

lab-sticc.univ-brest.fr/~babau/

Ingénierie du Développement Logiciel

Moodlescience/Informatique/M1/ [Ingénierie Du Logiciel](#)

Jean-Philippe Babau

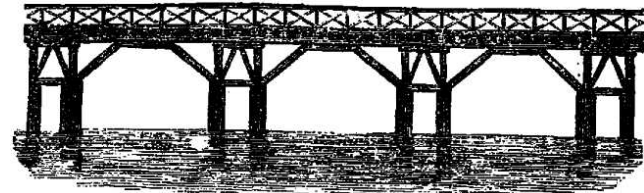
Département Informatique, UFR Sciences, UBO
Laboratoire Lab-STICC

Construire un pont

1



2



Pont Sublicius, Rome, 650 avant JC

3



le Ponte dei Salti (Lavertezzo, Suisse)



4



Viaduc de Millau

Mike Lehmann, [Mike Switzerland](#) March 2008

jean-philippe.babau@univ-brest.fr

Des outils pour la construction et pour l'organisation de la construction

1



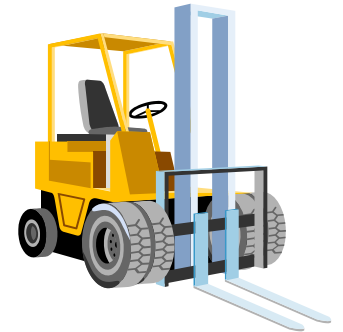
2



3



4



Construire une maison : documents et suivi

- Un projet : une idée de maison
- Des normes et règlements
 - Déclarations obligatoires
 - Documents de conformité
- Des plans
 - Schéma électrique, eau, chauffage, aération
 - Plan cadastral
 - Contrat de construction de maison individuelle « **avec fourniture de plans** »
- Des études techniques
 - Résistance des matériaux
 - Capacité d'isolation
- Des documents spécifiques
 - Contrats, actes, déclarations, plans, ...
 - Pour le notaire, l'état, le paysagiste, les techniciens, les constructeurs, les vendeurs
- Gestion du chantier
 - Planning, revues à base des contrats et des plans

Des contraintes légales sur le logiciel

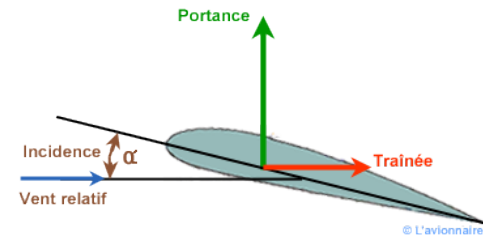
- **RGPD** : « Le présent règlement établit des règles relatives à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et des règles relatives à la libre circulation de ces données. » CNIL
- La loi de finances pour 2018, qui vise à l'obligation de l'utilisation d'un logiciel de caisse certifié **NF 525**, s'applique également aux sites marchands.
- Certains de ces logiciels sont des dispositifs médicaux (DM) ou des dispositifs médicaux de diagnostic in vitro (DM DIV). Ils doivent, de ce fait, être marqués CE comme tous les DM et DMDIV. **Le marquage CE atteste de leur conformité à la réglementation.** Ils entrent alors dans le champ de surveillance de l'**ANSM**.
- Les normes **ED-12C** et **DO-178C** (*Software considerations in airborne systems and equipment certification*) développées en commun et éditées respectivement par EUROCAE et RTCA Inc., fixent les conditions de sécurité applicables aux logiciels critiques de l'avionique dans l'aviation commerciale et l'aviation générale.

Construire un avion : des modèles ...

