



PROJET WIDE
Rapport d'analyse
Octobre - décembre 2013

Philippe ESTIVAL

philippe.estival@mri.cnrs.fr
MRI CNRS BIOCAMPUS UMS3426

Wide est une plateforme de sauvegarde, de partage et d'analyse de données biologiques. Ce rapport expose les résultats de l'examen logiciel du projet pour envisager son avenir sous des angles stratégiques et techniques. Le contexte et l'architecture sont présentés en première partie. La seconde vise à éclairer sur les bonnes pratiques à mener en conduites de projets informatiques d'envergures. Les résultats de l'audit sont à la fin. Vous pouvez commencer par la seconde partie si vous avez déjà eu connaissance du contexte.

1ère partie

MRI

Montpellier RIO Imaging est la plate-forme régionale d'imagerie du Languedoc-Roussillon. Labellisée IBISA (Infrastructures pour la Biologie, la Santé et l'Agronomie), elle permet la coordination des plate-formes de recherche en Science du Vivant ; RIO pour Réunion Inter-Opération. Lorsqu'une équipe de recherche a besoin d'un matériel, elle exprime la demande auprès du CNRS pour obtenir les fonds nécessaires à l'achat. Si la demande est exigible, le matériel est placé sous la responsabilité du MRI et rendu disponible pour d'autres équipes. La plate-forme est ouverte à l'ensemble des personnels de la production scientifique, du secteur publique au secteur privé sans limite géographique ni institutionnelle ou thématique. En 2009, MRI fournit 70900 heures de temps machine, forme les utilisateurs et partage son expertise. Ses activités de développements se concentrent sur l'imagerie des objets biologiques, l'imagerie du petit animal, l'électrophysiologie, la photométrie, l'imagerie à très haute résolution spatiale.

Présentation du projet

Wide – anciennement Cicéro – est un projet de plate-forme de services internet centralisée conçu pour la sauvegarde, le partage, l'analyse et la publication de données biologiques acquises par instruments de microscopie. Sur les sites pilotes où il a été déployé, des programmes de comptages et de facturations visent à améliorer l'assurance de mutualisation et de bonne gestion du matériel.

Le projet Wide a été initié en 2008.

Wide est inscrit dans le nœud Bio Image Informatics de l'IPDM et bénéficie à ce titre du financement de France Bio Imaging (FBI). Notons que sur le forum de FBI, une proposition de rapprochement avec les groupes de travail du réseau RBDD (Réseau Base de Donnée) du CNRS est soulevée, autour de questions d'interopérabilités et d'aspects juridiques.

La documentation indique que des étudiants stagiaires ont contribué à différentes étapes du développement de Wide. A partir de 2010, un développeur à temps plein a été affecté au projet.

Pourquoi centraliser les données ?

Le résultat de l'instrumentation produit des données à sauvegarder en lieu sûr. La pratique des utilisateurs se tourne naturellement vers l'usage de clés USB, fortement déconseillé, puis vers celui d'un serveur FTP, abondamment sollicité. Toutefois ceci comporte plusieurs risques, le pire étant la perte ou la compromission de ces données. En l'absence de centralisation :

- Traiter et analyser à posteriori les données acquises nécessite de les transférer sur un poste de travail, de réaliser des copies et des sauvegardes locales avant de téléverser à nouveau les résultats.
- Ni les versions successives ni l'authenticité originale de l'image ne sont assurés.
- Ces grands volumes de transferts provoquent la saturation des réseaux.
- Le nommage et l'identification des fichiers restent sous la responsabilité des utilisateurs et ne conserve pas d'empreinte.
- Aucun mécanisme ne protège d'une fausse manipulation qui entraînerait la suppression de toutes les données.

Pour ces raisons, un système centralisé (et redondant) garantit plus de sécurité et offrira graduellement les services et les fonctions nécessaires à l'amélioration du travail des ingénieurs et des chercheurs.

A cet effet, Wide propose un gestionnaire de fichiers accessible depuis le web pour organiser les données, visualiser les images, afficher puis traiter les méta-données extraites depuis les formats exotiques des constructeurs, interprète et unifie ces formats, pour enfin permettre d'analyser à distance de très grandes images.

Architecture

L'architecture proposée est résumée dans le schéma [1]. Y sont inscrits la majorité des développements déjà menés mais aussi des perspectives rationnelles pour anticiper le futur, ainsi que des propositions émises à ce jour. Avant de développer un de ces composants, il devra être détaillé dans un document qui expliquera la fonction qu'il doit remplir, les informations retournées, les dépendances et les moyens pour le déployer.

Les composants élémentaires pour obtenir une unité fonctionnelle de l'application sont :

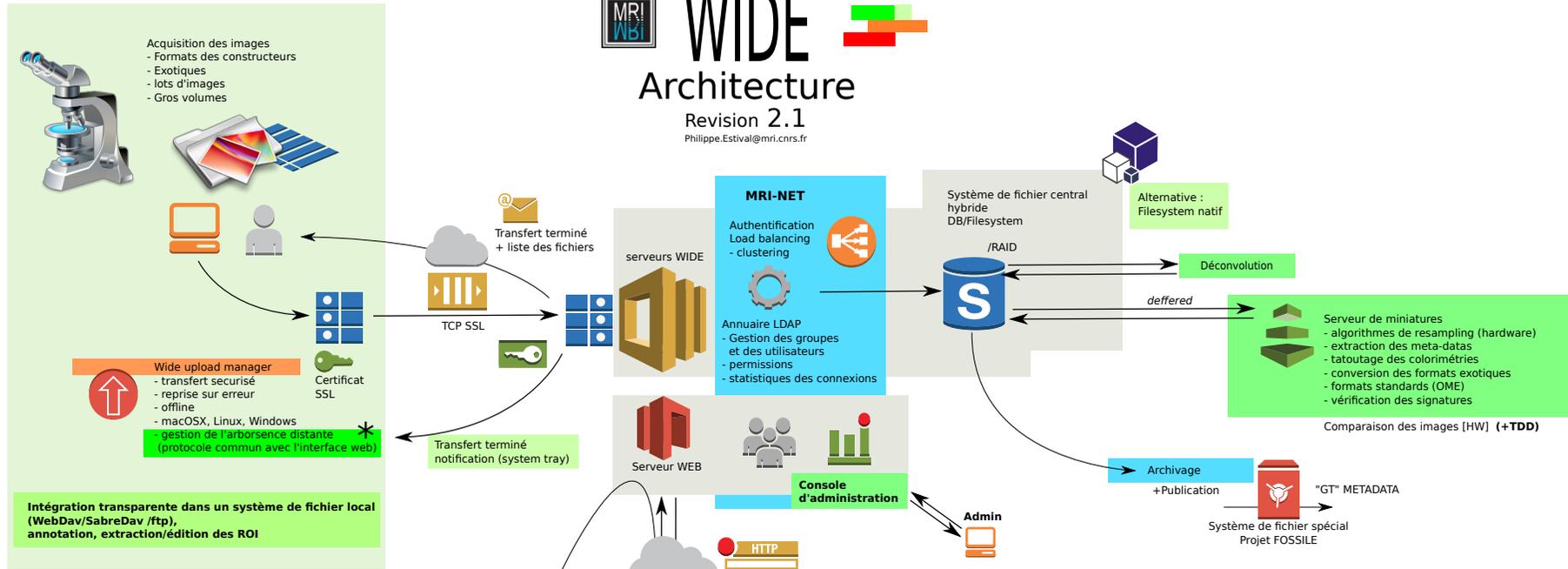
1. un serveur web doublé d'un serveur de fichiers et d'accès au dépôt.
2. un client de téléversement asynchrone pour le dépôt des données, qui poursuit le téléversement après qu'une session de l'utilisateur soit déconnectée.
3. Une extraction des méta-données associées ou décrites dans les données.
4. un gestionnaire (web-explorateur) de fichiers propre à chaque utilisateur pour l'organisation des fichiers, le téléchargement et la visualisation des images et de leurs méta-données associées.
5. L'authentification – périmétrique pour les connexions réalisées depuis l'intranet MRI – sécurisée par un échange de certificat SSL pour crypter les échanges.
6. Un programme de visualisation et de traitement à distance des images.
7. Des bibliothèques standards pour l'interprétation des données de microscopie [BioFormat, LOCI, OME data model]. Ce standard évolue en direction de SCIFIO, pour réorganiser et généraliser Bio-Format au-delà des sciences du vivant. L'appellation BioFormat elle-même décourage les ingénieurs venus d'un autre domaine, tandis que les problématiques d'imagerie ou de microscopie sont similaires, par exemple en rétroconception de microprocesseurs ¹. Les équipes impliquées dans OMERO sont celles qui définissent et implémentent ce standard.

