# Méthodes Agiles

## 4 – Agile Academy

# Argument

- L'évolution accélérée des exigences métier et logiciel imposent des pratiques de développement plus réactives/agiles, sans compromis de qualité
- Après plusieurs séances illustrant des pratiques agiles individuelles, il est temps de les intégrer
- Aujourd'hui, nous illustrerons plusieurs itérations d'un cycle de vie agile à travers quelques pratiques [xp] intégrées

# Agenda

I. Rappels agiles
II. Capture de "user stories" et cycle de négociation
III. Conception légère et développement itératif
IV. Tests unitaires et couverture du code
V. Propriété collective du code et gestion de configuration
VI. Métriques de stabilité et "refactoring" piloté / modèle

TP3 : Scénario agile intégré – « direct live / prime time »

# I. Rappels agiles

*De la niche au château*

# [Kent Beck, IEEE Computer' 99]

## XP Practices

Here is a quick summary of each of the major practices in XP.

***Planning game***. Customers decide the scope and timing of releases based on estimates provided by programmers. Programmers implement only the functionality demanded by the stories in this iteration.

***Small releases***. The system is put into production in a few months, before solving the whole problem. New releases are made often—anywhere from daily to monthly.

***Metaphor***. The shape of the system is defined by a metaphor or set of metaphors shared between the customer and programmers.

***Simple design***. At every moment, the design runs all the tests, communicates everything the programmers want to communicate, contains no duplicate code, and has the fewest possible classes and methods. This rule can be summarized as, "Say everything once and only once."

***Tests***. Programmers write unit tests minute by minute. These tests are collected and they must all run correctly. Customers write functional tests for the stories in an iteration. These tests should also all run, although practically speaking, sometimes a business decision must be made comparing the cost of shipping a known defect and the cost of delay.

***Refactoring***. The design of the system is evolved through transformations of the existing design that keep all the tests running.

***Pair programming***. All production code is written by two people at one screen/keyboard/mouse.

***Continuous integration***. New code is integrated with the current system after no more than a few hours. When integrating, the system is built from scratch and all tests must pass or the changes are discarded.

***Collective ownership***. Every programmer improves any code anywhere in the system at any time if they see the opportunity.

***On-site customer***. A customer sits with the team full-time.

***40-hour weeks***. No one can work a second consecutive week of overtime. Even isolated overtime used too frequently is a sign of deeper problems that must be addressed.

***Open workspace***. The team works in a large room with small cubicles around the periphery. Pair programmers work on computers set up in the center.

***Just rules***. By being part of an Extreme team, you sign up to follow the rules. But they're just the rules. The team can change the rules at any time as long as they agree on how they will assess the effects of the change.

- Réactions typiques

  A. **Sceptique** : « *ça marchera pô [chez nous]* »

  B. **Curieux** : « *ça pourrait marcher, mais quel effort pour quel gain ?* »

  C. **Convaincu** : « *ok, mais comment [faire] adopter ?* »

- Et vous ?

# RUP vs. XP

- Vision Rational [G. Booch]
  - Si le client veut une niche pour son chien,

    faites-la lui de façon ad-hoc : marteau, clous, bois, …
  - Si le client veut un château,

    présentez-lui un plan, planifiez les travaux, réservez les ressources, etc.
- Vision XP
  - Si le client veut un château,

    commencez malgré tout par une … niche,

    puis faites-la évoluer progressivement vers un château.

# XP

- Conséquence
  - Ceci réclame donc une organisation spéciale, capable de ne pas dégrader la cohérence, la stabilité et les performances pendant l'évolution
  - Ainsi que des outils adaptés
- Principes
  - La chose la plus simple qui puisse marcher
  - Pilotage par la valeur ajoutée (ROI++)
  - Feedback très, mais très court
  - Stabilité permanente et transparente
- Suite
  - Pratique / sondage / risques / outillage / démo / vote
  - Quels risques : identifiés, adressés, nouveaux ?

# II. Capture de "user stories" et cycle de négociation

# Exigences agiles

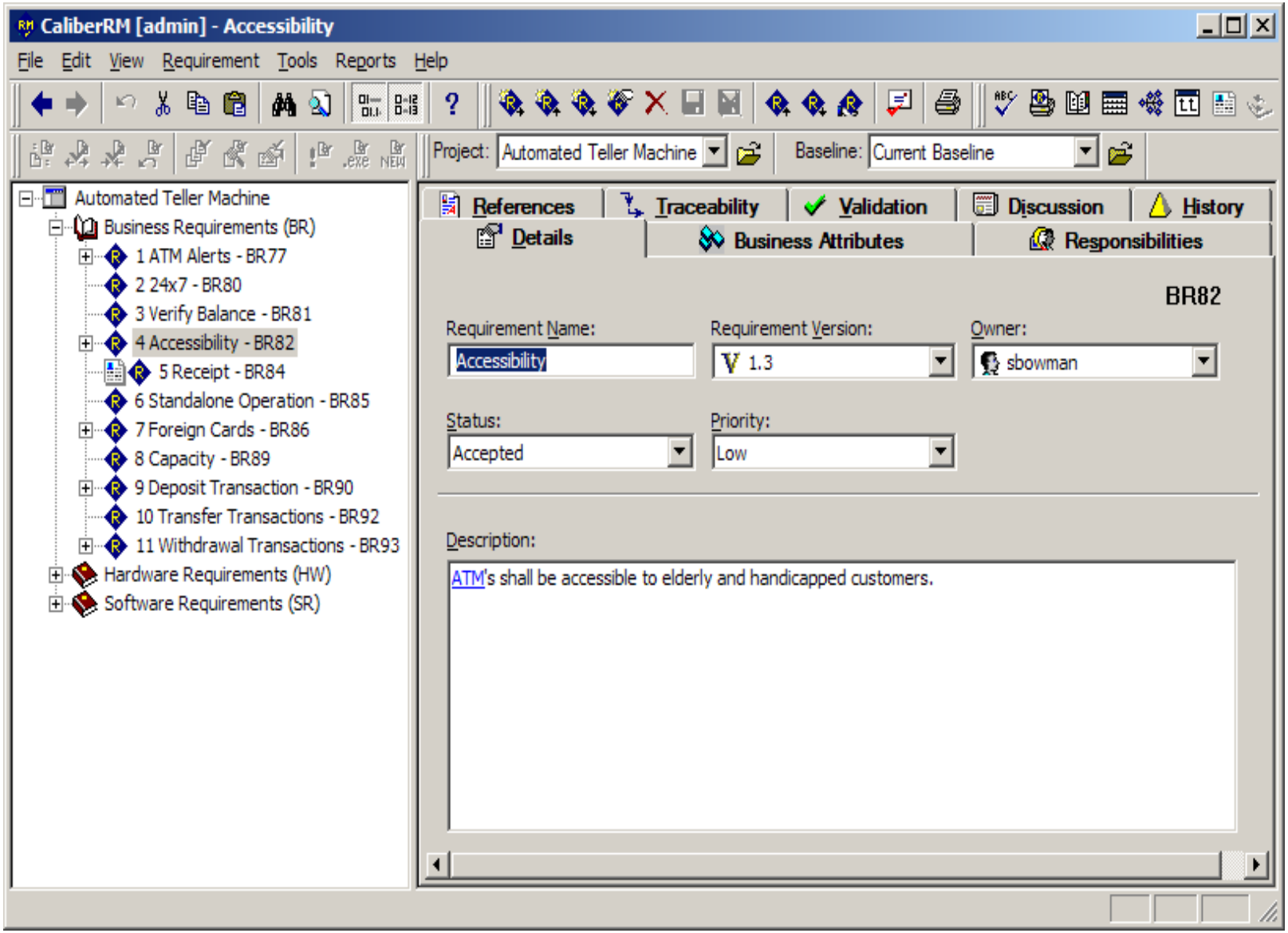
*Planning game*. Customers decide the scope and timing of releases based on estimates provided by programmers. Programmers implement only the functionality demanded by the stories in this iteration.

*Metaphor*. The shape of the system is defined by a metaphor or set of metaphors shared between the customer and programmers.

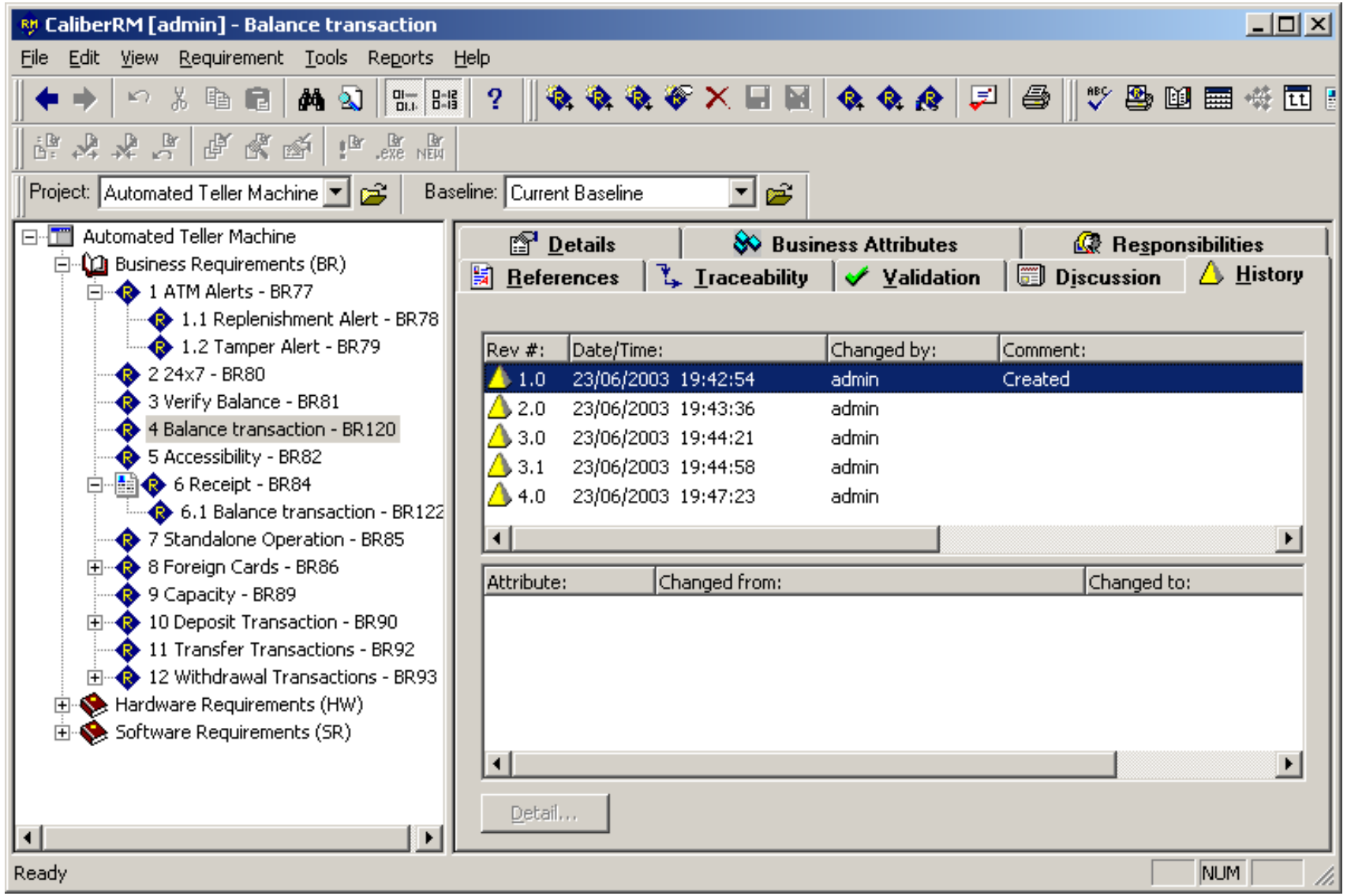
*Tests*. Programmers write unit tests minute by minute. These tests are collected and they must all run correctly. Customers write functional tests for the stories in an iteration. These tests should also all run, although practically speaking, sometimes a business decision must be made comparing the cost of shipping a known defect and the cost of delay.

*On-site customer*. A customer sits with the team full-time.

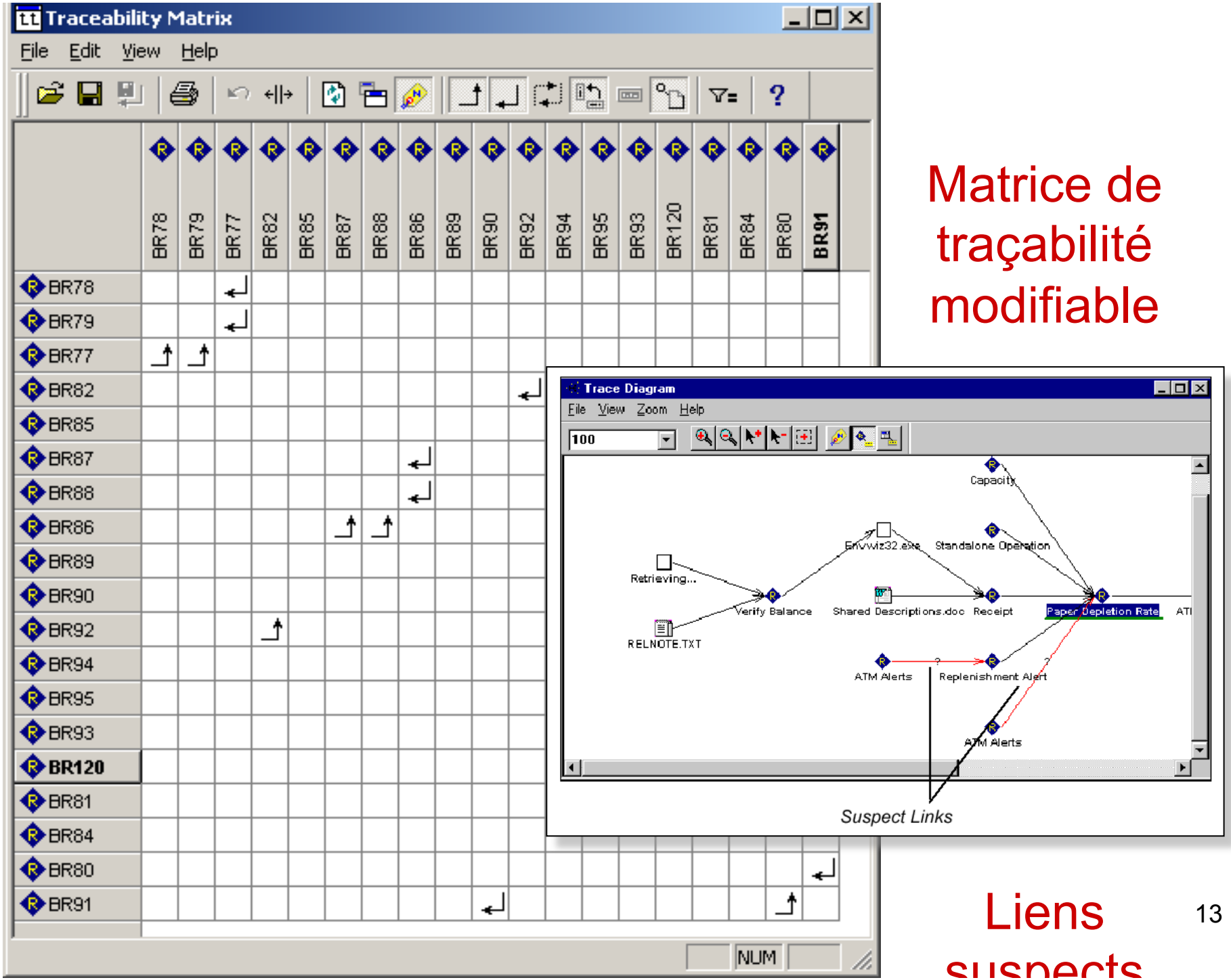
Risques : Trop vague → Sous-entendus → Malentendus → Anomalies → Temps perdu → Argent manqué → Chômage → Crise …




- Projet, types d'exigences, exigences, attributs personnalisés
- Exigence : nom, révision, propriétaire, statut, priorité, id, description




