|  |
| --- |
| Rathaxes |
| Plan d’assurance qualité |
|  |

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Document informations | | | |
| Auteur(s) | Relecteur | Approbateurs | Date |
| David Verrière  Christophe Fajardo  Sebastien Cassier  Marc Thompson | Mickael Dumont | David Verrière | 06/06/2008 |

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Revisions | | | |
| Revision | Description | Approbateur(s) | Date |
| 1.0 | First Draft | Sebastien Cassier | 19/04/2008 |
| 2.0 | PDCA ajouté | Marc Thompson | 24/05/2008 |
| 3.0 | Ajouts de schéma et descriptions | Christophe Fajardo | 07/06/2008 |
| 4.0 | Finalisation | David Verrière | 01/07/2008 |

Sommaire

[1) Intro 3](#_Toc202545431)

[Description du projet 3](#_Toc202545432)

[2) Référentiel 4](#_Toc202545433)

[a) Documents applicables 4](#_Toc202545434)

[b) Autres documents 4](#_Toc202545435)

[c) Outils, méthodes, normes et standards 4](#_Toc202545436)

[Outils 4](#_Toc202545437)

[Normes et standards 4](#_Toc202545438)

[d) Glossaire des termes 7](#_Toc202545439)

[e) Abréviations et sigles 7](#_Toc202545440)

[3) Organisation du projet 8](#_Toc202545441)

[1) Présentation des équipes 8](#_Toc202545442)

[Equipe hardware 9](#_Toc202545443)

[Equipe Langage 9](#_Toc202545444)

[Equipe Transverse 9](#_Toc202545445)

[Equipe Infra-Com 9](#_Toc202545446)

[2) Planning Prévisionnel 10](#_Toc202545447)

[3) Diagramme de Gant 11](#_Toc202545448)

[4) Instances internes 12](#_Toc202545449)

[5) Instances externes 12](#_Toc202545450)

[1) Partenaires 12](#_Toc202545451)

[2) Conférences 12](#_Toc202545452)

[3) Réunions 12](#_Toc202545453)

[4) Apport d’une expertise externe 12](#_Toc202545454)

[6) Prestation et produits livrable 13](#_Toc202545455)

[7) Démarche 14](#_Toc202545456)

[8) Gestion des événements 16](#_Toc202545457)

[9) Processus de support 17](#_Toc202545458)

[10) Maitrise des fournitures et Reproduction, production et livraison 17](#_Toc202545459)

[Fournitures 17](#_Toc202545460)

[Reproduction production et livraison 17](#_Toc202545461)

[11) Gestion du PAQ 17](#_Toc202545462)

[12) Bilan projet 18](#_Toc202545463)

[13) Garantie et réclamation client 18](#_Toc202545464)

# Intro

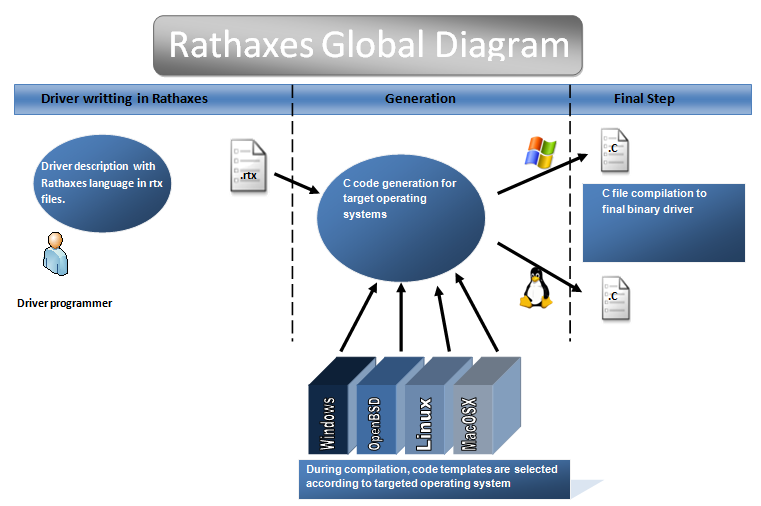
Le monde de l’informatique est soumis a des contraintes toujours plus importantes. Le souci de qualité est encore plus présent dans le domaine de recherche de Rathaxes. Les programmes générés par Rathaxes sont les plus sensibles dans un système d’exploitation, la défaillance d’un entraine la défaillance de tout le système.

