

MTDL - Intro: du développement de systèmes au développement de logiciels

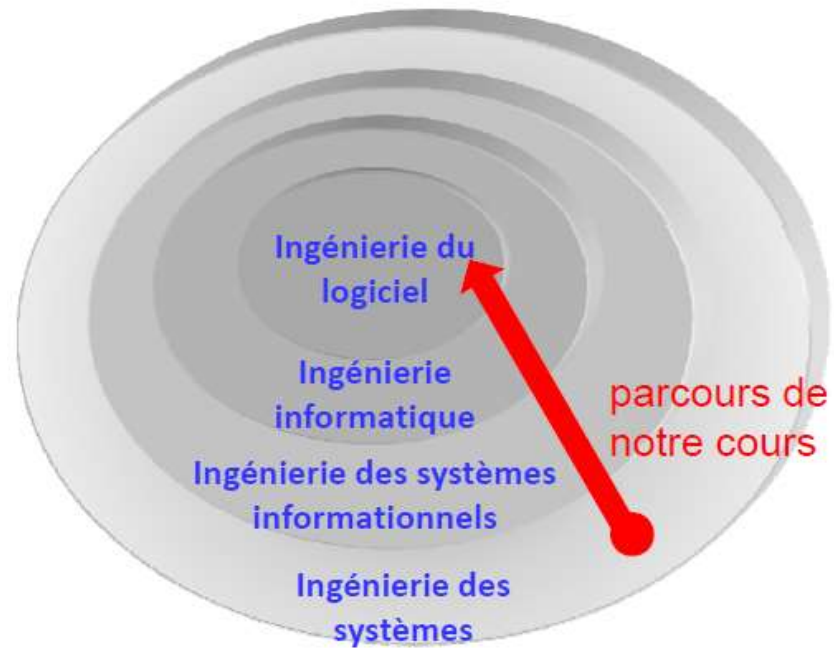
Conf. dr. Maria-Iuliana Dascalu

mariaiuliana.dascalu@gmail.com

www.mariaiulianadascalu.com

Faculté d'Ingénierie en Langues Étrangères, UPB

Quatre Ingénieries

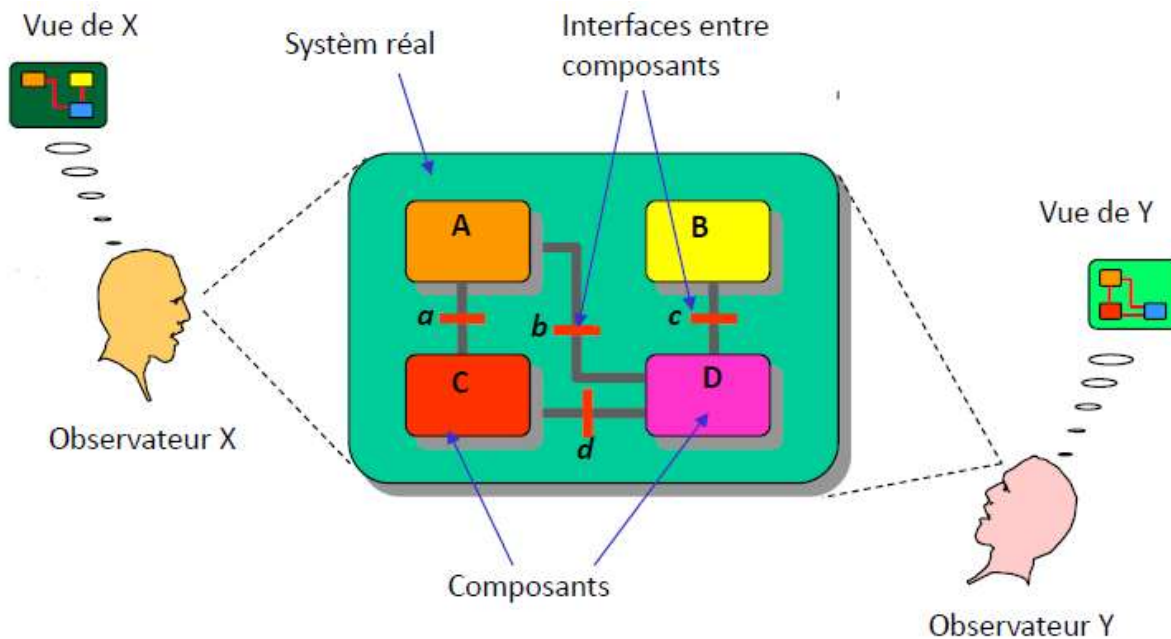


Système (1)

- **Ensemble intégré** d'éléments, de sous-systèmes ou d'assemblages permettant d'atteindre un **objectif défini**. Ces éléments incluent des produits (matériel, logiciels, micrologiciels), des processus, des personnes, des informations, des techniques, des installations, des services et d'autres éléments de support.
- An **integrated set** of elements, subsystems, or assemblies that accomplish a **defined objective**. These elements include products (hardware, software, firmware), processes, people, information, techniques, facilities, services, and other support elements.

(International Council on SE - INCOSE SE Handbook)

Systeme (2)



- État de système = une abstraction d'une situation stable d'un système
- Comportement = une séquence d'états de système (l'évolution du système)
- Une classification possibles des systèmes : naturel et artificiel
- Un observateur utilise sa propre mentale représentation du système qui est une vue simplifiée du système. De là un modèle du système peut être dérivée.
- Un modèle informationnel est fondamentalement une vue abstraite simplifiée de la réalité complexe traduite dans les informations.

Système d'Information (Système Informationnel)

- Un ensemble organisé de ressources permettant d'acquérir, traiter, stocker, communiquer des informations (sous forme de données, textes, images, sons, etc...) dans des organisations. Ces ressources sont inter connectés.
- Les composants principaux:
 - ordinateurs
 - données, compris bases de données
 - systèmes de télécommunications (l'infrastructure de communication)
 - personnel entraîné
 - procédures manuelle et automatisé.
- Le système d'information est donc par nature hétérogène, complexe, évolutif.
- Les organisations et les individus comptent sur les systèmes d'information pour gérer leurs opérations (planification, contrôle, coordination et aide à la décision), concourir dans le marché, fourniture des services, et enrichir la vies des personnes.
- Un système d'information peut être manuel ou même automatisé.

Système informatique, système de logiciel

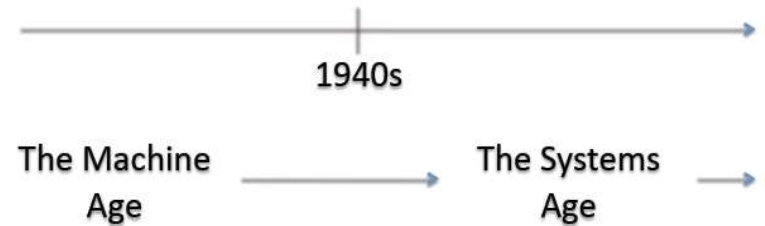
- **Système informatique** = un système d'information informatisé. Il contient:
 - matériel
 - logiciel
 - utilisateurs de logiciel
 - bases de données
 - documentation.
- **Système de logiciel** = un composant d'un système informatique. Il contient:
 - programmes
 - structures de données
 - documentation.

