

Rapport de Synthèse

Gestion de la R&D dans un projet interdisciplinaire en Open Source

Michel Nux

Andréa Iacovella
Responsable du Service Informatique R&D
Ecole Française d'Athènes
6 rue Didotou
106 80 Athènes (Grèce)

Guy Caplat
LIRIS (ex-LISI) - INSA (Lyon)

Jean-Louis Sourrouille
L3i et PRISMa - INSA (Lyon)

Aurélien Bénel
Doctorant – INSA (Lyon)

Résumé

L'équipe du projet de Recherche et Développement interdisciplinaire Porphyre a la volonté d'élargir la R&D aux institutions publiques et aux personnes voulant participer à l'avancement du projet. Mais dès que chacun peut participer à un projet de R&D de grande d'envergure, chacun risque d'avancer « dans son coin », ce qui engendrera une dispersion autour du projet. Il faut donc proposer une communauté structurée, mais aussi prévoir une organisation, une gestion du projet et une méthodologie de développement collaboratif pour permettre à ces personnes de partager leurs connaissances et leurs expériences, et de participer à l'évolution du projet commun.

Ainsi, en proposant un système Porphyre basé sur une communauté active, sur une répartition des responsabilités, sur une organisation par tâches et sur des méthodes et outils de travail adaptés, nous voulons assurer une viabilité à long terme de l'ensemble du projet de R&D.

Mots clefs

Développement collaboratif, communauté interdisciplinaire, gestion de projet, organisation par tâches, répartition des responsabilités, méthodologie de travail.

Abstract Research

The project Porphyre is an interdisciplinary Research and Development project. The team of the project has the motivation to open the R&D to the public institutions and to the persons who want to participate in the promotion of the project. But, when each person can take part in a significant project of R&D, a lot of people risks to go forward to different ways which will produce a dispersion around the project. Thus, it need to suggest a structured community, but also to offer an organization, a project management and a methodology of collaborative development to allow these persons to share their knowledge and their experiences, and to join the evolution of the common project.

So, by proposing a Porphyre system based on an dynamic community, on a distribution of the responsibility, on an organization by tasks and on methods and working tools adapted, we desire to guarantee that a durable viability of all the project of R&D.

Key words

Collaborative development, interdisciplinary community, project management, organization by tasks, distribution of the responsibilities, working methodology.

I. Identification du projet

1. R&D 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.

2. problématique

Dès lors que chacun peut participer à un projet de R&D de grande d'envergure, chacun 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 inévitablement, comme celui d'utiliser des outils incompatibles, ou encore comme la difficulté de communication entre tous les intervenants du projet.

3. finalité

Dans la volonté d'ouverture de la R&D à tout public désirant apporter une aide, des idées ou des réflexions, il faut créer et organiser une communauté de R&D autour du projet Porphyre, dans l'optique de regrouper ces personnes de spécialités différentes sur un projet de recherche interdisciplinaire à vocation publique. Il faut non seulement structurer une communauté de personnes, mais aussi prévoir une organisation, une gestion du projet et une méthodologie de développement collaboratif pour permettre à ces personnes de partager leurs connaissances et leurs expériences, et de participer à l'évolution du projet commun. Il

faut en quelque sorte repenser et reconstruire le nouveau système Porphyre en fonction de ses objectifs à long terme. Nous désignerons par le terme « système Porphyre » l'ensemble du projet de R&D, sa communauté ainsi que son organisation et sa gestion.

II. Modélisation du système

1. description de la communauté

Un des avantages les plus appréciables d'une communauté est l'évolution des idées. En effet, acceptant des personnes de formations et de compétences diverses, l'ensemble du projet peut alors profiter d'une évolution plus riche aussi bien du point de vue de la recherche et du développement que des problématiques et axes de recherche, ce qui n'est pas négligeable dans un projet de R&D.

