

SQUANER(System of QUality AnliZER)

HADERER Nicolas

Department of Computer and Software Engineering
École Polytechnique de Montréal, Québec, Canada

October 5, 2010

Introduction

Pourquoi mesurer la qualité
logicielle ?

La qualité logicielle
Monitorat du logiciel

Analyse de
l'existant

Gutsy

Squaner

Architecture
Modèle de Squaner
Détection des motifs
Les modèles de qualité
Analyse

Interface web

Perspective et
conclusion

Questions ?

Sommaire

Introduction

Pourquoi mesurer la qualité logicielle ?

La qualité logicielle

Monitorat du logiciel

Analyse de l'existant

Gutsy

Squaner

Architecture

Modèle de Squaner

Détection des motifs

Les modèles de qualité

Analyse

Interface web

Perspective et conclusion

Questions ?

Pourquoi mesurer la qualité logicielle ?

*"Au fur et à mesure qu'un logiciel évolue, sa structure a tendance à se complexifier. Il sera nécessaire de consacrer des ressources supplémentaires si l'on veut inverser sa dégradation".*Lehman 1974 [6]

Pourquoi mesurer la qualité logicielle ?

Des études réalisées sur le noyau Linux montre que [2], [10]
:

- ▶ croissance linéaire des fonctionnalités
- ▶ augmentation exponentielle du couplage entre les différents modules

Conclusion des études

Si aucun effort n'est fait pour altérer l'explosion du couplage, le noyau Linux ne pourrait plus être maintenu.

Pourquoi mesurer la qualité logicielle ?

Pourquoi ce phénomène ?

Parnas [9] en 1994 attribue la dégradation du logiciel à un phénomène appelé "*Ignorant surgery*" (chirurgien de l'ignorant).

Pourquoi mesurer la qualité logicielle ?

Difficultés supplémentaires pour les équipes Open-Source :

1. équipe souvent éloignée géographiquement
2. rotation fréquente
3. manque de documentation explicite
4. manque de planning général
5. manque de moyen formel pour contrôler l'évolution

La qualité logicielle

Premier modèle de qualité introduit par Jim McCall en 1977 pour US Air Force. Définition de trois perspectives majeures pour définir et identifier la qualité du produit logiciel.

- ▶ **Produit Révision**
 - ▶ *maintenabilité, flexibilité, testabilité*
- ▶ **Produit transition**
 - ▶ *réutilisabilité, interopérabilité :*
- ▶ **Produit opération**
 - ▶ *fiabilité, efficacité, intégrité, utilisabilité*

La qualité logicielle

Standardisation de l'évaluation d'un système par ISO (International Organization for Standardization)

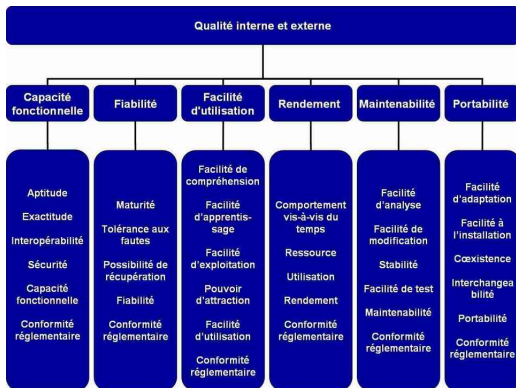


Figure: Extrait de la hiérarchie des caractéristiques internes et externes de l'ISO 9126 [8]

La qualité logicielle

QMOOD Bansiya et Davis [1].

6 équations pour l'évaluation des systèmes orientés objet.

- ▶ réutilisabilité
- ▶ flexibilité
- ▶ compréhensibilité
- ▶ fonctionnalité
- ▶ extensibilité
- ▶ efficacité

La qualité logicielle

Qualité et conception du logiciel :

- ▶ Patron de conception
- ▶ Anti-patron de conception

SQUAD [5] : Premier modèle de prédiction se basant sur le design du code.

