

Encadrants

Gildas AVOINE

Barbara KORDY

Glisir

Application au réseau STAR

Rapport de planification

Étudiants

Pierre-Marie AIRIAU

Valentin ESMIEU

Hoel KERVADEC

Maud LERAY

Florent MALLARD

Corentin NICOLE

Résumé

Glisir est le logiciel d'analyse d'arbres d'attaque et de défense que nous allons développer. Pour ce faire, nous allons améliorer ADTool, un outil d'édition d'arbres, avant de l'intégrer à notre logiciel.

Dans ce rapport, nous allons présenter les grandes versions du logiciel, en énonçant les tâches unitaires les composant. Ces tâches seront ensuite planifiées et réparties entre les membres du groupe. Par ailleurs, nous aborderons les risques associés, qui peuvent compromettre le développement.

10 décembre 2014

Table des matières

1	Introduction	5
2	Contexte	6
2.1	Acteurs	6
2.2	Besoins et objectifs	6
2.3	Éléments en entrée	6
2.4	Rétrospective	6
2.5	Calendrier	7
3	Organisation	7
3.1	La méthode SCRUM	8
3.2	Tâches unitaires et estimations	8
3.3	Organisation du groupe et répartition des tâches	10
3.4	Risques	11
4	Séquencement	12
4.1	Diagramme de Gantt	12
4.2	Répartition de la charge de travail	12
4.3	Utilisation des ressources	13
5	Rétrospective et conclusion	17

1 Introduction

La sécurisation des systèmes est une problématique majeure de la société moderne. En ce sens, de nombreuses méthodologies ont été développées dans le but d'identifier les risques et de les quantifier [3]. C'est avec cet objectif que le concept d'arbres d'attaque et de défense (« Attack-Defense Trees » en anglais, ou ADTrees) a vu le jour [2].

Lors de la phase de pré-étude, nous avons pu comprendre l'intérêt pratique de la construction des ADTrees. Leur utilisation permet d'identifier de manière précise les différentes attaques possibles contre un système et de les valuer en termes de coût, de probabilité, etc. ADTool (Attack-Defense Tree Tool), un logiciel développé pour l'implémentation de ces arbres sur support informatique, a été étudié pour ce projet [1]. Lors de sa prise en main, nous avons trouvé que le logiciel présentait quelques limitations. En effet, dans un cas concret d'expertise en sécurité, l'ADTree qui modélisera les attaques possibles sur le système sera parfois de très grande taille. Dans ce cas, il est difficile pour l'expert d'en extraire des informations pertinentes au premier coup d'œil. Or, ADTool ne fournit pas d'outil permettant à l'utilisateur de simplifier l'analyse de l'arbre.

L'objectif de ce projet est 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 [4], et nous avons créé son logo visible sur la FIGURE 1. Il disposera de trois fonctionnalités principales qui ont été détaillées dans le rapport de spécifications fonctionnelles :

- le paramètre de synthèse permet d'étendre le nombre de paramètres d'ADTool ;
- le filtre aide l'utilisateur à ôter de l'arbre les nœuds aux valuations hors d'un intervalle défini ;
- l'optimiseur sélectionne le meilleur chemin dans l'arbre selon une certaine valuation.

ADTool sera intégré dans une fenêtre au sein du logiciel, en tant qu'éditeur d'ADTrees.

Ce rapport présente la phase de planification et d'organisation. Dans un premier temps, il rappelle le contexte de ce projet. Ensuite, il détaille la méthode de gestion de projet utilisée, et le partitionnement du développement en tâches unitaires. Enfin, grâce à des diagrammes réalisés avec le logiciel Microsoft Project (souvent appelé MS Project), il précise le séquençement de ces tâches lors du second semestre.



FIGURE 1 – Logo du logiciel Glasir.

2 Contexte

La présente section rappellera les grandes lignes du projet, ses éléments principaux et présentera brièvement le travail déjà effectué.

2.1 Acteurs

Ce projet de 4ème année Informatique (4INFO) concerne un éventail réduit d'acteurs : en effet, il sera destiné aux experts en sécurité de façon générale, et non pas à un client précis. Il sera cependant illustré par l'étude d'un cas particulier : celui du Service des Transports en commun de l'Agglomération Rennaise (STAR).

