INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

Encadrants

Gildas AVOINE

Barbara KORDY

Glasir

Application au réseau STAR

Rapport de bilan de planification

Étudiants

Pierre-Marie AIRIAU

Valentin ESMIEU

Maud LERAY

Résumé

Ce rapport revient sur notre projet portant sur l'analyse de la sécurité des systèmes. Dans le cadre de ce projet, nous avons développé Glasir, un logiciel d'analyse d'arbres d'attaque et de défense. Pour ce faire, nous avons également amélioré ADTool, un outil d'édition d'arbres, afin de l'intégrer à notre logiciel.

Ce rapport rend compte de la conduite du projet au regard de la planification initialement retenue. Il rapporte donc les difficultés auxquelles nous avons fait face, ainsi que les ajustements mis en place pour y répondre.

26 mai 2015

Table des matières

1 Introduction **5**

2 Planification initale **6**
2.1 La méthode SCRUM 6
2.2 Versionnement du développement 6
2.3 Répartition des tâches 7

3 Écarts à la planification **9**
3.1 Gestion du temps 9
3.2 Décisions d'implémentation 9

4 Retour sur les risques **12**
4.1 Rappel des risques pressentis 12
4.2 Rétrospective 13

5 Pistes d'amélioration pour la gestion de projet **14**
5.1 Renforcement de la pré-étude 14
5.2 Meilleure définition du cahier des charges 14
5.3 Précision de la méthodologie de test 15

6 Conclusion **16**

1 Introduction

La sécurisation des systèmes est l'une des problématiques majeures de la société moderne. En ce sens, de nombreuses méthodologies ont été développées dans le but d'identifier les risques et de les quantifier. C'est ainsi que le concept d'ADTrees [1] [6], un formalisme permettant de représenter sous forme d'arbre les étapes d'une attaque contre un système, a vu le jour. L'un des logiciels destinés à leur modélisation informatique s'appelle ADTool [7].

Lors de la prise en main de ce logiciel, nous avons trouvé qu'il présentait quelques limitations. Tout d'abord, ADTool ne fournissait pas d'outils permettant à l'utilisateur de simplifier l'analyse des ADTrees. En effet, dans un cas concret d'expertise en sécurité, l'ADTree utilisé sera parfois de très grande taille et dans ce cas, il est difficile pour l'expert d'en extraire des informations pertinentes au premier coup d'œil. De plus, il manquait à ADTool certaines fonctionnalités « basiques » telles que le couper/copier/coller ou l'annulation d'une action.

L'objectif de ce projet était la réalisation d'un logiciel permettant de dépasser ces limites en fournissant à l'utilisateur, un expert en sécurité, des outils d'analyse des ADTrees. Ce logiciel a été nommé Glasir en référence à l'arbre aux feuilles d'or de la mythologie nordique [8], et son logo est visible sur la Figure 1. Il dispose de trois fonctionnalités principales qui ont été détaillées dans les rapports précédents : l'Éditeur de fonctions, le Filtre, et l'Optimiseur. ADTool a quant à lui été amélioré, et utilisé au sein de Glasir comme viewer d'ADTrees.

Ce rapport a pour but de revenir sur les phases de planification et d'organisation du projet, au vu de son implémentation réelle. Dans un premier temps, il rappellera les principales lignes de la planification initialement prévue. Ensuite, il détaillera et justifiera les écarts à cette planification, constatés à la fin du développement effectif du logiciel. Une rétrospective concluera ce rapport, afin de proposer quelques pistes à suivre pour améliorer la gestion de nos projets à venir.

Figure 1 – Logo du logiciel Glasir.

1. Abréviation d'« Attack-Defense Trees », ou « Arbres d'Attaque et de Défense » en français.

2 Planification initale

Cette section a pour but de rappeler les grandes lignes de la planification initiale du projet, qui avait été détaillée dans le rapport de planification [1]. Pour rappel, la méthode de gestion de projet que nous avons choisie d'appliquer est la méthode SCRUM, dont le principe est brièvement rappelé dans la SOUS-SECTION 2.1.

