

Compilation

Tests unitaires

Documentation

Profiling

QA

Intégration continue

# Outils

Sylvain GROSDÉMOUGE



## Compilation

Build configuration

Plateforme

Configuration

Complexité

Autobuild

Intérêt

Mise en place

Tests unitaires

Documentation

Profiling

QA

Intégration continue

## Build configuration

Permet de stocker, de manière nommée, un set d'options de compilation et de link pour un projet donné.

Sur un développement logiciel, nous avons souvent :

- Plusieurs plateformes
- Plusieurs configurations par plateforme

## Compilation

Build configuration

Plateforme

Configuration

Complexité

CMake

Autobuild

Intérêt

Mise en place

Tests unitaires

Documentation

Profiling

QA

Intégration continue

## Plateforme

Plateformes :

Windows (... , XP, W7, W8, W8.1, ...)

Linux (... , Fedora, Mandriva, Gentoo, Ubuntu, ...)

Android (Ice Cream Sandwich, JellyBean, KitKat, ...)

...

Au niveau du code :

```
#if defined(_WINDOWS)
```

```
...
```

```
#elif defined(...)
```

```
...
```

```
#endif
```

→ Nécessité d'avoir une partie du code source qui diffère selon la plateforme visée

## Compilation

Build configuration

Plateforme

Configuration

Complexité

CMake

Autobuild

Intérêt

Mise en place

Tests unitaires

Documentation

Profiling

QA

Intégration continue

## Configuration

Debug :

*Exécution moins rapide mais plus facile à debugger*

No optimization

Debug informations = on

...

Release :

*Exécution rapide mais moins facile à debugger*

Optimization Level 4

Debug informations = off

...

Profile :

Version permettant l'utilisation d'outils de mesure des performances.

Master :

Version finale du produit, distribuable (i.e. protection, gestion du numéro de série, ...)

## Compilation

Build configuration

Plateforme

Configuration

Complexité

CMake

Autobuild

Intérêt

Mise en place

Tests unitaires

Documentation

Profiling

QA

Intégration continue

## Complexité

Nombre de plateformes \* Nombre de configurations

Complexité :  $N^2$

Exemple : WINDOWS | LINUX | IOS | ANDROID

\*

DEBUG | RELEASE

= 8

WINDOWS(4) | LINUX(4) | IOS(1) | ANDROID(2)

\*

DEBUG | RELEASE | PROFILE | MASTER

= 44 !

→ Nécessité d'être capable d'automatiser les builds

# OUTILS

## Compilation

Build configuration

- Plateforme

- Configuration

- Complexité

CMake

Autobuild

- Intérêt

- Mise en place

Tests unitaires

Documentation

Profiling

QA

Intégration continue

## CMake

Permet de générer des configurations de compilation pour un certain nombre de plateformes.

# OUTILS

## Compilation

Build configuration

Plateforme

Configuration

Complexité

CMake

Autobuild

Intérêt

Mise en place

Tests unitaires

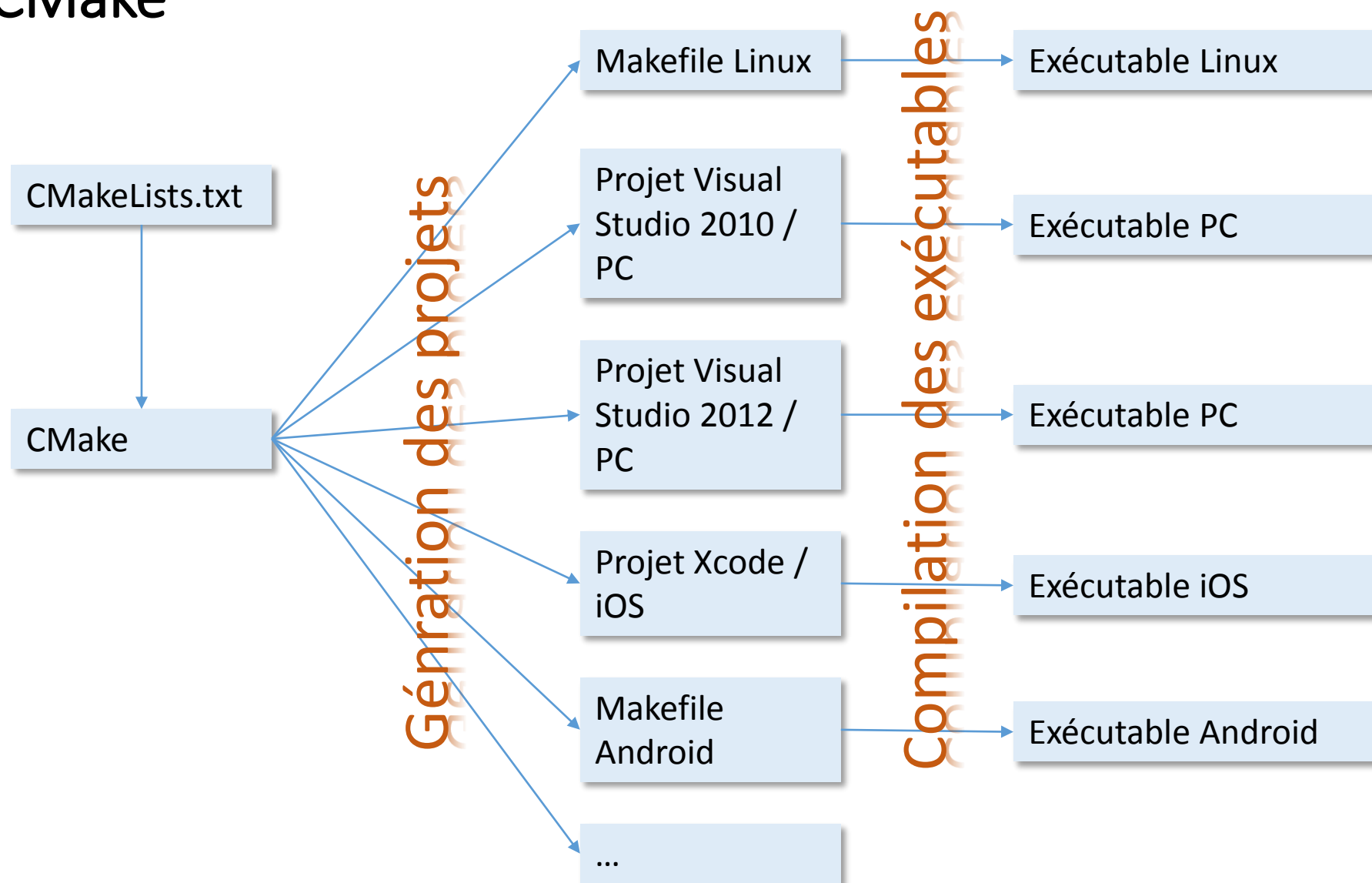
Documentation

Profiling

QA

Intégration continue

## CMake



## Compilation

Build configuration

- Plateforme

- Configuration

- Complexité

## CMake

Autobuild

- Intérêt

- Mise en place

## Tests unitaires

## Documentation

## Profiling

## QA

## Intégration continue

# Cmake

Exemple concret :

main.cpp

```
#include <stdio.h>

int main()
{
    printf("hello world.\n");
    return 0;
}
```

