

# Conduite de projets informatiques

Développement,  
analyse et pilotage

3<sup>ème</sup> édition

**Brice-Arnaud GUÉRIN**

Téléchargement  
[www.editions-eni.fr](http://www.editions-eni.fr)



Les exemples à télécharger sont disponibles à l'adresse suivante :  
**<http://www.editions-eni.fr>**  
Saisissez la référence ENI de l'ouvrage **DP3CPR** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

## Avant-propos

- 1. Objectifs du livre ..... 15
- 2. Structure du livre ..... 16

## Chapitre 1

### Un projet informatique

- 1. Les enjeux ..... 19
  - 1.1 Les buts d'un projet ..... 19
  - 1.2 Les protagonistes d'un projet ..... 20
  - 1.3 Exemples de projets ..... 21
- 2. Les moyens et les ressources ..... 21
  - 2.1 Les moyens techniques ..... 22
    - 2.1.1 Les postes de développement ..... 22
    - 2.1.2 Les serveurs de documentation de projet ..... 23
    - 2.1.3 Les autres types d'équipements ..... 23
  - 2.2 Les moyens financiers ..... 24
  - 2.3 Les moyens généraux ..... 24
  - 2.4 Les ressources humaines ..... 24
- 3. Les contraintes ..... 25
  - 3.1 Le cahier des charges ..... 25
  - 3.2 La maîtrise des coûts ..... 25
  - 3.3 La maîtrise de la qualité ..... 26
  - 3.4 La maîtrise des délais ..... 27
  - 3.5 La matrice d'ajustement ..... 28
- 4. Le cadre d'un projet ..... 30
- 5. Les objectifs de la conduite de projet ..... 30

# 2 — Conduite de projets informatiques

Développement, analyse et pilotage

6.	Trois exemples de projets . . . . .	31
6.1	Mise en œuvre d'un CRM . . . . .	31
6.1.1	Contexte . . . . .	32
6.1.2	Besoin exprimé . . . . .	32
6.1.3	Cadre du projet . . . . .	33
6.1.4	Les enjeux du projet . . . . .	33
6.2	Développement d'une application de pilotage d'activité . . . . .	34
6.2.1	Analyse de la situation et perspectives . . . . .	34
6.2.2	Résumé des exigences . . . . .	35
6.2.3	Dotation et contraintes . . . . .	36
6.2.4	Les enjeux du chef de projet . . . . .	36
6.3	Création d'un site de vente en ligne . . . . .	37
6.3.1	Des moyens conséquents au service d'un projet unique . . . . .	37
6.3.2	De la stratégie au plan projet . . . . .	38
6.3.3	Les enjeux d'une équipe . . . . .	38

## Chapitre 2

### Les aspects financiers

1.	La structure de coût d'un projet . . . . .	39
1.1	Les salaires et prestations de réalisation . . . . .	40
1.1.1	Des grilles de coûts . . . . .	40
1.1.2	Négocier les coûts journaliers en régie . . . . .	41
1.1.3	Négocier les prestations au forfait . . . . .	41
1.1.4	Le plan de charge financier . . . . .	42
1.2	Les prestations d'hébergement . . . . .	42
1.2.1	Les modes d'hébergement . . . . .	43
1.2.2	Le plan de charge financier . . . . .	43
1.3	Les prestations d'exploitation et de maintenance . . . . .	44
1.4	Les prestations de support aux utilisateurs . . . . .	44
1.5	Les coûts refacturés . . . . .	45
1.6	Les coûts non liés à l'informatique . . . . .	45
2.	Les budgets . . . . .	46
2.1	Constitution des budgets . . . . .	46