Classification des Systèmes

Category	Subcategory 1	Subcategory 2
Systèmes naturels	Systèmes physiques: galaxies, le système solaire, réseaux des fleuves, les montagnes, les systèmes moléculaires	
	Les systèmes vivants: animaux et plantes	
Systèmes artificiels	Les systèmes sociaux Systèmes d'idées Systèmes de transport Systèmes de communication Systèmes de production	Systèmes manuels d'information Systèmes d'information semi-automatiques Systèmes d'information automatisés

- **système analogique** - Un système qui prend un signal d'entrée à temps continu, et délivre en sortie un signal de sortie à temps continu (un signal est appelé à temps continu si elle est définie à chaque instant).
- **système numérique** ou **système à temps discret**. Il reçoit les signaux d'entrée en temps discret, et produit des signaux de sortie à temps discret (un signal est appelée à temps discret si il n'est défini que pour des points particuliers dans le temps).
- <http://www.technologystudent.com/elec1/anadig1.htm>

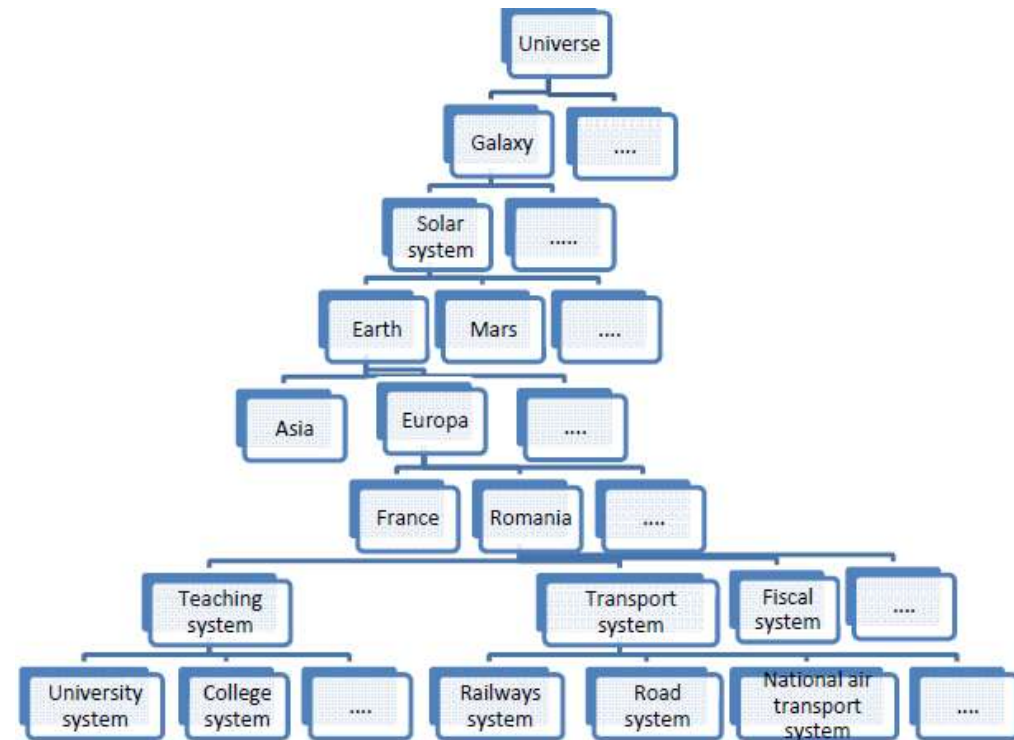
L'histoire



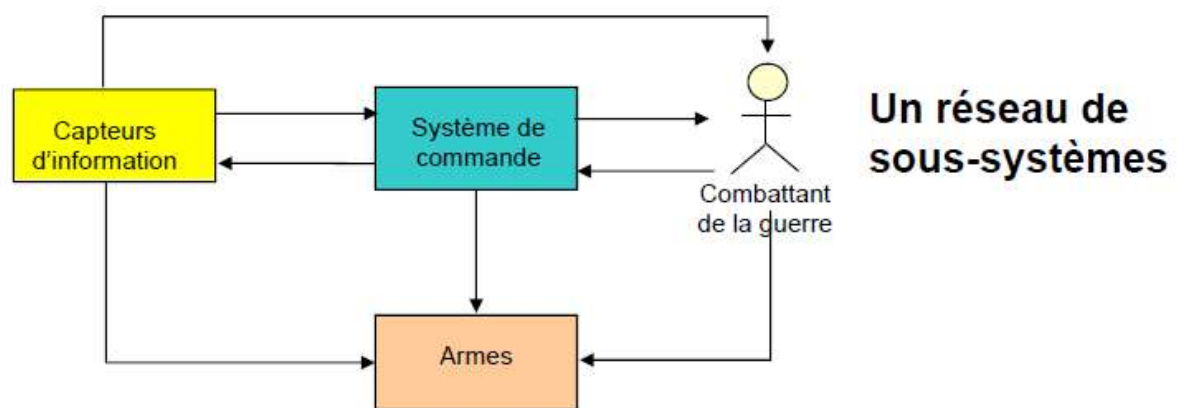
- Origines: après la seconde guerre mondiale
- L'âge de la machine:
 - Réductionnisme: tout peut être réduit, décomposé ou démonté en simples parties indivisibles
 - Pensée analytique: séparez ce qui doit être expliqué, expliquez les plus petites parties, le tout est la somme des parties
 - Mécanisme: cause et effet, pensée déterministe, pensée en système fermé (ignorer l'environnement dans lequel se produit un phénomène)
- L'âge du système:
 - Expansionnisme: tous les objets et événements font partie de plus grands ensembles, vue stochastique des systèmes: un système stochastique est un système dont l'état est non déterministe
 - Pensée synthétique: explique le rôle de ce dernier dans le système plus vaste; le tout n'est pas égal à la somme de ses parties
 - Orientation téléologique: les systèmes ont des objectifs; plus met l'accent sur l'aspect humain de la conception et de la gestion de l'organisation

Principes de l'ingénierie des systèmes

- Systèmes apparaissent dans des hiérarchies.
- Le point de vue de contexte du système (boîte noire-black box) .
- Le point de vue du contenu du système (boîte blanche-white box) .
- A tous les niveaux de la spécification du système, les décisions de conception fournissent des exigences pour les sous-systèmes de composants dans les niveaux inférieurs.
- Tout sous-système d'un système représentent un *partitionnement* du système. Systèmes, à tout niveau de la hiérarchie des systèmes, peuvent être considérés comme des pairs indépendants, formant un réseau. Systèmes du réseau communiquent via l'échange d'informations, du matériel ou d'énergie.
- Le même système peut avoir plusieurs partitionnement selon le point de vue particulier que nous utilisons pour le partitionnement.