Monitorat du logiciel

SONAR

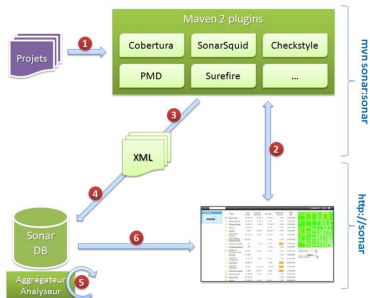


Figure: L'architecture de Sonar

Monitorat du logiciel

Squale

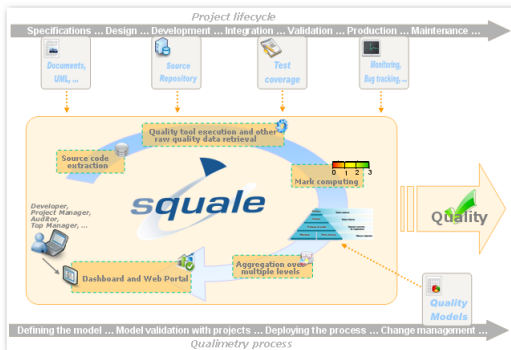


Figure: Diagramme de Squale

Analyse de l'existant

Un premier prototype *Gutsy*

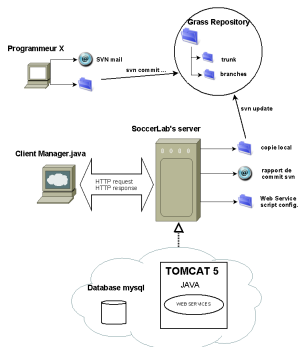


Figure: Première architecture Gutsy

Introduction

Pourquoi mesurer la qualité
logicielle ?

La qualité logicielle

Monitorat du logiciel

Analyse de
l'existant

Gutsy

Squaner

Architecture

Modèle de Squaner

Détection des motifs

Les modèles de qualité

Analyse

Interface web

Perspective et
conclusion

Questions ?

Analyse de l'existant

Défauts :

- ▶ Point d'entrée, les courriels
- ▶ Appel des outils d'analyse
- ▶ Utilisation des web-services

Gutsy

Nouvelle architecture proposée

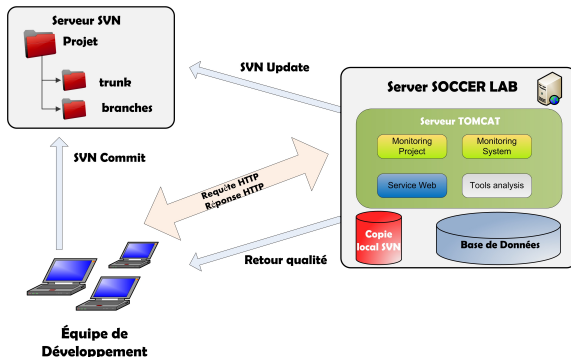


Figure: Nouvelle architecture de Gusty

Gutsy

Organisation des outils d'analyses

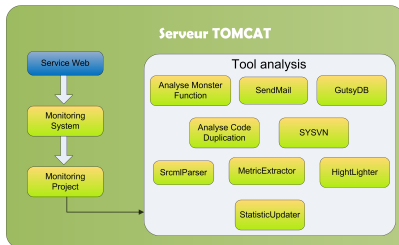


Figure: Architecture Tomcat

Gutsy

1) Téléchargement du code source

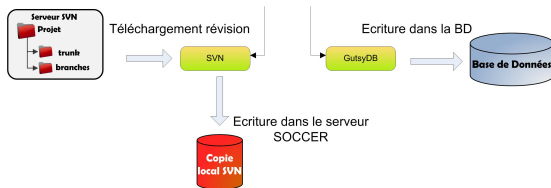


Figure: Étape 1 : Téléchargement du code source

Guttsy

2) Analyse du code source

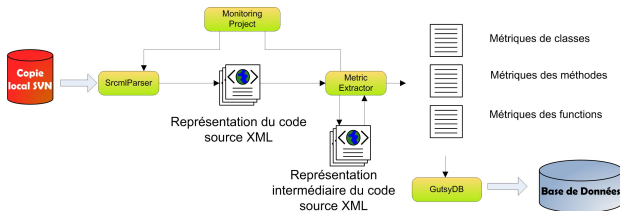


Figure: Étape 2 : Analyse du code source

3) Analyse de la qualité du code

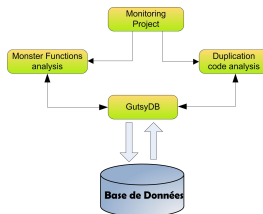


Figure: Étape 3 : Calcul qualité

Gutsy

Introduction

Pourquoi mesurer la qualité
logicielle ?

