

Quatre décennies de génie logiciel : 1968-2008

Les grandes étapes de l'évolution du génie logiciel, depuis l'émergence du concept en 1968 jusqu'à 2008, ses relations avec d'autres disciplines, notamment l'ingénierie de systèmes, et les tendances actuelles

Véronique Pelletier
Vice-Présidente ADELI

Dans le cadre de nos Rencontres autour d'un verre, Jean-Claude Rault a retracé un historique du Génie Logiciel, le mercredi 6 mai 2009. Depuis 40 ans, il côtoie cette discipline. Jusqu'à ce jour Jean-Claude Rault a consacré l'essentiel de ses activités à la conduite de projets de recherche appliquée, à la veille technologique et au transfert de technologie dans différents contextes d'entreprises et d'organismes de recherche, ceci principalement dans le domaine de l'informatique et plus particulièrement celui du génie logiciel depuis 1969. Aujourd'hui, ingénieur-conseil indépendant, il conserve le poste de rédacteur en chef de la revue trimestrielle Génie Logiciel et organise avec plusieurs organismes partenaires des conférences internationales sur le thème du génie logiciel.

Le conférencier

Jean-Claude Rault est centralien, de la même promotion qu'Alain Coulon (c'est un très bon cru semble-t-il !). Jean-Claude a commencé sa carrière en 1964 au Laboratoire de Recherche Physico-Chimique (RPC) de la compagnie CSF (Compagnie Générale de télégraphie Sans Fil), entreprise active dans les domaines de l'électronique militaire et professionnelle, des composants électroniques, de l'électronique grand public...L'aventure des circuits intégrés démarrait alors en France.

Jean-Claude Rault est parti aux États-Unis en 1967, chez Corning Glass (à Raleigh, en Caroline du Nord), l'équivalent américain de Saint-Gobain. Le lien entre le verre et le silicium...Corning Glass souhaitait alors se lancer dans l'industrie des circuits intégrés. Au cours de son séjour, il eut quelques contacts avec les trois Universités situées dans le Research Triangle Park. L'une d'entre elles, University of North Carolina à Chapel Hill, avait ouvert en 1964 un département d'informatique dirigé par Frederick Brooks, l'auteur du best-seller « Le mythe du mois-homme » et père de l'OS 360 d'IBM, Cette fréquentation a sensibilisé Jean-Claude Rault aux difficultés des développements de logiciels et à l'importance d'une bonne conduite des projets informatiques.

Jean-Claude Rault est revenu chez Thomson CSF (Thomson et CSF venant de fusionner) en 1969 où il est resté une dizaine d'années, dirigeant un service se consacrant au développement d'outils d'aide à la conception du matériel et du logiciel. C'est dans ce cadre qu'il a commencé des travaux sur le génie logiciel, notamment sur le test du logiciel, la fiabilité du logiciel, le prototypage rapide et les composants logiciels.

De 1979 à 1987, il intègre l'Agence De l'Informatique (ADI), dans la Direction Recherche et Développement, dirigée par Robert Mahl, où il est responsable du département Logiciel. Relevant du Ministère de l'Industrie, l'ADI était constituée au démarrage d'anciens de l'IRIA (maintenant INRIA). C'est dans ce cadre qu'il a fondé la revue trimestrielle « Génie Logiciel » (qu'il dirige de 1985 à ce jour) et organisé plusieurs manifestations internationales, notamment les Journées annuelles d'Avignon consacrées aux applications de l'intelligence artificielle et des journées annuelles sur le génie logiciel ayant pour objectif le transfert entre recherche et industrie.

À la fermeture de l'Agence De l'Informatique en 1987, il fonde une entreprise, héritière de ses activités à l'ADI, ayant une triple activité : le conseil, l'édition et l'organisation de conférences internationales traitant de technologies avancées.

Puis, à la fin de l'année 1998, il rejoint le CNAM comme consultant indépendant avec pour mission de développer les relations du pôle Informatique de l'organisme avec le monde industriel. Dans ce contexte, il a été conduit à organiser une série de Journées d'étude traitant de divers chapitres du génie logiciel et de l'ingénierie de systèmes ainsi qu'à poursuivre les Journées Internationales ICSSEA (International Conference on Software & Systems Engineering and their Applications) lancées en 1988 dans un autre contexte.

