



Ecole Française d'Athènes – Service informatique
6 rue Didotou, 106 80 Athènes, Greece.
Téléphone : +30 210 36 79 949

– Porphyre 2003 –

Gestion du développement collaboratif

- Choix d'une méthodologie de travail -

Version 2.0

Date de création : 15/04/03

Dernière modification : 22/08/03

Etat du document : validé

Nombre de pages : 14

Rédacteur : Michel NUX

SOMMAIRE

1.	INTRODUCTION.....	3
1.1.	PRESENTATION DU SYSTEME PORPHYRE	3
1.2.	OBJET DU DOCUMENT	3
1.3.	DOCUMENTS DE REFERENCE.....	4
2.	DES OUTILS DE TRAVAIL COLLABORATIF	5
2.1.	ATELIER UML.....	5
2.2.	GESTION DE FICHIERS / VERSIONS.....	7
2.3.	GESTION DE RESSOURCES ET DES TACHES.....	8
2.3.1.	<i>La liste des tâches.....</i>	<i>8</i>
2.3.2.	<i>L'annuaire de la communauté</i>	<i>9</i>
2.3.3.	<i>Le calendrier commun</i>	<i>9</i>
2.4.	DEVELOPPEMENT	9
2.4.1.	<i>La programmation.....</i>	<i>9</i>
2.4.2.	<i>Les tests de non-régression</i>	<i>10</i>
2.5.	REDACTION DES DOCUMENTS	10
2.6.	COMMUNICATION	10
2.6.1.	<i>L'Email</i>	<i>10</i>
2.6.2.	<i>Le « chat ».....</i>	<i>11</i>
2.7.	LE PORTAIL	11
3.	METHODES DE TRAVAIL.....	12
3.1.	UN VOCABULAIRE COMMUN	12
3.2.	LA MODELISATION OBJET : UML.....	12
3.3.	LA GESTION DE VERSIONS.....	13
3.4.	LES TESTS DE NON-REGRESSION.....	13
4.	SYNTHESE.....	14

1. Introduction

1.1. Présentation du système Porphyre

Le projet *Porphyre* est le fruit de plusieurs années de recherche et de développement initiées par l'Ecole Française d'Athènes (EFA). Il a été soutenu par le Ministère de la Recherche et le Ministère de l'Education Nationale et de l'Enseignement Supérieur. Il a fait l'objet de collaborations avec le LIRIS (ex-LISI, laboratoire de recherche) et le département informatique de l'INSA de Lyon. Il s'est concrétisé au travers de thèses et de nombreux stages.

Porphyre n'est qu'au début de son évolution que ce soit en terme de développement ou de recherche interdisciplinaire (Informatique et SHS-STIC : Sciences de l'Homme et de la Société – Sciences et Technologies de l'information et de la Communication). Par l'apposition des licences GNU GPL et FDL à l'ensemble du projet Porphyre, l'équipe Porphyre a une volonté d'élargir la recherche et le développement (R&D) aux institutions publiques et aux personnes voulant participer à l'avancement du projet.

Dès lors que chacun peut participer à un projet de R&D de grande d'envergure, beaucoup de monde risque d'avancer « dans son coin ». Quelques désagréments peuvent alors apparaître, dont celui de voir se développer autant de prototypes (incompatibles ?) que d'axes de recherche. La dispersion autour du projet de recherche Porphyre est donc à surveiller en premier lieu. Mais, il existe aussi d'autres problèmes qui apparaîtront inmanquablement, comme celui d'utiliser des outils incompatibles, ou encore comme la difficulté de communication entre tous les intervenants du projet. Pour éviter cela, une communauté Porphyre est proposée, possédant une méthodologie de développement collaboratif.

1.2. Objet du document

Ce document expose la méthodologie de travail répondant aux besoins du projet Porphyre pour permettre d'unifier les méthodes et outils utilisés en vue d'une gestion homogène du développement collaboratif et d'assurer une synergie autour d'un projet interdisciplinaire et de la communauté.

1.3. Documents de référence

Un ensemble de 4 documents de référence précise, décrit et organise aussi bien la communauté et le projet de R&D que l'interaction entre ces deux derniers (cf. figure 1.3).

