Retour d'expérience sur quelques outils d'audit de code CGI en C

Anne Cheylus

Plan

- Contexte
- Risques
- Gcc
- Sonar + plugin officiel cpp
- Sonar + community plugin
  - Valgrind
  - Vera++
  - Gcov/gcovr
  - Rats
- Bilan

Contexte

Expériences comportementales en sciences cognitives

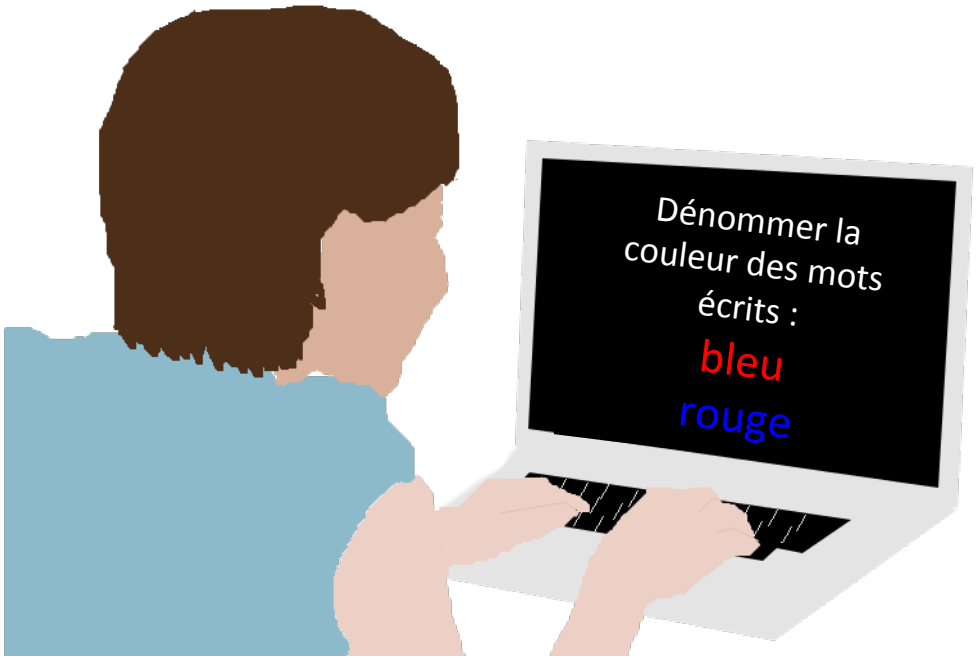
Différentes conditions expérimentales

Choix, temps de réaction, clics, éléments survolés…

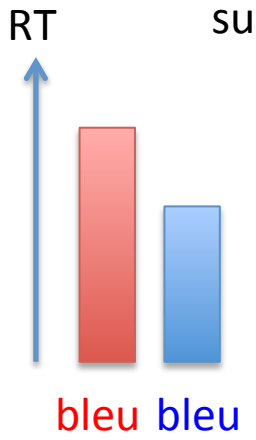
Contexte

Expériences comportementales : exemple : effet de Stroop (1935)

Différentes conditions expérimentales
Couleur congruente : bleu
ou non : bleu



Choix, temps de réaction, clics, éléments survolés…



Contexte

Expériences comportementales en ligne : avantages attendus

Plus de sujets

Meilleur échantillonnage

Accès à d'autres populations

inconvénients attendus

Tâches à adapter

Ordinateurs hétérogènes

Sujets inattentifs

Sujets en double

Robots

Contexte

Expériences comportementales en ligne : solutions existantes

Mechanical Turk d'amazon

Qualtrics

www.soscisurvey.de

Etc.

solution retenue

Développement maison

Linux, apache, html5/css3/ javascript, CGI en C

1 seul dev, budget minimaliste

Risques

Risques affectant la qualité des données recueillies

- Bugs, erreurs de conception
- Sujets sournois
- Robots
- Sujets inattentifs
- Déséquilibre du Plan d'Exp.
- Terminal imprévu

Risques d'atteinte au SI

- Disque plein
- Mémoire insuffisante
- Failles de l'OS
- Failles apache
- Failles CGI
  - Entrées détournées
  - Buffer overflow
  - Exécution de commandes
  - Lecture de données privées
  - Ecritures inappropriées

Risques pour mes CGI

Risques affectant la qualité des données recueillies

- Bugs, erreurs de conception
- Sujets sournois
- Robots
- Sujets inattentifs
- Déséquilibre du Plan d'Exp.
- Terminal imprévu

Risques d'atteinte au SI

- Disque plein
- Mémoire insuffisante
- Failles de l'OS
- Failles apache
- Failles CGI
  - Entrées détournées
  - Buffer overflow
  - Exécution de commandes
  - Lecture de données privées
  - Ecritures inappropriées

Risques

Analyse par risque :

- Probabilité d’occurrence ?
- Conséquences ?
- Traitement ?

Risques

Bugs, erreurs de conception

- Probabilité d'occurrence : élevée
- Conséquences : données faussées ou impossibles à analyser, risque engageant la crédibilité des chercheurs si un résultat faux est publié, évolutions du code de plus en plus pénibles
- Traitement : bonnes pratiques de programmation, tests unitaires et d'intégration automatisés

Risques

Déséquilibre du plan d'expérience

- Probabilité d'occurrence : élevée
- Conséquences : données faussées ou impossibles à analyser, risque engageant la crédibilité des chercheurs si un résultat faux est publié
- Traitement : l'un des CGI donne l'indice de pseudo-randomisation à appliquer, tests d'intégration automatisé

Risques

Disque plein

- Probabilité d'occurrence : modérée
- Conséquences : données incomplètes ou perte de données, fonctionnement altéré de l'OS
- Traitement : limiter le nombre de sujets, attribuer un espace disque suffisant, séparé du système, détecter les données incomplètes à l'analyse

Risques

Mémoire insuffisante

- Probabilité d'occurrence : faible
- Conséquences : données incomplètes ou perte de données, fonctionnement altéré de l'OS
- Traitement : limiter le nombre de connexions simultanées, contrôler les besoins en mémoire, adapter l'architecture, détecter les données incomplètes à l'analyse

Risques

Failles CGI : entrées détournées

- Probabilité d'occurrence : % popularité site
- Conséquences : données faussées, exploitation d'autres failles
- Traitement : contrôler la cohérence des entrées, interdire les caractères hors [A-Za-z0-9] lorsque c'est possible. Faire des tests automatiques pour s'assurer de l'efficacité de ces interdictions.

Risques

Failles CGI : buffer overflow

- Probabilité d'occurrence : élevée en C sans contrôles
- Conséquences : crash, exploitation d'autres failles
- Traitement : contrôler les allocations et la conformité avec les lectures/écritures en mémoire. Choisir des fonctions qui font ces contrôles (strncpy plutôt que strcpy…). Vérifier les calculs risquant d'aboutir à un débordement d'entier.

Gcc

- `gcc –Wall –Wextra` réalise déjà quelques tests automatiques à la compilation :

```c
#include <stdio.h>

int main(int argc, char *argv[])
{
    int variable_inutile=0;
    char * pointeur_non_initialise;

    pointeur_non_initialise[0]=0;
}
```

```
$ gcc -Wall -Wextra test.c -o tests
test.c:5:9: warning: unused variable 'variable_inutile' [-Wunused-variable]
    int variable_inutile=0;
        ^
test.c:3:14: warning: unused parameter 'argc' [-Wunused-parameter]
int main(int argc, char *argv[])
             ^
test.c:3:26: warning: unused parameter 'argv' [-Wunused-parameter]
int main(int argc, char *argv[])
                         ^
test.c:8:5: warning: variable 'pointeur_non_initialise' is uninitialized when
used here [-Wuninitialized]
    pointeur_non_initialise[0]=0;
    ^~~~~~~~~~~~~~~~~~~~~~~~~~
test.c:6:35: note: initialize the variable 'pointeur_non_initialise' to silence
this warning
    char * pointeur_non_initialise;
                                  ^
                                   = NULL
4 warnings generated.
```