- Aérodynamique
 - Modèles physiques de l'écoulement



Source : Photo ONERA, H.Werlé



$$\text{Drag} = \frac{1}{2} \rho V^2 C_D A$$

1) PFD: $P + P_{\text{stat}} S_1 = P_1 S_1 \Rightarrow P_1 = \frac{4P}{\pi d_1^2} + P_{\text{stat}}$ A.N. $P_1 = \frac{4 \cdot 62,84}{\pi \cdot 0,04^2} + 10^5 = 1,5 \text{ bar}$

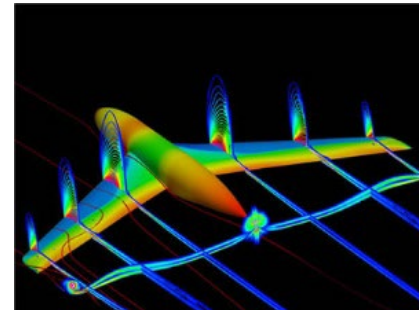
2) Equation de continuité: $V_1 S_1 = V_2 S_2$

$$\Rightarrow V_1 = V_2 \cdot \frac{S_2}{S_1} = V_2 \cdot \left(\frac{d_2}{d_1}\right)^2 \Rightarrow V_1 = \left(\frac{1}{4}\right)^2 V_2 \Rightarrow V_1 = \frac{1}{16} V_2$$

3) Equation de Bernoulli: $\frac{V_2^2 - V_1^2}{2} + \frac{P_2 - P_1}{\rho} + g(Z_2 - Z_1) = 0$ or $Z_1 = Z_2$ et $P_2 = P_{\text{atm}}$

et $V_1 = \frac{1}{16} V_2$ donc $V_2 = \sqrt{\frac{512 \cdot (P_1 - P_{\text{atm}})}{\rho}}$ A.N. $V_2 = \sqrt{\frac{512 \cdot (1,5 \cdot 10^5 - 10^5)}{1000}} = 10 \text{ m/s}$

4) $Q_v = \frac{\pi d_2^2}{4} V_2$ A.N. $Q_v = \frac{\pi \cdot 0,01^2}{4} \cdot 10 = 0,785 \cdot 10^{-3} \text{ m}^3/\text{s}$



Construire un avion : des modèles ...

- **Aérodynamique**
 - Modèles physiques de l'écoulement
 - Modèles physiques de la structure
- **Motorisation**
 - Modèles de combustion
 - Modèles thermiques
- **Systèmes de pilotage et de navigation**
 - Normes de sécurité
 - Normes de communication
- **Informatique**
 - Embarquée et débarquée (suivi) : sous-partie du système
 - Communications embarqué / débarqué
 - Aide à la conception : outils de manipulation des modèles physiques
 - SI des entreprises : outils de suivi des projets et des personnes
- **Intervenants hétérogènes**
 - Métiers
 - Cultures et langues



Construire un système

- Des systèmes de plus en plus complexes ... à produire de plus en plus vite
 - des systèmes personnalisables
- Des outils de conception et modélisation de plus en plus complexes
- Des schémas hétérogènes et spécifiques
 - pour évaluer, étudier
- Des acteurs divers et différents aux points de vue divers et différents
- Faire les bons choix au bon moment
- Maîtriser la gestion de projet
- *Automatisation des étapes et des processus*
- Des documents pour contractualiser et communiquer
- Des outils pour le suivi
 - coûts, délais, interaction avec les divers partenaires

Construire un logiciel

- Des systèmes de plus en plus complexe ... à produire de plus en plus vite
 - des systèmes personnalisables
- Des outils de conception et modélisation de plus en plus complexes
- Des schémas hétérogènes et spécifiques
 - pour évaluer, étudier
- Des acteurs divers et différents aux points de vue divers et différents
- Faire les bons choix au bon moment
- **Intégrer une vision Métier et une vision Informatique**
- Maîtriser la gestion de projet informatique
 - **Processus spécifiques**
- *Automatisation des étapes et des processus*
- Des documents pour contractualiser et communiquer
- Des outils pour le suivi
 - coûts, délais, interaction avec les divers partenaires

Le développement d'un logiciel : un projet qui concerne de nombreuses personnes

- Le **client** du produit développé
- Les **utilisateurs**
- Les administratifs
 - Coté client
 - Coté fournisseur