À l'heure actuelle, notre groupe de projet est constitué de six étudiants. Au second semestre, après le départ à l'étranger de trois d'entre nous, le travail sera donc réparti entre trois développeurs. Comme au premier semestre, la responsabilité de chef de projet sera tour à tour assumée par chacun des membres restants de notre équipe. Enfin, le projet est encadré par deux enseignants à l'INSA de Rennes : Gildas AVOINE et Barbara KORDY, avec lesquels nous avons des réunions hebdomadaires.

2.2 Besoins et objectifs

Comme présenté auparavant dans les rapports de pré-étude et de spécifications fonctionnelles, ce projet a pour but d'aider les experts en sécurité à modéliser et à analyser des situations d'attaque et/ou de défense. Il sera basé sur la théorie déjà bien établie des ADTrees pour aider à la modélisation et à l'étude de problématiques de sécurité. Le rendu final sera un logiciel nommé Glasir, destiné à des experts en sécurité supposés connaître un minimum le principe des ADTrees. L'objectif principal de l'application sera d'aider lesdits experts à exploiter les ADTrees pour obtenir rapidement et facilement les informations qui les intéressent, telles que le chemin le plus coûteux ou le chemin le plus court.

L'application sera développée et testée sous Windows.

2.3 Éléments en entrée

Glasir est un tout nouveau projet, dont les finalités ont été exposées dans les deux premiers rapports rendus. Il intégrera tout de même un logiciel déjà existant (ADTool) permettant de représenter les ADTrees sur support informatique. Quelques améliorations seront cependant apportées à ADTool avant son intégration, afin de le rendre plus ergonomique pour l'utilisateur. Les autres modules de Glasir, détaillés lors du précédent rapport, seront totalement nouveaux et leur création sera assurée par les trois étudiants qui travailleront sur le projet au second semestre. Il en est de même pour l'interface graphique du logiciel.

2.4 Rétrospective

Cette sous-section constitue un bref rappel du contenu des deux rapports précédemment rédigés lors de ce projet.

Rapport de pré-étude Nous avons commencé ce projet par une étude de l'existant : la théorie des ADTrees ainsi que la découverte d'ADTool, un logiciel permettant d'en créer et d'en manipuler. Nous avons relevé lors de cette étude des lacunes empêchant une analyse poussée des ADTrees. Le contexte - les attaques contre les transports en commun, plus particulièrement le STAR - a lui aussi été traité, afin de mieux cerner notre cas d'étude. Nous entendons par « attaque » tout événement nuisant au bon fonctionnement du réseau de transport, . Or il est apparu que ces attaques contre les transports en commun sont assez fréquentes, même si elles ne sont pas toujours volontaires. Suite à cette phase d'analyse, nous avons été en mesure de proposer un premier cahier des charges en date du 23 octobre. Ce dernier a été étoffé et précisé dans le rapport de spécifications fonctionnelles.

Rapport de spécifications fonctionnelles Les fonctionnalités du logiciel ont été décrites de façon exhaustive. Ce document a fourni une première ébauche de l'architecture de Glasir, comprenant les dépendances entre les modules, et établi les différents liens les unissant. Cela nous a donné une vision plus globale et claire du projet, nous permettant de hiérarchiser les tâches à effectuer. Pour finir, nous avons proposé une planification succincte, développée dans le présent rapport. Ce deuxième rapport a quant à lui été remis au rapporteur le 27 novembre.

Ces rapports nous ont aidé à préciser notre vision du projet, à identifier les modules de la solution que nous apporterons et définir les étapes dans sa réalisation. Avant de commencer la planification détaillée du projet, il semble important de rappeler les dates-clés et les périodes importantes associées à ce projet sur l'année.

2.5 Calendrier

Dans le cahier des charges commun à tous les projets de 4INFO, il figure des dates-clés indiquant un rapport à rendre ou une présentation orale. Nous allons donc rappeler ici les échéances que nous devons respecter. La phase de conception logicielle commencera dès les vacances de Noël, le 20 décembre. Elle se terminera le 12 février par un rendu de rapport. Les échéances suivantes du projet seront celles-ci :

- une page HTML sera livrée le 2 avril ;
- le projet final sera rendu sur CD-Rom le 28 mai ;
- une démonstration sera réalisée, également le 28 mai.