CMakeLists.txt

```
cmake_minimum_required(VERSION 2.8.9)

project(MyProject)

add_executable(hello main.cpp)
```



## Compilation

Build configuration

Plateforme

Configuration

Complexité

CMake

Autobuild

Intérêt

Mise en place

Tests unitaires

Documentation

Profiling

QA

Intégration continue

## Cmake

### Génération d'un makefile

```
$ cmake .  
-- The C compiler identification is GNU  
-- The CXX compiler identification is GNU  
-- Check for working C compiler: /usr/bin/gcc  
-- Check for working C compiler: /usr/bin/gcc -- works  
-- Detecting C compiler info  
-- Detecting C compiler info - done  
-- Check for working CXX compiler: /usr/bin/c++  
-- Check for working CXX compiler: /usr/bin/c++ -- works  
-- Detecting CXX compiler info  
-- Detecting CXX compiler info - done  
-- Configuring done  
-- Generating done  
-- Build files have been written to: ...
```

## Compilation

Build configuration

Plateforme

Configuration

Complexité

CMake

Autobuild

Intérêt

Mise en place

Tests unitaires

Documentation

Profiling

QA

Intégration continue

## Cmake

Compilation & generation de l'exécutable :

```
$ make
Scanning dependencies of target hello
[100%] Building CXX object
CMakeFiles/hello.dir/main.cpp.o
Linking CXX executable hello
[100%] Built target hello
```

Lancement de l'exécutable :

```
$ make
Scanning dependencies of target hello
[100%] Building CXX object
CMakeFiles/hello.dir/main.cpp.o
Linking CXX executable hello
[100%] Built target hello
```

## Compilation

Build configuration

Plateforme

Configuration

Complexité

CMake

Autobuild

Intérêt

Mise en place

Tests unitaires

Documentation

Profiling

QA

Intégration continue

## CMake

### Attention!

- Cmake est un outil permettant de simplifier la génération des configurations de **compilation**.
- Pour fonctionner sur chaque plateforme, le code doit être adapté en conséquence !

Pour aller plus loin : <https://cmake.org/>

# OUTILS

## Compilation

Build configuration

- Plateforme

- Configuration

- Complexité

CMake

Autobuild

- Intérêt

- Mise en place

Tests unitaires

Documentation

Profiling

QA

Intégration continue

## Autobuild

Ou comment « mettre en place un serveur permettant de compiler de manière automatique un ensemble de projets pour plusieurs plateformes et plusieurs configurations différentes »

# OUTILS

## Compilation

Build configuration

Plateforme

Configuration

Complexité

CMake

Autobuild

Intérêt

Mise en place

Tests unitaires

Documentation

Profiling

QA

Intégration continue

## Intérêt

Permettre de vérifier qu'une modification apportée à un programme et qui compile correctement dans une configuration, compile aussi dans toutes les autres configurations

# OUTILS

## Compilation

Build configuration

Plateforme

Configuration

Complexité

CMake

Autobuild

Intérêt

Mise en place

Tests unitaires

Documentation

Profiling

QA

Intégration continue

## Scripts :

Cmake + lancement de la compilation avec des scripts :

```
cmake .
```

```
rem Visual Studio
```

```
devenv InternalLibraries_2010_PC.sln /Clean
```

```
devenv InternalLibraries_2010_PC.sln /Build DebugEditor
```

```
devenv InternalLibraries_2010_PC.sln /Build ReleaseEditor
```

```
rem Android
```

```
...
```

## Compilation

Build configuration

Plateforme

Configuration

Complexité

CMake

Autobuild

Intérêt

Mise en place

Tests unitaires

Documentation

Profiling

QA

Intégration continue

## Limitation :

Pour certaines plateformes, nécessité d'avoir des architectures différentes :

- Mac + Xcode pour versions iOS & MacOS
- PC Windows + Visual Studio pour versions PC / Android / consoles
- PC Linux + GCC pour versions Linux / Android

→ Mise en place d'un protocole de compilation client / serveur :

- clients de compilation (pc windows, pc linux, mac)
- serveur de compilation qui joue le role de chef d'orchestre.

# OUTILS

## Compilation

Build configuration

Plateforme

Configuration

Complexité

CMake

Autobuild

Intérêt

Mise en place

Tests unitaires

Documentation

Profiling

QA

Intégration continue

## Il existe un certain nombre d'outils spécialisés :

- Ant : <http://ant.apache.org/>
- Maven : <https://maven.apache.org/>
- Cruise Control : <http://cruisecontrol.sourceforge.net/>
- Jenkins : <https://jenkins-ci.org/>
- Et beaucoup d'autres :  
[https://en.wikipedia.org/wiki/Comparison\\_of\\_continuous\\_integration\\_software](https://en.wikipedia.org/wiki/Comparison_of_continuous_integration_software)



Compilation

Tests unitaires

Intérêt

Exemple de test

Fonctionnement

Frameworks

Documentation

Profiling

QA

Intégration continue

## Test unitaire

« Vérifier que l'exécution d'une fonctionnalité appliquée à un certain nombre de variables d'entrée produit toujours le même résultat »

Un test unitaire permet de vérifier le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un programme.

Compilation

Tests unitaires

Intérêt

Exemple de test

Fonctionnement

Frameworks

Documentation

Profiling

QA

Intégration continue

## Intérêt

- Permettre de valider le bon fonctionnement de certaines parties considérées comme critiques d'un programme
- Permettre de s'assurer de leur bon fonctionnement quels que soient les conditions d'utilisation (paramètres, contraintes mémoires, contraintes de performance, ...)
- Permettre de s'assurer qu'elles sont en conformité avec une norme donnée

## Exemple :

```
void test_factorielle(void)
{
    int f = Factorielle(0);
    if (f != 1) printf("Factorielle(0) failed !\n");

    f = Factorielle(1);
    if (f != 1) printf("Factorielle(1) failed !\n");

    f = Factorielle(2);
    if (f != 2) printf("Factorielle(2) failed !\n");

    f = Factorielle(3);
    if (f != 6) printf("Factorielle(3) failed !\n");

    f = Factorielle(4);
    if (f != 24) printf("Factorielle(4) failed !\n");
}
```

Compilation

Tests unitaires

Intérêt

Exemple de test

Fonctionnement

Frameworks

Documentation

Profiling

QA

Intégration continue

## Fonctionnement

Un framework de tests permet de définir des tests unitaires, qu'on appelle des « tests cases », et qu'on peut regrouper dans des collections, appelées « test suites ».

On peut ensuite exécuter les tests et vérifier que les résultats qu'ils produisent sont conformes aux résultats précédemment enregistrés.

# Frameworks

**xUnit** : réaliser des tests unitaires dans un langage donné (dont l'initiale remplace « x » le plus souvent).

Framework :

Langage :

*Atoum*

[PHP](#)

*PHPUnit*

[PHP](#)

*CUnit*

[C](#)

*CppUnit*

[C++](#)

*Google Test*

[C++](#)

[JUnit](#)

[Java](#)

*NUnit*

[.NET](#)

...

Compilation  
Tests unitaires

Documentation

Automatisation  
Outils  
Exemples

Profiling  
QA  
Intégration continue

## Automatisation

Il est possible d'insérer, à l'intérieur d'un source code, des commentaires qui sont destinés à être analysés par un compilateur afin d'être transformés en une documentation mise en forme.