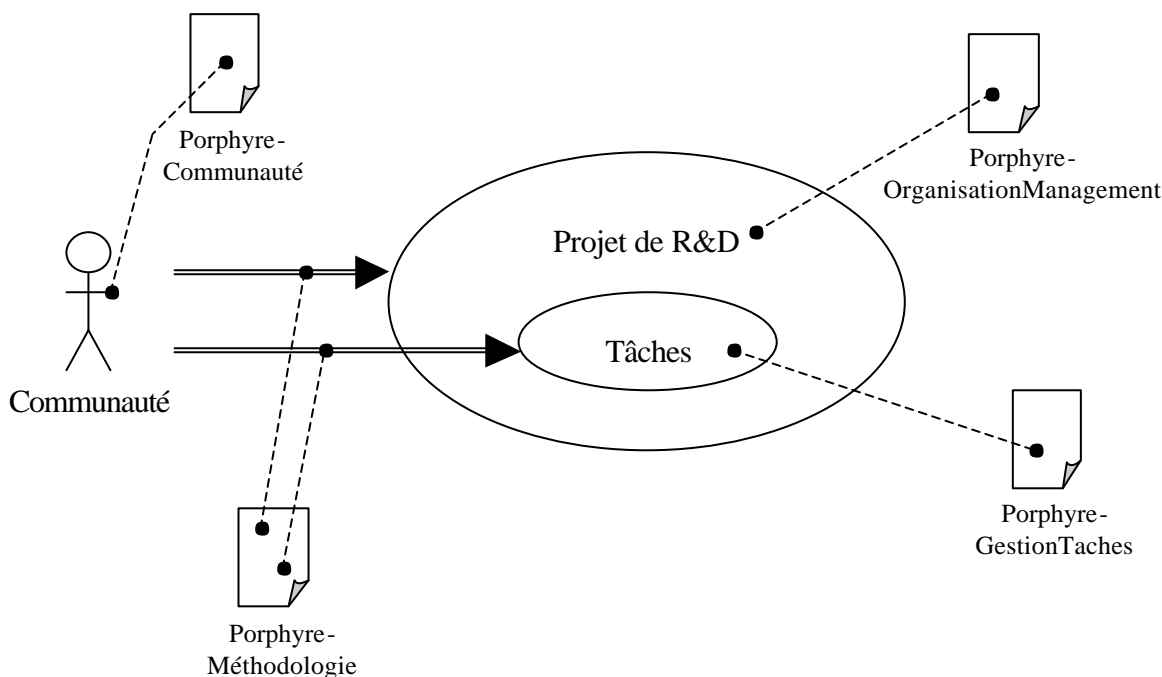


Figure 1.3 : Vue d'ensemble de l'application des documents de référence

La documentation décrivant la communauté Porphyre, ainsi que les règles organisationnelles et méthodologiques utilisées pour le développement collaboratif du projet de R&D Porphyre, est constituée de :

- Un document « **Porphyre-Communauté** » qui a pour but de présenter et définir la structure de la communauté Porphyre (qui s'est créée autour du projet de R&D) afin que chacun puisse se renseigner puis rejoindre à cette communauté.
- Un document « **Porphyre-OrganisationManagement** » qui décrit de façon générale les principaux concepts de la méthode de management du projet de R&D Porphyre ainsi que son organisation en tâches.
- Un document « **Porphyre-GestionTaches** » qui détaille le processus de gestion des tâches du projet de R&D ainsi que la forme et la composition des documents-clés et des livrables.
- Ce document même, « **Porphyre-Méthodologie** », qui présente la méthodologie de travail, répondant aux besoins du projet Porphyre, pour permettre d'unifier les méthodes et outils utilisés en vue d'une gestion homogène du développement collaboratif.

2. Des outils de travail collaboratif

Dans une optique d'homogénéisation, aussi bien dans le travail que dans le formatage des données, il faut proposer un ensemble d'outils commun à l'ensemble de la communauté ; en précisant également le formatage du travail et des informations issues de ces outils.

Dans l'intérêt d'accrocher un large public à l'investissement au projet Porphyre, il est préférable, voire indispensable, de proposer un ensemble d'outils « gratuits », c'est à dire libre d'utilisation dans le temps. Nous voulons réduire au maximum, voire rendre nul, le coût des outils indispensables pour travailler sur le projet. Ainsi, toute personne voulant s'investir au sein de la communauté pourra le faire sans déboursier d'argent pour l'achat de quelques licences de logiciels. Il pourra également, dans cette même optique, avoir accès et s'informer sur l'ensemble de l'étude se rapportant au projet, puis pourra librement l'étudier et le modifier.

