- ❑ **Ces changements externes peuvent expliquer l'apparition de techniques et d'outils nouveaux dans l'architecting des systèmes.**

2. Substitutions Technologiques pour les systèmes existants. (suite)

2.1 Conséquences de l'incertitude de l'objectif de la fin. (end purpose)

- ❑ L'incertitude de l'objectif de la fin peut avoir des conséquences sérieuses.
- ❑ La plus importante sont les erreurs faites dans les décisions affectant le design, le développement et la production du système.
- ❑ Les systèmes architecturés par les constructeurs sont souvent des solutions cherchant des problèmes et sont généralement vulnérables au pb de « travailler sur le mauvais problème ».
- ❑ L'incertitude dans les objectifs du système les affaiblit comme critère de management du design. Le plus important est de définir une ligne de base des objectifs et essayer de rester sur elle.

3. Réduire les risques de l'incertitude de l'objectif de la fin.

- ❑ Un guide d'architecting puissant pour se protéger contre les risques des objectifs incertains est de construire et maintenir des options.
- ❑ Avec des options disponibles, des décisions tôt peuvent être modifiées ou changées plus tard.
- ❑ Une possibilité est de faire une option pour stopper à des points définis pour garantir au moins une satisfaction partielle des objectifs de l'utilisateur sans perdre beaucoup en temps et argent.
- ❑ Une autre possibilité est de créer des options architecturales qui permettent de faire des ajouts par la suite (ça marche bien dans l'industrie de l'automobile).
- ❑ En software, une stratégie générale est l'utilisation d'architectures ouvertes (open architecture). On les utilise lorsque le marché commence à répondre.
- ❑ Enfin, il ya la partie heuristique: **faire une pause et réfléchir.** Pourquoi c'est dans cet endroit ...

3. Réduire les risques de l'incertitude de l'objectif de la fin. (suite)

3.1 Management du risque par objectifs intermédiaires.

- ❑ Une autre stratégie pour réduire le risque dans ce type de systèmes (introduire nouvelles technologies) est de programmer une série d'objectifs intermédiaires à atteindre par configurations partielles ou précurseurs.
- ❑ Ex: construire des systèmes, des démonstrateurs et des modèles partiellement permet d'évaluer la sensibilité de l'acceptation du client aux jugements de valeur de l'architecte ou constructeur.
- ❑ Les systèmes partiels (précurseurs) doivent être architecturés aussi bien que le produit final. Si leur démonstration est réussie, ça peut donner beaucoup de confiance.
- ❑ **Do the hard parts first**: une heuristique difficile à satisfaire surtout si la partie dure est la fonction unique du système tout entier.

3. Réduire les risques de l'incertitude de l'objectif de la fin. (suite)

3.1 Management du risque par objectifs intermédiaires. (suite)

- ☐ En management du risque, laisser la partie dure à la fin, le risque restera élevé et incertain jusqu'à la fin.
- ☐ La justification du système doit être alors très élevée et son support doit être très puissant.
- ☐ Pour des business privés ça veut dire « high-risk venture capital ».
- ☐ Pour les gouvernements, ça veut dire support par un processus politique, un facteur en acquisition de systèmes où peu d'architectes, d'ingénieurs et de managers techniques sont préparés.

4. Le dilemme « What next? » (Et quoi après)

- ❑ Un des plus importants risques à long terme que peut rencontrer un constructeur pour un système « technologie-driven » réussi est le manque ou bien l'échec de gagner une compétition pour la continuité du succès initial.
- ❑ **La 1^{ère} situation** peut être trouvée dans les compagnies start-up (incubateurs) où il ya manque du successeur du 1^{er} produit.
- ❑ Moins de ressources au début, pas de profit au début pour supporter l'effort de R&D pose pbl.
- ❑ Plus on a du succès avec le premier produit plus sera la compétition surtout des producteurs qui veulent gagner plus sur un produit sure.
- ❑ Tôt le premier produit devient à la portée de tous et sera reproduit à des couts et délais plus courts et donc moins de risques.
- ❑ Ainsi la compagnie start-up ne peut plus suivre son 1^{er} succès et sera donc en faillite ou en vente.

