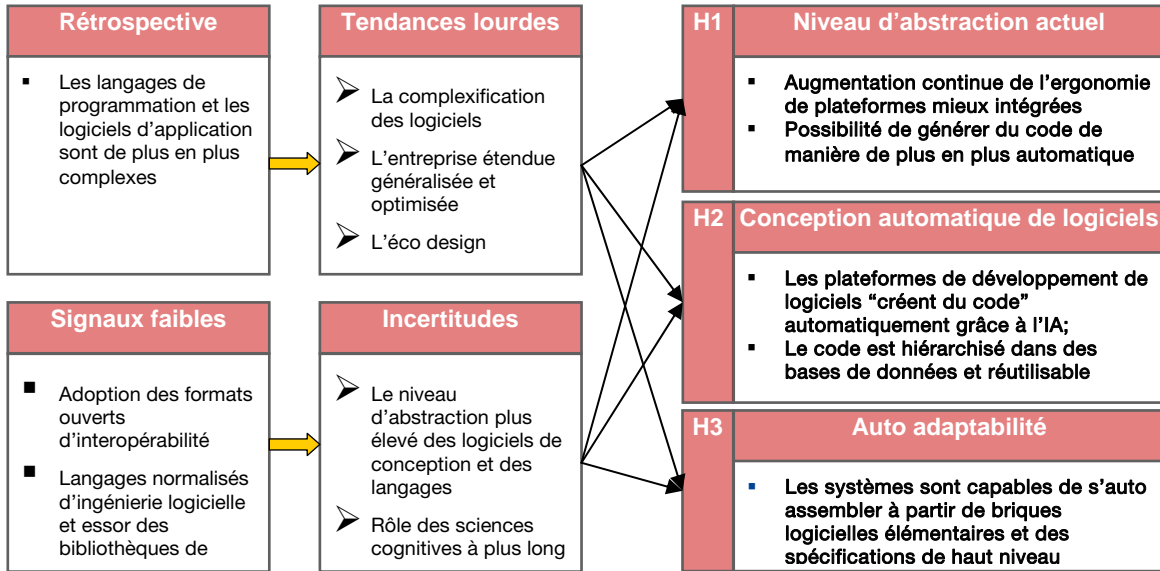


**1.2.1 – Variable Technologies de conception et de design**

*Membre référent pour la Commission à l'Economie numérique : Pierre Faure*

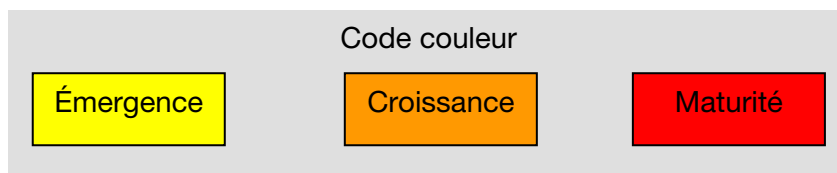


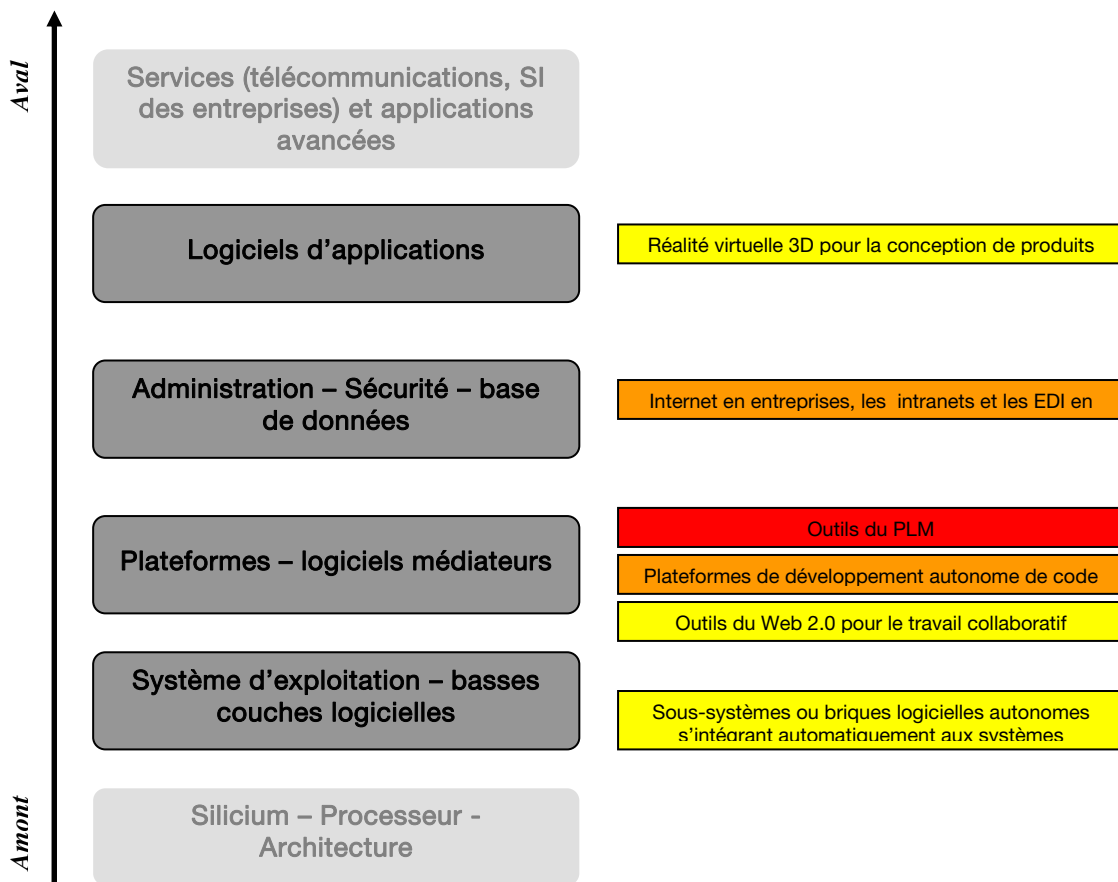
**1 – Définition de la variable**

Les technologies de conception et de design désignent l'ensemble des outils numériques utilisés par les organisations (entreprises, administrations, associations...) ou les personnes afin de créer, concevoir, tester et développer un produit. Il