Il est ainsi plus simple pour le développeur de mettre à jour la documentation, car il effectue les changements qu'il apporte au niveau des commentaires du code affectent directement le contenu de la documentation.

Compilation  
Tests unitaires

Documentation

Automatisation  
Outils  
Exemples

Profiling

QA

Intégration continue

## Outils

### Javadoc

<http://www.oracle.com/technetwork/java/javase/documentation/index-jsp-135444.html>

→ Génération de documentation pour Java, développé par Sun

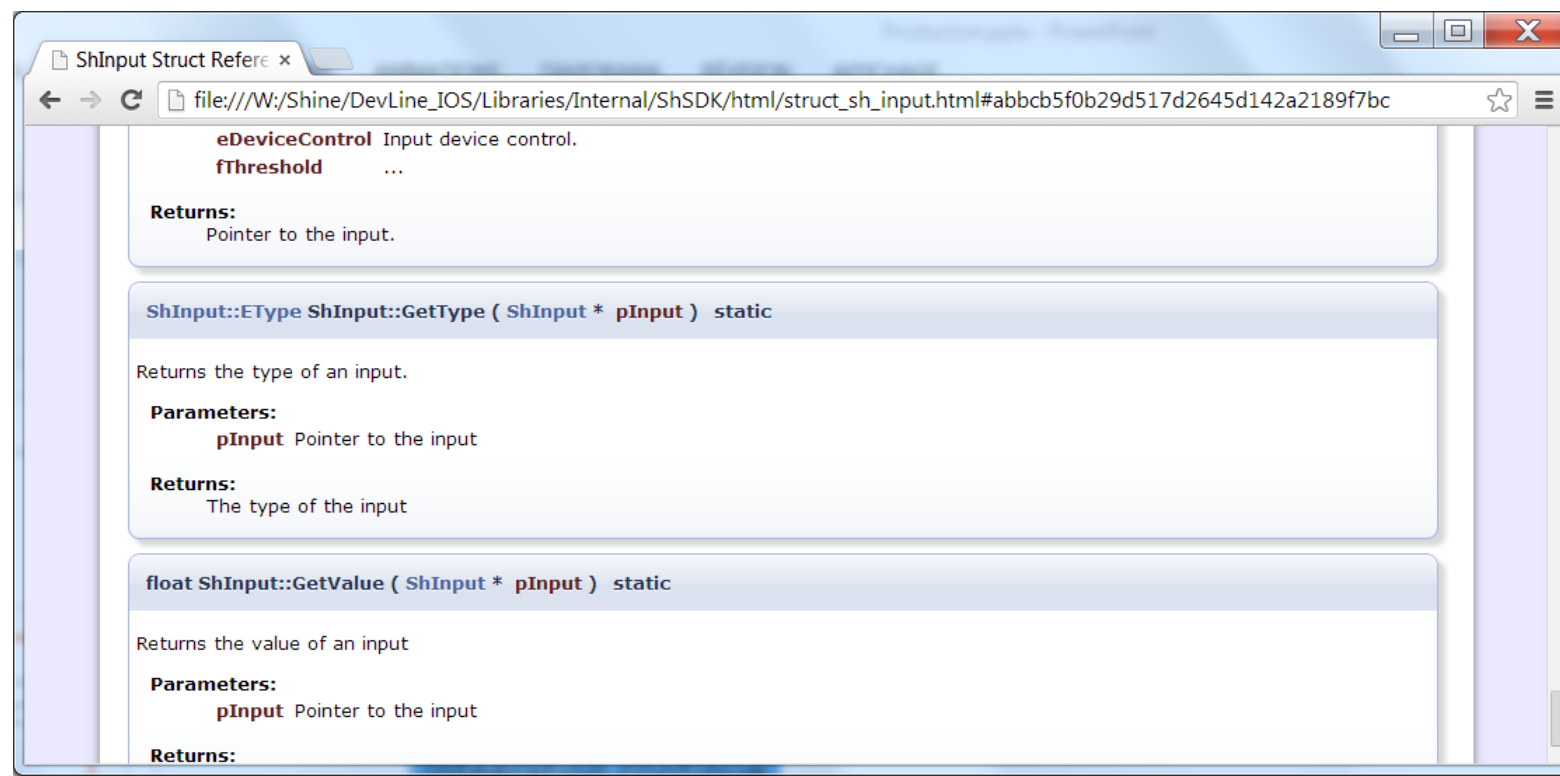
### Doxygen

[www.doxygen.org](http://www.doxygen.org)

→ Génération de documentation pour C/C++

# Doxygen

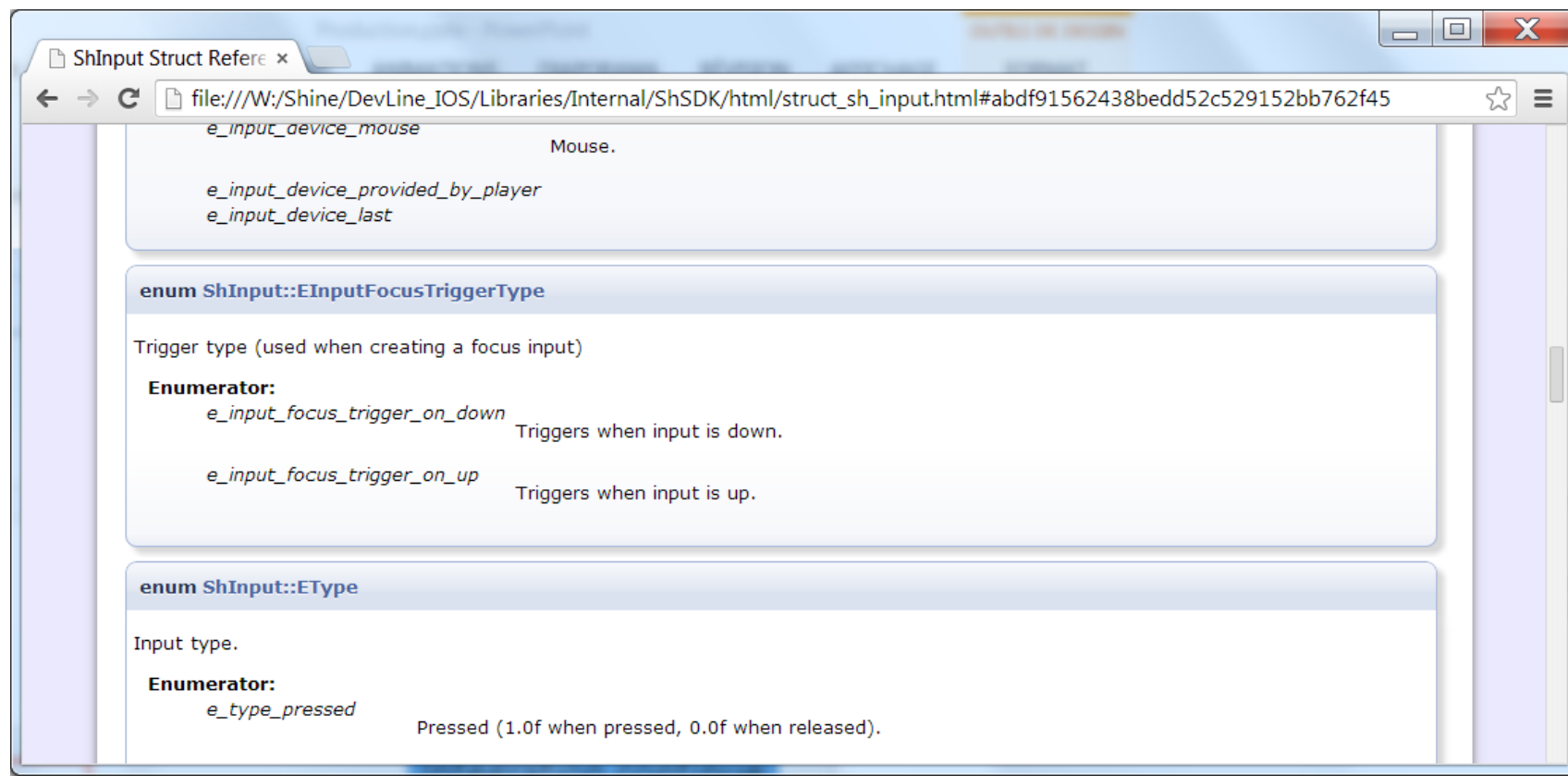
```
//-----
/// Returns the type of an input.
/// @param pInput Pointer to the input
/// @return The type of the input
//-----
/*static*/ ShInput::EType ShInput::GetType(ShInput * pInput)
{
    return( (ShInput::EType) ( (CShInput*)pInput)->GetType() );
}
```