2.1 La méthode SCRUM

En décembre dernier, après avoir déjà rédigé et rendu les rapports de pré-étude [3] et de spécifications fonctionnelles [2], il a fallu penser à la planification du développement de Glasir. Après discussions, nous nous sommes finalement dirigés vers une méthode agile proche du « SCRUM ». Ce choix s'est basé sur les constats suivants :

Depuis le début du projet, un « coordinateur » était responsable de la réalisation et du rendu d'un livrable (rapport, version logicielle, etc). Ce rôle, attribué à une nouvelle personne à chaque nouveau livrable, était assimilable au rôle de « Scrum Master » tel que défini dans la méthode SCRUM.

Le coordinateur endossait également la responsabilité de « Product Owner », partagée avec les encadrants du projet pour ce qui est de la définition des objectifs et surtout de l'acceptation des livrables.

Au lieu de la mêlée quotidienne du SCRUM, un rythme hebdomadaire a été adopté pour les réunions car Glasir n'est pas un projet à temps plein.

Enfin, le développement de chaque version a été divisé en sprints au cours desquels chaque développeur était associé à un sous-ensemble de tâches nécessaires à la réalisation de Glasir. Ces tâches ont constituées les « Users Stories » du projet, et sont détaillées dans la SOUS-SECTION 2.3. Avant de revenir sur cette répartition, quelques rappels sur le versionnement de l'implémentation s'imposent dans la SOUS-SECTION 2.2.

2.2 Versionnement du développement

L'implémentation de Glasir (et des améliorations apportées à ADTool en parallèle) a été séparée en trois versions (deux intermédiaires et une finale), chacune étant concrétisée par un rendu aux encadrants sous la forme d'un exécutable fonctionnel. Le contenu de ces trois livrables est rapidement décrit ci-dessous.

Version 0.1 Huit grandes tâches ont été identifiées pour cette version et sont présentées dans la TABLE 1.

Cible	Id	Tâche	Technologies	Durée
Glasir	1.1	Squelette interface	WPF	6h
	1.2	Gestion fichiers projet	C++	20h
	1.3	Intégration ADTool dans Glasir	JNI	20h
	1.4	Évaluateur de fonction	Java	12h
	1.5	Interface évaluateur	WPF	8h
ADTool	1.6	Valuation ADTrees	Java	18h
	1.7	Refonte langage des ADTrees		16h
	1.8	Vue globale des paramètres		12h
Total				**112h**

TABLE 1 – Tâches associées au développement de Glasir version 0.1.

Version 0.2 Le développement de la version 0.2 est découpé en cinq tâches, résumées dans la TABLE 2.

Cible	Id	Tâche	Technologies	Durée
Glasir	2.1	Algorithme filtrage	C++	24h
	2.2	Interface filtre	WPF	15h
	2.3	Multiples instances d'ADTool	C++, WPF	20h
	2.4	Affichage arbre filtré	Java, WPF	16h
ADTool	2.5	Couper/copier/coller	Java	25h
Total				**100h**

TABLE 2 – Tâches associées au développement de Glasir version 0.2.

Version 1.0 Quatre tâches ont été identifiées pour la version 1.0 de Glasir, présentées dans la TABLE 3.

Cible	Id	Tâche	Technologies	Durée
Glasir	3.1	Optimiseur	C++, WPF	30h
	3.2	Bibliothèque de modèles	C++, WPF	20h
	3.3	Harmonisation interface	WPF	16h
	3.4	Packaging	-	8h
ADTool	3.5	Ctrl-Z	Java	16h
Total				**90h**

TABLE 3 – Tâches associées au développement de Glasir version 1.0.

À l'issue de cette première division du travail à réaliser, il a fallu répartir entre les développeurs les tâches citées précédemment pour chacune des versions. Ces assignations sont rappelées dans la SOUS-SECTION 2.3.