Pareille architecture s'articule autour d'une console d'administration, pour gérer et connaître l'état des connexions et des espaces occupés, offrir une vision claire et synthétique de l'état du système, effectuer les opérations de maintenance. Elle semble avoir été oubliée dans les spécifications de Wide.

1. Echelle de 22nm, 72 millions de transistors par cm² [BLOCH].

WIDE Architecture

Revision 2.1
Philippe.Estival@mri.cnrs.fr



Fédération des composants

Définition des protocoles de communication
- évolutions **dissociées** des serveurs, du client lourd, et de l'interface web

La gestions des fichiers / des groupes / utilisateurs et systèmes de fichiers est une **problématique générique**
Peut appeler support communautaire très large
= solution opensource
= obtenir et participer au support du projet

Logiciel d'analyse
= **problématique commune**
= se concentrer sur les données et les metadatas,
= questions spécifiques au laboratoire et au traitement d'image

Questions :
Volumétrie attendue
- quantité de données
- nombre d'utilisateurs (700max)
- types/formats usuels des données
- où est la documentation du projet ?
- pas de console d'admin ?

Développement :
Test Driven Développement (TDD)
Serveur d'intégration
Tests fonctionnels

FIGURE 1 – Wide : architecture

2ème partie : le mythe du mois-homme

*« Faire de la bonne cuisine demande un certain temps.
Si on vous fait attendre, c'est pour mieux vous servir, et vous plaire. »*

Menu du restaurant Antoine, Nouvelle Orléans

Frédéric Brooks² était l'ingénieur en chef du projet OS/360 d'IBM qui visait à apporter des corrections majeures aux processeurs de la génération précédente. Persuadé de la viabilité du projet, la direction d'IBM y avait investi des sommes colossales et embauché une équipe d'ingénieurs malgré qu'il comportait un vice de conception qu'il fût tout simplement impossible d'extirper sans tout recommencer et ceci malgré les avertissements de Brooks.

Lorsque vous êtes responsable d'un projet à 7M\$, surtout à l'époque, vous comprenez aisément le besoin d'exprimer un mea-culpa quand il se termine ainsi. Il est l'auteur d'un livre appelé « *The Mythical Man-Month* » [BROOKS95], qui fait référence à une unité de coût de développement : le mois-homme ou plus communément homme-mois, c'est à dire le travail d'un homme pendant un mois. Le seul fait d'utiliser cette unité tend à faire croire que le travail de 1 personnes pendant n mois peut parfaitement être réalisé par n personnes pendant 1 mois. Selon cette idée on pourrait diviser le temps de travail par deux en mettant plus de personnel. Or expérimentalement, cela est faux. Le proverbe cité par Brooks pour exprimer cette idée est : « Neuf femmes ne font pas un enfant en un mois ».

Bien qu'il mentionne parfois des outils et des problèmes algorithmiques, il s'attache plutôt à l'organisation et aux méthodes de travail des équipes de développement : hiérarchie des responsables, documentation et prévision des délais de développement.

La loi de Brooks

Ajouter des ressources humaines à un projet en retard ne fait qu'accentuer ce retard : c'est la loi de Brooks. En effet, le personnel ajouté devra être formé au nouveau système, ce qui prend un temps non négligeable que ne peut compenser la productivité ajoutée par le personnel en question. Les nouveaux ont souvent besoin d'entraînement et de formation, et le travail d'être contrôlé et suivi. De plus, leur présence alourdit les canaux de communication.

L'effet « deuxième version »

Tout informaticien ayant réalisé la version originale d'un logiciel a tendance à incorporer dans la deuxième version toutes les fonctions qu'il n'avait pu intégrer, faute de temps, dans la première. Or cette initiative, apparemment heureuse, comporte de réels dangers :

- bouleverser l'aspect du logiciel auquel les utilisateurs avaient fini par s'habituer
- automatiser des séquences de tâches que l'utilisateur effectuait volontiers lui-même dans la première version, ce qui alourdit l'interface
- promouvoir des fonctionnalités déjà obsolètes
- faire exploser la taille du logiciel
- et surtout perdre l'unité conceptuelle (les spécifications d'origine) qui régissait la première version, en cherchant à imiter la concurrence (*par exemple*, un visualiseur d'image est un tout autre projet qu'un système de sauvegarde et de partage centralisé).

Un informaticien développant une deuxième version devrait donc être conscient du besoin de stabilité de l'utilisateur moyen et du risque de dépassement des spécifications exigées.

2. Médaille John Von Neumann en 1993 et prix Turing 1999 pour ses contributions en « architecture des ordinateurs, système d'exploitation et logiciel »



Le suivi de projet

Comment un gros projet peut-il être en retard d'un an sur l'horaire prévu ? A cause du retard pris chaque jour de ce projet. Des retards à différentes étapes ont un effet cumulatif et retardent globalement le projet. Pour prévenir ce retard, il faut que les gestionnaires agissent pour faire respecter l'horaire de livraison de chacune des étapes du projet...

L'unité conceptuelle

Tout système convivial possède nécessairement une unité conceptuelle laquelle est possible si l'architecture logicielle est séparée de son implémentation. Une « super idée » peut-être rejetée si elle ne s'insère pas naturellement dans le programme. Pour assurer la convivialité, un architecte logiciel peut décider que le programme offrira moins de fonctions que ce qu'il peut offrir. La raison est simple : si une fonction est trop longue à comprendre, elle ne sera pas utilisée.

Le prototype

Lorsqu'elle conçoit un nouveau type de système, une équipe va forcément concevoir un système bon à jeter, de manière intentionnelle ou non. Un tel système a en fait le rôle d'un prototype permettant de faire émerger des techniques qui causeront par la suite une complète refonte du système. C'est ce second système, plus « intelligent », qui devrait être celui qui sera livré au client, puisque livrer la version prototype ne causerait rien d'autre que l'agonie du client, ruinant par la même la réputation du programme, voire peut-être celle de l'entreprise.