s'agit donc essentiellement de *software*, destiné à faciliter le travail des acteurs économiques, dans le but final d'augmenter leur productivité.

En particulier, dans le secteur des TIC, les outils de conception ont un rôle fondamental dans le cycle de développement des logiciels (ingénierie ou génie logiciel) pour améliorer la productivité des développeurs et la qualité des logiciels produits.

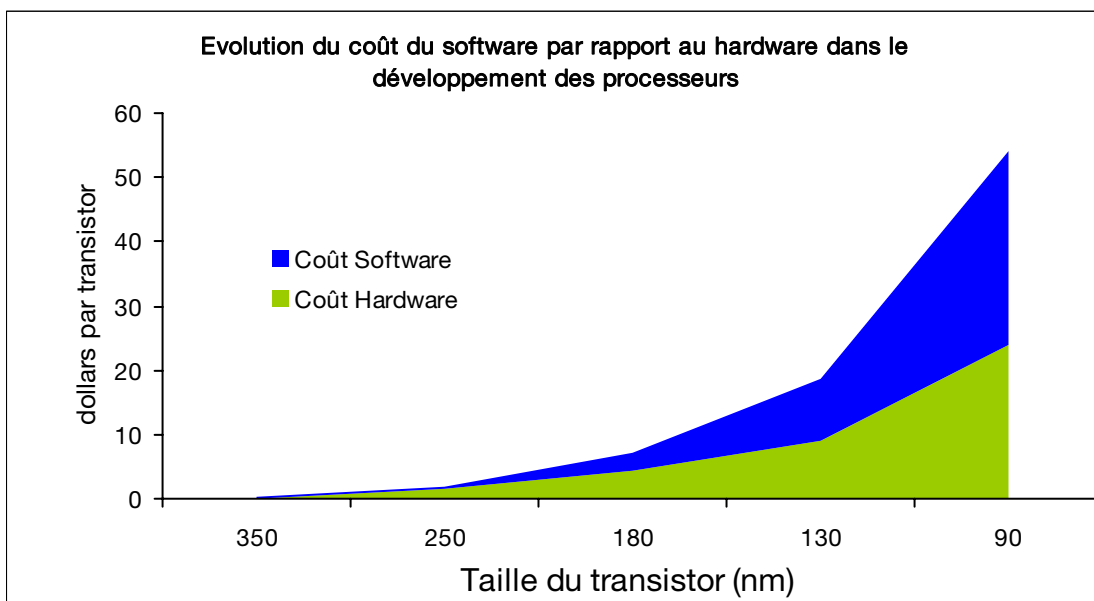
**2 – Position dans la chaîne de la valeur informatique et degré de maturité des technologies**