La qualité logicielle

Monitorat du logiciel

Analyse de l'existant

Gutsy

Squaner

Architecture

Modèle de Squaner

Détection des motifs

Les modèles de qualité

Analyse

Interface web

Perspective et conclusion

Questions ?

4) Retour d'information sur la qualité du code

```

Hello mmetz,

here's a report on your recent grass-SVN change for the revision : 43312

ANALYSIS

your changes to files/class/methods/functions :

*****
File :/include/iostream/ami_stream.h
*****
Class
AMLSTREAMCT>(Begins at: -9999, Ends at: -9999)
+-----+
| DIT | NOC | WMC | RFC | CBO | LCOM |
+-----+
| NEW Metrics | 1 | 0 | 38 | 66 | 0 | 66 |
+-----+
| OLD Metrics | 1 | 0 | 38 | 66 | 0 | 66 |
+-----+

Method
stream_len(Begins at: 382, Ends at:428)
+-----+
| New Metrics | Old Metrics |
+-----+
Complexity: | 4 | | OK | 4 |
ParamNBR: | 1 | | OK | 1 |
CalledNBR: | 20 | * | WARNING | 20 |
LOC: | 47 | | OK | 47 |
+-----+
Conclusion: | this function is a WARNING |
+-----+
at least 1 values are close to an outlier please consider revising code!

```

Figure: Extrait de courriel envoyé

Gutsy

Introduction

Pourquoi mesurer la qualité logicielle ?

La qualité logicielle

Monitorat du logiciel

Analyse de l'existant

Gutsy

Squaner

Architecture

Modèle de Squaner

Détection des motifs

Les modèles de qualité

Analyse

Interface web

Perspective et conclusion

Questions ?

Limitation de Gutsy :

- ▶ ne permet pas d'étudier l'évolution
- ▶ ajout de nouveaux modèles de qualité difficile

Squaner

► Squaner (System QUality AnalyzER)

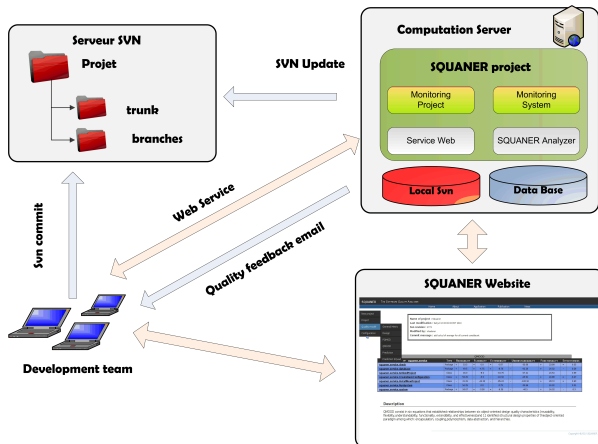


Figure: Architecture de Squaner

Squaner

Introduction

Pourquoi mesurer la qualité
logicielle ?

La qualité logicielle

Monitorat du logiciel

Analyse de l'existant

Gutsy

Squaner

Architecture

Modèle de Squaner

Détection des motifs

Les modèles de qualité

Analyse

Interface web

Perspective et conclusion

Questions ?