4. Le dilemme « What next? » (Quoi après)

- ❑ **La 2^{ème} situation (l'échec)** peut être trouvée dans les compagnies qui ont l'habitude d'inonder le marché et que leur nouveau système n'arrive pas à décrocher de nouveaux contrats (échec de suivre le succès).
- ❑ Dans ce cas la vraie force de succès d'un système (une architecture fine avec une organisation efficace) peut être sa faiblesse lors du changement de technologie.
- ❑ Les hypothèses et contraintes de l'architecture présente peuvent devenir un pb pour les participants difficile à changer.
- ❑ Pour une compagnie innovatrice, c'est une question de **control des caractéristiques architecturales critiques**.
- ❑ Pour un 1^{er} producteur réussi c'est la façon de connaître (bien à l'avant) qd les **objectifs ont changé** et que des **ré-architecting importants** sont nécessaires. (à expliquer chaque situation)

4. Le dilemme « What next? » (Quoi après)

4.1 Contrôle des caractéristiques architecturales critiques.

- ❑ La partie critique de la réponse d'une compagnie start-up à la question « what next » est le control de l'architecture de son produit à travers la notion de propriété de ses caractéristiques de base.
- ❑ Ex: systèmes opérationnels informatiques (windows...), protocoles de communication, configuration des micro puces, capacité de fabrication unique...
- ❑ Mais pour plus de vente, il vaut mieux que le produit soit améliorer au fur et à mesure. Chose qu'on peut le faire difficilement si on le garde pour soi (difficile de trouver des alternatives soi même). Il vaut mieux passer par une compétition dépendante (vendre les licences) pour trouver de nouvelles applications, ajouter des périphériques, développer le marché et renforcer la base architecturale.
- ❑ Heuristiquement: « **Successful architectures are proprietary, but open*** ».



4. Le dilemme « What next? » (Quoi après)

4.1 Contrôle des caractéristiques architecturales critiques. (suite)

- ❑ C'est la stratégie de Microsoft pour son système opérationnel des PC (open et licence). Alors que Apple refuse de le faire pour son Macintosh.
- ❑ Le résultat est le nbr important de PC clone utilisant Microsoft à l'inverse d'Apple.
- ❑ Mais il ya un inconvénient surtout dans le cas des PC hardware. Pour cela Microsoft travaille en « open » pour les systèmes opérationnels et en « propriété » pour ses programmes d'application.
- ❑ Un autre type de control architectural est celui de la compagnie « BELL telephone ». Sa technologie a été développée par Bell Laboratories, son équipement est produit par Western Electric et ses standards architecturaux sont maintenus par utilisation et régulation.

4. Le dilemme « What next? » (Quoi après)

4.1 Contrôle des caractéristiques architecturales critiques. (suite)

- ❑ Comme le cas de Xerox en photocopieurs, Kodak en caméra, HP en instrumentation...
- ❑ Toutes ses compagnies de ces produits ont commencé petites, ont contrôlé les caractéristiques de base puis ont prospéré rapidement.
- ❑ Donc, pour un innovateur, l'important d'un succès continue n'est pas seulement un bon produit, mais aussi la génération, la reconnaissance et le control de ses caractéristiques architecturales de base.
- ❑ Avec ses points importants, le succès continue et on peut dire que : **there is nothing like being the first success.**

4. Le dilemme « What next? » (Quoi après)

4.2 Abandon d'une architecture obsolète.

- ❑ Une nouvelle stratégie de réduction de risque est demandé pour une compagnie qui perd lorsqu'elle introduit des changements tout en pensant qu'elle améliore les performances, cout et délais.
- ❑ Ça peut arriver sous plusieurs formes. Les objectifs originellement satisfaits peuvent être atteints autrement et de façon meilleure.
- ❑ Les dactylographes (typewriters) sont remplacés par les ordinateurs (traitement de texte). Soit, les hypothèses conceptuelles de l'architecture originale ne sont plus valables, soit qu'il y'avait une grande compétition dans ce sens.
- ❑ Les PC ont révolutionné le marché.
- ❑ Pour cela, il faut qu'il ait une stratégie, bien travaillée à l'avance, pour profiter de l'architecture originale, y compris son organisation et essayer de créer une nouvelle.

4. Le dilemme « What next? » (Quoi après)

4.2 Abandon d'une architecture obsolète. (suite)

- ☐ Pour réussir le changement, il faut avoir une équipe d'architecting innovatrice au temps opportun.
- ☐ Equipe qui a déjà des succès dans le passé et capable de créer un remplacement réussi.

5. Création d'équipes innovatrices

- ❑ Les personnalités des membres de n'importe quelle équipe, et surtout l'équipe d'architecting innovatrice, doivent être compatibles.
- ❑ Selon certains chercheurs, le type de personnalité préférée pour les membres de l'équipe d'architecting est NT.
- ❑ i.e: les membres doivent avoir une analyse stratégique et systématique en résolvant les pbs.
- ❑ Un spécialiste a dit « **Systems architects are made and not born, but some people are more equal than others in terms of natural ability for the systems architecting process** » .
- ❑ Création d'équipes innovatrices n'est pas facile, bien que les membres travaillent bien ensemble.
- ❑ La compagnie start-up (incubatrice), ayant une chance minime, dépend de sa fortune pour recruter des experts, membre critique (expert) pas beaucoup dans la définition de nouvelles directions comme dans la définition des libertés, des droits d'accès, des contraintes, des responsabilités et les prérogatives de l'équipe.