# Doxygen

```
//
///< Trigger type (used when creating a focus input)
enum EInputFocusTriggerType
{
    e_input_focus_trigger_on_down = 0,    ///< Triggers when input is down.
    e_input_focus_trigger_on_up          ///< Triggers when input is up.
};
```



Compilation

Tests unitaires

Documentation

Profiling

Intérêt

Fonctionnement

Outils

QA

Intégration continue

## Profiling

« Analyser l'exécution d'un logiciel afin de connaître son comportement à l'exécution »

- Liste des fonctions appelées
- Temps passé dans chaque fonction
- Utilisation processeur
- Utilisation mémoire
- ...

# OUTILS

Compilation  
Tests unitaires  
Documentation

## Profiling

Intérêt  
Fonctionnement  
Outils

QA  
Intégration continue

## Intérêt

### CPU

Connaitre les allocations qui consomment le plus de mémoire  
Connaitre les méthodes qui consomment le plus de temps machine  
...

### GPU

Connaître les commandes qui consomment le plus de temps  
Etre capable de tracer pas à pas les commandes exécutées sur le GPU  
Etre capable de visualiser des parties de la mémoire vidéo  
...

# Fonctionnement

Méthode par instrumentation :

Rajout d'instructions au début et à la fin de chaque fonction afin de mesurer le temps écoulé à l'intérieur de la fonction.

Méthode par sampling :

Une interruption est générée au niveau du processeur à intervalles réguliers, et permet de récupérer la call stack à partir du pointeur d'instructions.

# OUTILS

Compilation  
Tests unitaires  
Documentation

## Profiling

Intérêt  
Fonctionnement  
Outils

## QA

Intégration continue

# Outils

## CPU

Intel® VTune™ Amplifier XE 2013

<http://software.intel.com/en-us/intel-vtune-amplifier-xe>



## GPU

NVIDIA Nsight

<https://developer.nvidia.com/nsight-eclipse-edition>

<https://developer.nvidia.com/nvidia-nsight-visual-studio-edition>



AMD CodeAnalyst

<http://developer.amd.com/tools-and-sdks/archive/amd-codeanalyst-performance-analyzer/>



# OUTILS

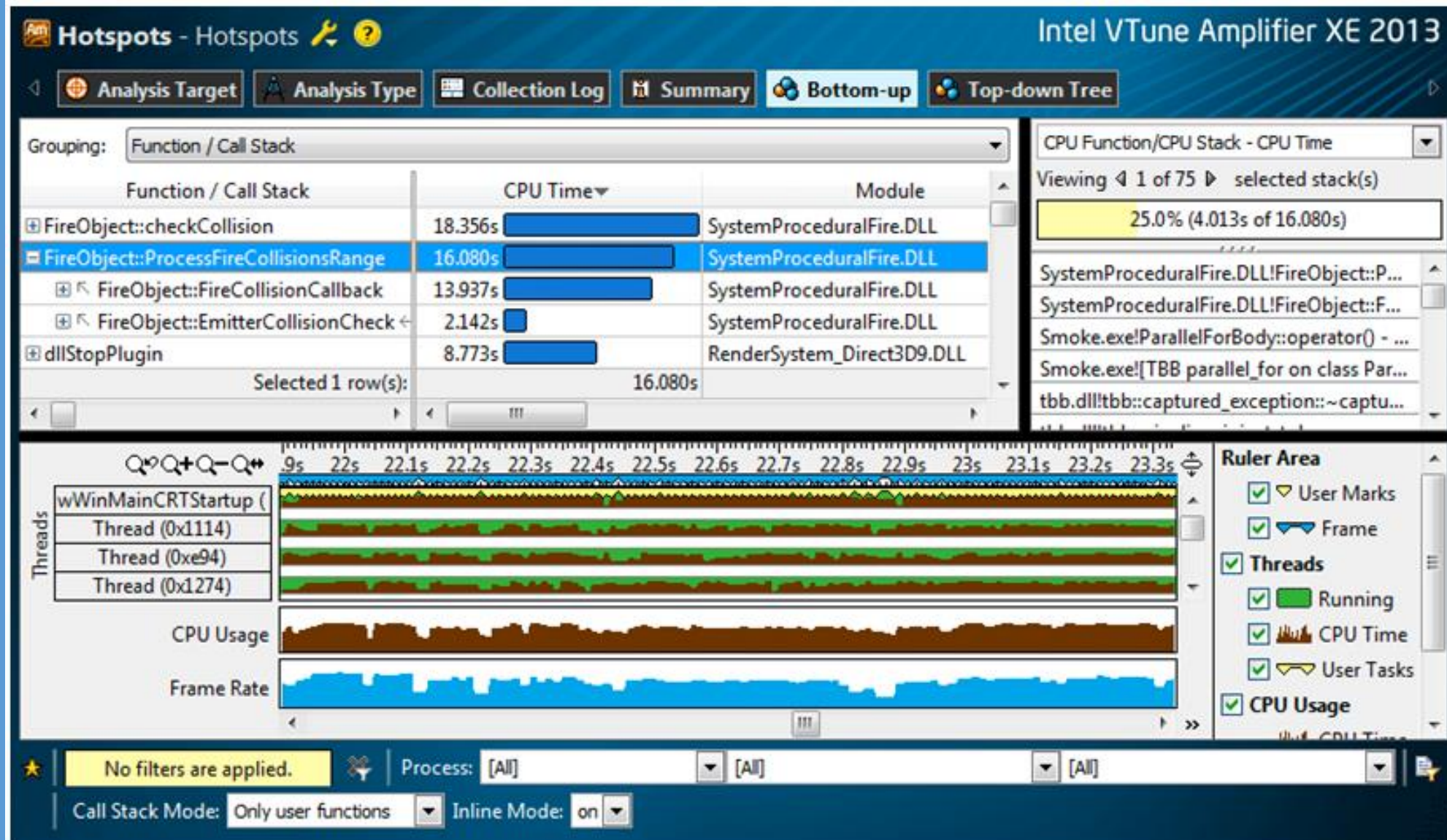
Compilation  
Tests unitaires  
Documentation