Depuis 2007, il est consultant indépendant, évaluateur de projets, rédacteur en chef de la revue « Génie Logiciel » et organisateur en collaboration avec plusieurs organismes partenaires de manifestations à caractère technologique.

Concept du génie logiciel

En octobre 1968, l'OTAN a organisé une première conférence, à Partenkirchen en Allemagne, sur l'industrialisation de l'élaboration du logiciel. C'est à cette occasion qu'est forgée l'expression « Software Engineering » pour donner une tournure délibérément industrielle au propos. À cette époque commençaient à apparaître des premiers grands projets logiciels et les affres qui les accompagnent.

Nous estimons qu'un historique du génie logiciel ne peut se faire sans mettre en parallèle l'ingénierie de systèmes. En effet, les deux disciplines sont proches l'une de l'autre, présentant un fort recoupement, voire des points redondants. Ceci est dû à leurs origines différentes, à leur émergence à des époques différentes et, sans aucun doute, aux effets de chapelles en concurrence. Née dans le contexte de grands projets multidisciplinaires à la fin des années 50, l'ingénierie de systèmes est indubitablement la plus ancienne. Le génie logiciel est apparu une décennie plus tard dans le seul milieu de l'informatique.

En conséquence, nous présenterons d'abord l'ingénierie de systèmes.

Ingénierie de systèmes

Apparue à la fin des années 50 dans divers organismes militaires américains, l'ingénierie de systèmes est antérieure de dix ans au génie logiciel. En 1957, est publié le livre fondateur : H. H. Goode et R. E. Machol : *Systems engineering: An introduction to the design of large-scale systems* ; McGraw Hill, 1957.

D'autres ont suivi au cours des années 60 :

- A. D. Hall : *A methodology for systems engineering* ; Van Nostrand-Reinhold, 1962 ;
- H. Chestnut : *Systems engineering tools* ; J. Wiley, 1965 ;
- H. Chestnut : *Systems engineering methods* ; J. Wiley, 1967 ;
- R. E. Machol : *System engineering handbook* ; McGraw Hill, 1965.

Plus tard (1974), le Ministère de la Défense des États-Unis (DoD) publie une première norme : *Systems Engineering Management*.

En 1992, un ouvrage marquant est publié par Andrew P. Sage, de l'Université George Mason : *Systems Engineering*, J. Wiley, 1992.

En France, la discipline n'a émergé que de façon tardive, en particulier avec la création, à la fin des années 80, d'une chaire intitulée « Intégration de Systèmes » au Conservatoire National des Arts et Métiers.

Le professeur titulaire de la chaire, Jean-Pierre Meinadier, est l'auteur des premiers ouvrages en langue française sur le sujet :

- Ingénierie et intégration des systèmes, Lavoisier, 1998
- Le métier d'intégration de systèmes, Lavoisier, 2002

C'est en 1990 qu'est créée aux États-Unis une association professionnelle internationale : INCOSE, (International Council on Systems Engineering).

Sa mission est de faire progresser l'état de l'art et les pratiques de l'ingénierie de systèmes dans l'industrie, l'éducation et les organismes gouvernementaux, basé sur l'interdisciplinarité, les approches mesurables, en produisant des solutions technologiques appropriées au confluent des besoins sociétaux.

Au fil du temps, INCOSE est devenu l'association mondiale sur le sujet avec des chapitres répartis dans le monde entier mais restant néanmoins d'obédience essentiellement américaine. L'association publie une revue trimestrielle (*Journal of Systems Engineering*) et une lettre d'information trimestrielle (INSIGHT). En 2003, INCOSE publie un guide, le G2SEBok (*Guide to Systems Engineering Body of Knowledge*).

En France, s'inspirant de INCOSE, s'est créée en 1999 une association équivalente sous l'impulsion de 13 grandes entreprises françaises (aujourd'hui au nombre de 25), l'AFIS (Association Française d'Ingénierie de Systèmes), association devenue de fait le Chapitre Français de l'INCOSE.

Les entreprises importantes membres de l'AFIS sont, entre autres :

- Aeroconseil ;
- Airbus ;
- Alstom Transport ;
- Alten ;
- Altran ;
- Areva TA ;
- CS SI ;
- Dassault Aviation ;
- DCNS ;
- DGA ;
- EADS DS ;
- EDF ;
- Nexter Systems ;
- RATP ;
- Renault ;
- Thales ;
- Thales Alenia Space.