Etant donné le caractère (fortement) évolutif des fonctionnalités, des versions comme des droits d'utilisations (licences) des logiciels, nous proposons un certain nombre d'outils actuellement très intéressants pour des raisons qui seront développées plus loin. Par contre, d'autres outils peuvent être utilisés ; notamment si la personne, l'institut ou le laboratoire de recherche, travaille déjà avec des outils dont il a l'habitude ou les licences. Il est très important de noter le caractère évolutif des outils de travail au sein de la communauté. Cette évolution pourra se faire par exemple par des propositions de personnes ayant travaillé, tester et approuver certains outils sur le projet en question.

2.1. Atelier UML

Le nombre d'ateliers UML existant actuellement est assez important. Mais ceux qui sont libres d'utilisation ou qui proposent le large panel de fonctionnalités que nous souhaitons, font partie d'une liste très restreinte. Nous n'allons pas en faire un catalogue, mais juste proposer les quelques uns qui nous semblent intéressants [cf. tableau 2.1].

Depuis l'adoption de la norme UML comme méthode de modélisation, beaucoup de logiciels « outils » se sont développés. Beaucoup même sont très complets d'un point de vue des fonctionnalités offertes telles que la génération de codes sources ou de documentations et le « reverse-engineering », ou d'un point de vue des OS nécessaires. Mais très peu proposent tout cela gratuitement. L'exemple de l'outil « Together » le montre bien : alors qu'il était en utilisation gratuite il y a quelques temps, il désormais payant, et il faut acheter une licence par poste.

Le fait de devoir acheter une licence par poste, ou par utilisateur, lorsque l'on s'adresse à une communauté n'est pas la meilleure solution, surtout lorsque l'on souhaite que des personnes s'investissent dans le projet et rejoignent la communauté. C'est pourquoi nous allons nous tourner de préférence vers des outils d'utilisation gratuite.

Par contre, un standard d'échange des informations UML existe, c'est une extension XMI de la norme XML. Nous allons donc dans la mesure du possible essayer d'utiliser ce format d'échange universel entre les différents outils UML existants (cf. figure 2.2).

Outil	Licence d'utilisation	Diagrammes UML	Génération de code Java	Importation de projet en XMI	Exportation de projet en XMI	O.S.	Divers
Argo UML	Gratuite	✓	✓	*		environnement Java	
Objecteering	Gratuite pour l'« Edition Personnelle »	✓		✓		- Linux - Windows	Génération de documentation [* .html ; *.rtf ; *.ps]
	Payante pour la « Version Professionnelle »	✓	✓	✓	✓	- Solaris	
Poseidon for UML	Gratuite	✓	✓	*		environnement Java	
Rose	Payante	✓	✓	✓	✓	- Unix - Solaris - Windows - SGI IRIX ...	« reverse engineering »
Together	Payante	✓	✓	✓	✓	- AS400 - Linux - Macintosh - Solaris - Windows	- Génération de documentation - « reverse engineering »
Umbrello UML	Gratuite	en partie	✓	✓		environnement KDE	

Tableau 2.1 : Tableau comparatif des outils UML intéressants pour le projet Porphyre

* NotaBene : « Poseidon for UML » et « ArgoUML » proposent l'importation et l'exportation de fichiers standards XMI. Mais après approfondissement, cela ne reste possible que seulement entre deux logiciels identiques (par exemple entre deux logiciels Poseidon). Pour que cela soit possible avec les autres logiciels du marché, il faut acheter la licence d'un logiciel jouant le rôle de « passerelle ».

L'atelier UML choisi dans le cadre du projet de la communauté est Objecteering « Edition Personnel » qui est une version gratuite. Objecteering est utilisable avec les OS les plus courants (Linux/Unix, Windows). Il permet d'exprimer l'ensemble des diagrammes UML pour la modélisation, il possède également des fonctionnalités de génération de documentation (RTF, HTML, PS) en rapport avec le modèle défini (ce qui permet de garder une cohérence entre la documentation et le modèle UML), et permet aussi d'importer des projets sous forme standard XMI. Ce format standard d'échange fonctionne très bien entre Objecteering et Rose, et cela dans les deux sens.