2.2	Les indicateurs liés aux budgets. . . . .	47
3.	Le compte de résultats (Profit And Loss) . . . . .	47
3.1	Le modèle économique . . . . .	47
3.1.1	Licence et maintenance . . . . .	48
3.1.2	L'abonnement (SaaS) . . . . .	48
3.1.3	Le modèle transactionnel (Pay as you go). . . . .	48
3.2	La projection financière . . . . .	49
3.2.1	Les flux de trésorerie. . . . .	49
3.2.2	L'analyse financière. . . . .	51
3.2.3	Les indicateurs clés VAN, TRI, break even. . . . .	52
4.	Le business case. . . . .	54
4.1	La proposition de valeur . . . . .	54
4.2	Le top line financier . . . . .	55
4.3	L'executive summary . . . . .	55
5.	Le suivi financier des projets . . . . .	56
5.1	Les comptes rendus d'activité . . . . .	56
5.2	La reconnaissance du chiffre . . . . .	56
6.	Étude financière du site de vente en ligne. . . . .	56
6.1	Structure de coûts et budgets estimés. . . . .	57
6.1.1	Budget d'investissement CAPEX (capital expenditure) . .	57
6.1.2	Budget de fonctionnement OPEX (operational expenditure) . . . . .	58
6.2	Business case et P&L. . . . .	60
6.2.1	Résumé exécutif . . . . .	60
6.2.2	Proposition de valeur . . . . .	60
6.2.3	Prévisions de ventes . . . . .	61
6.2.4	Analyse financière sur trois ans. . . . .	62
6.2.5	Résumé financier . . . . .	63
6.3	Scénarios de suivi financier . . . . .	63
6.3.1	Budget d'investissement CAPEX. . . . .	63
6.3.2	Identification d'économies sur le budget de fonctionnement OPEX . . . . .	65
6.3.3	Révision du P&L. . . . .	66

# 4 — Conduite de projets informatiques

Développement, analyse et pilotage

## Chapitre 3

### La prise en compte du risque

1. Les trois axes . . . . .	67
2. Le modèle de développement . . . . .	70
2.1 Le modèle cascade . . . . .	71
2.2 Le modèle en V . . . . .	73
2.3 Le modèle itératif . . . . .	74
2.3.1 Le modèle RAD . . . . .	75
2.3.2 Le modèle Extreme Programming . . . . .	76
2.3.3 Le modèle RUP (Rational Unified Process) . . . . .	78
3. Le modèle d'analyse . . . . .	80
3.1 Le principe de modélisation en analyse . . . . .	80
3.1.1 L'exemple du dictionnaire de termes . . . . .	82
3.1.2 L'exemple des figures géométriques . . . . .	85
3.2 Le modèle Merise . . . . .	88
3.3 Le modèle UML . . . . .	90
4. Le modèle de pilotage . . . . .	94
4.1 Les faits relatifs au projet . . . . .	97
4.2 Les indicateurs clés de performance (KPI) . . . . .	97
4.2.1 Indicateurs relatifs au planning . . . . .	97
4.2.2 Indicateurs relatifs à la qualité . . . . .	98
4.2.3 Indicateurs relatifs à la roadmap . . . . .	98
4.2.4 Indicateurs financiers . . . . .	98
4.2.5 Indicateurs relatifs à l'équipe . . . . .	98
4.2.6 Indicateurs liés au risque . . . . .	99
4.3 Le pilotage du projet . . . . .	99
4.3.1 Décisions sur l'issue du projet . . . . .	100
4.3.2 Décisions sur le contrôle du projet . . . . .	100
4.3.3 Conduite de la communication . . . . .	100
4.3.4 Constitution de la base de connaissances . . . . .	100
5. Prendre en compte le risque . . . . .	100
5.1 L'analyse des risques . . . . .	100

5.1.1	Les classes de risques . . . . .	101
5.1.2	L'identification des risques. . . . .	105
5.1.3	La caractérisation du risque . . . . .	106
5.2	Le plan de risques . . . . .	107
5.2.1	Les stratégies de gestion du risque. . . . .	108
5.2.2	Choisir le modèle de projet en fonction du risque . . . . .	111
6.	Étude du risque pour le projet de CRM . . . . .	112
6.1	Analyse des risques. . . . .	113
6.1.1	La faisabilité du projet . . . . .	113
6.1.2	La disponibilité des ressources . . . . .	114
6.1.3	La dimension personnelle. . . . .	114
6.1.4	Des aspects informatiques et techniques . . . . .	115
6.1.5	Facteurs de réussite du projet . . . . .	116
6.2	Plan de gestion des risques . . . . .	116
6.2.1	L'évaluation des risques . . . . .	117
6.2.2	La conduite du risque . . . . .	119
6.3	Choix des modèles . . . . .	123