Les documents de base

Tout gestionnaire de projet devrait rédiger des documents de base servant à établir le plan de travail : quels sont les objectifs, comment les atteindre, qui est responsable de quoi, quand doivent-ils être atteints et combien chacun d'eux coûtera ? Ces documents peuvent révéler des incohérences difficilement visibles autrement.

La planification

Lorsqu'un architecte logiciel planifie un projet, il doit se souvenir que l'usage d'une méthode d'intégration et d'un langage de haut niveau adapté peut substantiellement améliorer la productivité d'un informaticien. Il doit également tenir compte du temps pris par les tâches qui ne sont pas de nature techniques (réunions, rapports, congés, etc.).

La communication

Dans le but de prévenir les désastres, toutes les équipes affectées à un projet doivent maintenir le contact entre elles à l'aide du plus grand nombre de moyens possibles (courriel, téléphone, réunions, mémos, etc.). Plutôt que de faire des hypothèses sur une fonction, l'informaticien se réfère à l'architecte logiciel pour clarifier la nature de la fonction. Sinon il risque d'implémenter une fonction qui sera rejetée.

Le « chirurgien en chef »

Toute équipe de chirurgiens est menée par un chirurgien en chef, qui s'occupe en personne des tâches essentielles de l'opération tout en dirigeant son équipe, qui l'appuie et réalise des tâches moins essentielles. Il est raisonnable de voir une équipe logicielle de la même façon : le meilleur informaticien s'occupe des parties essentielles du composant à créer, alors que les autres s'occupent des autres tâches. Brooks spécule que les meilleurs informaticiens ont une production 5 à 10 fois supérieure aux pires.

La gestion de versions et le gel du code

Tout changement au code source doit être connu via un système de gestion de versions. En revanche, le logiciel est invisible. Pour cette raison, plusieurs fonctions ne deviennent visibles qu'après un certain temps de développement. A ce moment, les utilisateurs peuvent commencer à utiliser une partie du programme final. Leur expérience apporte des points de vue nouveaux, qui changent la perception de l'utilisateur ou la perception des besoins. A ce moment, le système doit aussi changer pour combler les besoins. Ces modifications sont permises à l'intérieur d'une certaine fenêtre de temps, sinon le produit final change continuellement, en attente d'être livré aux clients. A une certaine date, seuls les bogues doivent être réparés : le code source est gelé. Tous les changements conceptuels en attente d'implémentation se retrouveront dans une version future.

La diminution des coûts de développement logiciel

Selon Brooks, il existe deux techniques pour diminuer les coûts de développement :

- engager les informaticiens qui implémentent le système une fois les spécifications complètement établies par les architectes (une étapes qui prend du temps, période durant laquelle les informaticiens employés trop tôt n'ont rien à faire) ;
- acheter des logiciels déjà fonctionnels (« sur l'étagère »), tout simplement.

Résultats de l'audit

« La menace qui pèse sur les informaticiens, c'est le conservatisme, encouragé par les fournisseurs, on comprend qu'ils n'aient pas envie de voir remises en cause des compétences qui ont mises des années à être acquises, mais dans un secteur en évolution rapide, le conservatisme est suicidaire à terme, même si la croissance rapide du secteur peut en masquer les dangers. »

Laurent Bloch

*Chercheurs et ingénieurs en informatique.
Nécessaire (mais difficile) alliance.*

Le code source et la documentation de Wide sont librement disponible. ³

L'architecture, la conception et le codage de Wide ont des erreurs ; c'est courant en informatique et le reconnaître aide à la préparation d'un système plus robuste et mieux maîtrisé. En ceci, Wide est un bon prototype, mûr pour concevoir le système final.

Voici plusieurs remarques résultantes de l'analyse :

◊ Il s'agit d'un système de fichier virtuel construit à l'intérieur du système de fichier natif, un hybride base de données / système de fichier. Si l'un ou l'autre sont compromis ou qu'un accès à l'un ou l'autre des deux trousseaux est manquant, les données perdent leur sens. Or les expériences, les analyses et les interprétations en biologies sont longues et impliquent des recherches sur du long terme, parfois toute une vie. Qui peut garantir de la sûreté de ce système pendant 20 ans ? La préconisation est de maintenir la cohésion des données.

◊ L'environnement de développement (Eclipse Kepler) et les deux serveurs d'applications (GlassFish3.1) consomment beaucoup trop de ressources, ralentissant considérablement le travail des développeurs : 4Go de RAM sont insuffisants pour faire fonctionner l'ensemble en raison d'une *fuite de mémoire* dans le serveur d'EJB. Problème assez grave, quoique surmontable. Son origine reste à déterminer.

◊ En principe ce genre de programme sollicite des processus légers. Ce n'est que sous la charge d'un grand nombre de connexions et d'opérations simultanées que ces ressources peuvent être réquisitionnées.

◊ La structure de dépôts et de sauvegardes des codes sources (SVN) est demeurée une branche unique entre chaque versions produites. Or les bonnes pratiques recommandent aussi – la création de branches de développement pour garder intact des codes stabilisés et déployés entre chaque gel de version – mais aussi, et surtout, une version en cours de développement pour les correctifs apportés, et ceci à plus forte raisons si un nouveau codeur intervient dans la rédaction des sources. Comme la périodicité quotidienne du dépôt de sources ne permet de remonter que de deux semaines en arrière, la dernière version peut entrer dans un état dégénéré même du fait de très petites modifications. La création d'une branche SVN est donc nécessaire entre les *releases* si la version mineure (qui indique les *bugfixs*) progresse rapidement (ainsi la 1.3.16-dev mérite de produire auparavant une branche 1.3.8 par exemple).

◊ Numéroter automatiquement les versions pour un meilleur suivi des tickets et des correctifs.

◊ Le contrôle des régressions impose la mise en place d'un serveur d'intégration.

Wide est écrit en Java, XHTML et Javascript. Un modèle strict côté serveur respecte la convention Modèle-Vue-Contrôler (MVC). Il repose sur un serveur Glassfish v3 fourni par Oracle.

3. <http://dev.mri.cnrs.fr>

Recommandations

S'il fallait continuer avec les technologies Java, voici les points qui mériteraient d'être réexaminés.

Librairies

Bien qu'il soit mature et très largement adopté et pourvu d'une intégration avancée côté serveur, **Richfaces** est un choix à réévaluer. La documentation est brève et générique. La librairie, maintenue par une équipe réduite, semble à la traîne face aux solutions concurrentes. Une préconisation des experts du réseau RBDD2013 est de se tourner vers **GWT** pour de futurs développements d'applicatifs Java déployés sur le web.

Sécurité

La configuration utilisant les *ViewStates* et incluant les librairies aux versions suivantes : JSF ; Mojarra < v2.1.23 ; PrimeFaces < v3.5 ; Tomcat 7.0.42 (dernière version disponible) est **vulnérable** [MISC69]. La console de supervision de Glassfish porte elle aussi une faille importante connue de l'administration de MRI-TIGR.

Backup/Restore

Un moyen simple de sauvegarder régulièrement les bases et les fichiers via une console d'administration est impératif.

Serveur

Il peut être plus judicieux d'examiner des solutions alternatives au serveur Glassfish et se tourner vers la fondation Jakarta pour y trouver un serveur d'EJB plus rapide et plus léger. Encore une fois, un avis d'expert est nécessaire, la question pouvant faire l'objet d'un débat sur un forum spécialisé. Embarquer l'interface web sur **Tomcat** a permis d'obtenir un petit gain de performances.