2.3 Répartition des tâches

Pour répartir correctement les tâches unitaires présentées dans la SOUS-SECTION 2.2, nous avons fait le choix de « spécialiser » chacun des trois développeurs comme indiqué ci-après :

– Pierre-Marie Airiau était en charge des trois modules principaux de Glasir et des principaux problèmes algorithmiques ;
– Valentin Esmieu s'occupait surtout de l'interface de Glasir et de son fonctionnement global ;
– Maud Leray était responsable des améliorations apportées à ADTool.

Ce choix de planification permettait non seulement d'équilibrer le nombre d'heures de travail par personne, mais aussi de limiter pour chacun l'apprentissage de nouvelles technologies, ce qui aurait sinon entraîné un ralentissement dans l'avancement du projet. La répartition exacte des tâches est définie dans la table 4.

	Pierre-Marie A.		**Valentin E.**		**Maud L.**	
	Id tâche	**Durée**	**Id tâche**	**Durée**	**Id tâche**	**Durée**
Version 0.1	-	**38h**	-	**34h**	-	**40h**
	1.3	10h	1.1	6h	1.2	10h
	1.4	12h	1.2	10h	1.6	18h
	1.7	16h	1.3	10h	1.8	12h
	-	-	1.5	8h	-	-
Version 0.2	-	**39h**	-	**33h**	-	**28h**
	2.1	24h	2.3	20h	2.4	16h
	2.2	15h	2.5	13h	2.5	12h
Version 1.0	-	**25h**	-	**23h**	-	**42h**
	3.1	15h	3.1	15h	3.2	10h
	3.2	10h	3.4	8h	3.3	16h
	-	-	-	-	3.5	16h
Total		**102h**		**100h**		**110h**

Table 4 – Répartition des tâches, par personne et par version.

Ici s'arrête le résumé de la planification initiale, il est donc temps de passer à la comparaison de cette dernière avec la planification effective constatée à l'issue du développement de Glasir. C'est le but de la Section 3.

3 Écarts à la planification

Durant l'implémentation de Glasir, plusieurs évènements ont impacté la planification initialement prévue du projet. Pour rendre compte de cela, les diagrammes de Gantt d'avant et d'après l'implémentation sont respectivement présentés à la FIGURE 2 et à la FIGURE 3. Les sous-sections ci-après détaillent les changements majeurs du point de vue de la planification du projet.

3.1 Gestion du temps

L'un des premiers constats que l'on peut énoncer maintenant que l'implémentation est terminée, est que nous avions de manière générale légèrement surévalué la durée des tâches. Il est possible de le remarquer en comparant la FIGURE 2 (planification initiale) et la FIGURE 3 (adaptation de la planification suite au développement). Cette surévaluation a cependant permis de gagner du temps, qui a ensuite été utilisé pour améliorer certaines fonctionnalités du logiciel ou bien pour en créer de nouvelles. Ce gain de temps a également permis de surmonter les difficultés rencontrées sans prendre de retard.

Ainsi, nous avons pu gérer la sauvegarde de plus d'une modification de l'arbre dans ADTool, ce qui a permis l'annulation de plusieurs actions, au lieu d'une seule comme annoncé initialement.

Le temps supplémentaire a également permis d'autres améliorations mineures, comme le nommage des fenêtre, mais a surtout permis de développer un « view mode » pour ADTool. Son intérêt est expliqué plus loin, dans la SECTION 3.2.

Enfin, ce gain de temps a aidé à gérer les difficultés imprévues rencontrées lors de l'élaboration des algorithmes des modules principaux de Glasir, et ce sans faire prendre de retard au projet. Nous avons même été en mesure de fournir une version plus intelligente du Filtre que celle qui était prévue, puisqu'elle est capable de trouver les nœuds incorrects en-dessous des nœuds conjonctifs.

3.2 Décisions d'implémentation

Malgré le gain de temps évoqué dans la SOUS-SECTION 3.1, nous avons tout de même été confrontés à des difficultés que nous n'avons pas pu gérer. Deux éléments du logiciel initialement prévus ont en effet dûs être abandonnés : l'affichage multi-paramètres dans ADTool, et l'intégration complète d'ADTool au sein de Glasir.