## Profiling

Intérêt  
Fonctionnement  
Outils

QA

Intégration continue





# OUTILS

Compilation  
Tests unitaires  
Documentation

## Profiling

Intérêt  
Fonctionnement  
Outils

## QA

Intégration continue

supersonicsled (Debugging) - Microsoft Visual Studio (Administrator)

File Edit View Project Build Debug Team Nsight Data Tools Test Analyze Window Help

Debug Win32 RuntimeApiTrace\_t

Process: [4116] SupersonicSled.exe Thread: [0] Graphics Default Context Stack Frame: Shader: SledDarkMetal - Copy.psh - [0] m

SledDarkMetal - Copy.psh Graphics Inspector - SupersonicSled.exe

```
MRT_OUT main ( g2fConnector g2f )
{
    float3 N = normalize ( g2f . worldNormal );
    float3 T = normalize ( g2f . worldTangent );
    float3 B = normalize ( g2f . worldBinormal );
    float dispTiling = 4 ;
    float displacementHeightCodeScale = 0.0015 ;
    if ( dot ( N , cross ( T , B ) ) < 0 )
    {
        displacementHeightCodeScale *= - 1.0 ;
    }
    float mindesiredlength = 2048.0 ;
    float2 uv_offset_over = ddx ( g2f . tex );
    float l_over = length ( uv_offset_over );
    float desiredlengthover = max ( 1.0 / mindesiredlength , l_over );
    uv_offset_over *= desiredlengthover / l_over ;
    float2 uv_offset_up = ddy ( g2f . tex );
    float l_up = length ( uv_offset_up );
    float desiredlengthup = max ( 1.0 / mindesiredlength , l_up );
    uv_offset_up *= desiredlengthup / l_up ;
    float4 normalTap = s_dispTilesTex . Sample ( RepeatAniso8 , g2f . tex . xy * dispTiling );
    float4 normalTapOver = s_dispTilesTex . Sample ( RepeatAniso8 , g2f . tex . xy * dispTiling + uv_offset_over * di
    float4 normalTapUp = s_dispTilesTex . Sample ( RepeatAniso8 , g2f . tex . xy * dispTiling + uv_offset_up * dispTi
    float displacementCenter = displacementHeight * dot ( normalTap , dispWeights ) * displacementHeightCodeScale ;
    float displacementOver = displacementHeight * dot ( normalTapOver , dispWeights ) * displacementHeightCodeScale ;
    float displacementUp = displacementHeight * dot ( normalTapUp , dispWeights ) * displacementHeightCodeScale ;
    float3 tanNormalBump = normalize ( - cross ( float3 ( uv_offset_over . x , uv_offset_over . y , ( displacementOve
```

Graphics Focus Picker

Summary  
Show the summary of the current status

Vertices (3D View)  
Show the paused vertices in a 3D view

Vertices (Table View)  
Show the paused vertices in a table view

Geometry (Table View)  
Show the paused geometry shader threads

Pixels  
Show the paused pixels on the current render target

Compute  
Show the paused compute threads

Pixels  
RT0 - 1280x720 - (R16G16B16A16\_FLOAT)  
1 Paused Pixels  
1280x720  
576,302  
Set Pixel Rectangle

Locals

Name	Value	Type
worldNormal	{x = -0.0099726664, y = -0.03755236, z = 0.99924493}	float3
worldCoord	{x = -0.51191813, y = 3.4421973, z = -0.18642822, w = 1}	float4
worldBinormal	{x = 0.13819657, y = 0.98963982, z = 0.038570616}	float3
worldTangent	{x = 0.99034107, y = -0.13847685, z = 0.0046797399}	float3
tex	{x = 0.076345407, y = 0.86925811}	float2
projVelocity	{x = 0.00010310233, y = 6.8565554e-05}	float2
N	{x = -0.0099726655, y = -0.037552357, z = 0.99924481}	float3
x	-0.0099726655	float
y	-0.037552357	float
z	0.99924481	float

Shaders

Name	Shader Type	Symbolics Status	Language	Tec
69 main	Vertex	Ready.	HLSL	No
70 main	Pixel	Ready.	HLSL	No
71 main	Vertex	Ready.	HLSL	No
72 main	Pixel	Ready.	HLSL	No
73 main	Vertex	Shader has not yet been used by the application.	HLSL	No
74 main	Pixel	Shader has not yet been used by the application.	HLSL	No
75 main	Vertex	Shader has not yet been used by the application.	HLSL	No
76 main	Pixel	Shader has not yet been used by the application.	HLSL	No
77 main	Vertex	Shader has not yet been used by the application.	HLSL	No

Autos Locals Watch 1 Find Results 1

Ready

Shaders Call Stack Breakpoints Command Window Immediate Window Output

# OUTILS

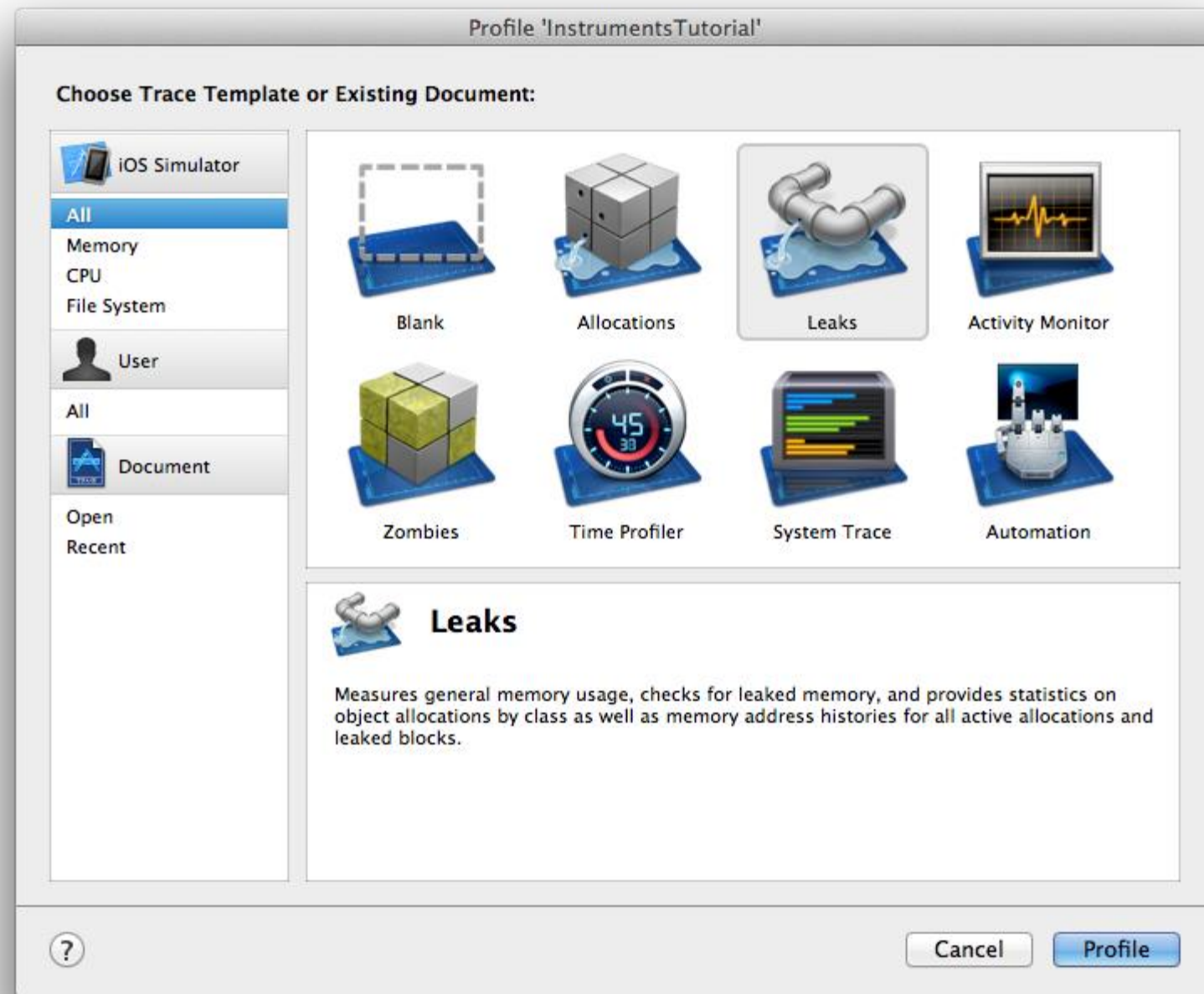
Compilation  
Tests unitaires  
Documentation