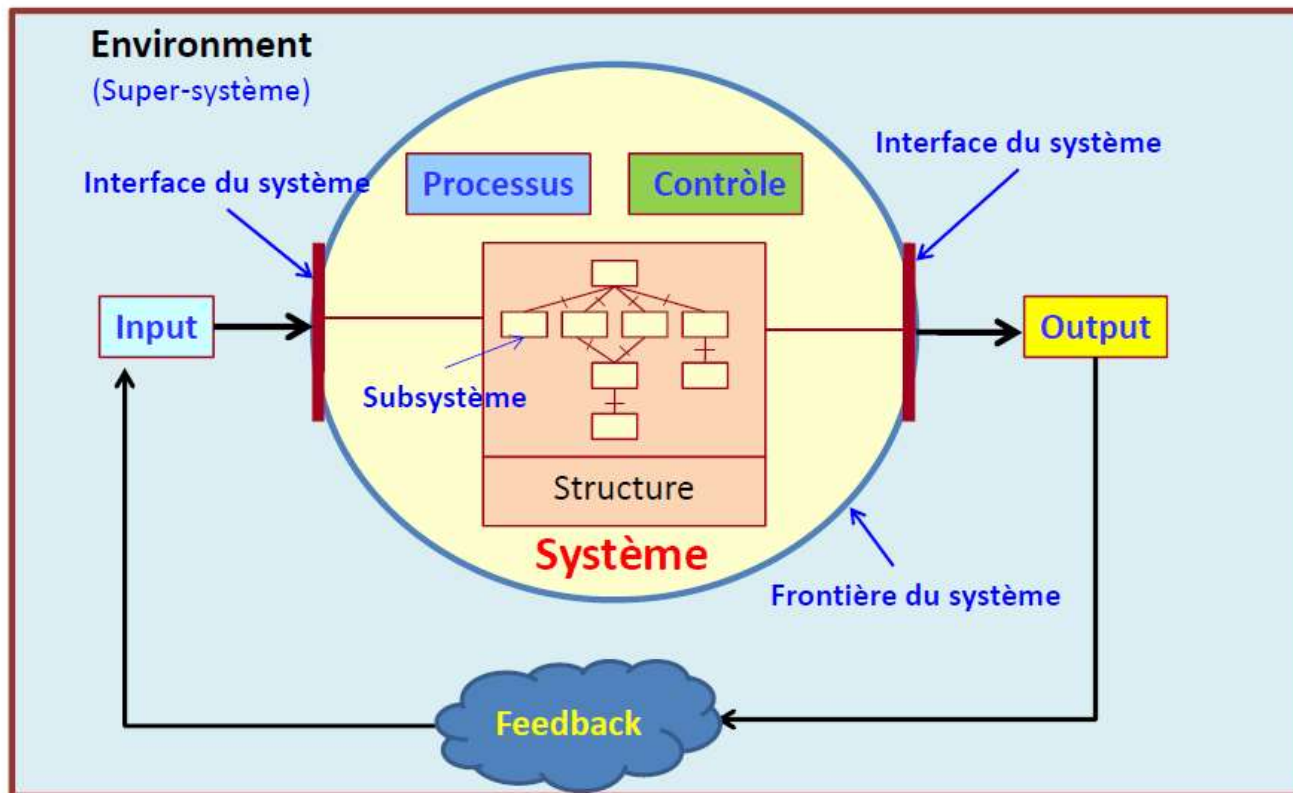


Principes supplémentaires pour les systèmes artificiels



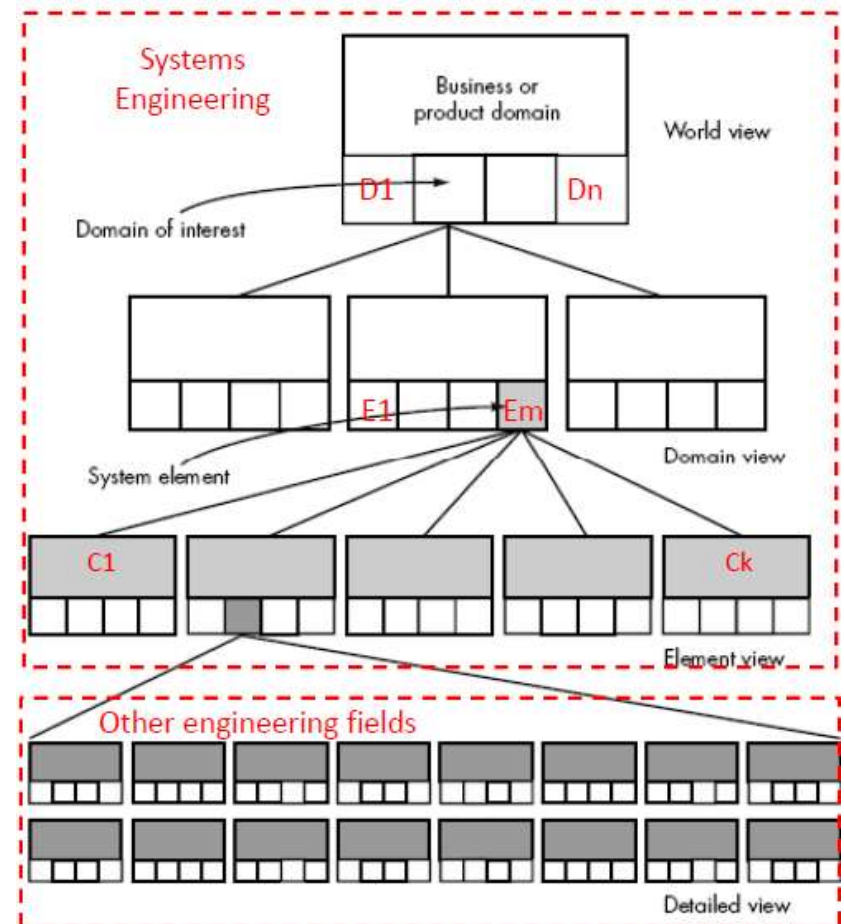
- Système artificiel: Un ensemble organisé d'éléments qui interagissent entre eux et avec leur environnement pour fournir des bénéfices aux gens.
- Les systèmes artificiels répondent aux principes spécifiques:
 - Plus un système est spécialisé, moins il est capable de s'adapter à des circonstances différentes.
 - Plus un système est grand, plus de ses ressources doivent être consacrés à la maintenance quotidienne.
 - Systèmes sont seulement dynamiques et grandissent toujours.

Concepts dans l'Ingénierie des systèmes



La hiérarchie de l'Ingénierie des systèmes

- L'ingénierie des systèmes englobe un ensemble de méthodes top-down et bottom-up pour naviguer dans hiérarchies de systèmes.
- La vision du monde (WV) est composé d'un ensemble de domaines (D_i). Chaque domaine peut être un système ou un système de systèmes. $WV = \{D1, D2, D3, \dots, Dn\}$
- Chaque domaine est composé d'éléments spécifiques (E_j) dont chacun sert un certain rôle dans l'accomplissement de l'objectif et les objectifs du domaine ou d'un composant: $D_i = \{E1, E2, E3, \dots, Em\}$
- Enfin, chaque élément est réalisé en précisant les éléments techniques (CK) qui réalisent la fonction nécessaire pour l'élément: $E_j = \{C1, C2, C3, \dots, Ck\}$. Pour les éléments de logiciel un composant pourrait être un programme informatique, un composant réutilisable de programme, un module, une classe ou un objet.