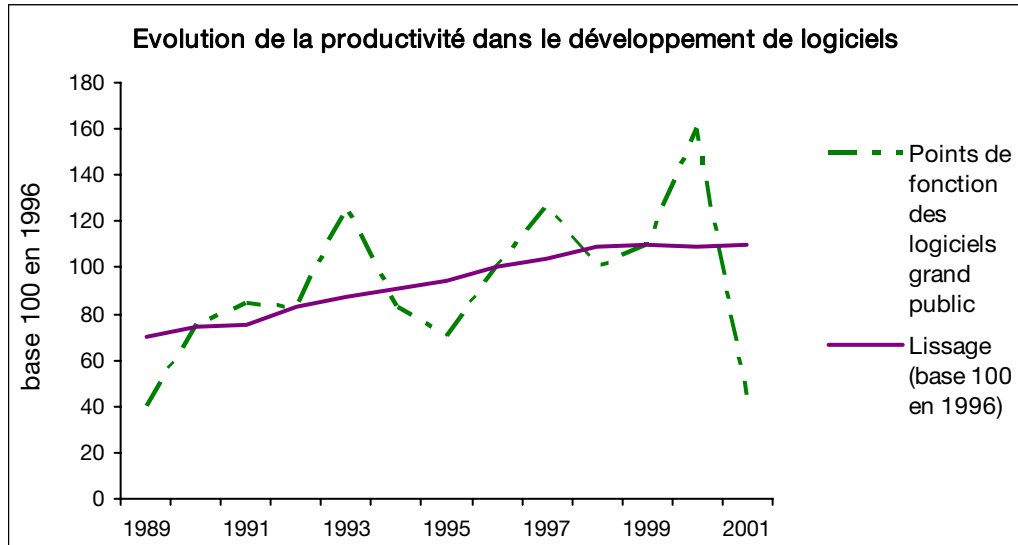


### 3 – Indicateurs pertinents (instruments de mesure pour évaluer la variable dans le temps)

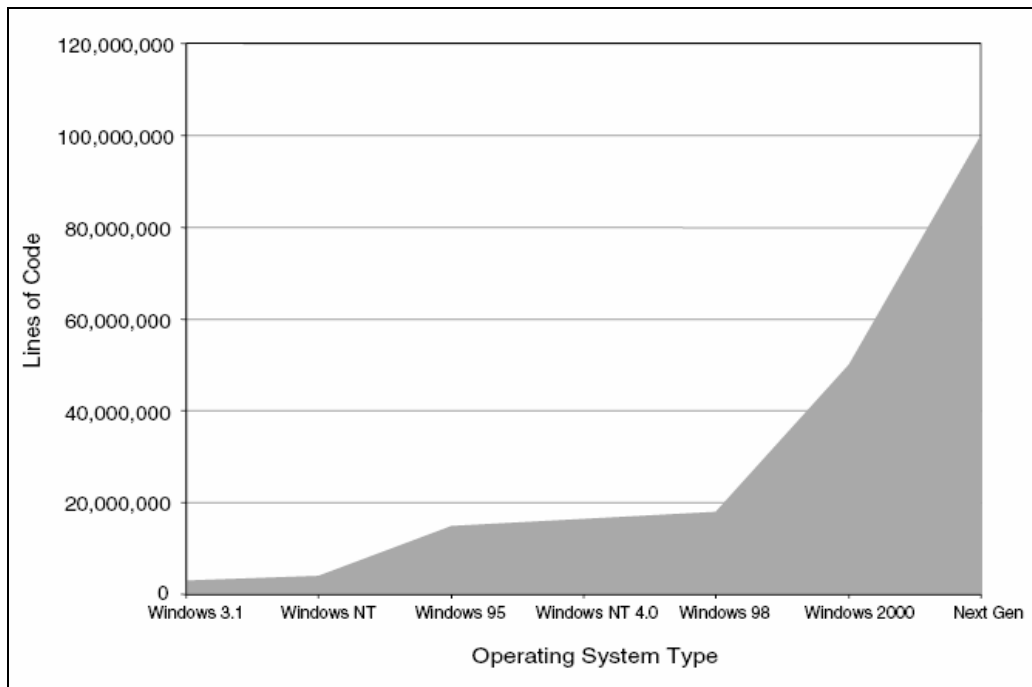
- Le coût de développement des logiciels embarqués (notamment la conception en amont des circuits électroniques).