Il faudra notamment que le «noyau» de la communauté se procure une licence de la «Version Professionnelle» du produit Objectteering pour exporter au format XMI le projet UML (ou tout autre outil UML permettant d'exporter au format standard XMI, par exemple «Rational Rose»). Ensuite, tout le monde pourra, via un outil UML permettant l'importation de projet en XMI, étudier et faire évoluer le modèle.

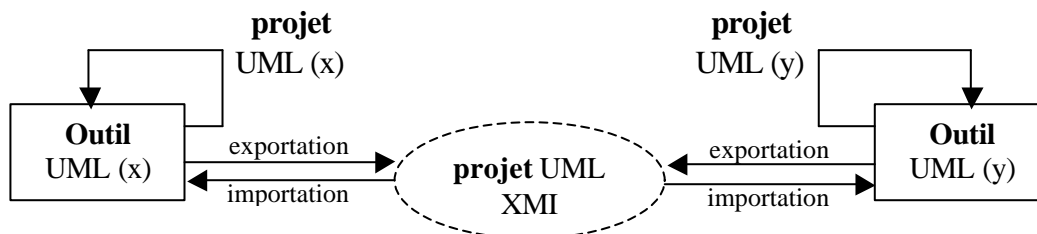


Figure 2.2 : Format d'échange entre deux outils UML

Dans un premier temps, alors que la communauté est en train de se construire, la documentation HTML et RTF/PS générée par Objectteering «Edition Personnel» convient tout à fait pour étudier le modèle UML par des personnes ne possédant pas d'outil UML.

2.2. Gestion de Fichiers / Versions

Actuellement, VSS (Visual Source Safe, produit Microsoft utilisé par l'EFA) est utilisé pour la gestion de versions de l'ensemble du projet de développement, alors qu'aucun serveur ne recense les travaux de recherche. Mais à l'utilisation de VSS, nous préférons un outil de Gestion de Version libre d'utilisation pour des raisons déontologiques vis à vis de la communauté.

Conçu tout d'abord pour des projets impliquant plusieurs développeurs, Concurrent Versions System (CVS) est le seul outil de Gestion de Version offrant un panel de fonctionnalités assez exhaustif et ayant opté pour une licence de Logiciel Libre GNU GPL. Ce qui permet à chacun de l'utiliser, voire le modifier, notamment par la communauté pour le projet Porphyre.

CVS est un serveur de gestion de fichiers, et il est la seule référence en terme de cohérence et d'intégrité des données au sein de la communauté. Chaque utilisateur accède seulement à une copie du fichier sur lequel il veut travailler. Quant à lui, le fichier original reste accessible par d'autres sur le serveur CVS. Ensuite, la mise à jour de fichiers sur CVS se fait par des fonctionnalités sécurisées propres au logiciel.

CVS n'est pas seulement un logiciel qui archive des fichiers (par duplication pour des versions différentes), mais il gère les révisions apportées aux documents grâce à des «Tag». Ainsi, on peut extraire la version "v2.5" du projet, même si le projet est beaucoup plus avancé et se trouve à la version "v3.1" par exemple. Aussi, CVS offre la possibilité de suivre l'évolution d'un projet par ses versions et, si nécessaire, de rebrousser chemin.

Par contre, l'outil CVS comporte des limites à son utilisation. Tout d'abord, on constate souvent un triplement du volume (en terme de taille mémoire, en byte) des fichiers sources «utiles» d'un projet de développement informatique. Mais cet encombrement est largement réduit par rapport à l'encombrement si nous avons dû conserver l'ensemble des versions successives du projet. Ensuite, CVS gère les versions concurrentes de fichiers, mais ne dispense en aucune manière la communication entre les membres de la

communauté. Par exemple, CVS ne peut en rien aider à la résolution de conflits relatifs à des « logiques » de programmation. Et pour finir, l'outil gère très bien tous les fichiers textuels, mais il ne possède aucun outil pour gérer les fichiers binaires, ni les fichiers graphiques (images, graphe, ...) ; ces derniers fichiers seront donc seulement archivés de manière classique sur le serveur de fichiers.

