

Introduction au développement du logiciel

Vers le génie logiciel

Pascal ANDRE

MIAGE
Université de Nantes

Master Miage M1



Plan

- 1 Introduction
- 2 Génie logiciel
- 3 Projet informatique
- 4 Méthode de développement
- 5 Qualité

Bibliographie sommaire

◇ Généralités et techniques de base

- [AV01a] Pascal André and Alain Vailly.
Conception de Systèmes d'Information, Panorama des méthodes et des techniques, volume 1 of Collection Technosup.
Editions Ellipses, 2001.
ISBN 2-7298-0479-X.
- [MR93] José Morejon and Jean-René Rames.
Conduite de projets informatiques.
InterEditions, 1993.
Principes et techniques s'appuyant sur la méthode MERISE, ISBN 2-7296-0457-X.
- [Vil95] Françoise Villeneuve.
Conduite de projet.
CNRS, 1995.
http://dsi-val.dsi.cnrs.fr/bureau_qualite/qualite/conduit.htm.

Introduction

- De l'idée au code
 - exprimer
 - programmer
 - vérifier

Introduction

- De l'idée au code
 - exprimer
 - programmer
 - vérifier
- avec méthode...
 - dans le bon ordre
 - éviter l'anarchie
 - travailler en groupe

Introduction

- De l'idée au code
 - exprimer
 - programmer
 - vérifier
- avec méthode...
 - dans le bon ordre
 - éviter l'anarchie
 - travailler en groupe
- ... et qualité
 - modèles (correct, fiable, évolutif...)
 - processus (efficace, rentable...)

Introduction

- De l'idée au code
 - exprimer
 - programmer → **MODELES**
 - vérifier
- avec méthode...
 - dans le bon ordre **PROCESSUS**
 - éviter l'anarchie → **GESTION DE PROJET**
 - travailler en groupe
- ... et qualité
 - modèles (correct, fiable, évolutif...)
 - processus (efficace, rentable...) → **QUALITE**

Le développement individuel

→ méthodes "agiles", XP

- Analyse légère ou dans la "tête"

Le développement individuel

→ méthodes "agiles", XP

- Analyse légère ou dans la "tête"
 - oublis
 - erreurs tardives → manque de réflexion

Le développement individuel

→ méthodes "agiles", XP

- Analyse légère ou dans la "tête"
- Programmation incrémentale et itérative sur le test

Le développement individuel

→ méthodes "agiles", XP

- Analyse légère ou dans la "tête"
- Programmation incrémentale et itérative sur le test
 - nombreux allers/retours
 - réactivité → manque d'organisation

Le développement individuel

→ méthodes "agiles", XP

- Analyse légère ou dans la "tête"
- Programmation incrémentale et itérative sur le test

⇒ problèmes à grande échelle

- magma d'organisation
- durées aléatoires
- expérience et capitalisation individuelle

Le développement individuel

→ méthodes "agiles", XP

- Analyse légère ou dans la "tête"
- Programmation incrémentale et itérative sur le test

⇒ problèmes à grande échelle

- magma d'organisation
- durées aléatoires
- expérience et capitalisation individuelle

⇒ pas de travail de groupe, pas rationnel

Le développement en groupe

- établir les tâches et les résultats

Le développement en groupe

- établir les tâches et les résultats
- définir les acteurs

Le développement en groupe

- établir les tâches et les résultats
- définir les acteurs
- organiser le développement

Le développement en groupe

- établir les tâches et les résultats
- définir les acteurs
- organiser le développement
- gérer la communication (documents, média, ...)

Le développement en groupe

- établir les tâches et les résultats
- définir les acteurs
- organiser le développement
- gérer la communication (documents, média, ...)
- superviser le développement

Le développement en groupe

- établir les tâches et les résultats
- définir les acteurs
- organiser le développement
- gérer la communication (documents, média, ...)
- superviser le développement
- améliorer la qualité

Le développement en groupe

- établir les tâches et les résultats
- définir les acteurs
- organiser le développement
- gérer la communication (documents, média, ...)
- superviser le développement
- améliorer la qualité

⇒ pour qui ?