# Historique du changement d'une exigence




Matrice de traçabilité modifiable

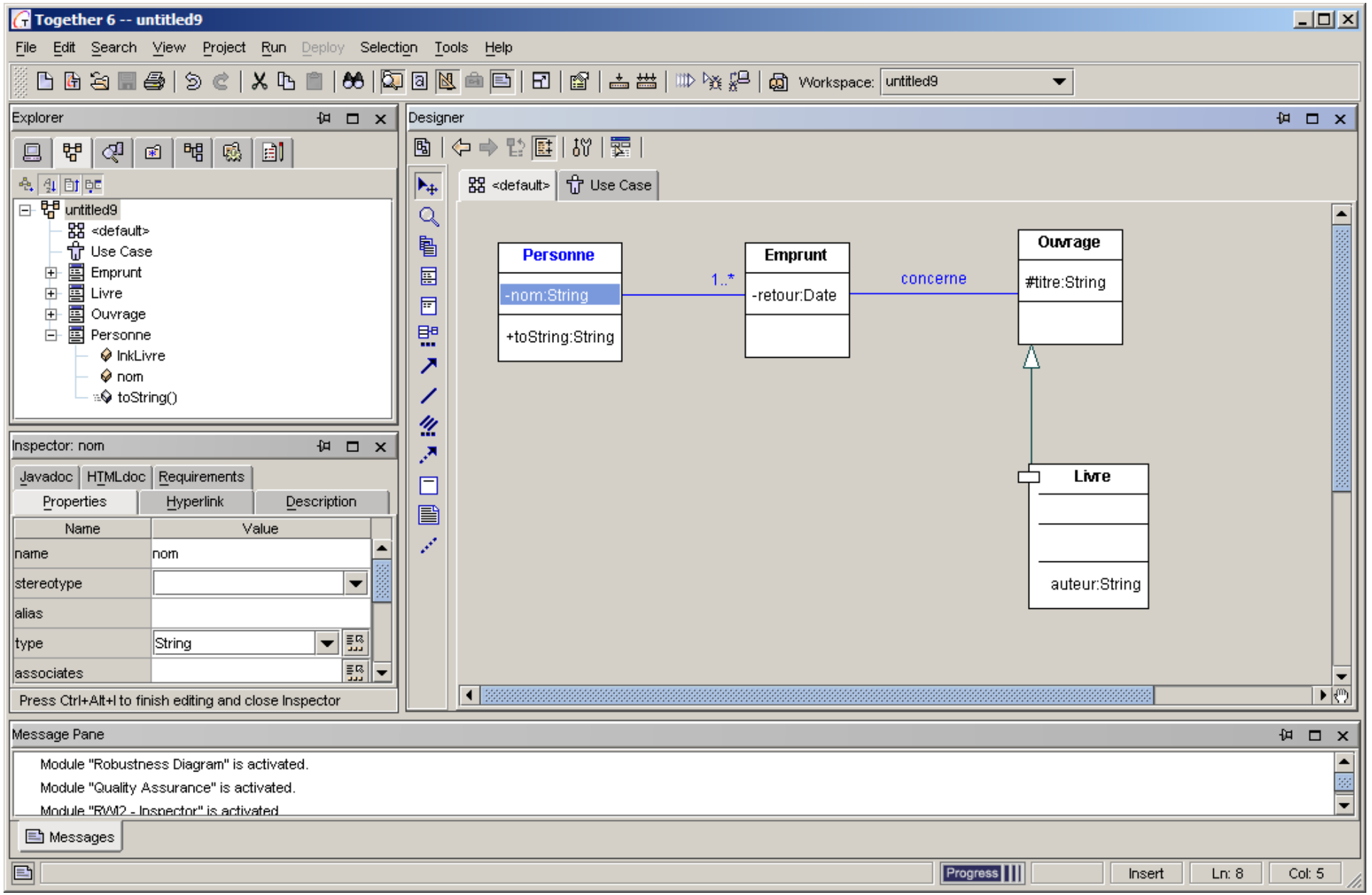
Liens suspects

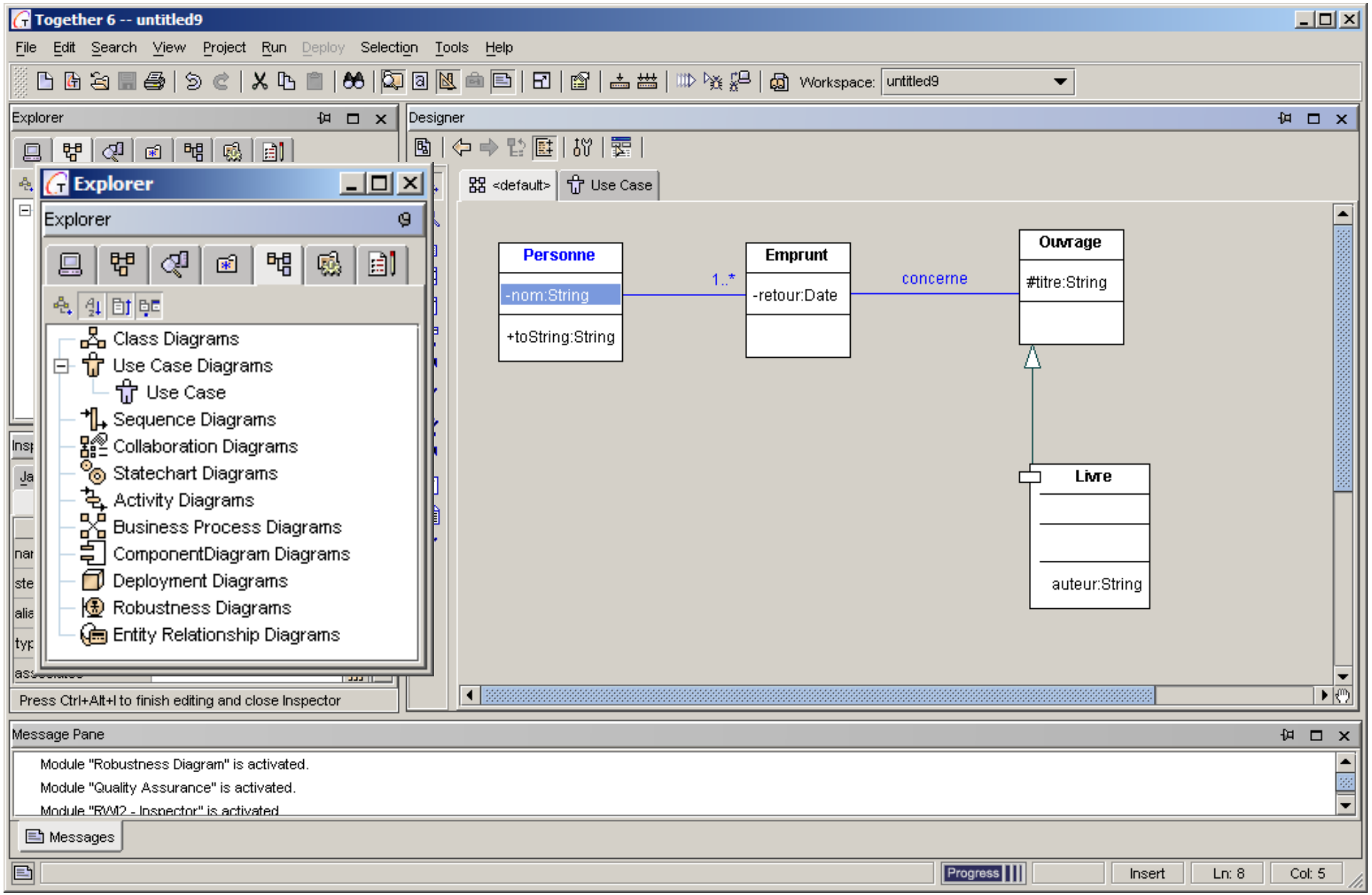
# III. Conception légère et développement itératif

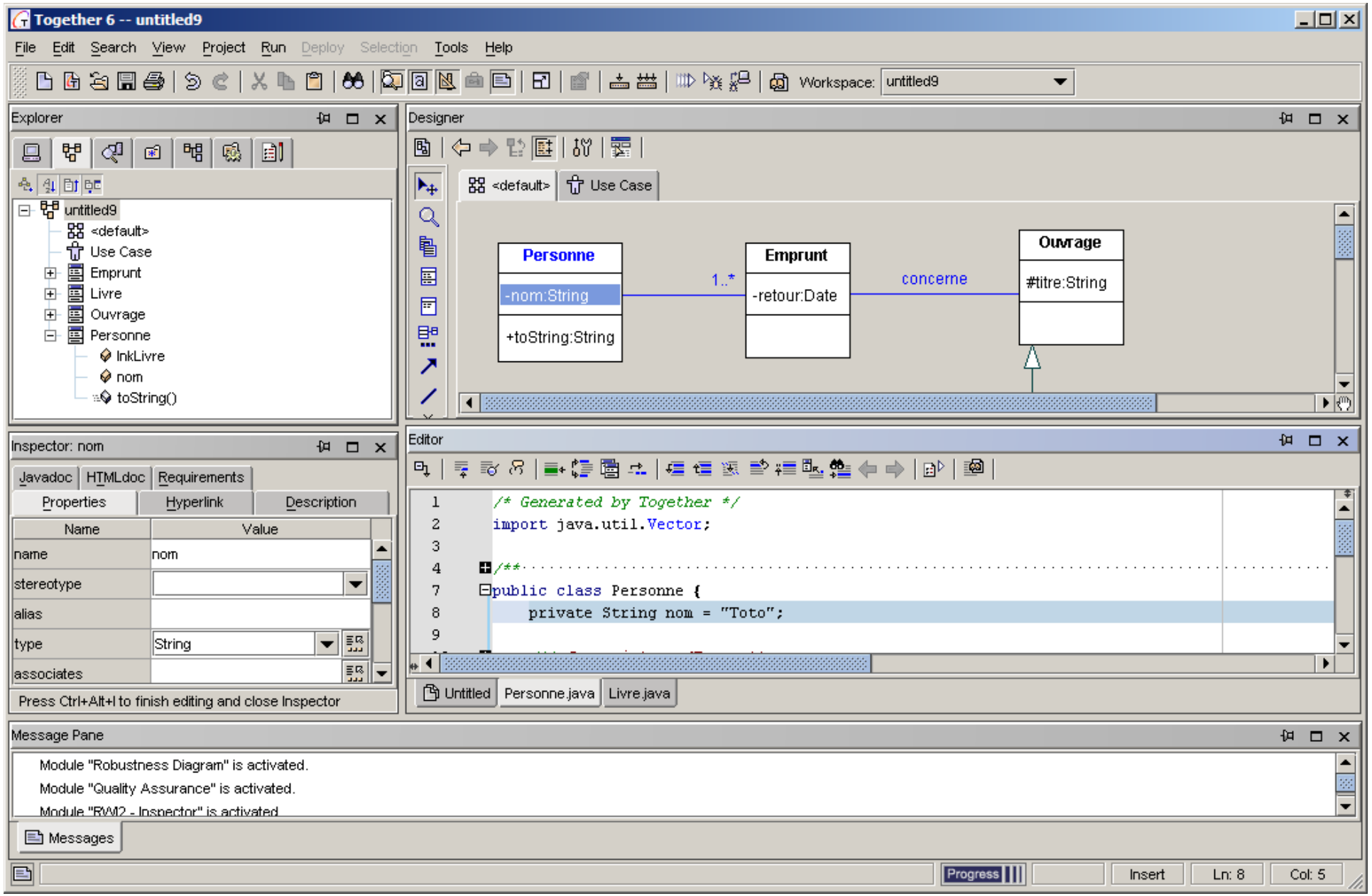
*Metaphor*. The shape of the system is defined by a metaphor or set of metaphors shared between the customer and programmers.

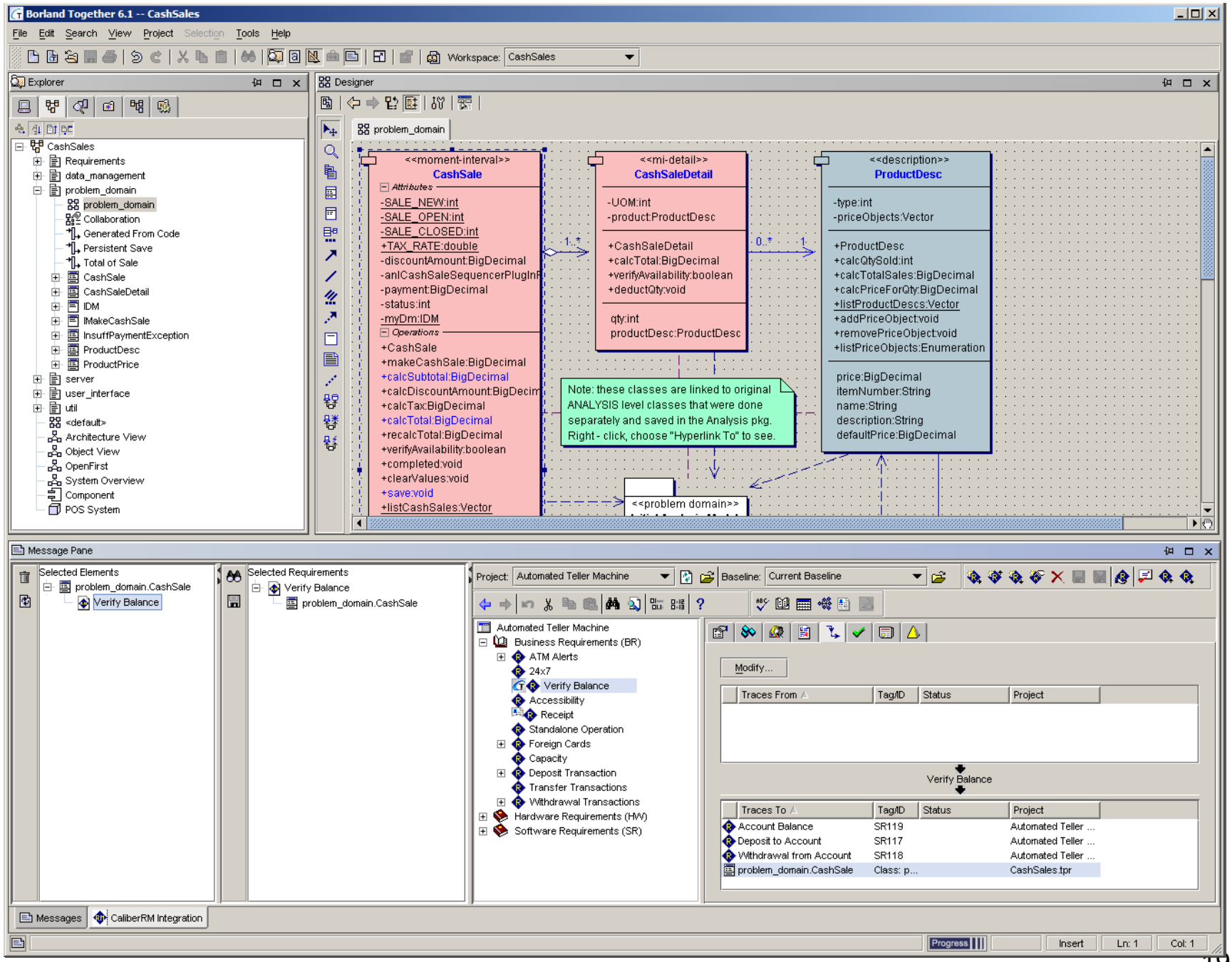
*Simple design*. At every moment, the design runs all the tests, communicates everything the programmers want to communicate. contains no duplicate code, and has the fewest possible classes and methods. This rule can be summarized as, "Say everything once and only once."