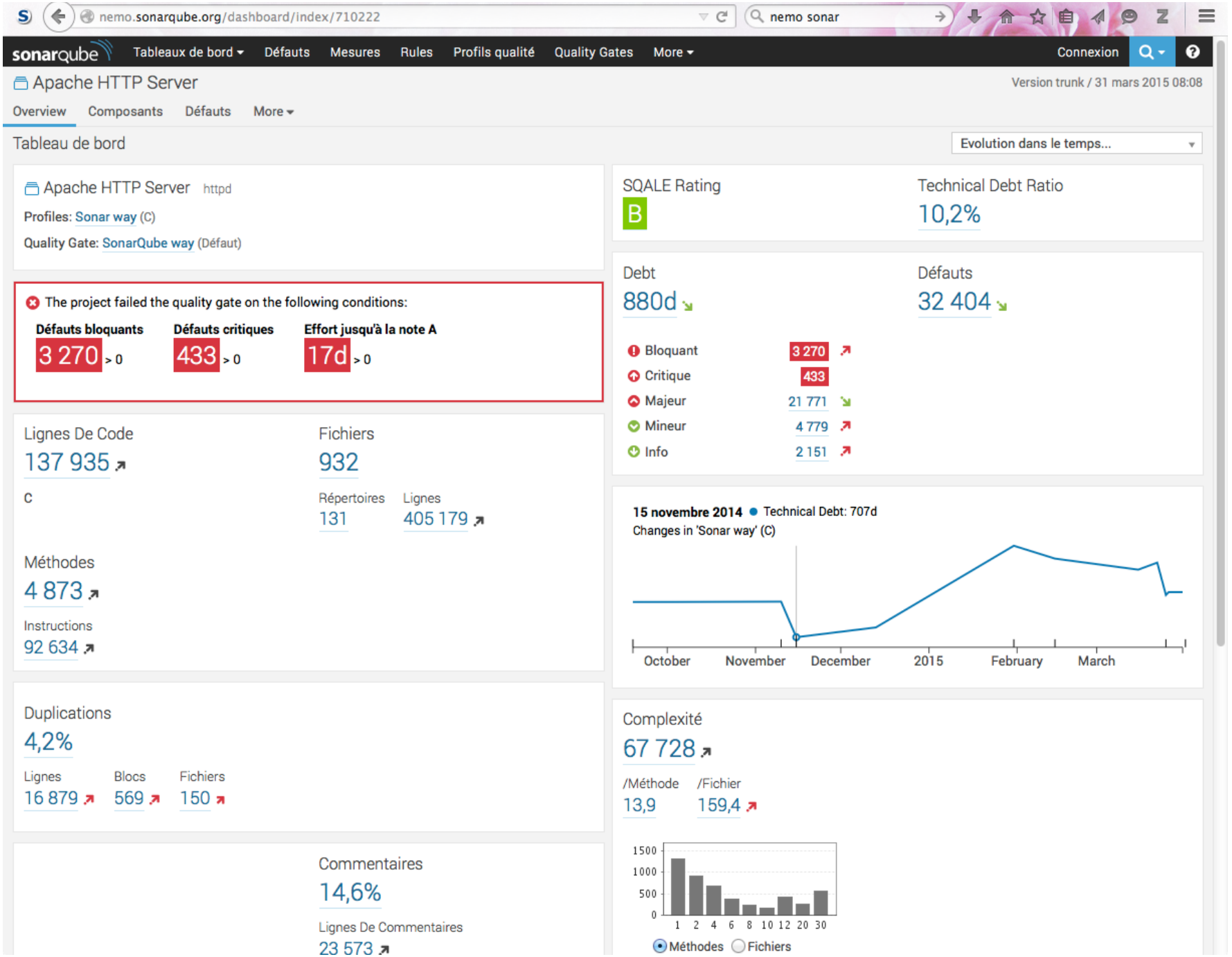
```c
#include <stdio.h>

int main(int argc, char *argv[])
{
    int variable_inutile=0;
    char * pointeur_non_initialise;

    pointeur_non_initialise[0]=0;
}
```

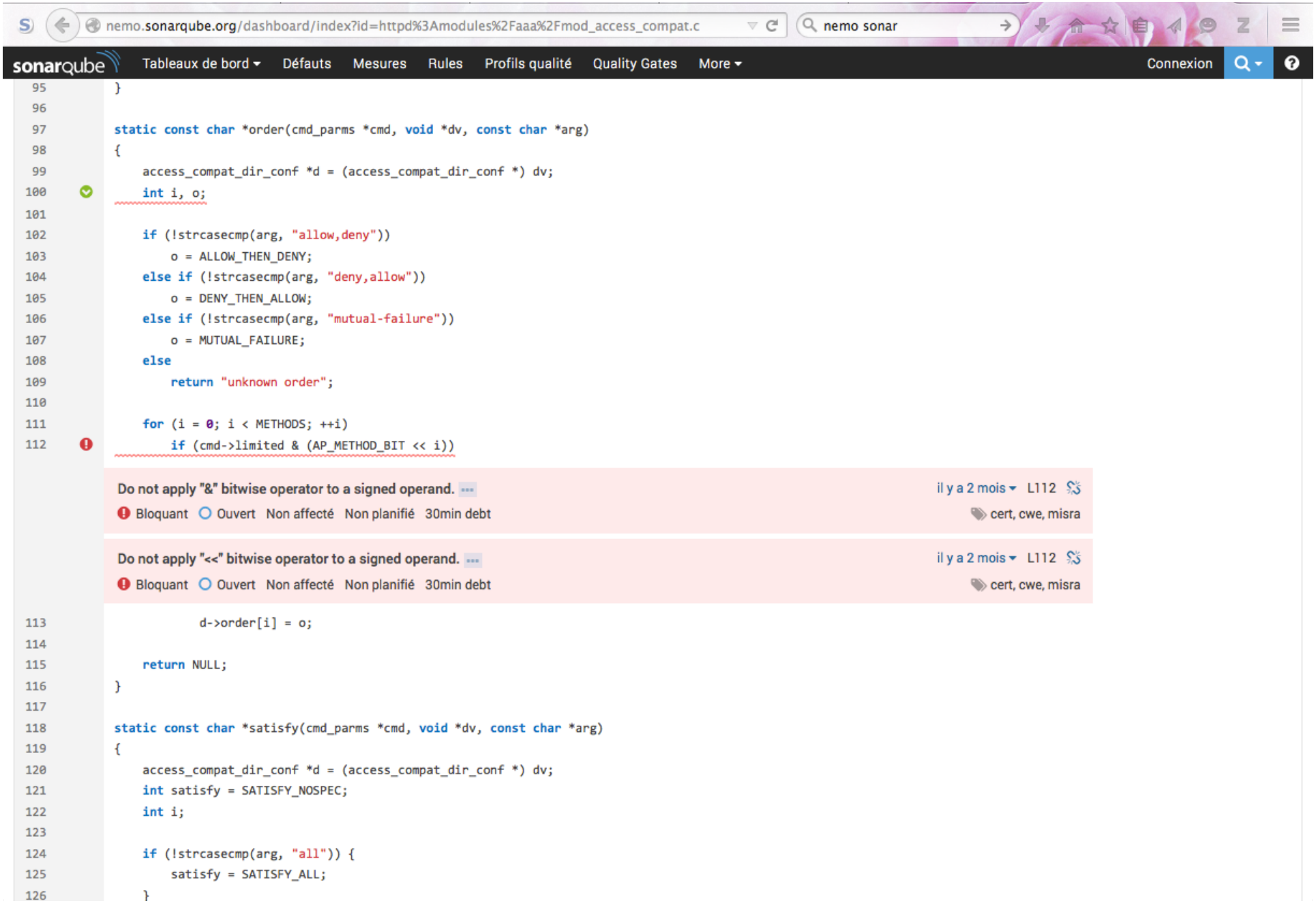
SonarQube

- Permet de regrouper sur une page web un bilan d'analyse de la qualité du code selon différentes métriques
- Licence LGPL v3
- http://docs.sonarqube.org/display/SONAR/Setup+and+Upgrade donne un tutoriel pour une installation d'évaluation rapide
- Il faut ajouter un plugin pour l'analyse de code en C