Les acteurs (cf. figure 1) qui se manifestent autour du projet Porphyre se classent en deux groupes appartenant à la communauté (Les Membres et les Modérateurs) et deux autres considérés comme « hors-communauté » (les Invités et les Utilisateurs) :

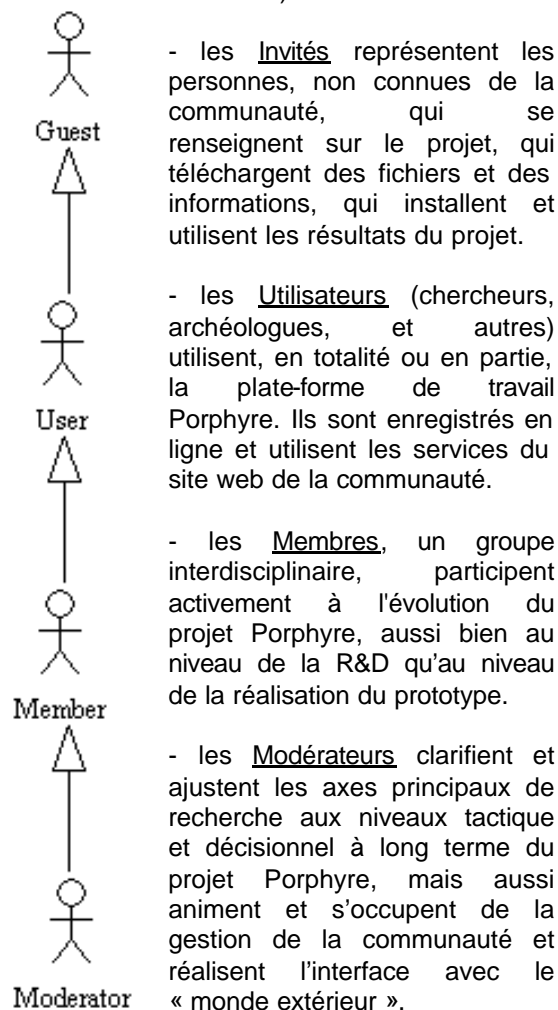


Figure 1 : Diagramme d'héritage des groupes

Cette communauté, dans un premier temps en tout cas, se verra composée principalement d'équipes de chercheurs et de développeurs qui formeront un noyau interdisciplinaire. Ce noyau de personnes pourra être également considéré comme le « moteur » du projet. Quant aux individus « seuls », ils ne seront pas très nombreux au début.

2. modélisation de la communauté

En réponse à cette communauté, plusieurs besoins du système Porphyre sont à prendre en compte et à organiser pour proposer ensuite une structure adaptée (cf. figure2).

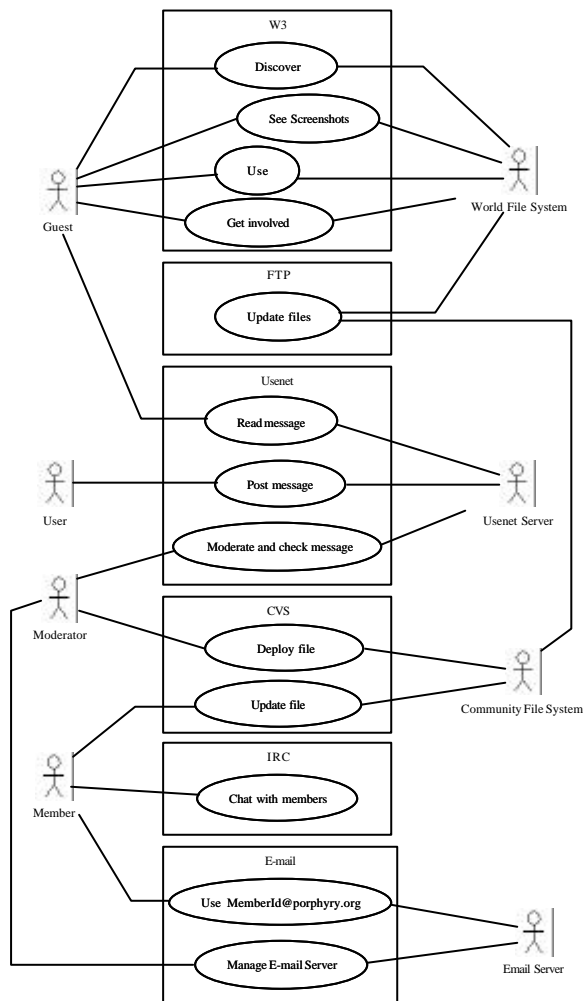


Figure 2 : modélisation du domaine

Cette structure de la communauté est primordiale pour garantir un lien entre les personnes et le projet Porphyre, et pour assurer la viabilité à long terme de l'ensemble du système Porphyre.