Clustering

Le *clustering* du serveur, c'est à dire sa répartition sur une grappe pour répondre à de fortes sollicitations, est réputé plus simple avec Glassfish. Les avis diffèrent pour savoir s'il s'agit d'un avantage ou non. C'est un réglage fin du système dans lequel la complexité doit rester visible au yeux de l'administrateur. Une solution "magique" qui la masque en automatisant cette tâche va forcément masquer aussi les réglages relatifs aux mouvements de la mémoire, à la gestion des caches et aux stratégies d'optimisation de bas niveau. Les questions de charges et de répartitions sont très largement traitées dans la littérature des réseaux informatiques et méritent qu'un peu de temps leur soient consacrées.

Corrections apportées

Les tableaux suivants récapitulent les bogues corrigés et découverts pendant les tests et les fonctionnalités prévues. Tous les composants passés au vert ont été réalisés par nos soins.

La version « de production » est demeurée intacte et c'est sur cette dernière (1.3.1) que j'ai verrouillé un accès.⁴

4. Le seul remède est de modifier manuellement la base de données.

✓ #	Projet	Tracker	Statut	Priorité	Sujet	Assigné à	Mis-à-jour	Début	Echéance	Temps estimé	% réalisé
<input type="checkbox"/> 802	WIDE	Bug	New	Normal	strange bug with drag and drop file	Philippe Estival	4/11/2013 19:30	1/10/2013			<div style="width: 100%;"></div>
<input type="checkbox"/> 801	WIDE	Bug	New	High	Multiple "download as zip" leads to Denial of Service.	Granier Alexandre	1/10/2013 14:44	1/10/2013			<div style="width: 0%;"></div>
<input type="checkbox"/> 800	WIDE	Bug	New	Normal	rename popup	Philippe Estival	25/11/2013 18:14	1/10/2013			<div style="width: 100%;"></div>
<input type="checkbox"/> 799	WIDE	Bug	New	Normal	page style problem in central form	Granier Alexandre	1/10/2013 14:55	1/10/2013			<div style="width: 0%;"></div>
<input type="checkbox"/> 798	WIDE	Bug	New	Normal	Impossible to correctly drag an drop an image in thumbnail mode	Philippe Estival	4/11/2013 19:36	1/10/2013			<div style="width: 100%;"></div>
<input type="checkbox"/> 797	WIDE	Bug	New	Normal	no mail with authenticated client	Granier Alexandre	1/10/2013 14:30	1/10/2013			<div style="width: 0%;"></div>
<input type="checkbox"/> 796	WIDE	Bug	Resolved	Low	The wait "Loading" logo is incorrect	Granier Alexandre	8/10/2013 11:45	1/10/2013			<div style="width: 100%;"></div>
<input type="checkbox"/> 795	WIDE	Bug	New	Low	Incorrect modification date for folders	Philippe Estival	25/11/2013 18:52	1/10/2013		2.0	<div style="width: 100%;"></div>
<input type="checkbox"/> 730	WIDE	Bug	Assigned	Normal	Web site : timeout of cookie does not lead to login page.	Granier Alexandre	5/07/2013 10:52	5/07/2013			<div style="width: 0%;"></div>
<input type="checkbox"/> 606	WIDE	Bug	New	Normal	High CPU charge when opening several Viewers	Granier Alexandre	28/05/2013 17:08	28/05/2013			<div style="width: 0%;"></div>
<input type="checkbox"/> 605	WIDE	Bug	New	Normal	Very hard to select a file or a folder in the thumbnail mode	Granier Alexandre	14/11/2013 13:50	28/05/2013			<div style="width: 100%;"></div>
<input type="checkbox"/> 604	WIDE	Bug	New	Low	Can't apply differents groups rights to subdirs.	Granier Alexandre	28/05/2013 16:45	28/05/2013			<div style="width: 0%;"></div>
<input type="checkbox"/> 603	WIDE	Bug	New	Normal	High CPU charge when applying rights.	Granier Alexandre	28/05/2013 16:43	28/05/2013			<div style="width: 0%;"></div>
<input type="checkbox"/> 602	WIDE	Bug	New	Low	Remaining upload size is not updated in the upload client	Granier Alexandre	28/05/2013 16:38	28/05/2013			<div style="width: 0%;"></div>
<input type="checkbox"/> 601	WIDE	Bug	New	Low	Upload emails in the wild.	Granier Alexandre	8/10/2013 16:45	28/05/2013			<div style="width: 0%;"></div>
<input type="checkbox"/> 600	WIDE	Bug	New	Low	Zip files everywhere!	Granier Alexandre	28/05/2013 16:32	28/05/2013			<div style="width: 0%;"></div>
<input type="checkbox"/> 599	WIDE	Bug	New	Immediate	Client now needs loci	Granier Alexandre	28/05/2013 11:12	28/05/2013			<div style="width: 0%;"></div>
<input type="checkbox"/> 568	WIDE	Bug	New	Low	Cicero is always here.	Granier Alexandre	12/04/2013 11:32	12/04/2013			<div style="width: 0%;"></div>
<input type="checkbox"/> 566	WIDE	Bug	Assigned	Normal	sharing images	Granier Alexandre	12/04/2013 11:16	12/04/2013			<div style="width: 0%;"></div>
<input type="checkbox"/> 564	WIDE	Bug	New	Normal	select remote folder on wide client don't work	Granier Alexandre	12/04/2013 11:30	12/04/2013			<div style="width: 0%;"></div>
<input type="checkbox"/> 450	WIDE	Bug	New	Normal	open with imagej don't work		8/10/2012 15:38	8/10/2012			<div style="width: 0%;"></div>
<input type="checkbox"/> 387	WIDE	Feature	New	Normal	add plugin entries into menus	Volker Baecker	14/06/2012 13:45	14/06/2012	18/06/2012		<div style="width: 0%;"></div>
<input type="checkbox"/> 380	WIDE	Bug	New	Normal	Multiple upload of same folder leads to corrupted folders	Granier Alexandre	7/06/2012 17:51	7/06/2012			<div style="width: 0%;"></div>
<input type="checkbox"/> 372	WIDE	Bug	New	Normal	ie 9 : we can't focus folder in thumnnail view.	Granier Alexandre	4/06/2012 17:36	4/06/2012			<div style="width: 0%;"></div>
<input type="checkbox"/> 370	WIDE	Bug	New	Normal	Empty folder are removed from zip files	Granier Alexandre	4/06/2012 15:56	4/06/2012			<div style="width: 0%;"></div>