Par défaut CVS fonctionne en mode « ligne de commandes » (sans interface graphique). Mais il existe un ensemble de clients graphiques gratuits :

- Cervisia (pour Unix)
- WinCvs (pour Windows)
- Tortoise (plugin qui permet de gérer les fichiers via l'Explorer de Windows)
- MacCvs (pour Macintosh)
- CvsWeb (permet d'explorer le contenu de CVS via un navigateur Web)

Par ailleurs, CVS propose également une documentation détaillée d'installation et d'utilisation qui reste accessible librement tout en respectant les termes de la licence apposée (comme pour l'ensemble du projet CVS à l'URL suivante : <http://www.cvshome.org/>). Un avantage supplémentaire est l'accès par différents sites de documentations d'installation et d'utilisation traduites en de nombreuses langues. Il en est de même pour chacun de ces clients CVS qui proposent également une documentation d'installation et d'utilisation qui reste accessible librement (tout en respectant les termes de la licence apposée).

2.3. Gestion de ressources et des tâches

Il existe plusieurs logiciels commerciaux qui permettent la gestion de projet avec les ressources associées au projet en question. Mais dans le monde du Logiciel Libre, il en existe très peu (Tutos, sous GNU GPL). Dans les deux cas, ces logiciels sont trop complets ou trop contraignants pour un projet aussi spécifique que Porphyre.

Ce que nous désirons pour la communauté et le projet Porphyre, c'est d'avoir des informations concernant les ressources (nom, prénom, Email, statut, etc...) et sur les tâches (titre, priorité, spécification, complexité, etc...), et surtout de pouvoir proposer et modifier la priorité des tâches.

Donc pour gérer les quelques ressources de la communauté et les tâches du projet, nous allons nous orienter vers une gestion « à la main » avec un fichier tableau ou texte, ou alors en créant une petite base de données (avec MySql ou PostgreSQL, par exemple, favorisant le lien avec le site officiel www.porphyre.org).

2.3.1. La liste des tâches

Une liste des tâches est consultable sur le site officiel (et une version se trouve également sur le serveur de fichiers CVS de la communauté). Cette liste recense l'ensemble des tâches du projet : celles en cours de réalisation, celles à faire et celles en discussion. Pour chaque tâche, nous connaissons son nom, sa priorité, sa difficulté, sa spécification (1 page généralement),...

Dans le cas où des tâches ne sont pas parallélisables lorsque certaines en prédéfinissent d'autres, nous utiliserons un diagramme de Gantt pour présenter leur enchaînement.

2.3.2. L'annuaire de la communauté

Un annuaire recensant les coordonnées des personnes, des organismes et des instituts appartenant à la communauté Porphyre existe pour faciliter la communication et l'échange entre individus.

Cet annuaire, accessible par le site officiel, présentera seulement les noms et Email des personnes, des organismes et des instituts de la communauté. Par contre, l'ensemble des informations concernant toutes les personnes de la communauté (adresse, téléphone, date anniversaire, emploi, société, etc...[sous réserve d'accord de la personne]) est accessible dans l'annuaire figurant dans le serveur de gestion de fichiers CVS, et donc accessible seulement à la communauté.

2.3.3. Le calendrier commun

Un calendrier porphyre, commun à toute la communauté, doit être utilisé pour indiquer les informations les plus importantes liées à la R&D du projet, à la communauté, etc...

Ce calendrier, mis à jour de façon pertinente, est consultable sur le site web officiel de la communauté, ou par CVS pour les membres de la communauté.

2.4. Développement

2.4.1. La programmation

La plate-forme de travail Porphyre est actuellement développée, en grande partie, en Java. Il faut donc à toutes personnes voulant contribuer au développement, d'abord installer l'environnement de travail Java «*jdk*» et «*jre*» (gratuit) et quelques bibliothèques supplémentaires qui permettent ensuite la compilation et l'exécution du projet.