3. modélisation du projet

Etant donné que le projet Porphyre allie à la fois de la Recherche en sciences humaines et du Développement, il est important d'éviter la dispersion et de favoriser une cohérence entre

le modèle de Recherche et le modèle de Développement, qui se déroulent tous les deux en parallèle (de là vient la difficulté). Pour cela, nous allons travailler avec des « cycles de vie » assez courts pour augmenter la réactivité entre ces deux modèles (cf. figure 3). Nous appellerons donc chaque cycle : une *itération*.

Ces séries d'itérations, définies par les Modérateurs quand ils le jugeront utile, permettent d'augmenter les liaisons entre la Recherche et le Développement ; ainsi, nous pouvons mettre à jour, voire recadrer, l'ensemble du projet de R&D et que cela soit du modèle de Recherche vers le modèle de Développement, ou vice-versa. De plus, cette série d'itérations permet de proposer à tous et assez fréquemment des résultats et des prototypes opérationnels, ce qui permet un retour et un apport continu d'expériences favorisant une évolution constante du projet.

Peu importe si les itérations des deux modèles se superposent ou pas, ou si la durée des itérations ne sont pas les mêmes, parce que nous nous trouvons face à l'évolution en parallèle de la recherche et de l'application des résultats de la recherche au prototype informatique (pour valider les résultats obtenus en recherche). Moins courant, mais fort probable, les résultats du modèle du développement informatique peuvent également influencer sur le modèle de recherche.

Le principe est plutôt de rester en cohérence avec l'ensemble du projet. En augmentant le nombre d'itérations dans le temps, nous pouvons donc réagir plus rapidement aux nouvelles idées et modifications intervenues dans l'ensemble du projet de R&D.

III. Management et Organisation

Pour gérer l'ensemble du projet de R&D Porphyre, nous devons penser à une méthode de gestion de projet dans son ensemble. Beaucoup de méthodes de gestion et de management existent et ont déjà fait leurs preuves. Nous en avons essayé quelques-unes, en totalité ou en partie, dont la gestion classique [KRUC 00] et la gestion agile [MOIN 03] avec la programmation extrême. Mais à cause des nombreuses incompatibilités face à la particularité du système Porphyre, nous avons pris dans la mesure du possible les points forts de chaque méthode de gestion que nous voulions adopter pour créer une nouvelle organisation de management de projet dédiée entièrement à ce système. Cette organisation et gestion de projet a été éprouvée avec succès par les derniers développeurs lors de l'optimisation de l'architecture de la plate-forme de travail.