Plugin cpp

- Commercial, version d'évaluation 2 semaines
- MISRA-C : Motor Industry Software Reliability Association
- Version de démo sur des logiciels libres : http://nemo.sonarqube.org


nemo.sonarqube.org/dashboard/index/710222
nemo sonar
sonarqube
Tableaux de bord
Défauts
Mesures
Rules
Profils qualité
Quality Gates
More
Connexion
Apache HTTP Server
Version trunk / 31 mars 2015 08:08
Overview
Composants
Défauts
More
Tableau de bord
Evolution dans le temps...
Apache HTTP Server httpd
Profiles: Sonar way (C)
Quality Gate: SonarQube way (Défaut)
The project failed the quality gate on the following conditions:
Défauts bloquants
3 270 > 0
Défauts critiques
433 > 0
Effort jusqu'à la note A
17d > 0
SQALE Rating
B
Technical Debt Ratio
10,2%
Debt
880d
Défauts
32 404
Bloquant 3 270
Critique 433
Majeur 21 771
Mineur 4 779
Info 2 151
Lignes De Code
137 935
C
Fichiers
932
Répertoires
131
Lignes
405 179
Méthodes
4 873
Instructions
92 634
15 novembre 2014 Technical Debt: 707d
Changes in 'Sonar way' (C)
October
November
December
2015
February
March
Duplications
4,2%
Lignes
16 879
Blocs
569
Fichiers
150
Complexité
67 728
/Méthode
13,9
/Fichier
159,4
1500
1000
500
0
1 2 4 6 8 10 12 20 30
Méthodes
Fichiers
Commentaires
14,6%
Lignes De Commentaires
23 573


nemo.sonarqube.org/dashboard/index?id=httpd%3Amodules%2Faaa%2Fmod_access_compat.c
nemo sonar
sonarqube
Tableaux de bord
Défauts
Mesures
Rules
Profils qualité
Quality Gates
More
Connexion
}
static const char *order(cmd_parms *cmd, void *dv, const char *arg)
{
access_compat_dir_conf *d = (access_compat_dir_conf *) dv;
int i, o;
if (!strcasecmp(arg, "allow,deny"))
o = ALLOW_THEN_DENY;
else if (!strcasecmp(arg, "deny,allow"))
o = DENY_THEN_ALLOW;
else if (!strcasecmp(arg, "mutual-failure"))
o = MUTUAL_FAILURE;
else
return "unknown order";
for (i = 0; i < METHODS; ++i)
if (cmd->limited & (AP_METHOD_BIT << i))
Do not apply "&" bitwise operator to a signed operand.
il y a 2 mois
L112
Bloquant
Ouvert
Non affecté
Non planifié
30min debt
cert, cwe, misra
Do not apply "<<" bitwise operator to a signed operand.
il y a 2 mois
L112
Bloquant
Ouvert
Non affecté
Non planifié
30min debt
cert, cwe, misra
d->order[i] = o;
return NULL;
}
static const char *satisfy(cmd_parms *cmd, void *dv, const char *arg)
{
access_compat_dir_conf *d = (access_compat_dir_conf *) dv;
int satisfy = SATISFY_NOSPEC;
int i;
if (!strcasecmp(arg, "all")) {
satisfy = SATISFY_ALL;
}

Plugin cpp

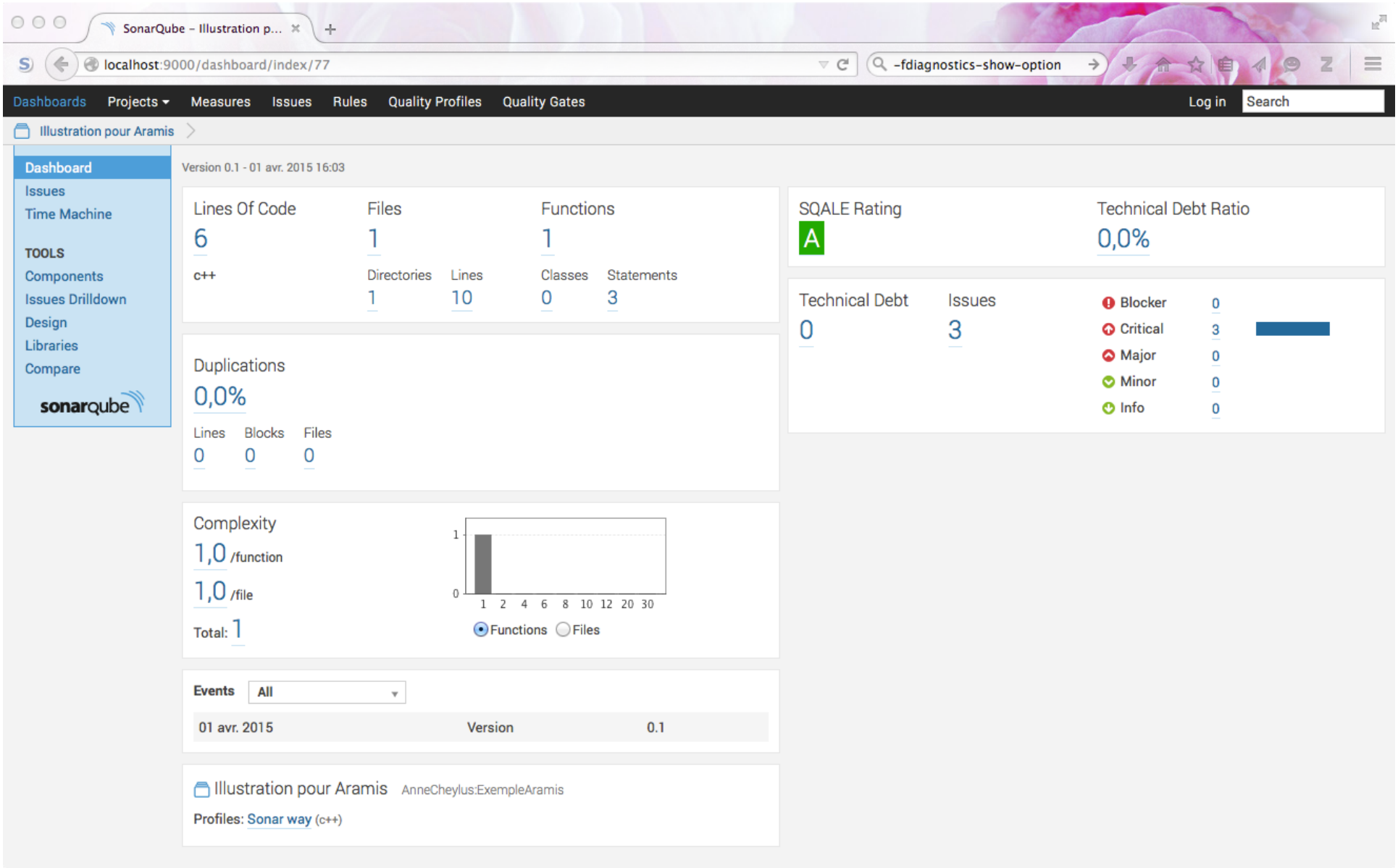
Sur mon projet (peu de temps pour tester) 😐

- Espaces en fin de ligne
- Tabulations (au lieu de 4 espaces)
- Non respect d'une convention de nommage
- Code commenté
- mot clé `TODO` dans les commentaires
- Utilisation de plusieurs `return()` dans une même fonction, de `continue()` ou de `break()`
- Fonctions trop complexes
- Variables magiques (`int n=2;`)
- Utilisation de constantes octales (`mkdir(path,0700)`)

Plugin community cxx

- https://github.com/wenns/sonar-cxx
- Installations et configuration supplémentaires pour chaque module d'analyse :
  - Lecture des sorties du compilateurs

```
$ gcc -Wall -Wextra test.c -o tests > gcc.log 2>&1
$ sonar-runner
SonarQube Runner 2.4
Java 1.6.0_65 Apple Inc. (64-bit)
Mac OS X 10.9.5 x86_64
INFO: Runner configuration file: /Users/cheylus/Documents/Anne/sonar/sonar-runner-2.4/conf/sonar-runner.properties
INFO: Project configuration file: /Users/cheylus/Documents/Anne/ExemplesAramis2015/sonar-project.properties
INFO: Default locale: "fr_FR", source code encoding: "UTF-8"
INFO: Work directory: /Users/cheylus/Documents/Anne/ExemplesAramis2015/./.sonar
INFO: SonarQube Server 4.5
16:03:03.999 INFO  - Load global referentials…
 […]
16:03:41.036 INFO  - ANALYSIS SUCCESSFUL, you can browse http://localhost:9000/dashboard/index/AnneCheylus:ExempleAramis
16:03:45.296 INFO  - Executing post-job class org.sonar.plugins.core.issue.notification.SendIssueNotificationsPostJob
16:03:45.435 INFO  - Executing post-job class org.sonar.plugins.core.batch.IndexProjectPostJob
16:03:45.498 INFO  - Executing post-job class org.sonar.plugins.dbcleaner.ProjectPurgePostJob
16:03:45.507 INFO  - -> Keep one snapshot per day between 2015-03-04 and 2015-03-31
16:03:45.508 INFO  - -> Keep one snapshot per week between 2014-04-02 and 2015-03-04
16:03:45.508 INFO  - -> Keep one snapshot per month between 2010-04-07 and 2014-04-02
16:03:45.509 INFO  - -> Delete data prior to: 2010-04-07
16:03:45.513 INFO  - -> Clean Illustration pour Aramis [id=77]
INFO: ------------------------------------------------------------------------
INFO: EXECUTION SUCCESS
INFO: ------------------------------------------------------------------------
Total time: 50.734s
Final Memory: 5M/108M
INFO: ------------------------------------------------------------------------
```