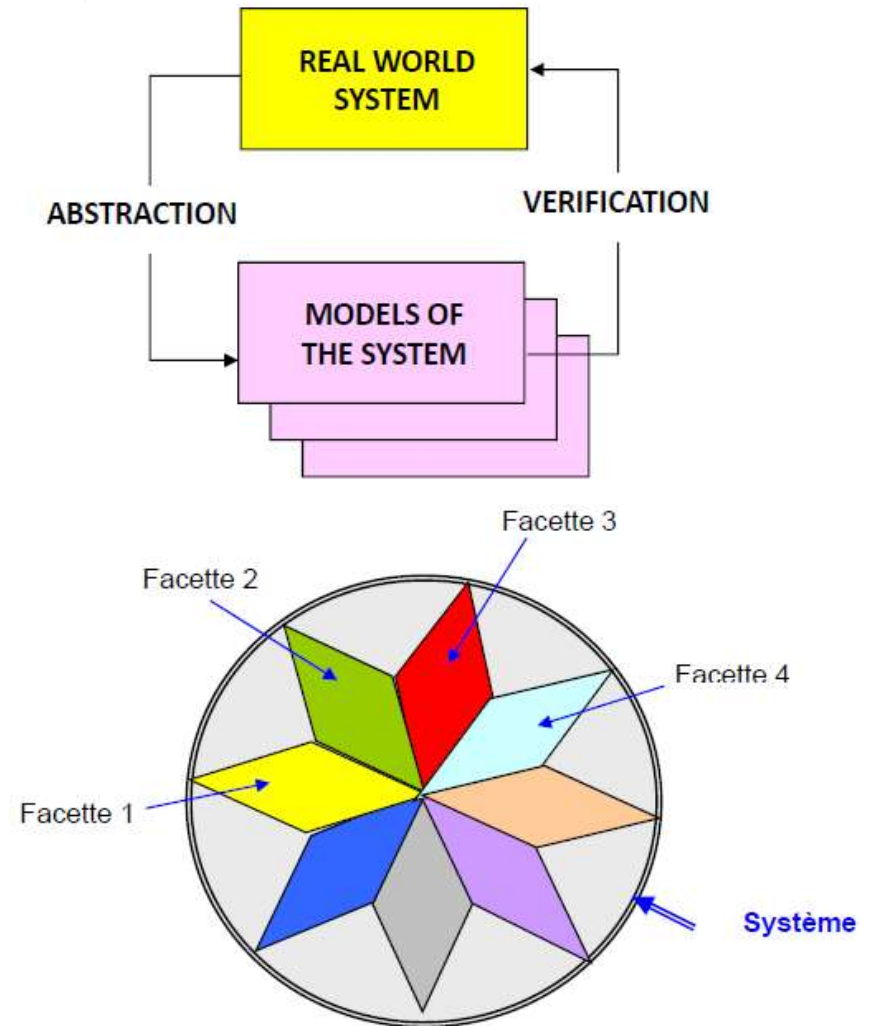


Complexité des systèmes

- Complexité tend à être utilisé pour caractériser quelque chose avec de nombreuses parties en arrangement complexe. La complexité d'un système ne dépend pas uniquement du nombre de parties reliées entre elles, le nombre de relations entre ces parties, mais des aspects non déterministes du système.
- **Classification:**
 - *Systèmes simples*: ceux qui ont toujours la même réponse à un stimulus donné. Le comportement d'ensemble est la somme du comportement de leurs parties.
 - *Systèmes complexes*: le tout est plus que la somme de ses parties. Leur ensemble présente de nouvelles propriétés qu'aucun des composants individuels possèdent.
 - *Systèmes organisés complexes* sont ceux qui, bien que complexe, peut être analysés, synthétisés et étudiés en utilisant des techniques d'ingénierie.
 - *Systèmes complexes non organisés* sont ceux qui sont si complexes qu'elles peuvent être étudiés seulement par les moyennes, les agrégats et les méthodes statistiques.
- **Exemples de systèmes complexes :**
 - colonies de fourmis, les économies et les structures sociales humaines , le climat, le système nerveux, les cellules et les êtres vivants, y compris les êtres humains, ainsi que les infrastructures modernes d'énergie ou de télécommunications.

Modèle

- L'ingénierie des systèmes est basée sur un processus de modélisation.
- *Modèle* = une abstraction d'un système du monde réel qui met l'accent sur certains aspects dans le but de décrire les aspects plus clairement.
- Un modèle a :
 - un objectif (la question que nous voulons qu'il réponde)
 - un point de vue (le point de vue d'un ou plusieurs acteurs: utilisateurs, développeurs etc.).
- Modèles abstraits réduisent la complexité du monde réel à des morceaux plus appropriés pour l'utilisation et plus simples à comprendre.
- Les modèles sont les véhicules de communication entre les différentes parties intéressées dans le système (acteurs, principalement les utilisateurs et les développeurs).
- Nous obtenons une représentation agrégée du système à partir de divers points de vue des systèmes, un ensemble des facettes, reliées entre eux.
- L'origine de facettes est dans le mécanisme abstraction que nous utilisons lors de la modélisation.



Qu'est-ce que l'ingénierie des systèmes ?

- L'ingénierie des systèmes est une approche interdisciplinaire et un moyen de permettre la réalisation de systèmes performants. Il s'attache à définir les besoins des clients et les fonctionnalités requises au début du cycle de développement, à documenter les exigences, puis à procéder à la synthèse de la conception et à la validation du système tout en tenant compte de l'ensemble du problème.
- L'ingénierie des systèmes intègre toutes les disciplines et tous les groupes de spécialités dans un effort d'équipe, formant un processus de développement structuré allant du concept à la production, puis à l'exploitation. L'ingénierie des systèmes prend en compte les besoins commerciaux et techniques de tous les clients dans le but de fournir un produit de qualité qui réponde aux besoins des utilisateurs.
- Systems Engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem.
- Systems Engineering integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation. Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs.

(International Council on SE - INCOSE SE Handbook)

Principes d'ingénierie des systèmes

- Gardez vos yeux sur le prix.
- Impliquer les principales parties prenantes/ stakeholders.
- Définissez le problème avant d'assumer une solution.
- Décomposez le problème en morceaux gérables.
- Retarder des choix technologiques spécifiques.
- Reliez les points entre les exigences et la conception.
- Testez tôt, testez souvent.