Ce document contient les procédures mises en place afin d’assurer la qualité du produit fini Rathaxes. Il est réalisé dans le cadre du projet de fin d’étude Epitech.

## Description du projet

Le titre complet du projet est « Approche langage à la génération de pilotes de périphérique robuste multiplateformes ». Ce projet a pour but de proposer un langage qui permettra d’écrire des pilotes de périphérique robuste pouvant être généré simplement pour différents systèmes d’exploitation.

Rathaxes apporte une approche différente à la conception de pilotes de périphériques en rendant l’écriture du code générique.



# Référentiel

Cette section est à pour but de décrire les documents applicable au projet.

## Documents applicables

Documents produits par Rathaxes :

|  |  |  |
| --- | --- | --- |
| Titre | Date | Emplacement |
| Documentation Utilisateur | 06/06/2008 | http://hosting.labpfe.epitech.net/rathaxes/fr/Documentation.html |
| Documentation Développeur | 06/06/2008 | http://hosting.labpfe.epitech.net/rathaxes/fr/Documentation.html |
| Documentations Techniques | 06/06/2008 | http://hosting.labpfe.epitech.net/rathaxes/fr/Documentation.html |

## Autres documents

Documents externes au projet :

|  |  |  |
| --- | --- | --- |
| Titre | Date | Emplacement |
| Approche langage au développement de pilotes de périphériques robustes | 2001 | http://www.inria.fr/rrrt/tu-0992.html |
| Osronline Windows driver documentation | 2008 | www.**osr**online.com |

## Outils, méthodes, normes et standards

### Outils

Nous utilisons différents outils pour Rathaxes :

* SVN : pour le contrôle de code source
* Track : pour le bug tracking et le suivi de projet
* Mailing liste pour la communication interne
* Calendrier partagé : pour la gestion des rendez vous et réunions

### Normes et standards

Nous utilisons différentes normes et standards pour le projet :

* MILSTD 498 : il s’agit d’une méthode de qualité logicielle appliquée par les militaires.
* Cycle en V : un processus de développement qui permet d’itérer sur différentes phase d’un projet allant de la conception à la livraison.
* Rathaxes utilise bien évidemment des normes de codages internes.
* ISO9000 : 2000 pour la qualité de projet

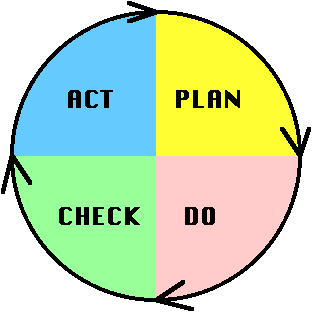
#### PDCA :

Le PDCA est une méthode qualité permettant d’améliorer la qualité d’un produit par l’enchainement de quatre étapes s’entrainant l’une l’autre. Souvent représenté sous la forme d’une roue, la Roue de Deming du nom du statisticien ayant inventé cette méthode, cette méthode vise la création d’un cercle vertueux entrainant une amélioration constante de la qualité.

La qualité est définit par la norme ISO 9000 : 2000 : qui définit les exigences pour les systèmes de management de la qualité, est maintenant fermement établie comme la norme appliquée mondialement pour donner l'assurance de la capacité de répondre à des exigences qualité et d'augmenter la satisfaction des clients dans les rapports clients-fournisseurs.

Les quatre étapes se suivent dans cet ordre ci :

* **P**lan - Planifier
* **D**o - Réaliser
* **C**heck - Vérifier
* **A**ct – Réagir



Plan :

Il s’agit de préparer le plan d’action. Cette étape peut se décomposer en trois autres étapes plus spécifiques :

* Identification du ou des problèmes
* Recherche des causes
* Recherche des solutions

Cette partie se solde par l’écriture d’un cahier des charges et la définition d’un planning.

