



**HAL**  
open science

## Vers de nouveaux processus de développement des projets : les recherches sur la contingence publiées dans la littérature américaine

Chantal Morley

### ► To cite this version:

Chantal Morley. Vers de nouveaux processus de développement des projets : les recherches sur la contingence publiées dans la littérature américaine. AFCET-CERAM 1991 : Association Française pour la Cybernétique Économique et Technique, Congrès "Autour et alentour de Merise", Apr 1991, Sophia-Antipolis, France. hal-02420657

**HAL Id: hal-02420657**

**<https://hal.science/hal-02420657v1>**

Submitted on 22 Apr 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**AFCET**

Congrès "Autour et alentour de Merise"

Sophia-Antipolis, 17-19 avril 1991

<b>VERS DE NOUVEAUX PROCESSUS DE DEVELOPPEMENT DES PROJETS</b>
--

<b>Les recherches sur la contingence publiées dans la littérature américaine</b>
--

**Chantal Morley**

**Compagnie Générale d'Informatique**

**Membre du groupe d'étude EUROMETHOD, phase 2**

**Résumé**

La persistance des difficultés que l'on rencontre dans le développement d'un système d'information conduit à s'interroger sur la pertinence d'une approche unique de développement, ainsi que sur un modèle de mise en pratique d'une méthode qui soit indépendant de ses conditions d'application.

Une ligne d'évolution de Merise, confirmée par les acteurs du projet Euromethod, est de dresser une synthèse des facteurs de contingence permettant de définir des approches adaptées aux multiples contextes de conduite des projets.

L'objet de cet article est de présenter les recherches publiées dans la littérature américaine sur les facteurs de contingence qui peuvent guider le choix d'une approche. Ces recherches sont peu nombreuses, mais marquent une tendance à appréhender l'activité de développement non comme un cas particulier de résolution de problème (problem-solving), mais comme un problème de pilotage de projet.

**Mots-clés:**

Système d'information; méthode de développement; conduite de projet; facteurs de contingence; risques; objectifs.

**1. Introduction**

Les problèmes rencontrés dans le développement d'un système d'information ont fait l'objet d'une littérature abondante depuis près de vingt ans. Selon une étude menée aux Etats-Unis il y a quelques années, trois quarts des développements entrepris n'arrivent pas à leur terme ou bien ne sont pas utilisés [Gladden,1982], sans que les causes de ces échecs en soient toujours bien comprises [Tait,1988]. La persistance de difficultés non maîtrisées contraste avec l'abondance de méthodes visant à guider le processus de développement. On a cependant remarqué que, bien qu'utilisant la même méthode, certaines entreprises se déclarent satisfaites et d'autres non, certains projets aboutissent, d'autres non. Pauvreté des méthodes ou mauvaise utilisation? Quelques chercheurs, peu nombreux toutefois, ont été conduits à rechercher **les relations entre une méthode pratique et son contexte d'utilisation**, persuadés que le choix d'une démarche, de principes d'organisation, de techniques, doit prendre en considération certaines caractéristiques-clés du contexte.

Cette nouvelle approche de développement peut être qualifiée d'**approche contingente** (par opposition à une approche basée sur un modèle unique). Le problème est alors de déterminer d'une part les facteurs qui jouent un rôle-clé ( les **facteurs de contingence**), et d'autre part les approches adaptées aux multiples contextes, caractérisés par des valeurs différentes des facteurs de contingence.

Nous allons d'abord rappeler la signification du concept de contingence en Sociologie des Organisations. Puis nous verrons quelle forme il prend en méthodologie de développement. Nous passerons alors en revue les différentes recherches, classées selon le facteur de contingence privilégié.

## 2. La contingence en Sociologie des Organisations

A partir de la fin des années 50, un courant de pensée s'est développé autour de l'idée que les contraintes qui pèsent sur une Organisation contribuent à lui donner sa forme. Ces contraintes, qui peuvent être externes (concurrence, innovation...) ou internes (technologie de production, taille de l'Organisation...), influent sur la structure de l'Organisation.

Ce courant a été baptisé "théorie de la contingence structurelle". Les chercheurs qui l'ont constitué (J.Woodward, T.Burns et J.M.Stalker, P.R.Lawrence et J.W.Lorsch...) ont étudié différents facteurs de contingence en se basant sur des études empiriques. Leurs recherches sont guidées par les affirmations implicites suivantes:

- Il n'y a pas une meilleure façon de structurer une Organisation.
- Toutes les structures, pour une Organisation donnée, à un moment donné, ne sont pas également efficaces.
- Pour favoriser la performance de l'Organisation, la structure doit être adaptée aux valeurs des facteurs de contingence.

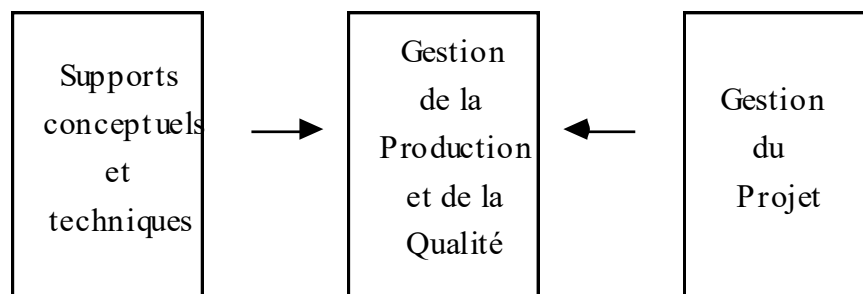
La théorie de la contingence structurelle, qui est peut-être moins une théorie qu'une orientation d'étude [Schoohoven,1981; Drazin,1985], a permis de poser clairement la question de l'ajustement d'une Organisation à ses contraintes. Les deux concepts-clés en sont l'**adéquation** ("fit") et la **performance**. Nous allons

voir quelle signification ils prennent quand on les transpose aux méthodes de développement.