Le développement pour un client

- 1 ajouter une dimension coût

Le développement pour un client

1 ajouter une dimension coût

- analyse de coûts : financier, temporel, humain...
- analyse de compétence : technologies requises (formation ?)
- analyse de performance : rentabilité...

Le développement pour un client

- 1 ajouter une dimension coût
- 2 ajouter de nouvelles étapes

Le développement pour un client

- 1 ajouter une dimension coût
- 2 ajouter de nouvelles étapes
 - étude d'opportunité
 - études de faisabilité
 - analyse de risques
 - gestion des risques
 - décisions

Le développement pour un client

- 1 ajouter une dimension coût
- 2 ajouter de nouvelles étapes
- 3 modifier le développement

Le développement pour un client

- 1 ajouter une dimension coût
- 2 ajouter de nouvelles étapes
- 3 modifier le développement
 - rôle du client
 - spécification
 - validation
 - décisions

Le développement pour un client

- 1 ajouter une dimension coût
- 2 ajouter de nouvelles étapes
- 3 modifier le développement
- 4 prévoir l'assurance qualité, l'automatisation...

Le développement pour un client

- 1 ajouter une dimension coût
- 2 ajouter de nouvelles étapes
- 3 modifier le développement
- 4 prévoir l'assurance qualité, l'automatisation...
 - qualité des résultats (validité, robustesse) = satisfaire le client
 - qualité du processus = respecter délais et coûts

Le développement par un prestataire

- 1 ajouter une dimension production

Le développement par un prestataire

1 ajouter une dimension production

- analyse de coûts : financier, temporel, humain...
- analyse de compétence : technologies requises (formation ?)
- analyse de performance : rentabilité...

Le développement par un prestataire

- 1 ajouter une dimension production
- 2 ajouter de nouvelles étapes

Le développement par un prestataire

- 1 ajouter une dimension production
- 2 ajouter de nouvelles étapes
 - formations
 - maintenance des applications (corrective, évolutive)

Le développement par un prestataire

- 1 ajouter une dimension production
- 2 ajouter de nouvelles étapes
- 3 prendre en compte d'autres préoccupations

Le développement par un prestataire

- 1 ajouter une dimension production
- 2 ajouter de nouvelles étapes
- 3 prendre en compte d'autres préoccupations
 - gestion des ressources humaines (formations, carrières)
 - gestion commerciale ou interne des applications

Le développement par un prestataire

- 1 ajouter une dimension production
- 2 ajouter de nouvelles étapes
- 3 prendre en compte d'autres préoccupations
- 4 rentabiliser la production

Le développement par un prestataire

- 1 ajouter une dimension production
- 2 ajouter de nouvelles étapes
- 3 prendre en compte d'autres préoccupations
- 4 rentabiliser la production
 - développer en série (portabilité, évolutivité, généralité)
 - développer en partie (réutilisabilité)
 - automatiser la production

Le développement par un prestataire

- 1 ajouter une dimension production
- 2 ajouter de nouvelles étapes
- 3 prendre en compte d'autres préoccupations
- 4 rentabiliser la production

⇒ produire du logiciel de qualité, rationaliser le développement, rentabiliser les investissements

Le développement par un prestataire

- 1 ajouter une dimension production
- 2 ajouter de nouvelles étapes
- 3 prendre en compte d'autres préoccupations
- 4 rentabiliser la production

⇒ produire du logiciel de qualité, rationaliser le développement, rentabiliser les investissements

- ◇ besoin de méthodes et d'outil de développement
- ◇ besoin de méthodes et d'outil de gestion de projets

⇒ une réponse : le Génie logiciel

Plan

- 1 Introduction
- 2 Génie logiciel**
- 3 Projet informatique
- 4 Méthode de développement
- 5 Qualité

Le Génie logiciel

- objectifs

Le Génie logiciel

■ objectifs

mettre en œuvre des moyens pour réaliser du logiciel de qualité en respectant des contraintes de coûts.