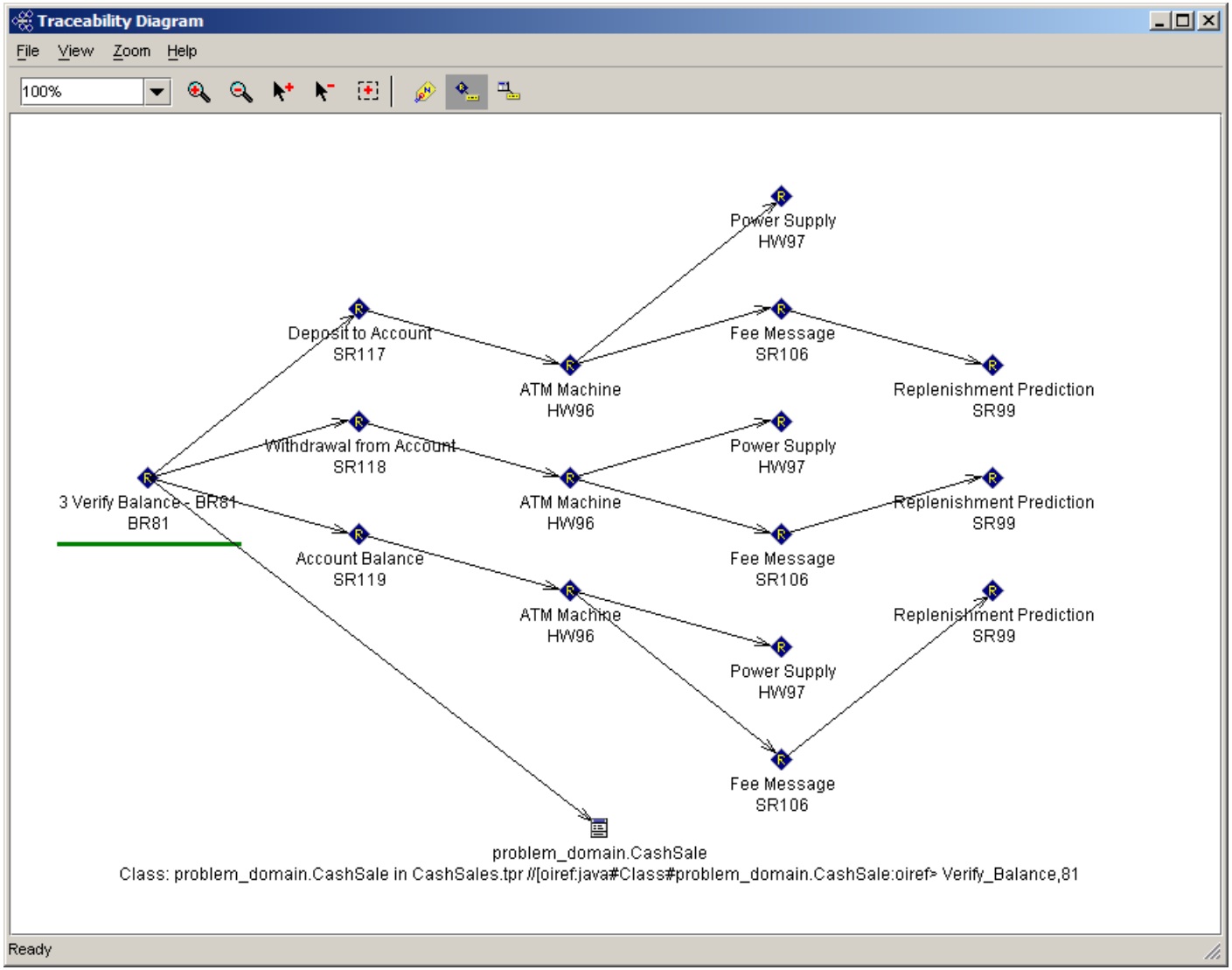
- Risques
  - Lien US-code
  - Rupture de cycle


Together 6 -- untitled9
File Edit Search View Project Run Deploy Selection Tools Help
Workspace: untitled9
Explorer
untitled9
<default>
Use Case
Emprunt
Livre
Ouvrage
Personne
lnkLivre
nom
toString()
Inspector: nom
Javadoc HTMLdoc Requirements
Properties Hyperlink Description
Name Value
name nom
stereotype
alias
type String
associates
Press Ctrl+Alt+I to finish editing and close Inspector
Designer
<default> Use Case
Personne
-nom:String
+toString:String
1..*
Emprunt
-retour:Date
concerne
Ouvrage
#titre:String
Livre
auteur:String
Message Pane
Module "Robustness Diagram" is activated.
Module "Quality Assurance" is activated.
Module "RWI2 - Inspector" is activated
Messages
Progress
Insert
Ln: 8
Col: 5


Together 6 -- untitled9
File Edit Search View Project Run Deploy Selection Tools Help
Workspace: untitled9
Explorer
Designer
Explorer
Class Diagrams
Use Case Diagrams
Use Case
Sequence Diagrams
Collaboration Diagrams
Statechart Diagrams
Activity Diagrams
Business Process Diagrams
ComponentDiagram Diagrams
Deployment Diagrams
Robustness Diagrams
Entity Relationship Diagrams
<default>
Use Case
Personne
-nom:String
+toString:String
1..*
Emprunt
-retour:Date
concerne
Ouvrage
#titre:String
Livre
auteur:String
Press Ctrl+Alt+I to finish editing and close Inspector
Message Pane
Module "Robustness Diagram" is activated.
Module "Quality Assurance" is activated.
Module "RWI2 - Inspector" is activated
Messages
Progress
Insert
Ln: 8
Col: 5


Together 6 -- untitled9
File Edit Search View Project Run Deploy Selection Tools Help
Workspace: untitled9
Explorer
untitled9
<default>
Use Case
Emprunt
Livre
Ouvrage
Personne
lnkLivre
nom
toString()
Designer
<default>
Use Case
Personne
-nom:String
+toString:String
1..*
Emprunt
-retour:Date
concerne
Ouvrage
#titre:String
Inspector: nom
Javadoc HTMLdoc Requirements
Properties Hyperlink Description
Name Value
name nom
stereotype
alias
type String
associates
Press Ctrl+Alt+I to finish editing and close Inspector
Editor
1 /* Generated by Together */
2 import java.util.Vector;
3
4 /**
7 public class Personne {
8 private String nom = "Toto";
9
Untitled
Personne.java
Livre.java
Message Pane
Module "Robustness Diagram" is activated.
Module "Quality Assurance" is activated.
Module "RWI2 - Inspector" is activated
Messages
Progress
Insert
Ln: 8
Col: 5

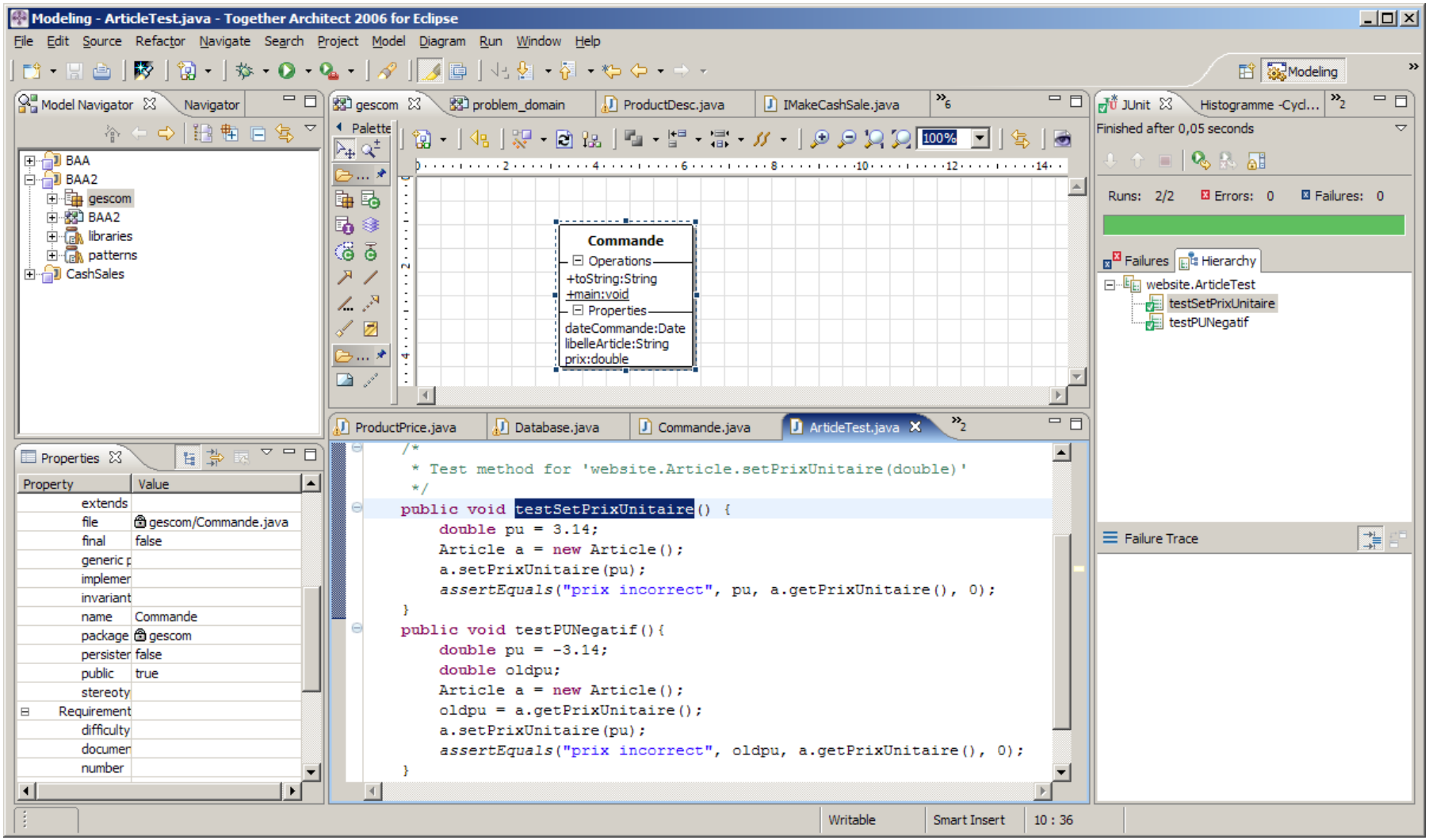



Traceability Diagram
File View Zoom Help
100%
Power Supply
HW97
Deposit to Account
SR117
Fee Message
SR106
ATM Machine
HW96
Replenishment Prediction
SR99
Withdrawal from Account
SR118
Power Supply
HW97
3 Verify Balance - BR81
BR81
ATM Machine
HW96
Replenishment Prediction
SR99
Account Balance
SR119
Fee Message
SR106
ATM Machine
HW96
Replenishment Prediction
SR99
Power Supply
HW97
Fee Message
SR106
problem_domain.CashSale
Class: problem_domain.CashSale in CashSales.tpr //[oiref:java#Class#problem_domain.CashSale:oiref> Verify_Balance,81
Ready

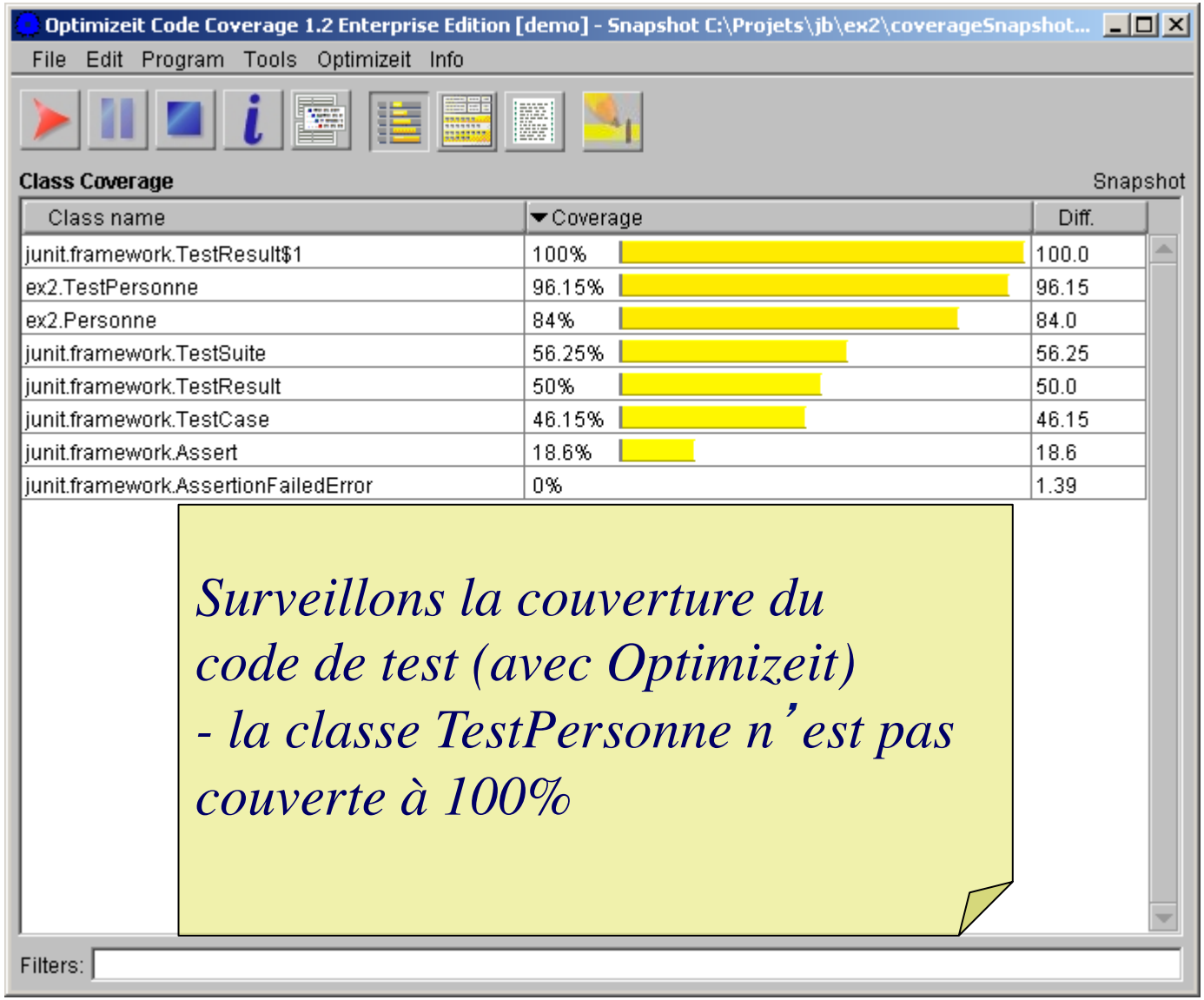
# IV. Tests unitaires et couverture du code

*Tests*. Programmers write unit tests minute by minute. These tests are collected and they must all run correctly. Customers write functional tests for the stories in an iteration. These tests should also all run, although practically speaking, sometimes a business decision must be made comparing the cost of shipping a known defect and the cost of delay.

- Risques
  - Stabilité incertaine
  - Visibilité réduite
  - Dépassement des délais
  - Débordement de budgets
- Solutions
  - Tests unitaires, exécution automatique
  - Stabilité permanente et transparente … disons « verte » ☺


Modeling - ArticleTest.java - Together Architect 2006 for Eclipse
File Edit Source Refactor Navigate Search Project Model Diagram Run Window Help
Model Navigator
Navigator
BAA
BAA2
gescom
BAA2
libraries
patterns
CashSales
gescom
problem_domain
ProductDesc.java
IMakeCashSale.java
Palette
100%
Commande
Operations
+toString:String
+main:void
Properties
dateCommande:Date
libelleArticle:String
prix:double
JUnit
Histogramme -Cycl...
Finished after 0,05 seconds
Runs: 2/2
Errors: 0
Failures: 0
Failures
Hierarchy
website.ArticleTest
testSetPrixUnitaire
testPUNegatif
Failure Trace
ProductPrice.java
Database.java
Commande.java
ArticleTest.java
Properties
Property
Value
extends
file
gescom/Commande.java
final
false
name
Commande
package
gescom
persister
false
public
true
Requirement
difficulty
number
/*
* Test method for 'website.Article.setPrixUnitaire(double)'
*/
public void testSetPrixUnitaire() {
double pu = 3.14;
Article a = new Article();
a.setPrixUnitaire(pu);
assertEquals("prix incorrect", pu, a.getPrixUnitaire(), 0);
}
public void testPUNegatif(){
double pu = -3.14;
double oldpu;
Article a = new Article();
oldpu = a.getPrixUnitaire();
a.setPrixUnitaire(pu);
assertEquals("prix incorrect", oldpu, a.getPrixUnitaire(), 0);
}
Writable
Smart Insert
10 : 36

# Doutes sur l'exhaustivité



# V. Propriété collective du code et gestion de configurations

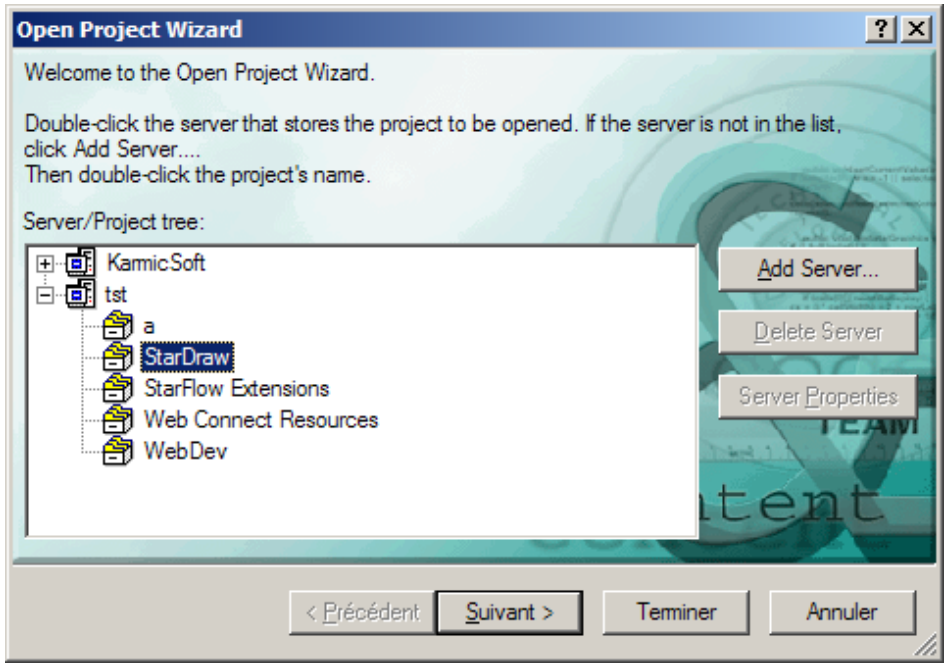
***Collective ownership***. Every programmer improves any code anywhere in the system at any time if they see the opportunity.

