



L'ALM avec Microsoft Team Foundation Server 2012 :

L'ingrédient clé pour la réussite de vos projets
informatiques





**L'ALM avec
Microsoft Team Foundation Server 2012 :
L'ingrédient clé pour la réussite de
vos projets informatiques**

Sommaire

1. AVERTISSEMENT	4
2. INTRODUCTION	5
3. PROJETS INFORMATIQUES : FAITS ET PROBLEMATIQUES	6
3.1. Quelques Chiffres	6
3.2. Les Principales causes d'échecs	7
3.2.1. Gestion de projet lacunaire	8
3.2.2. Spécifications incomplètes ou surréalistes	9
3.2.3. Manque de communication, information disséminée.....	10
3.2.4. Manque de réactivité face au changement	11
3.2.5. Tests et Qualité insuffisante.....	12
3.2.6. Mauvaise gestion des risques	13
3.3. Les Projets informatiques : toujours et encore le même scénario?	14
3.4. Les Notions Fondamentales	15
4. PREMIERS ELEMENTS DE REPONSE : METHODOLOGIE ET GESTION DE PROJET	18
4.1. Répartitions Statistiques des méthodologies utilisées	18
4.2. L'approche Agile.....	19
4.3. L'approche CMMI / PMI avec le Cycle en V.....	23
4.4. Comparatif des deux approches	25
5. L'ALM : LA SOLUTION ET LES OUTILS	26
5.1. L'ALM : Pourquoi et quels bénéfices ?	26
5.2. Démarrer avec l'ALM	29
6. LA SOLUTION ALM DE MICROSOFT : VISUAL STUDIO 2012	30
6.1. Architecture Technique et Fonctionnelle de TFS 2012	30
6.2. TFS 2012 Extensibilité et ouverture.....	65
7. CONCLUSION	66

1. Avertissement

Ce document est fourni uniquement à titre indicatif. MICROSOFT N'APPORTE AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, À CE DOCUMENT. Les informations figurant dans ce document, notamment les URL et les références aux sites Internet, peuvent être modifiées sans préavis. Les risques d'utiliser ce document ou ses résultats sont entièrement à la charge de l'utilisateur. Sauf indication contraire, les sociétés, les entreprises, les produits, les noms de domaine, les adresses électroniques, les logos, les personnes, les lieux et les événements utilisés dans ce document sont fictifs. Toute ressemblance avec des entreprises, noms d'entreprise, produits, noms de domaine, adresses électroniques, logos, personnes ou événements réels serait purement fortuite et involontaire.

Remerciements

Benoit Launay, Chef de produit Visual Studio chez Microsoft France, <http://visualstudio.fr>

Auteur



Philippe PUSCHMANN est Architecte Solution chez Avanade France. Membre de l'équipe CTO, Il est en charge du pôle ALM Avanade France, et assure également la fonction de Solution Delivery Manager. Philippe a développé une expertise sur l'ensemble des modules de l'offre TFS et s'est spécialisé sur la partie Méthodologie, Gestion de Projet, Agilité et Gestion de la Qualité. Il a également acquis une forte expérience terrain dans l'utilisation de la plateforme ALM de Microsoft ayant entre autre délivré un projet onshore de 2500 jours hommes sur une durée de 6 mois avec une équipe de 25 personnes en utilisant l'ensemble des fonctions de la plateforme TFS, Philippe PUSCHMANN est MVP Visual Studio 2012 et certifié PMP.



2. Introduction

Le déroulement incertain des projets informatiques, quelle que soit la méthodologie utilisée, que ce soit sur les aspects délai, budget, périmètre, qualité ou encore satisfaction du client, serait-il un fait avéré, une tautologie ?

Nous allons tenter, au travers de ce livre blanc, d'apporter un éclairage sur cette préoccupation majeure pour l'ensemble des acteurs participant au développement de solutions informatiques aujourd'hui. Malgré la multiplication et l'avancée des outils, des méthodologies et des technologies, la situation semble inébranlable, et pourtant des solutions existent, des solutions capables d'outiller et solutionner l'ensemble des problématiques abordées dans le cadre d'un projet informatique, ce panel de solution porte un nom : ALM (Application Lifecycle Management).

L'ALM serait-elle la clé à tous ces problèmes ? De nombreux retours d'expérience semblent attester ce fait, et nous allons essayer dans le cadre de ce livre blanc d'ouvrir des pistes apportant des solutions à la plupart des problèmes et difficultés rencontrés sur les projets aujourd'hui. Dans un premier temps, nous identifierons les problématiques à résoudre, puis nous revisiterons les solutions potentielles avant de définir ce qu'est l'ALM et comment Microsoft a réussi à outiller une solution ALM de premier rang comme facteur clé de succès des projets informatiques.

3. Projets Informatiques : Faits et Problématiques

3.1. Quelques Chiffres

Le premier artefact à regarder lorsque l'on aborde le domaine des projets informatiques est le CHAOS Report du Standish Group. Le Standish Group collecte de l'information et établit des statistiques sur les taux de réussite / échecs projets dans l'industrie des technologies de l'information dans le but de tracer les tendances et trouver des axes d'amélioration du taux de réussite et accroître la valeur et le ROI des investissements IT.

Chaque année, le Standish Group remet à jour le CHAOS Report et publie les statistiques en termes de résultats des projets IT en les classant dans 3 catégories : Les projets réussis, c'est-à-dire livrés dans les délais, le budget et au niveau de qualité requis, les projets « challengés » généralement sur un des éléments de la triple contrainte : Coûts, délais, périmètre et qualité et enfin les projets en échecs, ayant fait l'objet d'un No Go par exemple.

Le graphe à fin 2009 se présente comme suit :

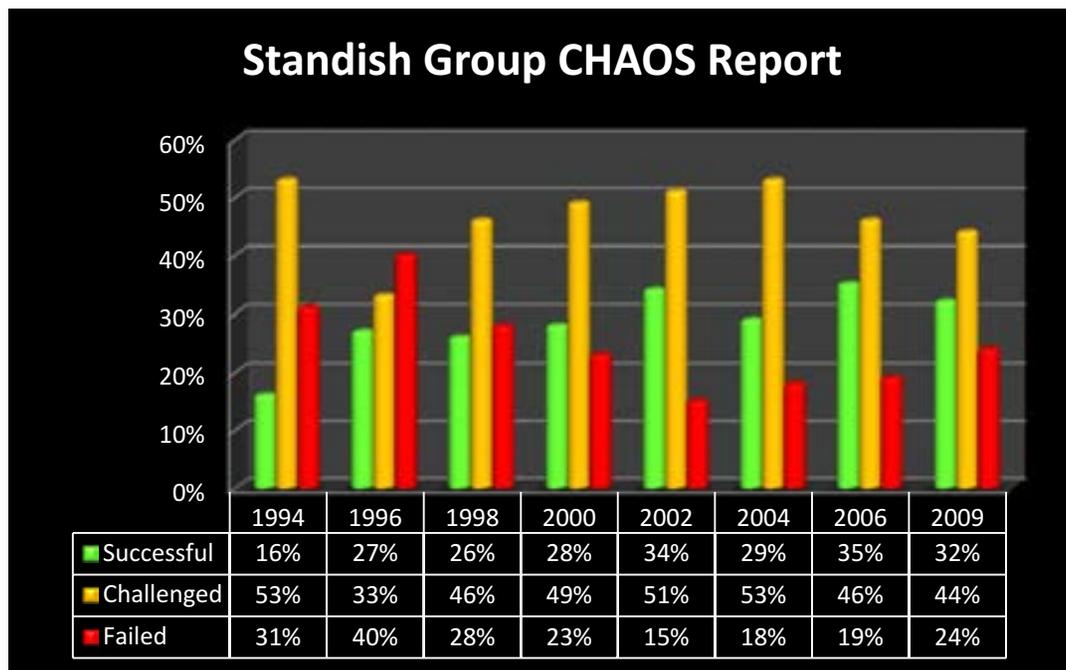


Figure 1 : Taux de réussite des projets informatiques sur 15 ans (Chaos Report Standish Group)

Une analyse rapide de ce graphique permet de dégager 2 informations clés :

- La tendance reste stable au cours du temps malgré les progrès technologiques.
- Une équipe projet qui démarre un projet aujourd'hui a 1 chance sur 3 de voir son projet se dérouler dans les meilleures conditions et 2 chances sur 3 pour que ça ne se passe pas bien à savoir des tensions sur les délais, le budget, le scope ou la qualité, voire pire un arrêt du projet par le client.

Un autre graphique du même organisme retient également toute notre attention. Ce graphique précise qu'en moyenne, 45% des fonctionnalités développées d'une application ne sont jamais utilisées. Ce qui signifie que des efforts considérables sont investis dans des fonctionnalités à faible valeur ajoutées alors que l'effort pourrait être concentré sur les fonctionnalités les plus utiles pour le métier du client.

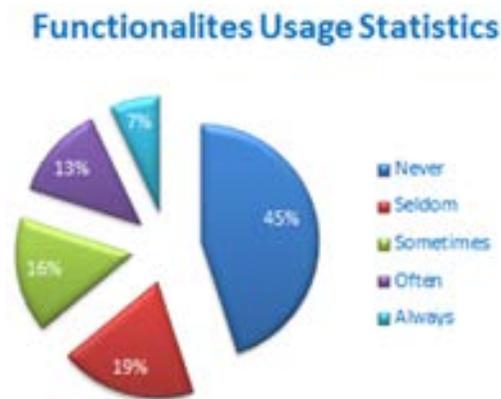


Figure 2 : % d'utilisation des fonctionnalités développées d'une solution (Source Standish Group)

3.2. Les Principales causes d'échecs

Dans un premier temps nous allons essayer de déterminer les raisons majeures qui font que la situation ne s'améliore pas.

On a tout d'abord le fait que les besoins des clients, généralement de grandes entreprises, sont toujours plus complexes, avec des contraintes en termes de Time to Market toujours plus agressives. Aujourd'hui, un client ne peut plus dans le cadre de la compétitivité mondiale se satisfaire de projets pluri annuels, il faut rajouter à cela, l'évolution incessante et très rapide des technologies, dont l'obsolescence ne fait que renforcer le point précédent. On a également la complexité du métier dont font l'objet les projets, les entreprises ont pour objectif d'améliorer leur compétitivité en s'appuyant notamment sur des « Core Solutions », dont le dessein reste d'outiller cette compétitivité métier au travers de solutions informatiques souvent très complexes devant à la fois exécuter un métier complexe avec de fortes contraintes en terme de performances pour supporter un business sans cesse

croissant.

D'un autre côté, nous avons les intégrateurs qui développent ces solutions, souvent en utilisant les dernières technologies sur lesquelles la plupart des développeurs ne sont pas suffisamment expérimentés avec des courbes d'apprentissage qui seront très rapides, à cela on ajoutera Framework et Patterns complexes pour ajouter sur le tout, une surcouche de complexité dont tout le monde se serait bien passé.

Ceci étant dit, l'exercice qu'il est intéressant de mener est ce que l'on appelle une « Root Cause Analysis ». L'idée ici étant de déterminer quels sont les facteurs qui vont contribuer le plus fortement au fait que les projets soient en difficulté.

Tout le monde connaît ces principales causes d'échecs, simplement, la difficulté va consister à les prioriser, nous allons donc proposer une liste :

1. Gestion de projet lacunaire
2. Spécifications incomplètes ou surréalistes
3. Manque de communication, information disséminée
4. Manque de réactivité face au changement
5. Mauvaise gestion des risques
6. Tests et Qualité Insuffisante

3.2.1. Gestion de projet lacunaire

La gestion de projet reste un élément clé de réussite. Avant de regarder pourquoi, la gestion de projet a tant d'importance, nous allons rapidement redéfinir ce qu'est la gestion de projet en nous appuyant sur un Framework de référence mondiale : le PMI (Project Management Institute).

Le Project Management Institute, fondé en 1969, est une association professionnelle à but non lucratif qui propose des méthodes de Gestion de projet. Son siège est à Philadelphie en Pennsylvanie (États-Unis), elle compte plus de 200 000 membres répartis dans 125 pays. Elle publie des standards relatifs à la gestion de projet et est en charge de la certification des processus de gestion de projet (la fameuse certification PMP). Le PMI met en exergue au travers d'un ouvrage incontournable, le PMBOK (Project Management Book of Knowledge), les bonnes pratiques en termes de gestion de projet. Au sens du PMBOK, il y a 9 zones de connaissances composées de 42 processus qui sont utilisées au cours de 5 phases d'un projet.

Les neuf zones de connaissances sont :



Les 5 Phases sont :



Sans entrer dans les détails, on remarque rapidement que ce Framework encapsule l'intégralité des concepts, processus et tâches nécessaires à la bonne conduite d'un projet avec notamment une gestion approfondie des éléments de la triple contrainte : Délai, Budget, Scope et Qualité tout au long du cycle de vie du projet.

Le constat est simple, il suffit de regarder sur un projet le niveau de maturité en termes de gestion de projet pour savoir si ce projet est à risque ou non !

La gestion de projet va permettre de contrôler simplement que l'exécution d'un projet est conforme aux attentes d'un client en se basant sur un certains nombres d'indicateurs et artefacts projets qui seront définis au démarrage du projet avec le client et ce que l'exécution soit faite en mode Agile, Sacrum ou Cycle en V.

3.2.2. Spécifications incomplètes ou surréalistes

La définition et gestion des spécifications constitue indéniablement une des principales

sources de problèmes sur les projets lorsque l'on interroge des parties prenantes. Tout le monde le sait et pourtant la situation continue, et quand bien même ce facteur est inclus dans une des zones de connaissances du PMI, Scope Management ou Gestion du Périmètre, il est tellement impactant qu'il est important de le faire ressortir. Démarrer un projet avec un périmètre insuffisamment défini est une ineptie qui va précipiter l'équipe en charge du projet dans des difficultés de tout instant. Attention à ne pas confondre Définition du besoin haut niveau avec Spécifications détaillées, en effet, il est également aberrant de démarrer un projet avec un trop haut niveau de détail, car comme on le sait, tout sera sujet à changement et il est préférable d'aborder le sujet en s'appuyant sur la technique connue sous le nom : Rolling Wave Planning ou élaboration progressive qui est une approche **Agile**.

3.2.3. Manque de communication, information disséminée

Pour faire simple, la communication est la clé de voûte d'un projet informatique, pour comprendre comment ce point est fondamental en projet, on parle souvent du syndrome de la Tour de Babel. La Tour de Babel a été un des premiers grands projets de l'humanité ayant subi un échec, il y a près de 5000 ans. Dans le cadre de ce projet les hommes voulaient construire une tour pour atteindre le ciel :



Ce projet, surréaliste au demeurant, a échoué à partir du moment où les bâtisseurs se mirent à parler des langues différentes, la leçon de cette histoire nous a démontré la nécessité qu'ont les parties prenantes de se parler, de se comprendre pour réaliser des grands projets,

mais aussi le risque de voir échouer ces projets quand chaque groupe de spécialistes se met à parler le seul jargon de sa discipline.

Dans nos projets de tous les jours, le manque de communication et de synchronisation entre les sponsors métiers, les utilisateurs finaux et les experts techniques sont souvent à la source de difficultés projets quand bien même les projets sont parfaitement bien gérés.

Le manque de visibilité, prédictibilité ou effet tunnel qui découlent d'une communication lacunaire, sont autant de difficultés pouvant inclure l'impossibilité de savoir qui est en charge, ou responsable de quoi, de même que l'incapacité à obtenir l'engagement des parties prenantes afin de produire des estimations précises et / ou procéder à des ajustements de planning ou encore des acceptances formelles.

3.2.4. Manque de réactivité face au changement

Entre le moment où un projet démarre et sa livraison, beaucoup de changements vont intervenir, c'est inévitable. Tout changement doit au contraire être accueilli favorablement comme avantage compétitif pour le client. Cet état de fait est particulièrement impactant dans des contrats de type forfait dans lesquels les intégrateurs s'engagent à délivrer un périmètre en s'appuyant sur le fameux cycle en V voire Waterfall (Cascade) qui rendent encore plus compliqué tout changement du fait de la lourdeur des processus. Pourtant la résistance au changement peut sembler légitime, en effet si l'on souhaite maîtriser le changement sur tout projet, une priorité sera sa traçabilité et si l'on se réfère au PMBOK, le premier réflexe dans le cadre d'une demande de changement est une analyse d'impact donc le processus se présente globalement comme suit :

- Evaluation de l'Impact sur le périmètre, exigence impactée
- Evaluation de l'Impact sur les délais et le budget
- Evaluation de l'Impact sur l'architecture
- Evaluation de l'Impact sur la qualité
- Evaluation de l'Impact sur les risques
- Evaluation de l'Impact sur l'expérience utilisateur
- Evaluation de l'Impact sur la documentation
- Justification du besoin métier
- Etc.

On se rend vite compte que cette bonne pratique si elle n'est pas d'une part bien outillée et exécutée va constituer un frein au changement avec par exemple des renégociations de contrats dans les cas de figure les plus courants.

De plus tout changement acté devra être réintégré dans les différents environnements d'in-

tégration, pré-production, production, et donc tout environnement projet n'intégrant pas une usine logicielle digne de ce nom rencontrera là encore des difficultés.

Nous verrons comment le principe d'usine logicielle est crucial pour tout ce qui concerne la gestion du changement, sa traçabilité et la réactivité au travers du principe de l'intégration continue.

Il reste à noter un élément fondamental : Tout changement doit être accueilli favorablement, comme avantage compétitif pour le client, et ceci sera un facteur clé de réussite du projet qui se résume dans la satisfaction du client (sponsor et utilisateurs finaux).

3.2.5. Tests et Qualité insuffisante

De l'importance des Tests et de la qualité : Il faut bien l'avouer, dans le domaine des projets informatiques, la qualité reste le parent pauvre, si l'on se référence à l'industrie par exemple, ou les tests avant mise en production constituent une activité de premier plan. Malheureusement, il arrive parfois que les projets informatiques fassent partie de projets industriels tel que dans l'automobile, l'aéronautique ou encore l'aérospatiale. L'exemple ci-dessous illustre parfaitement les conséquences catastrophiques d'une gestion lacunaire de la qualité, lorsque par exemple les campagnes de tests sont insuffisantes pour cause de coupes budgétaires.

Le bug informatique le plus onéreux de l'histoire

Le 4 juin 1996, a eu lieu le vol 501, vol inaugural de la fusée Ariane 5. Ce lancement s'est soldé par un échec. La fusée s'est brisée et a explosé en vol à 4000 mètres au-dessus du centre spatial de Kourou, 40 secondes après le décollage, à la suite d'une panne du système de navigation.



Figure 3 : Vol 501 - Ariane 5

Coût de l'opération : 370 Millions

L'incident, dû à un bug dans les appareils informatiques utilisés par le pilote automatique, qui a provoqué la destruction de la fusée ainsi que la charge utile – 4 sondes de la mission Cluster – d'une valeur totale de 370 millions de dollars, ce qui en fait le bug informatique le plus coûteux de l'histoire.