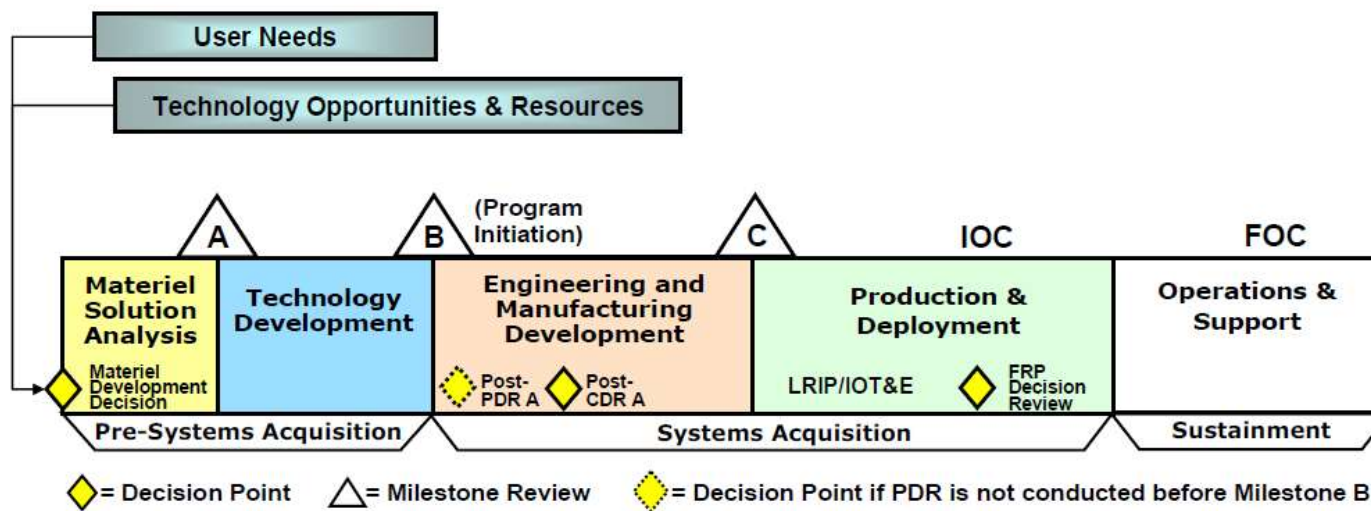
Approches pour le développement de systèmes

- Tous les modèles incluent les mêmes activités: spécification et analyse des exigences, conception, mise en œuvre, intégration, tests (unité, intégration, système, acceptation), mais la séquence et l'importance sont différentes.
- Lors du choix d'un modèle de développement, il faut tenir compte de la taille, de l'urgence et des ressources du projet.

Modèles de cycles de vie du système

- Department of Defense (DoD) Acquisition Management Model (DoD 5000.2)
- International ISO/IEC 15288 Model
- National Society of Professional Engineers (NSPE) Model
- SE Life Cycle Model by Kossiakoff&all

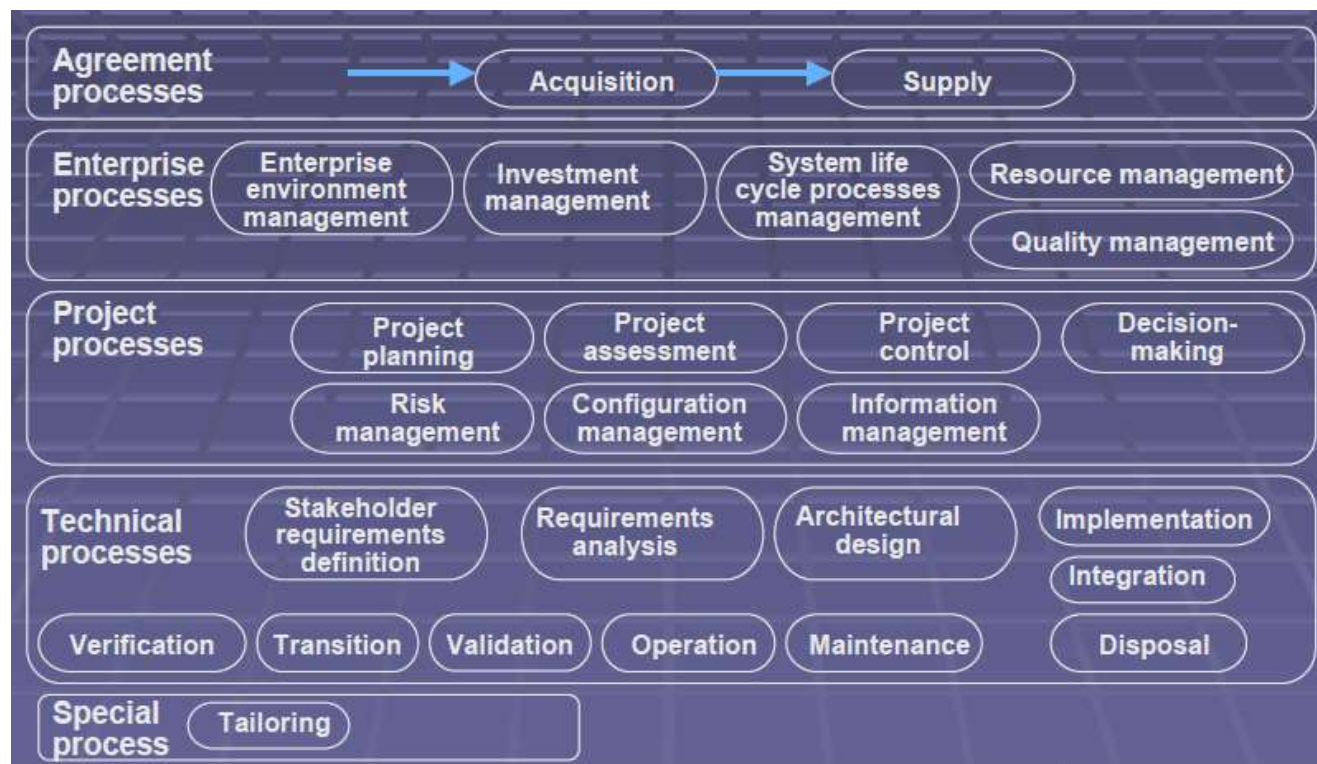
Department of Defense (DoD) Acquisition Management Model (DoD 5000.2)



Source: Operation of the Defense Acquisition System, USA Department of Defense, 2008

International ISO/IEC 15288 Model

- The primary processes:



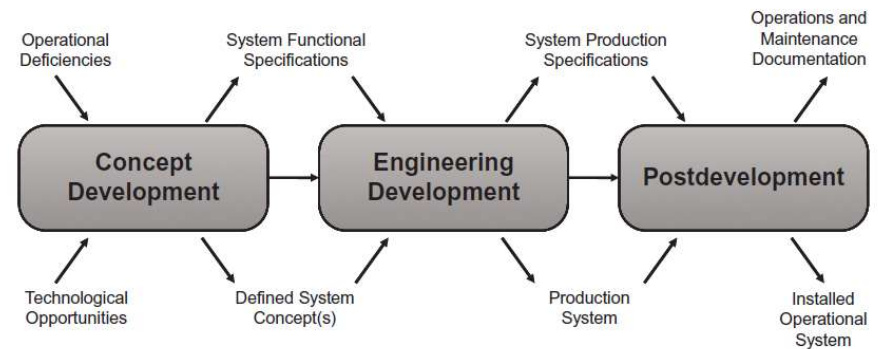
The NSPE Model - National Society of Professional Engineers