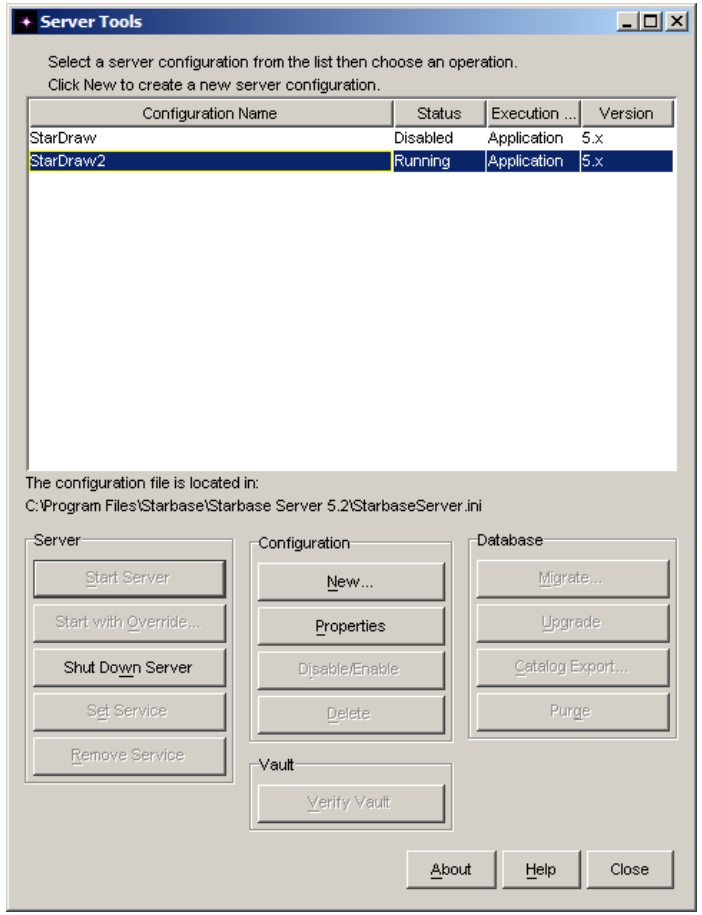
***Small releases***. The system is put into production in a few months, before solving the whole problem. New releases are made often—anywhere from daily to monthly.

***Continuous integration***. New code is integrated with the current system after no more than a few hours.

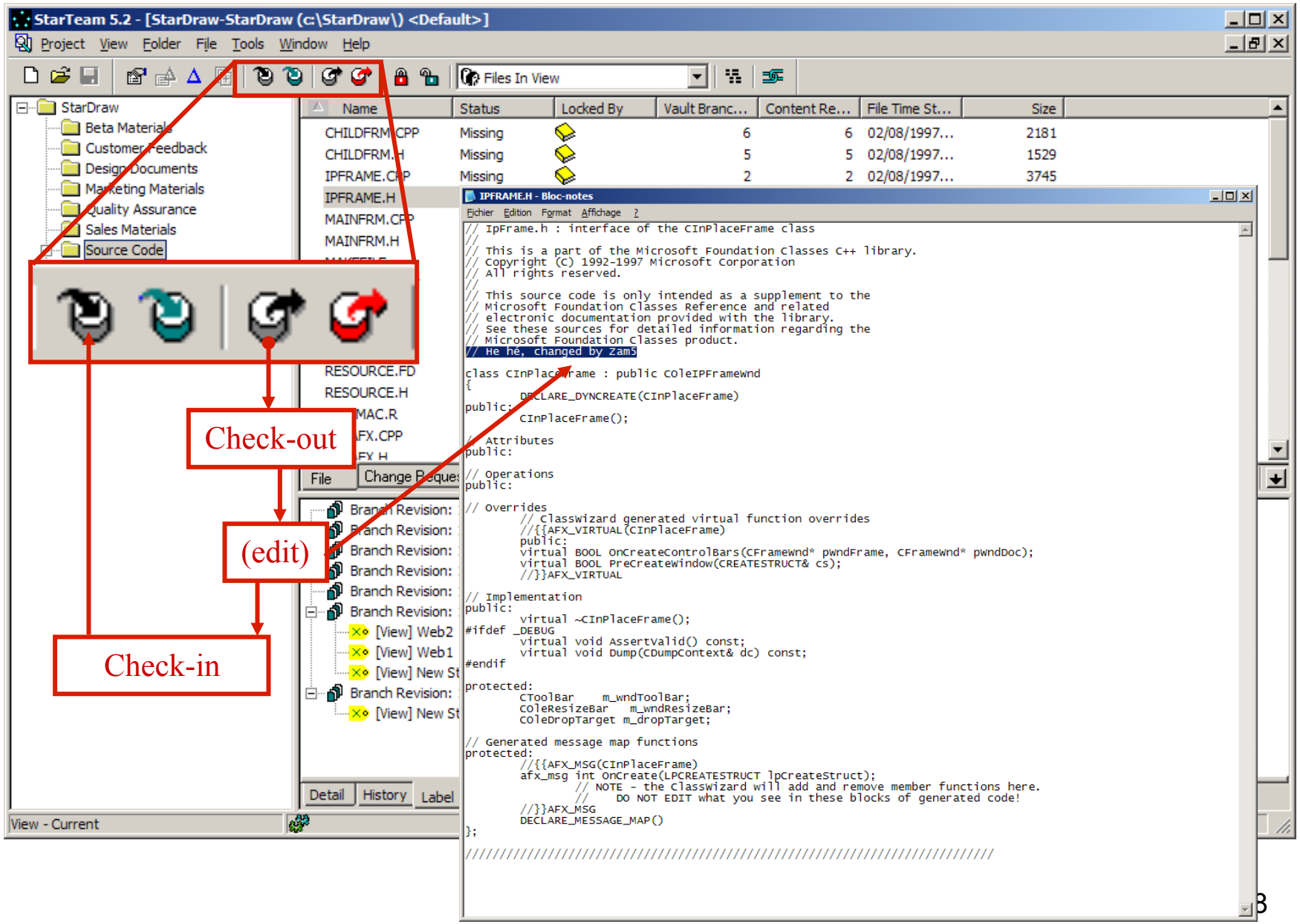
***Pair programming***. All production code is written by two people at one screen/keyboard/mouse.

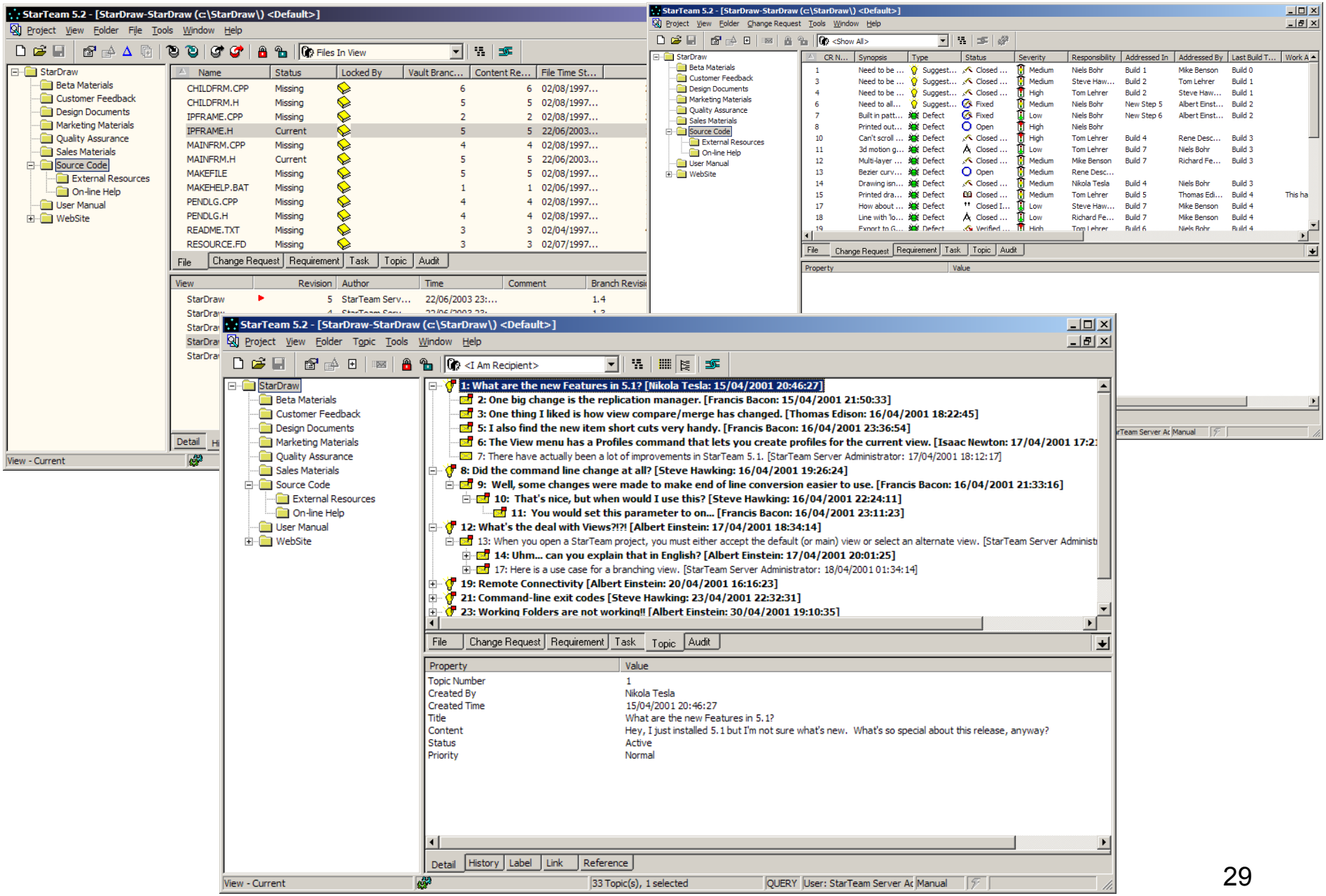
Risques
-Archaïsmes
-Rigidités
-Inerties
-Effet tunnel
-Turnover & passation



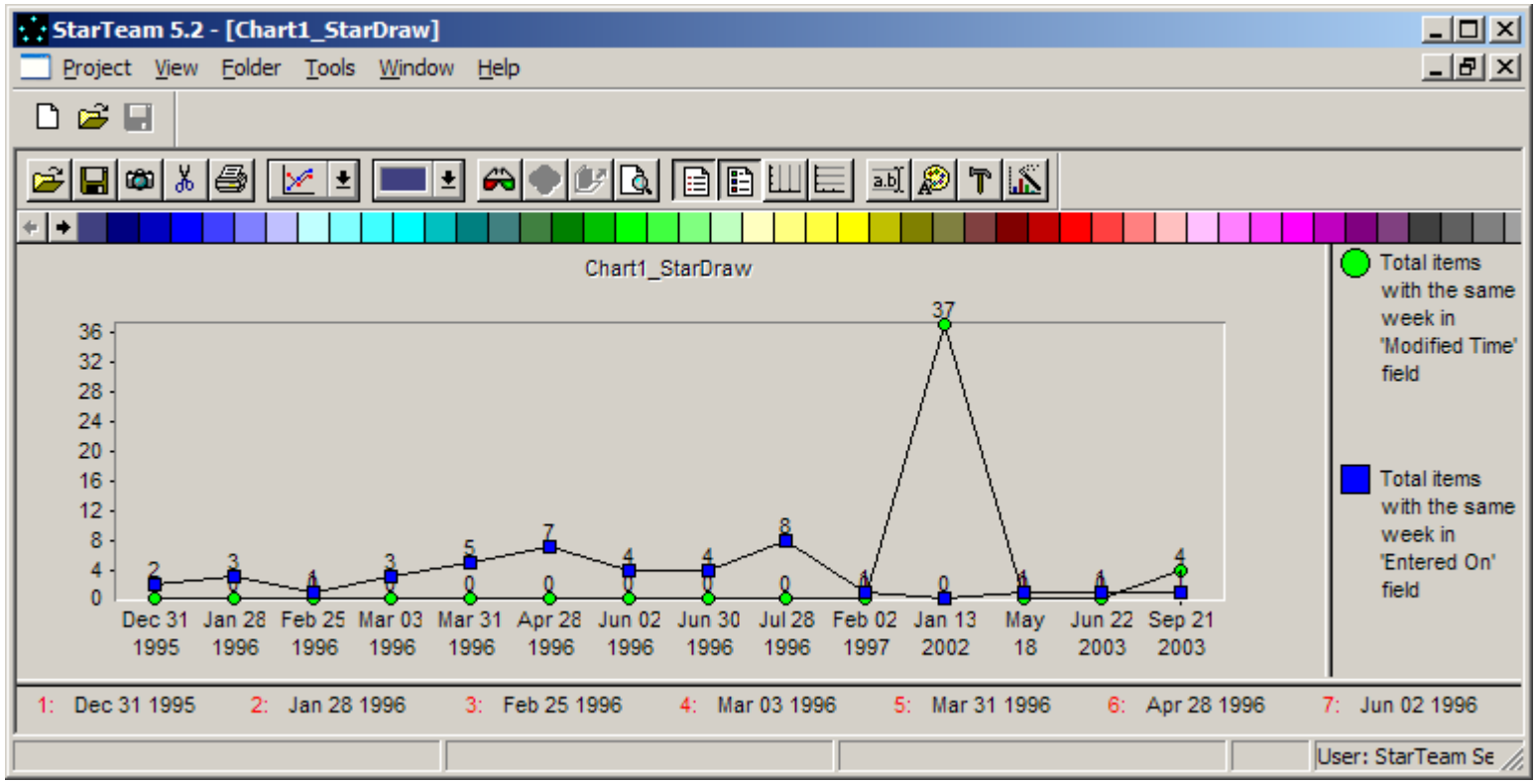




- Serveur
  - **Un référentiel partagé : SGBD**
- Client
  - **Répertoires de travail**
- Projet & éléments
  - **Organisation en répertoires**
  - **Fichiers, demandes d'évolution, exigences, tâches, topiques, audit**
  - **Cycle individuel : checkout – edition – checkin**
  - **Révision++, liens, comparaisons, étiquette (label), vue (filtres), configuration**