### 3. La nature d'une méthode

L'idée de méthode est basée sur la possibilité d'identifier des classes de problèmes qui appellent des réponses analogues. L'expertise et l'expérience peuvent ainsi être formalisées et réutilisées pour une variété de problèmes qui appartiennent à la même catégorie.

Une méthode apporte des aides pour accomplir les différentes activités nécessaires dans le développement d'un système d'information. Elle contient un ensemble de composants de diverses natures, qui s'architecturent de la façon schématique suivante:



*Figure 1 : Composants d'une méthode de développement de système d'information*

Le composant central est celui qui rassemble toutes les activités de production et les dispositions d'assurance qualité associées (facteurs et critères qualité, procédures...). C'est là que l'on trouve les modèles de développement préconisés, entendus comme des découpages du processus de développement en grandes étapes, ainsi que les activités composant ces étapes, éventuellement regroupées en phases, et la fourniture devant être produite à chaque étape, chaque phase, chaque activité. Les activités peuvent faire référence à des éléments de Supports.

Les éléments du composant Supports conceptuels et techniques sont les concepts proposés par la méthode pour appréhender le développement d'un système d'information, ainsi que les différentes techniques utilisables pour accomplir les activités de production.

Le composant Gestion de projet rassemble les recommandations qui peuvent être faites sur les aspects connexes à la production, selon trois centres d'intérêt:

- Gestion de la ressource humaine (organisation d'équipe, répartition du travail, suivi d'avancement...)
- Planification (estimation des charges, établissement d'un plan de travail...)
- Communication (coordination dans le groupe, relations avec les utilisateurs...).

Les éléments ci-dessus sont alimentés de façon inégale par les méthodes.

Le composant Gestion de projet est en général très peu fourni; les deux autres reçoivent la plus grande partie du corps méthodologique, bien que la gestion de la qualité n'y soit pas toujours intégrée.

En considérant l'ensemble de ses composants, on voit apparaître la **double nature** d'une méthode de développement de système d'information.

D'une part, c'est un **instrument individuel**, pour l'informaticien, le concepteur, le chef de projet., quand confronté à une situation nouvelle ou complexe, il cherche des réponses, basées sur des expériences analogues. Dans ce sens, une méthode est une aide à la résolution d'un problème.

D'autre part, c'est un **instrument collectif**, aidant à organiser le travail d'un service informatique ou d'un projet. Elle lui donne un cadre, une homogénéité et une communauté de culture. Elle guide le travail en commun des différents acteurs.

#### 4. Méthodes et théorie de la contingence

Examinons maintenant la signification que peuvent prendre les deux concepts-clés de la théorie de la contingence structurelle, quand on les transpose aux méthodes de développement.

La **performance** d'une méthode est évaluée en considérant les résultats auxquels on arrive en utilisant la méthode. Ceci peut être interprété de deux façons.

\* Soit on se réfère à l'efficacité de la méthode dans le processus de développement, c'est-à-dire à sa capacité à faire aboutir le projet. On y adjoint parfois la notion d'efficience de la méthode utilisée, mesurée par les ressources consommées (budget, délai).

\* Soit on se réfère à la pertinence du système d'information développé, c'est-à-dire sa capacité à améliorer le fonctionnement de l'Organisation. L'évaluation de cette performance a suscité de nombreux débats [Kim,1988]. Le plus souvent, on considère que le nouveau système d'information devrait augmenter la productivité du domaine couvert et/ou la capacité de décision des gestionnaires. La mesure de ces améliorations est cependant difficile à obtenir. Beaucoup de chercheurs considèrent qu'un bon indicateur de substitution est la satisfaction de l'utilisateur [Ives,1983].

L'**adéquation** signifie que le choix et/ou la mise en oeuvre d'une méthode doivent être adaptés aux contraintes (les facteurs de contingence). Il s'agit donc:

- soit de choisir entre plusieurs solutions (méthodes ou éléments de méthode);
- soit de choisir entre plusieurs façons de mettre en oeuvre une méthode (par exemple degré et modalités de participation utilisateur).

Les **recherches** qui ont cherché à mettre en évidence des démarches concurrentes et à identifier ce qui guide (ou doit guider) le choix, peuvent être classées en trois groupes.

1°) Dans le premier, qui ne comprend qu'une seule recherche, l'hypothèse de recherche est très proche de la théorie de la contingence structurelle. Le facteur de contingence est la nature des tâches des informaticiens: il s'inspire du facteur proposé par J.Woodward, le type de technologie [Woodward,1965]. L'adéquation porte sur le mode de coordination des informaticiens, c'est-à-dire le fonctionnement de la structure. La performance visée par l'adéquation est mesurée à l'aune de la satisfaction des utilisateurs. Ces derniers sont implicitement les gestionnaires.