Dans le rapport de la commission d'enquête, les points suivants ont été soulevés :

Des simulations en laboratoire ont en principe lieu avant le décollage. La réussite des simulations est nécessaire pour obtenir le certificat de vol. Le système de navigation, utilisé depuis longtemps sur Ariane 4, étant réputé fiable, le Centre national d'études spatiales a demandé de ne pas faire de simulations de vol pour ces appareils, et ainsi économiser 800 000 frs sur le coût des préparatifs.

Réalisées après la catastrophe, les simulations en laboratoire ont permis de vérifier que l'explosion était inéluctable.

Ce bug spectaculaire met en exergue l'importance des tests permettant de valider une solution avant sa mise en production, sachant qu'après la mise en production, les corrections coutent considérablement plus cher, ce fait historique nous le rappelle !

La leçon de cette histoire est que les tests doivent faire partie intégrante des activités du cycle de vie logicielle et adressés avec le même niveau d'expertise et d'exigence que celui mis en œuvre dans l'implémentation. Hors, fort est de constater qu'en France, contrairement aux pays anglo-saxons ou encore l'Allemagne, le profil du testeur est rarement de type Testeur professionnel certifié CFTL (Comité Français des Tests Logiciels) par exemple, mais plutôt le consultant qui a un peu de temps libre et qui passait par là !

3.2.6. Mauvaise gestion des risques

Qui gère ses risques sur un projet ? A cette question suit inéluctablement une réponse évasive. Ceci est pourtant une erreur stratégique. Pourtant, chaque membre de l'équipe projet est conscient des risques potentiels et pourtant, ils ne sont peu ou pas gérés.

Si vous affichez une liste type de risques potentiels à une équipe, celle-ci va immédiatement les rattacher à son projet.

Pour voir l'importance de la gestion des risques, il suffit de rejouer des scénarios critiques de projets avec et sans prise en compte des risques. Il y a quelques années, je suivais un projet sur lequel une contrainte était de démontrer l'intégration de deux produits. Dès le début du projet nous avons affiché ce risque avec une probabilité d'occurrence de 80%. Au premier comité de pilotage, le Senior Manager s'interrogea sur la présence de ce risque et l'attitude de l'ensemble des participants a été d'aider à mitiger le risque. La solution proposée a été de faire intervenir un expert du produit sensé s'intégrer pour le démontrer en prenant

une journée de budget sur le projet. L'expert est venu et reparti démontrant l'impossibilité de s'intégrer, il a donc été convenu en comité de pilotage de procéder à l'intégration en effectuant un développement personnalisé. Si le risque n'avait pas été affiché et géré, à la fin du projet, l'équipe aurait fait naturellement le développement personnalisé d'intégration et l'appréciation aurait été qu'elle n'avait pas réussi le projet, n'étant pas parvenue à démontrer ladite intégration. Cette petite histoire permet de bien comprendre l'enjeu de gérer ou ne pas gérer ses risques sur un projet.

Les risques sont essentiels pour tout ce qui concerne les problématiques de faisabilité technique et ouvrent un canal de communication permettant au moment de l'occurrence de ne pas mettre le client devant un fait accompli, mais plutôt d'entamer une démarche consensuelle de résolution qui bénéficiera aux deux parties. Les risques doivent faire partie intégrante des Rapports de Statut projets au même titre que les données d'avancement.

De manière plus générale, des exigences mal définies incomplètes ou surréalistes, mais aussi un périmètre changeant de manière incontrôlée, des estimés en deçà de la réalité, des objectifs métiers peu clairs, complexes dans un contexte technologique en perpétuel évolution participent à cette situation difficile et augmentent les risques projets.

Sachant bien sûr que la gestion des risques ne réside pas simplement dans le fait de les lister dans un Registre souvent appelé Risk Register mais bien de gérer les points suivants :

- Probabilité, Impact, Priorité, sévérité, coût et stratégie.
- Plan de Mitigation : Comment est-ce que j'agis en amont pour diminuer la probabilité d'occurrence du risque.
- Plan de Contingence : Comment est-ce que j'agis en amont pour diminuer l'impact du risque au cas où il s'avère.
- Plan de réponse ou action : Quel est le plan à suivre en cas d'occurrence du risque et qui est en charge de l'exécution du plan.

3.3. Les Projets informatiques : toujours et encore le même scénario?

On ne se lasse jamais de revoir ce scénario qui a fait le tour du monde et illustre à la perfection de manière caricaturale le déroulement type d'un projet en décrivant les contributions de chaque partie prenante au fur et à mesure de l'avancement du projet de même que le niveau de qualité des artefacts produits avec au final comme dans les fables de La Fontaine la morale de l'histoire de ce projet.

	<p>Le Client : Au début, on a toujours ce que le client a exprimé comme besoin avec le plus souvent de la complexité et une part de surréalisme</p>		<p>La Documentation du projet.....</p>
	<p>Ce que le Project Lead a compris</p>		<p>Les opérations mises en œuvre.....</p>
	<p>Comment l'architecte a conçu le Design</p>		<p>Ce que ça a coûté au final au client</p>
	<p>Comment le Développeur l'a développé</p>		<p>Puis, le support du Help Desk</p>
	<p>Comment le responsable commercial décrit le projet</p>		<p>Ce dont le client avait réellement besoin !</p>

3.4. Les Notions Fondamentales

De l'importance d'initialiser correctement un projet :



**A Project fails
Always at the
Beginning
never at the End**

Un projet échoue toujours au début et jamais à la fin, cette grande vérité met en exergue le fait que le plus souvent, les projets démarrent dans l'urgence en s'affranchissant des prérequis nécessaires et reportant à plus tard la prise en compte de réels problèmes mais également et surtout toute la phase d'initialisation et de planification du projet afin de s'assurer un démarrage dans les meilleures conditions.

« Festina Lente », ce proverbe romain, riche de sens dans ce contexte permet de se rappeler qu'il ne faut pas confondre rapidité et précipitation. Une fois le projet démarré, plus on avance dans le temps, plus il sera difficile d'opérer des changements structurels et plus ces changements seront lourds et onéreux.

De l'importance d'une planification :

**If you Fail to Plan
...
You Plan to Fail**

Cette seconde règle explique que si nous ne sommes pas capables de planifier un projet, nous planifions l'échec de notre projet. Effectivement, le fait de planifier un projet démontre notre capacité à clairement enchaîner les tâches de manière coordonnée et efficace dans une plage de temps prédéfinie en vue de délivrer le produit du projet en alignement avec les attentes du client et dans les délais attendus. Et de manière corollaire ne pas fournir de planning peut être vu soit comme un scope insuffisamment défini donc non planifiable soit un manque de professionnalisme de l'organisation qui exécute entraînant une visibilité réduite sur le statut du projet.

De l'importance de la qualité et des tests :

**Prevention
Over
Inspection**

Cette citation nous est inspirée par Deming et traite du domaine de la gestion de la qualité. Il est largement démontré que les coûts mis en œuvre sur un projet pour prévenir des problèmes de qualité sont de loin bien moins importants que ceux qui seront engagés pour résoudre ces problèmes, sans compter les conséquences souvent bien plus dramatiques quand les problèmes de qualité sont découverts en fin de projet. On considère que si l'on procède en mode prévention il en coûtera statistiquement de 5 à 12% du projet, alors qu'en mode inspection ce sera plutôt de l'ordre de 12 à 20% sans compter l'insatisfaction du client qui a également un coût et au final on parlera de COPQ (Cost of Poor Quality). Deming disait également : **Quality is a Management Problem** et donc le succès requerra la participation de l'ensemble de l'équipe et il est de la responsabilité du management de fournir les ressources nécessaires au succès.

De l'importance de la rationalisation et de l'optimisation :

**Reduce
waste**

Ce principe met en évidence le gaspillage de ressources qui sur les projets pourrait être considérablement diminué à condition d'optimiser, outiller et automatiser tout ce qui peut l'être et par là même permettre aux membres de l'équipe de se concentrer uniquement sur des tâches à haute valeur ajoutée, plutôt que des tâches répétitives, fastidieuses à faibles valeur ajoutée. Une équipe doit conserver le focus sur le fait de produire de la valeur tout en éliminant tout travail inutile, et donc rester concentré sur l'alignement avec les attentes du client et sa satisfaction, condition sine qua non de réussite d'un projet.

4. Premiers éléments de réponse : Méthodologie et Gestion de Projet

4.1. Répartitions Statistiques des méthodologies utilisées

Bien évidemment la méthodologie se positionne comme élément clé dans la réalisation d'un projet informatique, après avoir analysé les causes d'échecs, nous allons nous intéresser à l'état des lieux pour ce qui concerne le niveau d'utilisation des principales méthodologies.

Ce rapport du Forrester nous montre la répartition en pourcentage de l'utilisation des méthodologies sur un vaste échantillon de sociétés interrogées. On ne distingue clairement que l'Agilité arrive en tête avec notamment la mouvance SCRUM. Néanmoins, on parlera le plus souvent de *Scrum But* ou encore de *Water-Scrum-Fall* pour signifier que l'adoption des méthodes agiles a divergé de l'idée originale décrite dans le Manifeste Agile pour se transformer en un savant mélange de pratiques traditionnelles type WaterFall pour ce qui concerne l'initialisation du projet et le passage en release et de pratiques SCRUM pour la phase de développement.

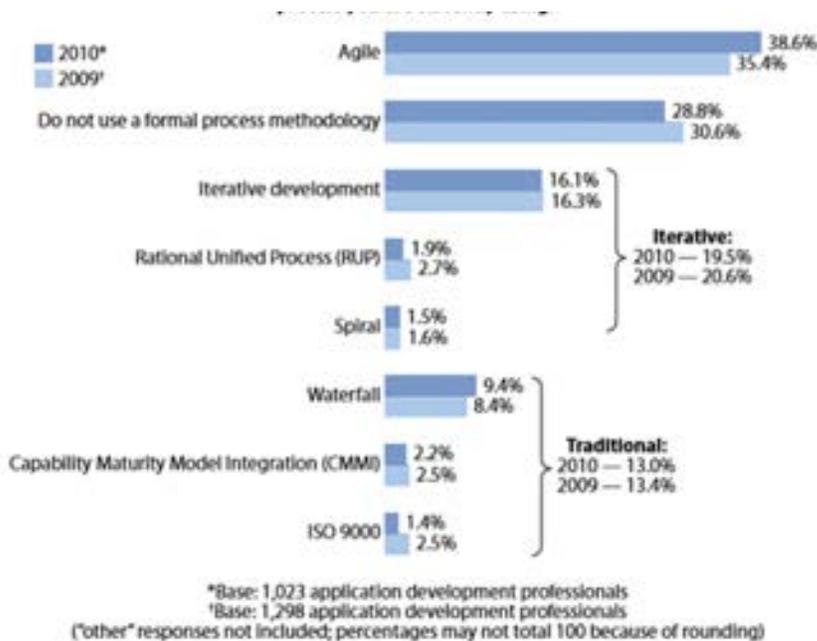


Figure 4 : Répartition des Méthodes (Source Forrester)

Le deuxième graphe nous montre la répartition des différentes pratiques Agiles sur un large échantillon de sociétés interrogées, bien évidemment, Scrum arrive largement en tête.

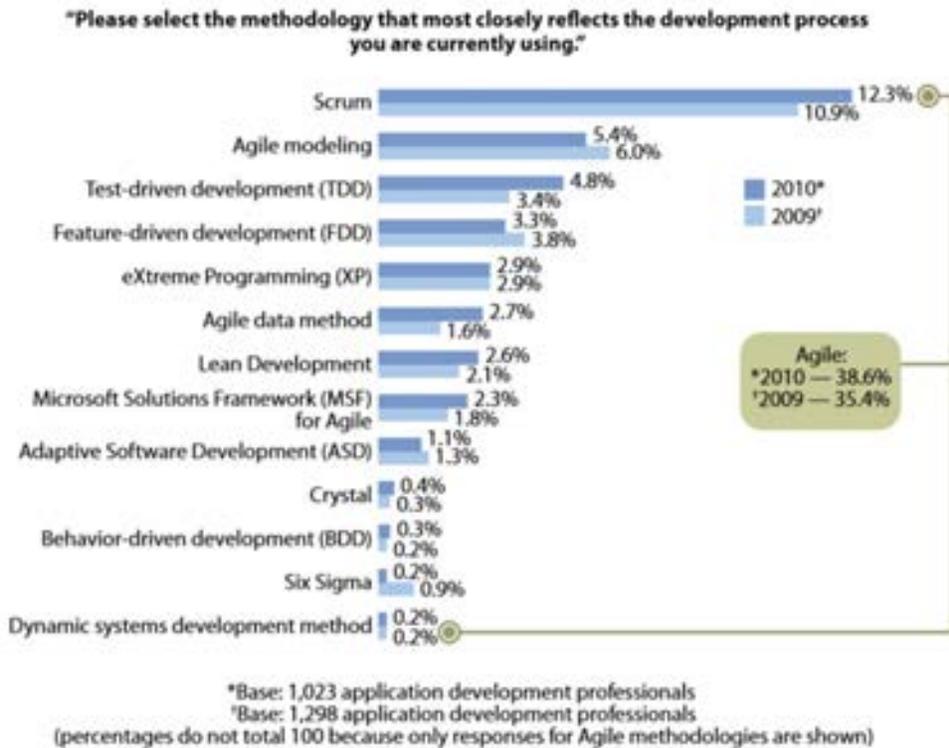


Figure 5 : Répartition des méthodologies Agile (Source Forrester)

4.2. L'approche Agile

Force est de constater donc que l'Agilité a gagné en Momentum sur le marché du développement ces dernières années et ce phénomène est sans cesse croissant.

"Agile" est un terme parapluie qui englobe un grand nombre de méthodologies qui promeuvent un travail effectué sur une base itérative où les exigences fonctionnelles et les solutions évoluent au cours du temps.

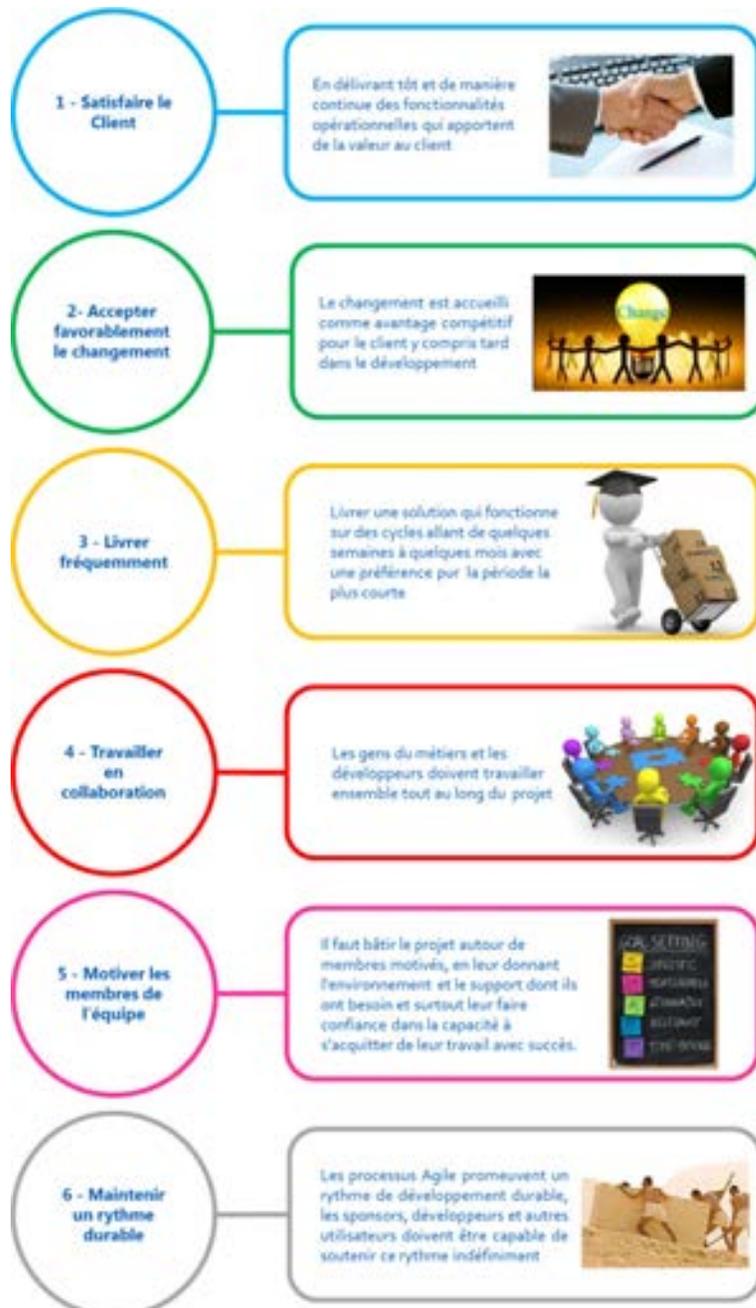
L'approche Agile va demander une grande disponibilité des équipes fonctionnelles du client afin de valider cette approche pas à pas où la vision du client va être implémentée progressivement et rediscutée régulièrement afin de s'assurer de la convergence vers l'objectif. Le seul juge de paix est le logiciel qui fonctionne plutôt que rapports et documents décrivant le logiciel que l'on verra plus tard... L'équipe est bien évidemment élargie pour y inclure le client et une collaboration s'installe autour d'un plan de travail itératif commun.

Il existe beaucoup de Mythes autour de l'agilité tels qu'en mode Agile il n'y a :

- Pas de planification
- Pas de documentation
- Pas d'engagement

- Des équipes Agiles qui ne sont pas disciplinées
- Pas de Gestion de Projet ni de Plan Qualité
- Etc.