- Les administrateurs
 - Coté client
 - Coté fournisseur

- Les instituts de normalisation

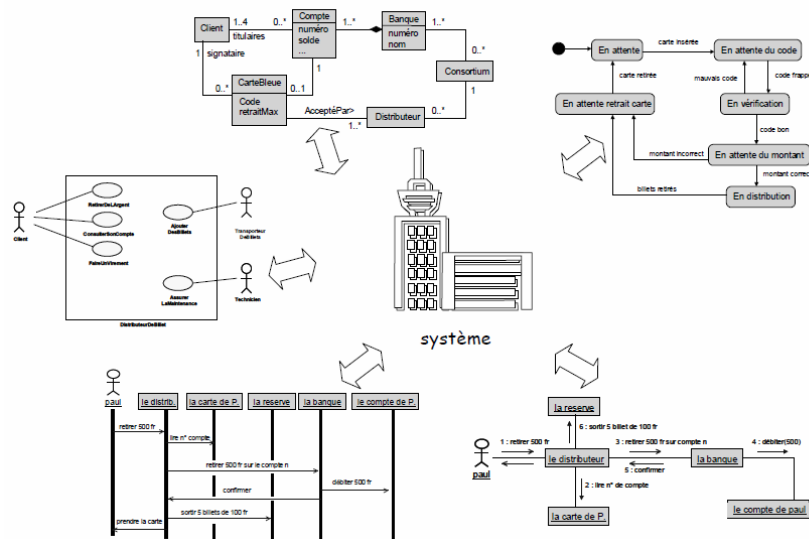
informaticiens

- Le chef de projet
- Les développeurs
 - Les architectes
 - Les codeurs
 - Les testeurs
 - Les intégrateurs
- Les fournisseurs d'outils et de matériel
- Les vendeurs
- Les formateurs

Développement informatique

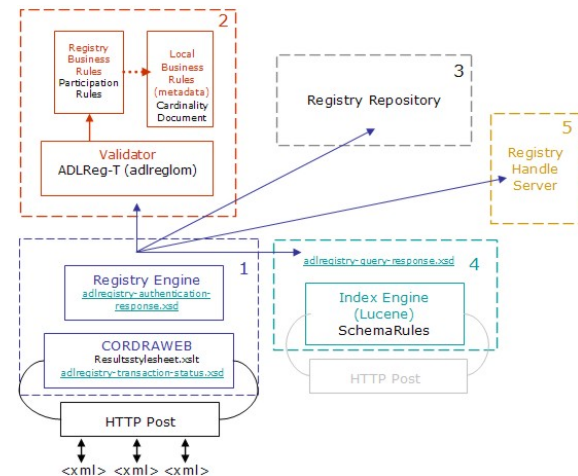
=>

intégration de multiples aspects



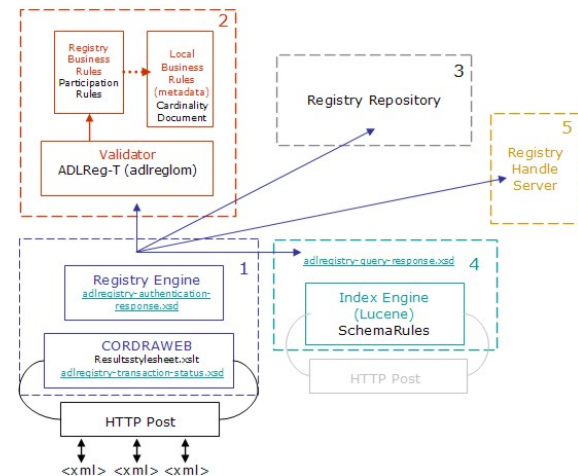
Les divers aspects liés au développement

- Les fonctionnalités
 - Données et algorithmes
- Les environnements de développement
 - IDE
 - Les frameworks
- La structure
 - Classes
 - Modules
 - Paquetages
 - Composants
- La correction des programmes
 - Tests
 - Preuves (sémantique formelle)