Dans le cas de Rathaxes cette action est effectuée régulièrement durant les réunions « responsables d’équipes »

Do :

Il s’agit la de mettre en œuvre le plan d’action spécifié plus tôt.

La période de développement du projet ou les équipes partent chacune de leur cotés effectuer les taches qui leurs sont assignées.

Check :

Par le biais de procédures de tests, on contrôle le travail réalisé en vérifiant :

* La fonctionnalité : Si cela fonctionne comme prévu
* L’utilité : Si cela solutionne les problèmes soulevés lors de la préparation.

En l’occurrence pour Rathaxes, cette procédure de vérification est effectué par la relation entre les équipes langage et hardware :

* L’équipe langage valide la qualité des documents produits par l’équipe hardware.
* L’équipe hardware vérifie la fonctionnalité et l’utilisabilité du langage produit par l’équipe langage.

Act :

Durant cette étape, on réagit par rapport aux tests effectués et à leurs résultats.

* S’ils sont probants, on peut décider de capitalisé sur certaine solutions
* S’ils ne le sont pas, il s’agit de trouver une solution rapidement.

Quoi qu’il en soit, cette étape entraine une autre étape plan puisqu’en fonction des résultats il y aura de nouvelles modifications à planifier, puis exécuter et contrôler et ainsi de suite.

## Glossaire des termes

* Driver : programme qui permet le dialogue entre un périphérique et un le système d’exploitation
* Compilateur : programme qui traduit le code d’un langage en un autre
* Librairie de template : ensemble de brique de codes

## Abréviations et sigles

* DMA: Direct Memory
* IO: Input Output
* LSE : Laboratoire Système et Sécurité EPITA EPITECH
* OS : Operating System
* SVN : Subversion

# Organisation du projet

## Présentation des équipes

* Interaction avec toutes les équipes
* Interaction avec toutes les équipes
* Suivi qualité
* Support de chaque équipe
* Veille Techno
* Support logistique
* Communication
* Conférences
* Conception
* Réalisation
* Tests
* Documentation
* Recherche
* Généralisation
* Tests
* Veille Techno
* Documentation

***Cross Team***

***Infra-Com***

***Langage***

***Hardware***

Rathaxes est composé de quatre équipes avec chacune un rôle bien distinct. Chaque équipe est supervisée par un chef d’équipe qui doit rendre compte des avancées de son groupe tous les mois durant les réunions chef.

Les équipes sont donc les suivantes :

### Equipe hardware

Membres :

* Responsable : Sylvestre Gallon
* David Amsallem
* Tomas Suarez
* Mickael Dumont

Cette équipe a pour rôle d’effectuer tout le travail de recherche nécessaire à Rathaxes. La première de ses recherche et de comprendre comment fonctionne les pilotes de périphériques pour chaque système d’exploitation visé (Windows, OpenBSD, Linux et MacOSX). Le deuxième travail de recherche est de généraliser l’écriture de pilotes, c'est-à-dire trouver les points commun entre les différents systèmes d’exploitation et les différences. Ce travail a pour conséquence la production d’un papier qui sera transmis à l’équipe langage.

### Equipe Langage

Membres :

* Responsable : David Giron
* Adrien Silvestre
* Vivien Jacquemoze

Cette équipe a en charge la conception du langage Rathaxes ainsi que de son compilateur. Cette production est rendue possible grâce au travail de recherche de l’équipe hardware. Un gros travail de recherche est également réalisé par cette équipe sur le fonctionnement des langages.

### Equipe Transverse

Membres :

* Responsable : David Verrière
* Christophe Fajardo
* Marc Thompson

Le rôle de cette équipe est de s’assurer de la qualité du projet, de coordonner les différentes équipes et d’apporter son soutien à chaque équipe en cas de besoin.

### Equipe Infra-Com

Responsable : Sébastien Cassier

Cette équipe s’occupe de la mise en place de toute la logistique nécessaire à Rathaxes ainsi que de la communication autour du projet. Elle assure également la préparation des conférences.