Le Développement Agile confère une approche itérative et flexible du développement logiciel en s'appuyant sur 12 Principes qui constituent le **Manifest AGILE** décrit ci-dessous :





Une mouvance du courant Agile est représentée par SCRUM qui suscite un vif intérêt actuellement. Cette approche très typée au niveau des principes et de la sémantique hérite d'un outillage riche notamment dans le monde de l'ALM et plus spécifiquement avec TFS 2012, SCRUM pousse la notion de performance jusqu'à des concepts tel que le « SCRUM of

SCRUM » pour ce qui concerne la parallélisation des Sprints par exemple, ci-dessous une représentation de cycles types simples et complexes SCRUM :

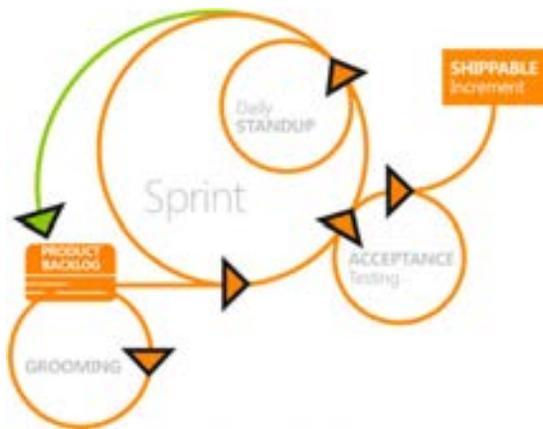


Figure 6: Modèle Simple

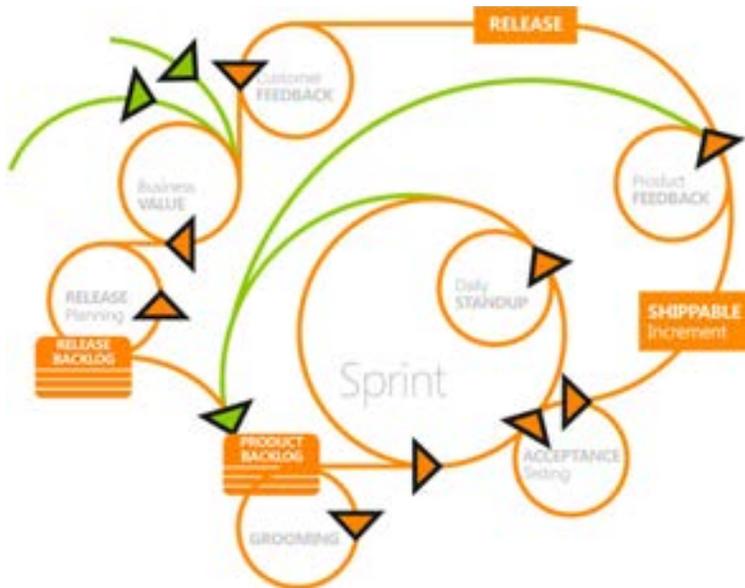


Figure 7: Modèle plus complet avec Intégration et Feedback en continu

Pour résumer, l'approche Agile privilégie une communication fréquente avec le client pour vérifier un alignement permanent avec le besoin métier, base son avancement sur une version opérationnelle de l'application et un Burndown Chart et s'engage à livrer fréquemment des modules fonctionnels finalisés par ordre de priorité d'importance métier. L'Agilité semble très bien adaptée à des équipes de taille petites à moyennes, et à des projets supportant le mode itératif. Dans ces conditions, les résultats obtenus sont extrêmement satisfaisants.

4.3. L'approche CMMI / PMI avec le Cycle en V

Le cycle en V tire son origine de l'armée et de l'industrie et a été adaptée à l'informatique dans les années 80. L'avantage du cycle en V est le haut niveau de détail dans la description du déroulement de la phase projet, de la manière par laquelle il sera réalisé. Bien que limpide par son modèle, il reste néanmoins très difficile à appliquer dans la réalité de manière stricto sensu. Le cycle en V doit être vu comme un modèle idéal vers lequel on doit tendre à l'instar d'autres modèles comme ITIL pour le management de systèmes d'information, ou encore OSI pour les réseaux.

Le modèle du Cycle en V est un paradigme de gestion de projet censé résoudre les problématiques de réactivité, de résistance au changement et de faible prédictibilité imposées par le modèle Cascade (Waterfall), notamment pour tout ce qui concerne la gestion du changement ou les actions correctives. Il inclut la notion d'itératif cher au paradigme Agile qui va permettre de ne pas attendre la fin d'un Cycle V (3 à 6 mois généralement) pour réagir.

Le cycle en V ne doit pas être confondu avec le système Waterfall dont les inconvénients sont bien connus après avoir mis bon nombre de projets en grande difficulté :

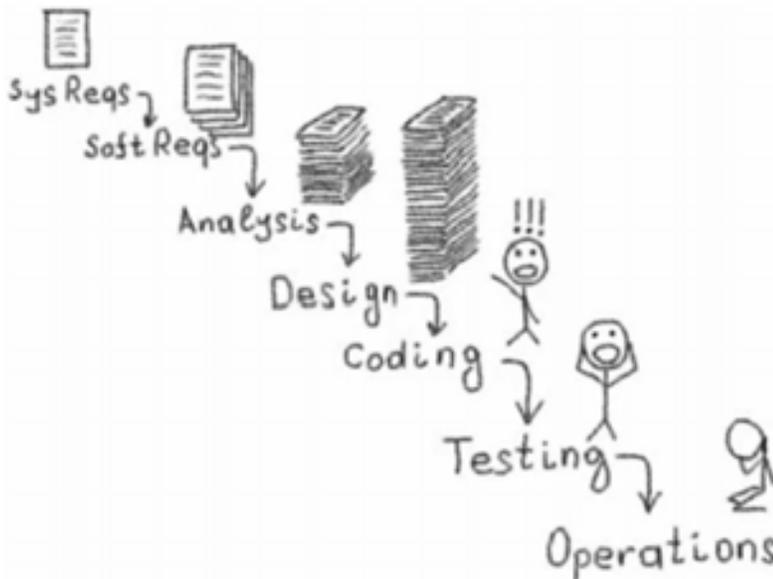


Figure 8 : Exemple de Structure Cascade ou Waterfall

Sur le principe, le cycle en V démarre par la réalisation en partant du plus général, l'analyse du besoin vers le plus détaillé, le développement puis remonte au travers des phases de tests du plus détaillé vers le plus général avec des actions correctives itératives, comme montré ci-dessous :

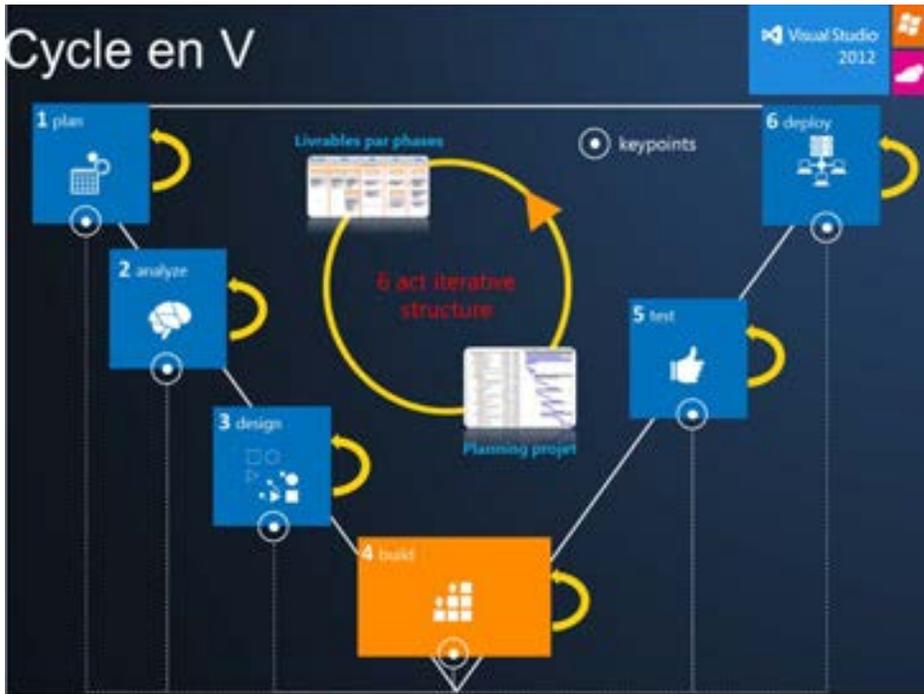


Figure 9 Diagramme du Cycle en V

Les modèle CMMI / PMI collent parfaitement à la mise en place d'un cycle en V sur projet avec la prise en considération des éléments haut niveau suivants :

- CMMI (Capability Maturity Model Integration)
 - Livrables par phase
 - Planning projet
 - Référentiel de bonnes pratiques
- PMI (Project Management Institute)
 - Fournit un Framework de gestion de projet reconnu dans le monde entier
 - Permet de contrôler les éléments de la triple contrainte : Budget, Délai, Scope et Qualité
 - Utilisation d'un **WBS Projet** : Définition et contrôle du Scope
 - Utilisation d'indicateurs de performance (KPIs) pour piloter
 - CPI : Cost Performance Index
 - SPI : Schedule Performance Index

Cette approche se veut plutôt rassurante pour le client qui connaît à l'avance l'ensemble des livrables qui seront produits et sur lesquels s'engage l'intégrateur mais également une vue planifiée de l'enchaînement des tâches avec en plus une surcouche de contrôle permettant de communiquer sur des métriques clés d'avancement, de consommé mais aussi la qualité et les risques. Cette approche demande bien évidemment un type de projet adapté, et une

maîtrise parfaite des concepts avec un chef de projet expérimenté, car si les livrables et/ou le planning ne sont pas réalistes au départ (SMART : Specific, Measurable, Achievable, Realistic, Time Bound) on revient sur les échecs douloureux bien connus de ce type de méthode.

Pour résumer, on a une approche phasée, avec des livrables, un planning, un engagement fort de l'intégrateur et un contrôle au travers de la mise en place de rapports permettant de communiquer en toute transparence sur la situation du projet avec le client et effectuer des actions correctives ou préventives si nécessaires, l'objectif principal étant de rester aligné avec les objectifs stratégiques du client tout en respectant l'engagement initial en termes de délai, budget, périmètre et qualité. Cette approche méthodologique est particulièrement bien adaptée aux grands projets avec de grandes équipes en mode forfait avec un périmètre relativement figé.

4.4. Comparatif des deux approches

Les 2 méthodes vues précédemment viennent avec un ensemble de caractéristiques pouvant être selon les projets des avantages ou des inconvénients selon le contexte. Nous verrons que l'ALM va permettre de tirer parti des avantages des 2 méthodes en fonction des contextes projets du fait de la souplesse apportée par l'outillage et les possibilités d'extensibilité et de configuration.

Ci-dessous un tableau récapitulatif permettant de comparer les différences au niveau de l'approche entre les deux méthodes :

Approche Standard Cascaded vs Approche Agile	
	
Cascade	Agile
<ul style="list-style-type: none">• Approche Déterministe• Pilote par la planification• Communication avec le client peu fréquente• Cycles Longs de Releases de 6-12 mois• Fonctionnalités livrées au début• Développement en phases avec des livrables intermédiaires• Développement en couche: Présentation, Mysql, etc.• Programmation vue comme implémentation• Intégration des couches en fin de phase• Les Tests sont effectués en fin de phase	<ul style="list-style-type: none">• Approche Empirique• Pilote par l'apprentissage• Communication continue avec le client• Cycles Courts de Releases, et 2-4 mois• Fonctionnalités évoluant tout au long du projet• Livraison de blocs opérationnels en itérations courtes• Développement par fonctionnalité en itérations• Programmation vue en comme moyen de conception• Intégration continue du code• Les Tests sont effectués tout au long de l'itération
<p>L'avancement est mesuré en termes d'artefacts de type livrable ou % d'avancement des tâches</p>	<p>Une application qui fonctionne est le facteur de mesure de l'avancement</p>

Ce qui est troublant lorsque l'on regarde ce tableau comparatif est de constater dans la réalité, qu'il n'y a qu'un pas pour faire de la « méthodologie politicienne » et donc qu'à partir du moment où l'on se trouve dans l'un ou l'autre modèle, rejeter catégoriquement toute pratique de l'autre modèle, et c'est précisément là que la bât blesse, car à la fois l'approche Cycle en

V et Agile sont à l'origine d'écueils projets et donc la meilleure approche va consister à tayloriser le meilleur des 2 méthodes (ou plus) afin de créer la méthode qui va nous permettre de réussir notre projet et tant pis si elle n'est pas tendance ou conforme à tel ou tel modèle.

Si l'on prend le cas de l'approche Agile, on pourra noter un certains nombres d'effets pervers ou dérives de la méthode dans les cas caricaturaux suivants par exemple :

- Suivre un client qui change d'avis comme une girouette, sous prétexte d'être réactif au changement.
- Travailler sur des itérations trop courtes qui non seulement vont épuiser les équipes et entraîner la suppression d'activités prévues telles que l'écriture des tests unitaires voire la réalisation de tests entraînant des livraisons incluant des problèmes de qualité et une dette technique grandissante.
- Des équipes qui s'auto organisent, avec des développeurs n'écrivant aucune documentation technique, des chefs de projets refusant de fournir un planning ou d'utiliser des outils de suivi...

On ne le rappelle jamais assez souvent, la réussite d'un projet c'est la satisfaction du client et non pas passer son temps dans des pratiques méthodologiques et gestion de projet au détriment de l'objet du projet lui-même !

5. L'ALM : La Solution et les Outils

5.1. L'ALM : Pourquoi et quels bénéfices ?

La problématique des projets IT est récurrente : gérer toujours plus de complexité, tant métier que technique, rester aligné avec le métier et réactif au changement pour respecter les priorités sans cesse changeantes avec une demande rémanente : faire toujours plus avec moins de moyens.

Pour faire face à ces challenges, une solution de type ALM est incontournable afin de suivre piloter et communiquer sur le statut d'une solution en cours d'élaboration afin de s'assurer à tout instant que le produit cible sera en mesure d'exécuter dans les conditions attendues le besoin métier. Cela reposera sur de l'industrialisation, de l'automatisation, de l'intégration continue, de la traçabilité et surtout du pilotage et du Reporting temps réel sur l'ensemble des artefacts produits par le projet et le produit du projet.

Définition : L'ALM peut être basiquement défini comme une collection de disciplines et d'outils qui vont permettre de transformer un besoin métier en une solution informatique opérationnelle.

Avec la version 2.0 de l'ALM, on note l'intégration de nouvelles disciplines telles que PPM, la gestion de projet et de portefeuille (Project and Portfolio Management) dans le but de

rapprocher les activités de développement avec le métier et le management dans les grandes organisations. Un nouveau besoin apparait donc dans le domaine de l'outillage des solutions ALM pour la prise en compte de ces nouvelles disciplines qui viennent briser les silos fonctionnels pour permettre encore plus de collaboration et de consistance au sein des organisations.

L'introduction et la conduite d'une stratégie ALM dans son entreprise n'est pas une tâche simple, car il va falloir dans un premier temps évaluer le niveau de maturité actuel, le niveau cible et planifier une conduite de changement réaliste et priorisée pour tirer au final les bénéfices de l'ALM qui sont globalement :

- Une amélioration du ROI sur les investissements IT
- Une amélioration du Time To Market
- Une amélioration de la Qualité
- Un meilleur alignement avec le Métier
- Un taux de réussite de projet en progression

Les trois piliers fondamentaux de l'ALM sont :

Traçabilité et liaisons entre les artefacts : Ceci est traditionnellement un processus manuel extrêmement lourd, dont l'effort varie selon la taille d'un projet et les variétés d'artefacts à relier. La conformité aux exigences fait de la traçabilité une nécessité. Ci-dessous, le schéma de base de la notion de traçabilité et de la liaison entre les différents artefacts projets et les parties prenantes tout au long du cycle de vie :

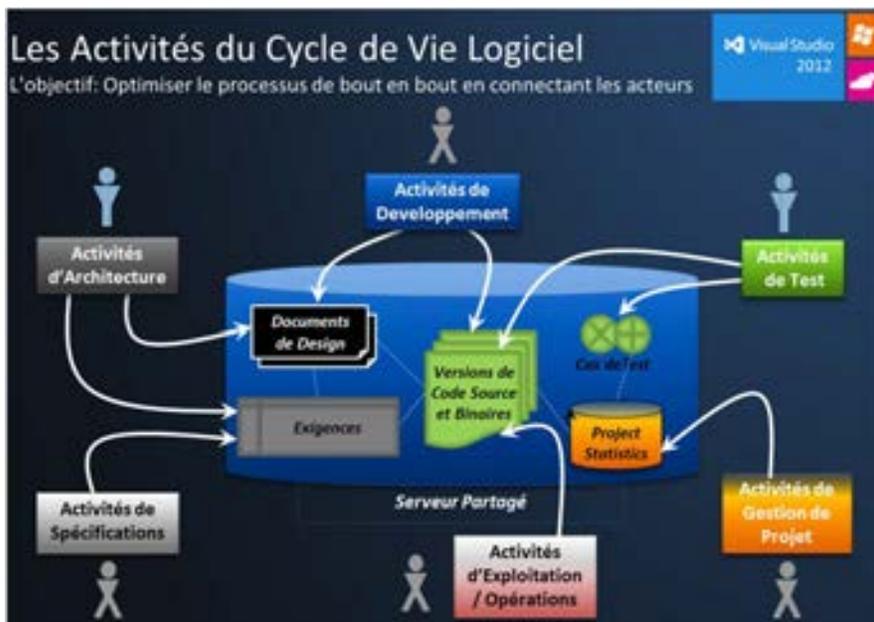


Figure 10 : Illustration de la notion de Serveur partagé au sens ALM

L'automatisation des processus : Les solutions ALM optimisent l'efficacité en fournissant une gamme de pistes pour l'automatisation de processus journaliers sur des aspects méthodologiques ou gestion de projet, et permettent également une communication fluidifiée avec une batterie d'outils permettant d'automatiser des processus liés à l'activité de développement ou de test alliant productivité et qualité. L'objectif in fine étant d'automatiser tout ce qui peut l'être afin de permettre à l'ensemble des acteurs de se concentrer sur le cœur de leur métier, les tâches à haute valeur ajoutée en se débarrassant de toutes les tâches répétitives et fastidieuses à moindre valeur ajoutée que l'on va déléguer à la plateforme ALM.

Le Reporting pour augmenter la visibilité : La plupart des chefs de projet ont le plus souvent une visibilité relative sur le niveau d'avancement du projet, le niveau de respect des éléments de la triple contrainte (budget, délai, périmètre et qualité). Le plus souvent la visibilité qu'ils ont provient d'échanges directs avec les membres de l'équipe (mode déclaratif) et repose donc sur des éléments subjectifs mais surtout pas objectifs. Le Reporting à partir d'une plateforme ALM profite de l'intégration de l'ensemble des artefacts pour fournir des informations temps réel sur l'état du projet à tout instant. et potentiellement permettre au chef de projet de faire des investigations plus poussées si nécessaires.

L'ALM est donc un paradigme de conduite outillée de projets informatiques pour gérer le développement de solutions IT en proposant un outillage, une structuration et des possibilités d'automatisation des processus de bout en bout, avec l'intégration de l'ensemble des informations au cours des différentes étapes du cycle de vie complet.

L'ALM permet à la fois une intégration **horizontale** et **verticale** des outils, avec par exemple l'intégration d'un EPM avec une usine logicielle permettant de connecter des activités de gouvernance directement à des activités de développement. L'ALM doit adresser les trois niveaux que sont la gouvernance, le développement et les opérations, et cet objectif est loin d'être trivial sachant que les outils ALM aujourd'hui en fonction des offres ne sont pas aussi bien intégrés qu'ils devraient et c'est justement le critère « Intégration » qui doit déterminer le juste choix d'une plateforme ALM.