Le Génie logiciel

- objectifs
- définition

Le Génie logiciel

- objectifs
- définition

Le Génie Logiciel est l'art de construire INDUSTRIELLEMENT du logiciel.

Le Génie logiciel

- objectifs
- définition
- large spectre : des langages de programmation à la gestion d'organisations

Le Génie logiciel

- objectifs
- définition
- large spectre : des langages de programmation à la gestion d'organisations
- soumis à l'évolution des technologies et des métiers

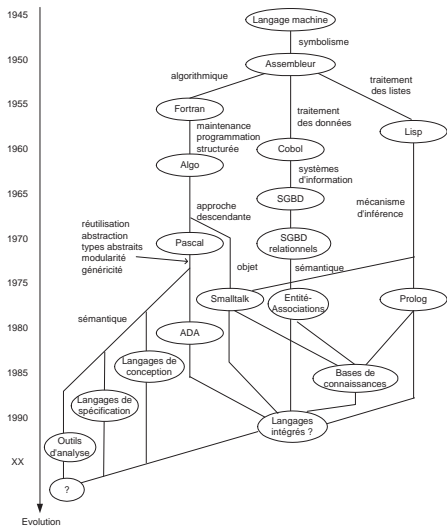
Le Génie logiciel

- objectifs
- définition
- large spectre : des langages de programmation à la gestion d'organisations
- soumis à l'évolution des technologies et des métiers
- Atelier GL indispensable

Le Génie logiciel

- objectifs
- définition
- large spectre : des langages de programmation à la gestion d'organisations
- soumis à l'évolution des technologies et des métiers
- Atelier GL indispensable
 - édition
 - vérification
 - génération automatique
 - documentation
 - réutilisation
 - personnalisation

Evolution du Génie logiciel



Plan

- 1 Introduction
- 2 Génie logiciel
- 3 Projet informatique**
- 4 Méthode de développement
- 5 Qualité

Projet informatique

Projet d'informatisation =

Projet informatique

Projet d'informatisation =

- aspects techniques

Projet informatique

Projet d'informatisation =

- aspects techniques ⇒ **méthode de développement**

Une méthode est une technique de résolution de problèmes [Lau86].

Un aspect clé de la méthode est de privilégier l'étude (l'analyse) avant de commencer à concevoir le logiciel.

Il faut bien poser le problème pour bien le résoudre.

Le terme de **méthode** recouvre plusieurs notions.

Projet informatique

Projet d'informatisation =

- aspects techniques ⇒ **méthode de développement**
 - philosophie (approche du problème)
 - démarche (processus)
 - notation (norme, langage)
 - outils

Projet informatique

Projet d'informatisation =

- aspects techniques ⇒ méthode de développement

Merise

- philosophie (approche du problème)
BD relationnelles, Prog structurée
- démarche (processus)
par niveau, par étapes
- notation (norme, langage)
E-A-P, Petri, SCOM...
- outils
AMCD, WinDesign...

Projet informatique

Projet d'informatisation =

- aspects techniques ⇒ méthode de développement
- aspects organisationnels

Projet informatique

Projet d'informatisation =

- aspects techniques ⇒ méthode de développement
- aspects organisationnels ⇒ gestion de projet

La conduite de projet (gestion de projet, management de projet) "définit l'environnement nécessaire pour que le projet se déroule dans les meilleurs conditions de coût, de délais ainsi que de qualité des résultats produits" [MR93].