## Profiling

Intérêt  
Fonctionnement  
Outils

QA

Intégration continue





# OUTILS

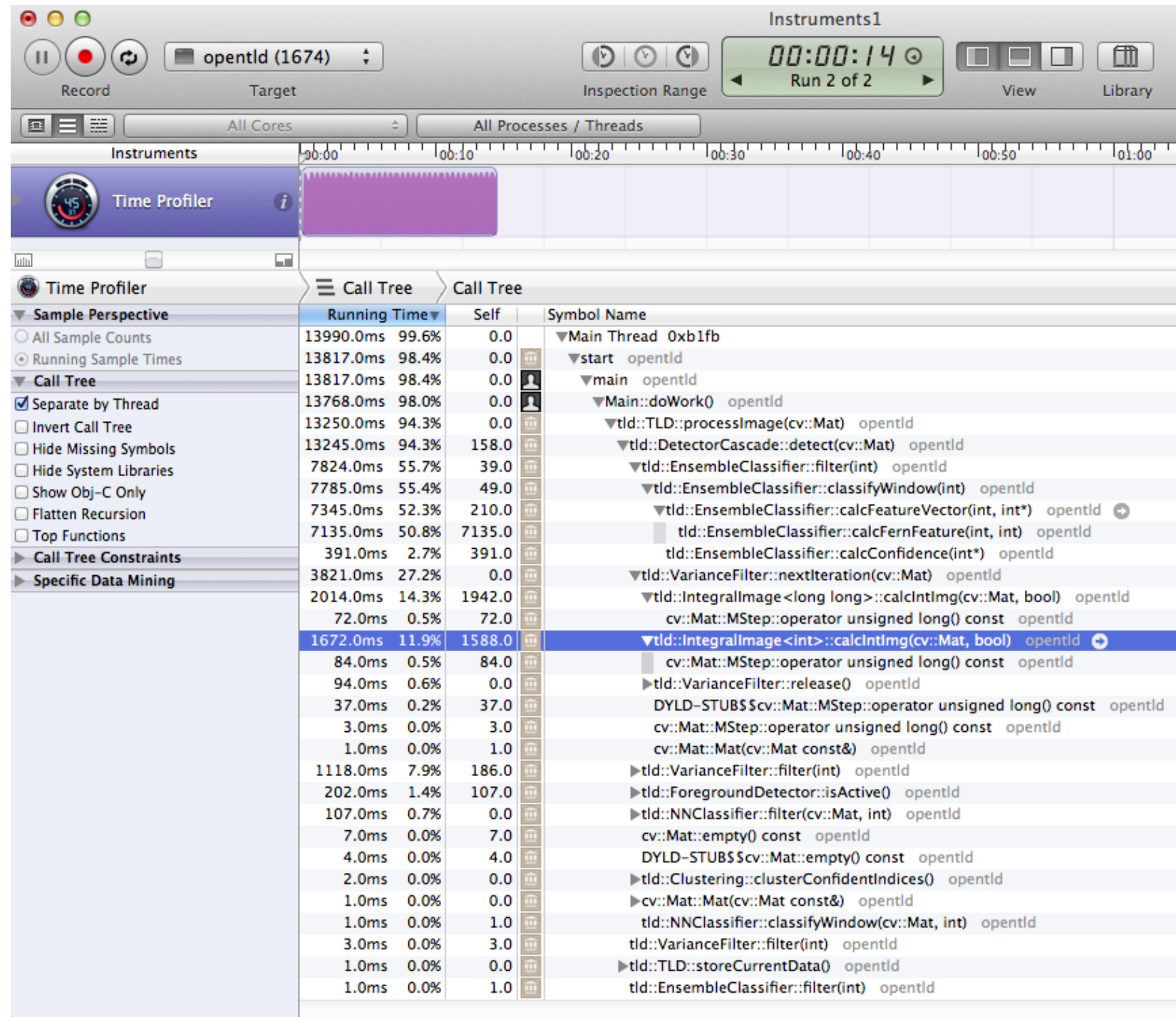
Compilation  
Tests unitaires  
Documentation

## Profiling

Intérêt  
Fonctionnement  
Outils

## QA

Intégration continue



Compilation  
Tests unitaires  
Documentation  
Profiling

Définition  
Fonctionnement  
Outils

Intégration continue

## QA – Quality Assurance

Mise en place de tests permettant de valider les différentes fonctionnalités d'un programme.

## Fonctionnement

Une équipe de testeur utilise le programme et relève les défauts d'exécution, qu'ils soient fonctionnels ou de toute autre nature.

Ils font part de ces défauts à l'équipe de développement afin que celle-ci se charge de les corriger.

Sur un projet logiciel, le nombre de bugs moyen en phase de QA peut se mesurer en milliers, voire en dizaine de milliers.

→ Nécessité d'avoir une interface de communication entre ces équipes.

Compilation  
Tests unitaires  
Documentation  
Profiling

## QA

Définition  
Fonctionnement

### Outils

FlySpray  
Mantis  
RedMine

Intégration continue

## Outils

Permettent de gérer une base de données de bugs.

Les testeurs peuvent ajouter des bugs dans la base de données, et y affecter certains critères :

- un facteur de gravité (Critique, Majeur, Mineur, ...)
- une priorité (Haute, Moyenne, Basse)
- un status (non confirmé, nouveau, assigné, résolu, vérifié, résolu...)

Les développeurs peuvent ensuite prendre la main sur un bug (non confirmé ou nouveau), changer le status (se l'assigner), le corriger, et le marquer comme résolu.

Les testeurs vérifient ensuite que les bugs résolus le sont bien, et peuvent alors changer le status (fermé)

# OUTILS

Compilation  
Tests unitaires  
Documentation  
Profiling

## QA

Définition  
Fonctionnement  
Outils  
FlySpray  
Mantis  
RedMine

Intégration continue

FLYSPRAY

Afficher la tâche #

Nom d'utilisateur

Mot de passe

Se souvenir de moi

Se connecter

S'inscrire

Mot de passe perdu ?