## Planning Prévisionnel

## Diagramme de Gant

**03/2008**

**12/2007**

**10/2007**

**BUS/DMA Research**

**IRQ Research**

**LKM Research**

**Language Team**

**Hardware Team**

**12/2008**

**BLT Writing**

**BUS/DMA MemoryMap Implementation**

**BUS/DMA Memory Map**

**BUS/DMA Research**

**BUS/DMA Implement**

**06/2008**

**09/2008**

**BUS/…**

**IRQ Implementation**

**LKM Implementation**

## Instances internes

Afin d’assurer un suivi régulier de l’avancement du projet des réunions seront organisées tous les mois.

Les réunions seront de deux types, les réunions chef d’équipes qui serviront à :

* suivre l’avancement du projet par équipe
* tirer les conséquences d’éventuels échecs
* décider des nouvelles orientations pour le projet
* Coordonner le travail de tous

Les réunions générales qui serviront à :

* Informer toute l’équipe de l’avancement du projet
* Auditer les développeurs
* Débattre sur différents thèmes au besoin

L’animation des réunions sera tenue par le chef de projet ou un membre de l’équipe transverse.

Les comptes rendu seront rédigés et communiqués par l’équipe de communication.

Enfin des réunions internes par équipe pourront être organisées selon le souhait des chefs d’équipe.

## Instances externes

### Partenaires

Rathaxes est en partenariat avec le laboratoire sécurité et système du pôle informatique Ionis. Il s’agit d’une relation bidirectionnelle avec d’une part l’apport de ressources aussi bien physique qu’intellectuelle de la part du laboratoire. En échange Rathaxes offre toutes ses recherches et avancées au laboratoire.

### Conférences

Dans le cadre de sa promotion Rathaxes animera différentes conférences de présentation du projet. Ces conférences seront réalisées au sein d’Epitech mais aussi dans différents salon tel que la RMLL. Le but de ces conférences est de faire connaitre le projet pour lui donner un maximum de visibilité sur celui-ci.

### Réunions

Dans le cadre du partenariat avec le laboratoire sécurité et système du pôle informatique il est possible que, lors de certaines de nos réunions, Lionel Auroux directeur du laboratoire soit présent et nous aiguille dans certains de nos choix technique ou organisationnel.

### Apport d’une expertise externe

Le concepteur du logiciel utilisé pour développer Rathaxes nous offre une expertise et propose de développer les possibles fonctionnalités manquante dans son logiciel. De plus il nous apporte un soutien technique sur son logiciel.

# Prestation et produits livrable

Rathaxes : Approche langage à la génération de pilotes de périphériques robustes multiplateformes

Sous ce long titre se cache une réelle solution à un des grands problèmes informatiques : l’écriture de pilote.

Cette écriture est très complexe et soumis à de nombreux bugs, de nombreuses études vont dans ce sens : le dernière rapport de crash de Windows Vista notifie que 30% de ses crashes systèmes sont dus aux pilotes NVIDIA, 18% de pilotes ATI ; On arrive donc à prêt de **50% de crashes système causés par des pilotes.**

Lorsque les pilotes sortent dans des versions à peu prêt stable, **ils ne sont pas portables**, le développeur doit donc réécrire son pilote pour un chaque système d’exploitation sur lequel il souhaite le porter.

Rathaxes apporte une vraie solution à tous ces problèmes en proposant **un langage de génération de pilote, un compilateur et une librairie de templates** par système d’exploitation cible.

Grâce à son langage simple et intuitif, Rathaxes permet de réduire les temps de développement de manière considérable et d’offrir une vision claire des différentes problématiques au développeur.

Le compilateur vérifiera les erreurs courantes de syntaxe et de sémantique et ainsi évitera de longues phases de débogages couramment réalisées par le développeur.

# Démarche

Dans un premier temps nous avons étudié les besoins puis nous avons étudié la faisabilité du projet, puis nous avons mis en places les exigences logicielles avant de mettre en place le plan de mise en œuvre du projet.