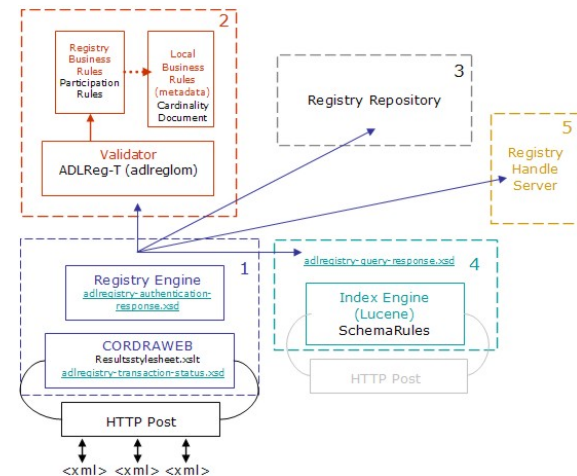
Les divers aspects liés au développement

- Le déploiement des applications
 - Versions et plateformes
- Les machines d'exécution
 - Machines et réseaux
 - OS
 - Middlewares
- La communication
 - Protocoles
- L'utilisation sûre de ressources et services
 - Sécurité
 - Disponibilité
 - Performances et temps-réel
 - Consommation



Les divers aspects liés au développement

- Les outils de développement
 - Édition
 - Génération d'exécutable
 - Mise au point
 - Partage de code
 - Qualité de code
- La méthodologie
 - Cycle de vie
 - Guide de style
 - Design pattern
 - Suivi de versions

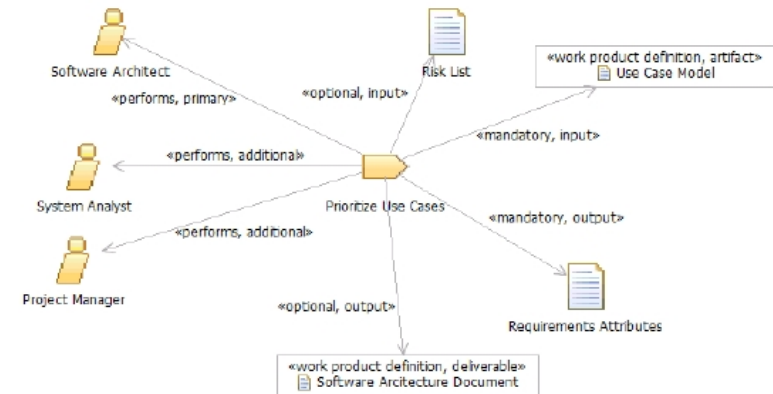


Limites de la compréhension humaine => séparation des préoccupations

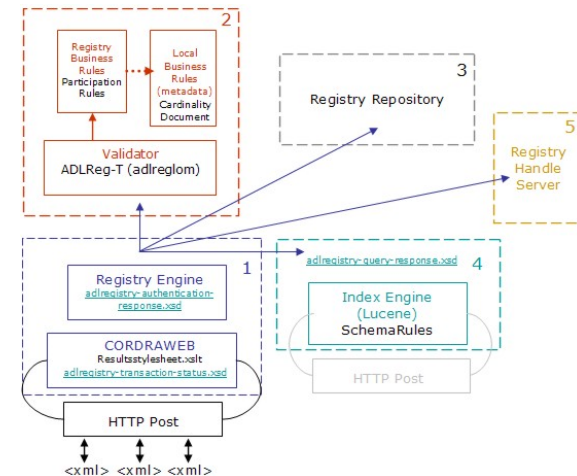
- Analyser un problème complexe
 - Décomposition du problème en sous-problèmes
 - Décomposition objet (UML)
 - Analyse architecturale
 - Modularité et abstraction
 - « Separation of concern »
 - Dijkstra, " On the role of scientific thought" 1974
 - Reade, " Elements of Functional Programming ", 1989

Un aspect du problème

- Une activité qui produit
 - des informations (modèles)