**Chapitre 4**

**Démarrer un projet**

1.	Le point de départ . . . . .	125
1.1	L'appel d'offres . . . . .	125
1.1.1	Les documents constitutifs d'un appel d'offres. . . . .	126
1.1.2	Les dispositions réglementaires . . . . .	133
1.2	La demande directe. . . . .	134
2.	Les premiers choix . . . . .	135
2.1	Le cadrage du projet . . . . .	135
2.1.1	Le contexte . . . . .	135
2.1.2	L'objet . . . . .	136
2.1.3	Un exemple de cadrage. . . . .	136
2.2	L'analyse préalable . . . . .	137
2.2.1	Le recueil des informations disponibles. . . . .	138
2.2.2	Conceptualisation de la solution . . . . .	139

# 6 — Conduite de projets informatiques

Développement, analyse et pilotage

2.2.3	La restitution (project master plan) . . . . .	139
2.3	L'organigramme des tâches . . . . .	140
2.3.1	Une approche générale . . . . .	140
2.3.2	Les unités d'œuvre propres au modèle cascade . . . . .	141
2.3.3	Les unités d'œuvre du cycle en V . . . . .	143
2.3.4	Les contributions du modèle itératif . . . . .	144
3.	La structuration du projet . . . . .	145
3.1	Le dimensionnement du projet . . . . .	145
3.1.1	La charge totale en jours.homme . . . . .	146
3.1.2	L'effectif participant au projet . . . . .	146
3.1.3	Les écueils habituels liés au chiffrage . . . . .	147
3.2	La maîtrise d'ouvrage . . . . .	149
3.3	La maîtrise d'œuvre . . . . .	150
3.4	La constitution de l'équipe . . . . .	151
3.4.1	La distribution académique des rôles . . . . .	151
3.4.2	Les configurations à privilégier et celles à éviter . . . . .	153
3.5	L'externalisation des ressources . . . . .	154
3.5.1	L'outsourcing . . . . .	155
3.5.2	La délocalisation (développement offshore) . . . . .	155
3.5.3	La sous-traitance . . . . .	155
3.6	Les équipes transversales . . . . .	156
3.7	Macroplanning . . . . .	156
3.8	L'initialisation (kick-off) . . . . .	157
3.9	La plate-forme technique . . . . .	158
4.	Démarrer le projet de pilotage d'activité . . . . .	158
4.1	Cadrage du projet . . . . .	159
4.1.1	Contexte . . . . .	159
4.1.2	Objet . . . . .	159
4.2	Structuration . . . . .	159
4.2.1	Expression de besoins . . . . .	159
4.2.2	Organigramme des tâches (WBS) et estimation des charges . . . . .	162
4.2.3	Dimensionnement . . . . .	165

- 4.3 Analyse des risques . . . . . 166
- 4.4 Choix des modèles . . . . . 167
- 4.5 Macroplanning et initialisation . . . . . 168

**Chapitre 5**  
**L'analyse**

- 1. Le domaine métier . . . . . 171
  - 1.1 Les bases de données relationnelles . . . . . 171
    - 1.1.1 Les tables . . . . . 172
    - 1.1.2 Les clés et les relations . . . . . 173
    - 1.1.3 Les vues et les requêtes . . . . . 174
    - 1.1.4 Les traitements et les procédures stockées . . . . . 175
  - 1.2 Les autres types de bases de données . . . . . 176
    - 1.2.1 Les bases de données hiérarchiques (XML) . . . . . 176
    - 1.2.2 Les bases de données objets . . . . . 179
    - 1.2.3 Les bases de données semi-structurées . . . . . 180
  - 1.3 Le modèle d'analyse Merise . . . . . 181
    - 1.3.1 Les modèles de délimitation du système . . . . . 181
    - 1.3.2 Le modèle conceptuel de données MCD . . . . . 182
    - 1.3.3 Le modèle logique de données MLD . . . . . 184
    - 1.3.4 Le modèle physique de données MPD . . . . . 185
  - 1.4 Les outils de modélisation . . . . . 188
- 2. Le périmètre fonctionnel . . . . . 188
  - 2.1 Les processus et les flux de travail (workflows) . . . . . 189
  - 2.2 La cartographie fonctionnelle . . . . . 189
  - 2.3 Le modèle d'analyse UML . . . . . 190
    - 2.3.1 Les modèles de délimitation du système . . . . . 190
    - 2.3.2 Les modèles associés aux diagrammes de classes . . . . . 196
    - 2.3.3 Les modèles de description logique et technique . . . . . 201
  - 2.4 Les outils de modélisation . . . . . 203
- 3. Le contexte technique en phase d'analyse . . . . . 203
  - 3.1 Les outils de recherche d'information . . . . . 204
    - 3.1.1 Le langage SQL . . . . . 204