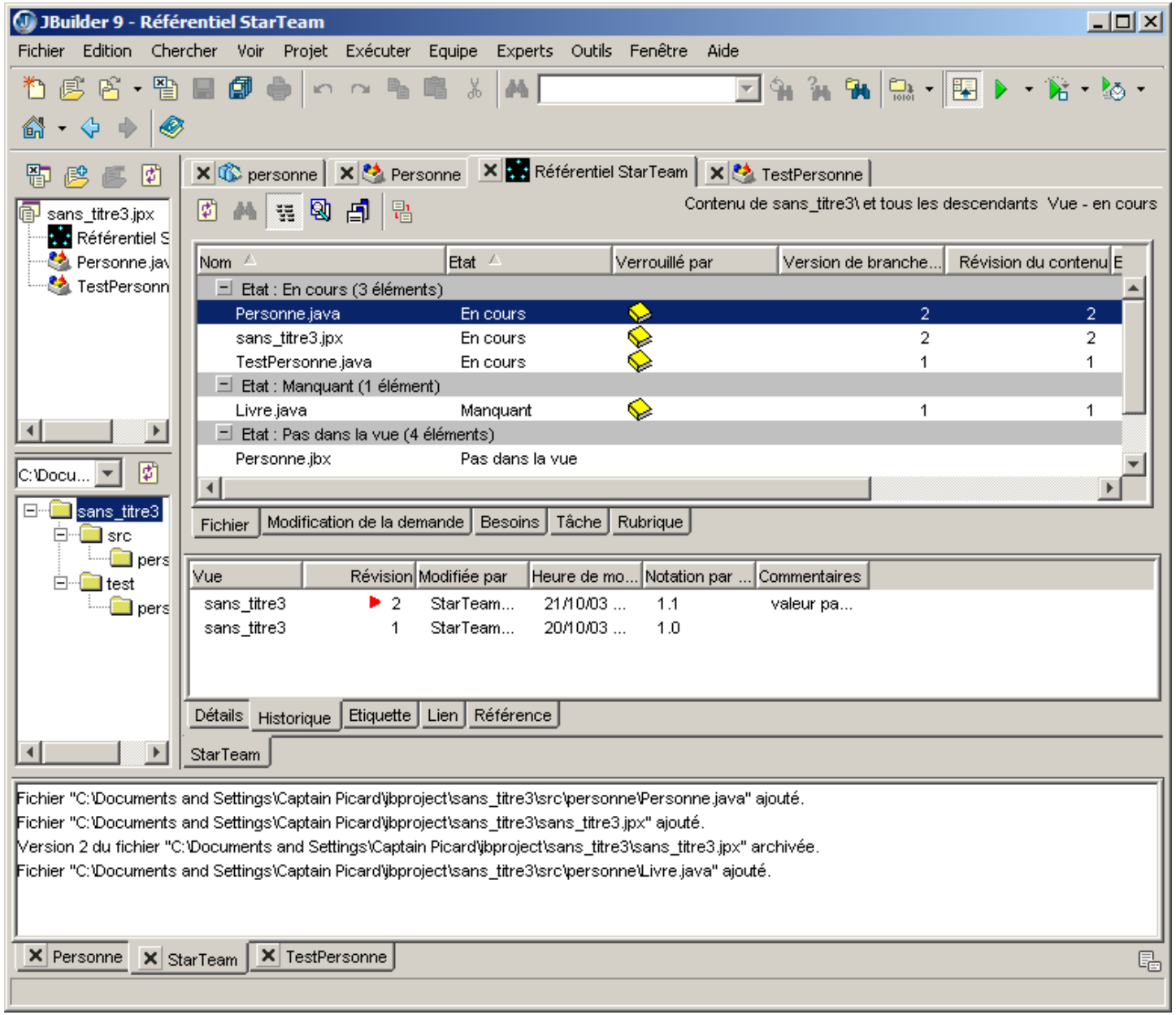

StarTeam 5.2 - [StarDraw-StarDraw (c:\StarDraw\) <Default>]
Project View Folder File Tools Window Help
Files In View
StarDraw
Beta Materials
Customer Feedback
Design Documents
Marketing Materials
Quality Assurance
Sales Materials
Source Code
Name
Status
Locked By
Vault Branc...
Content Re...
File Time St...
Size
CHILDFRM.CPP Missing 6 6 02/08/1997... 2181
CHILDFRM.H Missing 5 5 02/08/1997... 1529
IPFRAME.CPP Missing 2 2 02/08/1997... 3745
IPFRAME.H
MAINFRM.CPP
MAINFRM.H
RESOURCE.FD
RESOURCE.H
Check-out
(edit)
Check-in
File
Branch Revision:
[View] Web2
[View] Web1
[View] New St
Detail
History
Label
View - Current
IPFRAME.H - Bloc-notes
Fichier Edition Format Affichage ?
// IpFrame.h : interface of the CInPlaceFrame class
// This is a part of the Microsoft Foundation Classes C++ library.
// Copyright (C) 1992-1997 Microsoft Corporation
// All rights reserved.
// This source code is only intended as a supplement to the
// Microsoft Foundation Classes Reference and related
// electronic documentation provided with the library.
// See these sources for detailed information regarding the
// Microsoft Foundation Classes product.
// He hé, changed by Zam5
class CInPlaceFrame : public COleIPFrameWnd
{
DECLARE_DYNCREATE(CInPlaceFrame)
public:
CInPlaceFrame();
// Attributes
public:
// Operations
public:
// Overrides
// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CInPlaceFrame)
public:
virtual BOOL OnCreateControlBars(CFrameWnd* pWndFrame, CFrameWnd* pWndDoc);
virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
//}}AFX_VIRTUAL
// Implementation
public:
virtual ~CInPlaceFrame();
#ifdef _DEBUG
virtual void AssertValid() const;
virtual void Dump(CDumpContext& dc) const;
#endif
protected:
CToolBar m_wndToolBar;
COleResizeBar m_wndResizeBar;
COleDropTarget m_dropTarget;
// Generated message map functions
protected:
//{{AFX_MSG(CInPlaceFrame)
afx_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);
// NOTE - the ClassWizard will add and remove member functions here.
// DO NOT EDIT what you see in these blocks of generated code!
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
};
////////////////////////////////////////////////////////////////////////////


StarTeam 5.2 - [StarDraw-StarDraw (c:\StarDraw\) <Default>]
Project View Folder File Tools Window Help
Files In View
StarDraw
Beta Materials
Customer Feedback
Design Documents
Marketing Materials
Quality Assurance
Sales Materials
Source Code
External Resources
On-line Help
User Manual
WebSite
Name Status Locked By Vault Branc... Content Re... File Time St...
CHILDFRM.CPP Missing 6 6 02/08/1997...
CHILDFRM.H Missing 5 5 02/08/1997...
IPFRAME.CPP Missing 2 2 02/08/1997...
IPFRAME.H Current 5 5 22/06/2003...
MAINFRM.CPP Missing 4 4 02/08/1997...
MAINFRM.H Current 5 5 22/06/2003...
MAKEFILE Missing 5 5 02/08/1997...
MAKEHELP.BAT Missing 1 1 02/06/1997...
PENDLG.CPP Missing 4 4 02/08/1997...
PENDLG.H Missing 4 4 02/08/1997...
README.TXT Missing 3 3 02/04/1997...
RESOURCE.FD Missing 3 3 02/07/1997...
File Change Request Requirement Task Topic Audit
View Revision Author Time Comment Branch Revisi
StarDraw 5 StarTeam Serv... 22/06/2003 23:... 1.4
Detail
View - Current
StarTeam 5.2 - [StarDraw-StarDraw (c:\StarDraw\) <Default>]
Project View Folder Change Request Tools Window Help
<Show All>
CR N... Synopsis Type Status Severity Responsibility Addressed In Addressed By Last Build T... Work A
Property Value
StarTeam 5.2 - [StarDraw-StarDraw (c:\StarDraw\) <Default>]
Project View Folder Topic Tools Window Help
<I Am Recipient>
1: What are the new Features in 5.1? [Nikola Tesla: 15/04/2001 20:46:27]
2: One big change is the replication manager. [Francis Bacon: 15/04/2001 21:50:33]
3: One thing I liked is how view compare/merge has changed. [Thomas Edison: 16/04/2001 18:22:45]
5: I also find the new item short cuts very handy. [Francis Bacon: 16/04/2001 23:36:54]
6: The View menu has a Profiles command that lets you create profiles for the current view. [Isaac Newton: 17/04/2001 17:2
7: There have actually been a lot of improvements in StarTeam 5.1. [StarTeam Server Administrator: 17/04/2001 18:12:17]
8: Did the command line change at all? [Steve Hawking: 16/04/2001 19:26:24]
9: Well, some changes were made to make end of line conversion easier to use. [Francis Bacon: 16/04/2001 21:33:16]
10: That's nice, but when would I use this? [Steve Hawking: 16/04/2001 22:24:11]
11: You would set this parameter to on... [Francis Bacon: 16/04/2001 23:11:23]
12: What's the deal with Views?!?! [Albert Einstein: 17/04/2001 18:34:14]
13: When you open a StarTeam project, you must either accept the default (or main) view or select an alternate view. [StarTeam Server Administ
14: Uhm... can you explain that in English? [Albert Einstein: 17/04/2001 20:01:25]
17: Here is a use case for a branching view. [StarTeam Server Administrator: 18/04/2001 01:34:14]
19: Remote Connectivity [Albert Einstein: 20/04/2001 16:16:23]
21: Command-line exit codes [Steve Hawking: 23/04/2001 22:32:31]
23: Working Folders are not working!! [Albert Einstein: 30/04/2001 19:10:35]
File Change Request Requirement Task Topic Audit
Property Value
Topic Number 1
Created By Nikola Tesla
Created Time 15/04/2001 20:46:27
Title What are the new Features in 5.1?
Content Hey, I just installed 5.1 but I'm not sure what's new. What's so special about this release, anyway?
Status Active
Priority Normal
Detail History Label Link Reference
View - Current
33 Topic(s), 1 selected
QUERY
User: StarTeam Server Ac
Manual

# Rapports et graphiques



# *Intégration SCM-IDE*



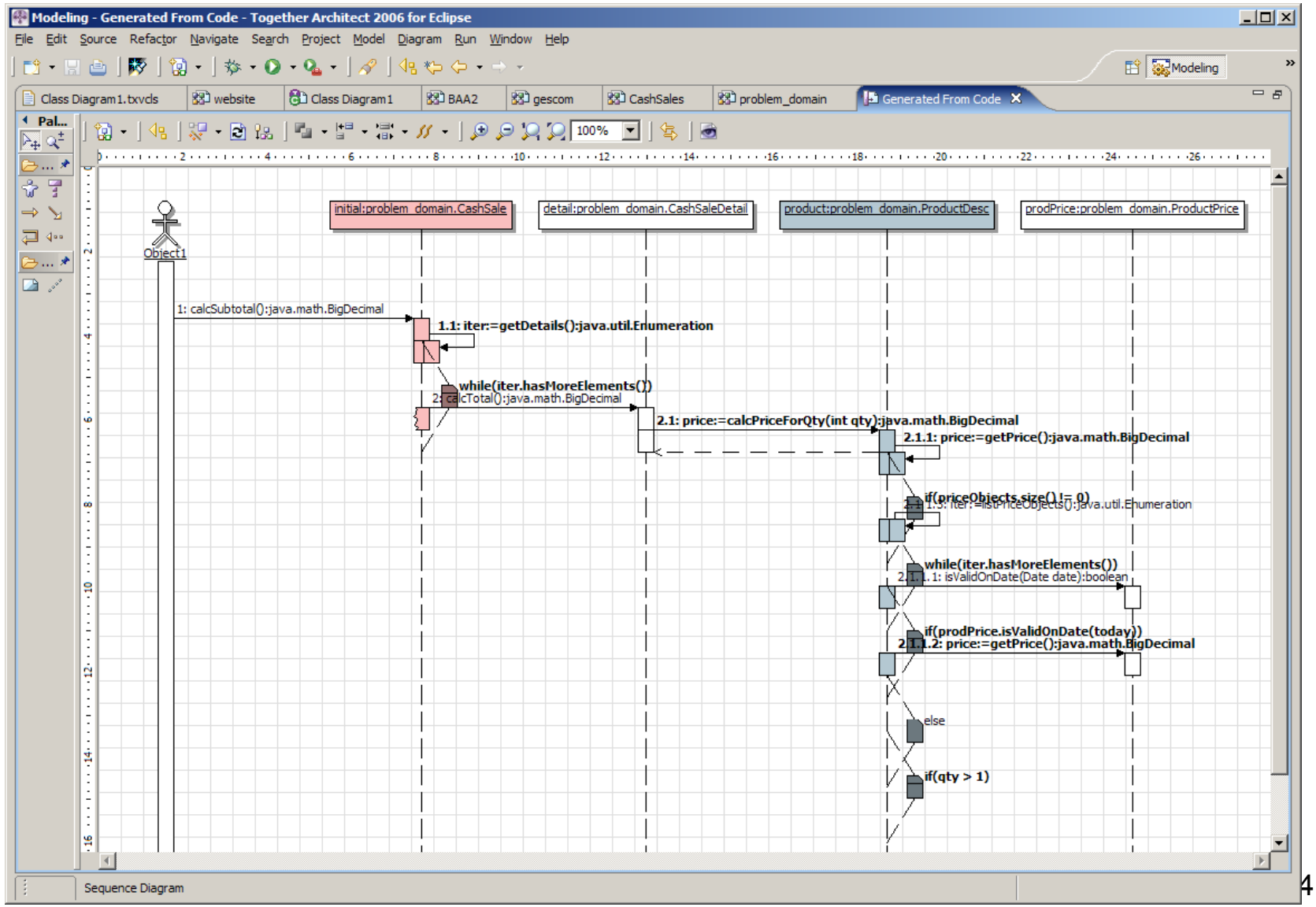
# VI. Métriques de stabilité et "refactoring" piloté par le modèle

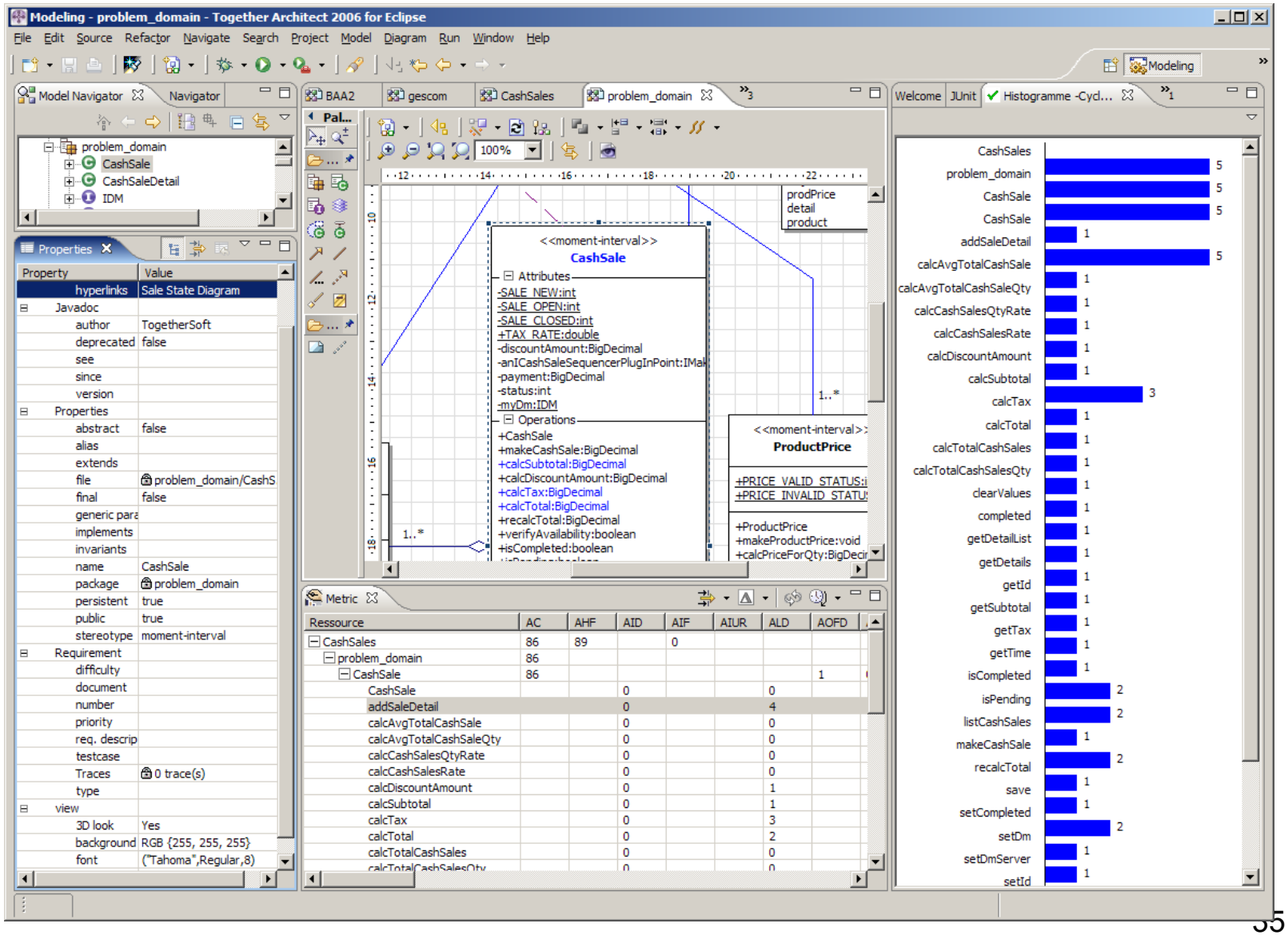
***Refactoring***. The design of the system is evolved through transformations of the existing design that keep all the tests running.

***Continuous integration***. New code is integrated with the current system after no more than a few hours. When integrating, the system is built from scratch and all tests must pass or the changes are discarded.