Selon le cabinet Forrester, la conscience de l'existence de l'ALM est élevée, alors que sa compréhension est relativement limitée, comme le montrent les diagrammes ci-dessous :

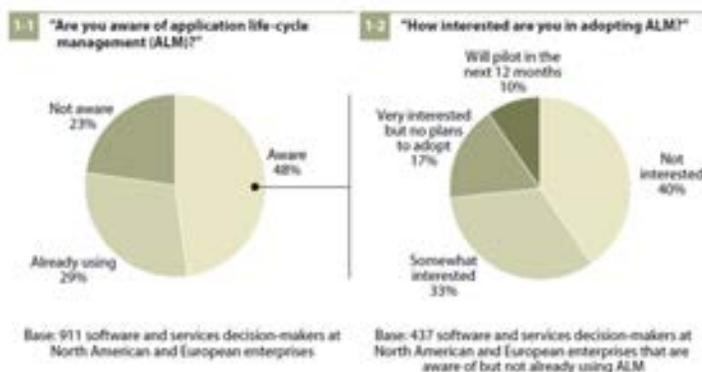


Figure 11 : Awareness of ALM Is High (Source Forrester)

Toujours selon le cabinet Forrester, une solution ALM se définit comme :

Un ensemble d'outils intégrés supportant et unifiant les activités du cycle de vie suivantes :

- L'analyse et la gestion des exigences
- Le design et la modélisation
- La gestion de projet
- La gestion du changement
- Le développement,
- La gestion de configuration (SCM)
- La gestion des Builds
- La gestion de Release
- L'activité de test et de déploiement.

En conclusion, on constate que de manière récurrente, il a toujours été très difficile pour des raisons diverses d'ailleurs d'avoir un suivi précis d'un projet, des automatisations, de la transparence etc., tous ces éléments qui font partie des fondamentaux à mettre en place pour réussir un projet. Ce problème est essentiellement dû à la multiplication d'outils non intégrés rendant difficile l'instanciation de ces trois piliers.

5.2. Démarrer avec l'ALM

Si l'on considère que seules 30% des entreprises aujourd'hui utilisent une solution ALM, un grand nombre d'entreprises vont donc démarrer dans les années à venir, et il est important de comprendre que l'adoption d'une solution ALM passe par un processus de conduite du changement qui n'est pas forcément trivial et devrait commencer par une phase d'audit permettant d'établir un plan d'action pour le déploiement de la solution ALM cible.

La mise en place d'une solution ALM ne doit jamais se faire en mode big bang mais plutôt en mode itératif. La première chose à faire est d'analyser les processus et les outils existants, faire une analyse de type SWOT (Strength, Weaknesses, Opportunities, Threats) afin de déterminer quels sont les points forts mais aussi les faiblesses de l'organisation en cours afin de planifier au mieux une implémentation ALM. Il est également important de recenser les outils existants et étudier comment réaliser une intégration optimale des nouveaux outils avec l'existant car l'intégration est un des points clé d'une solution ALM. Cette phase peut conduire à des arbitrages sur les outils existants et à venir.

Le but est d'établir une Roadmap de déploiement ALM tant au niveau processus qu'outils.

Il va falloir introduire les nouvelles pratiques et processus et l'outillage ALM de manière in-

crémentale, former les équipes à ces nouveaux outils et les intégrer dans le SI de l'entreprise.

6. La Solution ALM de Microsoft : Visual Studio 2012

6.1. Architecture Technique et Fonctionnelle de TFS 2012

La plateforme ALM Microsoft TFS 2012 fournit un ensemble d'outils très riche fonctionnellement permettant d'adresser l'ensemble des activités liées au cycle de vie du développement d'une application de la définition des exigences jusqu'aux phases de test, déploiement et opérations avec un support riche pour la méthodologie et la gestion de projet.

Ci-dessous la Roadmap fonctionnelle de l'ALM avec TFS 2012. La plateforme comme montrée sur le schéma ci-dessous adresse trois niveaux de maturité qui sont :

- Maturité Basse : Technique (Niveau Individu) → Domaine de l'Implémentation
- Maturité Medium : Méthodologie (Niveau Equipe) → Domaine de l'Organisation
- Maturité Elevée : Gestion de projet (Niveau Entreprise) → Domaine de la Gouvernance avec du Pilotage et du Contrôle projet.

Chaque niveau inclut bien évidemment les éléments du niveau précédent :

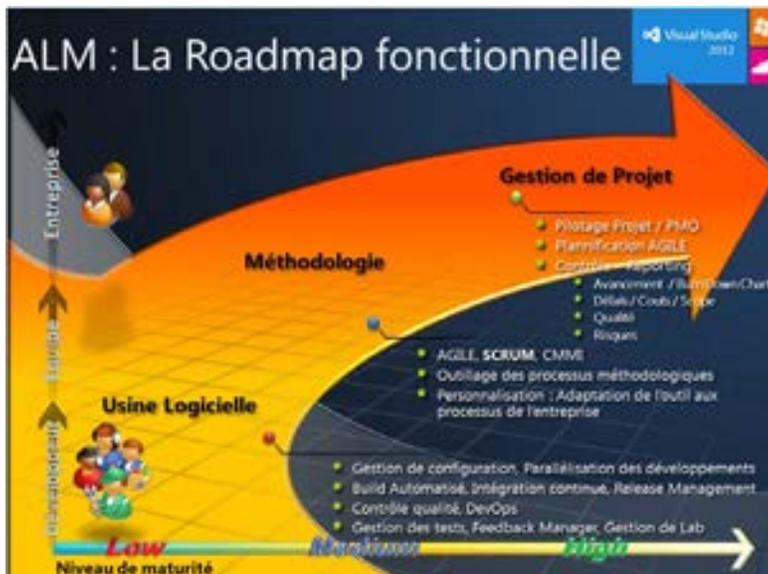


Figure 12 : Roadmap Fonctionnelle et Maturité ALM

Le Niveau de maturité basse ou niveau technique est le niveau de maturité le plus faible, l'ensemble des tâches réalisées sont techniques et les équipes exploitent les riches fonctionnalités de la plateforme usine logicielle pour délivrer le produit du projet sans

méthode ni contrôle. Généralement, à ce niveau, la gestion de configuration est le facteur majeur d'adoption de la plateforme. On va donc retrouver des activités de développement, de la gestion de release, de la gestion de configuration, de la parallélisation des développements, de l'intégration continue s'appuyant sur la plateforme de Build automatisée, et des activités de test et déploiement. Ce niveau est clairement à risque dans le cadre de projets informatiques du fait de la non coordination des acteurs et de l'absence de méthode et de contrôle.

Le niveau de maturité medium ou niveau méthodologique va impacter l'organisation et la structuration des processus de travail, on va donc ici s'appuyer sur une méthodologie, avec en standard sous TFS le choix entre AGILE, SCRUM ou CMMI. Bien évidemment la plateforme permet de personnaliser ces modèles de processus pour les adapter à tout type d'organisation pour par exemple outiller une méthodologie définie en interne dans laquelle l'entreprise a très largement investi. Ce niveau est qualifié de Medium, tout simplement car le fait de dire comment on va faire ne garantit pas que l'on va faire comme on a dit, car pour cela il faut monter d'un cran au niveau de la maturité et passer au niveau Gestion de projet pour apporter contrôle et gouvernance.

Le niveau de maturité élevé ou niveau Gestion de projet va permettre de boucler le cercle vertueux de l'ALM grâce aux outils de pilotage et de contrôle de la gestion de projet qui viennent également en standard avec la plateforme notamment grâce aux capacité de Reporting de la plateforme et la possibilité de s'appuyer simplement sur des rapports d'avancement temps réel jusqu'à la possibilité de faire un suivi de projet en s'appuyant sur la méthode de la valeur Acquis (ou Earned Value Analysis) permettant de communiquer sur les métriques clés du planning et du budget : CPI (Cost Performance Index), SPI (Schedule Performance Index), EV (Earned Value), AC (Actual Cost), PV (Planned Value), CV (Cost Variance), SV (Schedule Variance) et %Complete. Ci-dessous des exemples d'utilisation des résultats obtenus par la méthode de la valeur acquise.



Figure 13 : Exemple de Tableaux de bord produits à partir d'artefacts TFS

D'un point de vue architecture technique, on notera une évolution considérable depuis la précédente version TFS 2010 tel que montrée sur l'illustration ci-dessous avec l'intégration de nombreuses briques techniques supplémentaires tant coté client que serveur :



Figure 14 : Roadmap Technique TFS 2010 - 2012

La plateforme ALM TFS 2012 apporte un grand nombre de fonctionnalités à haute valeur métier pour la réalisation de projets dont :

Un support hiérarchique pour les Work Items : Déjà présente dans la version 2010, cette fonctionnalité permet une véritable activité de Gestion de projet avec le support d'un WBS Projet (Work Breakdown Structure qui représente l'arborescence hiérarchique des livrables d'un projet), ceci permet par exemple de gérer une arborescence de type *Projet / Phase / Module / Exigence / Tache* en mode CMMI ou bien *Epic / User Story / Task* pour un Carnet de produit en mode Agile ou SCRUM, et de se synchroniser avec un fichier MS Project ou encore avec Project Server. TFS 2012 intègre de plus un mécanisme de Roll up permettant la consolidation hiérarchique de données de consommé.

Une plateforme de test professionnelle : Avec Test & Lab Center, Microsoft propose une offre complète pour des équipes de test professionnelles, la plateforme est entièrement intégrée à TFS 2012 et propose un client dédié aux testeurs fonctionnels avec des outils de

planification de campagnes de test, un outil d'exécution des tests incluant des possibilités d'automatisation et de gestion de paramètres de tests pour améliorer la productivité et la fiabilité lors de la phase d'exécution.

Comme nouveauté, on retrouvera l'intégration d'un module de *Test Exploratoire* permettant de tester une application sans un script prédéfini. Le testeur utilisera sa créativité pour détecter et créer des bugs tout en générant en arrière-plan les étapes de reproductions. Le testeur pourra par la suite générer des cas de tests à partir de ces sessions exploratoires. A l'instar de Test Manager, on retrouvera de la capture audio, vidéo et screenshots, permettant d'encapsuler un contenu riche dans la session exploratoire. Un cas d'utilisation intéressant est de l'utiliser dans le cadre d'une activité de support, en rejouant en temps réel la manipulation du demandeur, pour constater l'anomalie puis transmettre le bug avec le cas de test généré au développeur.

Avec cette plateforme, Microsoft adresse le syndrome bien connu de la non reproductibilité des bugs en fournissant un mécanisme de collecte d'artefacts contextuels à l'exécution du test qui sont téléchargés automatiquement lors de la création d'un bug dont : les étapes détaillées du test avec les résultats, la capture Video de la session de test, les données IntelliTrace (Historique de Debug, breveté par Microsoft) permettant au développeur de connaître le code fautif avec les données source de l'exception, la pile d'appel (Stack Trace, Mini dump mémoire) et également des données de type Event Log et Informations Système.

Autre nouveauté avec Lab Center, Microsoft fournit une plateforme de Provisioning d'environnement pour les tests avec des capacités de déploiement des binaires applicatifs et d'exécution de tests de manière automatisés unitaires ou fonctionnels. Cette dernière fonctionnalité améliore considérablement la productivité et la qualité sur les projets car il est bien connu que le Provisioning et la maintenance d'environnements d'exécution pour les tests est coûteuse en temps et répétitive, tout ce que l'on veut éviter sur projet. Limité à un hosting de type Hyper V sous TFS 2010, la nouvelle version va permettre d'adresser tout environnement physique ou virtuel, Hyper-V ou VMware, ce qui devrait enfin permettre de l'utiliser massivement sur projet.

Un outil de collecte de Feedback avec Feedback Manager :

Ce nouvel outil a pour objectif d'engager les parties prenantes dans un processus de feedback continue, l'équipe initialise une demande de feedback sur une fonctionnalité développée en utilisant l'outil de feedback intégré à Web Access pour la requête de Feedback.

Un client gratuit est disponible pour effectuer la session de feedback et faire un retour riche sur la session et également entrer une notation.

Cet outil se positionne comme une formidable opportunité de communication avec le client permettant d'obtenir un feedback rapide de la part du client et donc typiquement inscrit dans une approche Agile.

Un outil de Storyboarding avec PowerPoint :

Cet outil manquait cruellement à la suite des composants de TFS, enfin un outil permettant de réaliser du prototypage et de simuler des cinématiques d'IHM. Le produit utilisé est PowerPoint qui a tout simplement été doté de fonctions de connections à TFS et également de bibliothèques de formes permettant de prototyper rapidement des interfaces d'applications, avec en plus la capacité de capturer des écrans d'applications existantes pour les modifier dans l'outil. Tout le monde sait utiliser Powerpoint ce qui devrait grandement favoriser l'adoption de cette superbe fonctionnalité.

Une intégration avec System Center pour les opérations en production :

Encore une fonctionnalité très attendue, qui va permettre une meilleure interaction entre les équipes de développement et les équipes d'opérations. Cette intégration va permettre directement à partir de System Center d'ouvrir des anomalies dans TFS et dans le meilleur des cas transmettre des traces Intellitrace aux développeurs pour apporter des informations très précises sur les erreurs survenues. L'activation d'intellitrace en production est également une nouveauté Visual Studio 2012.

Le Support de l'UML : On retrouve l'UML comme langage de modélisation logique dans la plateforme ALM avec :

- Les diagrammes d'activité
- Les diagrammes de composants
- Les diagrammes logiques de classe
- Les diagrammes de cas d'utilisation
- Les diagrammes de séquence

Des cas d'utilisation ou tout type de diagramme UML peuvent être créés avec ce module et surtout on notera la possibilité de les relier à n'importe quel autre artefact projet. Ci-dessous des illustrations de diagrammes obtenus avec ce module.



Figure 15 : Exemple de Use Case avec Visual Studio 2012

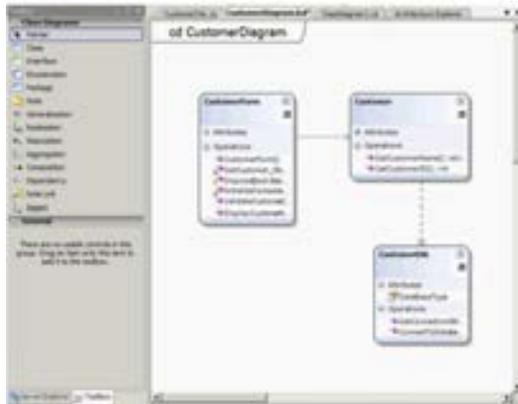


Figure 16 : Exemple de diagramme de classe avec Visual Studio 2012

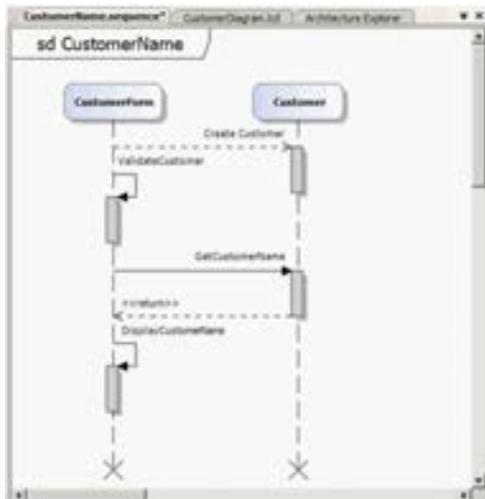


Figure 17 : Exemple de diagramme de séquence avec Visual Studio 2012

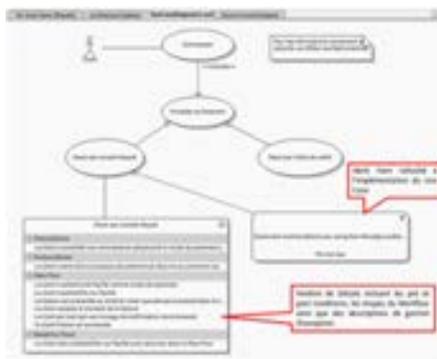


Figure 18 : Exemple de diagramme d'état avec Visual Studio 2012

Une plateforme Full 64 bits : L'ensemble des composants techniques de la plateforme supporte l'architecture 64 bits garantissant un haut niveau de performance pour :

- TFS 2012 le serveur applicatif
- SQL 2012 le serveur de base de données qui stocke l'ensemble des artefacts projets et expose les services Analysis Services et Reporting Services
- SharePoint 2010 le serveur collaboratif avec potentiellement un surcouche Project Server
- Build Server, la plateforme de Build pour l'intégration continue

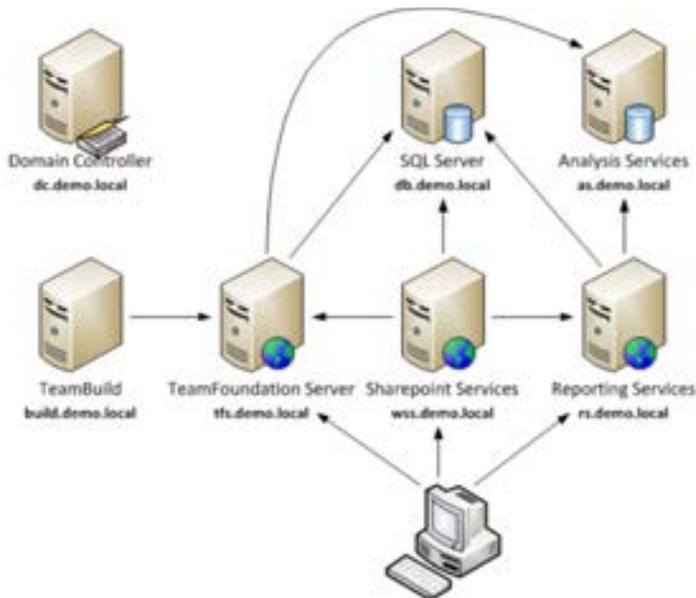


Figure 19 : l'architecture physique et logique d'un déploiement TFS 2012 en mode distribué

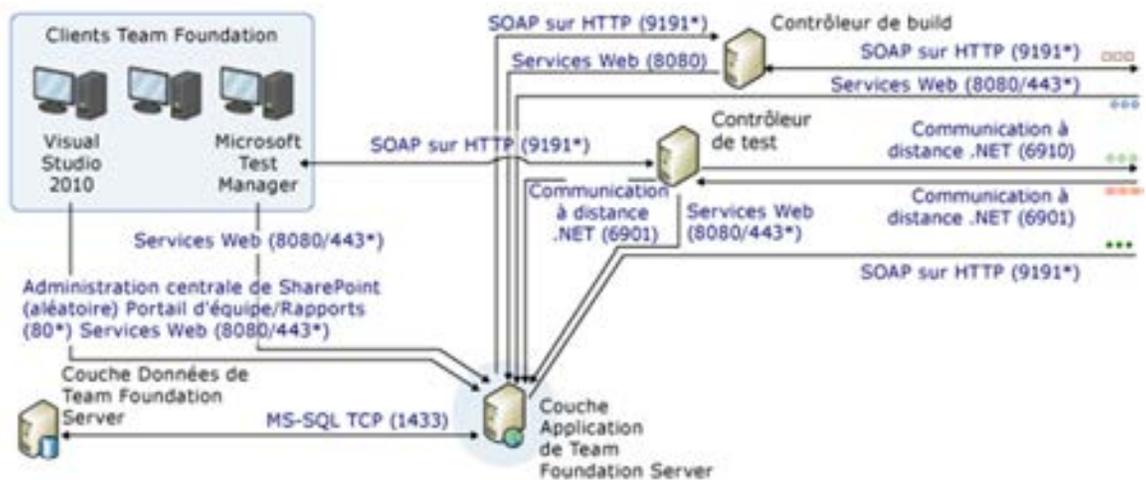


Figure 20 : Cartographie des flux

Une Plateforme de Reporting : Un des facteurs clé de TFS est un ensemble très complet de rapports fournis en standard qui correspondent à plus de 80% des besoins de Reporting. Les rapports proposés en standard sont adaptés par méthodologie, la plateforme propose des rapports spécifiques pour SCRUM si cette méthodologie a été choisie pour instancier le projet d'équipe.