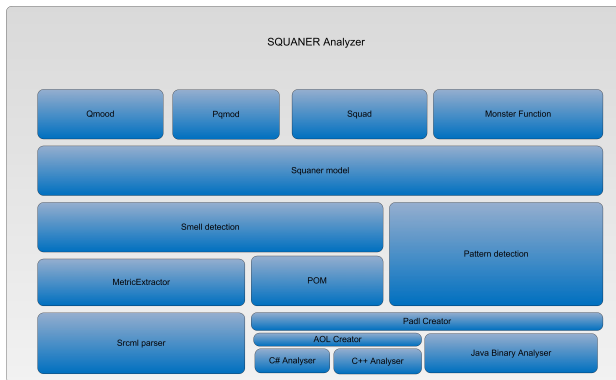


Figure: Architecture des composants de Squaner

Squaner

Modèle représentant l'évaluation d'un système

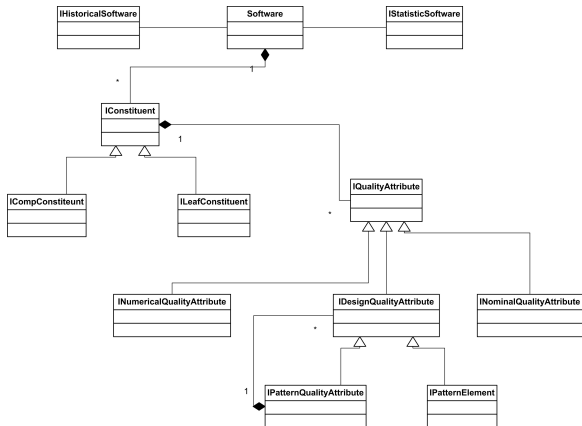


Figure: Squaner modèle

Squaner

Détection des motifs de conception avec Ptidej Framework [3]

- ▶ **PADL Creator** : Analyse du code source
- ▶ **POM** : Détection du code source
- ▶ **Pattern detection** : approche DeMINA [4] pour la détection des patrons de conception.
 - ▶ détection de 13 motifs : (*Abstract Factory, Adapter, Command, Composite, Decorator, Factory Method, Observer, Prototype, Singleton, State, Strategy, Template Method, and Visitor.*).
- ▶ **Smell detection** : approche DECOR [7] pour la détection des anti-patrons de conception.
 - ▶ détection de 11 motifs : *ClassDataShouldBePrivate (CDSBP), ComplexClass, Large-Class, LazyClass, LongMethod, LongParameterList (LPL), MessageChains, RefusedParentBequest (RPB), SpaghettiCode, SpeculativeGenerality (SG), SwissArmyKnife.*

Introduction

Pourquoi mesurer la qualité
logicielle ?

La qualité logicielle
Monitorat du logiciel

Analyse de
l'existant

Gutsy

Squaner

Architecture
Modèle de Squaner
Détection des motifs

Les modèles de qualité

Analyse

Interface web

Perspective et
conclusion

Questions ?

Squaner

QMOOD : 6 équations pour évaluer 6 caractéristiques des systèmes orientés objet : *réutilisabilité, flexibilité, compréhensibilité, fonctionnalité, extensibilité et efficacité.*

►
$$\text{Réutilisabilité} = -0.25 * \text{Coupling} + 0.25 * \text{Cohesion} + 0.5 * \text{Messaging} + 0.5 * \text{Design Size}$$

► ...

Squaner

Introduction

Pourquoi mesurer la qualité
logicielle ?

La qualité logicielle

Monitorat du logiciel

Analyse de l'existant

Gutsy

Squaner

Architecture

Modèle de Squaner

Détection des motifs

Les modèles de qualité

Analyse

Interface web

Perspective et conclusion

Questions ?

PQMOD : 7 règles pour établir une évaluation nominative sur 7 caractéristiques des systèmes orientés objet :
extensibilité, généralité, modularité, Modularité pendant l'exécution, réutilisation, passage à l'échelle et Compréhensibilité.

Squaner

HBugpredict et Bugpredict : modèles de prédiction de bugs

Probability Distribution Table For post

pre	AP	DP	YES	NO
'(-inf-1.5]	twoRole	twoRole	0,094	0,906
'(-inf-1.5]	twoRole	noRole	0,056	0,944
'(-inf-1.5]	twoRole	oneRole	0,025	0,975
'(-inf-1.5]	oneRole	twoRole	0,079	0,921
'(-inf-1.5]	oneRole	noRole	0,097	0,903
'(-inf-1.5]	oneRole	oneRole	0,036	0,964
'(-inf-1.5]	noRole	twoRole	0,035	0,965
'(-inf-1.5]	noRole	noRole	0,017	0,983
'(-inf-1.5]	noRole	oneRole	0,042	0,958
'(1.5-6.5]	twoRole	twoRole	0,25	0,75
'(1.5-6.5]	twoRole	noRole	0,5	0,5
'(1.5-6.5]	twoRole	oneRole	0,5	0,5
'(1.5-6.5]	oneRole	twoRole	0,25	0,75
'(1.5-6.5]	oneRole	noRole	0,5	0,5
'(1.5-6.5]	oneRole	oneRole	0,5	0,5
'(1.5-6.5]	noRole	twoRole	0,062	0,938
'(1.5-6.5]	noRole	noRole	0,1	0,9
'(1.5-6.5]	noRole	oneRole	0,5	0,5
'(6.5-inf]	twoRole	twoRole	0,25	0,75
'(6.5-inf]	twoRole	noRole	0,25	0,75
'(6.5-inf]	twoRole	oneRole	0,5	0,5

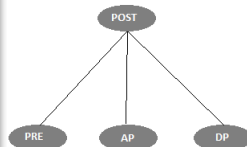


Figure: Réseau Bayésien du modèle HBugPredict

Introduction

Pourquoi mesurer la qualité
logicielle ?