En plus des méthodes et techniques de développement, des méthodes et techniques de gestion de projet sont mises en œuvre pour réussir le projet. ~~~>

Projet informatique

Projet d'informatisation =

- aspects techniques ⇒ méthode de développement
- aspects organisationnels ⇒ gestion de projet

↪ "La conduite de projet est un processus difficile à maîtriser car interviennent plusieurs facteurs de risque tels que les coûts et les délais à respecter, les technologies à maîtriser, les ressources humaines à gérer." [Vil95].

Projet informatique

Projet d'informatisation =

- aspects techniques ⇒ **méthode de développement**
- aspects organisationnels ⇒ **gestion de projet**
 - gestion du besoin (définition, validation, évolution)
 - gestion des ressources et des coûts (planification, suivi, optimisation, capitalisation)
 - gestion des risques (évaluation, suivi, décision)
 - gestion de la communication entre acteurs
 - gestion de la qualité des résultats produits

Projet informatique

Projet d'informatisation =

- aspects techniques ⇒ méthode de développement
- aspects organisationnels ⇒ gestion de projet

Projet informatique

- gestion du besoin (définition, validation, évolution)
réunions, revues, cahiers des charges
- gestion des ressources et des coûts (planification, suivi, optimisation, capitalisation)
humains, matériels, financiers
- gestion des risques (évaluation, suivi, décision)
- gestion de la communication entre acteurs
- gestion de la qualité des résultats produits
certification, contrôle qualité

Projet informatique

Projet d'informatisation =

- aspects techniques ⇒ méthode de développement
- aspects organisationnels ⇒ gestion de projet

→ domaine complexe

Projet informatique

Projet d'informatisation =

- aspects techniques ⇒ méthode de développement
- aspects organisationnels ⇒ gestion de projet

Dans ce qui suit, on se focalise sur le développement (la méthode au sens large).

La gestion de projet est hors contexte, voir des pointeurs à <http://www.sciences.univ-nantes.fr/info/perso/permanents/andre/COURS/cours.html>

Projet - Variantes

Les **applications informatiques** peuvent se classer en trois catégories majeures :

Projet - Variantes

Les **applications informatiques** peuvent se classer en trois catégories majeures :

- 1 Systèmes d'information de gestion **SIG**

Projet - Variantes

Les **applications informatiques** peuvent se classer en trois catégories majeures :

- 1 Systèmes d'information de gestion **SIG**
 - stockage volumineux
 - nombreux traitements triviaux (consultation, mises-à-jour)
 - quelques traitements complexes en mise-à-jour
 - assurer la cohérence des informations
 - soumis aux évolutions technologiques (distribution, persistance, IHM)

Projet - Variantes

Les **applications informatiques** peuvent se classer en trois catégories majeures :

- 1 Systèmes d'information de gestion **SIG**
- 2 Systèmes temps réel **STR**

Projet - Variantes

Les **applications informatiques** peuvent se classer en trois catégories majeures :

- 1 Systèmes d'information de gestion **SIG**
- 2 Systèmes temps réel **STR**
 - réactivité forte
 - qualité (sûreté, fiabilité...)
 - traitements souvent simples mais conditions complexes
 - assurer la cohérence des événements (rôle du temps)
 - soumis aux évolutions matérielles (capteurs, actionneurs, environnement)

Projet - Variantes

Les **applications informatiques** peuvent se classer en trois catégories majeures :

- 1 Systèmes d'information de gestion **SIG**
- 2 Systèmes temps réel **STR**
- 3 Systèmes intelligents, calcul scientifique **SI, CS**

Projet - Variantes

Les **applications informatiques** peuvent se classer en trois catégories majeures :

- 1 Systèmes d'information de gestion **SIG**
- 2 Systèmes temps réel **STR**
- 3 Systèmes intelligents, calcul scientifique **SI, CS**
 - algorithmes complexes
 - bases d'information plus ou moins importantes
 - qualité : efficacité
 - peu soumis aux évolutions technologiques (techniques de programmation, IHM)