Génie logiciel

La discipline du génie logiciel peut être datée très facilement, car, comme rappelé plus haut, l'expression a été créée à l'occasion de l'organisation d'une conférence internationale portant sur la difficulté de développer les logiciels de taille importante.

L'expression génie logiciel se voulait être provocatrice (à l'époque) pour donner à entendre que l'élaboration du logiciel devait reposer sur le genre de fondations théoriques et de disciplines pratiques rencontrées habituellement dans les diverses branches de l'ingénierie, comme, par exemple, le génie électrique, le génie maritime, le génie chimique, toutes disciplines d'ingénieurs qualifiées depuis un certain temps de « génie ».

Le génie logiciel est défini comme l'ensemble des méthodes, techniques, outils et processus pour spécifier, concevoir, mettre en œuvre, tester, valider et maintenir les produits logiciels. Il peut être vu au travers des activités de base qu'il implique, comme les suivantes :

- définition des besoins, des buts et objectifs ;
- définition des exigences ;
- analyse des fonctionnalités et des exigences ;
- établissement des spécifications ;
- choix d'architecture/structures ;
- affectation des exigences et spécifications ;
- évaluation des compromis et choix alternatifs ;
- conception de sous-systèmes ;
- analyse de sous-systèmes ;
- conception et développement du logiciel ;
- définition des interfaces ;
- planification ;
- estimation du coût global ;
- mesure des performances techniques ;
- gestion de projet ;
- analyse des risques ;
- assistance logistique (maintenance corrective, maintenance périodique, migration...)
- planification et anticipation d'améliorations potentielles ;
- fiabilité, maintenabilité, disponibilité ;
- analyses spécifiques à prendre en compte selon le contexte et correspondant à des exigences non fonctionnelles :
 - sûreté de fonctionnement,
 - sécurité,
 - contraintes environnementales,
 - compatibilité électromagnétique ;
- intégration ;
- test et évaluation ;
- gestion de configuration ;
- assurance de la qualité ;
- formation ;
- documentation ;
- production ;
- installation ;
- fonctionnement et maintenance ;
- évaluation du fonctionnement et des changements.

Cette liste d'activités est pratiquement la même que celle de l'ingénierie de systèmes. La différence réside essentiellement dans le fait que les outils et méthodes de l'ingénierie de systèmes pour conduire ces activités relèvent de génies particuliers multidisciplinaires alors que, dans le cas du génie logiciel il s'agit d'une seule discipline celle du logiciel.

Historique

Lancé en 1969-1970, soit plus de 10 ans après l'ingénierie de systèmes, le génie logiciel comparé à l'ingénierie de systèmes s'est développé plus rapidement. Ceci en raison de l'impulsion des grands projets spatiaux et militaires, grands consommateurs de logiciels et imposant des exigences de qualité et de sûreté de fonctionnement et du fait que les informaticiens du génie logiciel étaient plus enclins et plus à même de développer des outils logiciels d'aide à la conduite des activités.

Les conférences de l'OTAN ont été immédiatement suivies par de nombreuses autres conférences spécialisées sur tel ou tel thème du génie logiciel, comme par exemple la fiabilité du logiciel, le test du logiciel, la spécification du logiciel, la maintenance du logiciel...

En 1975 s'est tenue la première conférence véritablement dédiée à l'ensemble du génie logiciel, accompagnée d'une exposition de premiers outils, notamment pour l'aide au test. Depuis, l'IEEE tient une rencontre annuelle internationale. Cette première conférence a correspondu à la création, au sein de l'IEEE, d'un comité technique dédié au génie logiciel, animateur d'une série de conférences et ateliers récurrents et éditeur de deux revues dédiées au génie logiciel. Ceci s'est passé quinze ans avant la création de l'INCOSE.

La France, dans le domaine, n'était pas à la traîne, notamment grâce aux financements publics, militaires et civils. Le Ministère de l'Industrie, au travers de l'IRIA, a apporté son concours en favorisant des rencontres entre le milieu industriel et le milieu universitaire. Cela a culminé début 1979 par le lancement d'un premier appel à propositions de projets consacré totalement au génie logiciel ; c'est à l'occasion de cet appel à projets qu'a été introduite en français l'expression génie logiciel pour remplacer *software engineering*, utilisé jusqu'alors. L'expression a fait rapidement florès. Malheureusement, après un bon départ et 8 ans d'effort, ces actions ont été passées à la trappe à la suite de vicissitudes ministérielles intervenues en 1986.