L'affichage multi-paramètres, que nous estimions être une fonctionnalité relativement simple à implémenter, s'est avérée très complexe après l'étude approfondie du code source d'ADTool. En effet, l'implémentation de cette fonctionnalité aurait nécessité de réorganiser l'intégralité du code, ce qui n'était pas possible au vu du temps imparti dans le cadre de ce projet.

L'intégration complète d'ADTool au sein de Glasir a quant à elle été abandonnée, malgré de nombreuses tentatives. Nous n'avons simplement pas réussi à trouver des méthodes permettant l'exécution d'un thread Java au sein d'un processus C#. Pour remédier à cette impasse, une version « view mode »d'ADTool a été mise au point, dans laquelle toutes les fonctionnalités de création et de modification d'ADTrees sont désactivées. Cela permet d'utiliser ADTool en tant que processus extérieur à Glasir, sans craindre des incohérences dûes à des modifications d'arbres sous ADTool ignorées de Glasir.

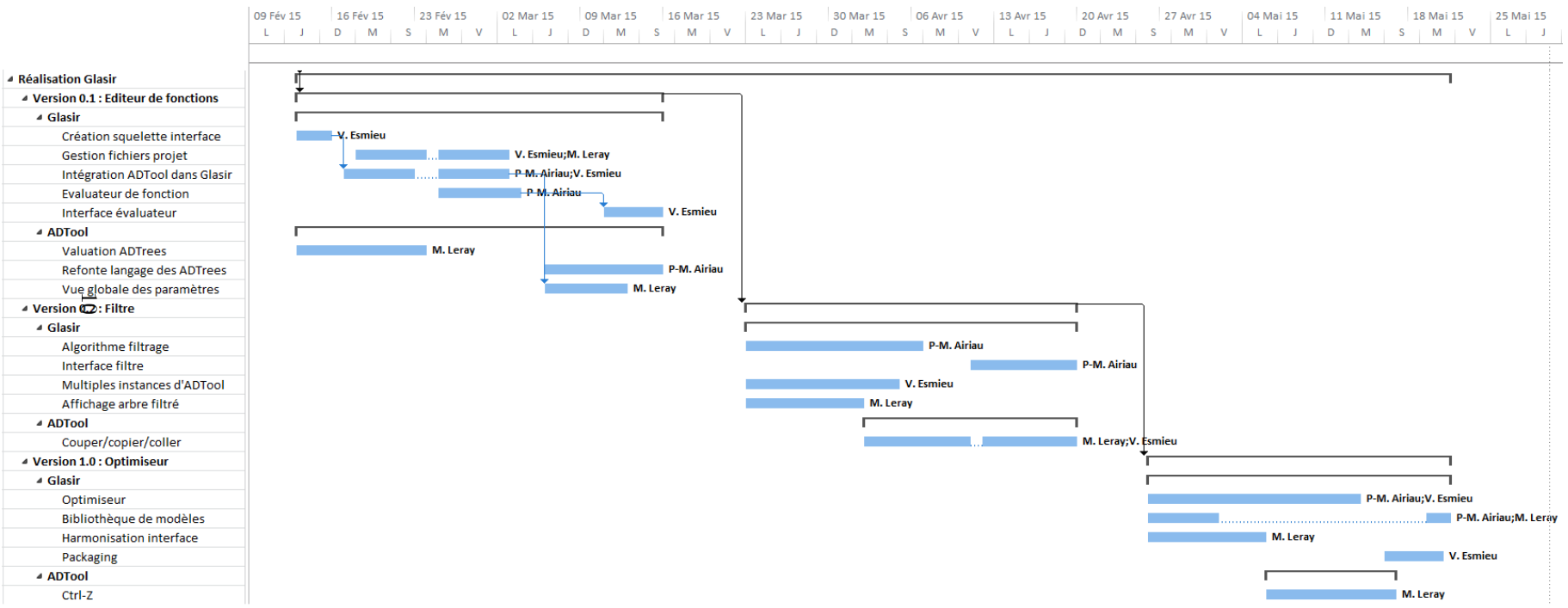

FIGURE 2 – Diagramme de Gantt au début du projet Glasir.