FIGURE 2 – Wide v1.3.1 : correctifs apportés

✓ #	Projet	Tracker	Statut	Priorité	Sujet	Assigné à	Mis-à-jour	Début	Echéance	Temps estimé	% réalisé
<input type="checkbox"/> 369	WIDE	Bug	New	Normal	ie8 : select all wont work	Granier Alexandre	4/06/2012 15:53	4/06/2012			
<input type="checkbox"/> 368	WIDE	Bug	New	Normal	Zip progress stuck if it ie8 lost focus		4/06/2012 15:51	4/06/2012			
<input type="checkbox"/> 366	WIDE	Bug	New	Normal	No messages when a zip ends succesfully.	Granier Alexandre	4/06/2012 15:46	4/06/2012			
<input type="checkbox"/> 365	WIDE	Bug	New	Normal	If i close the web page during thz zip progress, i lost my session	Granier Alexandre	4/06/2012 15:44	4/06/2012			
<input type="checkbox"/> 353	WIDE	Feature	New	Normal	Add auto completion to choose users in wide web interface related widget	Granier Alexandre	20/04/2012 09:56	20/04/2012			
<input type="checkbox"/> 336	WIDE	Feature	Assigned	Normal	clean the zip/tmp folder	Granier Alexandre	22/03/2012 11:53	22/03/2012			
<input type="checkbox"/> 335	WIDE	Bug	Assigned	Normal	Upload Client : progress bar display bad value when uploading multiple file image	Granier Alexandre	22/03/2012 11:13	22/03/2012			
<input type="checkbox"/> 281	WIDE	Task	Assigned	Normal	requirement analysis deconvolution in wide	Volker Baecker	10/01/2012 10:41	10/01/2012	27/01/2012	80.0	
<input type="checkbox"/> 243	WIDE	Bug	New	Normal	An expired session lead to a 404 error page.		15/11/2011 10:40	15/11/2011			
<input type="checkbox"/> 228	WIDE	Refactoring	Feedback	Normal	[ErgCI] Ergonomy upload client	Granier Alexandre	15/03/2012 17:00	24/10/2011		56.0	<div style="width: 56%;"></div>
<input type="checkbox"/> 227	WIDE	Feature	New	Normal	[Hist] History function	Granier Alexandre	24/10/2011 11:35	24/10/2011		96.0	<div style="width: 96%;"></div>
<input type="checkbox"/> 226	WIDE	Bug	New	Normal	[UploadApp] Write an upload applet for wide	Granier Alexandre	24/10/2011 11:32	24/10/2011		96.0	<div style="width: 96%;"></div>
<input type="checkbox"/> 225	WIDE	Feature	New	Normal	[Deconv] Deconvolution	Granier Alexandre	24/10/2011 11:29	25/10/2011	25/10/2011	576.0	
<input type="checkbox"/> 224	WIDE	Support	Assigned	Normal	[Mig] Migration from glassfish 2 to 3 and richfaces 3 to 4	Philippe Estival	25/11/2013 17:56	17/10/2011	10/11/2011	80.0	<div style="width: 80%;"></div>
<input type="checkbox"/> 223	WIDE	Feature	Assigned	Normal	[Login] Allow login out of MRI network	Granier Alexandre	4/06/2012 14:35	24/10/2011		56.0	<div style="width: 56%;"></div>
<input type="checkbox"/> 221	WIDE	Feature	Assigned	Normal	[ErgWeb] Improve web interface ergonomy	Granier Alexandre	24/10/2011 11:21	24/10/2011		80.0	<div style="width: 80%;"></div>
<input type="checkbox"/> 220	WIDE	Bug	Assigned	Normal	[Upload] Permit upload from web browser	Granier Alexandre	24/10/2011 11:19	24/10/2011		40.0	<div style="width: 40%;"></div>
<input type="checkbox"/> 219	WIDE	Feature	Assigned	Normal	[MetaDec] Metadata management for deconvolution purposes	Granier Alexandre	24/10/2011 11:22	24/10/2011		96.0	<div style="width: 96%;"></div>
<input type="checkbox"/> 217	WIDE	Bug	New	Normal	Bug on display IE 7	Granier Alexandre	21/10/2011 14:21	21/10/2011			
<input type="checkbox"/> 205	WIDE	Bug	New	Normal	The remove menu item still here when select a file in the trash	Granier Alexandre	24/10/2011 11:43	21/09/2011		5.0	<div style="width: 5%;"></div>
<input type="checkbox"/> 94	WIDE	Feature	New	Normal	replace "upload manager" by an another protocol if necessary		30/05/2011 10:29	11/07/2011			
<input type="checkbox"/> 93	WIDE	Task	New	High	to update "tools site"	Olivier Miquel	11/10/2011 10:38	30/05/2011	10/06/2011		<div style="width: 100%;"></div>
<input type="checkbox"/> 87	WIDE	Task	New	Low	MAJ descriptif installeur sous Redmine	Granier Alexandre	10/10/2011 15:55	23/05/2011			<div style="width: 20%;"></div>
<input type="checkbox"/> 83	WIDE	Bug	New	Low	No thumbnails for high-res png files	Granier Alexandre	23/05/2011 11:39	20/05/2011			
<input type="checkbox"/> 81	WIDE	Bug	New	High	corrupted image if uploaddaemon crash during image upload	Granier Alexandre	23/05/2011 11:44	20/05/2011			

FIGURE 3 – Wide v1.3.1 : correctifs apportés





✓ #	Projet	Tracker	Statut	Priorité	Sujet	Assigné à	Mis-à-jour	Début	Echéance	Temps estimé	% réalisé
<input type="checkbox"/> 79	WIDE	Bug	New	Normal	no answer or a long response time with IE 8	Corine Tran-Aupiais	23/05/2011 11:46	16/05/2011			<input type="checkbox"/>
<input type="checkbox"/> 71	WIDE	Bug	New	Low	Upload client: stuck scanning when rights problem	Granier Alexandre	23/05/2011 11:57	28/03/2011			<input type="checkbox"/>
<input type="checkbox"/> 70	WIDE	Bug	Assigned	Low	strange remaining time for upload progress	Granier Alexandre	8/04/2013 13:30	28/03/2011			<input type="checkbox"/>
<input type="checkbox"/> 68	WIDE	Bug	New	Low	Gif thumbnail is bigger than original file.	Granier Alexandre	23/05/2011 12:01	25/03/2011			<input type="checkbox"/>
<input type="checkbox"/> 7	WIDE	Bug	Assigned	Normal	script and/or instructions to setup the project	Granier Alexandre	11/10/2011 16:43	27/07/2010			<input type="checkbox"/>

✓ #	Projet	Tracker	Statut	Priorité	Sujet	Assigné à	Mis-à-jour	Début	Echéance	Temps estimé	% réalisé
<input type="checkbox"/> 883	WIDE	Refactoring	New	Normal	The cancel all button is too near to the low right corner		27/11/2013 17:38	27/11/2013		1.0	<input type="checkbox"/>
<input type="checkbox"/> 882	WIDE	Bug	New	Urgent	The build confirmation still send mails to Alexandre	Philippe Estival	25/11/2013 18:55	25/11/2013			<input type="checkbox"/>
<input type="checkbox"/> 881	WIDE	Bug	New	Normal	Impossible to empty trash	Philippe Estival	25/11/2013 18:51	25/11/2013			<input type="checkbox"/>
<input type="checkbox"/> 880	WIDE	Bug	New	High	The viewer does not open	Philippe Estival	25/11/2013 18:47	25/11/2013			<input type="checkbox"/>
<input type="checkbox"/> 879	WIDE	Refactoring	New	Normal	XHTML conformancy	Philippe Estival	25/11/2013 18:36	25/11/2013		1.0	<input type="checkbox"/>
<input type="checkbox"/> 878	WIDE	Bug	New	Normal	Download manager is stalled	Philippe Estival	25/11/2013 17:51	25/11/2013			<input type="checkbox"/>
<input type="checkbox"/> 877	WIDE	Bug	New	Normal	Selecting a folder still displays metadata		14/11/2013 13:47	5/11/2013			<input type="checkbox"/>
<input type="checkbox"/> 875	WIDE	Feature	New	Normal	add a confirm dialog on context > delete item	Philippe Estival	4/11/2013 13:07	4/11/2013		1.0	<input type="checkbox"/>
<input type="checkbox"/> 874	WIDE	Setup	Feedback	Immediate	next Milestone goes JSF 2.1.21	Philippe Estival	4/11/2013 19:29	31/10/2013			<input type="checkbox"/>
<input type="checkbox"/> 873	WIDE	Bug	New	Urgent	File system inconsistency when renaming	Philippe Estival	25/11/2013 18:19	29/10/2013		3.0	<input type="checkbox"/>
<input type="checkbox"/> 871	WIDE	Bug	New	Normal	New folder appears at the end of a list	Philippe Estival	31/10/2013 16:14	29/10/2013		1.0	<input type="checkbox"/>
<input type="checkbox"/> 870	WIDE	Bug	New	Low	The enter Key doesn't work in modal window	Philippe Estival	31/10/2013 15:03	29/10/2013			<input type="checkbox"/>
<input type="checkbox"/> 869	WIDE	Bug	New	Low	Text under miniature becomes grey	Philippe Estival	25/11/2013 18:42	31/10/2013		1.0	<input type="checkbox"/>
<input type="checkbox"/> 855	WIDE	Bug	New	Normal	Synchronize the tree and the list	Philippe Estival	4/11/2013 19:29	25/10/2013		2.0	<input type="checkbox"/>
<input type="checkbox"/> 854	WIDE	Bug	New	High	No rule to prevent weird files and folders names		12/11/2013 10:38	25/10/2013		2.0	<input type="checkbox"/>
<input type="checkbox"/> 851	WIDE	Refactoring	New	Normal	Liste des utilisateurs en partage	Granier Alexandre	24/10/2013 11:47	20/11/2013		24.0	<input type="checkbox"/>
<input type="checkbox"/> 845	WIDE	Bug	New	Normal	Window opens a secret area upon drag n drop	Philippe Estival	25/11/2013 17:56	17/10/2013		1.0	<input type="checkbox"/>
<input type="checkbox"/> 820	WIDE	Bug	New	High	renaming folders in tree view		31/10/2013 16:13	10/10/2013		1.0	<input type="checkbox"/>
<input type="checkbox"/> 810	WIDE	Bug	Feedback	High	Make a public link won't work most of the time	Granier Alexandre	9/10/2013 17:06	1/10/2013			<input type="checkbox"/>
<input type="checkbox"/> 809	WIDE	Bug	Resolved	Low	Incorrect title when we moidy a public link	Granier Alexandre	9/10/2013 17:06	1/10/2013			<input type="checkbox"/>
<input type="checkbox"/> 807	WIDE	Bug	New	High	We can't share files	Granier Alexandre	1/10/2013 15:09	1/10/2013			<input type="checkbox"/>
<input type="checkbox"/> 806	WIDE	Bug	Resolved	Normal	Share duplicate folder	Granier Alexandre	11/10/2013 17:09	1/10/2013			<input type="checkbox"/>
<input type="checkbox"/> 805	WIDE	Bug	New	Normal	Impossible to download "Metadata and Tif"	Granier Alexandre	1/10/2013 15:04	1/10/2013			<input type="checkbox"/>
<input type="checkbox"/> 804	WIDE	Bug	New	Normal	Issue with duplicate file with same name	Philippe Estival	25/11/2013 18:05	1/10/2013			<input type="checkbox"/>
<input type="checkbox"/> 803	WIDE	Bug	New	Normal	Renaming in right pane doesn't affect left pane.	Philippe Estival	4/11/2013 19:30	1/10/2013			<input type="checkbox"/>

FIGURE 4 – Wide v1.3.1 : correctifs apportés

Serveur d'intégration

Un serveur d'intégration pour l'exécution automatisée de tests unitaires permettra de prévenir que des modifications apportées, mêmes si elles peuvent sembler justes, ne provoquent pas d'autres problèmes inattendus. Un mail doit être envoyé quotidiennement pour annoncer le résultat de la compilation du projet sur le serveur de test. La configuration de cette fonction se trouve dans un paquet gelé (MRI-enterprise) auquel il faudra demander accès auprès des administrateurs MRI : les mails continuent d'être émis à l'adresse du responsable précédent du projet.

Systématiser la procédure à chaque modification solidifiera la méthode de développement.

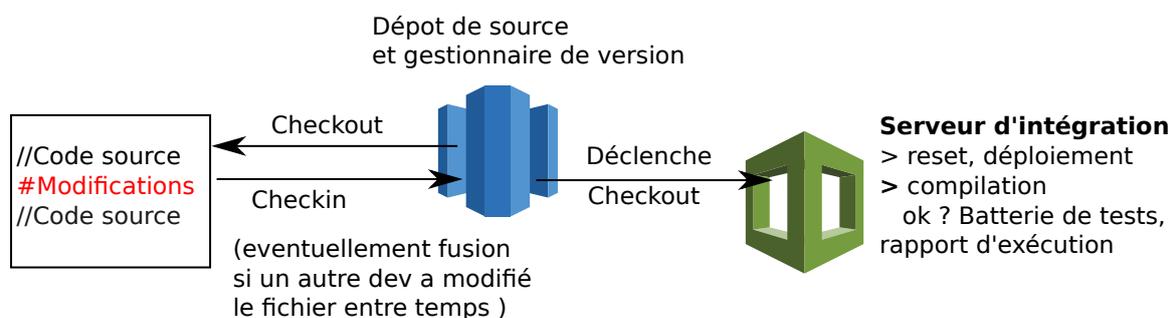


FIGURE 5 – Principe d'un serveur d'intégration

Si le rapport de compilation est positif, le serveur d'intégration réinitialise le déploiement complet de l'application puis procède d'abord aux tests unitaires, ensuite aux tests fonctionnels. On peut ensuite rédiger puis vérifier de nouveaux scénarios de tests. Parmi les solutions bâties pour répondre à ce problème, on peut citer **Cruise Control**, **Jenkins** et **TeamCity**.

Tests Fonctionnels

Compte tenu du rôle central joué par une application pourvue d'une interface graphique, l'absence de test fonctionnel automatisé (i.e. de l'interface graphique, au comportements simulés d'après des scénarios enregistrés) a aussi fait un vif débat au cours de nos réunions. Je maintiens qu'elle est indispensable : beaucoup de temps est perdu à tester des manipulations répétitives de dossiers et de fichiers. Le framework **Selenium** et sa suite, couplé avec un serveur d'intégration, servent précisément à surveiller les régressions.

Méthode

Pour maintenir ce projet en vie, nous invitons son futur responsable à consolider *les méthodes en priorité*. Pour cela, un changement dans la politique de gestion du projet est nécessaire. En l'état actuel, poursuivre les développements menés et fournir des délais d'estimation pour des corrections et de nouvelles fonctionnalités est une impasse ; développer de nouvelles fonctionnalités avant de stabiliser ce prototype est risqué : la liste des correctifs à apporter est longue et tant que la méthode n'est pas remise en question, toute nouvelle fonctionnalité développée ne fera que retarder encore plus le projet au point qu'il ne verrait jamais le jour.

Sur un cycle de développement en spirale (figure 6), Wide est entré en fin de cycle de son premier prototype opérationnel et l'itération suivante peut commencer.