2°) Dans le deuxième groupe, les chercheurs ont assimilé performance et aboutissement du projet. Ils se sont attachés à étudier le poids de l'incertitude et des risques du projet. Précisons ces deux concepts.

S'il n'y a pas d'incertitude sur un projet, c'est que l'on est sûr qu'il sera mené à terme. Dire qu'il y a incertitude, c'est pressentir que des obstacles peuvent surgir, qui compromettraient le développement du système d'information.

Les risques représentent ces obstacles potentiels. Ce peut être des événements<sup>1</sup> ou des erreurs (sur les choix de gestion ou de conception) dues à la complexité du projet (c'est-à-dire à l'impossibilité d'avoir immédiatement une vision claire de la réalité sur, dans et avec laquelle on travaille).

Nous avons relevé quatre recherches dans ce groupe, dont les facteurs de contingence s'expriment ainsi:

- incertitude du projet
- profil de risque du projet
- complexité et contraintes du projet
- types de risques du projet.

L'adéquation à apporter suivant le facteur de contingence porte sur l'un ou l'autre composant selon les auteurs.

3°) Les chercheurs du troisième groupe se sont attachés à décliner la notion de performance du système d'information développé. Selon la perception du rôle et de la nature de ce dernier, le même système d'information sera considéré comme ayant ou non amélioré le fonctionnement de l'Organisation.

La performance est ainsi mesurée par le degré de pertinence du système d'information: il s'agit d'évaluer s'il constitue une réponse appropriée à ce qu'on en attendait. Ces recherches ne débouchent pas sur l'expression de "lois de contingence", mais sur une typologie d'objectifs (souvent implicites) que l'on peut assigner à un système d'information, et qui induisent des choix et des mises en oeuvre de méthode.

Ces trois groupes de recherches sont schématisés dans le tableau ci-dessous.

---

<sup>1</sup>Par exemple, changement imprévu de réglementation, qui obligerait à revoir l'ensemble de la conception déjà effectuée; ou bien l'éclatement d'un conflit entre utilisateurs, qui empêcherait de trouver un consensus.

Facteurs contingence	Adéquation sur les composants			Evaluation performance
	Supports Conceptuels et Techniques	Gestion de la Production et de la Qualité	Gestion du projet	
Technologie tâches			Mode de coordination	Satisfaction utilisateurs
Incertitude/Risque	Méthode détermination besoins	Modèle de développement Choix des activités	Pilotage du projet Choix des ressources Evaluation délai Participation utilisateurs	Efficacité et efficacité du processus de développement
Objectifs		Modèle de développement	Participation utilisateurs	Pertinence système information

Figure 2: Rôle des facteurs de contingence sur les composants d'une méthode

Nous allons examiner successivement ces différentes recherches.

## 5. Nature des tâches des informaticiens

K.Kyu Kim [Kyu Kim,1988] a cherché à appliquer à **l'organisation du travail des informaticiens**, la théorie liant les caractéristiques technologiques des tâches et le mode de coordination des activités. Il a pris comme hypothèse que si une équipe de développement s'est dotée de la forme de coordination appropriée à ce qu'il nomme son "contexte organisationnel", cela améliore sa performance.

La méthode est donc ici envisagée comme un mode de coordination (personnelle ou impersonnelle) entre acteurs. Si l'on se réfère à une classification largement acceptée [Mintzberg,1984], une méthode peut être considérée comme relevant de l'un des trois types de coordination impersonnelle suivants:

- standardisation des procédés, de par la description des activités à accomplir;
- standardisation des résultats, de par la description des fournitures à produire;
- standardisation des qualifications, à travers la formation à des concepts et à une démarche.

Elle peut aussi aider à la coordination personnelle, en fournissant notamment un vocabulaire de référence et des éléments de suivi de projet.

Le contexte est défini par le type de technologie des tâches et la taille du groupe de travail. La typologie des technologies est élaborée à partir de trois variables. Les deux premières sont dues à C.B.Perrow [Perrow 1970]:

- le degré de *prévisibilité* de la tâche, qui indique le nombre de cas exceptionnels pouvant se produire ou bien la fréquence d'apparition d'évènements nouveaux ou inattendus devant être pris en compte par la tâche;
- le degré d'*analyse requis*, qui indique la possibilité ou l'impossibilité de décrire à l'avance la façon de traiter les problèmes.

La troisième variable est l'*interdépendance* des tâches qui indique dans quelle mesure le personnel d'une entité dépend d'une autre pour accomplir son travail. Les catégories d'interdépendance sont, par ordre croissant de dépendance: le "pool" (A et B travaillent de façon indépendante), la "séquence" (B travaille quand A a fini sa tâche), la réciprocité (A et B doivent se fournir mutuellement des éléments), l'équipe (A et B travaillent ensemble).

Les observations ont été faites au travers de questionnaires adressés aux informaticiens pour mesurer la forme de coordination et le contexte organisationnel, et aux utilisateurs pour mesurer la performance via leur satisfaction. Les hypothèses sont les suivantes:

- H1: Plus la tâche est prévisible, plus une coordination impersonnelle améliore la satisfaction de l'utilisateur.
- H2: Moins la tâche est prévisible, plus une coordination personnelle améliore la satisfaction de l'utilisateur.
- H3: Plus la tâche est analysable, plus une coordination impersonnelle améliore la satisfaction de l'utilisateur.
- H4: Moins la tâche est analysable, plus une coordination personnelle améliore la satisfaction de l'utilisateur.
- H5: Moins les tâches sont interdépendantes, plus une coordination impersonnelle améliore la satisfaction de l'utilisateur.
- H6: Plus les tâches sont interdépendantes, plus une coordination personnelle améliore la satisfaction de l'utilisateur.
- H7: Plus grande est la taille du groupe, plus une coordination impersonnelle améliore la satisfaction de l'utilisateur.