Ci-dessous les rapports proposés en standard :

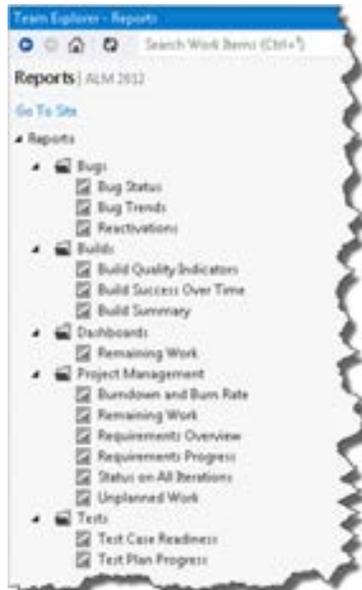


Figure 21 : Rapports Standard TFS

Par exemple, le rapport ci-dessous nommé « Stories Overview » qui donne une vue complète « 360° » du statut d'un projet avec les User Stories ou exigences du projet à gauche, et en regard, l'avancement, le reste à faire, le nombre de tests écrit par exigences, l'état et le résultat de l'exécution des tests et le nombre de bugs ouverts et résolus. On notera la notion de roll up des données du niveau exigence jusqu'au niveau projet.



Figure 22 : Vue 360° avec le rapport Requirement / Stories Overview

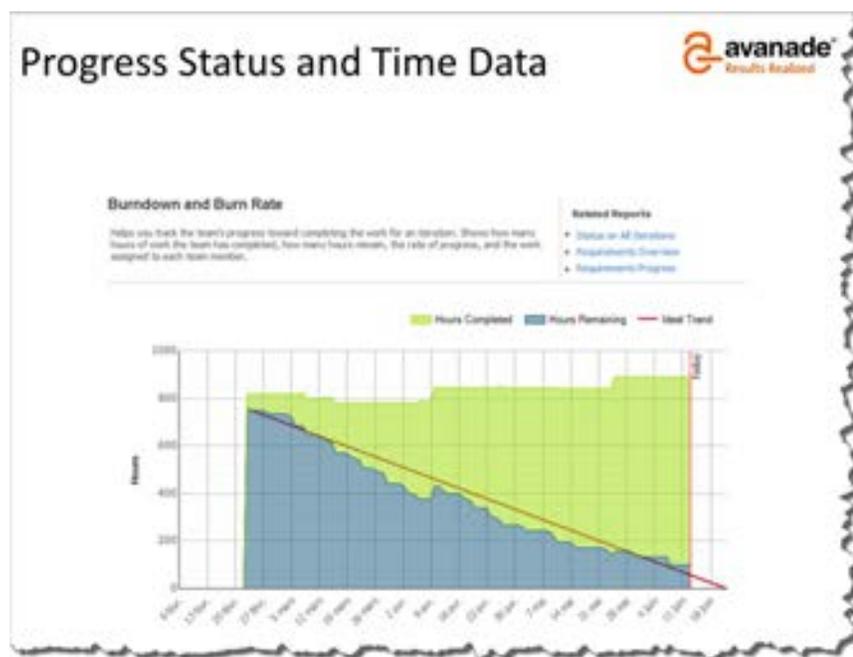


Figure 23 : Rapport Burn Down & Burn Rate

Il est également possible de créer des rapports personnalisés permettant de s'adapter aux besoins de Reporting de toute organisation :



Project Bug Status



Figure 24 : Exemple de rapport personnalisé

La plateforme ALM TFS 2012 permet à tout utilisateur de personnaliser des rapports simplement en utilisant Excel et les possibilités de connections aux bases relationnelles ou multidimensionnelles de TFS, la plateforme permet simplement à partir de requêtes « Work Item Queries » de générer des rapports à la demande, comme montré ci-dessous, simplement à l'aide du menu contextuel.

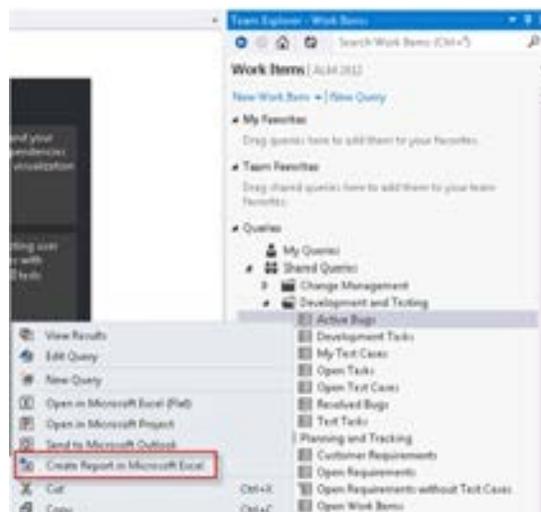


Figure 25 : Génération de rapport à partir de requêtes

L'utilisation en standard d'outils d'analyse multi dimensionnel tel que Power Pivot est également possible du fait de la compatibilité de TFS avec l'intégralité de la stack technique SQL Server 2008.

La publication des rapports pourra se faire au travers de SharePoint par l'élaboration de Dashboard composés de Web Part affichant des rapports Reporting Services ou des rapports Excel Services comme montré ci-dessous :

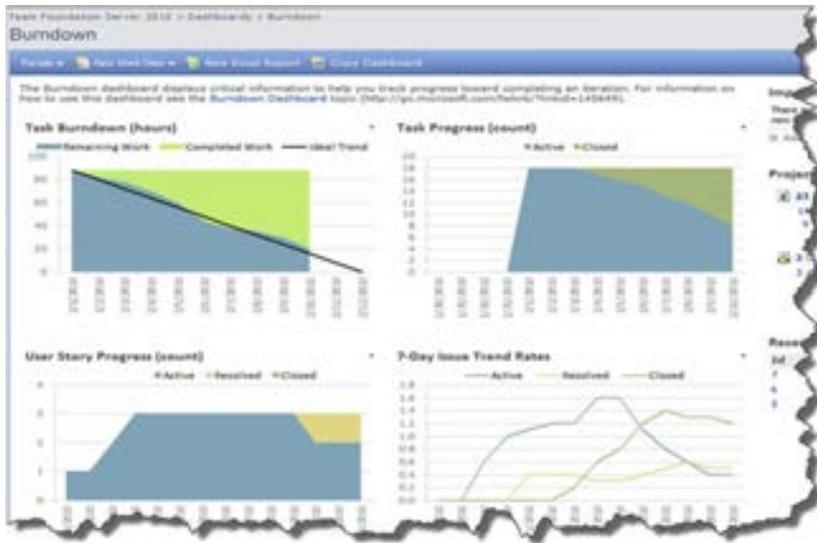


Figure 26 : Dashboard Burndown



Figure 27 : Dashboard Qualité

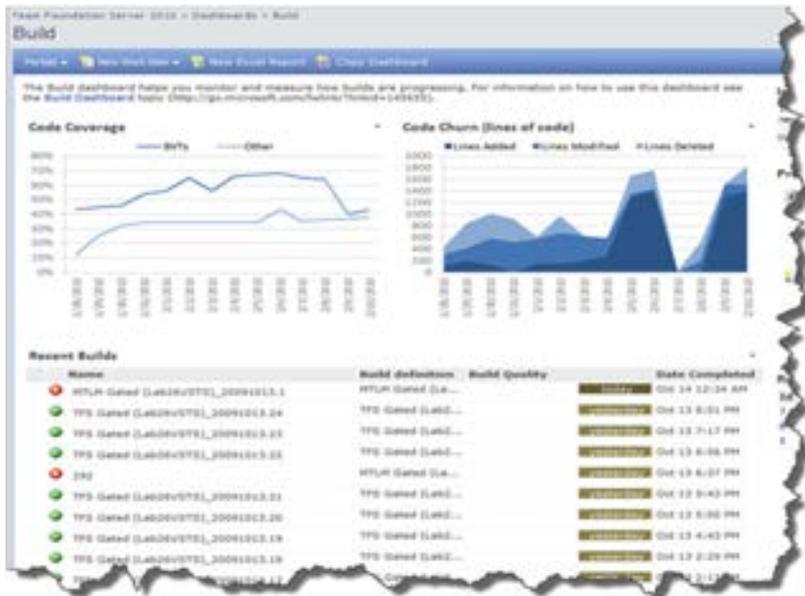


Figure 28 : Dashboard Build

Figure 29 : Exemple de Dashboard de rapports dans SharePoint

Program and Portfolio Management : TFS 2012 vient en standard avec une intégration avec l'EPM Project Server 2010 ou 2007 SP2 offrant des fonctionnalités de consolidation niveau entreprise pour la gestion des ressources, de Programmes et ou Portfolio au-dessus de TFS 2012 qui représente le suivi opérationnel projet par projet. Ci-dessous les fonctionnalités apportées par cette plateforme :



Figure 30 : Périmètre fonctionnel Project Server 2010

Dans les grandes organisations, la planification, l'arbitrage des ressources, le suivi budgétaire et le Reporting consolidé constituent un enjeu stratégique. Les fonctions EPM viennent avantageusement compléter l'offre TFS par la capacité à consolider les données de l'ensemble des projets gérés par TFS, mais également proposer toute la gestion amont avec le Portfolio Management et apporter au travers du BI Center des possibilités de Reporting consolidés et des KPI projets non disponibles dans TFS. Cette brique permet incontestablement d'intégrer les artefacts de développements de solutions informatiques avec la stratégie globale IT des grandes entreprises.

Une plateforme orientée services : TFS s'appuie sur une architecture SOA basée sur des web services métiers, et expose une API avec laquelle il est simple de développer pour exposer de nouveaux services étant soit des extensions de l'existant soit de nouvelles fonctionnalités, ci-dessous une vue plus détaillée de cette architecture fonctionnelle :

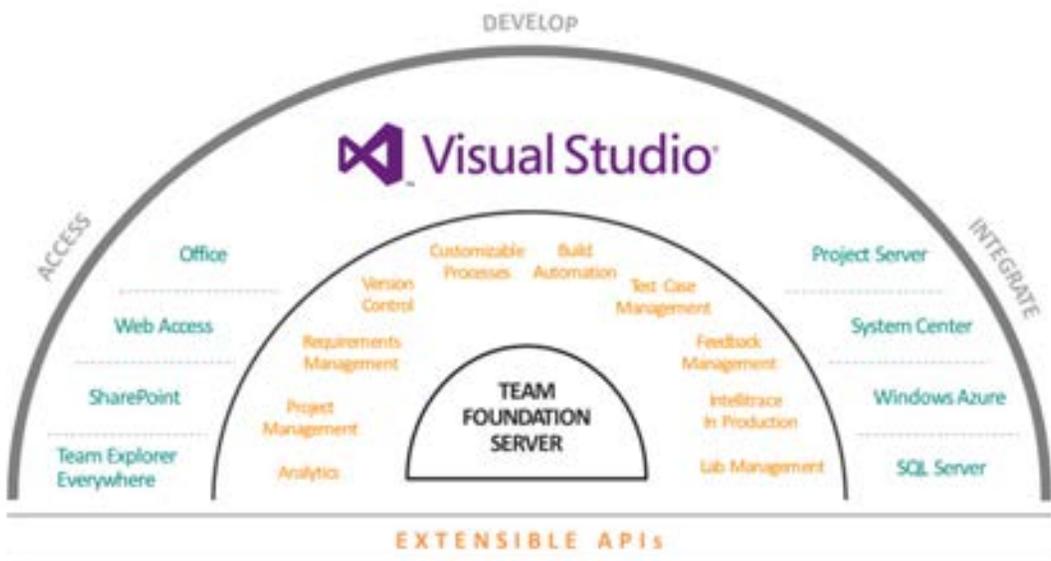


Figure 31 : Architecture Fonctionnelle de TFS

Parmi les services exposés coté serveur, on retrouve :

La gestion de projet

La gestion de projet outillée dans TFS 2012 permet d'englober l'ensemble des phases d'un projet en passant par l'initialisation avec la création d'un WBS projet pour fixer le périmètre ou scope du projet après la phase de collecte des exigences, également le pilotage et la partie contrôle avec la publication des tableaux de bord connectés directement aux métriques de l'usine logicielle permettant au chef de projet de fournir des rapports d'avancement en des temps records, ce qui d'ailleurs lui permet d'investir son temps sur des tâches à haute valeur ajoutée plutôt que sur des tâches fastidieuses et répétitives d'élaboration de rapports d'avancement manuels. Les artefacts suivis seront par exemple le consommé, le

reste à faire et le statut des différents éléments du WBS projet. TFS 2012 intègre également des notions d'itérations pour découper un projet en phase d'implémentation et d'Area Path pour la catégorisation technico fonctionnelle des éléments de travail. Ces deux éléments vont fournir des axes d'analyse pertinents pour la partie Reporting notamment.

TFS 2012 instancie des projets sous la forme de Projet d'Equipe, ces instances sont créées à partir de modèle de processus basés sur des templates méthodologiques tels qu'Agile, Scrum, ou CMMI en standard. Ces projets d'équipes vont exposer 5 types d'artefacts : Les Work items, les documents projet, les rapports, les Builds et le Contrôle de code source. La nouvelle version de Team Explorer fonctionne en mode web avec une page d'accueil, des boutons de navigation forward / backward. On notera également l'adjonction d'un lien My Work.

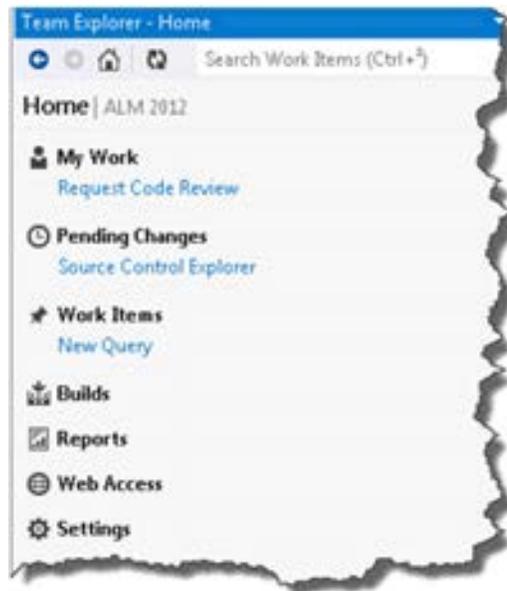


Figure 32 : Artefacts composant un Projet d'équipe vu dans Team Explorer

On notera le lien My Work qui permet de gérer son en cours de travail avec le travail en cours, le backlog de tâches disponibles, les demandes de Code Reviews ainsi que les sessions de travail suspendues disponibles. Cette dernière fonctionnalité (Suspend Resume) permet à un développeur de sauvegarder l'état de son environnement de travail à un instant t via l'option suspend pour par exemple adresser une urgence en production, et pouvoir restaurer plus tard son environnement à l'identique via la fonction resume. Toutes les fenêtres ouvertes sont retrouvées, les points d'arrêts jusqu'à la position du curseur de la souris.

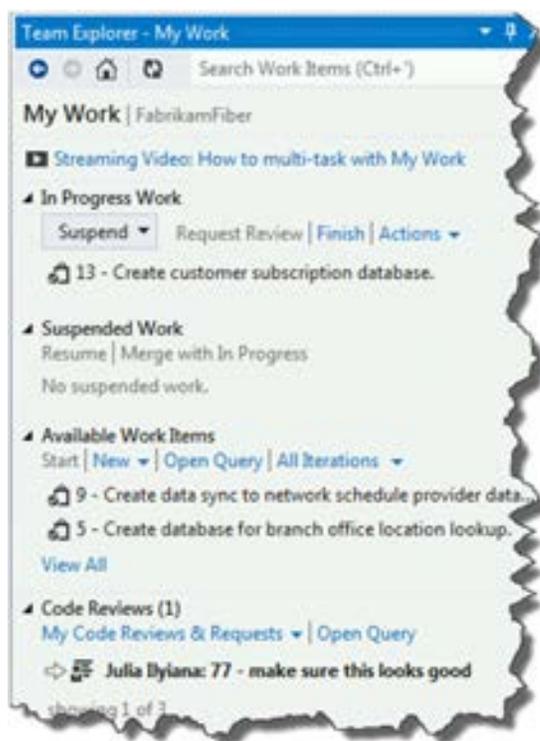


Figure 33 : Nouvelle fonction My Work

Les work items ou éléments de travail sont typés et extensibles, pour CMMI, les types sont :

- Requirement
- Task
- Risk
- Issue
- Test Case
- Bug
- Change Request
- Review

Pour chaque work item, le modèle se compose de champs, d'un layout (disposition des contrôles sur l'écran) et d'un workflow. Ces éléments sont simplement extensibles grâce à un outil nommé Process Template Editor. Cet outil va permettre de modéliser n'importe quelle méthodologie afin de s'adapter au mieux à votre organisation, ci-dessous, le designer permettant de modifier le workflow d'un Work Item :

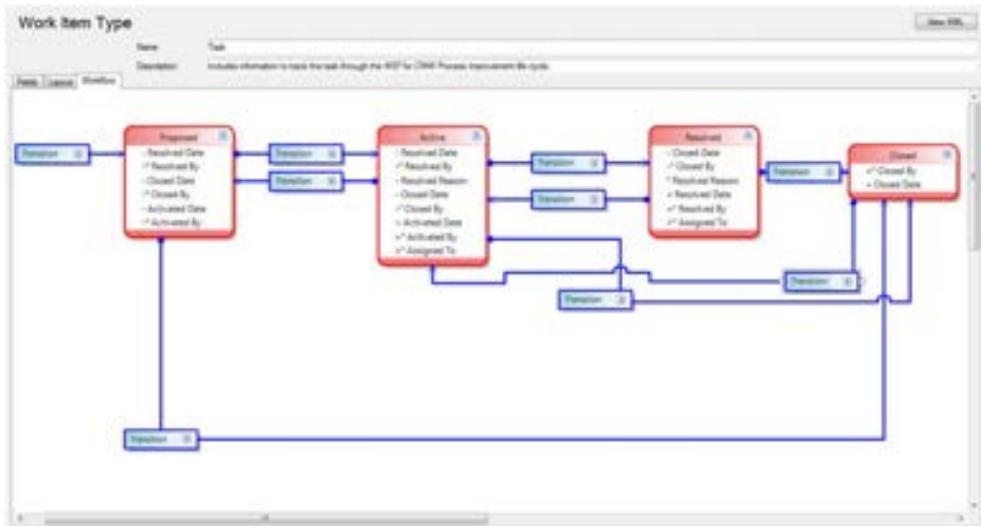


Figure 34 : Designer de Workflow pour Work Item

La plateforme fournit également un mécanisme de requête avancé avec les **Work Items Queries**, permettant de requêter le repository de Work items y compris avec des requêtes hiérarchiques et ce simplement à l'aide d'un Générateur de requêtes graphique. Ces requêtes sont simples à mettre en œuvre et à la portée de tous les utilisateurs en tant que requêtes personnelles ou requêtes d'équipes et pourront être utilisées pour créer des tableaux Excel ou alimenter des rapports à la volée, idéal pour un Reporting opérationnel.