**Etude des besoins :**

Les besoins ne sont pas clairement exprimés par les développeurs de pilotes de périphériques. Ils résultent de faits et d’études faites par lors de thèses (comme la thèse de Laurent Réveillère). Nous somme aussi parti du fait que les pilotes de périphérique sont briques logiciel sensible dans un système d’exploitation, Il faut donc qu’ils soient irréprochable sinon ils provoquent des instabilités du système d’exploitation que vous utilisez. Donc pour rendre plus simple et plus sécurisé le développement de pilotes de périphérique, une solution possible est : la génération de briques logiciel testé et considéré comme robuste.

**Etude de la faisabilité :**

Pour l’étude de la faisabilité du projet nous nous sommes basé sur le travail le Laurent Réveillère et de l’équipe du projet NDL. Qui prouvent, même s’ils ne sont pas complets, que le projet est réalisable. Après une études des différents types de pilotes de periphérique et de leur mise en œuvre nous avons détouré des couches identiques et répétitives. Cela nous permets donc, aussi, de dire que le projets est viable.

**Exigences logicielles :**

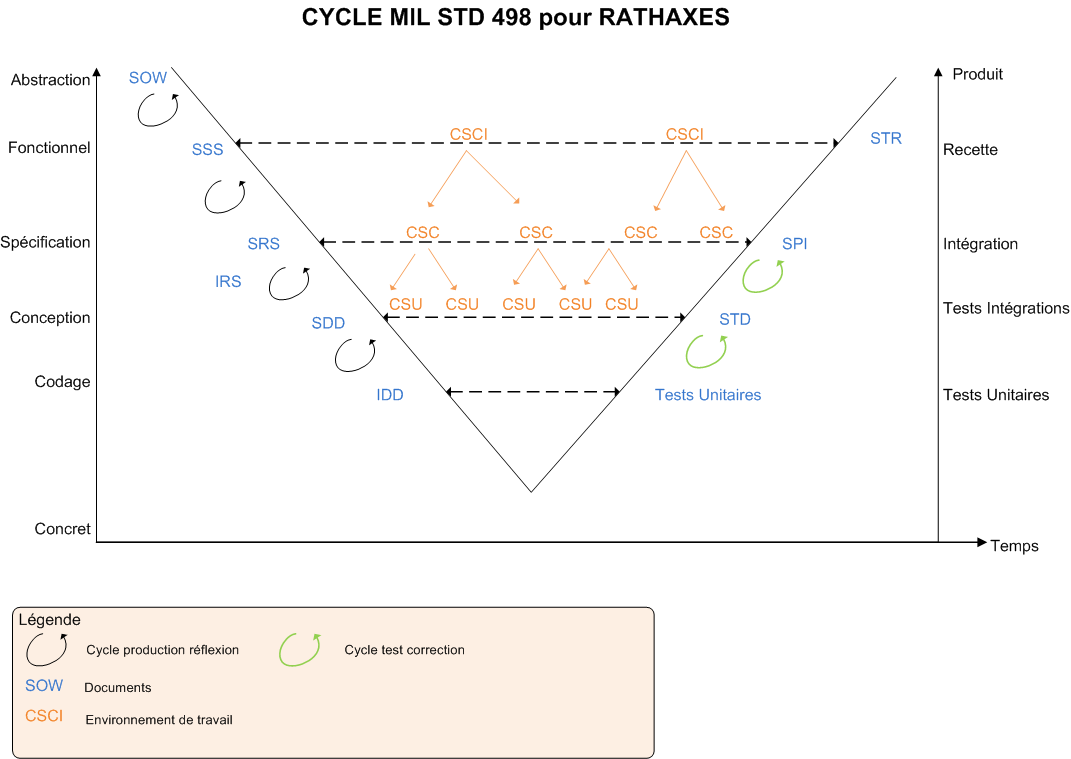
La première phase du projet a été de définir quels étaient les fonctionnalités et les limites du projet. Pour cela nous avons mis en place ce que la méthode de développement MIL-STD498 appel un Statement Of Work.