Les résultats de l'étude statistique menée sur 28 centres informatiques attachés à des hôpitaux ne permettent de vérifier que les hypothèses H1 et H2. Le chercheur a alors présumé, pour les variables autres que la prévisibilité de la tâche, que la diversité des cas présents dans son échantillon n'était pas suffisamment grande pour faire apparaître une interaction.

L'intérêt de cette recherche, bien que limitée dans ses résultats, est de permettre de mieux comprendre que le rôle que peut jouer une méthode n'est pas le même dans les différents types d'activités d'un service informatique. Ainsi, un projet de maintenance, un projet de réalisation et un projet de conception n'ont pas les mêmes caractéristiques technologiques. On ne peut pas recourir aux mêmes formes de standardisation. Les buts de la coordination personnelle sont différents (améliorer la productivité des informaticiens, contrôler l'avancement des travaux, guider le projet pour qu'il converge...).

Dans la mise en oeuvre d'une méthode, on voit alors se dégager deux plans: celui des dispositions permanentes, aidant à gérer la fonction informatique, où la coordination impersonnelle joue un rôle important ( le maître mot en est



"normalisation"); et celui d'un projet spécifique pour lequel la coordination à privilégier change de nature selon le stade d'avancement du projet (où le pilotage de convergence et consensus laisse place à un suivi formalisé de l'avancement).

## 6. Les approches orientées Risque

### 6.1 Incertitude du projet

G.Davis [Davis,1986] considère que l'incertitude doit guider le choix d'une **stratégie de développement** et celui d'une **méthode de détermination des besoins**.

L'incertitude du projet dépend de:

- la taille du projet,
- le degré de formalisation des activités à informatiser,
- la compréhension qu'ont les utilisateurs de leurs objectifs et leurs besoins,
- la connaissance qu'ont les informaticiens du domaine.

Les stratégies de développement varient selon le rôle que jouent les utilisateurs dans le processus. Ceux-ci sont à l'origine de la demande de système d'information et vont intervenir dans la validation des résultats livrés. Selon le degré d'incertitude globale du projet, on choisira (ordre croissant) l'une des stratégies suivantes:

1°) Les informaticiens se basent sur des *besoins initiaux*, préalablement définis, et la validation s'effectue une seule fois au terme de leur développement.

2°) Le projet est mené en suivant un *processus de certification linéaire*, c'est-à-dire qu'il suit le cycle de développement classique<sup>1</sup>; à la fin de chaque étape, une validation a lieu.

3°) Le *processus de certification itératif* suit le même schéma que le processus de certification linéaire, mais il autorise à boucler sur chaque étape si la validation complète n'est pas acquise.

4°) L'*expérimentation* est une stratégie de développement basée sur l'élaboration de prototypes.

Pour ce qui est de la détermination des besoins en information, il s'agit d'évaluer les risques spécifiques à cette phase. Les facteurs de risque sont les suivants:

- la *stabilité des besoins* en information,
- la *capacité des utilisateurs* à exprimer leurs besoins,
- la *capacité des analystes* à recueillir et évaluer les besoins.

Leur évaluation permet d'estimer une incertitude globale, qui conduit à choisir une des méthodes suivantes:

1. S'appuyer sur le recueil des souhaits: c'est l'enquête par les analystes auprès des utilisateurs.

---

<sup>1</sup>Voir par exemple la norme AFNOR Z67-101.

2. Etudier le système d'information existant ou un système d'information automatisé analogue.
3. Représenter le fonctionnement du domaine devant être couvert par le futur système d'information automatisé, c'est-à-dire étudier les activités du système opérant et en déduire les besoins en information.
4. Elaborer progressivement la définition des besoins en construisant un prototype<sup>1</sup>.

Les quatre types de méthodes de détermination des besoins ci-dessus sont classés en ordre croissant selon le degré d'incertitude auquel ils sont adaptés. Ainsi, une faible incertitude orientera vers l'enquête, une incertitude élevée fera préférer une approche prototypage.

Même si la notion d'incertitude globale peut être contestée, l'intérêt majeur de cette recherche est d'attirer l'attention sur l'importance des caractéristiques des acteurs (informaticiens et utilisateurs) dans le processus de développement. Cela rompt avec une tradition générale des méthodes qui recherchent l'objectivité, même elles contiennent souvent une image implicite du concepteur qui est fortement influencée par celle du chercheur dans sa communauté de recherche [Iivari,1983]. Cela conduit à identifier deux processus dans le développement d'un système d'information: l'un, rationnel, vise la production d'une application; l'autre, politique, cherche à ce que les décisions nécessaires soient prises.

## 6.2 Profil de risque du projet

N.Ahituv et S.Neumann [Ahituv,1984], deux chercheurs israéliens, ont proposé, en s'appuyant sur la littérature, un ensemble de facteurs qui appellent des **variations dans les activités de développement de système d'information**. Ainsi, le cycle de vie traditionnel ne doit-il plus être vu comme une séquence rigide d'activités, mais comme un cadre général à partir duquel on peut dériver le plan adapté à un projet spécifique. Les facteurs de contingence touchent au projet lui-même ou à son environnement et permettent d'établir un profil de risque du projet.

