



INOLAM



NUMERILIS
L'expérience digitale

Urbanisation / Architecture de SI



Sommaire

- 1. Introduction**
- 2. Objectifs**
- 3. Architectures : *Principes directeurs***
- 4. Architecture Applicative**
- 5. Logicielle**
- 6. Implémentation : Design Pattern**
- 7. Cas pratiques**



Urbanisation / Architecture de SI : Principes directeurs



Méthodologies d'Architecture (1/2)

- L'architecture a ses écoles, ses styles, ses courants ...
- Dans le domaine de l'architecture de SI aucune méthodologie n'a réussi à s'affirmer avec succès. Ces méthodologies se sont le plus souvent limitées à des projets spécifiques sans parvenir à se généraliser à l'échelle du SI
 - Approche « top down » (du processus au code), avec deux courants principaux:
 - * Approche « Données/Traitements » (Zachman, Merise...) : l'approche « Données/Traitements » centre l'analyse d'un problème sur la donnée manipulée;
 - * Approche « Composants » (RM-ODP, Catalysis...) qui adresse plus spécifiquement les architectures des systèmes distribués : ce modèle a été élaboré sous l'influence du framework Zachman mais guidé par le paradigme Orienté Objet.



Méthodologies d'Architecture (1/2)

Ces deux courants n'ont pas réussi à s'imposer pour deux principaux griefs :

- le dogmatisme : la croyance dans une démarche top-down séquentielle (de la stratégie au code)
- la lourdeur de ces méthodologies : méthodologies verbeuses, manquant de pragmatisme

Urbanisation des fonctions du SI

Les méthodologies de modélisation des processus métier sont le plus souvent basées sur des outils BPM (Business Process Modeling) ou BPR(Business Process Re-engineering) du marché (Mega, Casewise...), voire sur des outils de type Visio ou Word...

Il n'existe pas de standard en matière de méthodologie, chaque société de conseil ou chaque éditeur spécialisé dans le domaine propose généralement sa propre méthodologie de modélisation, qu'elle soit basée sur un formalisme UML ou un formalisme propriétaire...



Méthodologies d'Architecture (2/2)

Dans le domaine de l'architecture logicielle un consensus s'est créé ces dernières années autour du paradigme objet et des méthodologies basées sur UML (Unified Process, RUP) ou eXtreme Programming (XP) principalement pour les raisons suivantes :

- Utilisation d'un langage de modélisation formel et standardisé : UML (Unified Modeling Language)
- Puissance et adéquation du paradigme objet (abstraction, encapsulation) pour les activités d'analyse et de conception qui permet la modélisation à des niveaux successifs d'abstraction
- Démarche itérative, et non séquentielle, entre les phases de recueil des besoins, analyse, conception, grâce notamment aux niveaux d'abstraction proposés par les modèles
- Unification du langage de modélisation UML et des langages de développement (Java, C#, etc.) autour d'un même paradigme (l'objet), ce qui favorise la continuité entre les phases de conception et les phases d'implémentation
- Large utilisation de patterns dans les phases d'analyse et de conception (Analysis Patterns, Design Patterns)



L'urbanisation du système d'information sert de guide à l'architecture applicative

L'urbanisme du système d'information décrit l'agencement des fonctions et des informations et leurs mises en commun, indépendamment de la façon dont elles sont implémentées par le système Informatique

Elle met en évidence :

– des choix de modularité

But : assurer l'évolutivité du système d'information face aux évolutions du métier

– des choix de mise en commun

But : assurer le réemploi et la cohérence d'ensemble

L'architecture fonctionnelle lui sert de guide, mais elle obéit à d'autres contraintes :

- pour tenir compte des Architectures techniques et des performances
- pour tenir compte de l'existence de progiciels du marché

L'architecture applicative décrit l'agencement des traitements et des données