La Commission Européenne a pris dans une certaine mesure la relève et ses programmes ont permis une multitude de projets, souvent avec une forte composante universitaire, dont certains ont conduit à l'éclosion de jeunes pousses spécialisées.

À la fin des années 90, IEEE a lancé un projet au niveau mondial dont le but était d'établir un document de référence décrivant l'essentiel du génie logiciel, c'est-à-dire ce qui devrait être enseigné en la matière ou possédé par les ingénieurs spécialistes du logiciel. Il s'agit du projet SWEBOK (Software Engineering Body of Knowledge). La première version du SWEBOK est parue en novembre 1999.

NB : Cela ressemble étrangement à la publication du guide G2SEBok (Guide to Systems Engineering Body of Knowledge), ou en raccourci SEBOK, publié en 2003 par INCOSE. Dans le même genre, il faut citer aussi le guide PMBOK (Project Management Body of Knowledge) publié par une autre école, celle des spécialistes de la conduite de projets (quel que soit le domaine concerné).

Enfin, il faut citer l'Encyclopedia of Software Engineering, John Wiley, première édition en 1994, deuxième édition avec révision en 2002. C'est un ouvrage riche de 1800 pages

Les occasions manquées en France

En 1979, le Ministère de l'Industrie, intéressé par le concept de génie logiciel correspondant à un besoin identifié, a lancé une action pluriannuelle de financement d'appels d'offres ; ceci dans le but de favoriser l'émergence d'une industrie du génie logiciel en favorisant le passage de prototypes de recherche existants au stade de produits à vocation industrielle. Trois appels d'offres se sont succédé pour arriver à l'appel d'offres de 1982 par lequel était mis au concours la définition d'une plate-forme d'accueil devant permettre d'élaborer des ateliers de génie logiciel propres à divers domaines d'application. En 1983, la Commission Européenne lance un appel d'offres de même finalité et le projet retenu en France en 1982 remporte l'appel européen. Le projet s'intitule alors PCTE (Portable Common Tool Environment).

En parallèle, plusieurs projets de finalité voisine voyaient le jour ; c'était le cas au Ministère de la Défense Américain et chez IBM qui travaillait sur le projet AD/Cycle. Pour diverses raisons, ces deux projets connaissent des vicissitudes et sont arrêtés. Le Ministère de la Défense Américain s'intéresse alors au projet PCTE. Malheureusement, les entreprises partenaires du projet PCTE, au lieu de tirer parti d'une ouverture inattendue et inespérée, se brouillent, leurs liens se distendent, les financeurs se désengagent et le projet PCTE s'enlise.

Bref, la France, bien placée, n'a pas su saisir sa chance.

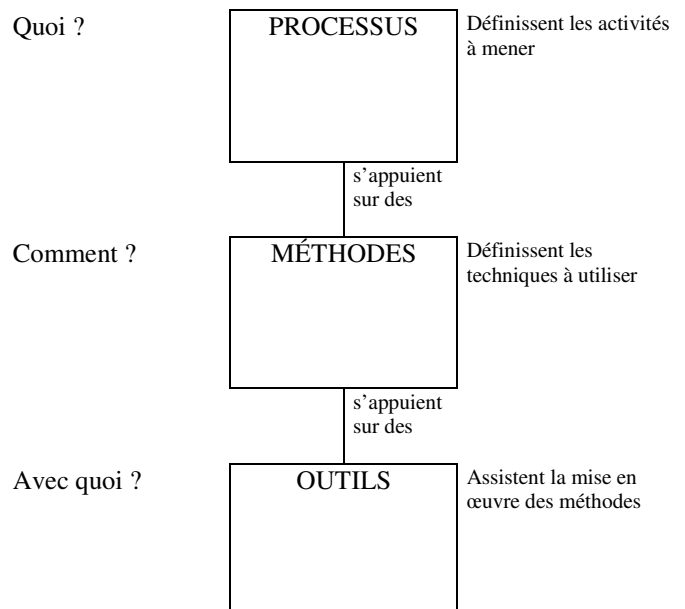
Outils, méthodes et modèles

Il existe plusieurs catalogues d'outils et documents de comparaison entre outils d'une même catégorie. Par exemple, les deux associations INCOSE et AFIS, publient une base de données relative aux outils à vocation commerciale utiles dans les activités de l'ingénierie de systèmes. Mais, il est souvent difficile d'attribuer un outil à telle ou telle discipline, car un

même outil peut servir dans les deux disciplines. Ces outils ont trait, entre autres, aux activités suivantes :