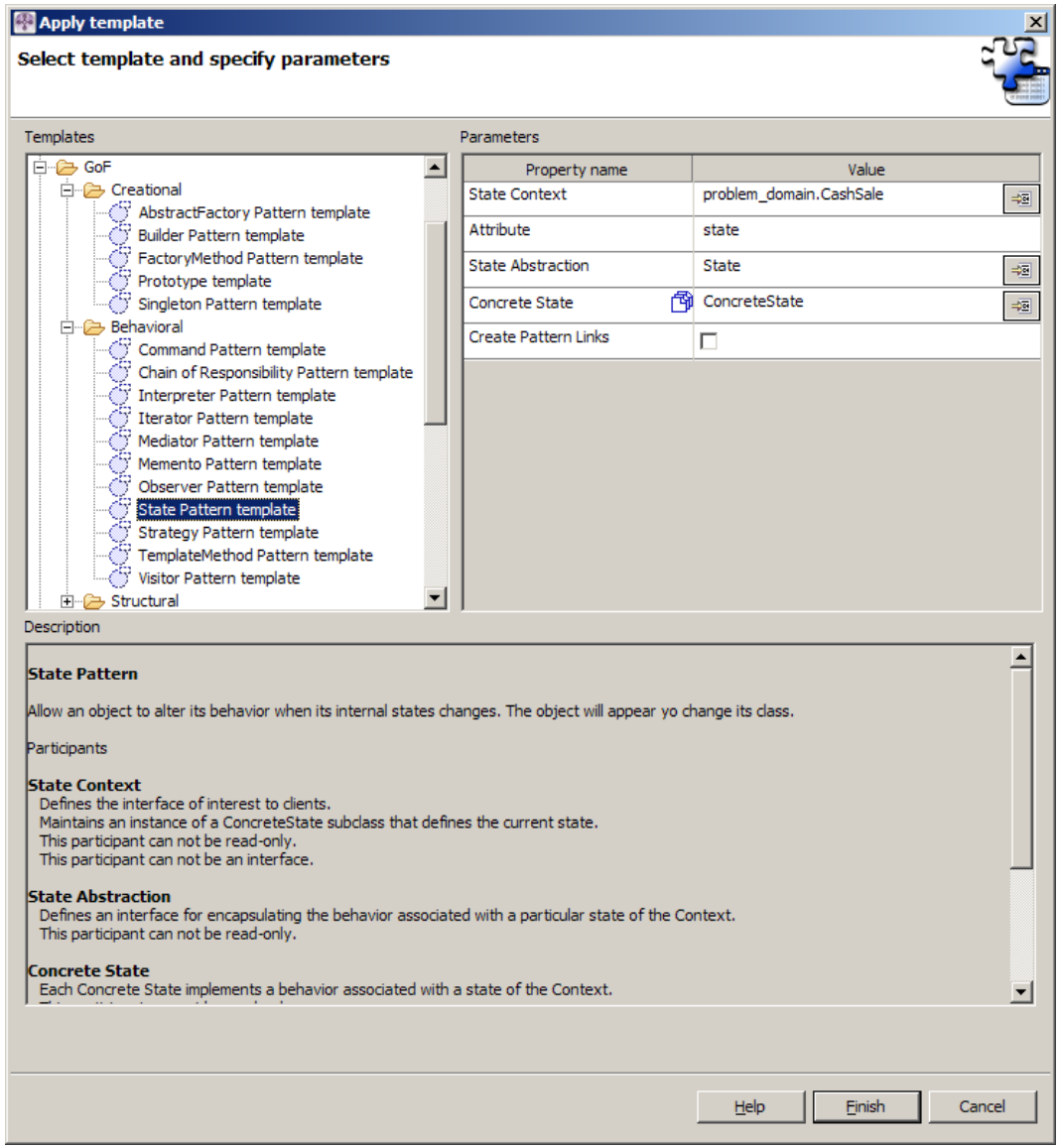
***Tests***. Programmers write unit tests minute by minute. These tests are collected and they must all run correctly. Customers write functional tests for the stories in an iteration. These tests should also all run, although practically speaking, sometimes a business decision must be made comparing the cost of shipping a known defect and the cost of delay.

Risques
- Mes diagrammes de séquence sont-ils toujours valables ?
- Mélange entre « présent » et « passé »
- Entropie croissante

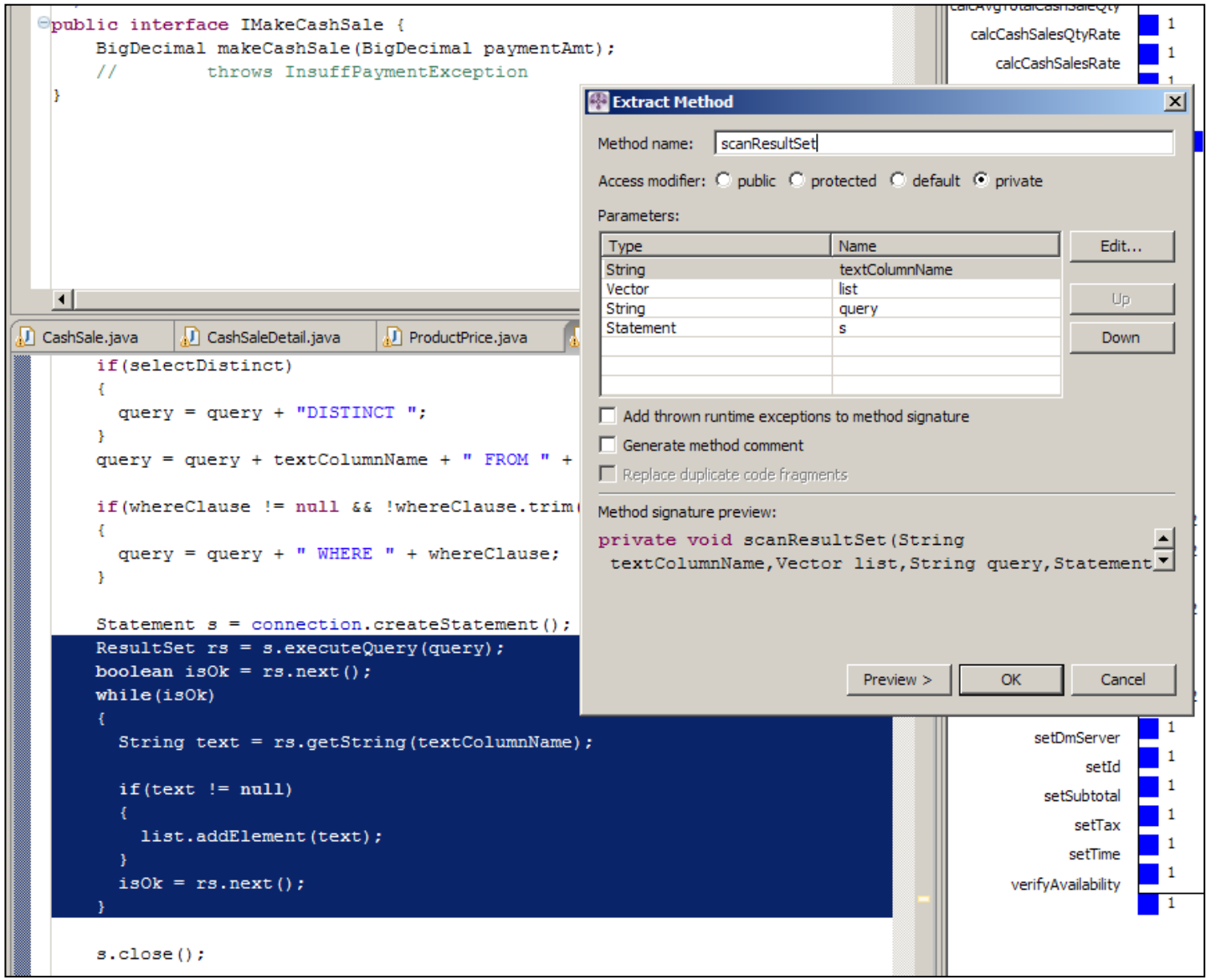

Modeling - Generated From Code - Together Architect 2006 for Eclipse
File Edit Source Refactor Navigate Search Project Model Diagram Run Window Help
Modeling
Class Diagram1.txvcls
website
Class Diagram1
BAA2
gescom
CashSales
problem_domain
Generated From Code
100%
initial:problem_domain.CashSale
detail:problem_domain.CashSaleDetail
product:problem_domain.ProductDesc
prodPrice:problem_domain.ProductPrice
Object1
1: calcSubtotal():java.math.BigDecimal
1.1: iter:=getDetails():java.util.Enumeration
while(iter.hasMoreElements())
2: calcTotal():java.math.BigDecimal
2.1: price:=calcPriceForQty(int qty):java.math.BigDecimal
2.1.1: price:=getPrice():java.math.BigDecimal
if(priceObjects.size() != 0)
while(iter.hasMoreElements())
2.1.1.1: isValidOnDate(Date date):boolean
if(prodPrice.isValidOnDate(today))
2.1.1.2: price:=getPrice():java.math.BigDecimal
else
if(qty > 1)
Sequence Diagram


Modeling - problem_domain - Together Architect 2006 for Eclipse
File Edit Source Refactor Navigate Search Project Model Diagram Run Window Help
Model Navigator
Navigator
problem_domain
CashSale
CashSaleDetail
IDM
Properties
Property
Value
hyperlinks
Sale State Diagram
Javadoc
author
TogetherSoft
deprecated
false
see
since
version
Properties
abstract
false
alias
extends
file
problem_domain/CashS
final
false
generic para
implements
invariants
name
CashSale
package
problem_domain
persistent
true
public
true
stereotype
moment-interval
Requirement
difficulty
document
number
priority
req. descrip
testcase
Traces
0 trace(s)
type
view
3D look
Yes
background
RGB {255, 255, 255}
font
("Tahoma",Regular,8)
BAA2
gescom
CashSales
problem_domain
Pal...
100%
<<moment-interval>>
CashSale
Attributes
-SALE_NEW:int
-SALE_OPEN:int
-SALE_CLOSED:int
+TAX_RATE:double
-discountAmount:BigDecimal
-anICashSaleSequencerPlugInPoint:IMak
-payment:BigDecimal
-status:int
-myDm:IDM
Operations
+CashSale
+makeCashSale:BigDecimal
+calcSubtotal:BigDecimal
+calcDiscountAmount:BigDecimal
+calcTax:BigDecimal
+calcTotal:BigDecimal
+recalcTotal:BigDecimal
+verifyAvailability:boolean
+isCompleted:boolean
prodPrice
detail
product
1..*
<<moment-interval>>
ProductPrice
+PRICE_VALID_STATUS:i
+PRICE_INVALID_STATU
+ProductPrice
+makeProductPrice:void
+calcPriceForQty:BigDecir
Metric
Ressource
AC
AHF
AID
AIF
AIUR
ALD
AOFD
CashSales 86 89 0
problem_domain 86
CashSale 86 1
CashSale 0 0
addSaleDetail 0 4
calcAvgTotalCashSale 0 0
calcAvgTotalCashSaleQty 0 0
calcCashSalesQtyRate 0 0
calcCashSalesRate 0 0
calcDiscountAmount 0 1
calcSubtotal 0 1
calcTax 0 3
calcTotal 0 2
calcTotalCashSales 0 0
Welcome
JUnit
Histogramme -Cycl...
CashSales 5
problem_domain 5
CashSale 5
CashSale 1
addSaleDetail 5
calcAvgTotalCashSale 1
calcAvgTotalCashSaleQty 1
calcCashSalesQtyRate 1
calcCashSalesRate 1
calcDiscountAmount 1
calcSubtotal 3
calcTax 1
calcTotal 1
calcTotalCashSales 1
calcTotalCashSalesQty 1
clearValues 1
completed 1
getDetailList 1
getDetails 1
getId 1
getSubtotal 1
getTax 1
getTime 1
isCompleted 1
isPending 2
listCashSales 2
makeCashSale 1
recalcTotal 2
save 1
setCompleted 1
setDm 2
setDmServer 1
setId 1

# Template de Design Pattern



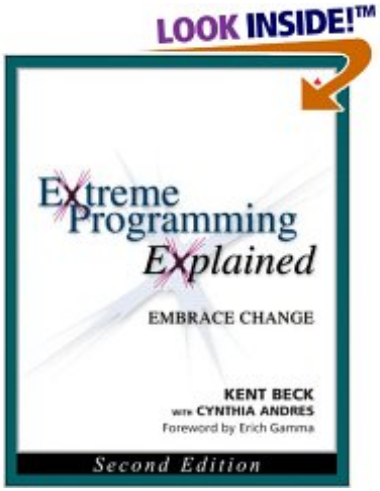
# *Refactoring : ExtractMethod*

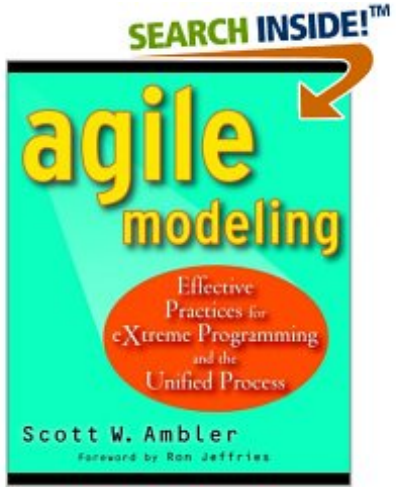


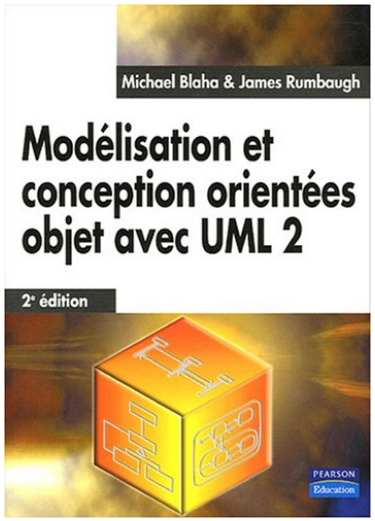
# Conclusion matérialiste et provisoire (promesse électorale)

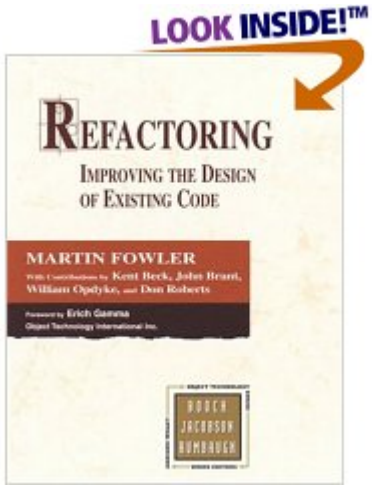
- Qualité globale = $€¥
- Un exemple
  - Nombre moyen de bugs /= 2..5
    - Calcul du glissement d'un mois pour un projet moyen (6 mois) :
    - 20 jours * 3 développeurs * 450 € = 27 K€ = 175 KF
  - Parallélisation accrue (flux tendu) => impact
    - 10 développeurs vs. 300 utilisateurs (qui font tourner le business et font entrer du CA)