5. Création d'équipes innovatrices (suite)

- ❑ Exemple: à qui l'équipe a à répondre et présenter le rapport et à qui elle ne le doit pas.
- ❑ Si l'équipe ne réussit pas à présenter une nouvelle et acceptable architecture, la compagnie peut échouer.
- ❑ Une des déclarations les plus discutées est « **the best architectures are the product of a single mind** ».
- ❑ Valable pour des projets modestes en taille, pour de grands projets ce n'est pas suffisant.
- ❑ La complexité et la charge de travail pour la création d'architectures de grands projets, multidisciplinaires, technologie-driven va demander beaucoup d'efforts de chaque individu.
- ❑ Ainsi la notion de « **single mind** » (un seul esprit) est remplacée par « **a team of single mind** » ou bien « **of a single vision** », composée d'idées, d'objectifs, de concepts et de priorités.



5. Création d'équipes innovatrices (suite)

- ❑ Dans le cas le plus simple, « single vision » pourra être celle de l'architecte en chef et l'équipe travaillera pour cette vision.
- ❑ Pour la pratique et pour la cohésion de l'équipe, la vision unique doit être l'idée partagée.
- ❑ Dans la vision partagée, l'équipe elle-même (et pas uniquement le chef) doit être vue comme créatrice, communicative, respectée, et avoir un seul esprit de ce que va être le système.
- ❑ Dans ce cas, l'équipe sera crédible dans l'accomplissement de ses responsabilités envers l'entrepreneur, le constructeur, le système et d'autres intervenants.
- ❑ D'autres disent que l'équipe innovatrice doit être **culturelle** en forme, **variée** en nature et presque **obsessive** en dévouement.
 - i. **Culturelle**: équipe caractérisée par une créativité informelle, relations interpersonnelles faciles, confiance et respect (caractéristiques nécessaires pour l'efficacité de l'équipe, échanges d'idées et une identification personnelle avec la vision partagée).

5. Création d'équipes innovatrices (suite)

II. Diversité: dans la spécialisation, une des raisons pour former l'équipe. A part égale, la diversité dans le style et l'expérience en programmation, égalité nécessaire pour assurer un esprit ouvert et pour déclencher une pensée créative dans d'autres.

The team that created and built a presently successful product is often the best one for its evolution – but seldom for creating its replacement.

III. Dévouement: un grand défi pour l'architecte (individuel ou leader d'une petite équipe) est de maintenir le dévouement non seulement à l'intérieur de l'équipe mais aussi dans la structure managériale de support. La vision doit être reformulée de façon continue lorsque de nouveaux participants arrivent comme les ingénieurs, les managers, les producteurs, les utilisateurs et les clients.

Plus difficile, cette vision doit être transformée lorsque le système passe du rêve au concret, à un faiseur de profit et à une production de qualité.

6. Architecting les systèmes « Révolutionnaires »

- ❑ Il faut faire la différence entre architecting dans un environnement précédent ou évolutif et architecting des systèmes sans précédent.
- ❑ les systèmes sans précédents ont un architecte ou bien une petite équipe architecte clairement identifiable.
- ❑ Leurs choix de base traduisent une vision unifiée et cohérente d'un seul individu ou petit groupe.
- ❑ Ces types de systèmes sont « **purpose and technology driven** ».
- ❑ Dans le purpose-driven l'architecte des fois fait partie de l'organisation du développeur des fois non.
- ❑ Dans la technology driven l'architecte est toujours dans l'organisation du développeur.
- ❑ L'architecte a le 1^{er} rôle technique mais ce rôle ne peut pas bien être exprimé en absence d'un bon management de projet.
- ❑ D'où, le duo **manager de projet – architecte de système** est une caractéristique de **systèmes réussis**.

6. Architecting les systèmes « Révolutionnaires » (suite)

- ❑ Cependant dans les systèmes complexes, il est difficile de combiner les 02 rôles.
- ❑ Le manager de projet est confronté aux pbs à court terme alors qu'une bonne architecture nécessite une longue réflexion et communication.
- ❑ La plus importante leçon apprise des systèmes révolutionnaires est que le succès ne se trouve pas là où on a supposé il sera lors de la conception originale.
- ❑ Ex: Ordinateur Macintosh a connu un succès grâce à l'édition de bureau et non pas au marché (comme supposé auparavant).
- ❑ Ce modèle de nouveaux systèmes devient réussi parce que de nouvelles applications sont devenues tellement communes dans l'industrie de l'informatique qu'on leur a donné le nom de « **killer app**(lication) ».
- ❑ Une application (killer app) est une application si précieuse qu'elle entraîne la vente d'une plateforme d'ordinateur particulière.