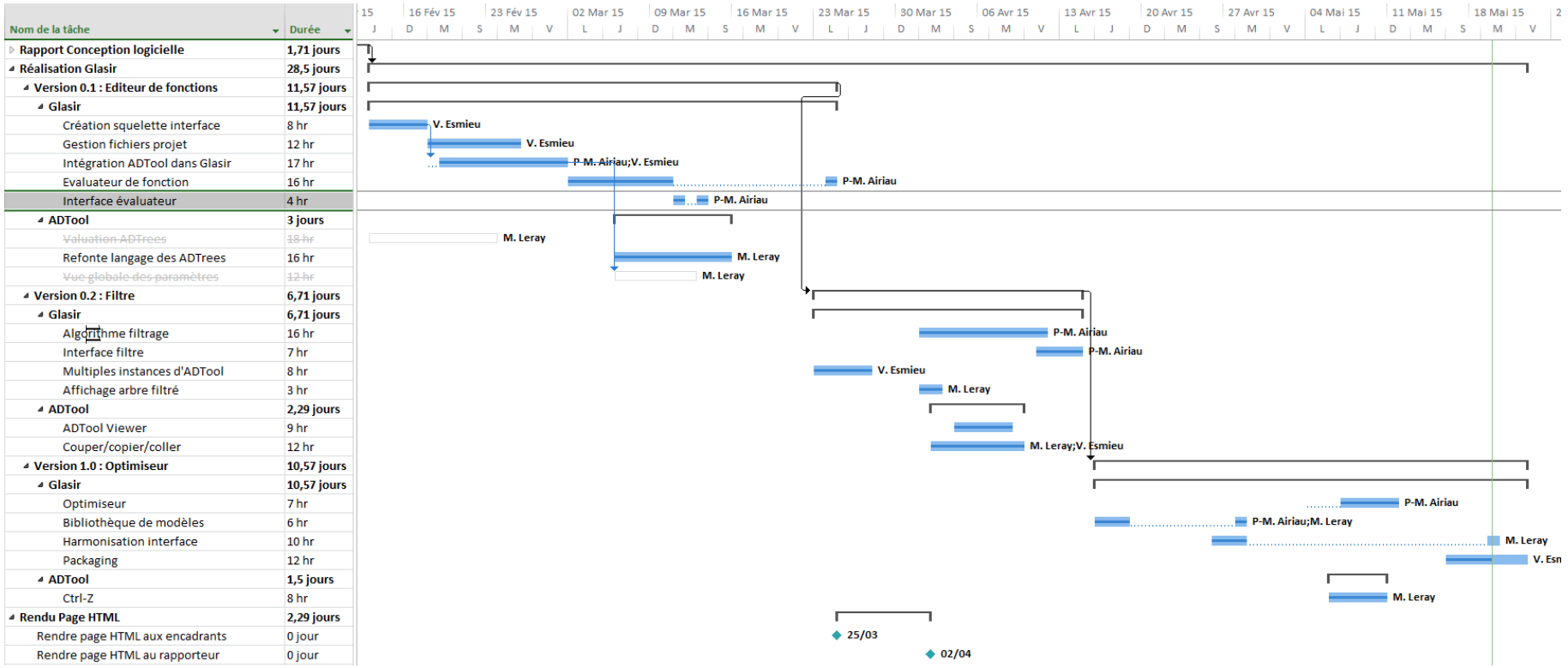

FIGURE 3 – Diagramme de Gantt à la fin du projet Glasir.

4 Retour sur les risques

Le rapport de planification [1], en plus de la répartition des tâches, comportait une section sur les risques potentiels concernant le développement de Glasir. La présente section commence par rappeler ces risques dans la SOUS-SECTION 4.1, avant de revenir sur ceux rencontrés lors de l'implémentation dans la SOUS-SECTION 4.2.

4.1 Rappel des risques pressentis

Afin de délivrer ce projet dans les délais prévus, une évaluation des risques a été effectuée lors de la rédaction du rapport de planification [1]. Pour chacun de ces risques, les notions de probabilité (Pr) et de criticité (Cr) ont été estimées. Les valeurs possibles sont les suivantes : L pour « Low » (faible), M pour « Medium » (moyenne), H pour « High » (élevée). Dans le but de réagir au mieux en cas d'apparition de ces aléas, des solutions ont été proposées. Le résultat de cette réflexion est présenté dans la TABLE 5.