SonarQube - Illustration p...
localhost:9000/dashboard/index/77
-fdiagnostics-show-option
Dashboards
Projects
Measures
Issues
Rules
Quality Profiles
Quality Gates
Log in
Search
Illustration pour Aramis
Dashboard
Issues
Time Machine
TOOLS
Components
Issues Drilldown
Design
Libraries
Compare
sonarqube
Version 0.1 - 01 avr. 2015 16:03
Lines Of Code
6
c++
Files
1
Directories
1
Lines
10
Functions
1
Classes
0
Statements
3
SQALE Rating
A
Technical Debt Ratio
0,0%
Technical Debt
0
Issues
3
Blocker
0
Critical
3
Major
0
Minor
0
Info
0
Duplications
0,0%
Lines
0
Blocks
0
Files
0
Complexity
1,0 /function
1,0 /file
Total: 1
1
0
1 2 4 6 8 10 12 20 30
Functions
Files
Events
All
01 avr. 2015
Version
0.1
Illustration pour Aramis
AnneCheylus:ExempleAramis
Profiles: Sonar way (c++)

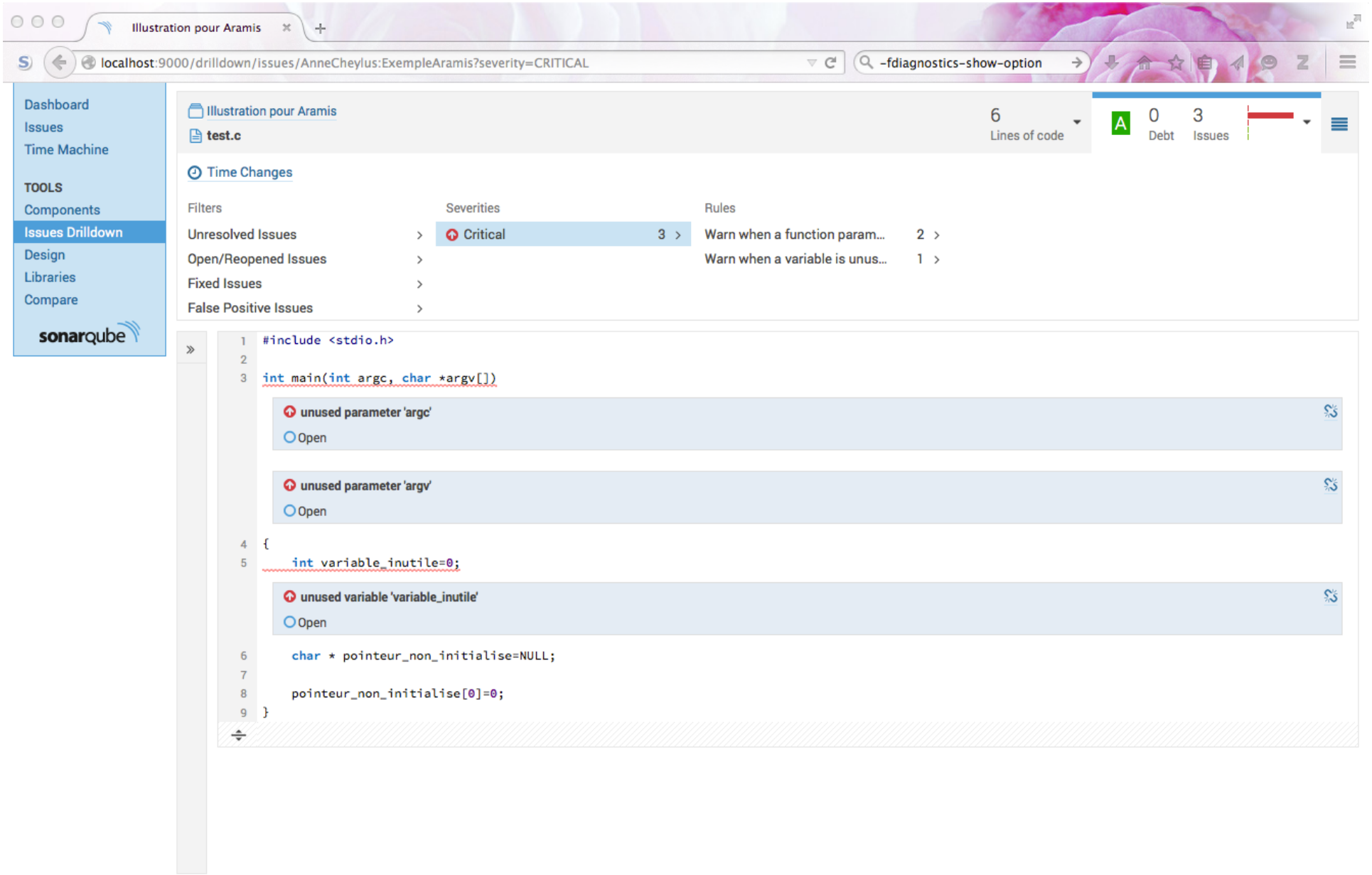


Illustration pour Aramis
localhost:9000/drilldown/issues/AnneCheylus:ExempleAramis?severity=CRITICAL
-fdiagnostics-show-option
Dashboard
Issues
Time Machine
TOOLS
Components
Issues Drilldown
Design
Libraries
Compare
sonarqube
Illustration pour Aramis
test.c
6
Lines of code
A
0
Debt
3
Issues
Time Changes
Filters
Unresolved Issues
Open/Reopened Issues
Fixed Issues
False Positive Issues
Severities
Critical 3
Rules
Warn when a function param... 2
Warn when a variable is unus... 1
1 #include <stdio.h>
2
3 int main(int argc, char *argv[])
unused parameter 'argc'
Open
unused parameter 'argv'
Open
4 {
5 int variable_inutile=0;
unused variable 'variable_inutile'
Open
6 char * pointeur_non_initialise=NULL;
7
8 pointeur_non_initialise[0]=0;
9 }

Plugin community cxx

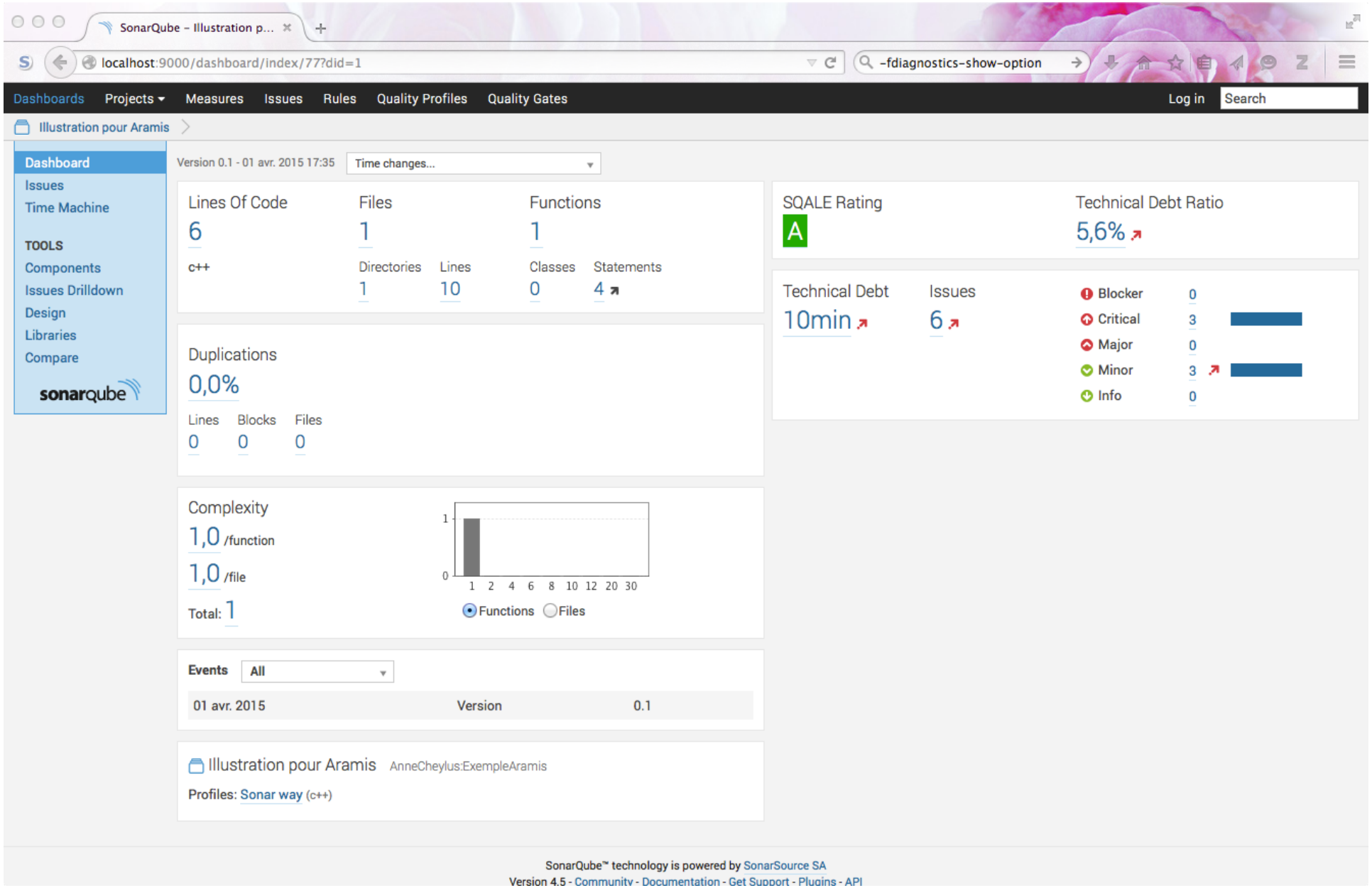
- https://github.com/wenns/sonar-cxx
- Installations et configuration supplémentaires pour chaque module d'analyse :
  - Lecture des sorties du compilateurs
  - Vera++