Nous avons pus definir les exigeances suivante :

1. Rathaxes doit être un langage de description de driver : décrit le comportement d’un driver indépendamment de la plateforme hôte.
2. Rathaxes doit inclure le compilateur du langage de description en code exploitable pour une plateforme de destination.
3. Rathaxes doit fournir une librairie de Templates extensible
4. Rathaxes doit fournir un outil de déploiement pour chaque plateforme supportée.
5. Rathaxes doit fournir une documentation couvrant tous les aspects sus cités
6. Le projet doit être maintenable et maintenu
7. Le projet doit être connu et reconnu
8. Le projet sera soumis à des licences open sources

**La mise en œuvre du projet :**

La mise en œuvre du projet suis un modèle de développement dit «  de cycle en V » ici défini par la méthode de développement MIL-STD498. La MIL-STD498 fonctionne de la façon suivante.



Pour chaque étapes du projets nous allons définir les briques logiciels avant des les développé pour les redéfinir si l’implémentation de celles-ci ne conviennent pas. Rathaxes étant un projet de recherche il est normal d’adopté une telle méthode. Cella nous assurera que chaque briques logiciel concervent une coherance entre elle, puisqu’elle ne seront pas développé au même momment.

# Gestion des événements

**Problem with systems headers**

**Mai 2008**

**Cache has been developed for each OS**

**MACOSX Driver**

**are in C++**

**We don’t have C++ parser, MacOSX development was suspend for a time**

**April 2008**

# Processus de support

Les processus de support sont l’ensemble des procédures logistiques existantes au sein du groupe :

* Utilisation d’un logiciel de versionning : SVN.
* Procédure de diffusion d’information au sein du groupe :
  + Générales : Mailing lists organisée par thèmes
  + Bug Report : Création de ticket et assignation sur trac
  + Documentation : site web
* Procédure de sauvegarde du serveur : Script Shell créé par l’équipe Infra-Com exécuté de manière automatique.
* Réunions mensuelles planifiées

# Maitrise des fournitures et Reproduction, production et livraison

### Fournitures

Les fournitures sont composés à la fois de matériel et de technologies utilisées dans le développement du projet.

D’un point de vue matériel, sont utilisées :

* 4 machines de test sous différents systèmes d’exploitation.
* Un serveur centralisant les outils logiciel de gestion de projet.

D’un point de vue technologique, sont utilisées :

* Codeworker : Un langage de script orienté vers la description de grammaire et de parseurs, utilisés pour l’exécutable Rathaxes.
* Latex : Standard dans la rédaction de papier scientifique, utilisés pour nos documentations.

### Reproduction production et livraison

* Licence AS-IS documentations , BSD pour le code et les templates, LGPL pour la langage
* Ces licences permettent à Rathaxes d’être open source tout en gardant un contrôle sur le projet

# Gestion du PAQ

Le PAQ est principalement géré par les équipes Com et transverse de Rathaxes mais des intervenants des équipes Hardware et Langage participent à la relecture de ce dernier.

Le PAQ est mis à jour après chaque intervention du responsable de matière EPITECH.

# Bilan projet

Après plus d’un an de travail sur le projet Rathaxes, les délais ont été respecté jusque la. Le projet génère des versions simples de code de pilotes de périphérique. Le projet a remplit la partie promotion en participant le 4 juillet 2008 aux réunions mondiales du logiciel libre prés de Bordeaux.

Le projet a suivi les exigences du PAQ et de la MILST498 ce qui a permis de régler rapidement les problèmes.

Nous utilisons nos propres outils de versionning et de communication, nous avons décliné l’offre de support de l’équipe PFE.

Nous avons du malheureusement abandonner les recherches sur un des OS, MacOSX car nous n’avons pas actuellement de parseur C++.

# Garantie et réclamation client

Le projet n’a actuellement aucune garantie de part sa nature non commerciale. Les réclamations des utilisateurs sont prises en compte via la mailing liste et le site internet du projet. L’un des buts du projet est de créer une communauté open source de sorte de le maintenir.