# 8 — Conduite de projets informatiques

Développement, analyse et pilotage

3.1.2	Les langages LINQ et HQL .....	206
3.1.3	Les langages XPath et XQuery .....	210
3.2	Les outils de structuration d'application (design pattern) .....	211
3.2.1	Motifs de structuration logique .....	211
3.2.2	Motifs de structuration technique .....	215
4.	La documentation de projet .....	215
4.1	Le cahier des charges .....	215
4.1.1	Le contenu d'un cahier des charges .....	216
4.1.2	Élaborer un cahier des charges .....	219
4.2	Les référentiels de spécifications .....	220
4.2.1	Le dossier d'intégration fonctionnelle (DIF) .....	220
4.2.2	Le dossier d'intégration technique (DIT) .....	220
4.2.3	Les spécifications détaillées .....	221
4.2.4	Les normes de développement .....	221
5.	L'analyse du projet de pilotage d'activité .....	222
5.1	Le dossier d'intégration fonctionnelle .....	222
5.2	Le dossier d'intégration technique .....	224

## Chapitre 6

### Les référentiels techniques

1.	Les problématiques d'intégration .....	229
1.1	Les développements spécifiques .....	229
1.2	Les progiciels et l'intégration d'applications d'entreprises (EAI) .....	231
1.2.1	Le modèle fonctionnel .....	232
1.2.2	Le modèle technique .....	233
2.	Les architectures distribuées .....	233
2.1	Le découpage en couches .....	233
2.2	Les composants distribués .....	235
3.	Les architectures orientées services (SOA) .....	238
3.1	Les middlewares orientés message (MOM) .....	238
3.2	Les services web .....	240

- 3.3 Les services web AJAX et le modèle REST . . . . . 241
- 4. Les plates-formes types . . . . . 244
  - 4.1 Les généralistes J2EE et .NET. . . . . 244
    - 4.1.1 Java et J2EE . . . . . 244
    - 4.1.2 Windows et .NET. . . . . 248
  - 4.2 Les serveurs de bases de données SQL. . . . . 251
  - 4.3 Les environnements décisionnels et les infocentres . . . . . 252
  - 4.4 Les serveurs intégrés ERP, CRM, CMS . . . . . 253
  - 4.5 Les serveurs d'intégration (EAI). . . . . 254
  - 4.6 Les serveurs virtuels . . . . . 256
- 5. De l'ASP au cloud computing . . . . . 257
  - 5.1 L'accès locatif aux applications . . . . . 257
  - 5.2 Le client mobile : du portable à la tablette . . . . . 258
  - 5.3 L'infrastructure à la demande . . . . . 260

**Chapitre 7**  
**Le développement du projet**

- 1. La maîtrise de la qualité du code . . . . . 263
  - 1.1 Les normes de codage . . . . . 263
    - 1.1.1 L'indentation. . . . . 263
    - 1.1.2 Les noms dans les programmes . . . . . 264
    - 1.1.3 La notion de modularité. . . . . 266
  - 1.2 La gestion de la documentation. . . . . 266
    - 1.2.1 Les commentaires . . . . . 266
    - 1.2.2 La documentation générée. . . . . 267
    - 1.2.3 Les régions. . . . . 269
  - 1.3 Les revues de code . . . . . 271
    - 1.3.1 Définir les règles de codage . . . . . 271
    - 1.3.2 Utiliser des outils de contrôle durant le codage . . . . . 271
    - 1.3.3 Préparer une revue de code. . . . . 273
    - 1.3.4 Effectuer une revue. . . . . 273
  - 1.4 Les outils d'analyse et de couverture . . . . . 273