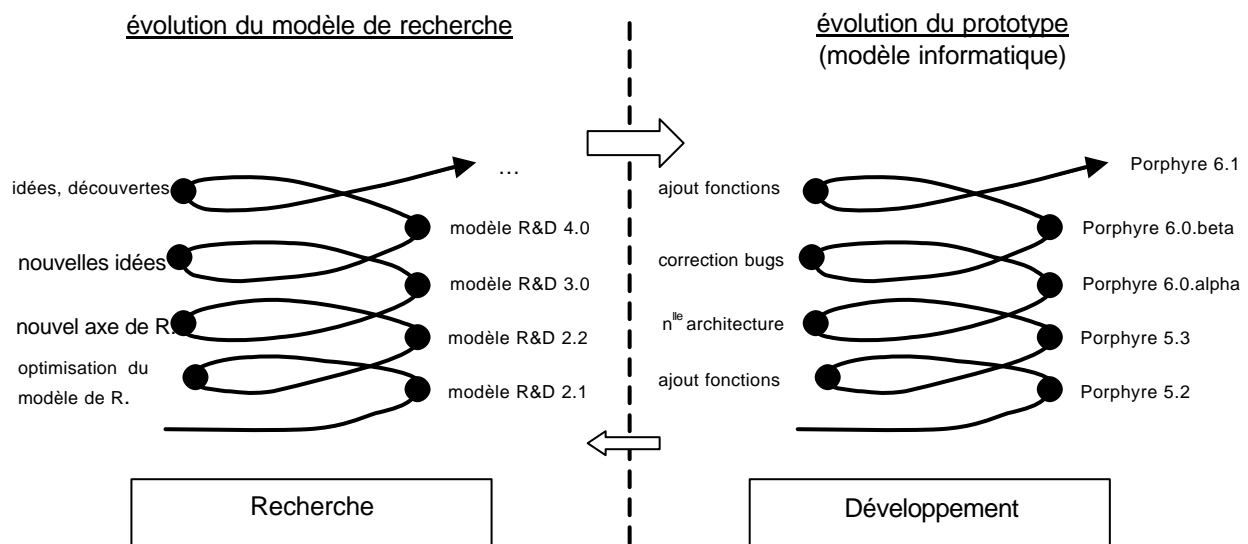


Figure 3 : Itérations du projet de R&D Porphyre

1. problématique et besoins

La gestion de projet doit répondre aux besoins et favoriser au mieux l'évolution de la R&D et de la plate-forme de travail associée. Pour cela, il faut une gestion qui puisse être très réactive vis à vis des nouvelles découvertes ou ré-orientations venant de la R&D (cf. figure3). Ainsi, nous allons nous orienter dans des itérations très courtes, pour permettre de remettre en cause assez fréquemment l'ensemble du projet et son évolution.

Les points importants à prendre en compte :

- l'aspect humain des individus de la communauté (personne n'a les mêmes méthodes de travail, de plus l'adaptation est une difficulté face à l'habitude).
- éviter un goulot d'étranglement lié à une minorité qui centralise, vérifie, intègre, corrige et valide le travail effectué par la communauté.
- pas de d'échéance, due à la notion de Recherche et Développement.
- laisser le choix au membre de la nature du travail à accomplir suivant son domaine ou ses expériences.

2. organisation par tâches

Nous appellerons « organisation par tâches » le fait que l'ensemble de la communauté découpe le projet Porphyre en tâches (sous contrôle et validation des Modérateurs), avec les priorités, les spécifications synthétiques et les niveaux de difficulté associés.

Lors de la validation d'une itération, des tâches peuvent être encore en cours de réalisation par les Membres de la communauté. Mais cela ne doit pas empêcher les Modérateurs de valider un état stable et de proposer une nouvelle itération. La tâche n'a pas de « deadline », mais elle peut être indispensable pour la

réalisation d'autres tâches (diagramme de Gantt).

Nous définissons le positionnement de la tâche au sein du projet, son interaction avec d'autres tâches, le tableau de bord de la tâche (interface avec le reste du projet, entrée/sortie), les documents livrables avec cette tâche. Ensuite, une fois la tâche bien définie dans son environnement projet, la tâche devient un projet à son tour.

Un (ou plusieurs) membre de la communauté peut ensuite prendre en charge la réalisation et l'intégration d'une tâche. Du moment où il respecte ce qui a été défini et prend en compte l'ensemble du projet, il peut s'organiser et gérer sa tâche comme bon lui semble.

Avec cette organisation, nous ne voulons pas imposer une gestion de travail à des personnes qui ont déjà des habitudes, mais nous leur fournissons une vision de l'ensemble de l'organisation de la communauté et du management de l'ensemble du projet de R&D dans laquelle s'inscrit sa tâche.

3. répartitions des responsabilités

Pour éviter que la prise de décisions soit dédiée à un groupe de personnes limité (dirigeants), ce qui pourrait créer un «goulot d'étranglement », nous nous orienterons vers une distribution des responsabilités face aux tâches à réaliser pour le projet Porphyre. Ainsi, une grande partie du travail de centralisation n'aura plus lieu, et sera distribuée par les différentes tâches entre les Membres de la communauté, tandis que les Modérateurs se concentreront à l'animation de la communauté et à la validation des tâches de Recherche et de Développement, et des itérations du modèle de Recherche et du prototype informatique.

4. objectifs

Nous recherchons entre autre à travers cette organisation par tâches :

- l'apport de problématiques, d'idées et de solutions nouvelles lors de la définition des tâches par l'ensemble des Membres de la communauté.
- des interfaces et une définition des tâches clairement établies au sein du projet.
- une mobilisation de plusieurs membres, suivant leur disponibilité, pour la réalisation d'une ou plusieurs tâches.
- une entraide au sein de la communauté lorsque la quantité et la qualité de travail sont importantes sur une tâche ou un ensemble de tâches.
- un dégagement rapide des axes de R&D, grâce entre autre aux priorités des tâches.

IV. Outils et Méthodes

Dans une optique d'homogénéisation, aussi bien dans le travail que dans le formatage des données, et de synergie autour du projet Porphyre, il faut proposer un ensemble d'outils et de méthodes communs à l'ensemble de la communauté.