- Pas de document produit : pas d'activité



Des informations produites pour chaque activité

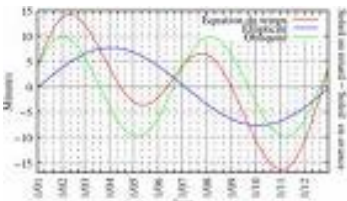
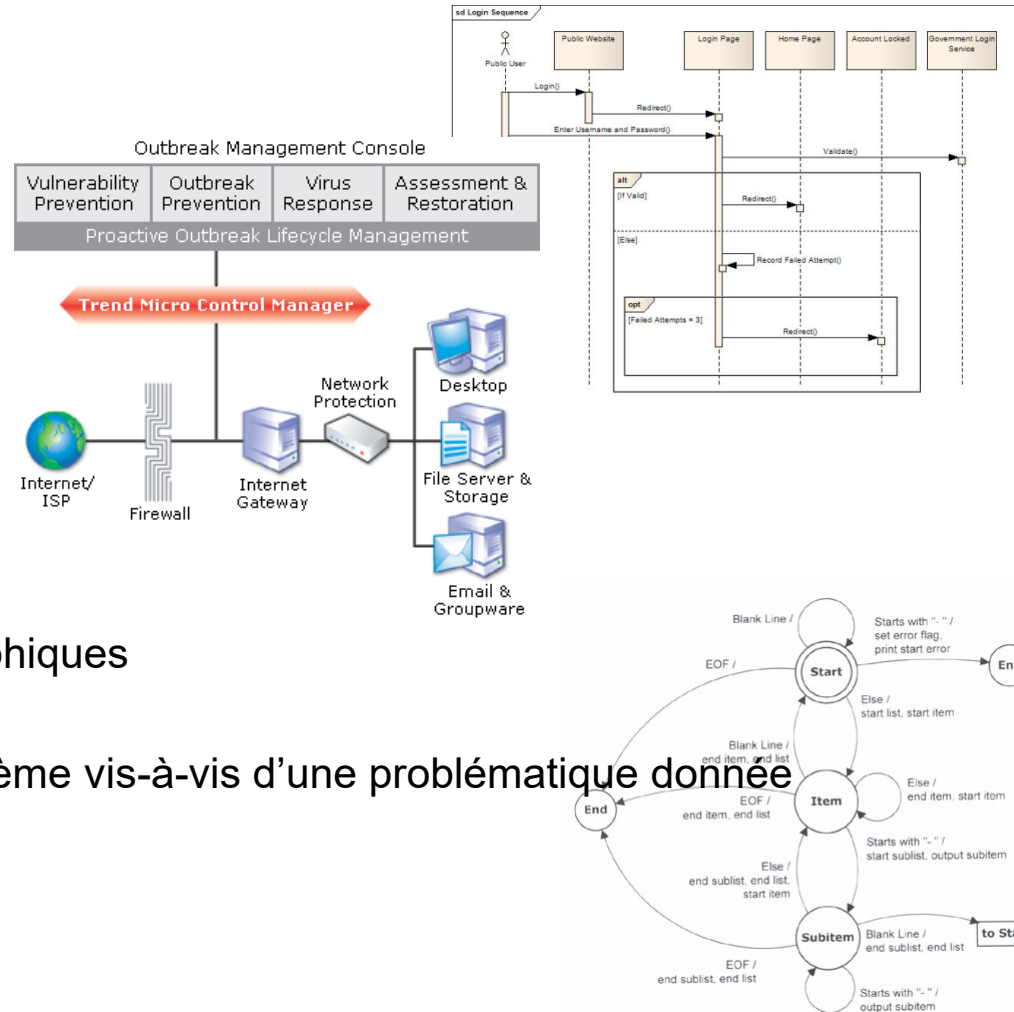
- *Analyse des exigences*
 - Faire un modèle du domaine
 - Décrire les fonctionnalités attendues
 - Lister les contraintes de déploiement
- *Accompagner le développement*
 - Décrire la structure (modules, architecture, BD, ...)
 - Décrire le comportement
 - Editer les plans de tests
 - Expliquer le choix des outils (POC, tests, critères, ...)
 - Formaliser le suivi du développement (Gantt, ...)
- *Assurer le suivi*
 - Documents pour les utilisateurs et les installateurs
 - Formulaire de gestion d'erreurs (maintenance corrective)
 - Formulaire d'évolutions (maintenance évolutive)