# 10 — Conduite de projets informatiques

Développement, analyse et pilotage

1.5	Les référentiels de code source . . . . .	275
1.5.1	Installation du serveur et création d'un référentiel . . . . .	276
1.5.2	Création d'un projet lié au référentiel . . . . .	279
1.5.3	Extraction et archivage de fichiers . . . . .	281
1.5.4	Gestion des historiques . . . . .	283
1.5.5	Opérations avancées . . . . .	284
2.	La gestion des versions . . . . .	284
2.1	Production d'une version . . . . .	284
2.1.1	Les numéros de version . . . . .	284
2.1.2	Les release notes . . . . .	286
2.2	Montée de version . . . . .	286
2.3	L'intégration continue . . . . .	286
3.	Les tests . . . . .	287
3.1	Tests unitaires . . . . .	287
3.2	Tests d'intégration . . . . .	288
3.3	Tests fonctionnels . . . . .	289
3.3.1	Boîte noire . . . . .	289
3.3.2	Boîte blanche . . . . .	290
3.4	Le banc d'essais (benchmark) . . . . .	291
3.4.1	Tests de charge et de performances . . . . .	291
3.4.2	Tests de stress et de montée en charge . . . . .	291
3.4.3	Tests d'endurance . . . . .	292
3.4.4	Tests aux limites . . . . .	292
3.5	Les méthodologies de test . . . . .	292
3.5.1	Cycles de test . . . . .	292
3.5.2	Le plan de tests . . . . .	294
3.5.3	Stratégie de tests descendants . . . . .	295
3.5.4	Stratégie de tests ascendants . . . . .	296
3.5.5	Tests de non-régression . . . . .	296
3.6	L'utilisation d'un logiciel de suivi, activité et bug tracking . . . . .	297
3.6.1	Installation et paramétrage d'un outil . . . . .	297
3.6.2	Création de projets et de versions . . . . .	298
3.6.3	Enregistrement de défauts (bogues) . . . . .	299

- 3.6.4 Établissement de rapports ..... 300
- 4. L'industrialisation. .... 301
  - 4.1 La configuration ..... 301
  - 4.2 L'installation ou setup ..... 302
  - 4.3 Les tests de déploiement et de mise à jour ..... 303
- 5. Le développement du projet de pilotage d'activité. .... 304
  - 5.1 Organisation du développement ..... 304
    - 5.1.1 La gestion du code source et des éléments de travail ... 304
    - 5.1.2 La documentation. .... 305
    - 5.1.3 Les revues ..... 305
    - 5.1.4 La production des versions. .... 305
    - 5.1.5 La mesure de l'avancement du développement ..... 306
  - 5.2 Organisation des tests ..... 306
    - 5.2.1 La préparation des scénarios de tests. .... 306
    - 5.2.2 Les tests automatisés ..... 307
    - 5.2.3 La non-régression ..... 308
    - 5.2.4 Les jeux d'essais. .... 308
    - 5.2.5 La planification des tests ..... 308
    - 5.2.6 La gestion des anomalies ..... 310
    - 5.2.7 Les statistiques utiles ..... 310
  - 5.3 Le déploiement ..... 312
    - 5.3.1 La préparation des environnements ..... 312
    - 5.3.2 Les procédures de déploiement ..... 313
    - 5.3.3 Les mises à jour ..... 314

**Chapitre 8**  
**La planification et le chiffrage**

- 1. L'estimation des charges ..... 315
  - 1.1 Les charges et les délais. .... 316
  - 1.2 L'estimation par abaque ..... 316
  - 1.3 L'estimation analytique ..... 317
  - 1.4 L'estimation imposée ..... 318