Dans l'intérêt d'accrocher un large public à l'investissement au projet Porphyre, il est préférable, de proposer un ensemble d'outils « gratuits ». Par contre, étant donné le caractère (fortement) évolutif des versions comme des droits d'utilisation (licences) des logiciels, nous ne faisons que proposer un certain nombre d'outils qui pourra évoluer par la suite.

Par contre, dans l'optique d'une utilisation efficace des outils et méthodes de travail choisis, une sensibilisation à la méthodologie de la communauté sera à mettre en place.

1. la modélisation objet : UML

UML est le langage de modélisation informatique la plus intuitive et évolutive (en terme de modélisation Orientée Objet). Il 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.

L'atelier UML choisi depuis quelques années est Objecteering Edition Personnel (version gratuite). Il permet d'exprimer tous les diagrammes UML pour la modélisation, la génération de code Java et de documentation

(RTF, HTML, PS), et permet aussi d'importer des projets sous forme standard XML. De plus, il est utilisable avec les systèmes d'exploitation les plus courants.

2. la gestion de versions : CVS

La 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.

Cette méthode de gestion de versions, que nous continuerons à utiliser, présente un intérêt pour les Membres de la communauté puisqu'ils pourront suivre plus facilement la succession et l'évolution du projet, des axes de recherche et des versions stables du prototype. Ils pourront également revenir à une version antérieure, si toutefois la version en cours n'aboutissait pas.

La communauté adoptera Concurrent Versions System comme outil de Gestion de Version, car il est le seul à offrir un panel de fonctionnalités assez exhaustif et à opter pour une licence de Logiciel Libre GNU GPL. Par défaut, CVS fonctionne en mode « ligne de commandes » (sans interface graphique), mais il existe un ensemble de clients graphiques gratuits, dont :

- Cervisia (pour Unix).
- WinCvs ou Turtoise (pour Windows).
- MacCvs (pour Macintosh).

3. la gestion des ressources

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, GNU GPL). Dans les deux cas, ces logiciels sont trop complets ou trop contraignants pour un projet aussi spécifique que Porphyre.

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 des fichiers tableau ou texte, ou avec une petite base de données favorisant le lien avec le site officiel www.porphyry.org.

Au minimum, ces fichiers seront une liste des tâches du projet avec toutes les informations utiles, un annuaire des membres de la communauté et un calendrier commun à la communauté.

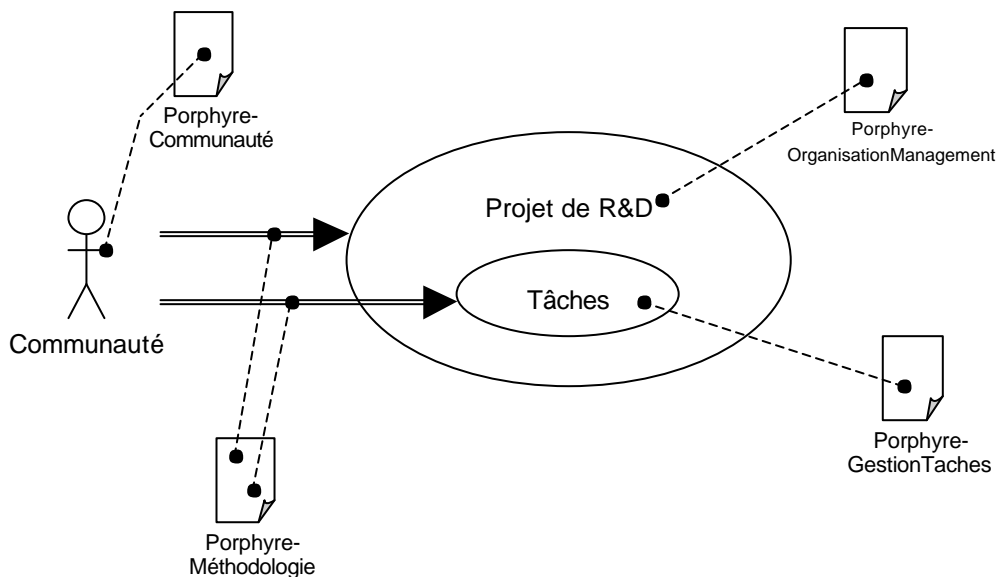


Figure 4 : Vue d'ensemble de l'application des documents de références

4. le développement

Le développement du projet est actuellement réalisé en Java. Le prototype représente déjà près de 18000 lignes de codes. La programmation se poursuivra donc en JAVA. Les avantages de cette technologie sont nombreux. À commencer par le fait que l'environnement de compilation et d'exécution est gratuit, fiable et portable (multi-systèmes d'exploitation). En ce qui concerne les éditeurs de codes JAVA, plusieurs sont disponibles gratuitement, comme entre autre « Xemacs » et « Kate KDE ». Pour la compilation, si les éditeurs de codes n'incluent pas la fonction de compilation de projet, 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 « makefile ».