TFS 2012 intègre un mécanisme de roll up permettant de consolider les données de Work items hiérarchiques jusqu'au niveau projet mais aussi de comparer l'actuel aux éléments d'une Baseline ce qui permet de mesurer le niveau de performance projet pour la partie planning et budget et anticiper tout dérapage.

Le rapport ci-dessous montre que l'on dépasse légèrement le budget initial :



Figure 35 : Exemple de rapport comparant le planifié avec le réel

La gestion du changement, la gestion des risques, le suivi du niveau d'anomalies sont autant d'éléments que le chef de projet pourra suivre au travers de rapports et Dashboard sans effort de collecte d'information, puisque l'ensemble des données collectées par la plateforme sont stockées dans un data warehouse puis processées dans le cube Olap permettant la création de rapports avancés.

La gestion des exigences

La gestion des exigences est effectuée en reliant des documents de spécifications aux exigences et tâches définies dans TFS. Ce principe s'appuie sur le fait qu'un Work item (élément de travail) est un artefact projet qui a la capacité d'être relié à d'autres artefacts projets tels que d'autres Work items (Requirements, risques, tâches, tests etc.) mais aussi des documents, diagrammes d'architecture ou spécifications détaillées, mais encore des hyperliens etc... Cette structure est au cœur de la stratégie de gestion des exigences avec TFS 2012. Visual Studio 2012 permet de gérer pour les exigences ou User Stories des Storyboards pour décrire graphiquement les écrans d'une application, cette fonctionnalité a été intégrée à PowerPoint grâce à un add ins ce qui permettra de rapidement et simplement décrire une application et relier ces éléments aux spécifications correspondantes dans TFS. L'avantage de cette nouvelle fonctionnalité est une prise en main extrêmement rapide et une productivité accrue.

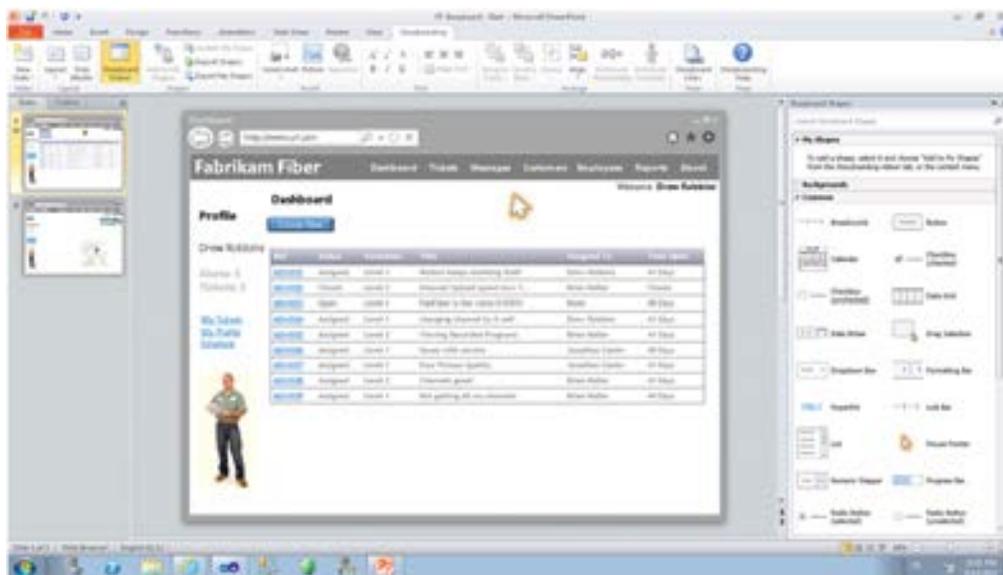


Figure 36 : Outil de Storyboarding de Visual Studio 2012 intégré à PowerPoint

L'outil de Storyboarding permet via une fonction d'image clipping d'intégrer des captures d'écrans d'une application existante pour par exemple spécifier une évolution. On notera également la possibilité de créer des cinématiques d'écran pour illustrer le comportement dynamique d'une application.

Au-delà du Storyboarding, la plateforme va permettre le suivi des changements qui impactent les exigences, mais également de rattacher aux exigences les artefacts d'implémentation tel que tâches, code source d'implémentation et les tests garantissant un suivi très fin du cycle de vie des exigences ou Requirement.

La Gestion de Configuration avec Le contrôle de code source

La plateforme ALM TFS 2012 expose des fonctions avancées de gestion de configuration, le code source est stocké dans des branches, la synchronisation entre les différentes branches est effectuée via de puissantes fonctions de fusion pour livrer une version, un hot fix ou une nouvelle fonctionnalité. Parmi les scénarios de branches on pourra implémenter des scénarios personnalisés permettant par exemple d'isoler le flux de développement sur une branche Dev, un flux Main pour les tests et des branches de release majeures, les labels permettront de tagger des versions mineures sur les différentes branches ce qui permettra de retrouver très facilement n'importe quel niveau de version d'une application. De plus, en standard, TFS propose un puissant mécanisme d'historisation permettant de tracer toutes les modifications effectuées sur le code. L'archivage du code source repose sur 2 mécanismes : les opérations d'archivage ou check in conventionnelles mais également le mécanisme de mise sur étagère ou shelve permettant à un développeur de stocker du code en cours de développement sur le serveur sans l'intégrer dans la branche donc sans déclencher de Build automatisé en cas d'intégration continue. Cette technique est très intéressante notamment pour mettre à disposition du code pour Revue par exemple, ou encore développer plusieurs scénarios sans créer de branches coté serveur.

Ci-dessous un exemple de scénario tel qu'il peut être implémenté sous TFS :

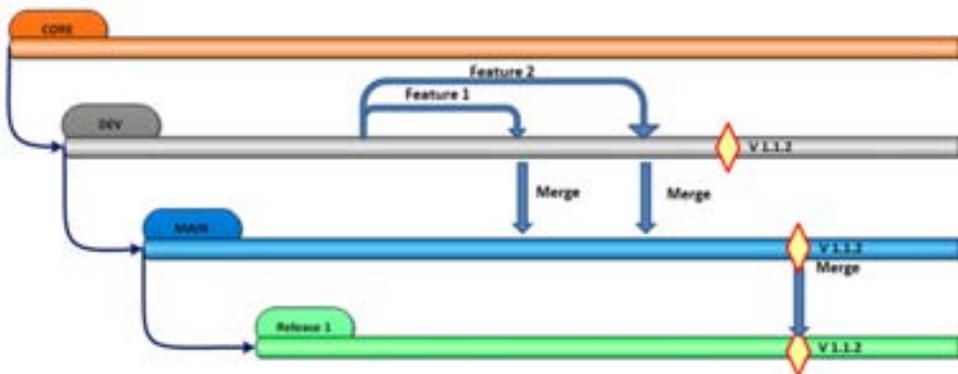


Figure 37: Exemple de Modèle de Branche

Chaque branche peut de plus contenir bien évidemment des sous dossiers sur lesquels on pourra également utiliser les fonctions de branches, labels et fusion également. A noter qu'il est possible d'effectuer des livraisons ou fusion au niveau fichier, dossier ou branche.



Figure 38 : Exemple de structure de Branches et Sous dossiers

Visual studio propose un afficheur en WPF permettant de suivre les structures des branches ainsi que les modifications effectuées sur les branches en mode hiérarchique ou temporel.



Figure 39 : Visualiseur de hiérarchie de branches

Ci-dessous une visualisation temporelle du cycle de vie des opérations de fusion sur les branches de développement, d'intégration et de release. Cette interface graphique permet également de procéder à la livraison de fonctionnalités par Drag & Drop de branche à branche :

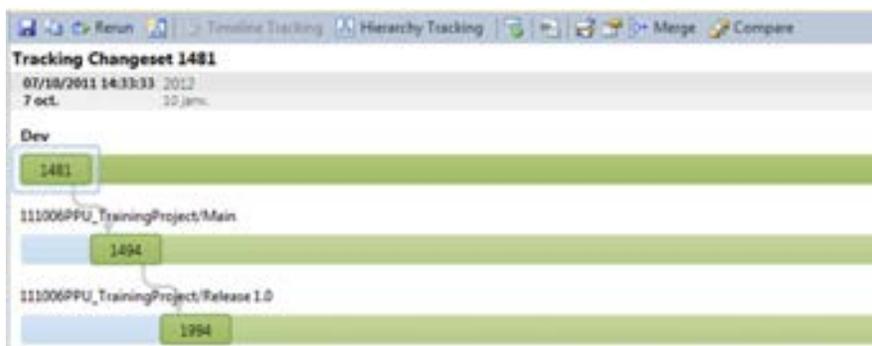


Figure 40 : Suivi des opérations de fusion d'un Changeset sur la ligne Temps

Un Centre de qualité logiciel avec le module de gestion des tests

La qualité est depuis quelques années au centre des préoccupations de Microsoft dans le cadre de sa proposition ALM. Avec Test et Lab Center, TFS embarque un véritable centre de qualité logicielle, et nul n'ignore l'impact de la qualité sur le taux de réussite des projets.

TFS permet d'intégrer de manière puissante l'activité de test dans le cycle de vie du projet à la fois pour la partie planification d'une campagne de test mais également la phase d'exécution et de Reporting sur l'exécution de ces tests d'une part et l'impact en termes de qualité sur les implémentations réalisées. Cette plateforme va permettre de mettre en œuvre l'intégralité de type de tests en partant du plus détaillé vers le plus général : des tests unitaires jusqu'aux tests d'acceptances utilisateur (UAT) en passant par les tests de charge.

Les points forts de cette offre sont :

- La capacité d'automatisation des tests pour la partie tests manuels avec de surcroît la notion de « Coded UI Tests » qui sont des tests automatisés d'interface utilisateurs avec la validation d'assertions ce qui signifie que le test s'auto-valide ou invalide en fonction des retours de l'interface.
- La capacité de coupler les tests unitaires et automatisés au processus de Build pour valider le niveau de qualité via des rapports et également augmenter la productivité en déportant une partie des tests manuels à l'automate de Build.
- La capacité via Lab Management de provisionner des environnements de Test avec les dernières versions des binaires applicatifs déployés et ce en s'appuyant simplement sur un processus de Build.
- La capacité de traiter la création de bug directement à partir de l'environnement d'exécution des tests ce qui permet de générer des bugs reproductibles grâce à un mécanisme de collecte, *cette fonctionnalité est d'ailleurs inédite dans l'offre des outils de test, la collecte peut contenir : des captures d'écran, le recording vidéo et ou audio de la session de test, la trace IntelliTrace, les éléments de l'événement log, les informations systèmes de l'environnement de tests, les étapes du test ainsi que les résultats.*
- La capacité d'analyser les tests impactés, à la fois pour les tests manuels que les tests unitaires. Sur un projet il est très courant d'avoir des centaines de tests unitaires et des volumétries analogues pour les tests manuels. Au départ tout va bien, mais chaque fois que le code est modifié, il est très difficile sans outillage de savoir quels sont les tests qui ont été impactés et donc qu'il faudra rejouer pour s'assurer qu'aucune régression n'affectera l'application. Ce point est parfaitement géré avec Visual Studio tant pour les tests unitaires que manuels ce qui constitue un gain énorme de productivité puisque seuls les tests impactés seront rejoués pour validation.

- La capacité à effectuer des Tests exploratoires
- La capacité à soumettre des demandes de Feedbacks Client

Test Manager se présente sous la forme d'un client dédié avec une interface client lourd de type WPF.

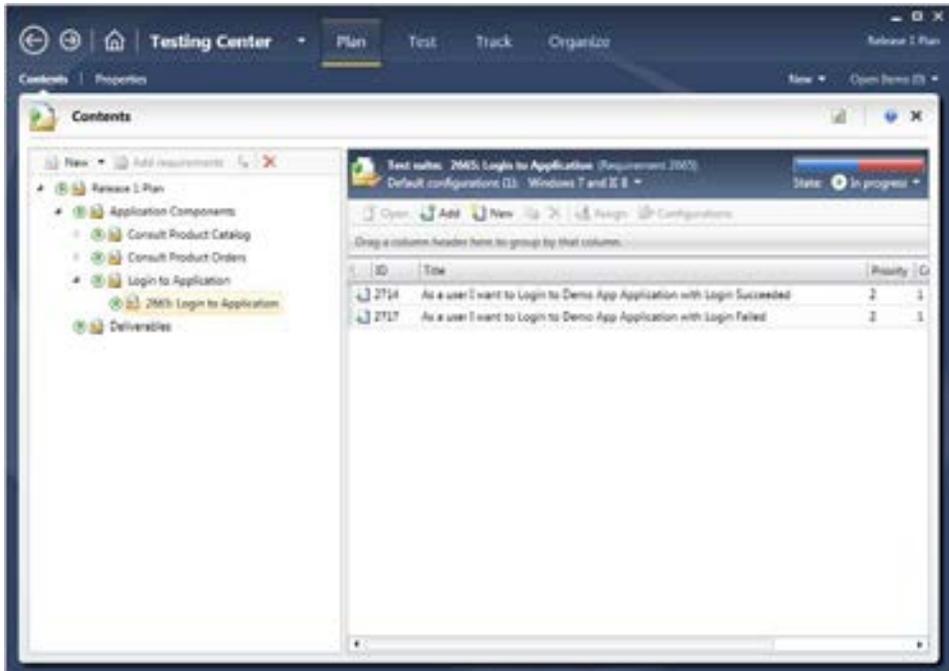


Figure 41 : Client Test Manager

Dans la nouvelle version, chaque étape de test peut s'afficher sur plusieurs lignes et contenir du contenu formaté (police, style et couleur).

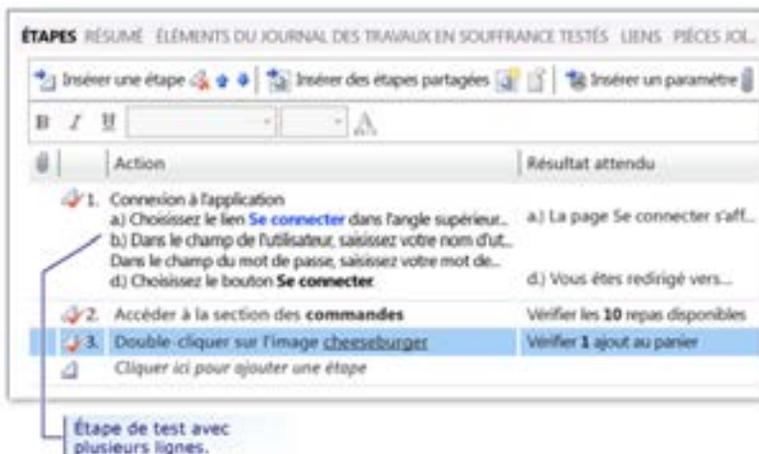
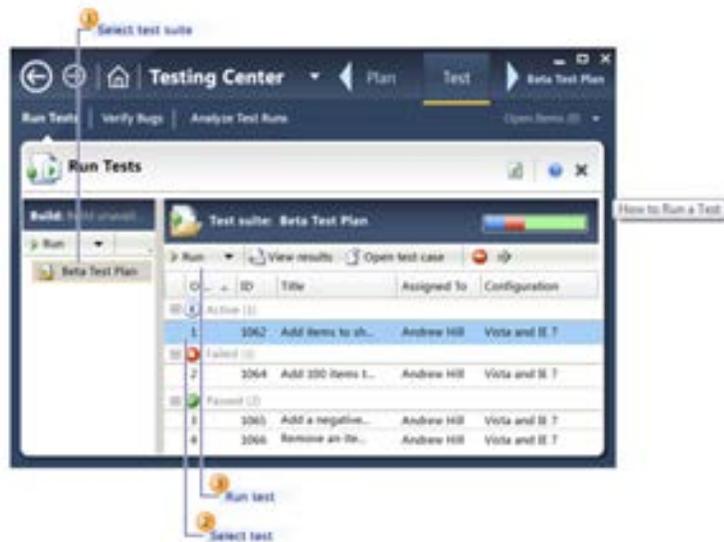


Figure 42 : contenu Texte riche et multi ligne pour les étapes de cas de test

Le mode Run permet de consulter le résultat d'exécution et de lancer l'exécution d'un ou plusieurs Tests :



Test Manager propose également un outil pour l'exécution des tests Test Runner apportant un panel de fonctionnalités de premier ordre pour améliorer la productivité du Testeur. Le testeur effectue chaque étape du cas de test, indique le résultat, on note le bouton Play pour jouer une automatisation sur une ou plusieurs étapes du cas de test.

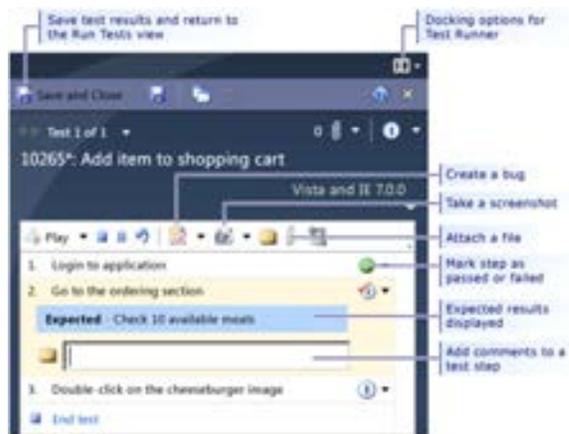


Figure 43 : Test Runner - une collection complète d'outils pour l'exécution des Tests

On a au niveau de Test Runner les possibilités suivantes :

- Possibilité d'automatiser toute ou partie des étapes manuelles.
- Utilisation de paramètres pour éviter de ressaisir à chaque fois les mêmes données lors du test. Les paramètres peuvent être injectés par copier-coller dans les IHM

testées évitant les erreurs de saisies et accélérant les phases de test.

- L'ouverture de bug ainsi que la prise de capture d'écran sont intégrées à l'environnement d'exécution.
- Une fonction permettant d'attacher des données riches telles que l'enregistrement vidéo de la session de tests, les données IntelliTrace avec les exception Data qui permettront aux développeurs de résoudre beaucoup plus rapidement le bug avec également les threads d'exécution, la stack Trace, les informations systèmes et les étapes du cas de test avec les résultats.