Id	**Risque**	**Tâches concernées**	**Pr.**	**Cr.**	**Solution**
1	Mauvaise estimation des durées nécessaires aux tâches unitaires	Toutes	H	H	Prioriser les tâches restantes
2	Apparition d'un bug difficile à corriger	Toutes	H	H	Faire des tests unitaires
3	Difficulté à créer une grammaire	1.4	L	M	Simplifier les expressions à évaluer
4	Manque de connaissances techniques	Toutes	H	H	Approfondir nos connaissances
5	Rédaction trop tardive de la documentation	Toutes	M	L	Commenter le code au fur et à mesure
6	Mauvaise compréhension du code d'ADTool	Toutes celles sur ADTool	M	M	Contacter le développeur d'ADTool
7	Mauvaise communication entre ADTool et Glasir	1.4, 2.1, 3.1	M	H	Utiliser le fichier XML lisible par ADTool
8	Perte de temps sur une tâche secondaire	Toutes	M	L	Compter ses heures, faire le point lors des réunions
9	Algorithme ralentissant le logiciel	3.1, 2.2	L	L	Optimiser l'algorithme
10	Échec de l'intégration d'ADTool dans Glasir	1.2	L	H	Lancer ADTool séparément

TABLE 5 – Tableau des risques du rapport de planification [1].

Maintenant que les risques potentiels ont été rappelés, il est intéressant de revenir sur les risques réellement rencontrés.

4.2 Rétrospective

En ayant connaissance de la SECTION 3, il apparaît que la majorité des risques repérés lors du 1er semestre ne se sont finalement pas présentés. Cependant, cela vaut la peine de revenir sur certains d'entre eux :

Ainsi, le risque n ° 1 s'est bel et bien produit, mais avec un effet plutôt positif finalement puisque les durées avaient été sur-estimées, comme cela a été expliqué dans la SECTION 3, et non sous-estimées. Cela a permis de gagner du temps et d'améliorer certains aspects de Glasir et d'ADTool, comme le Filtre ou le Undo par exemple.

Le risque n ° 10 s'est également produit, en conséquence du risque n ° 4. Nous avons en effet manqué des connaissances techniques pour réaliser l'inter-connectivité totale entre deux langages (C# et Java), et nous n'avons pas eu le temps de progresser suffisamment pour résoudre le problème. Nous avons donc réalisé un compromis entre ce qui avait été prévu et les difficultés rencontrées : Glasir utilise bien ADTool, mais comme afficheur d'ADTrees uniquement et en tant que processus séparé. Ce compromis a été rendu possible par la création du « view mode » d'ADTool, explicité dans la SECTION 3.

Enfin, le risque n ° 7 n'a pas vraiment pu se réaliser car il a été décidé dès le début de l'implémentation de travailler sur les ADTrees sous format XML, comme le préconisait la solution envisagée. Cela est dû au fait que le format des fichiers par défaut d'ADTool (.adt) est de type binaire, et donc très difficile à exploiter.

Toutes les parties du rapport de planification [1] ayant été revues, il est a présent nécessaire de prendre du recul par rapport à la gestion de projet qui a été réalisé. C'est pourquoi la SECTION 5 propose quelques pistes d'améliorations pour éviter les écarts de planification mis en évidence précédemment.

5 Pistes d'amélioration pour la gestion de projet

La présente section a pour objectif de revenir sur les défauts de notre gestion de projet afin d'en tirer de bonnes pratiques pour l'avenir. La liste des améliorations possibles indiquées ci-dessous n'est pas exhaustive, mais elle contient nos principaux constats à l'issue de ce projet.

5.1 Renforcement de la pré-étude

Le contexte du projet avait été bien établi lors de la phase de pré-étude du projet (en septembre dernier) : notions de sécurité, formalisme des ADTrees, etc. Cependant, nous avons réalisé lors du développement que les détails techniques liés à l'implémentation auraient quant à eux nécessité des recherches plus poussées. Deux exemples seront cités ci-dessous afin d'illustrer cette problématique.