L'emploi du temps prévoit des semaines « bloquées », pendant lesquelles nous n'aurons pratiquement aucun cours afin de nous concentrer pleinement sur le développement de Glasir. Ainsi, les semaines du 18 au 23 mai et du 26 au 29 mai sont donc exemptes de cours.

3 Organisation

Cette section va à présent décrire la méthode de travail que nous avons décidé d'utiliser, ainsi que l'organisation que nous avons adoptée au sein du groupe.

3.1 La méthode SCRUM

Afin de définir la méthode de gestion de projet à mettre en place pour la réalisation de Glasir, nous avons discuté ensemble des différentes méthodes à notre disposition. À la suite de nos réunions, nous nous sommes dirigés vers une méthode agile proche du « SCRUM ». Ce choix s'est fait sur les points qui sont détaillés ci-dessous :

L'organisation des rôles que nous avons mise en place depuis le début du projet est basée sur un rôle de « coordinateur », qui est responsable de la réalisation et du rendu d'un livrable (rapport, version logicielle, etc). Ce rôle est attribué à une nouvelle personne à chaque nouveau livrable, de sorte que chacun des membres de l'équipe l'exerce au moins une fois. Le rôle de coordinateur que nous avons défini nous a donc semblé correspondre au rôle de ScrumMaster tel que défini dans la méthode SCRUM.

Le coordinateur endosse également la responsabilité de Product Owner, qu'il partage 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, nous nous réunissons une fois par semaine afin de discuter de l'avancement de nos tâches respectives et de faire part de nos difficultés. Nous avons décidé d'adopter un rythme hebdomadaire plutôt que quotidien car nous ne travaillerons pas à temps plein sur le projet : ce rythme nous a donc semblé plus adéquat. Nous définirons en fonction des retours en réunion les éventuels ajustements à effectuer la semaine suivante, ainsi que dans le reste de la planification.

Enfin, le développement de chaque version sera composé de sprints au cours desquels nous nous consacrerons à un sous-ensemble de tâches nécessaires à la réalisation d'une version de Glasir. Ces tâches constitueront les Users Stories du projet, et sont détaillées dans la SOUS-SECTION 3.2.

3.2 Tâches unitaires et estimations

Trois versions de Glasir seront livrées : deux intermédiaires et une finale. Le développement de chacune de ces versions a été divisé en tâches unitaires, détaillées dans les TABLES 1, 2 et 3.

Pour déterminer le temps nécessaire à l'accomplissement de chaque tâche, nous avons combiné les méthodes d'estimation « Analogique » et « Expertise ». En effet, la méthode « Analogique » consiste à estimer la durée d'une tâche par analogie avec une tâche similaire déjà effectuée (au sein d'un autre projet par exemple). La méthode « Expertise », quant à elle, repose sur le jugement d'un expert. Bien qu'aucun membre de notre groupe ne puisse réellement se considérer comme tel, nos stages respectifs nous ont tout de même permis d'apporter des avis constructifs sur l'estimation de la durée des tâches.

Les résultats de ces différentes estimations sont décrits ci-après pour chaque version. Il est à noter que les tests et la rédaction de la documentation sont pris en compte dans l'estimation du temps nécessaire à la réalisation de chacune des tâches.

3.2.1 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.

La réalisation de cette version sera découpée en trois sprints, présentés ci-dessous :

Sprint 1 Création de l'interface utilisateur de base de Glasir, intégration du logiciel ADTool en tant que fenêtre de Glasir, possibilité de création d'un nouveau projet et sa sauvegarde, affichage des différents arbres d'un projet dans un dock sous forme d'arborescence.

Sprint 2 Reconnaissance d'une formule mathématique par le logiciel ADTool, création d'un paramètre en fonction de l'évaluation de cette formule, et amélioration de la représentation des arbres sous forme syntaxique au sein d'ADTool.

Sprint 3 Affichage de plusieurs paramètres par arbre, possibilité de communication entre le module éditeur de fonction de Glasir et la partie création d'un paramètre de synthèse implémentée précédemment dans ADTool.