Limites de la compréhension humaine => abstraction

- Lao Tseu
 - *"Trop de précision tue"*
- Nombre maximal de « token » dans un schéma
 - De 3 à 7 au maximum, mais plutôt 3
 - Limité par les capacités de compréhension du lecteur
 - Toujours inférieur aux capacités du producteur des modèles concernés
 - Le producteur connaît son modèle, le lecteur le découvre
- **Des modèles simples**
 - Attention : il faut savoir faire « simple mais pas simpliste »

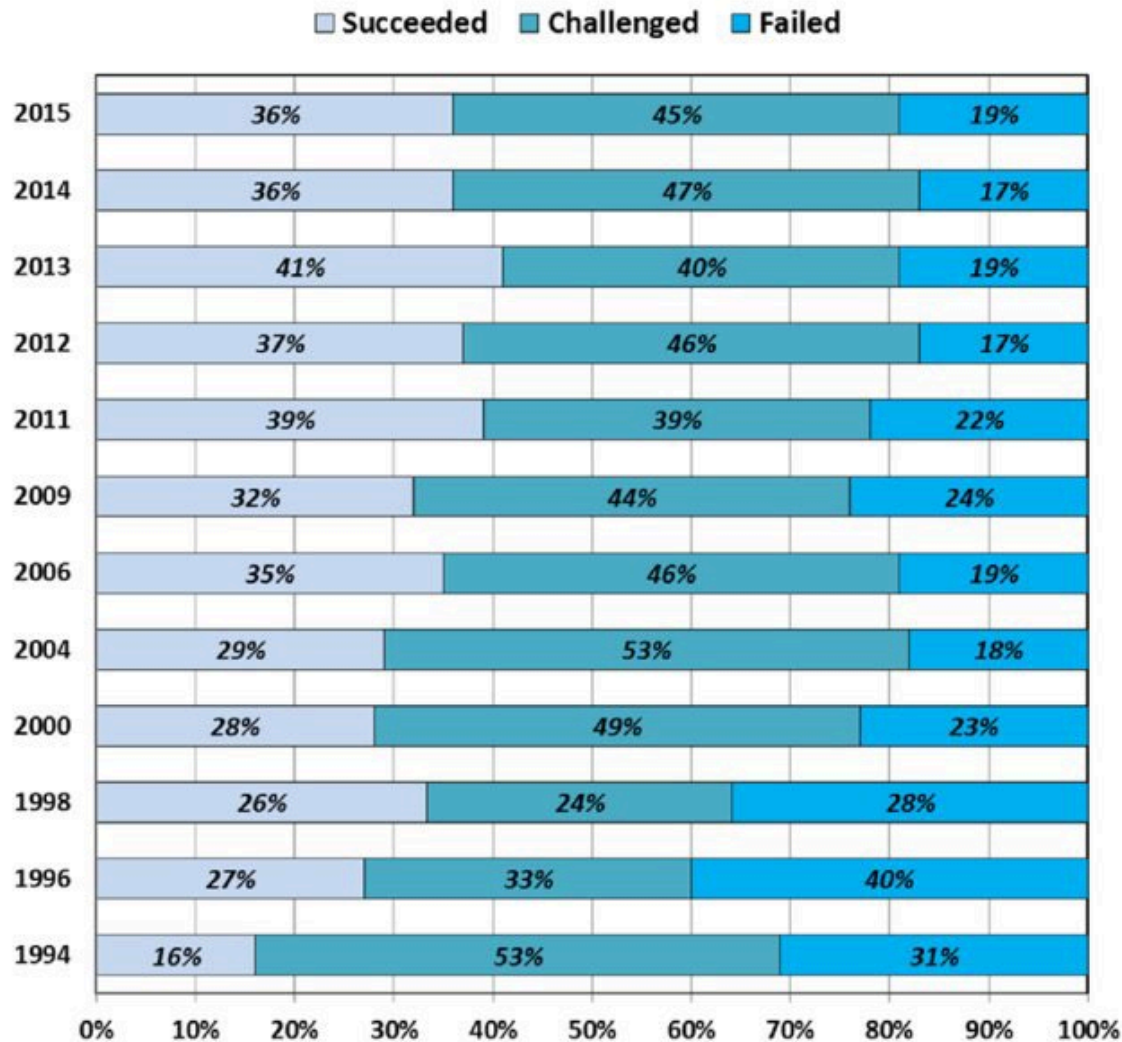
Des modèles pour ...