https://bitbucket.org/verateam/vera Boost Software License

vera++ vérifie si le code suit un certain nombre de règles de programmation, ces règles sont extensibles. Il est également capable de donner des statistiques sur le respect de ces règles ou d'effectuer certaines corrections sur le code source.

Par exemple, mettre un espace après if, ne pas laisser plusieurs lignes vides consécutives ou des espaces en fin de ligne, utiliser des noms de fichiers pas trop longs

```
$ find ./ -name "*.[ch]" | vera++ -s -d | ./veraxml.pl > vera++.xml
$ sonar-runner
```


SonarQube - Illustration p...
localhost:9000/dashboard/index/77?did=1
-fdiagnostics-show-option
Dashboards
Projects
Measures
Issues
Rules
Quality Profiles
Quality Gates
Log in
Search
Illustration pour Aramis
Dashboard
Issues
Time Machine
TOOLS
Components
Issues Drilldown
Design
Libraries
Compare
sonarqube
Version 0.1 - 01 avr. 2015 17:35
Time changes...
Lines Of Code
6
c++
Files
1
Directories
1
Lines
10
Functions
1
Classes
0
Statements
4
SQALE Rating
A
Technical Debt Ratio
5,6%
Technical Debt
10min
Issues
6
Blocker
0
Critical
3
Major
0
Minor
3
Info
0
Duplications
0,0%
Lines
0
Blocks
0
Files
0
Complexity
1,0 /function
1,0 /file
Total: 1
1
0
1 2 4 6 8 10 12 20 30
Functions
Files
Events
All
01 avr. 2015
Version
0.1
Illustration pour Aramis
AnneCheylus:ExempleAramis
Profiles: Sonar way (c++)
SonarQube™ technology is powered by SonarSource SA
Version 4.5 - Community - Documentation - Get Support - Plugins - API

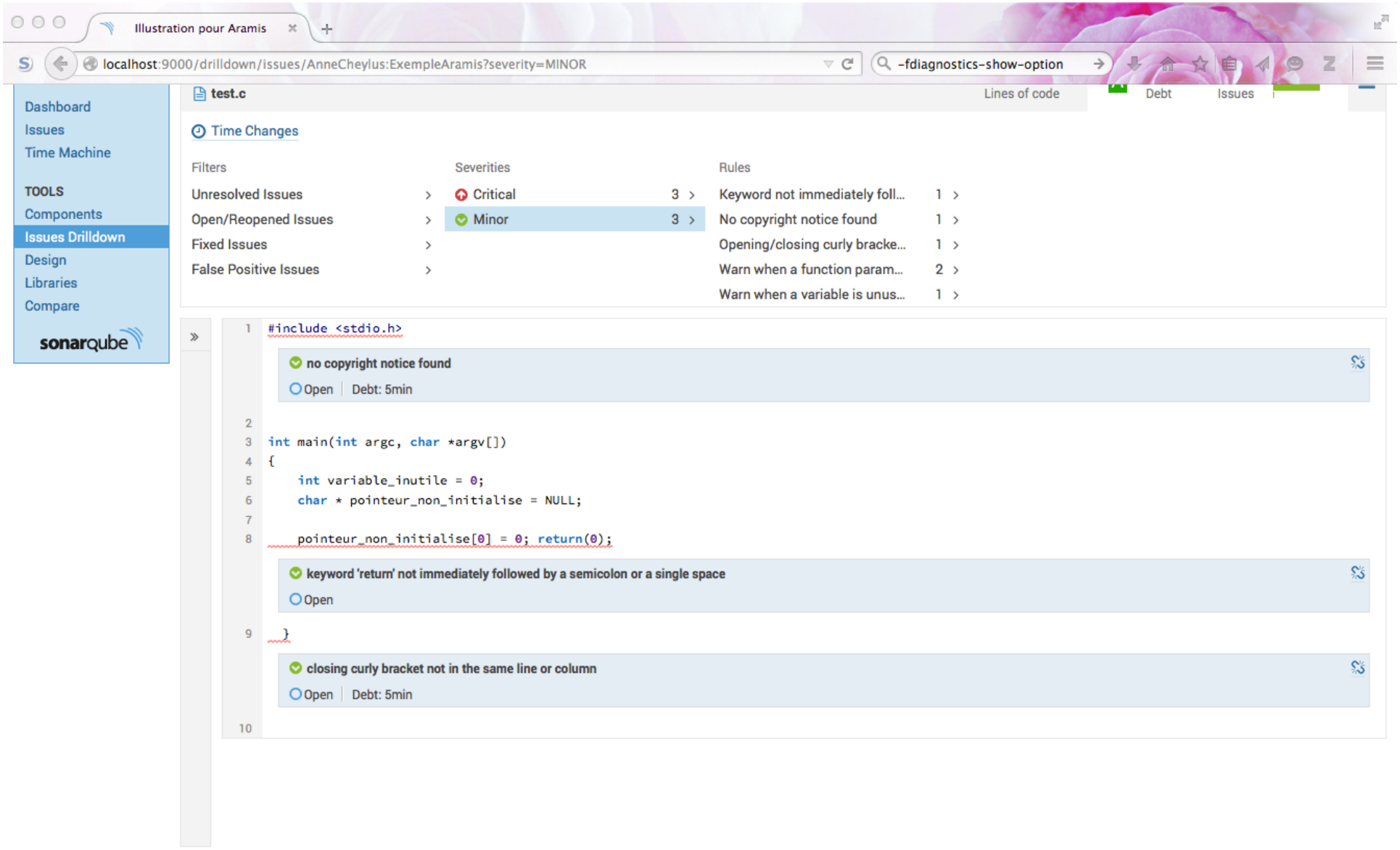


Illustration pour Aramis
localhost:9000/drilldown/issues/AnneCheylus:ExempleAramis?severity=MINOR
-fdiagnostics-show-option
Dashboard
Issues
Time Machine
TOOLS
Components
Issues Drilldown
Design
Libraries
Compare
sonarqube
test.c
Lines of code
Debt
Issues
Time Changes
Filters
Unresolved Issues
Open/Reopened Issues
Fixed Issues
False Positive Issues
Severities
Critical 3
Minor 3
Rules
Keyword not immediately foll... 1
No copyright notice found 1
Opening/closing curly bracke... 1
Warn when a function param... 2
Warn when a variable is unus... 1
1 #include <stdio.h>
no copyright notice found
Open | Debt: 5min
2
3 int main(int argc, char *argv[])
4 {
5 int variable_inutile = 0;
6 char * pointeur_non_initialise = NULL;
7
8 pointeur_non_initialise[0] = 0; return(0);
keyword 'return' not immediately followed by a semicolon or a single space
Open
9 }
closing curly bracket not in the same line or column
Open | Debt: 5min
10
SonarQube™ technology is powered by SonarSource SA
Version 4.5 - Community - Documentation - Get Support - Plugins - API