Tous les projets

Basculer

Aperçu

Liste des tâches

Rechercher dans ce projet

Avancé

ID	Projet	Résumé	Date d'ouverture	Etat	Progression	Catégorie		Type de tâche	Sévérité	Date d'échéance	Dernière édition	Assignée à	Basée sur	
386	DZ_testerelli	DZ2 - contact client	20.10.08	Open		Backend / Core	0	Specs	Critique		23.10.08	super	1.0	3
371	abc123	asdasdasd	15.10.08	geschlossen		labo1	0	Orga	Critique	15.10.08		john, +1	1.0	0
336	SIGL 2009	ghjfhgj	06.10.08	Open		Magazzino	0	Programmierung	Critique			super	1.0	0
314	Proyecto1	Texto resumido de la tarea	01.10.08	in bearbeitung		Backend / Core	0		Critique		14.10.08		1.0	2
309	Test 082008	Change workname into call-ID	29.09.08	in bearbeitung		RFC	0		Critique			alex	1.0	0
308	Line Parking	Mise à jour du machin	29.09.08	geschlossen		Backend / Core	0		Critique		05.07.09	super	1.0	3
297	Proyecto1	This is a task	25.09.08	in bearbeitung		Backend / Core	0	Orga	Critique		14.10.08		1.0	1
291	Projeto Teste	tarefa teste	24.09.08	in bearbeitung		Backend / Core	0	Orga	Critique	30.09.08		super	1.0	0
257	Oma_skodaus	Buff ovfl. blablabla	02.09.08	Open		Backend / Core	0	QS	Critique	30.08.08	03.09.08	super	1.0	1
252	dupowaty	2336456413	01.09.08	in bearbeitung		Systemy	0	Orga	Critique	23.09.08	16.10.08	super	1.1 beta 1	2
224	Oma_skodaus	Nouvelle tâche	26.08.08	Open		Backend / Core	0	Orga	Critique		18.09.08	super	1.0	3
91	picaoclc	F& geht nicht	25.07.08	offen		OUS	0	makros	Critique			aaaaa	1.0	0
69	Kipróbáalom projekt	NEMTOM	20.07.08			SQL Server	0		Critique	19.07.08	17.10.08	super		4
44	Test-Account	Gebäude-Repp funzt nicht	15.07.08	Open		Backend / Core	0	Orga	Critique		23.09.08	super	1.0	2
16	dupowaty	Narrator needs a complete rewrite!	09.07.08	Open		test	0	Orga	Critique	08.07.08		super	1.0	0
365	,0,2	Cheese missing	14.10.08	in bearbeitung		Menu	0	Programmierung	Haute	21.10.08		super	1.0	0
356	aaaaTEST	test	06.07.09	Open		Aplicación 2 módulo 1	0	Orga	Haute	28.10.08	06.07.09	super	1.0	2
350	Line Parking	Test-Task	09.10.08	Open		Backend / Core	0	Testing	Haute			super	1.0	0
333	My little project	Crash when you enter the word crash...	04.10.08	in bearbeitung		Module 2	0	Programmierung	Haute		07.10.08		1.0	1
321	Oma_skodaus	probleme printer	02.10.08			Printer	0		Haute				1.0	0

Tâches 1 - 20 sur 200

Page 1 sur 10 -- 1 - 2 - 3 - 4 - 5 - Suivant > Dernier >>

# OUTILS

Compilation  
Tests unitaires  
Documentation  
Profiling

## QA

Définition  
Fonctionnement  
Outils  
FlySpray  
Mantis  
RedMine

Intégration continue

Mozilla Firefox

http://192.168.1.139/

Getting Started Latest Headlines

JumpBox Admin | Register | Support | Applications Login | Project Page | Support



Logged in as: admin (administrator) - administrator 11-20-2007 21:11 MST Project: Test Switch

Main | My View | View Issues | Report Issue | Change Log | Summary | Docs | Manage | Edit News | My Account | Logout Jump

Reporter:	Monitored By:	Assigned To:	Category:	Severity:	Resolution:	Profile:
any	any	any	any	any	any	any
Status:	Hide Status:	Product Build:			Priority:	
any	closed (And Above)	any			any	
Show:	View Status:	Show Sticky Issues:	Changed(hrs):	Use Date Filters:	Relationships:	
50	any	Yes	6	No	any	
Sort by:	Last Update Descending					

Search: Apply Filter [ Advanced Filters ] Reset Filter Save Current Filter

Viewing Issues (1 - 1 / 1) [ Print Reports ] [ CSV Export ]

	P	ID	#	Category	Severity	Status	Updated	Summary
<input type="checkbox"/>			0000001		feature	new	11-20-07	JumpBox is Incredible

☐ Select All Move OK

new feedback acknowledged confirmed assigned resolved

Done

Brothersoft

S3Fox



OUTILS

Compilation  
Tests unitaires  
Documentation  
Profiling

QA

Définition  
Fonctionnement  
Outils  
FlySpray  
Mantis  
RedMine

Intégration continue

Redmine

Recherche:

Aperçu Download Activité Roadmap Demandes Annonces Wiki Forums Dépôt

Demandes

Filtres

☒ Statut

ouvert

Ajouter le filtre:

✓ Appliquer  Effacer

✓ #	Tracker	Statut	Priorité	Sujet	Mis à jour	Catégorie
<input type="checkbox"/> 3480	Defect	New	Normal	Bot filter plugin crashes when missing useragent	2009-06-11 08:50	Plugins
<input type="checkbox"/> 3479	Defect	New	High	SEGFALT in Mysql	2009-06-11 07:15	Tickets
<input type="checkbox"/> 3477	Patch	New	Normal	Fix access handler to remove the need for a separate svn-private/git-private	2009-06-10 21:54	
<input type="checkbox"/> 3476	Defect	New	Normal	Right-floating TOC funky in the roadmap	2009-06-10 19:52	UI
<input type="checkbox"/> 3473	Feature	New	Normal	Can Redmine support notification when a issue will be overdue?	2009-06-10 07:17	Emails
<input type="checkbox"/> 3472	Defect	New	Normal	Role given to a non-admin user who creates a project	2009-06-10 06:37	
<input type="checkbox"/> 3471	Defect	Resolved	Normal	Project managers should be able to assign "sub-project of"	2009-06-10 05:49	Projects
<input type="checkbox"/> 3470	Feature	New	Normal	Projects should inherit documents and files from sub-projects	2009-06-10 23:08	Projects
<input type="checkbox"/> 3469	Defect	New	Normal	senAS	2009-06-10 04:48	Administration
<input type="checkbox"/> 3468	Defect	Resolved	Normal	Subversion : View Differences 500 error	2009-06-09 21:21	SCM
<input type="checkbox"/> 3467	Feature	New	Normal	Due date sort order should sort issues with no due date to the end of the list	2009-06-09 08:12	Tickets
<input type="checkbox"/> 3466	Patch	New	Normal	ja label for text_status_changed_by_changeset	2009-06-09 05:06	
<input type="checkbox"/> 3465	Feature	New	Urgent	Default project	2009-06-09 04:46	
<input type="checkbox"/> 3464	Feature	New	Normal	columns "user_id" & "created_on" in tables like projects, documents & custom_fields...	2009-06-08 21:29	Administration
<input type="checkbox"/> 3463	Feature	New	Low	Export (all) Wiki-Pages to PDF/DOC	2009-06-09 09:02	Wiki
<input type="checkbox"/> 3462	Defect	New	High	CVS path encoding problems	2009-06-08 16:24	SCM
<input type="checkbox"/> 3461	Patch	New	Normal	Manage permission on issue assigment	2009-06-10 20:41	Permissions
<input type="checkbox"/> 3457	Defect	New	Normal	Defaul value on log text fields	2009-06-08 11:01	Custom fields
<input type="checkbox"/> 3454	Defect	New	Normal	Mercurial Repository Browsing Disappearing	2009-06-06 13:20	SCM
<input type="checkbox"/> 3453	Feature	New	Normal	Issue creation via email by anonymous	2009-06-07 05:51	Emails
<input type="checkbox"/> 3452	Feature	New	Normal	Per project email notifications settings panel	2009-06-07 16:38	Projects
<input type="checkbox"/> 3451	Defect	Resolved	Normal	Issue Creation Via Email not Working	2009-06-10 12:34	Emails
<input type="checkbox"/> 3450	Feature	New	Normal	Project WorkFlow	2009-06-07 00:31	
<input type="checkbox"/> 3449	Defect	New	High	Redmine Takes Too Long On Large Mercurial Repository	2009-06-10 12:35	SCM
<input type="checkbox"/> 3448	Feature	New	Normal	Add issue watcher that isn't a maintainer	2009-06-10 11:31	