Intégration Java dans un programme C# La planification initiale prévoyait d'intégrer ADTool (codé en Java) directement dans Glasir, développé en C#. Au moment de l'implémentation effective, de nombreux essais ont été réalisés dans ce but mais aucun n'a été concluant. Il a donc fallu conserver les deux logiciels séparés. Il est à noter que ce choix permet tout de même de mieux séparer l'édition des ADTrees, réalisée par ADTool, de leur analyse par Glasir. Ceci reste cependant un écart à la planification qui aurait pu être évité si nous nous étions mieux renseignés sur la compatibilité des deux langages lors de la phase de pré-étude.

Reprise d'un code existant Il était prévu d'apporter à ADTool un certain nombre d'améliorations, ce qui paraissait simple dans la mesure où le langage de programmation nous était déjà familier. Nous n'avons cependant pas su totalement anticiper les complications qu'induisait la reprise d'un code déjà existant, et n'avons pas suffisamment étudié le code source d'ADTool lors de la phase de pré-étude. Ceci explique notamment que nous ayons annoncé la fonctionnalité d'affichage multiple des paramètres, qui aurait nécessité de re-structurer l'intégralité du programme et donc d'y consacrer un temps beaucoup trop élevé.

À travers les deux exemples précédents, il est évident qu'un manque de renseignements quant au développement technique d'un projet peut poser des soucis en terme d'accomplissement des objectifs. C'est donc un point à améliorer lors de la phase de pré-étude de nos projets futurs.

5.2 Meilleure définition du cahier des charges

Cette sous-section concerne les trois modules principaux de Glasir que sont l'Éditeur de fonctions, le Filtre et l'Optimiseur. Ces derniers ont été annoncés dans le rapport de pré-étude [3], puis détaillés dans les rapports de spécifications fonctionnelles [2] et de conception [4]. La planification initiale prévoyait de les implémenter au fur et à mesure, au rythme d'un nouveau module par version. Ces échéances ont été correctement respectées, mais quelques modifications furent nécessaires sur chaque modules après leur implémentations suite à des redéfinitions de leur fonctionnement. Ces petits écarts auraient pu être évités si les fonctionnalités du cahier des charges avaient été définies plus en détails dès le début, en se basant par exemple sur une étude de cas permettant d'illustrer l'ensemble des possibilités.

5.3 Précision de la méthodologie de test

La méthodologie de test appliquée au cours des différentes étapes du développement a été détaillée dans le rapport final du projet [5]. C'est justement la rédaction de ce dernier qui a mis en évidence le fait que les tests n'avaient pas été suffisamment rigoureux, malgré le versionnement du développement. En effet, chaque phase de test (correspondant à chaque rendu de version, sauf pour les développeurs qui ont testé en continu) aurait mérité d'être plus réfléchie. Les tests ont une importance capitale dans un projet informatique, c'est pourquoi il s'agit d'un point à améliorer dans le cadre de projets futurs. Cela peut être réalisé en suivant les pistes ci-dessous, dont la liste n'est pas exhaustive :

- définition d'une procédure précise de test ;
- description claire des objectifs des tests (ergonomie du logiciel, cohérence des résultats, etc.) ;
- conduite de tests par des individus totalement extérieurs au projet ;
- conduite de tests par un futur utilisateur potentiel (dans notre cas, un expert en sécurité connaissant les ADTrees).

Les constats précédents entraînent une rétrospective sur le projet qui est présentée dans la SECTION 6.