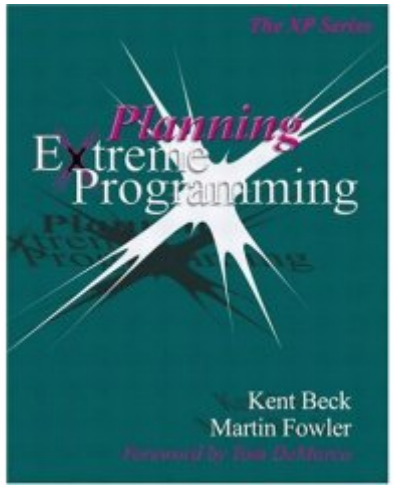
# Bibliographie agile

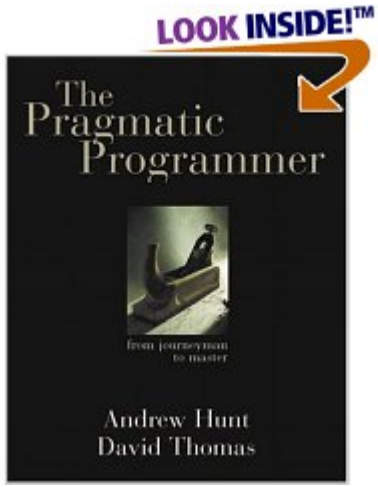


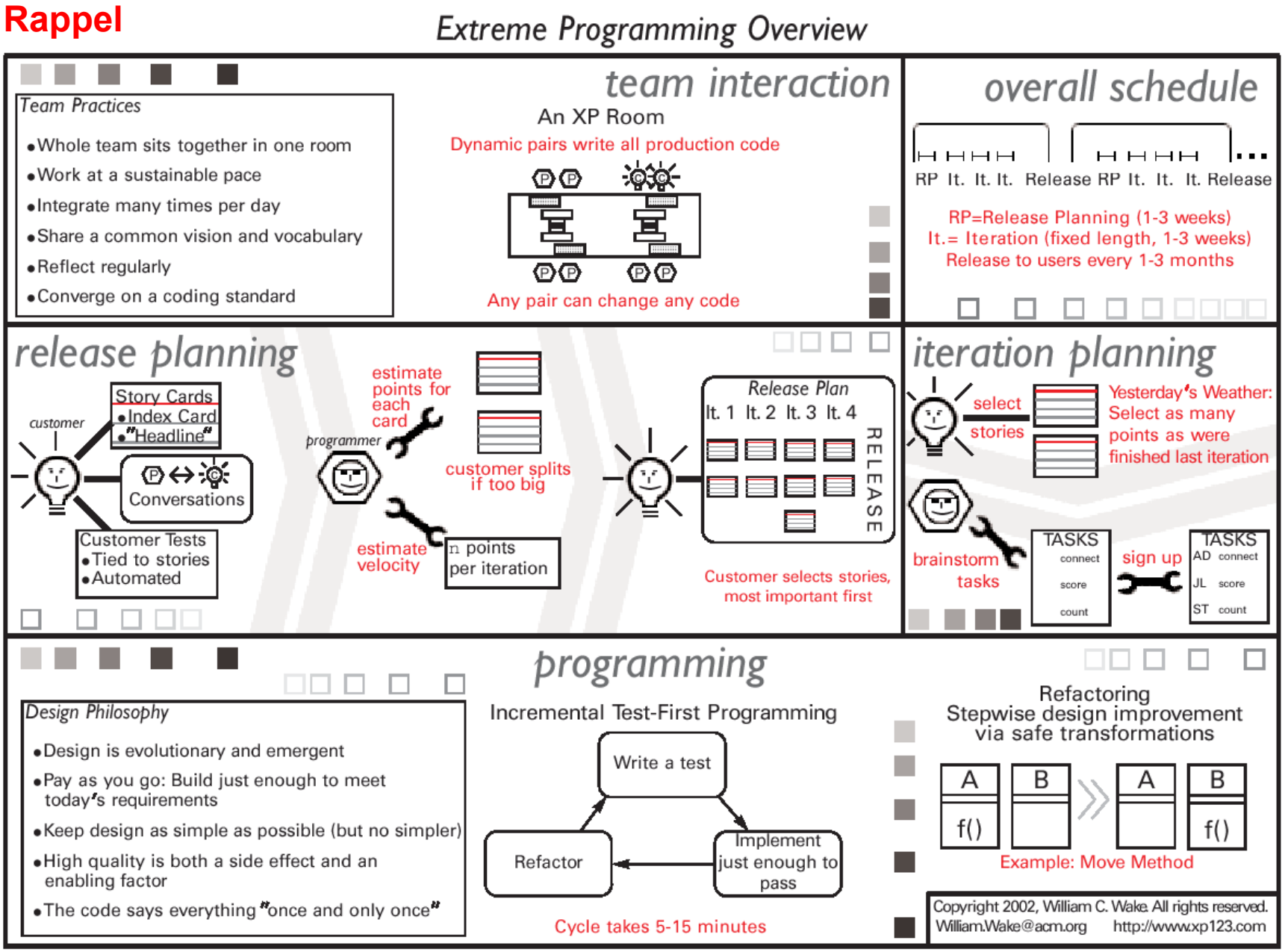



# TP3
# Intégration de pratiques agiles

En direct-live du château

**Rappel**


Extreme Programming Overview
team interaction
Team Practices
•Whole team sits together in one room
•Work at a sustainable pace
•Integrate many times per day
•Share a common vision and vocabulary
•Reflect regularly
•Converge on a coding standard
An XP Room
Dynamic pairs write all production code
Any pair can change any code
overall schedule
RP It. It. It. Release RP It. It. It. Release
RP=Release Planning (1-3 weeks)
It.= Iteration (fixed length, 1-3 weeks)
Release to users every 1-3 months
release planning
customer
Story Cards
•Index Card
•"Headline"
Conversations
Customer Tests
•Tied to stories
•Automated
programmer
estimate points for each card
customer splits if too big
estimate velocity
n points per iteration
Release Plan
It. 1 It. 2 It. 3 It. 4
RELEASE
Customer selects stories, most important first
iteration planning
select stories
Yesterday's Weather: Select as many points as were finished last iteration
brainstorm tasks
TASKS
connect
score
count
sign up
TASKS
AD connect
JL score
ST count
programming
Design Philosophy
•Design is evolutionary and emergent
•Pay as you go: Build just enough to meet today's requirements
•Keep design as simple as possible (but no simpler)
•High quality is both a side effect and an enabling factor
•The code says everything "once and only once"
Incremental Test-First Programming
Write a test
Implement just enough to pass
Refactor
Cycle takes 5-15 minutes
Refactoring
Stepwise design improvement via safe transformations
A
B
f()
A
B
f()
Example: Move Method
Copyright 2002, William C. Wake. All rights reserved.
William.Wake@acm.org http://www.xp123.com

# Scénario d'intégration agile

1. Rédiger les **User Stories** (exigences)
2. Planifier les **Itérations** (quoi / quand)
3. Ecrire le code et les **tests unitaires**
4. **Archiver** les implémentations
5. Exécuter et tester interactivement
6. Signaler et corriger les **anomalies** et suivre **l'avancement**
7. Assurer la reproductibilité

   ➔ **développement parallèle**

# 1. User Stories (1)

**US01 – Enregistrer les commandes – Auteur : MZ**
*Pour chaque commande, enregistrer le libellé de l'article, la quantité et le prix unitaire.*
*Le numéro de la commande (entier) et le total doivent être automatiquement calculés.*
• TST01 : construction de commandes cohérentes et la validité du total
• TST02 : refuser les prix et les quantités négatifs

**US02 – Gérer les articles – Auteur : MZ**
*Stocker le catalogue d'articles, avec libellé et prix unitaire.*
*Associer les commandes aux articles.*
• TST03 : construction d'articles cohérents vérifier les prix vides ou non négatifs
• TST04 : ne pas autoriser des prix négatifs ou non renseignés pour les articles associés aux commandes

# User Stories (2)

**US03 – Gérer les clients – Auteur : MZ**
*Stocker les clients – avec nom, prénom, âge et email – et les associer aux commandes.*
• TST05 : construction de clients cohérents : ne pas autoriser les noms vides et les adresses email invalides.
• TST06 : toute commande doit être associée à un client cohérent

<< Un mois plus tard >>
**US21 – Gérer les commandes hétérogènes – Auteur : MZ**
*Les commandes pourront contenir plusieurs lignes, chacune portant vers un article différent. Autoriser les ristournes*
• TST32 : ….

# Compréhension

- Pour bien estimer la charge de travail, il vaut mieux comprendre les exigences, mais parfois c'est un peu difficile.
- Esquisser des diagrammes peut aider à estimer la complexité, sans forcément nécessiter un approfondissement du métier du client …

  => topologie du graphe sémantique

# UML (ou pas)

- Client 1-- * Commande * -- 1 Article
- Observons la topologie, le graphe des dépendances
- Plissons les yeux et libérons-nous de la sémantique
- Effectuons les estimations sur la base structurelle + complexité des comportements à implémenter …

# User Stories – estimation (1)

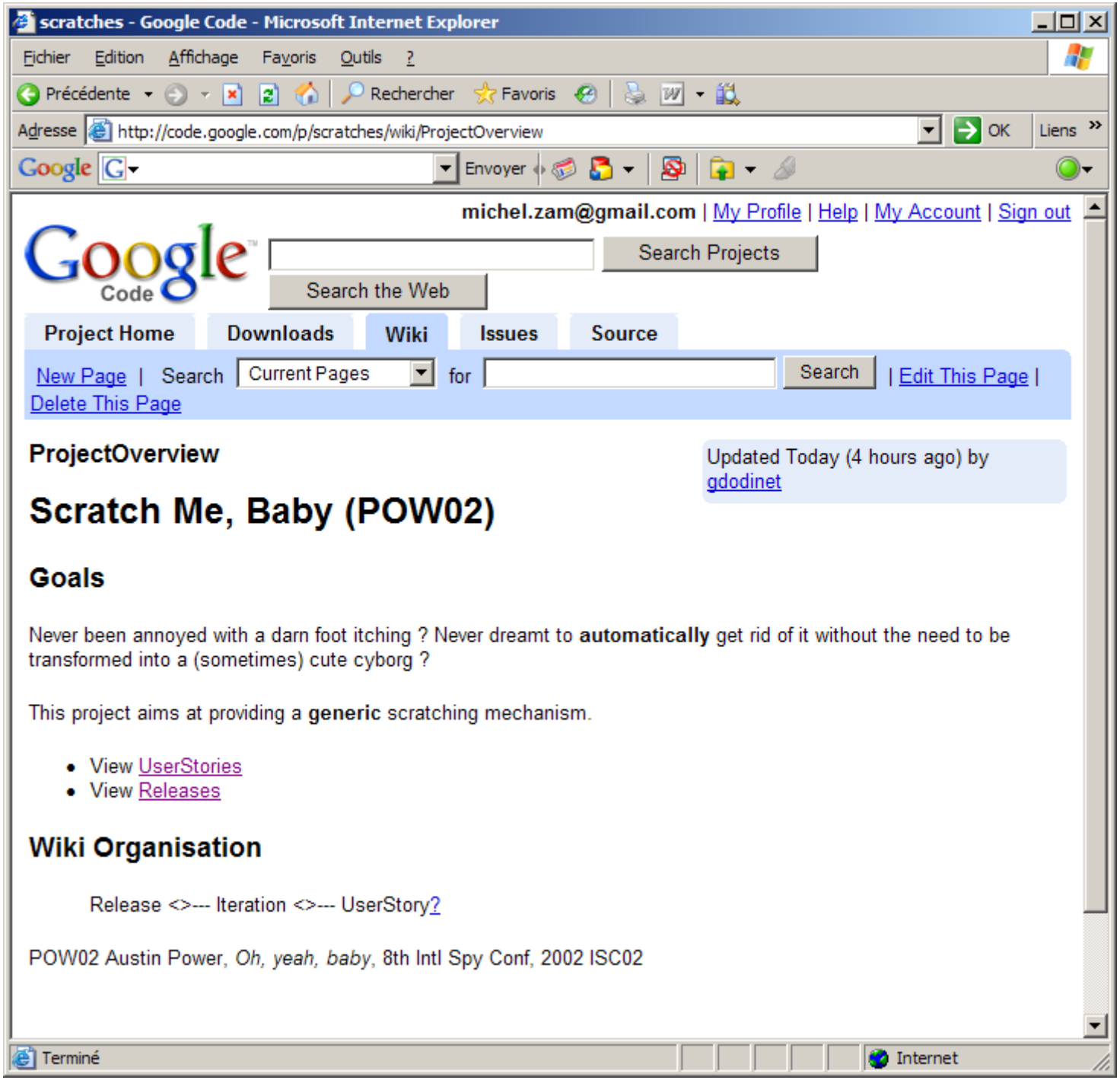
**US01 – Enregistrer les commandes – Auteur : MZ**
*Pour chaque commande, enregistrer le libellé de l'article, la quantité et le prix unitaire.*
*Le numéro de la commande (entier) et le total doivent être automatiquement calculés.*
• TST01 : construction de commandes cohérentes et la validité du total
• TST02 : refuser les prix et les quantités négatifs
Estimation de charge [GD] : 6 points

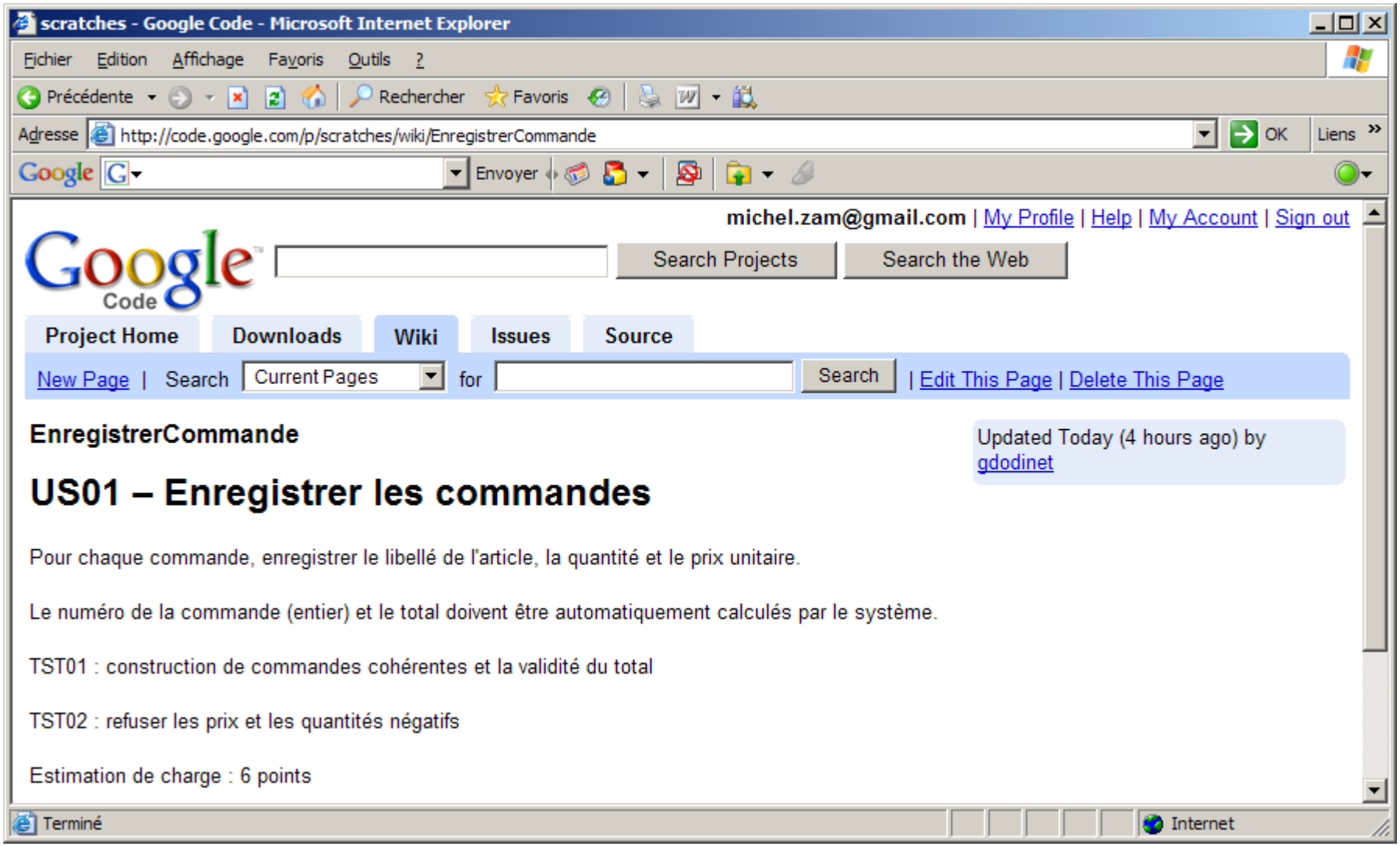
**US02 – Gérer les articles – Auteur : MZ**
*Stocker le catalogue d'articles, avec libellé et prix unitaire.*
*Associer les commandes aux articles.*
• TST03 : construction d'articles cohérents vérifier les prix vides ou non négatifs
• TST04 : ne pas autoriser des prix négatifs ou non renseignés pour les articles associés aux commandes
Estimation de charge [GD] : 4 points

# User Stories – estimation (2)

**US03 – Gérer les clients – Auteur : MZ**
*Stocker les clients – avec nom, prénom, âge et email – et les associer aux commandes.*
• TST05 : construction de clients cohérents : ne pas autoriser les noms vides et les adresses email invalides.
• TST06 : toute commande doit être associée à un client cohérent
Estimation de charge [GD] : 4 points

<< Un mois plus tard >>
**US21 – Gérer les commandes hétérogènes – Auteur : MZ**
*Les commandes pourront contenir plusieurs lignes, chacune portant vers un article différent. Autoriser les ristournes*
• TST32 : ….
Estimation de charge [GD] : 5 points


scratches - Google Code - Microsoft Internet Explorer
Fichier Edition Affichage Favoris Outils ?
Précédente Rechercher Favoris
Adresse http://code.google.com/p/scratches/wiki/ProjectOverview
OK Liens
Google Envoyer
michel.zam@gmail.com | My Profile | Help | My Account | Sign out
Google Code
Search Projects
Search the Web
Project Home Downloads Wiki Issues Source
New Page | Search Current Pages for Search | Edit This Page | Delete This Page
ProjectOverview
Updated Today (4 hours ago) by gdodinet
Scratch Me, Baby (POW02)
Goals
Never been annoyed with a darn foot itching ? Never dreamt to automatically get rid of it without the need to be transformed into a (sometimes) cute cyborg ?
This project aims at providing a generic scratching mechanism.
View UserStories
View Releases
Wiki Organisation
Release <>--- Iteration <>--- UserStory?
POW02 Austin Power, Oh, yeah, baby, 8th Intl Spy Conf, 2002 ISC02
Terminé
Internet


scratches - Google Code - Microsoft Internet Explorer
Fichier Edition Affichage Favoris Outils ?
Précédente Rechercher Favoris
Adresse http://code.google.com/p/scratches/wiki/EnregistrerCommande
OK Liens
Google Envoyer
michel.zam@gmail.com | My Profile | Help | My Account | Sign out
Google Code
Search Projects
Search the Web
Project Home
Downloads
Wiki
Issues
Source
New Page | Search Current Pages for Search | Edit This Page | Delete This Page
EnregistrerCommande
Updated Today (4 hours ago) by gdodinet
US01 – Enregistrer les commandes
Pour chaque commande, enregistrer le libellé de l'article, la quantité et le prix unitaire.
Le numéro de la commande (entier) et le total doivent être automatiquement calculés par le système.
TST01 : construction de commandes cohérentes et la validité du total
TST02 : refuser les prix et les quantités négatifs
Estimation de charge : 6 points
Terminé
Internet