Demandes

Voir toutes les demandes  
Résumé  
Historique

Rapports personnalisés

Documentation issues  
Open defects  
Open features  
Plugin issues  
Translation patches  
User Interface Issues

Donate

Compilation  
Tests unitaires  
Documentation  
Profiling  
QA

## Intégration continue

Coût d'une intervention  
Serveur de build  
Exemple de mise en place

## Intégration continue

Permettre, lors de chaque modification, de :

- compiler le programme sur toutes les plateformes / configurations
  - lancer un certain nombre de tests unitaires
  - Informer sur les éventuelles regressions
  - éventuellement déployer les versions 'stables' (pas d'erreurs sur les tests)
- être réactif par rapport aux nouvelles regressions, afin de converger vers une version stable plus rapidement.



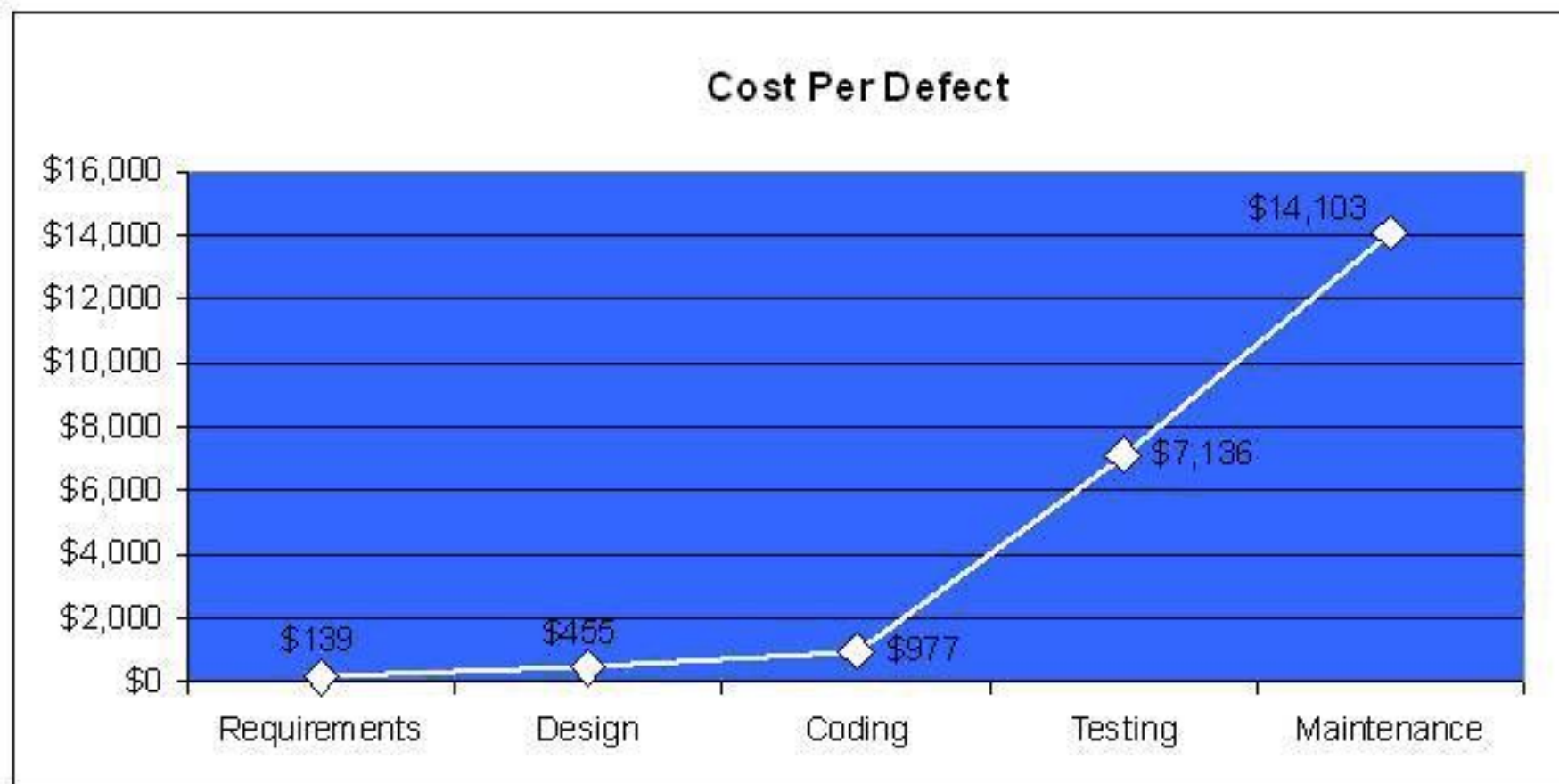
# OUTILS

Compilation  
Tests unitaires  
Documentation  
Profiling  
QA

## Intégration continue

Coût d'une intervention  
Serveur de build  
Exemple de mise en place

## Coût d'une intervention



# OUTILS

Compilation  
Tests unitaires  
Documentation  
Profiling  
QA

## Intégration continue

Coût d'une intervention  
Serveur de build  
Exemple de mise en place

## Serveur de build

### Principales tâches :

- Synchronisation des codes sources avec le serveur de sources
- Compilation des données
- Compilation des codes sources
- Lancement des tests unitaires
- Déploiement des versions
- Génération des documentations

Il permet de rapidement détecter un dysfonctionnement et ainsi de réduire les coûts liés à sa correction.

# OUTILS

Compilation  
Tests unitaires  
Documentation  
Profiling  
QA

## Intégration continue

Coût d'une intervention  
Serveur de build  
Exemple de mise en place

# L'intégration continue en images