Jusqu'à présent, nous avons utilisé l'outil de développement «*Kawa*». Mais depuis peu, cet éditeur de codes n'est plus libre d'utilisation comme auparavant. Désormais, *Kawa* propose seulement une version limitée dans le temps de 15 à 30 jours suivant la version. Nous ne pouvons pas proposer d'un point de vue légal au téléchargement (sur le site officiel de porphyry.org) une version antérieure et libre d'utilisation dans le temps de *Kawa*, pour que les membres d'une communauté puissent travailler librement. Par contre, les personnes possédant une version ultérieure gratuite de *Kawa* peuvent l'utiliser car cet outil a déjà fait ses preuves durant plusieurs années pour le développement de la plate-forme de travail Porphyre.

Il nous faut donc proposer d'autres outils d'édition de codes qui soient libres d'utilisation dans le temps. En ce qui concerne les éditeurs de textes et de codes, plusieurs sont disponibles gratuitement, comme «*Xemacs*», «*CygWin*» ou «*Kate KDE*» qui sont des éditeurs de codes Java (avec coloration syntaxique). Mais les logiciels classiques de développement Java et les éditeurs classiques de textes (comme «*Vi*» pour Linux ou «*NotePad*» pour Windows) conviennent parfaitement.

Par ailleurs, l'outil «*astyle*» (*artistic style*) permet entre autre d'uniformiser l'indentation (notamment pour les sources Java), et donc de régler les problèmes dans les codes sources dus à l'utilisation de différents éditeurs.

Pour la compilation, étant donné que tous les éditeurs de codes n'incluent pas à chaque fois une fonction ou un lien vers un compilateur Java, il est toujours possible d'ouvrir une console et d'y indiquer les lignes de commandes pour compiler le projet, voire d'élaborer un fichier «*make-file*».

2.4.2. Les tests de non-régression

Les tests de non-régression seront disponible au fur et à mesure en fonction de l'évolution du prototype informatique Porphyre. A chaque version stable de la plate-forme, nous devons la proposer accompagnée d'un jeu de tests qui seront alors les tests de non-régression. Ces tests de non-régression doivent être validés par toutes les autres versions stables qui la suivront dans le temps (sous condition d'une continuité de travail dans un même axe de R&D), et ces futures nouvelles versions stables seront également accompagnées d'un jeu de tests de non-régression.

Pour le moment, aucun outil de tests n'est officiellement utilisé pour élaborer des jeux de tests, pour éprouver et valider la plate-forme de travail Porphyre, mais l'outil « Junit », atelier de tests qui a une bonne réputation, peut-être d'une aide précieuse pour les développeurs et les testeurs. D'autres outils peuvent aussi être utilisés pour élaborer des jeux de tests.

2.5. Rédaction des documents

Le choix des outils de rédaction de documents est laissé au principal intéressé : le rédacteur. Le plus souvent, il s'agira des outils de bureautique les plus classiques et répandus : Microsoft Word (Windows & Mac), Star Office (Linux).

Sur le serveur de fichiers de la communauté, les fichiers en cours de rédaction apparaîtront sous leur format de travail (suivant l'outil de rédaction). Une fois le document finalisé, le fichier apparaîtra également sous un format permettant une consultation gratuite.

Sur le site officiel de la communauté, les documents seront proposés dans des formats dont les outils pour les consulter sont gratuits pour un large panel de systèmes d'exploitation : PDF et PostScript.

2.6. Communication

Sans outil de communication, il est impossible de coopérer au sein d'une communauté et de collaborer au développement d'un seul et unique projet.

2.6.1. L'Email

L'Email est sans doute l'outil de communication le plus utilisé pour communiquer à distance via le réseau entre deux ou plusieurs personnes. Devenue comme une boîte aux lettres virtuelle, la boîte Email est utilisée à la fois à des fins professionnelles et à des fins personnelles. Mais peu de personnes l'utilisent correctement, ce qui entraîne que certains Emails se perdent face à la quantité des messages reçus. « *Le gros problème du mail, c'est lorsque la capacité d'émission des messages dépasse la capacité de digestion des messages* »¹.

¹ explique Serge Levan, consultant en travail collaboratif chez Main Consultants, lors d'une interview pour le dossier « Travail collaboratif » de JDNet.