Parmi ces facteurs, on trouve:

- l'étendue organisationnelle, c.à.d. le nombre d'entités concernées;
- l'importance du projet, mesurée par exemple par son budget;
- la maturité organisationnelle, mesurée soit par la position de l'entreprise dans les étapes de Nolan<sup>1</sup>, soit plus simplement par le nombre d'applications déjà installées;

---

<sup>1</sup>Le recours à un prototype apparaît de deux façons: soit comme une stratégie de développement, opposée à celles qui s'appuient sur un processus linéaire de développement; soit comme un moyen de déterminer les besoins.

- l'organisation de la fonction informatique, c.à.d. son degré de centralisation;
- le degré de structuration du système d'information à développer;
- le degré d'innovation technologique;
- le type de développement, c.à.d. par exemple: développement d'un nouveau système d'information, installation d'un progiciel, modification d'un système existant, installation d'une nouvelle version d'un progiciel, maintenance corrective ou faiblement évolutive.

Les valeurs de ces facteurs doivent être prises en compte à chacune des étapes du déroulement du projet selon cinq aspects - cinq "dimensions"-, qui peuvent nécessiter une adaptation en fonction des valeurs prises par les facteurs:

- Les activités de base, parmi une liste de neuf activités (Etude de l'organisation; étude des besoins; étude des coûts/bénéfices; conception logique; conception physique; programmation; rédaction des procédures; formation; tests), en précisant quel est l'objectif de l'activité dans l'étape (pour ce projet), son niveau de détail, la façon de la mener...
- Le pilotage du projet : choix d'une organisation, procédures de suivi, procédures d'audit, dispositions d'assurance qualité.
- Le choix de la ressource humaine: détermination des types de profils nécessaires.
- Le choix des autres ressources: détermination des ressources nécessaires (temps machine, matériels...)
- Le délai nécessaire: la valeur de l'un ou l'autre facteur peut induire des délais accrus.

Les auteurs ne sont pas normatifs, ils n'indiquent pas de règles à suivre dans chaque dimension, selon la valeur de chaque facteur et selon l'étape du cycle de vie. Ils préconisent d'établir, à la fin de chaque étape, le plan de l'étape suivante, en se basant sur une matrice d'analyse Facteurs de contingence / Dimensions d'un projet.

La proposition de N.Ahituv et S.Neumann repose sur l'idée que chaque facteur de risque a un impact différent sur chacun des aspects de la conduite de projet, alors que la notion d'incertitude globale de G.Davis suppose qu'il y a compensation entre les différents facteurs, et qu'un degré analogue d'incertitude, quelle que soit sa source, appelle une réponse identique.

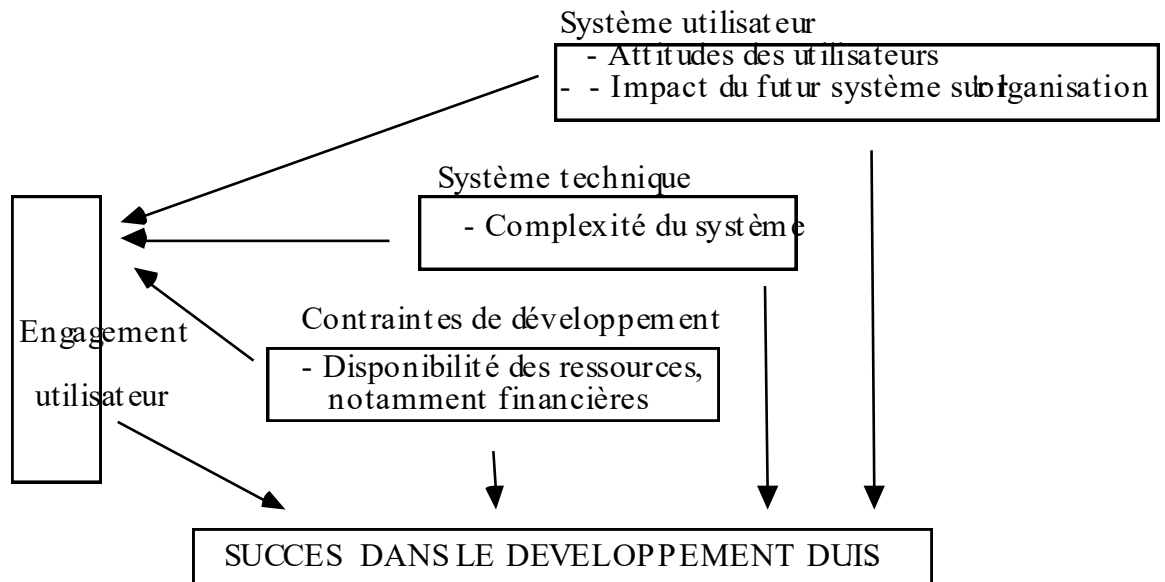
### 6.3 Complexité et contraintes du projet

P.Tait et I.Vessey [Tait,1988], deux chercheurs australiens, sont partis du fait que les recherches sur les effets de la **participation des utilisateurs** aboutissent à des résultats parfois contradictoires [Ives,1984]. Ils en ont déduit une approche contingente, à travers un modèle dans lequel les éléments qui affectent

---

<sup>1</sup>Selon R.L.Nolan [Nolan,1974], l'informatique se développe dans une Organisation en passant par quatre stades de croissance, caractérisés par le nombre d'applications, la spécialisation des personnels informatiques et le degré d'intégration et de contrôle de la fonction informatique.

l'engagement des utilisateurs (Système utilisateur, Système technique et Contraintes de développement ) sont également ceux qui ont un effet sur la pertinence du système d'information et la bonne fin de son développement.