- Le ratio entre le coût du logiciel et le coût du matériel ou celui entre l'évolution des performances matérielles et la productivité des développeurs<sup>1</sup>.



- Les lignes de code (ou de points de fonction<sup>2</sup>) au cours du temps des logiciels grand public, c'est-à-dire ayant plusieurs versions sur plusieurs années, tels que Windows, Office, Linux, Catia... donnent une indication sur la complexification des logiciels.



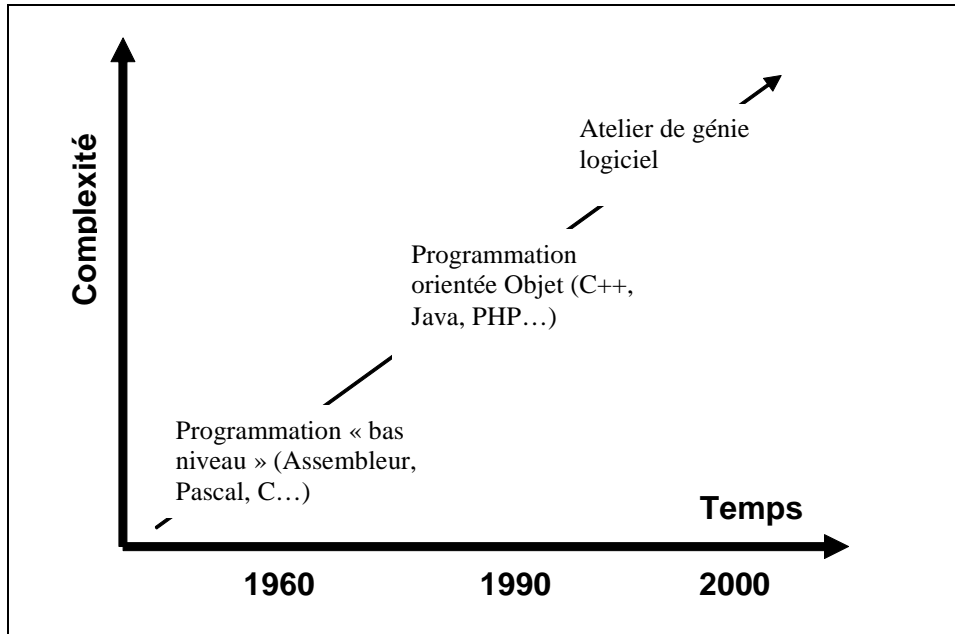
Source: *Software, growth and the future of the U.S. economy*, National Academies of Science, 2007

(1) La productivité des développeurs est souvent le facteur limitant dans les secteurs utilisant de nombreux outils numériques. Elle a augmenté de 5,8 % par an tandis que les performances du matériel augmentait de 58 % par an (correspondant à peu près à la loi de Moore).

(2) La méthode des points de fonction est apparue dans les années 1980 pour tenter de mesurer la productivité dans le développement des logiciels. Elle reste toutefois très empirique et approximative (cf. « *Measuring and sustaining the new economy, Software, Growth and the Future of the U.S Economy*, Dale Jorgenson, National Academies Press, 2006).

#### 4 – Rétrospective (les 20 dernières années ; quoi, comment et par qui ?)

La complexification des outils de conception et de design est visible dans l'évolution des langages de programmation et l'histoire des logiciels. Entre le langage machine des années 1940 et les langages modernes comme JAVA, le niveau d'abstraction des langages de programmation a considérablement augmenté.