3.2.2 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.

Nous avons réparti les tâches en trois sprints :

Sprint 4 Implémentation de l'algorithme de filtrage, possibilité d'ouverture de plusieurs instances d'ADTool au sein de différents onglets de l'interface de Glasir.

Sprint 5 Affichage du panneau relatif à ce module au sein de l'interface utilisateur de Glasir, implémentation de la fonction couper/copier/coller.

Sprint 6 Affichage de l'arbre filtré.

3.2.3 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.

Les sprints se décomposeront de la façon suivante :

Sprint 7 Implémentation de l’algorithme de l’optimiseur, création d’une bibliothèque de modèles.

Sprint 8 Affichage du panneau de l’optimiseur dans Glasir, possibilité d’annuler une action dans ADTool.

Sprint 9 Harmonisation de l’interface, et réalisation du packaging.

3.3 Organisation du groupe et répartition des tâches

Lors de la phase de développement de Glasir, notre groupe de projet sera réduit à trois étudiants : Pierre-Marie AIRIAU, Valentin ESMIEU et Maud LERAY. Les trois autres, Florent MAL-LARD, Hoel KERVADEC ainsi que Corentin NICOLE partiront en effet étudier à l’étranger.

Nous comptons donc répartir les tâches selon trois composantes générales : Maud L. travaillera principalement sur les améliorations d’ADTool, Pierre-Marie A. sur les algorithmes et sur la communication entre ADTool et Glasir, et Valentin E. s’orientera sur l’interface utilisateur de Glasir. Chacun sera considéré comme « spécialiste » dans son domaine, cependant les réunions permettront d’être au courant de l’avancée des autres tâches. Ainsi, il sera possible de donner un avis sur les méthodes utilisées, et nous serons à même de nous entraider. La TABLE 4 donne la répartition des tâches plus en détails.

	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.

3.4 Risques

Afin de délivrer ce projet dans les délais prévus, une évaluation des risques a été effectuée. Celle-ci a pour but d'identifier les différents facteurs qui pourraient engendrer des difficultés dans la réalisation des tâches, telles qu'une incapacité à implémenter une partie du cahier des charges ou un retard dans le planning. Nous avons associé à chaque élément de cette liste les notions de probabilité (Pr) et de criticité (Cr). 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 définies. Le résultat de cette réflexion est présenté dans la TABLE 5.

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

TABLE 5 – Tableau des risques

4 Séquencement

Pour établir notre planification de la réalisation de Glasir, nous avons utilisé le logiciel MS Project. Les diagrammes présentés dans cette section tiennent compte des différents jalons qui ont été posés, soit imposés à tous les groupes de projet de 4INFO, soit définis par nous-même.

4.1 Diagramme de Gantt

Le diagramme de Gantt visible sur la FIGURE 2 illustre le séquencement de notre projet. Chaque tâche a été attribuée à une personne, et les durées ont été calculées pour une moyenne d'une heure et demie de travail personnel par jour. Lors des semaines du 18 et du 25 mai, bloquées pour les projets de 4INFO, la quantité de travail personnel a été augmentée. Si une tâche est réalisée par plusieurs personnes à la fois, les contributions de ces différentes personnes sont séparées par des pointillés.

4.2 Répartition de la charge de travail

Le planning de la FIGURE 3 illustre pour chaque étudiant la répartition du temps de réalisation des tâches, ainsi que le cumul des heures de travail. Nous avons essayé d'être aussi homogènes que possible dans la répartition du travail, les différences de répartition étant la conséquence de l'irrégularité des durées des tâches. Les quantités de travail de la dernière semaine de chacune des

versions ont été volontairement abaissées afin de permettre de rattraper d'éventuels retards dans la réalisation.

4.3 Utilisation des ressources

Le dernier graphique, celui de la FIGURE 4, rend compte de l'utilisation du temps de travail pour l'ensemble du groupe, que nous avons fixé à environ une heure et demie par jour en semaine normale. Les trois versions se distinguent facilement grâce au taux d'occupation des ressources. Ce dernier est en effet moindre une semaine entre chaque rendu de version, comme précisé dans la SECTION 4.2.