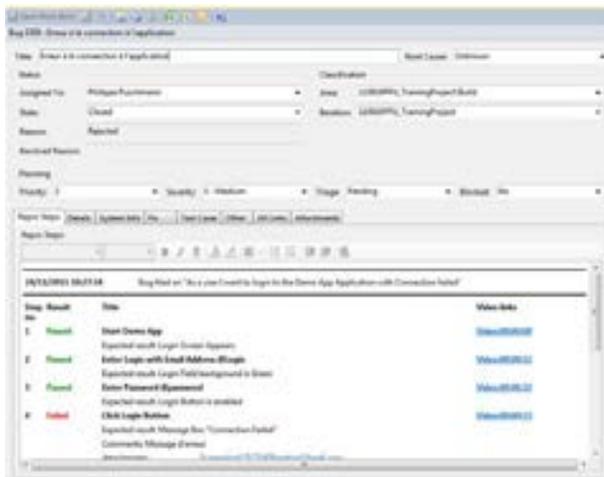


Figure 44 : Bug affichant les étapes, les résultats et liens vers les captures vidéo

Enfin, on notera surtout enfin un excellent module de reporting avec filtrage par suite.



Figure 45 : Rapport de résultat de tests par suite

Le développeur peut également avoir accès aux données d'intellitrace qui ont été collecté dans le bug soumis :

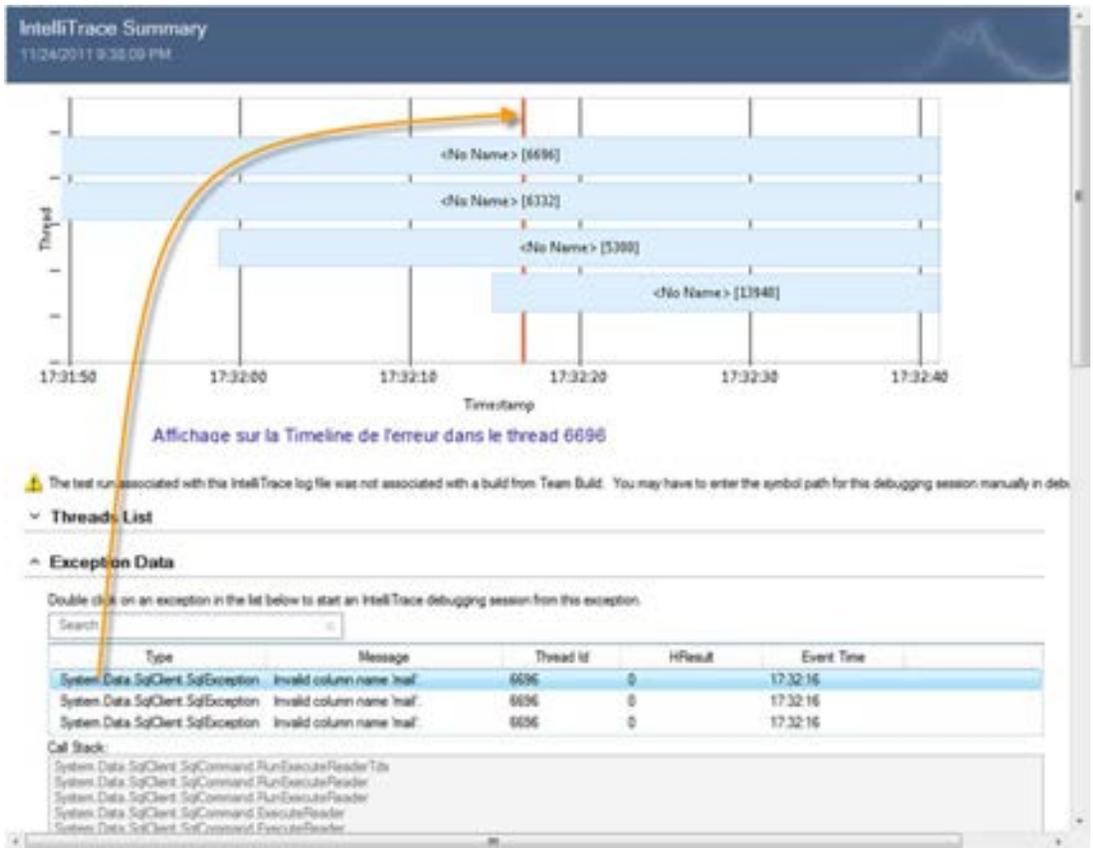


Figure 46 : Exemple de données Intellitrace uploadées automatiquement par le Test Agent dans le bug

En double cliquant sur le thread 6696, Intellitrace permet d'afficher la page de code en cause et surligne en rouge la ligne qui a déclenché l'erreur, un temps énorme gagné pour la résolution d'anomalies, à noter que le testeur de son côté n'a fait que jouer le test, alors que toutes les données capturées à part la capture d'écran sont le fait du Test Agent qui effectue la collecte des données et le téléchargement des éléments dans le Bug en tant que pièces jointes.

Un outil de Test Exploratoire :

Cet outil est en fait une extension de Test Manager, permettant à un testeur d'initier une session de test sans script avec la possibilité de transformer cette session en cas de test. Cette dernière fonction est absolument bluffante dans le sens où l'outil de capture sera capable de transformer les différentes étapes des tests avec la définition de l'action en automatique. Par exemple :



Figure 47 : Interface de Test exploratoire

Un outil de Feedback Client

Cet outil est au cœur des processus agiles et va permettre d'obtenir simplement et rapidement des feedbacks de la part du client, avec une interface permettant d'entrer des commentaires avec des captures d'écran, un système de rating basé sur une jauge de 5 étoiles mais aussi potentiellement des commentaires audio ou même du recording vidéo pour fournir un feedback sur une cinématique d'écran.

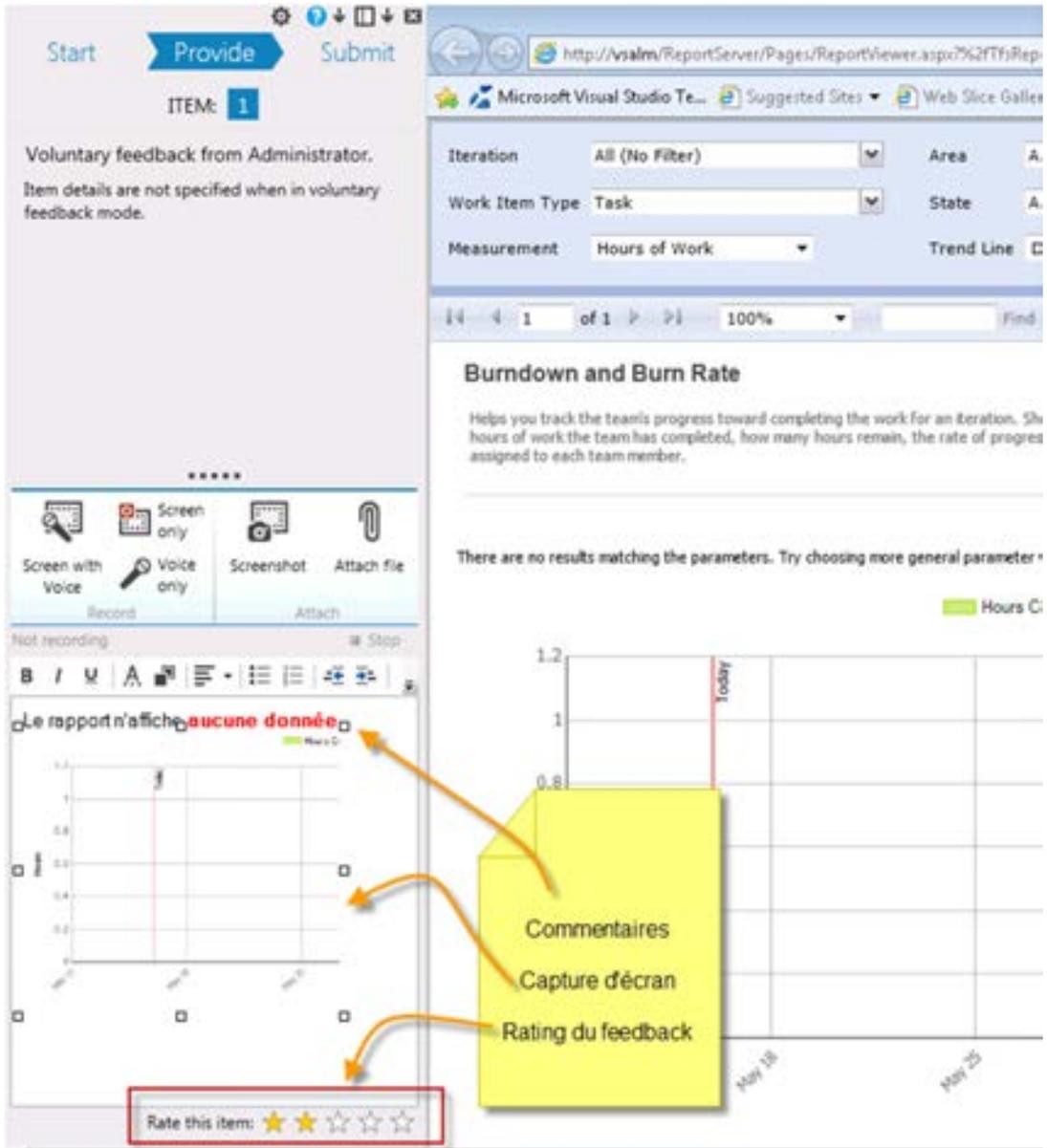


Figure 48 : Interface du Client Feedback

La gestion des builds

La nouvelle version de Team Build basée sur le moteur dernière génération Microsoft Workflow Foundation 4.0 s'impose comme un outil exceptionnel pour l'ensemble des concepts d'intégration continue pour une usine logicielle. Cette nouvelle version permet en plus des options standards d'Analyse de qualité du code, d'exécution de listes de tests unitaires avec

le calcul de la couverture de code, de personnaliser simplement un processus de Build avec par exemple le rajout d'activités de déploiement en fin de Build ou encore la génération de release note automatique.

Les possibilités de personnalisation sont énormes puisque le moteur vient avec en standard des activités pour team Build, des activités pour Lab Management mais également l'intégralité des activités de workflow standard et la possibilité de créer ses propres activités pour adapter le workflow de Build à son besoin pour y rajouter par exemple des activités de déploiement automatisé par exemple.

A noter BRDLite, créée par l'équipe ALM Rangers pour Build Release Deploy qui fournit des templates de build permettant le déploiement automatisé des types de solutions suivantes :

- SharePoint 2010/2013
- Office 365
- CRM
- Azure
- Click Once
- ASP.NET

Les Builds sont au centre du processus d'intégration continue, ce puissant outil va permettre en quelques clics de souris de mettre en place les types de Builds nécessaires au cycle de développement d'une application, en couplant de l'analyse de la qualité du code, l'exécution de tests unitaires, le calcul de la couverture de code et enfin la génération de rapports. D'importantes améliorations ont été apportées à la nouvelle version avec une possibilité de Load Balancing entre différents Controllers pour supporter la charge, la possibilité de prioriser, mettre en attente ou annuler des Builds en cours d'exécution.

Un des facteurs clés de la plateforme de Build se trouve dans les possibilités d'extension du processus de Build. Basé sur Workflow Foundation 4.0, tout processus de Build peut être édité dans un Designer XAML et modifié ou étendu. Il est effectivement possible de développer des activités personnalisées en .Net et d'utiliser ces nouvelles activités par exemple, la compression zip de fichiers, la manipulation de Work Item ou toute autre action et consommer ces activités dans le workflow du processus de Build. Les extensions les plus communément implémentées sont des actions de déploiement automatisées, la génération d'installateurs ou encore la génération d'artefacts de release tel que les tags de versions d'assemblies ou la génération de release note. Ci-dessous une illustration du designer de processus de Build :

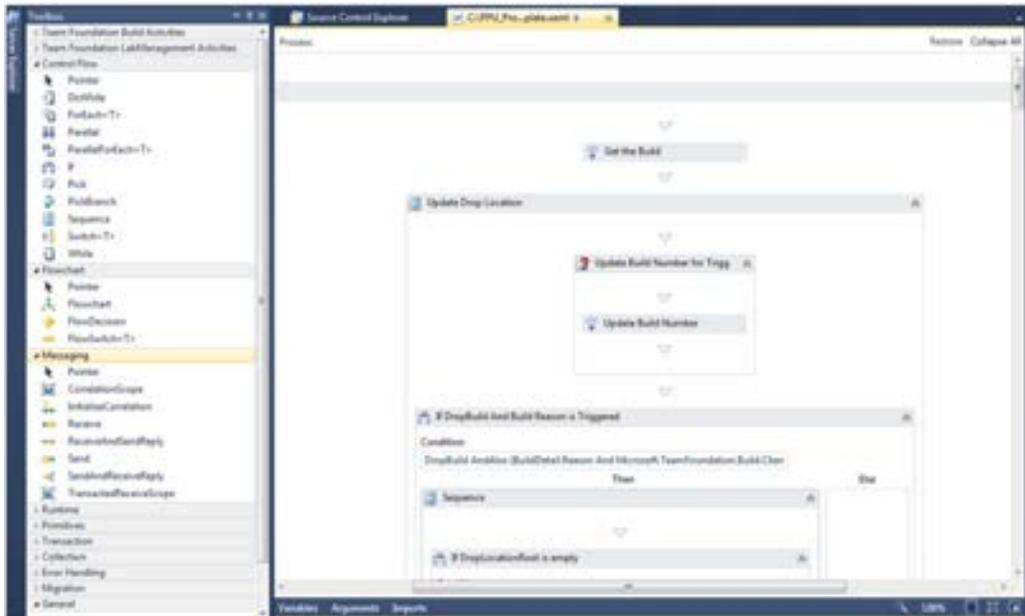


Figure 49 : Designer pour la personnalisation d'un processus de Build

Les Builds à mettre en place dans le cadre d'un projet sont généralement au nombre de trois :

Les Builds d'intégration continue : Un Build est déclenché chaque fois qu'un développeur check in du code dans TFS. Ce mécanisme permet de s'assurer que le Build n'est pas cassé grâce à des mécanismes de notification temps réel.

Les Nightly Builds : Ces Builds vont permettre de contrôler toutes les nuits non seulement le fait que l'application compile mais également le contrôle de la qualité du code, le résultat des tests avec les métriques de couverture de code. Le tout étant publié au travers de plusieurs rapports, par exemple le rapport journalier avec les tests et le niveau de couverture ci-dessous un exemple permettant de voir qu'aucun test n'est joué mais que le processus de Build est stable :



Figure 50 : Exemple de rapport de Build avec ligne de temps et niveau de qualité

Les Builds de Release : Ces Builds sont exécutés à chaque livraison d'un release manuellement, ce qui permet de communiquer sur le niveau de qualité de la version mais également de générer un release note incluant les nouvelles fonctionnalités incluses depuis la dernière version ainsi que les bugs fixés

Coté client, TFS démontre une implémentation ALM avec une prise en compte des utilisateurs en offrant un large panel d'outils adaptés et dédié évitant d'imposer Visual Studio à l'ensemble des parties prenantes. On a donc pour la partie cliente :

Client Visual Studio 2012: L'IDE de développement apportant aux développeurs un outil de premier ordre permettant d'améliorer la productivité et la qualité des développements. Microsoft propose plusieurs versions pour Visual Studio 2012 comme indiqué ci-dessous, la version de base étant la version Professional, pour chaque version supérieure sont affichées les fonctionnalités en plus :

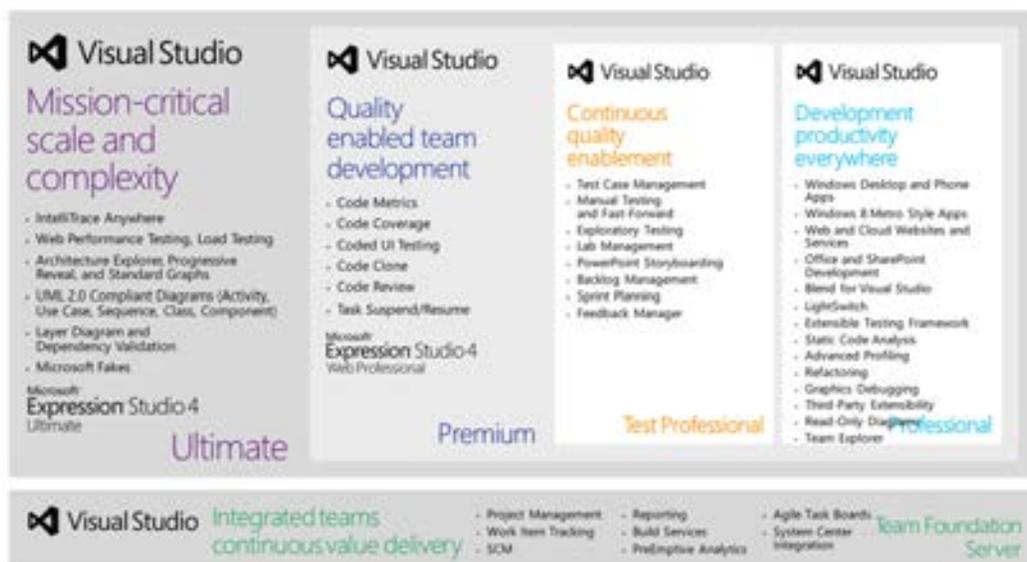


Figure 51 : Fonctionnalités par version de Visual Studio 2012

Client SharePoint 2010 : Plateforme collaborative par excellence, dédiée au stockage et suivi de version de l'ensemble de la documentation du projet, l'outil se décline sous plusieurs formes : la version Gratuite : SharePoint Foundation et la version complète permettant notamment de s'intégrer avec Project Server. Cet outil client est à destination des parties prenantes fonctionnelles pour le stockage de l'ensemble des documents de spécifications par exemple. SharePoint sera également le client dédié à la publication de Dashboard grâce à la technologie des Web parts permettant la personnalisation de pages, ci-dessous un exemple de Dashboard publié sous SharePoint :



Clients Microsoft Office : La plateforme permet de connecter les différents outils Office à TFS via un mécanisme de Publish / Refresh.

Microsoft Excel et TFS pour manipuler en mode productivité les tâches, exigences, anomalies stockés dans TFS sous forme de Work Items ou encore produire des rapports personnalisés en quelques minutes grâce aux possibilités de connexion au Cube OLAP de TFS pour analyser les statistiques projets. Ces Dashboard Excel sont alors publiables sur SharePoint grâce à Excel Services, ceci est particulièrement intéressant pour une approche Agile des aspects Reporting. Ci-dessous un exemple de vue hiérarchique d'un WBS projet :

The image shows an Excel spreadsheet displaying a hierarchical Work Breakdown Structure (WBS) for a project. The spreadsheet has columns for 'Project', 'Task ID', 'Task Name', 'Task Type', 'Task Status', 'Task Priority', 'Task Owner', 'Task Start Date', 'Task End Date', 'Task Duration', and 'Task Cost'. The tasks are organized into a tree structure, with 'Project' at the top level, followed by 'Phase 1', 'Phase 2', and 'Phase 3'. The tasks are listed in a table format, with the 'Task Name' column containing the task details. The 'Task Status' column shows the current status of each task, and the 'Task Cost' column shows the estimated cost for each task. The spreadsheet is displayed in a grid view, with the 'Task Name' column highlighted in blue. The 'Task Status' column is also highlighted in blue, and the 'Task Cost' column is highlighted in blue. The spreadsheet is displayed in a grid view, with the 'Task Name' column highlighted in blue. The 'Task Status' column is also highlighted in blue, and the 'Task Cost' column is highlighted in blue.