Projet - Variantes

Les **applications informatiques** peuvent se classer en trois catégories majeures :

- 1 Systèmes d'information de gestion **SIG**
- 2 Systèmes temps réel **STR**
- 3 Systèmes intelligents, calcul scientifique **SI, CS**

il y en a d'autres

Projet - Variantes

Les **applications informatiques** peuvent se classer en trois catégories majeures :

- 1 Systèmes d'information de gestion **SIG**
- 2 Systèmes temps réel **STR**
- 3 Systèmes intelligents, calcul scientifique **SI, CS**

il y en a d'autres

différents besoins \implies différentes méthodes

Plan

- 1 Introduction
- 2 Génie logiciel
- 3 Projet informatique
- 4 Méthode de développement**
- 5 Qualité

Méthode de développement (en bref)

Méthode =

Modèles, produits

Méthode de développement (en bref)

Méthode =

Modèles, produits

+ Processus

Méthode de développement (en bref)

Méthode =

Modèles, produits

+ Processus

+ Outils

Méthode de développement (Modèles)

- Un **modèle** est une interprétation explicite par son utilisateur de l'idée qu'il se fait d'une situation.

Méthode de développement (Modèles)

- Un **modèle** est une interprétation explicite par son utilisateur de l'idée qu'il se fait d'une situation.
- Il peut être exprimé par des mathématiques, des symboles, des mots, mais, essentiellement, c'est une description d'entités et de relations entre elles.

Méthode de développement (Modèles)

- Un **modèle** est une interprétation explicite par son utilisateur de l'idée qu'il se fait d'une situation.
- Il peut être exprimé par des mathématiques, des symboles, des mots, mais, essentiellement, c'est une description d'entités et de relations entre elles.
- Un modèle est correct s'il permet de répondre aux questions qu'on se pose.

Méthode de développement (Modèles)

- Un **modèle** est une interprétation explicite par son utilisateur de l'idée qu'il se fait d'une situation.
- Il peut être exprimé par des mathématiques, des symboles, des mots, mais, essentiellement, c'est une description d'entités et de relations entre elles.
- Un modèle est correct s'il permet de répondre aux questions qu'on se pose.
- Un modèle est **opérationnel** s'il peut être exécuté par une machine.

Méthode de développement (Modèles)

Les éléments d'un système peuvent se classer selon trois aspects majeurs :

Méthode de développement (Modèles)

Les éléments d'un système peuvent se classer selon trois aspects majeurs :

- 1 Structure, données \implies Point de vue statique

Méthode de développement (Modèles)

Les éléments d'un système peuvent se classer selon trois aspects majeurs :

- 1 Structure, données \implies Point de vue statique
 - informations
 - structure
 - contraintes
 - répartition

Méthode de développement (Modèles)

Les éléments d'un système peuvent se classer selon trois aspects majeurs :

- 1 Structure, données \implies Point de vue statique
- 2 Evolution dans le temps \implies Point de vue dynamique

Méthode de développement (Modèles)

Les éléments d'un système peuvent se classer selon trois aspects majeurs :

- 1 Structure, données \implies Point de vue statique
- 2 Evolution dans le temps \implies Point de vue dynamique
 - événements
 - actions
 - contraintes temporelles
 - concurrence

Méthode de développement (Modèles)

Les éléments d'un système peuvent se classer selon trois aspects majeurs :

- 1 Structure, données \implies Point de vue statique
- 2 Evolution dans le temps \implies Point de vue dynamique
- 3 Traitements \implies Point de vue fonctionnel

Méthode de développement (Modèles)

Les éléments d'un système peuvent se classer selon trois aspects majeurs :

- 1 Structure, données \implies Point de vue statique
- 2 Evolution dans le temps \implies Point de vue dynamique
- 3 Traitements \implies Point de vue fonctionnel
 - algorithmes
 - transactions et sécurité
 - assertions (contraintes)
 - parallélisme

Types de spécification

Les spécifications peuvent être classées selon leur forme ou leur degré de formalisme.