Plugin community cxx

- https://github.com/wenns/sonar-cxx
- Installations et configuration supplémentaires pour chaque module d'analyse :
  - Lecture des sorties du compilateurs
  - Vera++
  - cppcheck

http://cppcheck.sourceforge.net/ GPLv3

Cppcheck réalise une analyse statique du code source dans le but de mettre en évidence des accès à des espaces mémoire non alloués, une mauvaise gestion de la mémoire, l'utilisation de fonctions obsolètes, du code risqué de par sa formulation.

```
$ cppcheck -v --xml-version=2 test.c 2> test.cppcheck.xml
Checking test.c...
$ sonar-runner
```

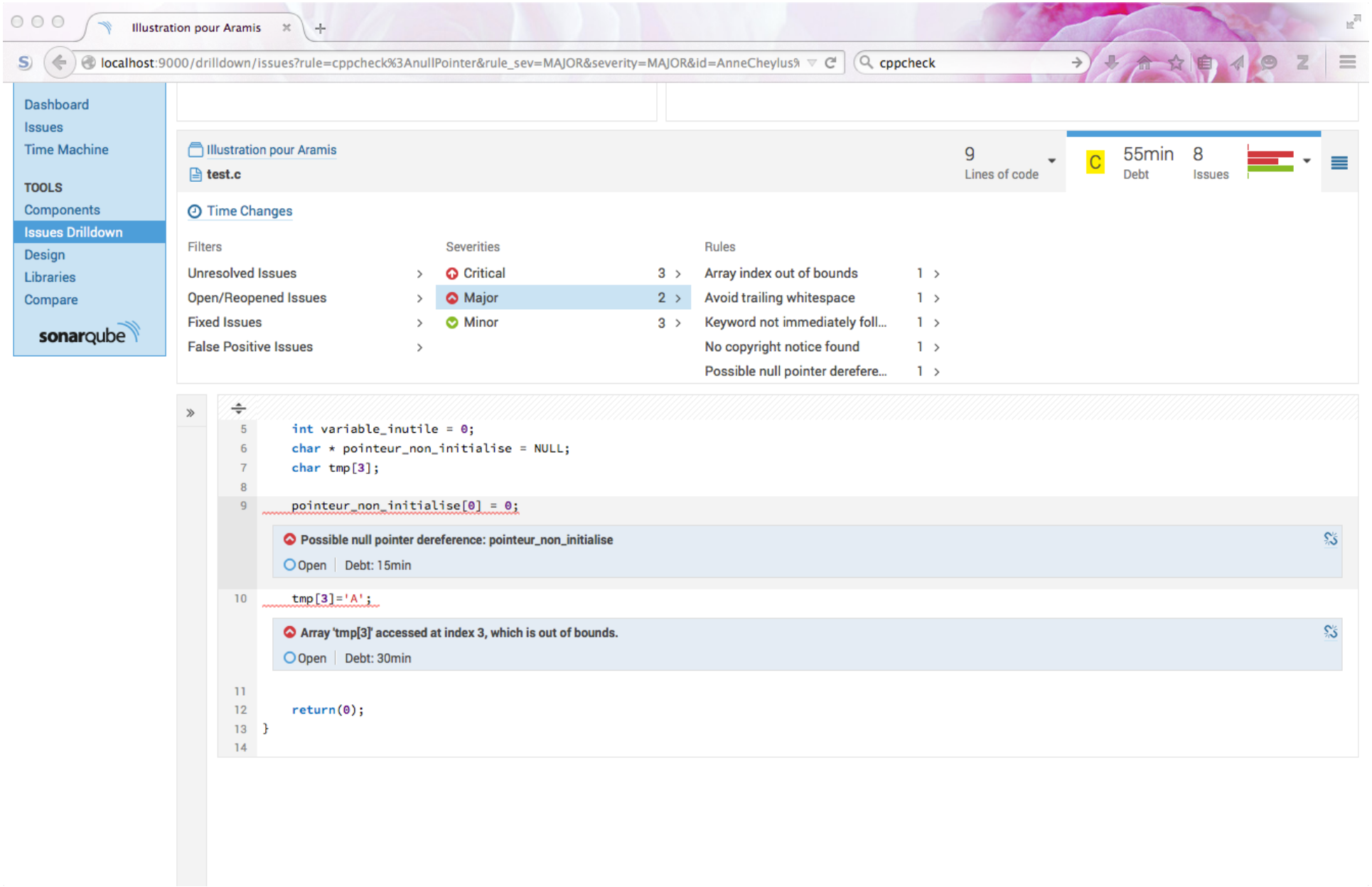


Illustration pour Aramis
localhost:9000/drilldown/issues?rule=cppcheck%3AnullPointer&rule_sev=MAJOR&severity=MAJOR&id=AnneCheylus
cppcheck
Dashboard
Issues
Time Machine
TOOLS
Components
Issues Drilldown
Design
Libraries
Compare
sonarqube
Illustration pour Aramis
test.c
9
Lines of code
C
55min
Debt
8
Issues
Time Changes
Filters
Unresolved Issues
Open/Reopened Issues
Fixed Issues
False Positive Issues
Severities
Critical 3
Major 2
Minor 3
Rules
Array index out of bounds 1
Avoid trailing whitespace 1
Keyword not immediately foll... 1
No copyright notice found 1
Possible null pointer derefere... 1
5 int variable_inutile = 0;
6 char * pointeur_non_initialise = NULL;
7 char tmp[3];
8
9 pointeur_non_initialise[0] = 0;
Possible null pointer dereference: pointeur_non_initialise
Open | Debt: 15min
10 tmp[3]='A';
Array 'tmp[3]' accessed at index 3, which is out of bounds.
Open | Debt: 30min
11
12 return(0);
13 }
14

Plugin community cxx

- https://github.com/wenns/sonar-cxx
- Installations et configuration supplémentaires pour chaque module d'analyse :
  - Lecture des sorties du compilateurs
  - Vera++
  - Cppcheck
  - Valgrind

http://valgrind.org/

Valgrind vérifie les fuites de mémoire dans le programme lors de son exécution. Pour qu'il rapporte correctement les numéros de ligne, il faut avoir compilé avec l'option –g sans optimisation

```
$ gcc -Wall -Wextra test.c -g -o tests > gcc.log 2>&1
$ find ./ -name "*.[ch]" | vera++ -s -d | ./veraxml.pl > vera++.xml
$ cppcheck -v --xml-version=2 test.c 2> test.cppcheck.xml
Checking test.c...
$ valgrind --xml=yes --xml-file=test.valgrind.xml --leak-check=full ./tests
$ sonar-runner
```