6. Architecting les systèmes « Révolutionnaires » (suite)

- ❑ De façon générale, une application (killer app) est toute nouvelle utilisation de système si précieuse qui par elle-même entraîne la diffusion du système.
- ❑ Donc un des approches pour les systèmes sans précédents est de trouver « killer app » qui peut entraîner son succès.

7. Architecting des systèmes et recherche de base

- ❑ Il faut bien établir une relation entre les architectes et ceux qui sont engagés dans la recherche de base et le développement technologique.
- ❑ Chacun peut intéresser l'autre.
- ❑ L'architecte peut apprendre sans conflit d'intérêt. Et le chercheur est plus susceptible de devenir conscient des sponsors et utilisateurs potentiels.
- ❑ Nouvelles technologies permettent de nouvelles architectures.
- ❑ Aussi, des architectures innovatrices fournissent le bon raisonnement pour la recherche, souvent à un niveau de base.
- ❑ Bien que l'architecting innovatrice et la recherche de base explorent l'inconnu et le sans précédent, il ya peu de contact au début entre les architectes et chercheurs.
- ❑ Ex: l'architecture de machines intelligentes, l'aérodynamisme chaotique des surfaces actives, la sociologie des systèmes de transport intelligents et la résolution de conflits dans les réseaux multimédias sont des exemples d'intérêts communs

7. Architecting des systèmes et recherche de base (suite)

- ❑ Les nouvelles architectures, conduites par des fins bien perçus, sponsorisent plus de recherche de base et de développement technologique.
- ❑ Ex: systèmes d'armement sponsorisent les sciences de matériaux et l'électromagnétique.
- ❑ L'aviation sponsorise la mécanique des fluides.
- ❑ Les systèmes de l'exploration de l'espace sponsorisent la compréhension et l'acquisition du savoir.
- ❑ Il est donc de l'intérêt des 02 de se connaître.
- ❑ Les architectes gagnent de nouvelles options alors que les chercheurs gagnent des supports bien motivés.

8. Heuristiques pour les systèmes conduits technologiquement (technology-driven)

8.1 Général

- ❑ Une clairvoyance (insight) vaut mieux que 1000 études de marché
- ❑ Le succès est défini par le consommateur et non pas par l'architecte
- ❑ En architecture d'un nouveau programme, toutes les erreurs importantes sont faites les premiers jours.
- ❑ Les plus dangereuses hypothèses sont celles qui sont implicites.
- ❑ Le choix entre les produits dépend sur quel groupe de désavantages les utilisateurs gèrent le mieux.
- ❑ Plus le temps de livraison diminue, plus l'utilité de l'utilisateur augmente.
- ❑ Si vous pensez que votre design est parfait c'est que vous ne l'avez pas montré à quelqu'un d'autres.

8. Heuristiques pour les systèmes conduits technologiquement (suite)

8.1 Général (suite)

- ☐ Si vous ne comprenez pas le système existant vous ne pouvez pas être sûr de construire un meilleur.
- ☐ Fait le plus dur en premier.
- ☐ L'équipe qui a créé et construit un produit réussi présentement est généralement la meilleure équipe pour son évolution, mais rarement la meilleure pour créer son remplacement. (elle peut être bloquée par des hypothèses implicites qui ne tiennent plus).

8. Heuristiques pour les systèmes conduits technologiquement (suite)

8.2 Spécialisée

- ☐ Les bons produits sont insuffisants (leurs caractéristiques doivent être gagnées).
- ☐ Les mises en œuvre ont de l'importance (elles aident à établir le control architectural).
- ☐ Les architectures réussies sont des propriétés mais ouvertes (maintient le control sur les standards, protocoles... qui les caractérisent mais elles les rendent disponibles pour les autres qui peuvent étendre le marché pour le gain des autres).

9. Résumé

- ❑ **Les systèmes conduits technologiquement, architecturé par le constructeur, avec leur grande incertitude d'acceptation par le client, court des risques architecturaux plus importants que les systèmes conduits par objectifs.**
- ❑ **Les risques peuvent être réduits en faisant attention à l'introduction des options, à la structuration des équipes innovatrices et à l'application des heuristiques trouvées intéressantes ailleurs.**
- ❑ **En même temps, ils ont des leçons à apprendre dans le control des caractéristiques du système critiques et la réponse à la compétition rendue possible par de nouvelles technologies.**

Systems Architecting

Abdellatif MEGNOUNIF

Semaine Prochaine

Systèmes de Fabrication

Merci. Fin du chapitre 4