# 12 — Conduite de projets informatiques

Développement, analyse et pilotage

2.	L'emploi du temps du chef de projet . . . . .	319
2.1	Les charges d'encadrement . . . . .	319
2.2	Les tâches d'organisation . . . . .	322
3.	La gestion des ressources . . . . .	324
3.1	Le plan de charge . . . . .	324
3.2	La montée en charge et la disponibilité . . . . .	325
3.3	La surcharge . . . . .	326
4.	La planification . . . . .	327
4.1	Les éléments d'un planning . . . . .	327
4.1.1	Les unités d'œuvre et les tâches . . . . .	327
4.1.2	Les jalons (points de phase) . . . . .	329
4.1.3	Les dépendances entre les tâches et les contraintes . . . . .	329
4.1.4	Les ressources . . . . .	330
4.1.5	Le calendrier . . . . .	331
4.2	La création du planning . . . . .	332
4.2.1	Le recensement des tâches . . . . .	332
4.2.2	Le recensement des ressources . . . . .	333
4.2.3	La définition de l'horizon temporel . . . . .	333
4.2.4	L'identification du plan de charge . . . . .	334
4.2.5	La constitution du planning à partir du plan de développement . . . . .	335
4.3	La représentation des plannings . . . . .	335
4.3.1	Le diagramme de Gantt . . . . .	335
4.3.2	Le diagramme PERT . . . . .	337
4.3.3	Le chemin critique . . . . .	337
4.4	Les outils de planification . . . . .	338
5.	La planification du projet de pilotage d'activité . . . . .	338
5.1	Différentes estimations . . . . .	339
5.1.1	Estimation selon la méthode des points fonctionnels . . . . .	339
5.1.2	Estimation par abaque . . . . .	339
5.1.3	Estimation ad hoc . . . . .	342
5.2	Le planning du projet . . . . .	343
5.3	Analyse des périodes critiques . . . . .	346

**Chapitre 9**  
**Le suivi et le pilotage**

- 1. Le suivi de projet . . . . . 347
  - 1.1 Le compte rendu d'activité . . . . . 347
  - 1.2 L'avancement des tâches . . . . . 348
  - 1.3 L'évolution du planning . . . . . 350
  - 1.4 Les réunions de projet . . . . . 352
    - 1.4.1 Les réunions d'équipe . . . . . 352
    - 1.4.2 Les sessions de spécification ou d'analyse . . . . . 352
    - 1.4.3 Les séances de briefing et de formation . . . . . 353
    - 1.4.4 Les coups d'envoi (kick-off) . . . . . 353
  - 1.5 La centralisation de la documentation . . . . . 354
- 2. La gestion des imprévus . . . . . 355
  - 2.1 Les complications techniques . . . . . 355
    - 2.1.1 Détection et diagnostic . . . . . 356
    - 2.1.2 Les causes possibles . . . . . 356
    - 2.1.3 Les solutions . . . . . 357
    - 2.1.4 La base de connaissances . . . . . 358
  - 2.2 Les conflits . . . . . 358
    - 2.2.1 Les types de conflits . . . . . 358
    - 2.2.2 Les techniques de résolution . . . . . 359
  - 2.3 Les aléas organisationnels . . . . . 360
    - 2.3.1 Retards . . . . . 360
    - 2.3.2 Qualité médiocre des livraisons . . . . . 361
    - 2.3.3 Communication défaillante . . . . . 362
- 3. Le comité de pilotage . . . . . 362
  - 3.1 Les règles de pilotage . . . . . 362
  - 3.2 Les réunions de pilotage . . . . . 363
  - 3.3 Les modifications de planning . . . . . 364
  - 3.4 L'allocation de ressources . . . . . 365
  - 3.5 L'interruption ou l'arrêt d'un projet . . . . . 365