6 Conclusion

Au cours de cette année, nous avons pu être confrontés à toutes les étapes de la gestion d'un projet. Partant d'une problématique et d'un logiciel open source existant (ADTool), nous avons commencé par définir le contexte dans lequel nous allions travailler ainsi que le profil d'utilisateur ciblé (un expert en sécurité). Cela a donné naissance au 1er rapport, celui de pré-étude [3], qui contenait également une ébauche du cahier des charges. Nous avons ensuite détaillé ce dernier dans le rapport de spécifications fonctionnelles [2], qui contenait en plus un aperçu de l'architecture logicielle. Il a ensuite fallu planifier le développement de Glasir en répartissant les tâches entre les développeurs, ce qui a été décrit dans le rapport de planification [1]. Un dernier rapport, celui de conception [4], a été nécessaire avant l'implémentation pour mieux appréhender les technologies utilisées et pour définir une structure plus précise du logiciel.

Ce projet a également été une bonne occasion de découvrir le travail en équipe, et tout ce qu'il implique : répartition des tâches, rôle de coordinateur, gestionnaire de versions (Git), etc. C'est une situation que nous rencontrerons fréquemment en entreprise, et à laquelle il faut donc se préparer.

La phase de développement de Glasir arrivant à sa fin, un rapport final de projet [5] a été rédigé afin de comparer l'état final de Glasir à celui promis dans les rapports qui ont précédé. Ce rapport final fut également l'occasion de fournir un compte-rendu des tests effectués ainsi qu'un manuel utilisateur du logiciel.

Le présent rapport de bilan de planification a fait état de quelques écarts constatés par rapport à la planification initiale rappelée dans la SECTION 2. Ces différences ont été décrites et justifiées dans la SECTION 3. Des idées d'amélioration de gestion de projet, destinées à éviter ces écarts à l'avenir, ont ensuite été données dans la SECTION 5.

Mais d'une façon générale, la planification initialement prévue dans le rapport de planification [1] a été suivie. Les délais fixés ont été respectés, et les fonctionnalités annoncées ont presque toutes été implémentées, et ce de manière fonctionnelle. Nous retiendrons donc qu'il est important de soigner la planification d'un projet, mais aussi d'avoir un regard critique à son égard à l'issue du développement afin de progresser de manière continue dans le domaine de la gestion de projet.

Références

[1] Pierre-Marie Airiau, Valentin Esmieu, Hoel Kervadec, Maud Leray, Florent Mallard, and Corentin Nicole. Glasir - Rapport de planification, décembre 2014. Rapport de spécification fonctionnelle, projet de quatrième année, INSA de Rennes.

[2] Pierre-Marie Airiau, Valentin Esmieu, Hoel Kervadec, Maud Leray, Florent Mallard, and Corentin Nicole. Glasir - Rapport de spécifications fonctionnelles, novembre 2014. Rapport de spécifications fonctionnelles, projet de quatrième année, INSA de Rennes.

[3] Pierre-Marie Airiau, Valentin Esmieu, Hoel Kervadec, Maud Leray, Florent Mallard, and Corentin Nicole. Rapport de pré-étude, octobre 2014. Rapport de pré-étude, projet de quatrième année, INSA de Rennes.

[4] Pierre-Marie Airiau, Valentin Esmieu, and Maud Leray. Glasir - Rapport de conception, février 2015. Rapport de conception logicielle, projet de quatrième année, INSA de Rennes.

[5] Pierre-Marie Airiau, Valentin Esmieu, and Maud Leray. Glasir - Rapport final, mai 2015. Rapport final de projet, projet de quatrième année, INSA de Rennes.

[6] Barbara Kordy, Sjouke Mauw, Saša Radomirović, and Patrick Schweitzer. Attack–Defense Trees. *Journal of Logic and Computation*, 24(1) :55–87, 2014.

[7] Piotr Kordy and Patrick Schweitzer. The ADTool. `http://satoss.uni.lu/software/adtool`, 2012.

[8] Andy Orchard. *Dictionary of Norse Myth and Legend*. Cassell, 1997.

INSA Rennes
20 Avenue des Buttes de Coësmes
CS 70839
35708 Rennes Cedex 7

Tél. +33 (0) 2 23 23 82 00
Fax +33 (0) 2 23 23 83 96

www.insa-rennes.fr

INSA

UNIVERSITÉ EUROPÉENNE DE BRETAGNE

Commission des Titres d'Ingénieur

RÉPUBLIQUE FRANÇAISE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE