Équipe FACT
Conception et développement d'applications

FACT Development

http://fact-team.github.io

UNIVERSITÉ TOULOUSE III PAUL SABATIER

Rédigé avec LaTeX
Version du 18 avril 2015

Bilan du Projet

FactDev

Université Toulouse III – Paul Sabatier

— **F**lorent Berbie
— **A**ntoine de Roquemaurel
— **C**édric Rohaut
— Andriamihary Manan**T**soa Razanajatovo

Rédigé par
Équipe FACT

Approuvé par
Frédéric Migeon

18 avril 2015

Table des matières

Introduction

FactDev est un logiciel de devis et de facturation réalisé dans le cadre de l'UE Projet.

Ce projet s'est fait en réponse à un problème de l'un des membres du groupe : Antoine De Roquemaurel. En effet Antoine, développeur *Freelance*, rédigeait pour ses clients les factures et devis « à la main ». La tâche était répétitive et le risque d'erreurs humaine important :

— erreur dans le calcul des montants
— perte de facture
— plusieurs factures différentes pour un même projet et donc risque de ne pas faire le bon travail demandé par le client

Face à ces difficultés, Antoine a soumis le projet devant répondre aux spécifications suivantes :

— Gestion des clients
— Gestion des projets associées aux clients
— Calculs des tarifs
— Génération de documents
— Recherche

Les autres membres ont vu dans ce projet l'opportunité de développer de nouvelles compétences, aussi bien sur le plan technique qu'organisationnel. D'un point de vue technique avec l'utilisation du *LaTeX* et du *C++* accompagné du *framework Qt*. Sur le plan organisationnel avec la mise en place de la méthode Agile *Scrum*.

Ce document fait état des parties prenantes, des outils et des méthodologies de développement appliquées au projet. Il fait le point sur les résultats obtenus au niveau de la méthodologie, du logiciel et en terme de respect du plan qualité.

1 Le projet

1.1 Le logiciel FactDev

Le logiciel FactDev a pour but de faciliter la création de devis et la conversion de ces devis en factures.

Il répond aux exigences définies lors de l'introduction en permettant l'enregistrement d'un nouveau client dans la base de données et de projets associés à ce clients. Par exemple, un client X peut demander la réalisation d'un logiciel pour le management de son entreprise et un site web pour promouvoir son activité. On a donc deux projets distincts qui peuvent cependant faire l'objet d'une seule facture. De plus, une facture est fréquemment précédé d'un devis, c'est pourquoi il doit être facile de transformer un devis en facture. Enfin le logiciel devra tenir de certaine réglementation «légale » tel que l'impossibilité de modifier une facture ayant été payée.

1.1.1 Présentation des Parties Prenantes

1.1.1.1 Client : Antoine de Roquemaurel

Développeur Freelance et membre de l'équipe de développement.

☎ 06 54 33 52 93
https://antoinederoquemaurel.github.io
✉ antoine.roquemaurel@gmail.com

1.1.1.2 Encadrant : Fédéric Migeon

Maître de conférence à l'Université Toulouse III – Paul Sabatier

☎ 05 61 55 (62 46)
✉ Frederic.Migeon@irit.fr
IRIT1 / Niveau 3, Pièce : 361

1.1.1.3 Responsable de l'UE Projet : Bernard Cherbonneau

☎ 05 61 55 (63 52)
✉ Bernard.Cherbonneau@irit.fr

IRIT1 / Niveau 4, Pièce : 413

1.1.1.4 Titulaire : Équipe FACT

Étudiants en M1 Informatique Développement Logiciel à l'Université Toulouse III – Paul Sabatier

Florent Berbie
☎ 06 85 31 92 90
✉ florent.berbie@gmail.com

Antoine de Roquemaurel
☎ 06 54 33 52 93
✉ antoine.roquemaurel@gmail.com

Cédric Rohaut
☎ 06 74 80 12 67
✉ rohaut@icloud.com

Manantsoa Andriamihary Razanajatovo
☎ 06 01 71 53 02
✉ manantsoa.razana@gmail.com

1.2 Les outils

2 La méthodologie

2.1 La méthode Scrum

2.2 L'intégration continue

2.3 La revue de code

La revue de code représente une démarche que nous avions mis en avant dans le plan qualité. L'objectif visé est de tendre vers un projet dont l'intégralité du code a été revu.