- établissement des exigences et des besoins ;
- spécification des exigences ;
- définition et analyse de l'architecture de systèmes ;
- affectation des exigences aux spécifications ;
- établissements de compromis, choix alternatifs ;
- répartition en sous-systèmes ;
- modélisation et simulation ;
- mesures ;
- gestion des configurations ;
- test ;
- analyse de risques ;
- estimation de coût ;
- suivi des anomalies ;
- support au travail coopératif.

Ces outils interviennent dans les processus de la façon suivante :



On constate qu'une grande partie de ces outils, fréquemment informatisés, sont nés dans le contexte du génie logiciel avant leur utilisation dans le contexte de l'ingénierie de systèmes. Du point de vue des outils, les deux disciplines se recoupent de façon importante.

Parmi les outils connus on peut citer :

- DOORS et RequisitePro (IBM) pour la consignation et la validation des exigences ;
- la notation UML pour la modélisation, notation dont a été dérivée de façon très récente la notation SysML pour l'ingénierie de systèmes ;
- l'outil CLEARCASE pour la gestion de configuration ;
- l'outil STATEMATE, pour la conception, la simulation et le prototypage de systèmes enfouis ;
- l'outil RHAPSODY, un environnement de développement et de test guidé par les modèles, reposant sur UML et SysML, destiné aux systèmes temps réel ou enfouis.

Les modèles

On distingue les trois catégories de modèles suivantes :

Les modèles de maturité : Dans cette rubrique, il faut citer le modèle CMM (Capability Maturity Model) conçu à la fin des années 80 par le Software Engineering Institute (SEI), un organisme financé par le Ministère de la Défense des États-Unis et hébergé

par l'Université Carnegie Mellon, à Pittsburgh. Le SEI a adapté les principes de la qualité des produits proposés par Deming et Joseph Juran. Il en est résulté un cadre conceptuel inspiré par l'ouvrage mondialement connu de Philip Crosby : *Quality is free*. Ce cadre a été adapté au contexte du génie logiciel par un ingénieur d'IBM, Watts Humphrey. Ce dernier l'a ensuite apporté au SEI. Au passage a été introduit le concept de niveau de maturité.

Ainsi, la structure du modèle est la suivante :

Niveau		CMM	CMMI
5	Optimisation	Prévention des défauts Gestion des changements technologiques Gestion des changements de processus	L'entreprise améliore de façon continue ses processus
4	Maîtrisé	Gestion quantitative de processus Gestion de la qualité du logiciel	L'entreprise utilise le contrôle statistique de processus et d'autres techniques quantitatives
3	Défini	Focalisation sur les processus Définition du processus Programme de formation Gestion du logiciel intégrée Ingénierie des produits logiciels Coordination intergroupe Revue par des pairs	Les meilleures pratiques sont répertoriées et introduites dans les processus de l'entreprise. Les produits et processus sont mesurés.
2	Reproductible	Gestion des exigences Planification de projet Suivi et supervision des projets Gestion des sous-contrats Assurance de la qualité Gestion de configuration	Les pratiques de bases de la gestion de projet sont présentes. Mais le comportement de l'entreprise est plutôt réactif.
1	Initial	Processus ad hoc et imprévisibles	

Le modèle CMM a connu dès son lancement un grand succès et a été adopté par de nombreuses entreprises dans le monde. D'autres modèles CMM adaptés à d'autres contextes que celui du seul logiciel sont apparus ; ainsi ont été proposés des modèles CMM pour l'ingénierie de systèmes, pour l'acquisition, pour la gestion d'équipes. Pour éviter une dispersion et des coûts inutiles, le DoD et la National Defense Industrial Association ont lancé en 1998 un projet intitulé CMMI (i pour intégré), dont le but était d'intégrer de multiples disciplines dans un cadre unique et cohérent d'amélioration de processus. Il en est résulté le modèle CMMI dont la première version est sortie en août 2000. Son modèle est indiqué dans la colonne de droite du tableau ci-dessus.