Ainsi, l'Email peut faire perdre beaucoup de temps à son utilisateur. En effet, il faut à chaque instant séparer les messages utiles des messages superflus. Ou pire encore, certaines informations particulièrement utiles sont perdues dans la masse de messages : « *trop d'info tue l'info !* ». Aussi, il est important d'utiliser cet outil de façon correcte. Attention, il n'y a pas de règles universelles, chacun se doit de réfléchir et de l'utiliser à bon escient. Il suffit de bien « hiérarchiser et personnaliser » l'information. Le destinataire du/des message(s) aura ainsi des informations plus ciblées et le concernant. Reste à chacun de bien structurer sa boîte Email, voire de se réserver un Email strictement à usage professionnel. C'est ce qui est conseillé vis à vis de l'utilisation de l'Email des membres de la communauté : « [nom.prénom]@porphyry.org ».

La communauté Porphyre offre à tous ses Membres une adresse Email du nom de la communauté. Mais attention à en faire un usage correct et principalement destiné au travail collaboratif au sein de la communauté. L'ensemble de l'outil Email sera à la charge des Modérateurs de la communauté (il serait intéressant qu'un modérateur en prenne la responsabilité et la gestion pour devenir le Webmaster de la communauté).

D'autre part, il est ensuite possible (à la charge du Modérateur) de créer des listes de diffusion suivant les besoins de la communauté. Mais là aussi, attention à ne pas proposer une extension de l'outil qui surchargerai les destinataires de messages non-importants. Par exemple, les listes de diffusion suivantes semblent indispensables : « community@porphyry.org », « bug@porphyry.org », « member@porphyry.org », « moderator@porphyry.org ».

2.6.2. Le « chat »

Le « chat » est un outil permettant à plusieurs personnes de discuter (en mode texte) en direct dans un espace virtuel. Chaque personne peut intervenir dans le cours de la discussion. Cet outil permet à plusieurs personnes (en des lieux différents) de s'entretenir sur un thème ou un sujet pour en dégager une problématique, une solution ou encore un avis.

L'utilité d'ouvrir un canal IRC propre à la communauté Porphyre pour communiquer entre Membres, semble peu intéressante face aux divers outils (ICQ, Messenger, Trilian, ...) existant déjà sur Internet et dont les fonctionnalités sont déjà assez exhaustives.

2.7. Le portail

Le portail du site web officiel de la communauté a pour but la valorisation du projet Porphyre, et offre à chacun une compréhension du projet (accès à tous aux informations), une utilisation de la plateforme de travail Porphyre (distribution suivant la licence GPL) et propose à chacun d'adhérer à la communauté Porphyre.

En plus des outils sur le site web présentant l'évolution du projet, le portail possèdera également un espace permettant de poster des messages : FAQ (Frequently Asked Questions). Pour déposer des messages et/ou répondre à des messages déjà déposés, l'Utilisateur devra préalablement s'enregistrer en ligne (nom, prénom, âge, profession,...) pour recevoir en retour un accès (login et password).

3. Méthodes de travail

Dans le désir de création d'une communauté interdisciplinaire autour du projet Porphyre, nous nous devons de réfléchir aux méthodes de travail que nous souhaitons utiliser et faire appliquer. Ces méthodes doivent préciser les méthodes utilisées et donner une inertie à l'ensemble du projet dans un sens défini et précis. Elles sont un peu des guides pour que le projet progresse avec une certaine Qualité ; aussi bien dans la documentation technique et les dossiers de recherche que dans l'implémentation, l'intégration et les tests du développement.

3.1. Un vocabulaire commun

Lors d'un travail collaboratif, et encore plus dans une communauté où les participants pratiquent des disciplines diverses et pas toujours « compatibles » du point de vue du domaine et du vocabulaire, il faut obligatoirement instaurer un vocabulaire commun pour le projet. En effet, trop souvent un même terme fait référence, pour diverses disciplines, à des idées très différentes.

L'ensemble du vocabulaire spécifique au projet, ou à une des disciplines, doit donc figurer dans un glossaire. Ce glossaire sert principalement à mettre en accord les usages du vocabulaire entre les diverses disciplines. Il pose pour chaque terme la définition, les hypothèses et/ou les limites d'usage pour le projet en question.