La revue de code se fait au moment de l'intégration, c'est pourquoi toutes intégrations nécessitent la création préalable d'une *Pull Request*. Pour cela, une fois le travail correspondant à une *User story* est fini [1], le développeur crée une *Pull Request* via l'outil *Github*. Cette *Pull Requests*'accompagne d'une description plus technique de la *User story* à laquelle elle est liée. La liste des *commits* associés et le code source ajouté et/ou modifié est accessible comme l'on peut le constater ci-dessous.

FIGURE 2.1 – Exemple de *Pull Request* du projet *FactDev*

1. selon nos critères définis lors de la présentation de la méthode *Scrum* 2.1

À partir de cette *Pull Request*, les autres membres de l'équipe reçoivent une notification pour indiquer qu'ils doivent procéder à la revue de code. Au moins l'un deux doit s'assurer que le code est valide. Un code est dit valide lorsqu'il :

est lisible Le code doit être facile à lire.

est compréhensible Le code doit être facilement compréhensible, avoir un niveau de complexité minimum.

respecte les conventions d'écritures Respect des conventions d'écritures (convention de nommage et de mise en forme).

Cette vérification est aisé via l'outil *Github* qui permet de rajouter des commentaires *« inline »* c'est-à-dire d'ajouter un commentaire aux lignes précise de code à modifier. Ainsi, sans toucher au code, l'on sait précisément l'endroit où l'on doit procéder à des changements.

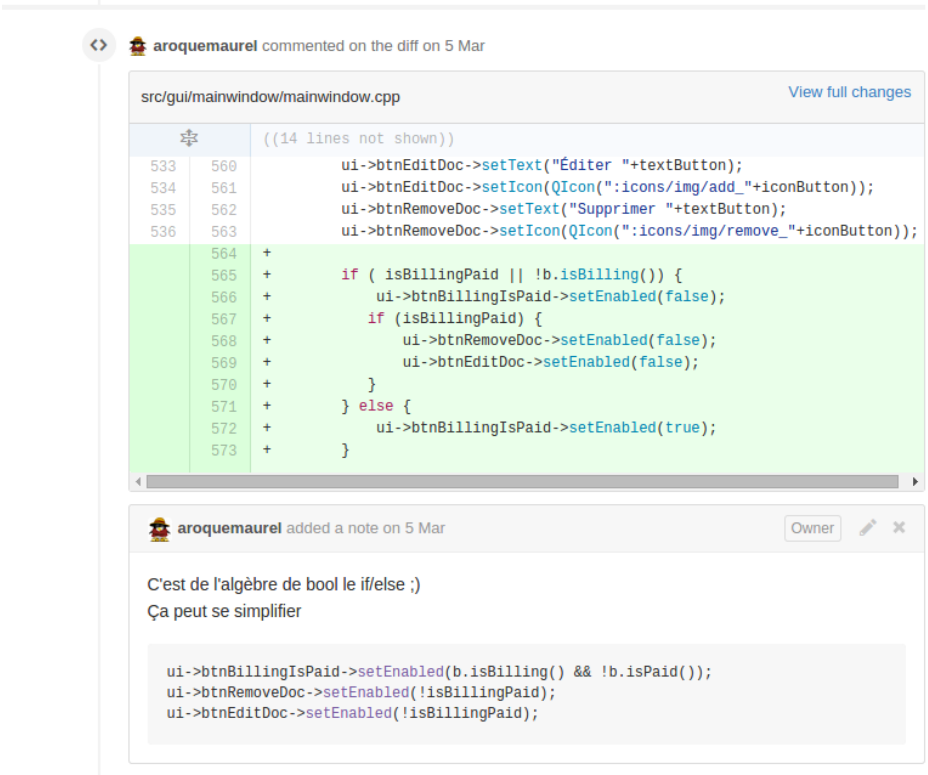

Figure 2.2 – Ajout d'un commentaire « inline » lors d'une *Pull Request*

L'on procède également à la vérification de la documentation. Chaque méthode et attribut doit être documenté. Là aussi, il faut que la documentation respecte les conventions d'écritures.

Une fois les remarques faites sur le code et sa documentation, l'on passe aux tests fonctionnels. On se rend donc sur la branche *Git*en question et l'on vérifie que la fonction répond bien à la *User story*et qu'il n'existe aucun bug. Bien entendu, on s'assure que ça n'a pas entraîné de régression sur d'autres parties du logiciel. C'est aussi le moment de proposer des modifications sur le plan ergonomique si besoin est.

3 Les résultats

L'UE projet était l'occasion de se positionner dans une situation très proche de celle que allons rencontré en milieu professionnel. Dans ce cadre là, il était nécessaire de prendre des mesures organisationnel afin de parvenir aux résultats escomptés.

3.1 La méthodologie

La qualité du logiciel *FactDev* est principalement dû à la rigueur dans l'application de la méthodologie de développement.

3.1.1 Le respect de la méthode Scrum

Au niveau de la gestion du développement par la méthode *Scrum* nous avons tenu un *Backlog* A. Dans celui-ci se trouve l'ensemble des *User stories* et *Technical stories* que nous avions à réaliser durant le projet. Ces *Stories* ont été déterminée lors de *Planning Poker*durant les mêlées. C'est durant ces séances que nous déterminions le poids attribué à chacune des *Stories* et comment les répartir sur les six *Sprints* des deux *Releases*.

Lorsque nous ne pouvions nous voir pour assurer les mêlées quotidiennes, nous nous retrouvions sur une zone de chat *« #irc »* que nous avions créé pour la circonstance. Ainsi, nous faisions le point sur l'évolution du projet et débattre sur des choix de conception.

Le *BurnUp Chart* ci-dessous montre l'assiduité et le respect de la méthode *Scrum* durant le projet.

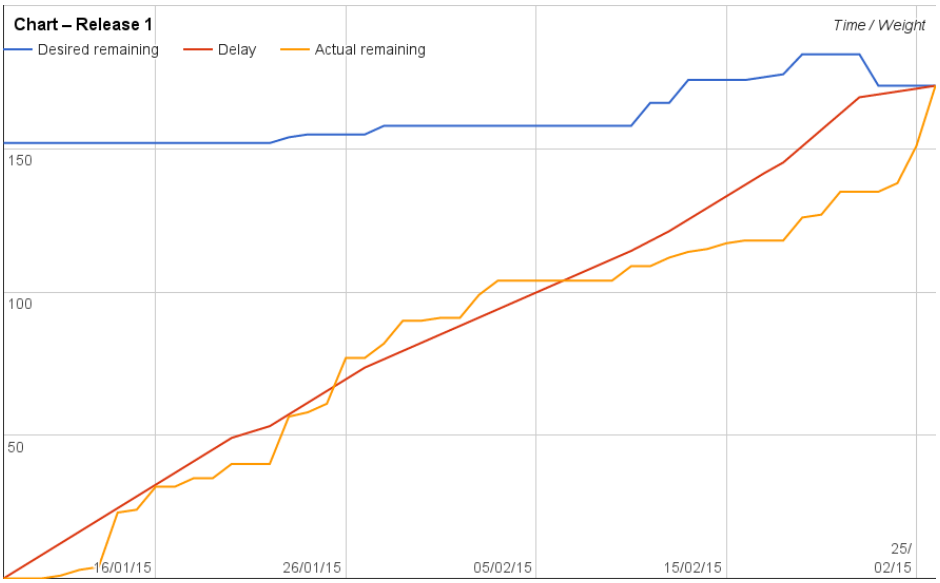

Figure 3.1 – Burn Up chart de la première release

La courbe rouge représente l'évolution théorique du projet durant les *Sprints* de la première *Release*. On constate que notre avancé (courbe orange) suit globalement la courbe théorique. Cependant, l'on observe une baisse de notre activité durant la semaine du 5 février qui correspondait à notre période de vacance. De plus, on note également que lors du *Sprint* 3 nous nous sommes trouvés en dessous de l'objectif. Cela s'explique par une augmentation du nombre de *Stories* (courbe bleu) qui correspondait à des modifications ergonomiques où à la résolution de bugs. Néanmoins, nous sommes parvenus à finir nos *Sprints* dans les temps.