*Figure 3 : Facteurs affectant l'engagement des utilisateurs*

Ce modèle a été testé sur quarante deux projets, dans trente entreprises en Australie, à travers l'envoi d'un questionnaire.

Il en ressort que:

- Il n'y a pas de relation directe entre le degré d'engagement de l'utilisateur et le succès dans le développement du système d'information.
- Il n'y a pas de relation significative entre le Système utilisateur (attitude vis-à-vis du nouveau système et impact de ce futur système sur l'organisation) et son engagement.

Par contre:

- Si les ressources sont peu disponibles (temps et budget), les chances de succès diminuent.
- Si la complexité du système est élevée, l'engagement de l'utilisateur diminue le risque d'échec.
- Si les ressources sont faibles, la participation de l'utilisateur empire la situation; un processus de développement non participatif semble préférable.

L'intérêt de cette recherche est de ne plus considérer la participation des utilisateurs comme bonne ou mauvaise, dans l'absolu.

Ses limites tiennent à ce que les différentes formes de participation ne sont pas différenciées. Or, les objectifs poursuivis par la participation sont variés<sup>1</sup> et ils se traduisent par des modalités différentes.

#### 6.4 Type de risque du projet

La critique d'une approche linéaire et unique de développement a conduit à la notion de "**modèle de développement**" ("process model"), qui est défini par sa fonction: un modèle de développement fournit les étapes à parcourir lors du développement d'un système d'information, en énumérant, classant, spécifiant et ordonnant les tâches à accomplir.

**B.W.Boehm** [Boehm,88] a une vision diachronique et centrée sur la production d'une application. Il considère ainsi que les modèles de développement ont suivi une évolution où chacun se construit en cherchant à dépasser les défauts du précédent:

- Le modèle "*code-and-fix*" des premiers jours de l'informatique de gestion était articulé en deux étapes:
  - programmation
  - détermination des besoins à travers la mise au point.
- Le modèle de la "*cascade*" ("waterfall model"), succession unique d'étapes fixées à l'avance, fut largement répandu.
- Le modèle du "*développement évolutif*" ("evolutionary design model") est basé sur une succession de cycles constitués des trois étapes suivantes:
  - programmation
  - expérimentation
  - détermination des besoins,

B.W.Boehm considère à certains égards ce modèle proche du modèle "code-and-fix".

- Le modèle de la "*transformation automatique*" ("transform model") est basé sur la possibilité de transformer automatiquement des spécifications en programmes. La non-disponibilité de tels outils logiciels limitent le recours à ce modèle.
- Le dernier modèle, celui de la "*spirale*" ("spiral model") est considéré comme le plus évolué car il intègre les modèles précédents et permet de déterminer la meilleure combinaison pour une situation donnée. Le développement se fait à travers une succession de cycles, qui comportent chacun les étapes génériques suivantes:
  - analyse du risque
  - développement d'un prototype
  - simulation et essais
  - détermination des besoins (à des mailles différentes selon le cycle)
  - validation des besoins
  - planification du cycle suivant.

---

<sup>1</sup>Par exemple, obtenir une connaissance sur un domaine, ou chercher l'acceptation du futur système, ou encore faciliter les prises de décision.

Le dernier cycle se termine de façon classique sur les tâches finales d'intégration, recette et implantation du logiciel, dont divers lots ont pu être issus des cycles antérieurs.

D'après B.W.Boehm, ce modèle en spirale peut être utilisé de façon flexible selon les **types de risques**.

- Si le risque majeur du projet est de dépasser le *budget* ou les *délais*, alors que la détermination des besoins ou les performances ne sont pas source de risque, l'application du modèle conduira à un seul cycle, équivalent au modèle en cascade.
- Si les besoins sont stables et que le risque majeur est lié à des *erreurs de programmation*, on sera conduit à réduire la spirale à un seul cycle à deux étapes: spécifications précises, suivies d'une programmation sans modification des spécifications.
- Si les contraintes (budget et délai) sont faibles et s'il n'y a pas de problème d'intégration, alors que le risque majeur réside dans une mauvaise *définition du dialogue homme-machine* ou des *besoins d'aide à la décision*, alors le modèle en spirale se ramènera au modèle de développement évolutif.
- Si des *outils de développement automatisé* sont disponibles, le modèle en spirale les utilisera soit pour développer les prototypes si les risques sont importants, soit en se ramenant au modèle de la transformation automatique s'ils le sont moins.
- Si un projet présente une combinaison de différents risques, alors l'analyse des risques permettra, selon B.W.Boehm, de recourir à une combinaison des modèles ci-dessus lors de la détermination de chaque cycle.

## 7. Les approches orientées objectifs

### 7.1 Nature du système d'information et du processus de développement

K.Lyytinen [Lyytinen,1987] considère que les modèles de développement correspondent à différentes visions sur la nature du système d'information et sur celle du processus de développement. Ces visions s'appréhendent à travers deux critères:

- la **cible visée**: veut-on obtenir un système technique ou un système social?
- la **dimension humaine** dans le processus de développement: est-ce un problème individuel ou un problème collectif?

Le positionnement selon ces deux critères permet d'identifier trois classes de modèles, aux orientations différentes:

- une orientation *ingénierie* : la cible est technique;

- une orientation *apprentissage*: la cible est un système social, avec une dimension individuelle;
- une orientation *dialogue*: la cible est également un système social, mais la dimension est collective.

Les sept modèles retenus sont les suivants:

- Le modèle du *cycle de vie*, analogue au modèle en cascade.
- Le modèle du *prototypage*, où le développement d'un prototype permet de définir les besoins.
- Le modèle *PSC*, proposé par des chercheurs finlandais [Kerola,1981], qui affine le modèle du cycle de vie, en distinguant trois niveaux de préoccupation: le niveau "pragmatique" (P), où l'on étudie les impacts et les objectifs du système; le niveau "sémantique" (S), où l'on conçoit la solution fonctionnelle; le niveau "constructif"(C), où l'on construit la solution technique. Ce modèle est comparable à celui proposé par Merise.
- Le modèle de *développement évolutif*, comme celui de la classification de B.W.Boehm.
- Le modèle du "*changement organisationnel*" propose de calquer le développement d'un système d'information sur les étapes à suivre pour faire adopter une innovation. L'approche socio-technique d'E.Mumford [Mumford,1981] ou le guide Actif [A.C.T.I.F.,1981] relèvent de cette catégorie.
- Le modèle de la "*négociation*" considère le développement d'un système d'information comme une intervention politique dans un ordre social précédemment négocié. Ce modèle est inspiré d'une pratique dans les pays scandinaves, notamment en Norvège, où tout développement de système d'information doit faire l'objet d'un accord avec les syndicats et les représentants des différents groupes, dans le souci de donner un droit de regard à l'utilisateur final.
- Le modèle du "*discours*" est une recherche participative pour trouver des raisons rationnelles pour agir, à travers un échange d'arguments, de "discours", entre les différentes parties. Cela permet de réconcilier des points de vue extrêmes lors de la conception et l'évolution d'un système d'information. Ce modèle est peu opérationnel.

La répartition en classes est représentée sur le tableau suivant:

Cible \ Dimension humaine		<i>Individuelle</i>	<i>Collective</i>
		<i>Systeme technique</i>	<b>INGENIERIE</b>
	Cycle de vie	Prototypage	
<i>Systeme social</i>	<b>APPRENTISSAGE</b>	<b>DIALOGUE</b>	
	Développement évolutif Changement organisationnel	Négociation Discours	

Figure 4: Modèles de développement

Le passage d'un modèle unique à une diversité de types de modèles donne la possibilité de sélectionner une approche en adéquation avec les objectifs poursuivis, même si la réflexion sur le choix d'un modèle reste encore limitée.

## 7.2 Les paradigmes du développement de système d'information

R.Hirschheim et H.K.Klein [Hirschheim,1989] considèrent que lorsque l'on développe un système d'information, on s'appuie (en général, implicitement) sur une conception de la nature d'une organisation humaine et de la nature des tâches de conception. Ces présupposés orientent le processus de développement et ses résultats.

On peut caractériser une catégorie de présupposés par les réponses que l'on apporte à ces deux groupes de questions:

1°) Comment acquérir la connaissance nécessaire pour construire le système d'information (problème de la détermination des besoins)? Y a-t-il une vision objective de la réalité ou bien toute représentation de la réalité ne dépend-elle pas de l'observateur et de son point de vue?

2°) Qu'est-ce qu'une organisation sociale? Est-ce un système ordonné où chacun peut trouver une place? Ou bien est-ce un lieu d'oppositions irréductibles, le terrain de conflits entre des intérêts opposés?



Les auteurs, en combinant les réponses, proposent quatre "paradigmes"<sup>1</sup> qui induisent un rôle du concepteur et une démarche spécifique.

①	Ordre	②
Connaissance		Connaissance
③	Conflit	④
objective		subjective

*Figure 5 : Paradigmes de développement*

Examinons-les successivement:

1. Le concepteur est un expert; il doit rechercher la meilleure démarche.
2. Le concepteur est un catalyseur, son art relève de la maïeutique: il doit permettre aux gestionnaire de définir le système le plus pertinent. Il puise pour cela dans son arsenal d'outils méthodologiques, sans être lié par une démarche pré-établie.
3. Le concepteur se situe du côté des utilisateurs finals dont il doit défendre les intérêts, notamment en concevant un système qui améliore leur qualité de vie au travail.
4. Le concepteur doit favoriser le progrès social de l'Organisation. Il va chercher dans les méthodes des éléments permettant de construire un consensus entre groupes aux intérêts communs.

Ces différentes visions du rôle du concepteur se traduisent par des modalités de participation spécifiques (choix des utilisateurs, objectifs de participation, organisation du travail de conception..) et par le recours à une démarche adaptée aux acteurs impliqués.

Selon l'approche adoptée, le système d'information construit sera différent<sup>2</sup>. Les différences pourront porter sur:

- les flux d'information,
- l'architecture technique,
- les contrôles que le système d'information permet sur les différents groupes d'utilisateurs,
- les autorisations d'accès aux informations,
- le suivi des erreurs.