3.2. La modélisation objet : UML

UML est la méthode de modélisation informatique la plus intuitive et évolutive (en terme de modélisation Orientée Objet). Elle permet de modéliser à plusieurs niveaux les systèmes informatiques, leurs actions avec le monde extérieur ainsi que le flux de données. L'atout principal d'UML est de permettre d'exprimer et d'élaborer ces modèles objets, indépendamment de tout langage de programmation. Cela devient très intéressant dans le cadre de notre communauté où se côtoient des personnes de spécialités différentes (dans notre cas, recherche en sciences humaines et recherche en informatique). Ainsi, lors de la R&D, nous avons le loisir de bien spécifier le modèle de recherche sans se préoccuper de l'implémentation (pour les personnes qui ne connaissent pas bien les nouvelles technologies informatiques).

UML possède un standard connu et reconnu de tous les informaticiens, et utilisé par la grande majorité d'entre eux dans les spécifications et modélisations de projets. UML est défini par un méta-modèle et des normes (cf. norme UML 1.3 pour exemple).

UML est utilisable de manière transparente à tous les niveaux des cycles de développement de logiciels ou de systèmes informatiques. En effet, l'intégration d'UML dans un projet ne bouleverse pas et n'influe guère sur la méthode de travail choisie. Par contre, elle la complète et implique donc l'adoption d'une démarche méthodologique à son égard.

Comme la méthode UML est utilisable de façon transparente, elle offre quelques aspects intéressants dans l'approche de la méthode de travail sur un projet. Ainsi, de par son aspect modulaire et en permettant de définir précisément les tâches et leurs interfaces, elle facilite aussi bien le développement à une seule personne qu'à plusieurs personnes (et cela que les personnes soient dans une même équipe localement ou dans des équipes distantes), et même quand les domaines de compétences sont très hétérogènes. Ce fait de faciliter le travail de personnes distantes nous conforte dans ce choix pour la communauté Porphyre.

3.3. La gestion de Versions

En vu de promouvoir le projet Porphyre et sa communauté, nous allons proposer l'ensemble du projet en accès libre sur le site web pour les « Invités » et sur un serveur de données pour les membres de la communauté. Pour suivre plus facilement l'évolution du projet, que ce soit les codes sources ou la documentation de développement et de recherche, nous allons adopter l'instauration de versions comme il est d'usage en informatique lorsque l'on fait évoluer un projet.

Cette méthode de gestion de versions est utile et voire indispensable dans tout projet informatique comportant des itérations donnant à chaque fois des versions stables du projet. De même pour proposer une mise en ligne des résultats, il est plus correct vis à vis des « Invités » de leur offrir un ensemble structuré du projet où ils pourront se retrouver assez facilement.

Ensuite, la méthode de gestion de versions présente un intérêt pour les personnes de la communauté puisqu'elles pourront suivre plus facilement la succession et l'évolution du projet, des axes de recherche et des versions stables du prototype. Et si besoin, elles pourront revenir à une version antérieure ; si toutefois la version en cours n'aboutissait pas pour répondre aux besoins exprimés, ou si par exemple une modification des axes de recherche obligerait à repartir d'une version précédente.

3.4. Les tests de non-régression

Dans ce projet interdisciplinaire Porphyre, qui évolue sans cesse suivant des axes de recherche, il est important que chacun puisse bien contrôler qu'il n'y ait pas « régression ». Pour contrôler les éventuelles régressions possibles, nous pouvons proposer un jeu de tests appelé « tests de non-régression ». Ainsi, pour chaque version stable les membres de la communauté auront également un jeu de tests de non-régression. Et pour que la nouvelle version soit acceptée en tant que telle, il faudra d'abord qu'elle passe avec succès l'ensemble des tests de non-régression des versions précédentes.

Ainsi, cette méthode de tests de non-régression assure soit un avancement constant du projet, soit un stagnement (possibilité de stagnement lors, par exemple, d'une refonte de l'architecture de la plate-forme de travail). Elle est essentielle pour proposer des bases solides d'avancement dans la recherche et le développement du prototype.

4. Synthèse

Ce document propose un ensemble d'outils et de méthodes de travail afin d'homogénéiser le travail produit concernant le projet de R&D Porphyre, mais aussi pour assurer la viabilité de la communauté autour du projet de R&D et pour créer une synergie autour du projet et de la communauté.

Il est bien entendu et même indispensable que cette méthodologie de la communauté évolue (en même temps que l'évolution de la communauté et de la gestion de projet retenue) pour répondre et coller au mieux aux nouveaux besoins du projet et aux attentes des personnes appartenant à la communauté.