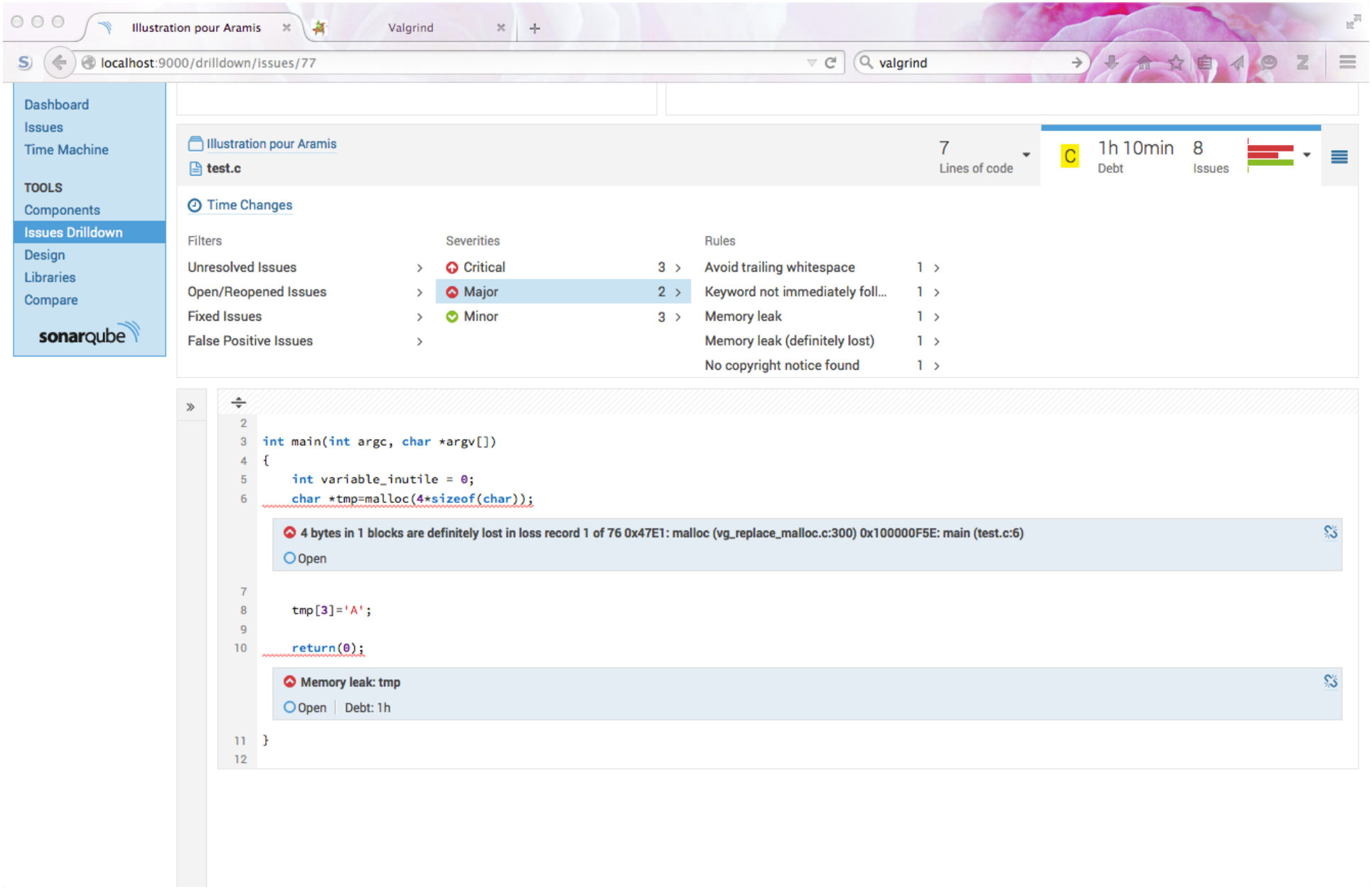


Illustration pour Aramis
Valgrind
localhost:9000/drilldown/issues/77
valgrind
Dashboard
Issues
Time Machine
TOOLS
Components
Issues Drilldown
Design
Libraries
Compare
sonarqube
Illustration pour Aramis
test.c
7
Lines of code
C
1h 10min
Debt
8
Issues
Time Changes
Filters
Unresolved Issues
Open/Reopened Issues
Fixed Issues
False Positive Issues
Severities
Critical 3
Major 2
Minor 3
Rules
Avoid trailing whitespace 1
Keyword not immediately foll... 1
Memory leak 1
Memory leak (definitely lost) 1
No copyright notice found 1
2
3 int main(int argc, char *argv[])
4 {
5 int variable_inutile = 0;
6 char *tmp=malloc(4*sizeof(char));
4 bytes in 1 blocks are definitely lost in loss record 1 of 76 0x47E1: malloc (vg_replace_malloc.c:300) 0x100000F5E: main (test.c:6)
Open
7
8 tmp[3]='A';
9
10 return(0);
Memory leak: tmp
Open | Debt: 1h
11 }
12

Plugin community cxx

- https://github.com/wenns/sonar-cxx
- Installations et configuration supplémentaires pour chaque module d'analyse :
  - Lecture des sorties du compilateurs
  - Vera++
  - Cppcheck
  - Valgrind
  - Gcov/gcovr

gcov/gcovr

Principe :

1. compiler avec l'option `--coverage`
2. Exécuter les tests unitaires/d'intégration
3. Compter le nombre d'exécutions pour chaque ligne de code, chaque embranchement, pour révéler des tests manquants

https://github.com/gcovr/gcovr ©Sandia Corporation

Pour lancer gcovr, il faut compiler avec l'option --coverage

```
$ gcc -Wall -Wextra test.c --coverage -g -o tests > gcc.log 2>&1
$ find ./ -name "*.[ch]" | vera++ -s -d | ./veraxml.pl > vera++.xml
$ cppcheck -v --xml-version=2 test.c 2> test.cppcheck.xml
Checking test.c...
$ valgrind --xml=yes --xml-file=test.valgrind.xml --leak-check=full ./tests
\-/|
$ gcovr -x -r . > gcovr.xml
$ sonar-runner
```

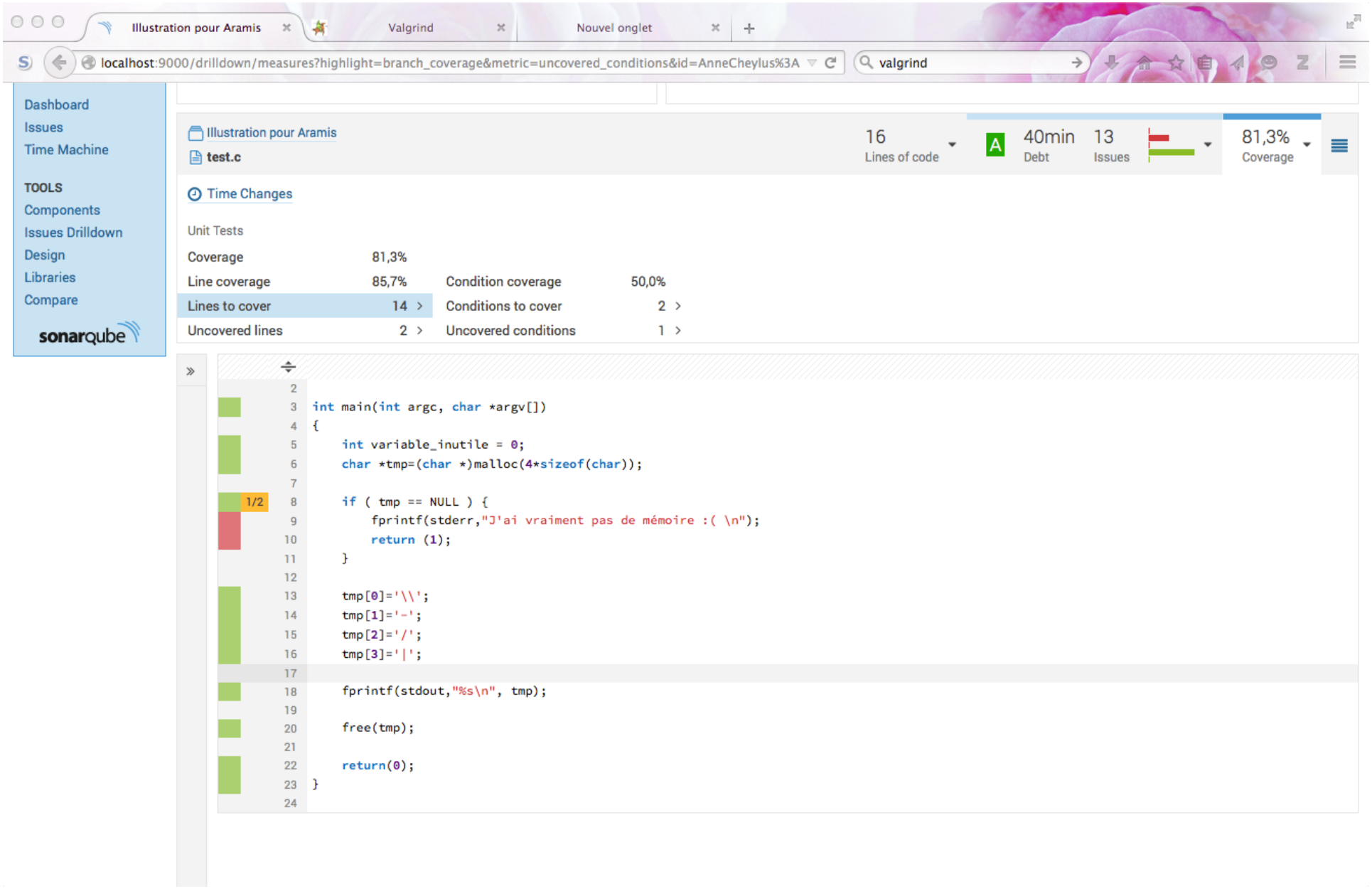


Illustration pour Aramis
Valgrind
Nouvel onglet
localhost:9000/drilldown/measures?highlight=branch_coverage&metric=uncovered_conditions&id=AnneCheylus%3A
valgrind
Dashboard
Issues
Time Machine
TOOLS
Components
Issues Drilldown
Design
Libraries
Compare
sonarqube
Illustration pour Aramis
test.c
16
Lines of code
A
40min
Debt
13
Issues
81,3%
Coverage
Time Changes
Unit Tests
Coverage 81,3%
Line coverage 85,7%
Condition coverage 50,0%
Lines to cover 14
Conditions to cover 2
Uncovered lines 2
Uncovered conditions 1
1/2
int main(int argc, char *argv[])
{
int variable_inutile = 0;
char *tmp=(char *)malloc(4*sizeof(char));
if (tmp == NULL) {
fprintf(stderr,"J'ai vraiment pas de mémoire :(\n");
return (1);
}
tmp[0]='\\';
tmp[1]='-';
tmp[2]='/';
tmp[3]='|';
fprintf(stdout,"%s\n", tmp);
free(tmp);
return(0);
}