- Communiquer
- Documenter
 - Rapports, contrats
- Expliquer
 - Présentations
 - Groupes de travail
- Réfléchir
 - Schémas, dessins, graphiques
- Evaluer des propriétés
 - Point de vue sur le système vis-à-vis d'une problématique donnée



- Le développement : une ingénierie complexe
- Nombreuses activités
 - Spécification, documentation, architecture, gestion de projet
- Nombreux intervenants
 - Points de vue différents, spécialités différentes
- Nombreux documents
 - Pour les informaticiens et les non informaticiens
- Qualité de la communication
 - On ne communique pas avec un code
- Abstraction et séparation des préoccupations
- Travail en équipe
 - Organisation et suivi
- Manipuler des modèles (documents) et avoir de la méthode
 - Production et communication autour des modèles
 - Organisation des activités
 - **QQOQCCP** : **Q**ui fait **Q**uoi **Q**uand, **O**ù **C**omment **C**ombien et **P**ourquoi

Le CHAOS report du Standish Group



- *Succeeded* : dans le respect du délai, du budget et des fonctionnalités
- *Challenged* : 1 point n'est pas respecté
- *Failed* : 2 à 3 points ne sont pas respectés

Le *CHAOS* report du Standish Group

- En 2018, le Standish Group, dans son *CHAOS Report*, a évalué à 34% la part des projets qui aboutissent dans les conditions prévues
- 47 % sont en retard ou ont un coût supérieur au budget
 - 42,7 dépassent de 189% leur budget prévisionnel
- 19% sont en échec
 - 31,1 % des projets sont stoppés avant d'être terminés
- En 2020, le Standish Group, dans son *CHAOS Report*, a évalué à 31% la part des projets qui aboutissent dans les conditions prévues
- 50 % sont en retard ou ont un coût supérieur au budget
- 19% sont en échec

Le *CHAOS* report du Standish Group

CHAOS FACTORS OF SUCCESS

FACTORS OF SUCCESS	POINTS	INVESTMENT
Executive Sponsorship	15	15%
Emotional Maturity	15	15%
User Involvement	15	15%
Optimization	15	15%
Skilled Resources	10	10%
Standard Architecture	8	8%
Agile Process	7	7%
Modest Execution	6	6%
Project Management Expertise	5	5%
Clear Business Objectives	4	4%

Aspects humains
(45 %)

Aspects techniques
(33 %)

Aspects processus
(22 %)

Le CHAOS report du Standish Group

CHAOS RESOLUTION BY PROJECT SIZE

	SUCCESSFUL	CHALLENGED	FAILED
Grand	2%	7%	17%
Large	6%	17%	24%
Medium	9%	26%	31%
Moderate	21%	32%	17%
Small	62%	16%	11%
TOTAL	100%	100%	100%

The resolution of all software projects by size from FY2011-2015 within the new CHAOS database.