# Itérations

**Planning des itérations de la livraison R1 :**

Itération no 1 – Gabarit : 10 points

- US01 [6p] : état : effectué
- US03 [4p] : état : en cours

  Etat global : en cours

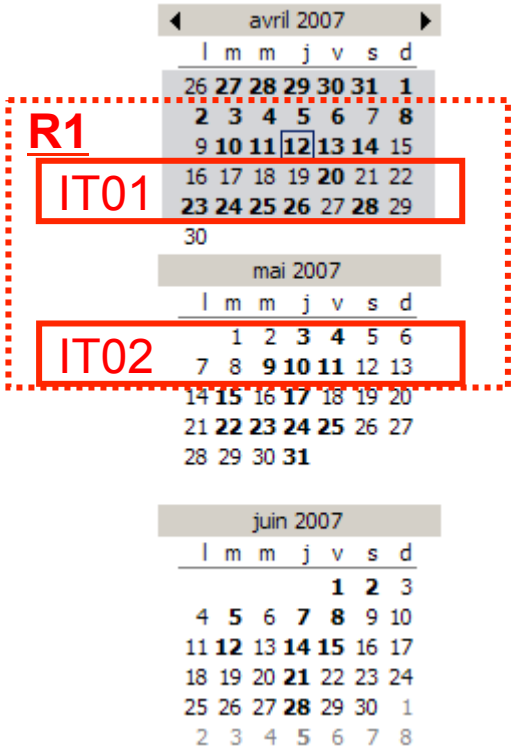
Itération no 2 – Gabarit : 10 points

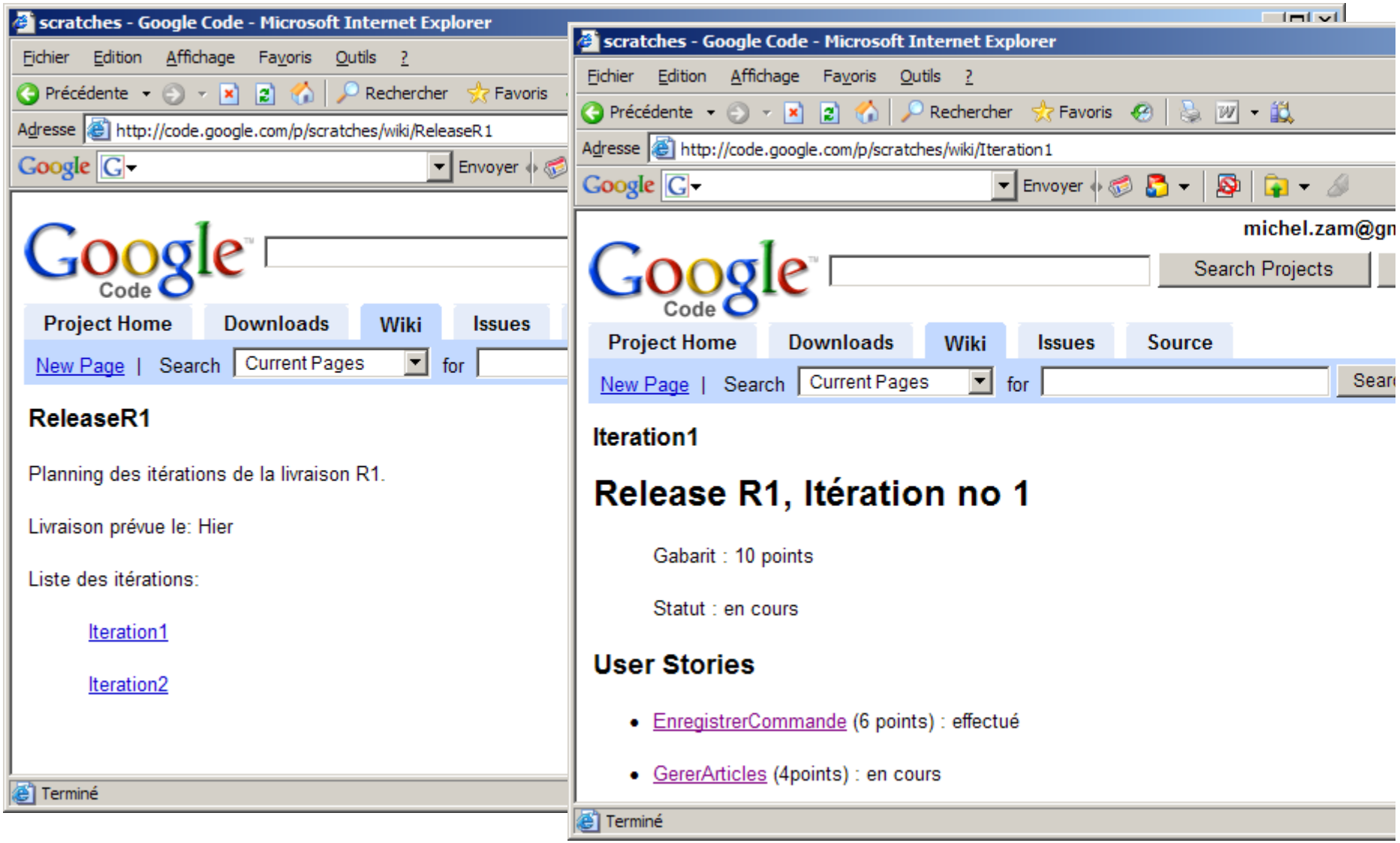
- US02 [4p] : état : en cours
- US04 …
- TK01 : passage en EJB

  Etat global : non commencé

….

➔ Assurer un rythme durable




scratches - Google Code - Microsoft Internet Explorer
Fichier Edition Affichage Favoris Outils ?
Précédente Rechercher Favoris
Adresse http://code.google.com/p/scratches/wiki/ReleaseR1
Google Envoyer
Google Code
Project Home Downloads Wiki Issues
New Page | Search Current Pages for
ReleaseR1
Planning des itérations de la livraison R1.
Livraison prévue le: Hier
Liste des itérations:
Iteration1
Iteration2
Terminé
scratches - Google Code - Microsoft Internet Explorer
Fichier Edition Affichage Favoris Outils ?
Précédente Rechercher Favoris
Adresse http://code.google.com/p/scratches/wiki/Iteration1
Google Envoyer
michel.zam@gn
Google Code
Search Projects
Project Home Downloads Wiki Issues Source
New Page | Search Current Pages for
Iteration1
Release R1, Itération no 1
Gabarit : 10 points
Statut : en cours
User Stories
EnregistrerCommande (6 points) : effectué
GererArticles (4points) : en cours
Terminé

# 3. Tests unitaires

```
/*
 * Test method for 'website.Article.setPrixUnitaire(double)'
 */
public void testSetPrixUnitaire() {
    double pu = 3.14;
    Article a = new Article();
    a.setPrixUnitaire(pu);
    assertEquals("prix incorrect", pu, a.getPrixUnitaire(), 0);
}
public void testPUNegatif(){
    double pu = -3.14;
    double oldpu;
    Article a = new Article();
    oldpu = a.getPrixUnitaire();
    a.setPrixUnitaire(pu);
    assertEquals("prix incorrect", oldpu, a.getPrixUnitaire(), 0);
}
```

TST02A

# 4. Archivage

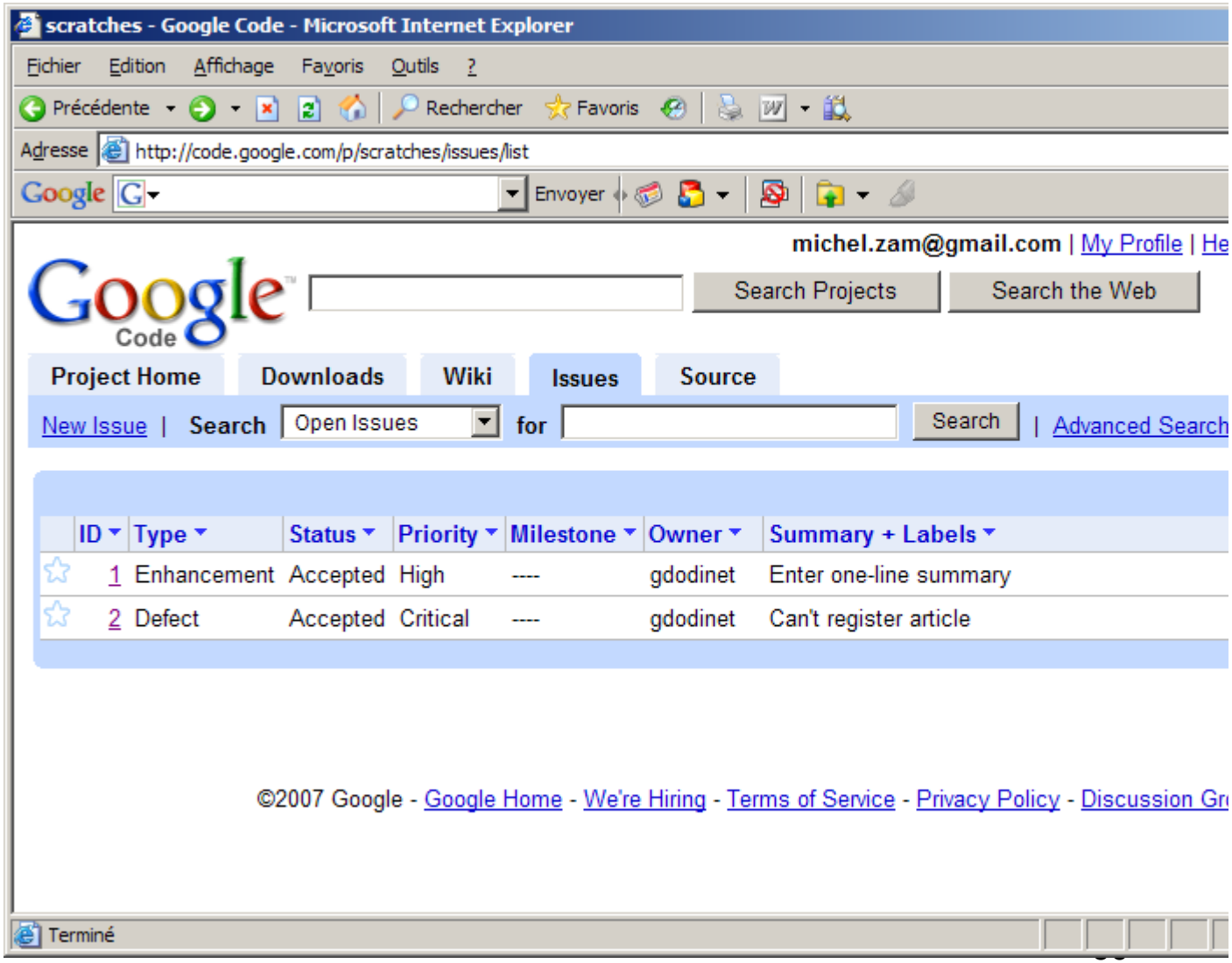
- Référentiel partagé (dépôt)
- Répertoire de travail (privé)
- Archivage (v++) et Extraction
- Etat d'un fichier
  - nouveau, courent, obsolète, …
- Travail collaboratif
  - Évitement et résolution des conflits

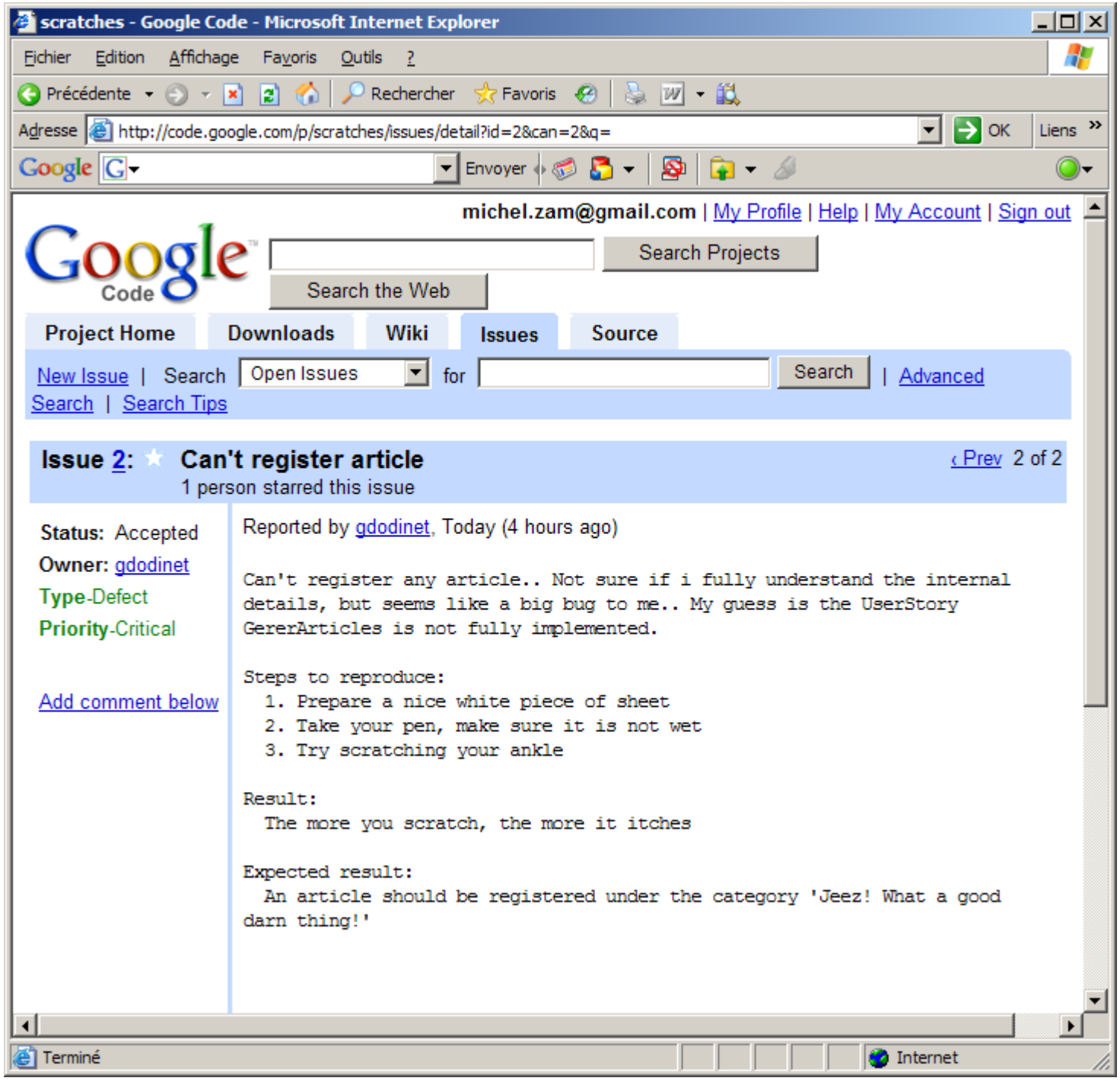
# 5. Tests interactifs

- Suivent les tests fonctionnels explicitement rédigés lors des US : valeurs attendues pour un jeu de paramètre donné
- Doivent comprendre aussi les tests génériques préconisé par le framework interactif : menus, écrans, onglets, ordre de tabulation, tri par clic sur colonne, …
- Exécution manuelle ou automatisée (outillage adapté)

# 6. Anomalies

- Signaler
- Corriger
- Suivre




scratches - Google Code - Microsoft Internet Explorer
Fichier Edition Affichage Favoris Outils ?
Précédente Rechercher Favoris
Adresse http://code.google.com/p/scratches/issues/detail?id=2&can=2&q=
OK Liens
Google Envoyer
michel.zam@gmail.com | My Profile | Help | My Account | Sign out
Google Code
Search Projects
Search the Web
Project Home Downloads Wiki Issues Source
New Issue | Search Open Issues for Search | Advanced Search | Search Tips
Issue 2: Can't register article
1 person starred this issue
‹ Prev 2 of 2
Status: Accepted
Owner: gdodinet
Type-Defect
Priority-Critical
Add comment below
Reported by gdodinet, Today (4 hours ago)
Can't register any article.. Not sure if i fully understand the internal details, but seems like a big bug to me.. My guess is the UserStory GererArticles is not fully implemented.
Steps to reproduce:
1. Prepare a nice white piece of sheet
2. Take your pen, make sure it is not wet
3. Try scratching your ankle
Result:
The more you scratch, the more it itches
Expected result:
An article should be registered under the category 'Jeez! What a good darn thing!'
Terminé
Internet

# 7. Développement parallèle

On nous signale un bug sur une livraison antérieure

1) on tente de le reproduire le bug
   - Checkout dans un nouveau projet Eclipse d'une "config" du passé précédemment tagguée
   - Compilation, exécution, constat visuel du bug
   - Ecriture d'un test unitaire qui met en évidence le bug par une barre rouge
     - Remarque: notez ici le passage en mode « test driven development » (TDD)
   - Archivage
2) correction du bug
   - Verdir la barre junit
   - Création d'une nouvelle branche (B2)
   - Archivage de la correction dans B2
   - Pose d'un nouveau « tag » et livraison

3) propagation du bug dans la branche B1

# Pratiques agiles en UML

- Modéliser les concepts suivants et leurs relations :
- Diagramme de classe :
  - Release, Itération, User Story, Test, Test unitaire, Fichier source, Référentiel, Révision, Etat fichier
- Diagramme d'activités cyclique :
  - UserStory ➔ Test unitaire ➔ Code ➔ Exécution scénario de test ➔ Statut d'achèvement et nouvelle userstory

*« La différence entre la théorie et la pratique est que selon la théorie il n'y en a pas »*

Martin Fowler