5. la non-régression

Dans ce projet interdisciplinaire, 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 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. Cette méthode de tests de non-régression assurera un avancement constant du projet. Elle sera essentielle pour proposer des bases solides d'avancement dans la recherche et le développement du prototype.

6. la rédaction

Le choix des outils de rédaction de documents sera laissé au principal intéressé : le rédacteur. Le plus souvent, il s'agira des outils

de bureautique les plus répandus : Microsoft Word (Windows & Mac), Star Office (Linux). Une fois le document finalisé et validé, le fichier apparaîtra également sous un format permettant une consultation gratuite : PDF et/ou PostScript.

7. la 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 [LEVA 02]. L'outil de travail collaboratif le plus usuel est sans aucun doute l'Email. Donc, tous les Membres posséderont 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. Ensuite, le « chat » est un outil bien apprécié pour permettre à plusieurs personnes de discuter (en mode texte) en direct dans un espace virtuel. Mais l'utilité d'ouvrir un canal IRC propre à la communauté pour communiquer entre les Membres semble peu intéressante face aux divers outils existants déjà sur Internet et dont les fonctionnalités supplémentaires sont déjà assez exhaustives. Enfin, le portail du site web offrira un autre moyen de communication : unidirectionnel dans le cas de mise en ligne de publication de recherches ou de résultats de développement, multi-directionnel dans le cas des messages postés sur le serveur de News (Frequently Asked Questions, par exemple).

V. Etat des travaux et perspectives

Un ensemble de quatre 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 4). Mais il est bien entendu et même indispensable que ce système Porphyre évolue pour répondre et coller au mieux aux nouveaux besoins du projet et aux attentes des personnes appartenant à la communauté.

Néanmoins, ce système assure :

- des références communes (méthodes, outils, données) valables pour l'ensemble du projet de R&D.
- une compréhension commune de l'environnement du projet de R&D et de ses objectifs.
- la participation, l'implication et la responsabilisation via la délégation et la confiance.
- le partage et la circulation des informations au sein de la communauté et vers l'extérieur via son site web.
- la planification par la priorité des tâches.
- le suivi des tâches par la communauté.
- l'intégration et la validation de chaque tâche permettant l'évolution du projet de façon non-régressive.
- la valorisation des compétences et expériences de la communauté interdisciplinaire.
- un retour rapide et efficace de la part des utilisateurs membre de la communauté.

Par contre, ce système comporte encore quelques risques dont :

- l'intégration de la personne dans la communauté qui peut échouer, due à une « non-habitude » face aux outils et à la méthodologie de travail.
- les conflits entre des personnes de la communauté qui n'ont pas les mêmes points de vue ou les mêmes idées.
- la définition et le choix de tests de non-régression et de versions stables.

Un point sera à surveiller : un système « d'autogestion » est, par expérience [BARN 03], voué à l'échec. Il faudra donc, de la part des Modérateurs, conserver une animation et coordination de la communauté sur le long terme et sans relâche.

Cela peut être en partie, ou en totalité, résolu par le maintien, par les Modérateurs, d'une dynamique du projet et d'une animation perpétuelle de la communauté. Les Membres de la communauté doivent former un groupe d'individus soudés, une équipe de travail volontaire et disponible.

En bref, il faut construire une communauté vivante qui soit en premier lieu le reflet des personnes y appartenant, puis du projet.

Références bibliographiques

- [BARN 03] BARNI M. Manager une équipe à distance, Editions d'Organisation, février 2003.
- [KRUC 00] KRUCHTEN P. Introduction au Rational Unified Process, 2000.
- [LEVA 02] LEVANS. Les logiques et les règles d'usage des outils de travail collaboratif, article-interview JDNet, 2002
- [MOIN 03] MOINE A., DAVID F., FOULETIER B. Synthèse bibliographique : Agile Methods & Extreme Programming, février 2003.