Chacun des *Sprints* a été réalisé dans les temps et fut l'objet d'une démonstration auprès de M. Migeon.

Dans l'ensemble la courbe suit bien la courbe théorique malgré une tendance en dessous du délai attendu. La coupure du milieu correspond à la semaine de vacances ou nous avons effectué une pause, sur la fin nous avons eu un retard à combler à cause de l'augmentation des stories liée aux bugs. Nous avons du transferérer quelques unes de ces dernières à la release d'après.

Une des manières la plus simple pour savoir si la méthodologie choisie correspondait bien aux résultats attendus et était bien adaptée est d'analyser les chiffres et les résultats que nous avons obtenus.

La méthode agile *Scrum* que nous avons choisie nous a permis de bien répartir le travail en fonction de la disponibilité de chacun et de ses compétences techniques. En effet avec une moyenne de 10 *User stories* par *Sprint* sur un total de 54, la durée des sprints (2 semaines) était bien adaptée à notre cycle de développement. Nous avons réalisé 6 sprints avec deux *Releases*. Cette volonté de découper le travail par tranche de deux semaines était bénéfique dans le sens ou nous avons pu nous adapter aux attentes et aux évolutions des besoins du client. Nous reviendrons sur l'organisation plus bas avec le burnup chart.

Par ailleurs, la couverture des tests qui est aux alentours de 90% sur la fin du projet nous a permis d'éviter un grand nombre de bugs ainsi que d'en trouver certains qui n'auraient pas forcément été détectés si il n'y avait pas eu de tests. Cependant les bugs par rapport à l'interface (clic sur un bouton qui provoque un plantage de l'application) n'ont pas pu être testé et nous avons du procéder à des tests manuels.

Enfin l'utilisation de *Github* en étroite collaboration avec la méthode agile *Scrum* a permis de pouvoir travailler depuis chez soit ce qui était indispensable dans ce contexte (nous ne pouvions programmer des mêlées tous les jours à cause des cours, etc). Le nombre de *Pull Request* (95), issues (145), branches (91) et commits (1200) en est une preuve ainsi que la preuve de toute la communication que nous avons mis en place pour gérer ce projet.

3.2 Le logiciel

Conclusion

Nous tenons avant tout à remercier toutes les parties prenantes de ce projet qui nous ont permis de le mener à bien que se soit l'équipe pédagogique de l'université, le client ou l'équipe de développement.

Le projet FactDev nous a en effet permis de pouvoir mener un projet comme dans le cadre d'une entreprise dans le monde professionnel. Malgré les difficultés que nous avons pu rencontrer comme la difficulté de communiquer, d'exprimer nos points de vues, nos problèmes, des fonctionnalités que nous n'avons pu implanter, de nombreux points positifs en ressortent.

Au delà de l'expérience professionnelle que nous avons acquise grâce à la pédagogie mise en place et du plus dans nos CV, plusieurs points bénéfiques en ressortent. Tout d'abord le logiciel sera sous licence GPL, il sera utilisable et utilisé par son client et éventuellement d'autres personnes dans le futur. Au niveau technique, une multitude de compétences ont été acquises, les principales sont le langage C++ ainsi que les méthodes agiles mais d'autres moins principales mais tout aussi nécessaires ont été vues comme la gestion du projet avec GitHub ou encore les tests avec Travis et Coveralls.

C'est pourquoi cette pédagogie innovante qui nous permet d'acquérir une certaine autonomie est nécessaire dans un monde qui évolue tout les jours notamment celui du numérique avec ses innovations.