Un autre projet porte sur l'évaluation de la maturité en matière de génie logiciel. Il s'agit du projet SPICE lancé en janvier 1993 par le comité international des standards en génie logiciel ISO/IEC JTC1/SC7. Les travaux du comité ont donné naissance à la norme 15504. Les travaux ultérieurs ont pour but d'étendre

la portée de la norme en y gommant les aspects trop spécifiques du génie logiciel.

C'est une approche similaire à celle du modèle CMMi.

Les modèles de coût : Au début des années 80 sont apparus divers modèles quantitatifs de coût. Il s'agit de modèles mathématiques permettant d'estimer, à partir de la description d'un projet et de celle de son contexte, le coût et le délai de réalisation d'un projet. Deux de ces modèles bien connus sont celui intitulé COCOMO, conçu par Barry Boehm, un des acteurs marquants du génie logiciel, et le modèle PRICE-S issu de GE et RCA.

Les modèles de fiabilité : Très tôt, au début des années 70, diverses approches pratiques ont été proposées pour estimer la fiabilité d'un logiciel, en tenant compte de sa structure et des opérations de test auxquelles ce logiciel a été soumis. Certains de ces modèles sont inspirés des approches relatives au matériel. Un des modèles les plus connus est celui proposé en 1973 par John Musa, des Bell Laboratories. Microsoft s'y est intéressé.

Formation

Pour les deux disciplines, existent diverses possibilités de formation, tant dans les universités que dans les écoles d'ingénieurs. Mais le plus souvent ces formations sont partielles, car elles se concentrent sur les fondements théoriques plus que sur ses aspects pratiques. En France, on peut citer deux exceptions : le Conservatoire National des Arts et Métiers (CNAM) et l'ETGL.

La difficulté de la formation en la matière est de pouvoir reproduire un contexte réel quand il s'agit de projets à grande échelle. Constatant ce problème et tenant compte de l'importance économique du logiciel, la société américaine Wang a lancé au cours des années 80 un institut dédié à la formation au génie logiciel, le Wang Institute of Software Engineering.

Le recrutement des enseignants était ouvert au monde des ingénieurs chevronnés de l'industrie et les émoluments étaient comparables à ceux de l'industrie. L'initiative n'a duré que quelques années.

L'idée a été reprise au début des années 90 par le groupe Thomson-CSF (maintenant Thalès) qui, pour ses propres besoins, a créé l'École des Techniques du Génie Logiciel (ETGL). En 1991, Thomson-CSF a été rejointe par d'autres grandes entreprises (Renault, Alcatel, MBDA, France Telecom, IBM, OSIATIS...) pour former l'AFTI, une association de formation par apprentissage. L'originalité est de former dans des contextes réels, les apprentis étant affectés à des projets réels dans les entreprises d'accueil. Les deux disciplines, génie logiciel et ingénierie de systèmes, sont prises en compte.

On doit citer aussi l'École de Technologie Supérieure, de Montréal, qui présente des points communs avec le CNAM et l'ETGL.

Tendances & Perspectives

Les tendances actuelles impliquent :

- un effort important sur les méthodes et outils destinés à consigner les exigences et à améliorer leur validation. au-delà de l'aide apportée par les outils de traitement de texte ;
- la prise de conscience de l'importance et de la difficulté du test qui se traduit par l'apparition de nouveaux outils de test, par une meilleure formalisation des processus de test et par l'émergence d'une offre de services relative à la professionnalisation des opérations de test et aux tests en tierce partie ;
- l'ingénierie guidée par les modèles ; approche du développement dans laquelle le code source n'est plus considéré comme l'élément central d'un logiciel mais comme un élément découlant de modèles ;
- de nouvelles approches de la réutilisation et de l'architecture des systèmes, comme SOA et SOC ;
- un intérêt, encore modeste mais croissant, pour les méthodes formelles et des tentatives de création d'outils permettant de masquer leur complexité. ▲

**Jean-Claude Rault : genie-logiciel@orange.fr
veronique.pelletier@adeli.org**

Sigles

ADI	Agence De l'Informatique
CSF	Compagnie Générale de Télégraphie Sans Fil
ESA	European Space Agency
ETGL	École des Techniques du Génie Logiciel
GE	General Electric
LCR	Laboratoire Central de Recherche
MDA	Model-Driven Architecture
PCTE	Portable Common Tool Environment
PMBOK	Project Management Body Of Knowledge
PMI	Project Management Institute
RCA	Radio Corporation of America
SWEBOK	Software Engineering Body of Knowledge
UML	Unified Modeling Language