La qualité logicielle

Monitorat du logiciel

Analyse de l'existant

Gutsy

Squaner

Architecture

Modèle de Squaner

Détection des motifs

Les modèles de qualité

Analyse

Interface web

Perspective et conclusion

Questions ?

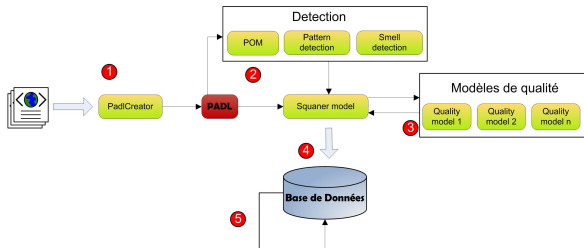


Figure: Étapes pour l'analyse complète du code source

Interface web

Introduction

Pourquoi mesurer la qualité
logicielle ?

La qualité logicielle
Monitorat du logiciel

Analyse de l'existant

Gutsy

Squaner

Architecture
Modèle de Squaner
Détection des motifs
Les modèles de qualité
Analyse

Interface web

Perspective et
conclusion

Questions ?

Introduction

Pourquoi mesurer la qualité
logicielle ?

La qualité logicielle
Monitorat du logiciel

Analyse de
l'existant

Gutsy

Squaner

Architecture
Modèle de Squaner
Détection des motifs
Les modèles de qualité
Analyse

Interface web

**Perspective et
conclusion**

Questions ?

Perspectives et conclusion

- Analyse de l'impact des motifs de conception sur la qualité du code
- Validation de modèle de qualité
- Amélioration des outils de détection

Introduction

Pourquoi mesurer la qualité
logicielle ?

La qualité logicielle
Monitorat du logiciel

Analyse de l'existant

Gutsy

Squaner

Architecture
Modèle de Squaner
Détection des motifs
Les modèles de qualité
Analyse

Interface web

Perspective et
conclusion

Questions ?

Questions ?

Merci pour votre attention !



J. Bansiya and CG Davis.

A hierarchical model for object-oriented design quality assessment.

IEEE Transactions on Software Engineering, pages 4–17, 2002.



M.W. Godfrey and Q. Tu.

Evolution in open source software: A case study.

In *Proceedings of the International Conference on Software Maintenance*, pages 131–142. Citeseer, 2000.



Y.G. Guéhéneuc.

Ptidej: Promoting patterns with patterns.

In *European Conference on Object Oriented Programming, workshop on Building a System with Patterns, Glasgow, Scotland*. Citeseer, 2005.



Y.G. Guéhéneuc and G. Antoniol.

DeMIMA: A multi-layered framework for design pattern identification.

Introduction

Pourquoi mesurer la qualité
logicielle ?

La qualité logicielle
Monitorat du logiciel

Analyse de
l'existant

Gutsy

Squaner

Architecture
Modèle de Squaner
Détection des motifs
Les modèles de qualité
Analyse

Interface web

Perspective et
conclusion

Questions ?

Transactions on Software Engineering, 34(5):667–684,
2008.



Foutse Kohmh.

Patron et qualité des programmes orienté objet.

PhD thesis, Faculté des arts et de la science, Université
de Montréal, Avril 2010.



M. Lehman.

Laws of software evolution revisited.

Software process technology, pages 108–124.



N. Moha, Y.G. Guéhéneuc, L. Duchien, and A.F.

Le Meur.

DECOR: A method for the specification and detection of
code and design smells.



oumoumsack.

La qualité logiciel.

<http://lil.univ-littoral.fr/oumoumsack>.



D.L. Parnas.

Software aging.

In *Proceedings of the 16th international conference on Software engineering*, pages 279–287. IEEE Computer Society Press, 1994.



S.R. Schach, B. Jin, D.R. Wright, G.Z. Heller, and A.J. Offutt.

Maintainability of the Linux kernel.

IEE Proceedings-Software, 149(1):18–23, 2002.