- Adapté au développement de systèmes commerciaux
- Séparé en 6 étapes:
 - Conceptuel
 - Faisabilité technique
 - Développement
 - Validation commerciale et préparation du produit
 - Production à grande échelle/ full-scale
 - Support produit

SE Life Cycle Model by Kossiakoff&all

- Séparé en 3 étapes et 8 phases distinctes:
- Développement de concept
 - Analyse des besoins
 - Exploration de concept
 - Définition du concept
- Développement technique:
 - Développement avancé
 - Conception technique
 - Intégration et évaluation
- Post développement:
 - Production
 - Opérations et support

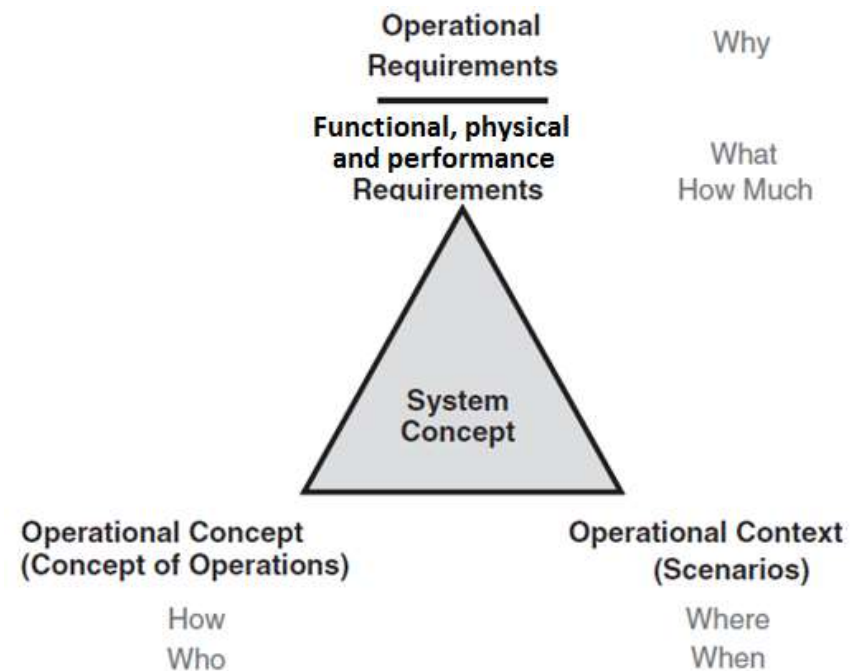
•Entrées et sorties entre processus:



Source: SE Principles and Practice, A. Kossiakoff & all

Types de développement de système

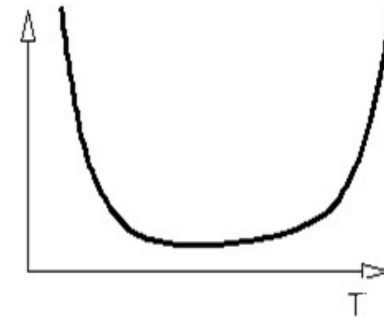
- Développement axé sur les besoins: on découvre une grave déficience dans un système actuel conçu pour répondre à un besoin opérationnel important
- Développement axé sur la technologie: un développement technologique promet une avancée majeure par rapport aux systèmes disponibles pour répondre à un besoin



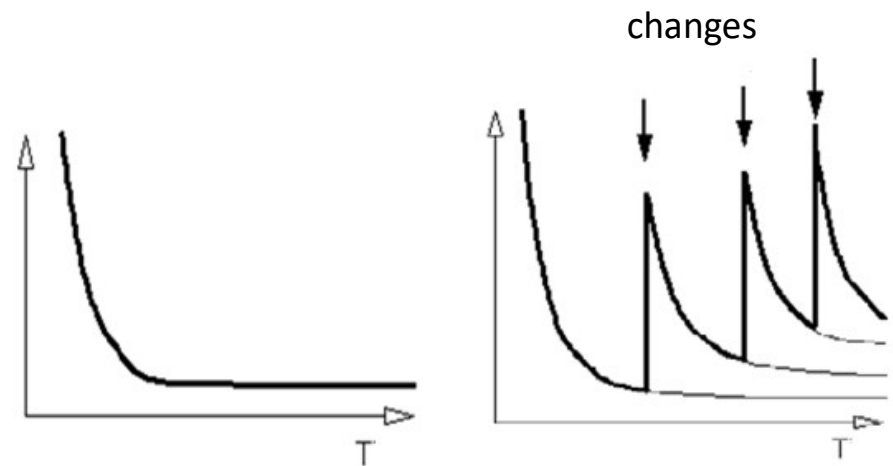
Software

- code source, code exécutable, bibliothèques + documentation (pour conception, installation, utilisation)
- n'est pas un produit physique
- algorithmes, traitement de données / événements - pas d'interactions physiques
- développé, non fabriqué; il n'y a pas de processus de fabrication de logiciel
- les programmes ne peuvent pas être "assemblés" entièrement à partir de composants - ils doivent répondre à des exigences spécifiques et particulières
- pas "vieillissant" mais évoluant à travers la maintenance.

Hardware Fall Rate



Software Fall Rate



Développement logiciel - historique

- Années 50 à 60: le début du développement de logiciels; les premiers langages de programmation et systèmes d'exploitation
- 1965-1985: la période du logiciel "de crise": beaucoup de projets échoués et de sérieux problèmes de développement de logiciels
- 1970-1990: balles d'argent: outils, disciplines, méthodes formelles, processus, professionnalisme
- 1990-2000: boom de l'Internet, nouvelles technologies, systèmes complexes, distribué
- Directions actuelles: Agile, expérimental, dirigée par les modèles

Crise logicielle de longue durée (1)

- Stimulus: développement de hardware permanent et utilisation accrue des ordinateurs ---> la nécessité de développer rapidement des systèmes logiciels complexes et de qualité
- Problèmes ('70-'80 ans)
 - Projets dépassant le temps ou le budget alloué au développement
 - Logiciel inachevé
 - Erreurs majeures / critiques
 - Logiciel de basse qualité
 - Logiciel inefficace
 - Logiciels non conformes aux exigences
 - Coûts de maintenance très élevés et difficultés

Crise logicielle de longue durée (2)

- Conscient que:
 - les méthodes de développement étaient inadéquates
 - le logiciel et son développement ont des caractéristiques différentes de celles des autres produits et disciplines d'ingénierie
 - l'effort de développement augmente plus que linéaire avec la taille du programme
 - un programme n'est pas une entité statique, il évolue dans le temps en raison de l'évolution des exigences et de l'environnement d'utilisation
 - le code doit être compris et adapté par des personnes différentes de celles qui l'ont développé
- G. Booch: *“building and maintaining software is hard and getting harder; building quality software in a repeatable and predictable is harder still”*