Plugin community cxx

- https://github.com/wenns/sonar-cxx
- Installations et configuration supplémentaires pour chaque module d'analyse :
  - Lecture des sorties du compilateurs
  - Vera++
  - Cppcheck
  - Valgrind
  - Gcov/gcovr
  - Rats

rats

Rough Auditing Tool for Security

https://code.google.com/p/rough-auditing-tool-for-security/

Recherche de vulnérabilités répertoriées pouvant affecter la sécurité d'un logiciel : buffer overflow, race condition, TOCTOU (développement datant de 2009 avec un patch en 2013)

```
$ gcc -Wall -Wextra test.c --coverage -g -o tests > gcc.log 2>&1
$ find ./ -name "*.[ch]" | vera++ -s -d | ./veraxml.pl > vera++.xml
$ cppcheck -v --xml-version=2 test.c 2> test.cppcheck.xml
Checking test.c...
$ valgrind --xml=yes --xml-file=test.valgrind.xml --leak-check=full ./tests
\-/|
$ sed -e "s/Aramis2015/Aramis2015\/./g" -i "" test.valgrind.xml
$ gcovr -x -r . > gcovr.xml
$ rats -w 3 --xml ./test.c > rats.xml
$ sonar-runner
```

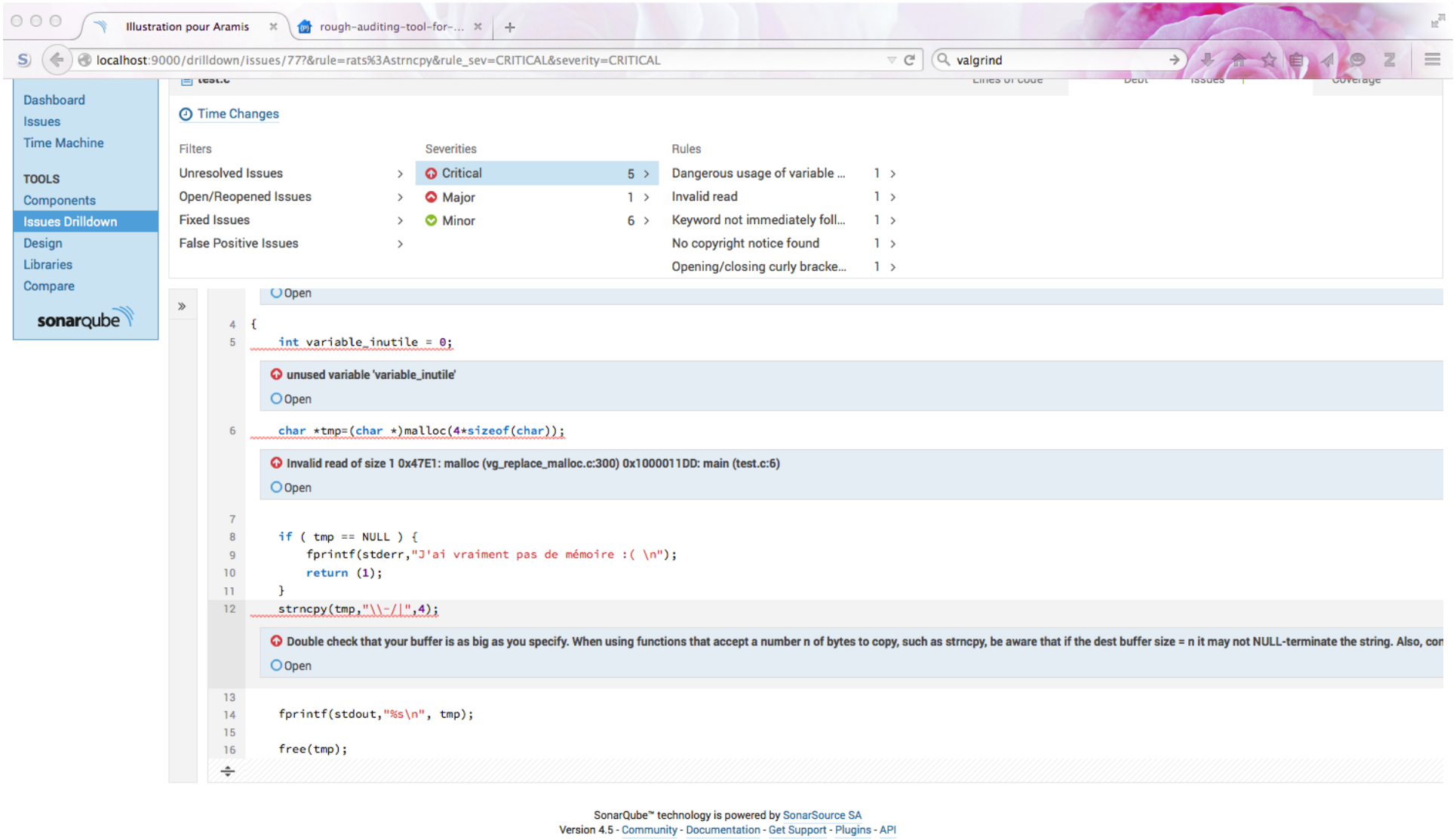


Illustration pour Aramis
rough-auditing-tool-for-...
localhost:9000/drilldown/issues/77?&rule=rats%3Astrncpy&rule_sev=CRITICAL&severity=CRITICAL
valgrind
Dashboard
Issues
Time Machine
TOOLS
Components
Issues Drilldown
Design
Libraries
Compare
sonarqube
Time Changes
Filters
Unresolved Issues
Open/Reopened Issues
Fixed Issues
False Positive Issues
Severities
Critical 5
Major 1
Minor 6
Rules
Dangerous usage of variable ... 1
Invalid read 1
Keyword not immediately foll... 1
No copyright notice found 1
Opening/closing curly bracke... 1
Open
{
int variable_inutile = 0;
unused variable 'variable_inutile'
Open
char *tmp=(char *)malloc(4*sizeof(char));
Invalid read of size 1 0x47E1: malloc (vg_replace_malloc.c:300) 0x1000011DD: main (test.c:6)
Open
if (tmp == NULL) {
fprintf(stderr,"J'ai vraiment pas de mémoire :(\n");
return (1);
}
strncpy(tmp,"\\-/|",4);
Double check that your buffer is as big as you specify. When using functions that accept a number n of bytes to copy, such as strncpy, be aware that if the dest buffer size = n it may not NULL-terminate the string. Also, con
Open
fprintf(stdout,"%s\n", tmp);
free(tmp);
SonarQube™ technology is powered by SonarSource SA
Version 4.5 - Community - Documentation - Get Support - Plugins - API

Bilan

- Découverte de plusieurs défauts dans le code
- Une fois l'installation effectuée et le process automatisé, retour rapide sur certaines erreurs de programmation ou de conception
- La première installation prend du temps
- Les solutions commerciales sont très chères pour un usage « personnel »
- En intégration continue, il existe un risque de se sentir conforté dans ses erreurs par cette multitude de tests qui n'est pas pour autant exhaustive.

Liens utiles

- https://www.projet-plume.org/fiche/sonar
- https://aresu.dsi.cnrs.fr/IMG/pdf/failles_de_securite_v1-3.pdf
- https://www.projet-plume.org/fr/ressource/owasp-open-web-application-security-project-securite-web
- https://www.owasp.org
- http://www.sans.org/top25-software-errors/