- les spécifications informelles, en langue naturelle, rédigées sans contraintes de forme,

Types de spécification

Les spécifications peuvent être classées selon leur forme ou leur degré de formalisme.

- les spécifications informelles, en langue naturelle, rédigées sans contraintes de forme,
- les spécifications standardisées, en langue naturelle, mais avec une structure, un format et des règles précises (notations, glossaire, index, historique...),

Types de spécification

Les spécifications peuvent être classées selon leur forme ou leur degré de formalisme.

- les spécifications informelles, en langue naturelle, rédigées sans contraintes de forme,
- les spécifications standardisées, en langue naturelle, mais avec une structure, un format et des règles précises (notations, glossaire, index, historique...),
- les spécifications semi-formelles, utilisant un langage de spécification textuel ou graphique, doté d'une syntaxe précise et d'une sémantique assez faible),

Types de spécification

Les spécifications peuvent être classées selon leur forme ou leur degré de formalisme.

- les spécifications informelles, en langue naturelle, rédigées sans contraintes de forme,
- les spécifications standardisées, en langue naturelle, mais avec une structure, un format et des règles précises (notations, glossaire, index, historique...),
- les spécifications semi-formelles, utilisant un langage de spécification textuel ou graphique, doté d'une syntaxe précise et d'une sémantique assez faible),
- les spécifications formelles, exprimées dans un langage à syntaxe et sémantique précises, construites sur une base théorique solide.

Méthode de développement (Processus)

Le processus (la démarche) précise

- les tâches (activités) à réaliser,

Méthode de développement (Processus)

Le processus (la démarche) précise

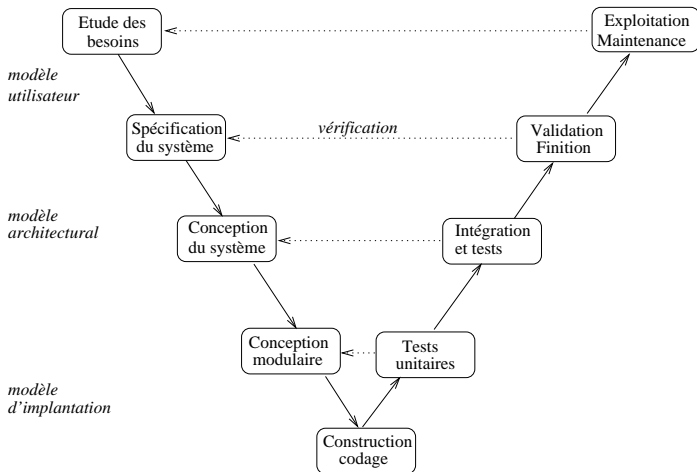
- les tâches (activités) à réaliser,
- les résultats attendus (modèles, produits, spécifications),

Méthode de développement (Processus)

Le processus (la démarche) précise

- les tâches (activités) à réaliser,
- les résultats attendus (modèles, produits, spécifications),
- l'ordre qui organise les tâches (le cycle de développement).

Processus : le cycle de référence



Modèles de processus

Principaux cycles

- linéaires (cascade, V, fontaine),

Modèles de processus

Principaux cycles

- linéaires (cascade, V, fontaine),
- contractuels (méthodes formelles),

Modèles de processus

Principaux cycles

- linéaires (cascade, V, fontaine),
- contractuels (méthodes formelles),
- itératifs (spirale),

Modèles de processus

Principaux cycles

- linéaires (cascade, V, fontaine),
- contractuels (méthodes formelles),
- itératifs (spirale),
- variantes (X, Y).

Méthode de développement (Processus)

Ne pas oublier

- Vérification : c'est contrôler que le produit respecte le cahier des charges. La vérification est donc plus une "affaire" de spécialistes.