A Backlog product

Release	Sprint	Statut	Vélocité Poids	Vélocité Nb Storie	Poids	Prio	#Issue	Type Story	En tant que	Je Souhaite	Afin de
		Done			2	Must	1	User Story	Utilisateur	Créer un nouveau client	Ajouter ses informations et prochianement le lier à un devi
		Done			3	Should	2	User Story	Utilisateur	Editer un client	Modifier les informations d'un client
		Done			2	Would	3	User Story	Utilisateur	Supprimer un client	Corriger une erreur, un client ajouté par mégarde
		Done			5	Must	11	User Story	Utilisateur	Afficher la liste des clients	Pouvoir prochainement afficher leurs factures
		Done			5	Must	12	User Story	Utilisateur	Renseigner mes données	
		Done			8	Could	14	User Story	Utilisateur	Afficher les informations d'un client particulier	Obtenir des informations détaillé sur le client
		Done			5	Could	20	User Story	Utilisateur	Chercher un client à partir de son nom	Filtrer l'affichage des clients
		Done			1	Would	22	User Story	Utilisateur	M'informer sur le développement du projet	
		Done			1	Must	16	Technical Story	-	Création de la fenêtre d'ajout/modification d'un client	
	Sprint 1	Done	40	10	8	Would	15	User Story	Utilisateur	Afficher un menu contextuel	*Editer, Ajouter, Supprimer...*
		Done			5	Must	13	User Story	Utilisateur	Créer un nouveau projet pour un client	pouvoir l'associé à un client
		Done			13	Must	7	User Story	Utilisateur	Créer un nouveau devis	estimer le coût d'un développement
		Done			5	Must	30	User Story	Utilisateur	Afficher les devis d'un projet	Avoir une estimation du coût du projet
		Done			8	Must	44	Technical Story	-	Récupérer un devis dans la bd avec son id	
		Done			1	Should	24	User Story	Utilisateur	Editer un projet existant	modifier ses informations
		Done			3	Must	25	User Story	Utilisateur	Lister les projets d'un client	Afficher la liste des projets d'un client particulier
		Done			5	Must	34	Technical Story	-	User Manual	
		Done			8	Must	37	User Story	Utilisateur	Créer une prestation	
		Done			8	Would	47	Technical Story		Sécurité des champs	
		Done			0.5	Could	50	Anomalie		Titres de fenêtres	
		Done			0.5	Could	49	Anomalie		Noms docks	
		Done			0.5	Could	48	Anomalie		Affichage Fax	
		Done			0.5	Must	52	Technical Story		créer un jeu d'essais de devis et projets	
		Done			3	Must	65	Technical Story		Insérer dans tripatte	
	Sprint 2	Done	64	15	3	Should	26	User Story	Utilisateur	Supprimer le projet d'un client	
		TODO			1	Must	8	User Story	Utilisateur	Créer une nouvelle facture	
		TODO			1	Must	39	User Story	Utilisateur	Afficher les factures d'un projet	
		TODO			1	Must	41	User Story	Utilisateur	Editer une facture existante	
		TODO			2	Should	45	User Story	Utilisateur	Signaler une facture comme payée	
		TODO			3	Could	27	User Story	Utilisateur	Editer un devis existant	le générer de nouveau
		TODO			3	Would	26	User Story	Utilisateur	Supprimer un devis	supprimer un devis d'un client
		TODO			3		40	Technical Story	-	User Manual	
		TODO			5		42	Technical Story	-	Plan Qualité	
		TODO			13		43	Technical Story	-	Préparer soutenance	
		TODO			5	Would	33	User Story	Utilisateur	Rechercher un devis ou une facture	Accéder facilement un devis ou une facture spécifique
		TODO			2	Could	35	User Story	Utilisateur	Editer une prestation	Pouvoir l'ajouter à un devis ou une facture
		TODO			8	Could		Technical Story		Sécurité des champs	
					5	Should	28	User Story	Utilisateur	Générer le fichier .tex d'un devis	pouvoir l'avoir ensuite en .pdf
Release 1	**Sprint 3**	TODO	59	13	20		38	Technical Story	-	Déploiement	
Totaux	3		163	38	163						
Moyennes			52	14	4.35135						

B Table des figures