Les deux recherches ci-dessus mettent l'accent sur la marge de liberté qui existe théoriquement dans le processus de développement. Les choix sociaux, d'organisation du travail, de qualification de postes, de contrôles, etc. ne sont pas les conséquences du développement d'un système technique. A l'inverse, ces

<sup>1</sup>Ils entendent par "paradigme" un ensemble de postulats qui guident la résolution du problème de développement.

<sup>2</sup>Sans que l'on puisse toutefois établir une correspondance entre paradigme et type de système d'information.

orientations influent à la fois sur les modalités de développement et sur le résultats.

## Conclusion

Les recherches sur la contingence marquent une tendance à appréhender l'activité de développement non comme un cas particulier de résolution de problème (problem-solving), mais comme un problème de pilotage de projet. Les difficultés naissent, en effet, de l'intervention d'acteurs multiples et de la nécessité de prendre simultanément en compte des aspects divers (techniques, humains, financiers, stratégiques...).

Ainsi, ce sont moins les types de problèmes de conception qui appellent une réponse adaptée que les objectifs retenus ou les risques identifiés. Le choix et la mise en oeuvre d'une méthode apparaissent donc liés à l'aspect collectif d'un projet et à son pilotage.

Cet aspect devrait conduire les méthodes à proposer plusieurs types de démarches ainsi qu'un guide permettant d'opter pour la plus appropriée ou de construire des démarches spécifiques à chaque contexte de projet.

Ce dépassement d'une approche unique de développement constitue bien une ligne d'évolution de Merise, confirmée par les acteurs du projet Euromethod, qui cherchent notamment à définir un modèle de mise en pratique d'une méthode intégrant ses conditions d'application.

## BIBLIOGRAPHIE

- A.C.T.I.F. (*Amélioration des Conditions de Travail et d'Informatique*). *Informatisation et vie au travail. Un guide pour maîtriser les impacts sociaux du développement de l'informatique* - Mission à l'informatique, Ministère de l'industrie Paris, Ed. d'Organisation, Paris, 1981.
- Ahituv N., Hadass M. et Neumann S. "A Flexible Approach to Information System Development", *MIS Quarterly*, juin 1984.
- Boehm B.W. "A spiral Model of Software Development and Enhancement" *Computer*, mai 1988.
- Davis G.B., Olson M.H., Ajenstat J. et Peaucelle J.L. *Systèmes d'Information pour le Management*, 2 vol., Editions Economica, Paris, 1986.
- Drazin R. et Van de Ven A. "Alternative forms of fit in contingency theory" *Administrative Science Quarterly*, décembre 1985.
- Gladden G.R. "Stop the life-cycle, I want to get off" *ACM SIGSOFT Software Engineering Notes* 7,2. 1982.
- Hirschheim R. et Klein H.K. "Four Paradigms of Information Systems Development" *Communications of the ACM*, octobre 1989.

- Iivari J. et Kerola P. "A Socio-cybernetic Framework for the Feature Analysis of Information System Design Methodologies" in [Olle,1983].
- Ives B. et Olson M. "User Involvement and MIS Success: A Review of Research", *Management Science*, Volume 30, N°5, mai 1984.
- Ives B., Olson M. et Baroudi J. "The Measurement of User Information Satisfaction", *Communications of the ACM*, octobre 1983.
- Kerola P. et Freeman P. "A comparison of life-cycle models" in *Proceedings of the 5th International Conference on Software Engineering*, IEEE, New York, 1981.
- Kyu Kim K. "Organizational Coordination and Performance in Hospital Accounting Information Systems: An Empirical Investigation", *The Accounting Review*, juillet 1988.
- Lyytinen K. "Different perspectives on Information Systems: Problems and Solutions" *ACM Computing Surveys*, Vol.19, N°1, mars 1987.
- Mintzberg H. - *Structure et dynamique des organisations* - Ed. d'Organisation, Paris 1984.
- Mumford E. et Henshall D. - *A participative approach to computer system design* - Associated Business Press, Londres, 1979 .
- Mumford E. - "System Design and Human Needs" - in *The Impact of Systems Change in Organisations* - Alphen aan den Rijn, Ed. Sijthoff et Noordhoff, Hollande, 1980.
- Nolan R.L. et Gibson C.F. "Managing the Four Stages of EDP Growth", *Harvard Business Review*, janvier-février 1974. Traduction française: *Informatique nouvelle*, juin 1977.
- Nolan R.L. "Gérer les crises de l'informatique " *Informatique nouvelle*, juillet-août 1979, reproduit de la *Harvard Business Review*, mars-avril 1979.
- Olle T.W., Sol H.G. et Tully C.J. - *Information Systems Design Methodologies: A Feature Analysis* - North-Holland, Amsterdam, 1983.
- Perrow C.B. *Organizational Analysis: A Sociological View*, Wadsworth Publishing Co., 1970.
- Schoonhoven C.B. "Problems with Contingency Theory: Testing Assumptions Hidden within the Language of Contingency" Theory", *Administrative Science Quarterly*, sept. 1981.
- Tait P. et Vessey I. "The effect of User Involvement on System Success: A Contingency Approach", *MIS Quarterly*, mars 1988.
- Woodward J. *Industrial organization : theory and practice*, Oxford University Press, 1965.