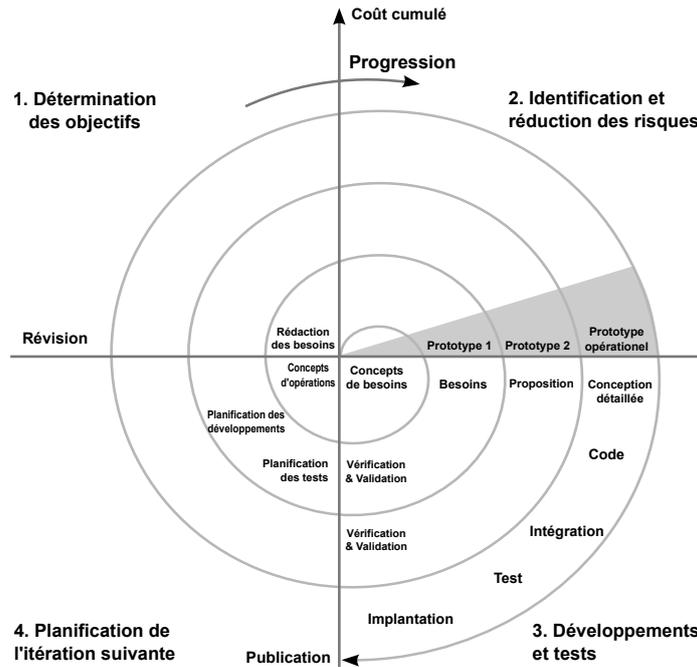


FIGURE 6 – Modèle en spirale

Un seul développeur pour ce projet

Nous allons voir pourquoi le choix de l'opensource plutôt qu'une solution intramuros ou une autre est la plus raisonnable, pour de multiples raisons dont des raisons purement méthodologiques en termes de développement logiciels.

Une problématique transverse

Les critiques de ImageJ et de Bio-Format, naturelles du fait de l'évolution des technologies, font que les efforts se tournent à présent vers la généralisation et la restructuration des programmes. En prenant un peu de recul, on peut voir que la gestion conviviale de gros volumes de données distantes est doublement concernée par ce phénomène. A plusieurs étages la problématique est institutionnellement transverse.

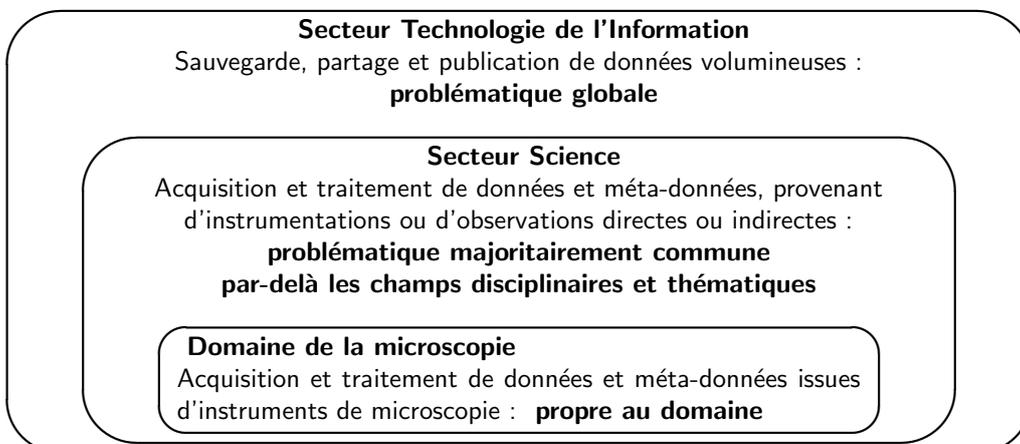


FIGURE 7 – Repositionnement du contexte

L'innovation arrive ailleurs

L'innovation arrive ailleurs, simplement parce qu'il y a plus d'ailleurs que d'ici. Aussi naïf que cela peut paraître, c'est la réalité : aussi intelligente, créative et innovante que vous pensiez que votre organisation puisse être, il y aura toujours des gens plus malins, plus créatifs et plus novateurs en dehors de votre organisation qu'à l'intérieur. De plus, la majorité des "ailleurs" ne sont pas spécialement concernés par les réalisations à l'intérieur de votre espace. Mais les utilisateurs qui se servent déjà d'un outil pour travailler sont en bonne position pour donner des suggestions à propos des directions dans lesquelles cet outil devrait évoluer. Même si ces utilisateurs n'ont pas de suggestions concrètes, la façon dont ils utilisent les outils peut vous fournir des indices⁵ sur le moyen de les améliorer [GG05].

Comme Henry Chesbrough l'a fait remarquer [CHES03], le profil des dépenses en R&D suggère qu'il est de moins en moins fréquent de trouver des gens innovants dans de grandes sociétés monopolisant virtuellement des aires technologiques spécifiques. Comme il est impossible de déclencher et de concrétiser toutes les innovations en interne, le jeu entre davantage dans la connection que dans la domination.

Pour y parvenir, les sociétés ont besoin de trouver les moyens d'utiliser les innovations extérieures et de devenir partie intégrante d'une fabrique d'innovation distribuée, au travers d'une combinaison de licences et de dons bien ciblés. Bien que ce concept de dons puisse ne pas bien cadrer avec ce marché de libre-échange capitaliste, il l'est, lorsqu'on réfléchit au contexte de collaboration pour la construction d'infrastructures de bases. L'*open source*, c'est d'abord l'exploitation des moteurs de l'innovation logicielle.

Le respect des délais

En quoi ceci améliore le respect des délais de livraisons ? Grâce à la confiance et à la solidité d'un composant logiciel sous licence libre. Si un paquet logiciel open source a un nombre suffisamment important d'utilisateurs (où « suffisamment important » commence entre 5,000 et 10,000), vous pouvez être sûr qu'il est fonctionnel, fiable et bien documenté. Pourquoi ? Si les utilisateurs ne sont pas experts, ils ne pourraient pas être 5,000 à l'utiliser sans qu'il possède ces propriétés. Le développeur ne pourrait pas gérer les questions, les bugs et les rapports de bugs. S'ils le sont et qu'ils sont suffisamment nombreux, ils les corrigeront et ils peuvent le faire puisque le paquet est sous licence libre.

Poursuivre ce projet avec un seul développeur est donc possible, à condition de se tourner vers la bonne solution active, bénéficiant d'un support communautaire et proprement documentée. En toute logique, rejoindre une équipe qui résoud déjà la problématique commune de la gestion des fichiers va permettre d'accélérer et de consolider cet aspect, d'offrir un visuel agréable à l'utilisateur et de se concentrer sur les données en elles-mêmes, les besoins spécifiques aux laboratoires et fournir des outils d'analyses plus évolués.

La solution la plus rationnelle que nous suggérons repose sur PYDIO⁶, une base puissante qui offre de nombreux avantages :

- Un support communautaire, car elle largement diffusée sous licence libre.⁷
- Une documentation à jour.
- Des partenaires de confiance (*RedHat, Lacie, Qnap, DreamHost*).
- Un logiciel déjà implanté dans l'enseignement supérieur.
- Un développement maîtrisé.
- Une philosophie de conception modulaire.

5. Voir le projet Centriole Screen du Gönczy Lab de l'EFPL. <http://centriolescreen.vital-it.ch/>

6. Créé par une société française qui a remporté le Trophée Starup Numérique 2012. <http://pyd.io>.

7. 500'000 téléchargement en septembre 2013.



Et OMERO ?

OME Remote Objects (OMERO) est une alternative en avance sur Wide grâce à l'activité, la communication, une documentation claire et des contributeurs qu'elle a réussi à agréger. Toutes les informations sur OMERO sont sur www.openmicroscopy.org.

Les technologies exploitées par Omero sont en avance sur Wide, quoique très similaires.

Contribuer à Omero apparaît comme un choix d'allégeance pour MRI. Attention toutefois, choisir cette stratégie fait perdre la transversalité institutionnelle des composants logiciels, qui risquent alors de rester cantonnés au et connus du domaine seul de la microscopie. Les contributeurs à l'initiative SCIFIO l'ont bien compris mais Omero est déjà parti dans un cycle de développement sur lequel il ne retournera probablement pas avant plusieurs versions, contrairement au modèle en spirale sur lequel Wide démarre sa nouvelle itération.

Les principaux avantages que nous avons par rapport à Omero sont la remise en question du prototype, la fédération d'une force active de développement qui va au-delà du secteur biologique et un plan de conception basé sur la modularité des programmes.

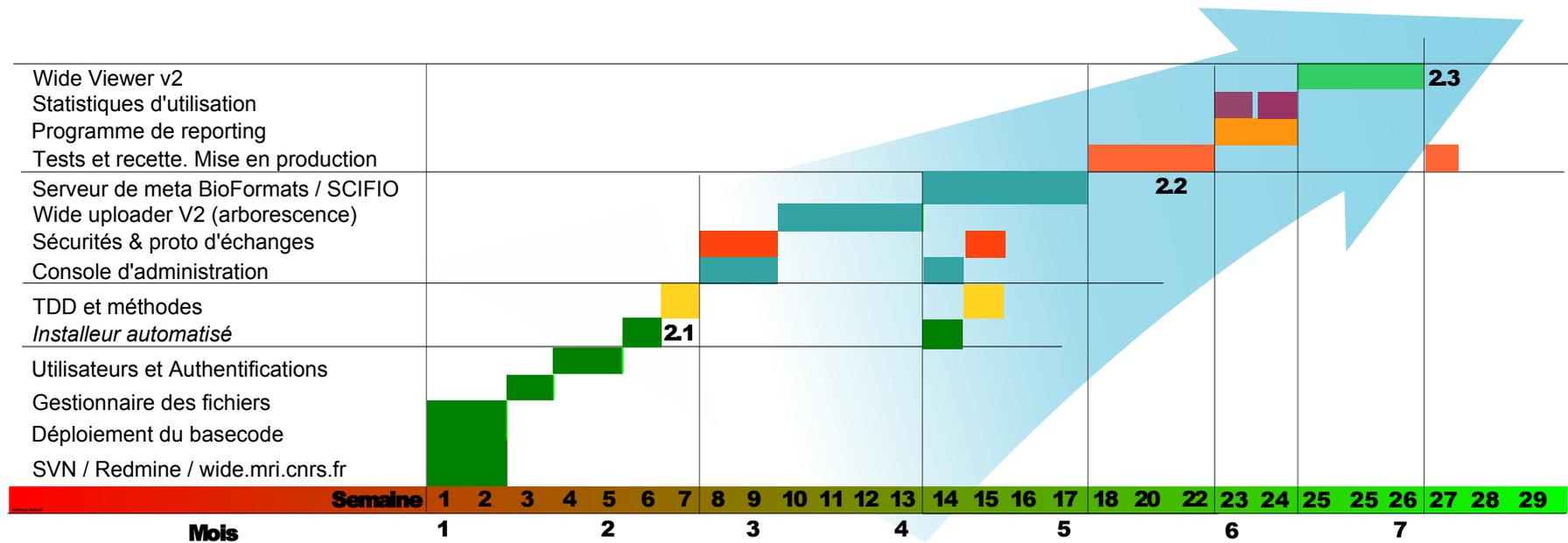
Cette planification devra être accompagnée d'un suivi attentif des développements de Icy⁸ pour permettre de déployer graduellement toute l'architecture en cours d'année 2014. Le projet ImageJ2 peine à décoller, tandis que Icy est déjà opérationnel et assure la compatibilité ascendante avec ImageJ.

Voir plus loin : une solution pérenne de sauvegardes et d'authentications des données scientifiques.

Même si cela peut sembler déborder des spécifications premières, le montage d'une ANR a été proposé pour démontrer une solution de pérennisation et d'authentification des données [DOUZ12], qui puisse garantir leur persistance sur du très long terme. Certaines briques logiques de l'informatique moderne sont considérées comme acquises et nous les remettons en question. Ce regard déconstructiviste appuyé d'un long travail de recherche ont pointé du doigt que malgré les solutions que nous apportons, une problématique de fond demeure et un vaste champs de solutions peut être apporté sur cette base canonique et axiomatique ; Un cas de recherche aux implications théoriques et technologiques en attente immédiate de transfert – que nous avons tenu à mentionner ici, car le projet Wide, s'il est préparé à accueillir ce modèle, peut l'inclure d'une façon naturelle pour amener davantage de contributeurs et un meilleur développement.

8. <http://icy.bioimageanalysis.org/>

Roadmap 2014



Les délais d'estimation pour atteindre une plate-forme stable et opérationnelle sont de 7 mois. Sur la durée totale du projet et au vu des recommandations mentionnées précédemment, le décalage pessimiste maximal que nous prévoyons est de dix semaines. Une estimation optimiste aménage plus de temps pour anticiper des fonctionnalités supplémentaires : déconvolution, visualiseur 3d de piles d'images, l'accélération des conversions et des découpages des images ndpi (nanozoomer), le tatouage et la sauvegarde des colorimétries appliquées, un serveur d'archivage, la signature et la comparaison des images. la planification du service de traitements et d'analyses distantes pourra être planifié après que la version 2.2 soit achevée.



Conclusion

Nous avons expliqué comment développer Wide dans une deuxième version stable, fonctionnelle et évolutive. Pour garantir que le projet soit correctement piloté dans des délais raisonnables, nous recommandons de fédérer une communauté de développeurs, d'opter pour un développement modulaire, le remplacement de technologies obsolètes, la mise en place d'une méthode rigoureuse et de contrôles de tests plus sévères. Au terme de quoi, la plateforme WIDE pourrait devenir un élément informatique central d'une infrastructure de recherche.

Références

- [GG05] Ron Goldman, Richard P. Gabriel, *Innovation Happens Elsewhere, Open Source as Business Strategy*.
- [CHES03] Chesbrough, Henry (2003). *Open Innovation : The New Imperative for Creating and Profiting from Technology*. Cambridge, MA : Harvard Business School Press.
- [BROOKS95] Brooks, Frederick P. (1995). *The Mythical Man-Month : Essays on Software Engineering*. Reading, MA : Addison-Wesley.
- [DOUZ12] Vincent Douzal, Nicolas Bercher, *From natural hazards to outer space and to Plan9. Proceedings of the 6th International Workshop on Plan 9, oct 20-21, 2011, pp.1-10. ISBN 978-84-694-9010-5*.
- [MISC69] MISC n°69 sept-oct. 2013
- [BLOCH] www.laurentbloch.org/MySpip3/spip.php?article272

Remerciements à Olivier Micquel, Alexandre Granier, Stéphane Laborie et Vincent Douzal.