Dans le même temps, les applications développées grâce à ces langages<sup>1</sup> ont intégré des fonctions plus nombreuses et plus ergonomiques pour l'utilisateur. L'évolution de Microsoft Word en est un bon exemple. La version initiale (1983) de ce logiciel comportait le saut de ligne, plusieurs polices différentes et des styles différents (italique...); les versions suivantes vont incorporer les fonctions de dessin, d'aide (compagnon office) jusqu'à la prise en charge actuellement des nouveaux formats tels que XML.

#### 5 – La situation actuelle (dynamique en cours et signaux faibles)

Le changement de paradigme entre langages « bas niveau » et langage orienté objet se produit actuellement pour la plupart des logiciels. Cette nouvelle approche de la programmation permet de « sauter » un niveau d'abstraction dans le développement de logiciels, de réutiliser plus aisément les bibliothèques de code existantes...

Par conséquent, le cycle de développement d'un logiciel est raccourci et les logiciels écrits dans ces langages offrent eux-mêmes des niveaux supérieurs d'abstraction tout en améliorant l'ergonomie.

Autre tendance de fond plus « contextuelle » que technologique, l'interopérabilité, sous l'impulsion notamment des logiciels libres, est appelée à modifier les comportements des entreprises développant les logiciels.

---

(1) Par exemple, la suite Microsoft Office est écrite dans le langage Visual Basic datant de 1991. Les prochaines versions d'Office devraient être codées en langage objet (Visual Basic .NET). C'est un changement total de paradigme pour ces suites logicielles très utilisées.

Voici quelques signaux faibles sur le sujet :

- Le format d'interopérabilité XML pour le Web est déjà à l'origine de nouvelles applications (pratique du *mash-up*, cf. fiche Technologies cognitives).
- Les ateliers d'ingénierie logicielle se développent depuis quelques années ; ils associent la gestion des projets de programmation (allocation des ressources en développeurs, tenue des délais, réalisation des objectifs...), l'aide à l'écriture de code (tenue de base de données de bibliothèques, tests et vérification, compilation...) mais également la génération de code à partir d'un cahier des charges. Ils sont le fruit de l'adoption de méta langages tels que UML (*Unified Model Language*) et devraient probablement être le premier pas vers la génération automatique de code.

## 6 – Les principaux acteurs concernés

	France	Europe	Reste du Monde
<b>Recherche fondamentale et appliquée</b>	CNRS-LAAS ; INRIA		Software Engineering Institute (Carnegie Mellon University)
<b>Industrie</b>	Dassault Systems ; Ilog ; Business Objects ; de nombreuses SSII	SAP	IBM ; Borland ; Sun ; Microsoft ; Oracle
<b>PP, instances de régulation, de normalisation et de concertation, etc.</b>	ASSEMI/CNAM ; Images et réseaux (Bretagne), System@tic (Ile-de-France), Aerospace Valley (Midi-Pyrénées)	Commission Européenne	

## 7 – Les usages associés (enjeux et impacts en termes d'applications pour les e-activités)

Les technologies numériques de design et de conception sont déjà au cœur des processus de création dans de nombreux secteurs « d'ingénierie » (automobile, high tech, aéronautique...). Elles commencent à se diffuser dans d'autres secteurs pour lesquels la technologie doit rester plus ou moins « transparente » (urbanisme et architecture, industrie du luxe, publicité, industrie du multimédia...) ; elles seront aussi utilisables par les particuliers, capables dès lors de créer leurs propres objets sur-mesure :

- Les modeleurs 3D : architecture, décoration, habillement...
- Les suites de logiciel de CAO dans l'industrie
- Les logiciels de traitement multimédia et de création artistique, etc.

Pour les entreprises, les XPM et XRM<sup>1</sup> seront rapidement une nécessité vitale pour leur compétitivité (et de l'ensemble de la filière à laquelle elle se rattache), avant de devenir un facteur de différenciation par l'innovation, notamment pour les fonctions de conception (groupware, Knowledge Management, datamining...).