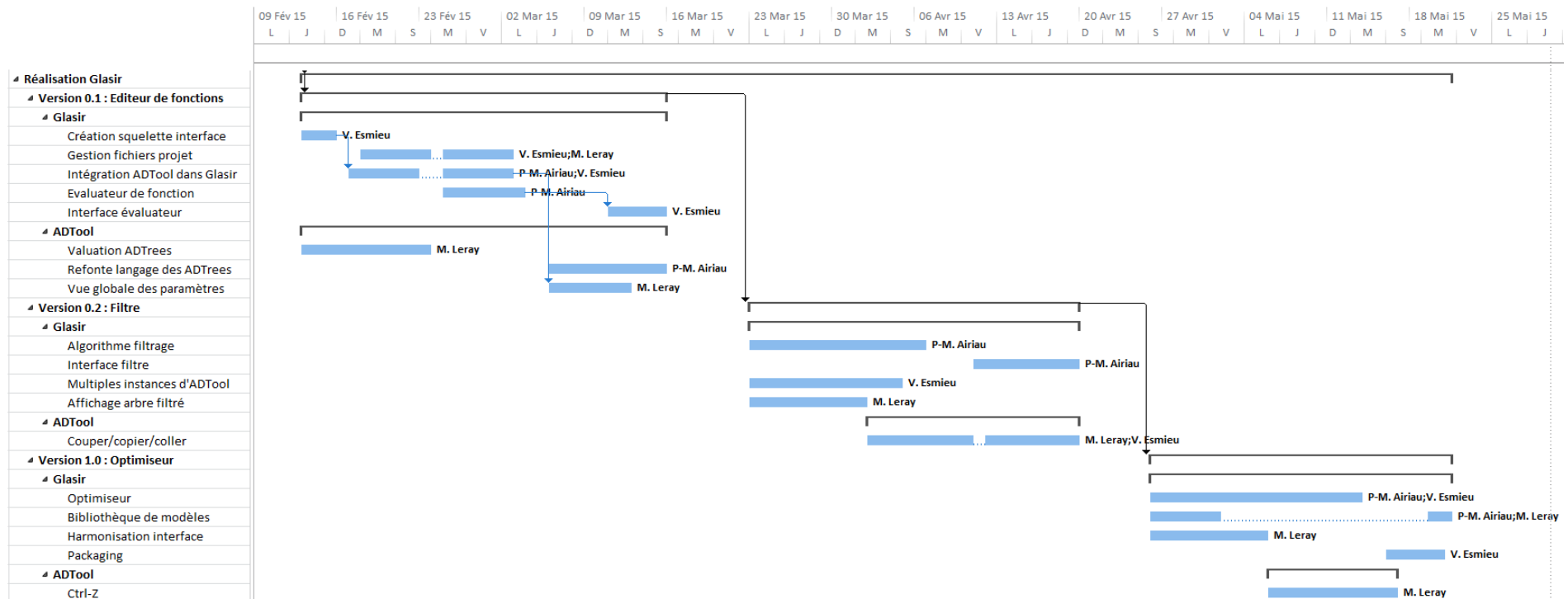


FIGURE 2 – Diagramme de Gantt présentant la chronologie des tâches.