# 14 — Conduite de projets informatiques

Développement, analyse et pilotage

4. Terminer un projet . . . . .	366
4.1 La livraison . . . . .	366
4.1.1 Les livrables . . . . .	366
4.1.2 Les modalités de livraison . . . . .	366
4.2 La réception et la recette . . . . .	367
4.3 La gestion du changement . . . . .	367
4.3.1 Le pilote . . . . .	367
4.3.2 La formation . . . . .	368
4.3.3 Le support . . . . .	368
4.4 La maintenance . . . . .	369
5. Suivi du projet de pilotage d'activité . . . . .	370
5.1 Organisation et suivi des réunions de projet . . . . .	370
5.1.1 Les réunions daily standup meetings . . . . .	370
5.1.2 Les réunions d'avancement . . . . .	371
5.2 Quelques imprévus . . . . .	374
5.2.1 Les réunions de 30 minutes s'éternisent . . . . .	374
5.2.2 Démarrage trop tardif de tâches sans alerte . . . . .	375
5.2.3 La charge des testeurs en dent de scie . . . . .	376
5.3 Suivi du projet par le comité de pilotage . . . . .	377
5.3.1 Quand le projet vire à l'orange . . . . .	377
5.3.2 La malédiction des données de production . . . . .	377
5.4 Le go live . . . . .	379
5.4.1 Le processus de lancement . . . . .	378
5.4.2 L'adhésion au nouvel outil . . . . .	379
5.4.3 Le bilan . . . . .	380
Glossaire . . . . .	381
Index . . . . .	387

## Chapitre 3

# La prise en compte du risque

### 1. Les trois axes

Le chef de projet en devenir est parfois perplexe face au nombre des éléments qu'il doit organiser pour aboutir. Dans quelle direction partir ? Par quoi commencer et comment continuer ?

Pour répondre à ces questions, il faut d'abord se représenter le processus du projet tel que nous l'avons décrit au chapitre Un projet informatique. Cette fois-ci, nous choisissons un espace structuré par trois axes, le temps (cycle de vie), l'analyse et le pilotage.

Chaque dimension de cet espace correspond elle-même à une méthode plus ou moins standard, et qui donnera lieu à un processus que le chef de projet devra suivre.



Sur le dernier axe sont représentés les aléas, les points de contrôles, les prises de décision... Il s'agit de l'axe de pilotage (cycle de décision). Par rapport aux deux autres axes, c'est celui qui réserve le plus de "surprises", mais il ne faudrait pas non plus le laisser totalement libre. Des méthodes se sont forgées au fil des projets pour organiser le pilotage de manière méthodique et efficace.

Ainsi le "secret" du démarrage d'un projet résiderait-il dans le choix des méthodes associées à chaque dimension ? Ce n'est pas tout à fait suffisant, car il faut encore s'assurer d'un mélange harmonieux de ces trois composantes, d'une communication efficace, bref d'un sens pratique sans lequel les meilleures recommandations du monde resteront vaines. Autrement dit, l'application isolée, cloisonnée, d'une méthode sur un seul axe ne donne pas de meilleurs résultats qu'un projet dans lequel on oublierait l'existence des autres axes.

Le tableau suivant donne un aperçu du résultat lorsque les méthodes sont appliquées seules (indépendamment les unes des autres) ou lorsqu'elles font défaut :

Méthode/axe	Application seule ou indépendante	Non application (absence)
Développement	En général, la méthode d'analyse est censurée ou ne donne aucun résultat probant.	Non-conformité (l'expression des besoins est prise en compte tardivement). Dérive du projet (difficulté à stabiliser la solution par manque de tests).
Analyse	Vision beaucoup trop conceptuelle ; le projet devient techniquement risqué car les tests d'intégration sont oubliés.	Non-conformité (focus sur des fonctions mineures, alors que les fonctions majeures manquent). Code non optimal entraînant des difficultés à le faire évoluer (pas de factorisation).

# 70 — Conduite de projets informatiques

Développement, analyse et pilotage

Méthode/axe	Application seule ou indépendante	Non application (absence)
Pilotage	Inadéquation du pilotage par rapport aux caractéristiques du projet.	Non tenue des objectifs, manque de contrôle. Explosion de l'équipe projet en cas de montée en charge. Conduite du changement inexistante : rejet de l'application livrée.

Ce tableau récapitule donc six constructions (antipatterns) qu'il convient absolument d'éviter. Et pourtant, les occasions de se tromper ne manquent pas, soit parce que les habitudes ont la vie dure, soit parce qu'une nouvelle méthode n'est pas toujours à même de s'intégrer facilement dans le paysage d'un référentiel de conduite de projets.

## ■ Remarque

*Les exemples d'achoppement sont légion. Le chef de projet doit toujours garder en tête que les trois composantes sont indispensables, et qu'elles doivent coopérer pour être efficaces.*

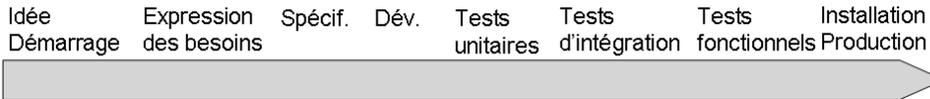
## 2. Le modèle de développement

Le modèle de développement constitue l'un des premiers choix que doit effectuer le chef de projet. Il façonne de manière explicite l'organisation temporelle du projet et a une grande influence sur le résultat final.

Il n'y a pas de bon et de mauvais modèle, du moins parmi les grandes classes dont l'énumération suit. Cependant, chaque projet a des caractéristiques qui rendent l'application d'un modèle judicieuse ou au contraire inefficace.

### 2.1 Le modèle cascade

On l'appelle aussi modèle linéaire ou modèle nominal. C'est certainement le modèle le plus simple, ce qui ne le déprécie pas pour autant !



Dans ce modèle chaque étape suit la précédente, sans autre nécessité que "d'attendre" son issue ; dès que l'expression des besoins est terminée, les spécifications sont écrites. Au terme de ce travail, le développement est réalisé jusqu'à son achèvement. Les tests unitaires s'enchaînent, et ainsi de suite.

Avant d'expliquer l'appellation cascade, intéressons-nous aux différentes étapes prévues par ce modèle.

Idée - démarrage	C'est le point de départ du projet, le moment où les conditions matérielles sont remplies, et où la vision du projet est partagée par l'équipe.
Expression des besoins	Après la décision d'enclencher un projet, on refait le tour des utilisateurs pour recueillir plus en détail leurs exigences fonctionnelles. Ces requêtes sont consignées dans un classeur (papier ou tableur), identifiées, et finalement regroupées ou recoupées.
Spécifications	L'ensemble des requêtes est considéré selon différentes approches dictées par le modèle d'analyse : importance, priorité, difficulté technique, risques... Il ressort des spécifications un découpage technico-fonctionnel de l'application.
Développement	Les spécifications sont traduites en code, en projetant le découpage modulaire (composants) sur la plate-forme choisie par l'architecte.

# 72 — Conduite de projets informatiques

Développement, analyse et pilotage

Tests unitaires	Chaque composant est testé indépendamment des autres. Son comportement doit être conforme aux spécifications, dans la plage de fonctionnement prévue.
Tests d'intégration	Les composants sont le plus souvent assemblés sur une plate-forme techniquement hétérogène. La lecture d'enregistrements SQL depuis du code Java ou .NET peut provoquer des problèmes de sécurité, de charge, de robustesse (gestion des valeurs NULL par exemple...). Les tests d'intégration s'assurent que l'édifice final est stable dans toutes les occasions prévues par le cahier des charges.
Tests fonctionnels	L'ensemble des processus couverts par l'application est passé au crible. Ils doivent respecter la méthodologie retenue pour l'utilisateur, répondre justement et dans des délais acceptables.
Installation	L'application est déployée sur sa plate-forme définitive (cette étape peut être précédée d'une phase d'industrialisation pour les logiciels packagés). Des ajustements de sécurité sont parfois nécessaires, des chaînes de connexion SQL sont à redéfinir, etc.

Ce modèle est nominal ; chacun est libre d'y ajouter d'autres étapes, ou de le remanier, c'est une base de travail. Il est aussi linéaire, aucune étape n'est traitée parallèlement à une autre.

D'où vient l'appellation cascade ? La remontée est difficile, comme pour des poissons qui cherchent à gagner les sources d'une rivière. Les écluses, barrages, ou cascades, sont difficilement franchissables. Dans le modèle cascade, un coût unitaire "à la descente" est décuplé s'il faut faire machine arrière. Autrement dit, une erreur de conception se paye de manière exponentielle ! Quelque chose qui a oublié d'être spécifié coûtera 10 fois plus à développer. Si l'oubli est réparé en phases de tests unitaires, c'est 100 fois plus...