Méthode de développement (Processus)

Ne pas oublier

- Vérification : c'est contrôler que le produit respecte le cahier des charges. La vérification est donc plus une "affaire" de spécialistes.
- Validation : c'est contrôler que le (produit) résultat correspond à ce qui était attendu. La validation est un contrôle qui fait intervenir largement les "utilisateurs".

Méthode de développement (Processus)

Ne pas oublier

- Vérification : c'est contrôler que le produit respecte le cahier des charges. La vérification est donc plus une "affaire" de spécialistes.
- Validation : c'est contrôler que le (produit) résultat correspond à ce qui était attendu. La validation est un contrôle qui fait intervenir largement les "utilisateurs".

Moyens

- Revues
- Tests
- Preuves
- Prototypage

Plan

- 1 Introduction
- 2 Génie logiciel
- 3 Projet informatique
- 4 Méthode de développement
- 5 Qualité**

Qualité

- Qualité des spécifications

Qualité

■ Qualité des spécifications

⇒ éviter

- le **bruit**
- le **silence**
- la **sur-spécification**
- la **contradiction**
- l'**ambiguïté**
- la **référence avant**
- le **vœu pieu**

Qualité

- Qualité des spécifications

Qualité du logiciel

Qualité

- Qualité des spécifications

Qualité du logiciel

- la **validité**
- la **robustesse**
- l'**extensibilité**
- la **réutilisabilité**
- la **compatibilité**

Qualité

■ Qualité des spécifications

Qualité du logiciel

- la **validité**
- la **robustesse**
- l'**extensibilité**
- la **réutilisabilité**
- la **compatibilité**

D'autres qualités du logiciel sont moins cruciales :
l'**efficacité**, la **portabilité**, la **vérifiabilité**, l'**intégrité**, la
facilité d'utilisation

Qualité

- Qualité des spécifications

Qualité du logiciel

- Qualité du processus

Qualité

- Qualité des spécifications

Qualité du logiciel

- Qualité du processus
 - **sûreté**
 - **terminaison**
 - **rigueur**
 - **cohérence, complétude**
 - **souplesse**
 - **accessibilité**
 - **rentabilité**

Qualité

- Qualité des spécifications

Qualité du logiciel

- Qualité du processus

Plus de détails ⇒ [AV01]

Qualité

- Qualité des spécifications

Qualité du logiciel

- Qualité du processus

Plus de détails \Rightarrow [AV01]

Améliorer la qualité :

Qualité

- Qualité des spécifications

Qualité du logiciel

- Qualité du processus

Plus de détails \Rightarrow [AV01]

Améliorer la qualité :

- rigueur
- structuration modulaire

Qualité

- Qualité des spécifications

Qualité du logiciel

- Qualité du processus

Plus de détails \Rightarrow [AV01]

Améliorer la qualité :

- rigueur \rightsquigarrow méthodes formelles
- structuration modulaire \rightsquigarrow méthodes à objets

Qualité

- Qualité des spécifications

Qualité du logiciel

- Qualité du processus

Plus de détails \Rightarrow [AV01]

Améliorer la qualité :

- rigueur \rightsquigarrow méthodes formelles Z
- structuration modulaire \rightsquigarrow méthodes à objets UML



Pascal André and Alain Vailly.

Conception de systèmes d'information ; Panorama des méthodes et des techniques,
volume 1 of Collection Technosup.

Editions Ellipses, 2001.

ISBN 2-7298-0479-X.



Jean.-Louis Laurière.

Résolution de problèmes par l'Homme et la machine.

Eyrolles, 1986.



José Morejon and Jean-René Rames.

Conduite de projets informatiques.

InterEditions, 1993.

Principes et techniques s'appuyant sur la méthode MERISE, ISBN
2-7296-0457-X.



Françoise Villeneuve.

Conduite de projet.

CNRS, 1995.

http://dsi-val.dsi.cnrs.fr/bureau_qualite/qualite/conduit.htm.