			2 Fév 15			16 Fév 15			02 Mar 15			16 Mar 15			30 Mar 15			13 Avr 15			27 Avr 15			11 Mai 15			25 Mai 15		
Nom de la ressource	Travail	Détails	M	D	V	M	L	S	J	M	D	V	M	L	S	J	M	D	V	M	L	S	J	M	D	V			
▲ P-M. Airiau	102 hr	Trav.			3h	7h	6h	6h	7,5h	7,5h	1h	4,5h	7,5h	7,5h	4,5h	6h	7,5h	1,5h	6h	4h	7,5h	7,5h							
Intégration ADTool dans Glasir	10 hr	Trav.			3h	7h																							
Evaluateur de fonction	12 hr	Trav.					6h	6h																					
Refonte langage des ADTrees	16 hr	Trav.							7,5h	7,5h	1h																		
Algorithme filtrage	24 hr	Trav.										4,5h	7,5h	7,5h	4,5h														
Interface filtre	15 hr	Trav.														6h	7,5h	1,5h											
Optimiseur	15 hr	Trav.																	0h	0h	7,5h	7,5h							
Bibliothèque de modèles	10 hr	Trav.																	6h	4h									
▲ M. Leray	111 hr	Trav.		1,5h	7,5h	7,5h	7,5h	4h	7,5h	4,5h		4,5h	7,5h	7,5h	7,5h	2h			6h	7,5h	7,5h	7,5h	3,5h	10h					
Gestion fichiers projet	10 hr	Trav.			0h	0h	6h	4h																					
Valuation ADTrees	18 hr	Trav.		1,5h	7,5h	7,5h	1,5h																						
Vue globale des paramètres	12 hr	Trav.							7,5h	4,5h																			
Affichage arbre filtré	16 hr	Trav.										4,5h	7,5h	4h															
Couper/copier/coller	13 hr	Trav.												3,5h	7,5h	2h													
Bibliothèque de modèles	10 hr	Trav.																	0h	0h	0h	0h	0h	10h					
Harmonisation interface	16 hr	Trav.																	6h	7,5h	2,5h								
Ctrl-Z	16 hr	Trav.																			5h	7,5h	3,5h						
▲ V. Esmieu	90 hr	Trav.		1,5h	6h	7,5h	7h	4h		7,5h	0,5h	4,5h	7,5h	7,5h	0,5h	4,5h	7,5h	1h	6h	7,5h	1,5h		6h	2h					
Création squelette interface	6 hr	Trav.		1,5h	4,5h																								
Gestion fichiers projet	10 hr	Trav.			1,5h	7,5h	1h																						
Intégration ADTool dans Glasir	10 hr	Trav.			0h	0h	6h	4h																					
Interface évaluateur	8 hr	Trav.								7,5h	0,5h																		
Multiples instances d'ADTool	20 hr	Trav.										4,5h	7,5h	7,5h	0,5h														
Couper/copier/coller	13 hr	Trav.												0h	0h	4,5h	7,5h	1h											
Optimiseur	15 hr	Trav.																	6h	7,5h	1,5h								
Packaging	8 hr	Trav.																			1,5h		6h	2h					

FIGURE 3 – Planning des charges réparties par personne.