Du côté des loisirs, l'industrie du jeu vidéo - dont le chiffre d'affaires dépasse déjà celui de l'industrie du disque - devrait se diversifier et se renforcer à travers les *serious games*, alternatives intéressantes à la formation continue traditionnelle. Les interfaces homme machine (IHM) pourront être les écrans actuels ou bien des dispositifs de réalité virtuelle (ou augmenté) en trois dimensions.

Pour les applications des outils de conception et de design dans le secteur des TIC, il faut noter le rôle fondamental des techniques de l'ingénierie des logiciels, tant dans la conception efficace de programmes fiables que de circuits électroniques ou d'architectures informatiques.

---

(1) Ces acronymes regroupent des termes tels que Business Process Management, Customer Relationship Management, Supply Chain Management, Supplier Relationship Management...

## 8 – Prospective (les 20 prochaines années)

(Les tendances lourdes et les incertitudes majeures, la différence entre les hypothèses étant construites sur les incertitudes)

- Les tendances lourdes :
  - ⇒ une complexification des outils numériques de conception et de design, avec d'un côté le besoin d'applications très pointues, utilisables uniquement par des spécialistes (CAO, logiciels de modélisation et de calcul...) et de l'autre côté, des outils de conception « grand public » puissants mais « transparents » (modeleurs 3D ergonomiques...). Ces outils s'appuieraient par exemple sur des IHM intuitives et des moteurs d'intelligence artificielle en appui à l'utilisateur ;
  - ⇒ l'importance grandissante de l'éco design ;
  - ⇒ l'inertie représentée par la « *legacy* » et l'empilement des systèmes dans le temps ;
  - ⇒ l'intégration des outils de conception pure (CAO, calcul...) au sein de nouvelles plateformes d'entreprises va se poursuivre, de même que l'intégration des systèmes entre entreprises. Ainsi « l'entreprise étendue » va devenir nécessaire pour les plus petites PME ;
  - ⇒ pour garder le rythme des innovations, le secteur des TIC lui-même – et en particulier la micro/nanoélectronique – devra optimiser autant l'architecture des systèmes que les briques élémentaires qui les constituent. Pour cela, les outils d'aide à la conception seront fondamentaux.
- Les incertitudes :
  - ⇒ un niveau d'abstraction plus élevé des logiciels de conception (cf. fiche Technologies cognitives) se traduisant par la génération de code à partir de spécifications de haut niveau. A quand des ateliers de génie logiciel version 2.0 ?
  - ⇒ la convergence entre programmation graphique (exemple du logiciel Labview...), interfaces intuitives (écran tactile, interaction avec un hologramme ou un écran 3D, reconnaissance...) et des technologies cognitives rudimentaires (tests en temps réel des programmes graphiques) peut apporter des gains de productivité importants dans le développement de certaines applications (robotique, systèmes complexes...);
  - ⇒ à plus long terme, les technologies cognitives pourraient apporter une plus grande rupture par le développement de briques logicielles s'auto intégrant au sein de systèmes complexes. Ces briques logicielles seraient également capables d'évoluer - par exemple en modifiant leur propre code - sans intervention humaine, en fonction des besoins.

### **Hypothèse 1 : Niveau d'abstraction actuel**

- les fonctionnalités des outils de conception et de design restent peu ou prou les mêmes qu'actuellement ;
- l'ergonomie des outils de conception augmente au rythme actuel. Ils sont capables de communiquer avec des applications différentes (usage de formats ouverts tels que XML...);
- les plateformes de développement commencent à prendre en charge de la génération automatique de code.

### **Hypothèse 2 : La généralisation de la conception automatique de logiciels**

- les plateformes de développement de logiciel génèrent du code évolué à partir de 2015-2020, grâce à des algorithmes d'intelligence artificielle ou l'exploitation « intelligente » des bases de données (bibliothèques) de code. La productivité du secteur du logiciel augmente fortement et influence tous les secteurs économiques ;
- les plateformes sont plus fiables, interopérables et indépendantes ;
- l'entreprise étendue, plus efficace, s'est imposée chez la plupart des industriels.

### **Hypothèse 3 : L'auto adaptabilité**

- les systèmes sont capables de s'auto assembler à partir de briques logicielles élémentaires et des spécifications de haut niveau.

*Rédacteur : Jean-Loup Loyer*

*Mise à jour : 08/10/2008*