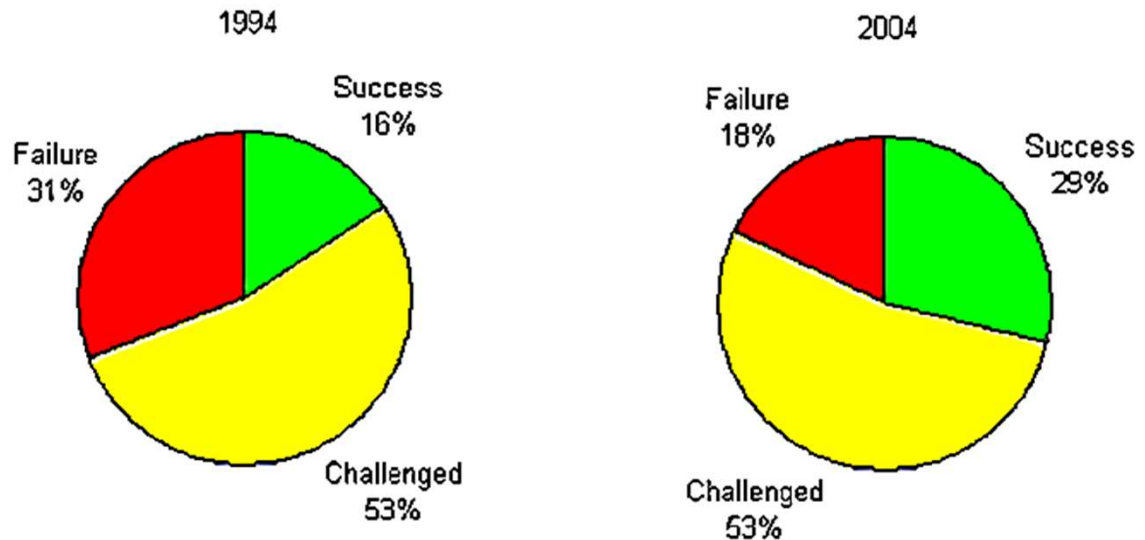
Résultat: Génie logiciel

- Le développement de Sw commence à être perçu comme une industrie distincte
 - Le logiciel traité comme un produit d'ingénierie spécifique
 - Cadre pour la construction de logiciels: Génie logiciel (Génie)
 - Définition des techniques de production justifiées par la théorie et la pratique
-
- Génie logiciel (voir norme IEEE 1993): "Appliquer une approche systématique, disciplinée et mesurable dans développement, exploitation et maintenance de logiciels "

Pour surmonter la crise du logiciel...

- Clarification d'activités spécifiques (développement SW)
- Modèles de développement (guides)
- Méthodes et langages d'analyse et de
- Langages de programmation
- Principes et techniques de réutilisation des composants logiciels
- Environnements de développement intégrés (IDE)
- Générateurs d'applications
- Environnements de test
- etc.

Taux de réussite / échec: <https://www.infoq.com/articles/standish-chaos-2015>



	2011	2012	2013	2014	2015
SUCCESS	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

Activités dans un projet de logiciel

- Activités techniques:
 - Analyse et spécification des besoins
 - Conception
 - Mise en œuvre des unités de programme (modules)
 - Intégration
 - Test du système, test d'acceptation etc
 - Maintenance et exploitation
- Activités d'assurance qualité
- Activités de gestion de projet

Analyse et spécification des exigences

- Besoins de l'utilisateur:
 - Décrivez le point de vue des utilisateurs du futur produit.
 - Il existe des exigences en matière de: performances, performances, sécurité, interface utilisateur, autres interfaces, etc.
 - Défini dans: URD (User Requirements Document) - Document sur les exigences de l'utilisateur, parfois appelé Document de définition système.
 - URD → partie du contrat client
 - URD → base pour les tests de réception.
- Exigences logicielles
 - Ils sont plus précis, techniques; sont déterminés par l'analyse, la reformulation plus technique de l'URD.
 - Spécifié dans SRD (Software Requirements Document); parfois, pour les systèmes très complexes avec composants hw et sw, le document de configuration système est également créé
 - Ils sont utilisés pour clarifier et planifier le développement
 - SRD → Base de test du système
- Les exigences (URD, SRD) sont "abstraites" - indépendantes de la mise en oeuvre (sauf pour certaines contraintes).

Conception

- Conception architecturale
 - Définir l'architecture du système qui implémentera les exigences:
 - technologies
 - sous-systèmes (modules principaux)
 - Choisir la solution optimale parmi les alternatives possibles
 - Toutes les exigences doivent être couvertes par l'architecture
 - Résultat: ADD (Architectural Design Document): les sous-systèmes, le rôle et les interfaces de chaque module, le mode de coopération entre les modules.
 - ADD est utilisé pour spécifier des tests d'intégration
- Détails de conception
 - Conceptions individuelles et décomposition successive de modules en modules de plus en plus petits, jusqu'aux unités d'implémentation (en fonction du projet)
 - Résultat: DDD (Detailed Design Document): le rôle et les interfaces de chaque module / sous-module, la manière dont les modules fonctionnent ensemble.
 - DDD est utilisé pour définir les tests unitaires

Mise en œuvre des unités de programme (modules)

- = encodage + tests unitaires (individuels) de modules
- les activités les plus maîtrisées et "équipées":
 - langages de programmation, compilateurs, débogueur
 - IDE (Environnements de développement intégrés)
 - outils logiciels pour les tests unitaires
- ~ 15-20% de l'effort global de développement d'un logiciel

Intégration

- Les modules qui ont été testés indépendamment sont progressivement intégrés dans des sous-systèmes, jusqu'au niveau du système.
- La communication et l'interaction entre les composants intégrés sont testés.