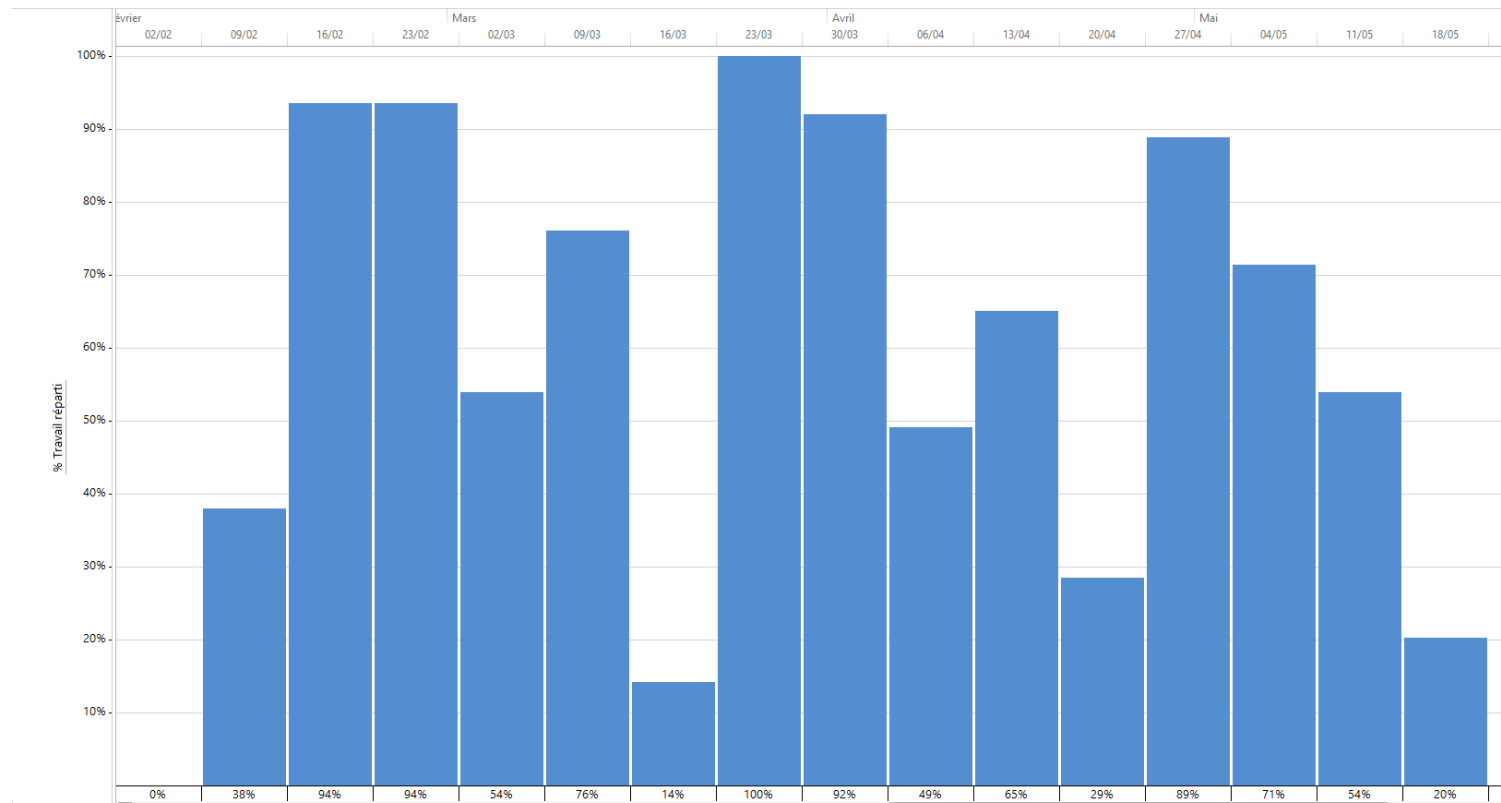


FIGURE 4 – Taux d'utilisation global des ressources.

5 Rétrospective et conclusion

Nous avons commencé par lister les tâches que nous allons devoir réaliser pour que notre projet soit mené à son terme. Nous les avons ensuite regroupées selon la version pour laquelle elles seront effectuées. Le développement de chacune des trois versions définies a quant à lui été divisé en sprints. Ces derniers sont caractéristiques de la méthode SCRUM, méthode agile à laquelle nous avons aussi emprunté le concept de réunion de concertation sur l'avancement, même si dans notre cas elles ne sont qu'hebdomadaires. Nous avons ensuite envisagé les différents obstacles qui pourraient nous retarder, ce qui nous a permis d'estimer une durée nécessaire à chaque tâche en tenant compte de ces risques. Notre planification est le résultat de ces estimations. Elle sera soumise à des changements, mais ceci fait partie intégrante des méthodes agiles. Nous pouvons maintenant commencer la conception de Glasir, dans laquelle seront détaillées son architecture et son interface graphique.

Références

- [1] Barbara Kordy, Piotr Kordy, Sjouke Mauw, and Patrick Schweitzer. ADTool : Security Analysis with Attack-Defense Trees. In Kaustubh R. Joshi, Markus Siegle, Mariëlle Stoelinga, and Pedro R. D'Argenio, editors, *QEST*, volume 8054 of *LNCS*, pages 173–176. Springer, 2013.
- [2] Barbara Kordy, Sjouke Mauw, Saša Radomirović, and Patrick Schweitzer. Attack-defense trees. *Journal of Logic and Computation*, 24(1) :55–87, 2014. Available at <http://logcom.oxfordjournals.org/content/24/1/55.full.pdf+html>.
- [3] Barbara Kordy, Ludovic Piètre-Cambacédès, and Patrick Schweitzer. DAG-Based Attack and Defense Modeling : Don't Miss the Forest for the Attack Trees. *Computer Science Review*, 2014. DOI : 10.1016/j.cosrev.2014.07.001.
- [4] 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



Cti
Commission
des Titres d'Ingénieur