Quantification des activités

- Répartition des activités
 - Analyse et conception
 - Réalisation et tests unitaires
 - Codage
 - Intégration et validation :

Quantification des activités

- Répartition des activités
 - Analyse et conception 45%
 - Réalisation et tests unitaires : 35%
 - Codage 15 à 20% du total
 - Intégration et validation : 25%
- Dérives dans les estimations
 - Étude préalable : de 10 à 25 %
 - Conception : de 10% à 35 %
 - Réalisation : de 30% à 40%
 - Mise en œuvre : de 5% à 20%
- LOC
 - 10 LOC/j , Fred Brooks « The Mythical Man-Month »
 - 16 to 38 LOC/j , Caper Jonesz
 - 20 to 125 LOC/j (projets 10 k LOC), 1,5 to 25 LOC (projets 10 000 k LOC), MC Connell

- Une définition
 - Ensemble de méthodes, techniques et outils pour la production et la maintenance de composants logiciels de qualité
- Principes
 - Rigueur et formalisation
 - Séparation des préoccupations : découpage d'un problème en sous-problème
 - Modularité, abstraction, généricité, faible couplage
 - Modélisation
- Pourquoi appliquer des principes de génie logiciel
 - Logiciels de plus en plus importants et complexes
 - En terme de fonctionnalités et de propriétés extra-fonctionnelles attendues
 - Intégration de technologies diverses en constante évolution
 - Enjeux stratégiques pour les entreprises et la société

Bilan : les points clés de la construction de logiciel

- Intégrer les besoins des utilisateurs
 - Relation client / fournisseur
 - Considérer l'ensemble des exigences de nature hétérogène
 - prix, délais
 - fonctionnalités
 - matériel
 - IHM
 - performance
 - sécurité
 - formation
 - déploiement
 - maintenance
 - Valider les solutions vis-à-vis des attentes du client

Bilan : les points clés de la construction de logiciel

- Maitriser le lien Informatique et Métier
 - Echanger des informations dans les deux sens
 - Intégrer les concepts et les approches spécifiques métier au logiciel
 - Expliquer l'impact du traitement automatique de l'information au métier
 - Changement d'outil => impact fort sur le système et son utilisation
 - Exemple : intégration d'un simulateur de conduite
 - Impact sur l'apprentissage
 - Impact sur les constructeurs (tests en situation limite)

Bilan : les points clés de la construction de logiciel

- Choisir (ou développer) les bons outils
 - Pour le développement
 - Pour le déploiement
 - Pour la gestion de projet
- Evaluer les solutions
 - Etude de l'existant
 - Solutions faisables
 - Cout, complexité, disponibilité, ...
 - Choix justifiés sur des critères objectifs
 - Proposer des solutions de bonne qualité
 - En suivant un processus maîtrisé et reproductible (normes qualité)
 - En s'appuyant sur des règles métier
 - En augmentant la réutilisation

Les points clés de la construction de logiciel

- Maitriser la gestion du projet informatique
 - Modèles spécifiques au développement logiciel
 - Méthodes spécifiques au développement logiciel
 - Les outils supports du développement logiciel
- Maitriser le développement
 - Etapes
 - Spécification, analyse et conception, codage et tests
 - Organisation du logiciel (architecture)
 - Déploiement
 - Maintenance corrective et évolutive

Les défis

- **Maitriser le lien Informatique et Métier**
 - Echanger des informations dans les deux sens
- **Maitriser le processus de développement**
 - Étude, conception, codage et tests, livraison et suivi
- **Maitriser la communication technique**
 - On ne communique pas avec un code ...

Bibliographie

- OMG et UML
 - <http://www.omg.org/>
 - <http://www.uml.org/>
- Cours de Jean-Marc Jézéquel
 - <http://www.irisa.fr/prive/jezequel/enseignement/>
- Cours de Laurent Audibert
 - <http://laurent-audibert.developpez.com/Cours-UML/html/>
- Cours d'Olivier Caron
 - <http://www2.lifl.fr/~carono/>
- Cours de Cédric Dumoulin
 - <http://www2.lifl.fr/~dumoulin/enseign/>
- Jean-Marie Nicolle
 - « Histoire des méthodes scientifiques », Paris, Bréal 2006