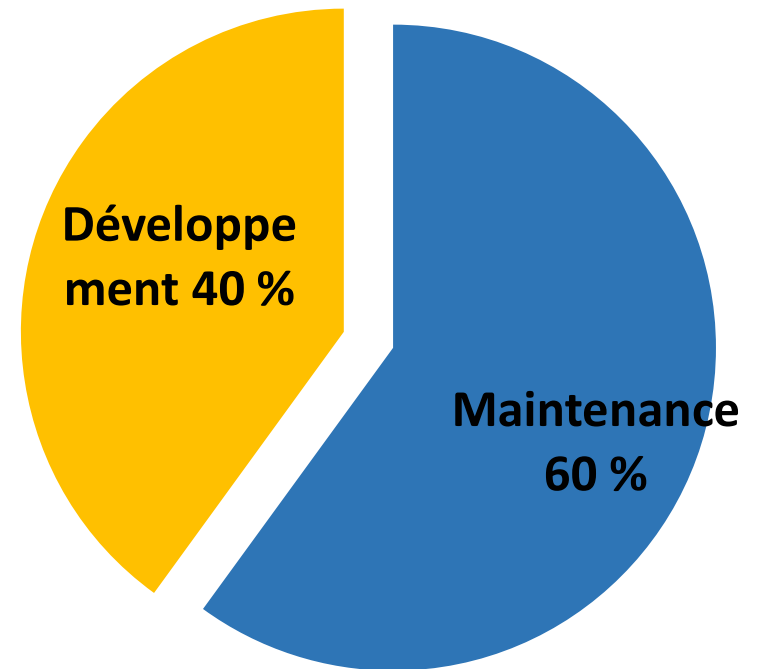
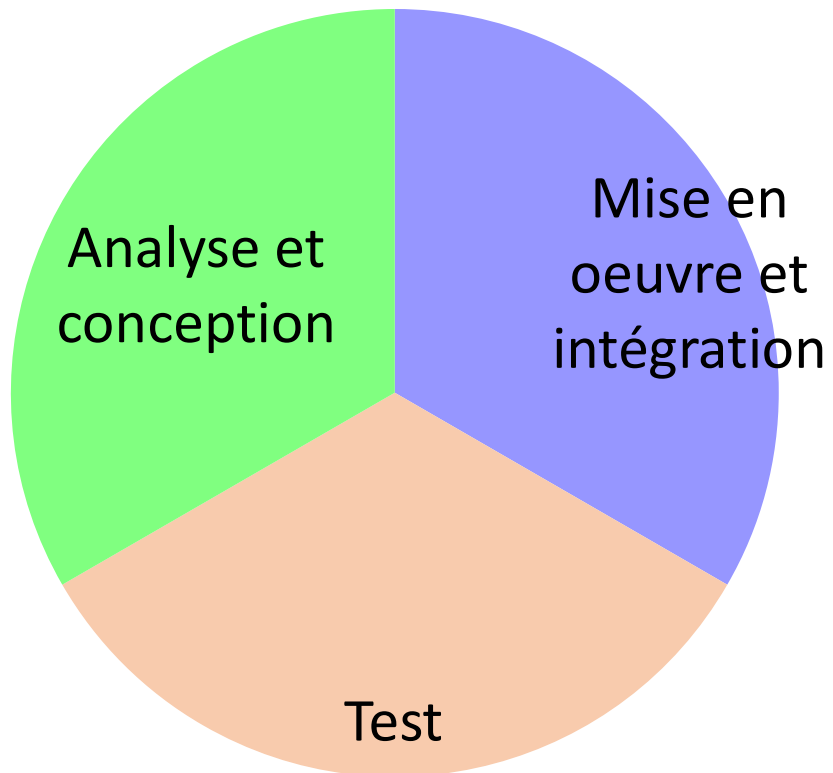
Test

- Tests unitaires
- Tests d'intégration
- Test du système
 - Fait à l'intérieur du développeur
- Test d'acceptation
 - Il est réalisé par une équipe de test incluant le client.

Maintenance et exploitation

- Utilisation du logiciel dans le monde réel
- Changements:
 - Correction des bugs découverts
 - Améliorer les fonctionnalités existantes
 - Adaptation aux nouvelles technologies
 - Ajouter de nouvelles fonctionnalités
- La maintenance peut être effectuée par des développeurs autres que ceux d'origine

Distribution d'effort



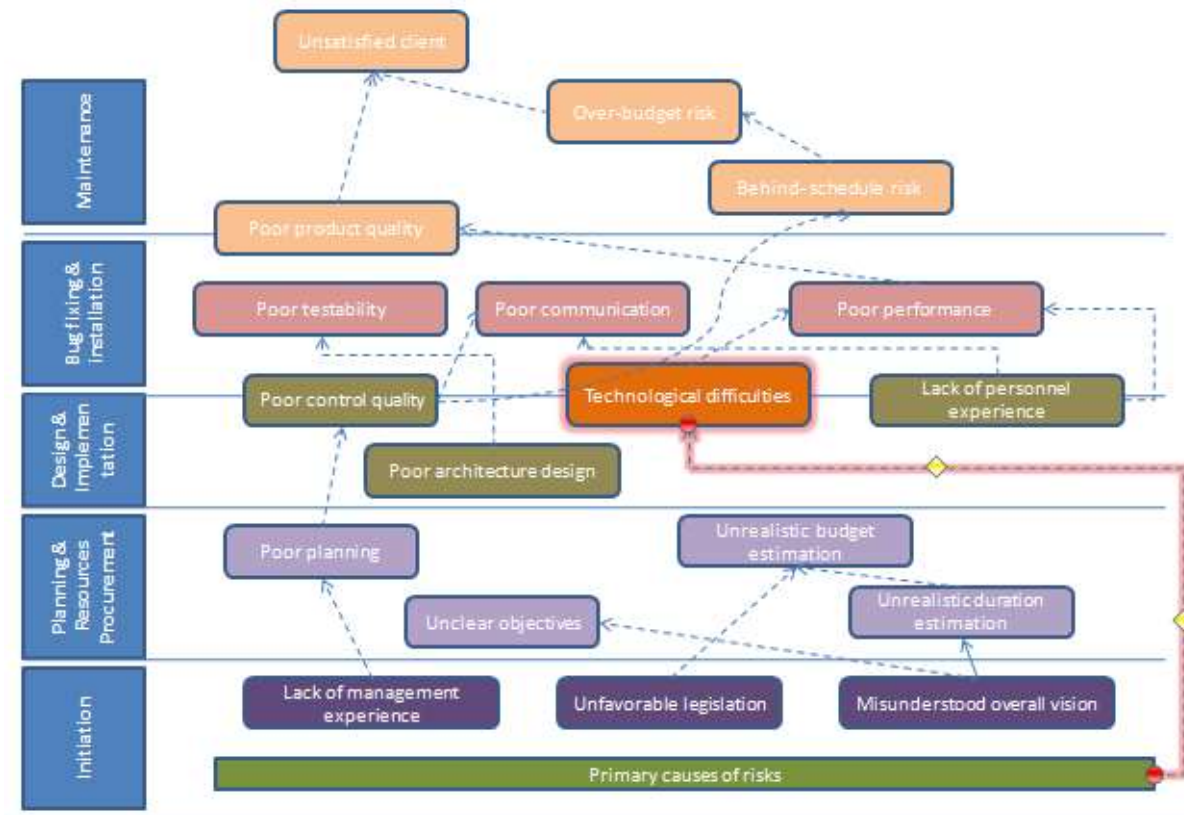
Assurance qualité

- Objectif: garantir les exigences techniques et les normes de qualité pour:
 - dans le processus de développement
 - le produit final
- Activités:
 - Choix des méthodes et des normes de spécification, de conception et de mise en œuvre
 - Révisions tout au long du processus de développement
 - Définir des stratégies de test
 - Définir des méthodes de documentation
 - Définir les mesures de mesure pour les produits et les outils de mesure

Gestion de projet

- Estimation des besoins en ressources
- Mise en scène et planification
- Sélection et évaluation du personnel
- Suivi du développement
- Identifier et surveiller les risques
- Allocation de (re) ressources
- Révisions
- Rapports

Des risques



Source: CN Bodea, MI Dascalu - IT Risk Evaluation Model Using Risk Maps and Fuzzy Inference, International Journal of Information Technology Project Management (IJITPM) , vol. 1 (2), pp. 79-97, 2010