Coté Tableau de bord, un exemple d'analyse de la production de code avec possibilité de drill down sur l'échelle de temps. On notera l'utilisation du formatage conditionnel avec Excel permettant de produire des rapports pertinents et très lisibles.

Lines Added	Antoine Barot	Damien Deroux	Fabrice Roux	Virginie Dubois	Arnaud Delan	Laurent Bert	Samuel Bouchamp	Mathieu Erzog	Nicolas Piot	Grand Total	
= 2008	75 128	82 736	55 290	17 745	107 468	8 699	88 820	57 652	35 187	5 535	813 983
> Week of 18 février	245	105	377	144	150		5 189	1 781	379	295	8 508
> Week of 25 février	2 274	6	2 510	505	2 348		15 898	821	741	233	24 139
> Week of 3 mars	10 339	11 430	9 895	892	23 483		1 241	348	2 188		58 437
> Week of 10 mars		56 688	2 268	388	82 428		18 529	9 001	778	84	122 494
> Week of 17 mars	9 401	3 486	2 570	14 317	1 520		998	4 814	3 063	1 186	38 343
> Week of 24 mars	337	975	524	368	712		941	381	178	336	4 034
> Week of 31 mars	1 182	9 779	2 057	1 587	283		418	9 825	10 818	2 835	34 034
> Week of 7 avril	2 481	4 705	3 014		2 380		110	1 134	700	3 300	17 880
> Week of 14 avril	2 194	3 410	18 812		88 040		1 875	1 246	1 296	900	95 518
> Week of 21 avril	1 474		2 748		348		38	3279	627	3 878	12 378
> Week of 28 avril	24 304		2 048				960	2 258	11 341	1 412	30 924
> Week of 5 mai		518	7 776		88 004		1 488	2 644	38	1 042	76 818
> Week of 12 mai	8 128		4 180		88 008		1 272	3 213	6 893	2 187	88 704
> Week of 19 mai	3 877		2 586		28 180		1 564	4 348	8 025	10 110	51 482
> Week of 26 mai	21 958		2 449		122 380		584	21 011	259		178 238
Grand Total	75 128	82 736	55 290	17 745	107 468	8 699	88 820	57 652	35 187	5 535	813 983
Average per day	1 083	1 237	717	267	9 300	NA	1 284	748	503	847	

Figure 52 : Exemple de tableau croisé dynamique pour l'analyse du Code Churn

Ou encore un tableau d'analyse des temps passés

Figure 53 : Exemple de tableau croisé dynamique pour l'analyse du consommé et du reste à faire

Développeurs	Nbre de Taches	Travail Complété	Travail restant
Antoine Barot	9	72,0	36,0
Damien Deroux	4	0,0	128,0
Fabrice Roux	11	85,0	116,0
Virginie Dubois	22	229,0	0,0
Arnaud Delange	22	137,5	24,0
Laurent Berthaux	2	20,0	28,0
Mathieu Erzog	25	184,3	4,7
Nicolas Piot	41	293,5	2,0
Nicolas Vezin	14	64,0	48,0
Nicolas Armand	32	317,03	0,0
Olivier Perrin	20	315,0	12,0
Philippe Sarrain	9	73,0	43,0
Samuel Dupuy	11	152,0	0,0
Xavier Lambert	14	160,4	0,0
Grand Total	236	2 102,73	441,7

Microsoft Project pour synchroniser un WBS projet, généralement constitué d'une arborescence hiérarchique de Work items de type Requirement et Task. Cet outil est incontournable pour tout chef de projet souhaitant transformer un planning projet en une collection hiérarchique de Work items et tâches en quelques secondes et basculer en mode suivi. Project 2012 vient avec des nouvelles fonctionnalités telles que la Time line permettant en un clic de souris d'afficher les grandes phases d'un projet à partir des données du diagramme de Gantt, cette représentation est connectée sur les données du diagramme de Gantt et est donc remise à jour automatiquement :

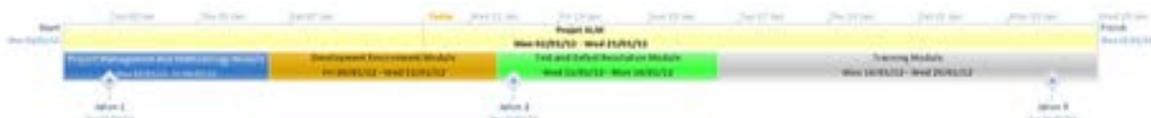


Figure 54 : Exemple de Timeline générée avec Project 2012

Une vue « Team Planner » permet également de gérer efficacement ses ressources graphiquement et de résoudre facilement les problématiques de surutilisation : un outil idéal pour une planification efficace.

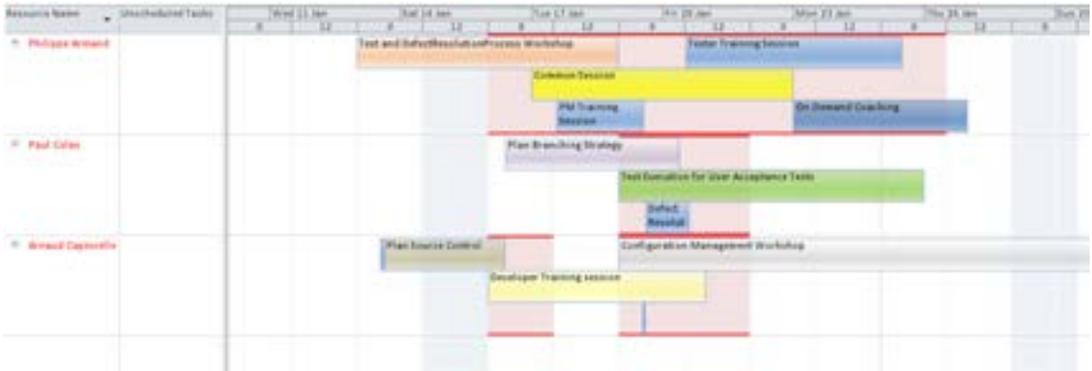
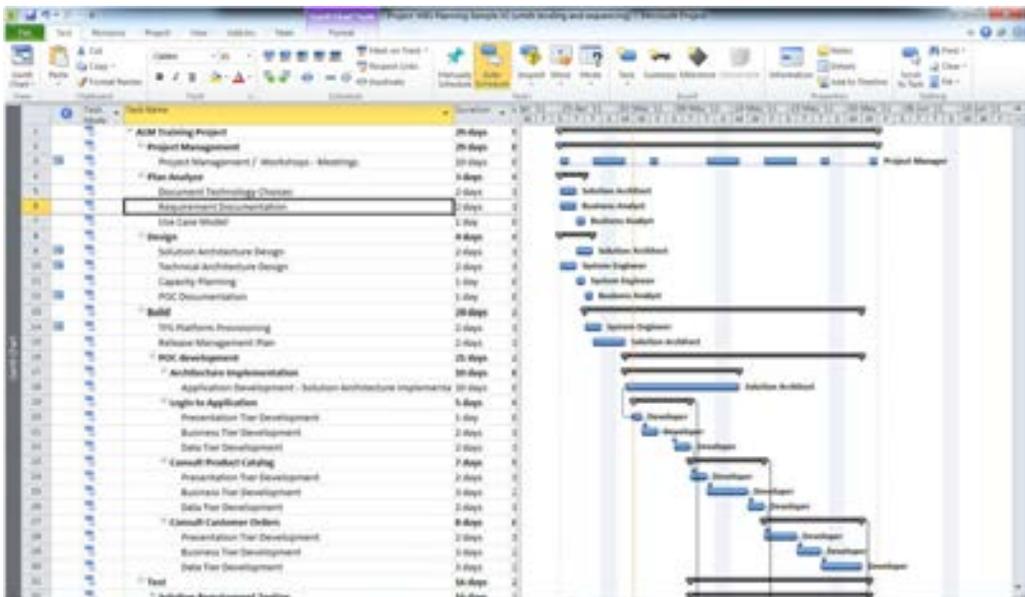


Figure 55 : L'outil Team Planner pour la gestion de l'affectation des ressources aux tâches

Enfin un mode manuel permet de faire des plannings façon Excel sans subir le recalcul du moteur de Gantt de Project, un outil idéal pour l'élaboration d'un WBS projet sans notion de date et ou de durée des tâches.

Et son équivalent dans TFS vu avec Visual Studio sous forme de Tâches et Requirements Ci-dessous, un planning projet publié dans TFS vu dans Microsoft Project.



Et son équivalent dans TFS vu avec Visual Studio sous forme de Tâches et Requirements.

ID	Type	Requirements	Work Item	Assigned To	Status	Progress	Original To	Task To
1	Operational	Requirement	Requirement		Proposed	0	0	Planned
2	Operational	Requirement	Requirement		Proposed	0	0	Planned
3	Operational	Requirement	Requirement		Proposed	0	0	Planned
4	Operational	Requirement	Requirement		Proposed	0	0	Planned
5	Operational	Requirement	Requirement		Proposed	0	0	Planned
6	Operational	Requirement	Requirement		Proposed	0	0	Planned
7	Operational	Requirement	Requirement		Proposed	0	0	Planned
8	Operational	Requirement	Requirement		Proposed	0	0	Planned
9	Operational	Requirement	Requirement		Proposed	0	0	Planned
10	Operational	Requirement	Requirement		Proposed	0	0	Planned
11	Operational	Requirement	Requirement		Proposed	0	0	Planned
12	Operational	Requirement	Requirement		Proposed	0	0	Planned
13	Operational	Requirement	Requirement		Proposed	0	0	Planned
14	Operational	Requirement	Requirement		Proposed	0	0	Planned
15	Operational	Requirement	Requirement		Proposed	0	0	Planned
16	Operational	Requirement	Requirement		Proposed	0	0	Planned
17	Operational	Requirement	Requirement		Proposed	0	0	Planned
18	Operational	Requirement	Requirement		Proposed	0	0	Planned
19	Operational	Requirement	Requirement		Proposed	0	0	Planned
20	Operational	Requirement	Requirement		Proposed	0	0	Planned
21	Operational	Requirement	Requirement		Proposed	0	0	Planned
22	Operational	Requirement	Requirement		Proposed	0	0	Planned
23	Operational	Requirement	Requirement		Proposed	0	0	Planned
24	Operational	Requirement	Requirement		Proposed	0	0	Planned
25	Operational	Requirement	Requirement		Proposed	0	0	Planned
26	Operational	Requirement	Requirement		Proposed	0	0	Planned
27	Operational	Requirement	Requirement		Proposed	0	0	Planned
28	Operational	Requirement	Requirement		Proposed	0	0	Planned
29	Operational	Requirement	Requirement		Proposed	0	0	Planned
30	Operational	Requirement	Requirement		Proposed	0	0	Planned
31	Operational	Requirement	Requirement		Proposed	0	0	Planned
32	Operational	Requirement	Requirement		Proposed	0	0	Planned
33	Operational	Requirement	Requirement		Proposed	0	0	Planned
34	Operational	Requirement	Requirement		Proposed	0	0	Planned
35	Operational	Requirement	Requirement		Proposed	0	0	Planned
36	Operational	Requirement	Requirement		Proposed	0	0	Planned
37	Operational	Requirement	Requirement		Proposed	0	0	Planned
38	Operational	Requirement	Requirement		Proposed	0	0	Planned
39	Operational	Requirement	Requirement		Proposed	0	0	Planned
40	Operational	Requirement	Requirement		Proposed	0	0	Planned
41	Operational	Requirement	Requirement		Proposed	0	0	Planned
42	Operational	Requirement	Requirement		Proposed	0	0	Planned
43	Operational	Requirement	Requirement		Proposed	0	0	Planned
44	Operational	Requirement	Requirement		Proposed	0	0	Planned
45	Operational	Requirement	Requirement		Proposed	0	0	Planned
46	Operational	Requirement	Requirement		Proposed	0	0	Planned
47	Operational	Requirement	Requirement		Proposed	0	0	Planned
48	Operational	Requirement	Requirement		Proposed	0	0	Planned
49	Operational	Requirement	Requirement		Proposed	0	0	Planned
50	Operational	Requirement	Requirement		Proposed	0	0	Planned

Client Web Access : Le module Web Access a été entièrement réécrit, avec un Look & Feel Metro et des interfaces asynchrones induisant des temps de réponse excellents, les écrans suivant montrent des exemples de cette nouvelle version de Web Access. La plateforme web permettant de connecter l'ensemble des parties prenantes aux artefacts projets, l'avantage de cet outil est de pouvoir démarrer sans installer quelque composant que ce soit sur les postes des utilisateurs favorisant donc le déploiement de la solution ALM notamment au sein des grandes organisations où l'installation de composant spécifiques sur les postes peut requérir un certain délai. La version 2012 se présente comme suit avec la page d'accueil se présentant sous la forme d'un Dashboard, reprenant la charte Graphique Metro avec les tuiles permettant de visualiser de l'information avant même de cliquer et visualiser le détail :

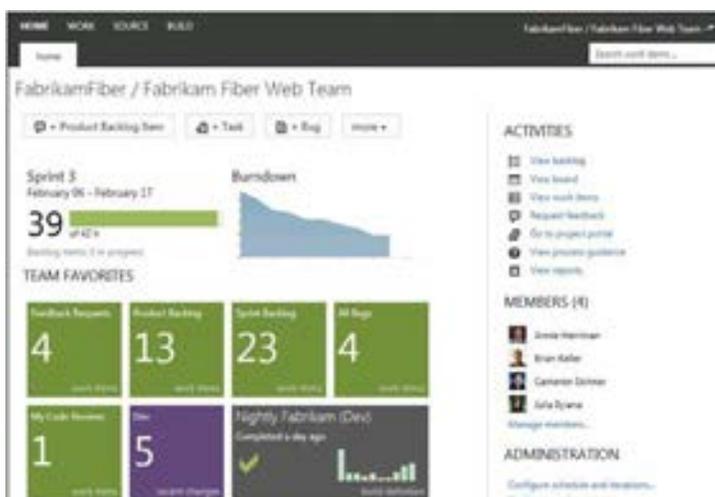


Figure 56 : Nouvelle Interface TFS Web Access

Cette nouvelle version propose également entre autre une vue Board pour visualiser les tâches en cours par exigence / User Story ou par membre de l'équipe sous la forme de post it, les tâches peuvent être manipulées en mode cliquer glisser, on notera le Burndown Chart en haut à droite. La cible de cette interface sera bien évidemment de grands écrans plats tactiles utilisables lors des Daily Scrum meetings par exemple :

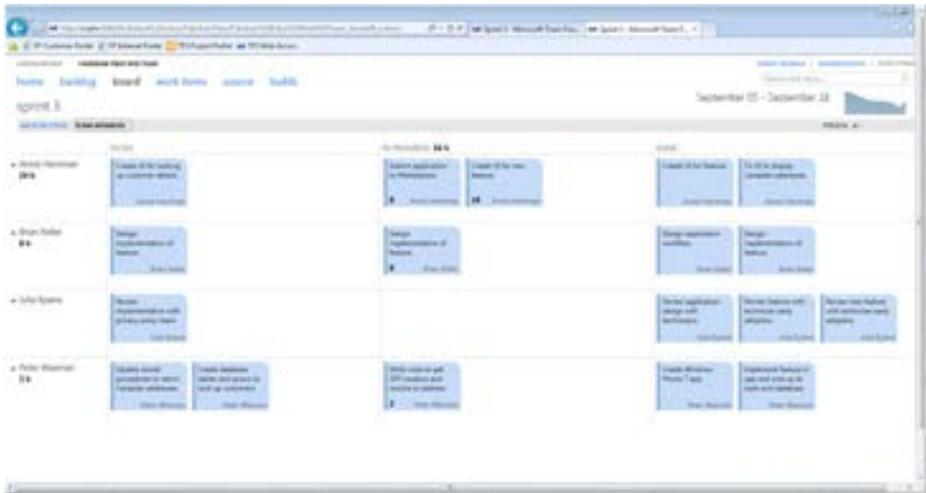


Figure 57 : Aperçu du Web Access Task board de Visual Studio 2012

Autre fonctionnalité du Taskboard, la possibilité d'éditer interactivement l'assignation de la ressource, ainsi que le reste à faire et le statut tel que montré ci-dessous :



Figure 58 : Manipulation interactive des post it

La vue Product Backlog permet la priorisation des items par simple cliquer glisser, de même que l'affectation d'une story à un Sprint.

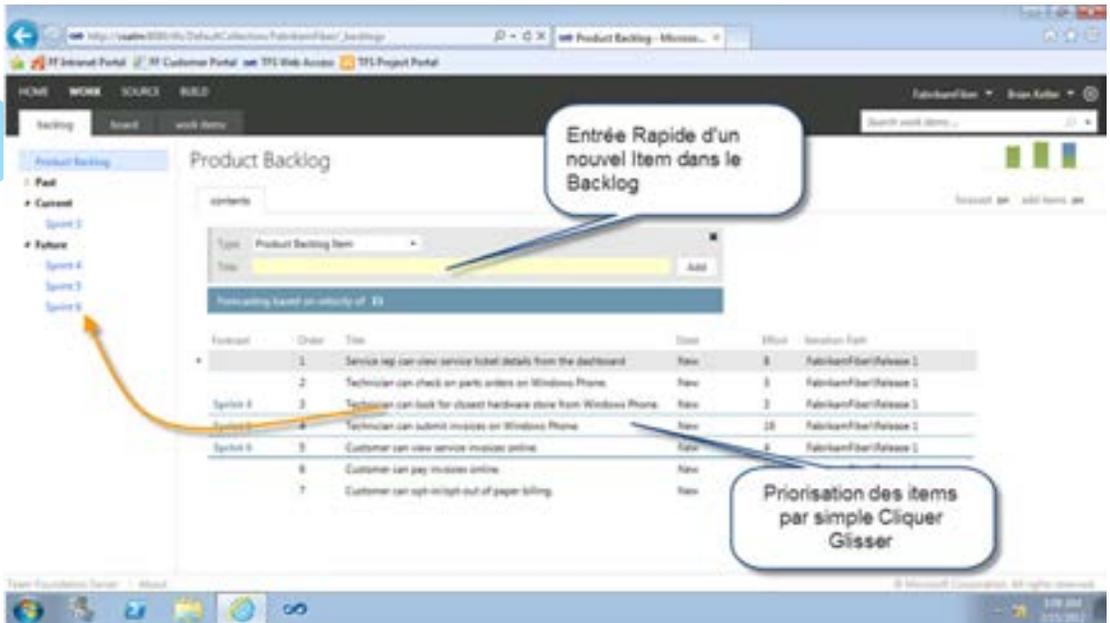


Figure 59 : Manipulation interactive du Product Backlog

L'outil permet également la planification Agile avec un affichage temps réel de l'état d'utilisation des ressources : rouge pour surutilisation et vert pour sous-utilisation. Un outil ergonomique, interactif et très efficace.

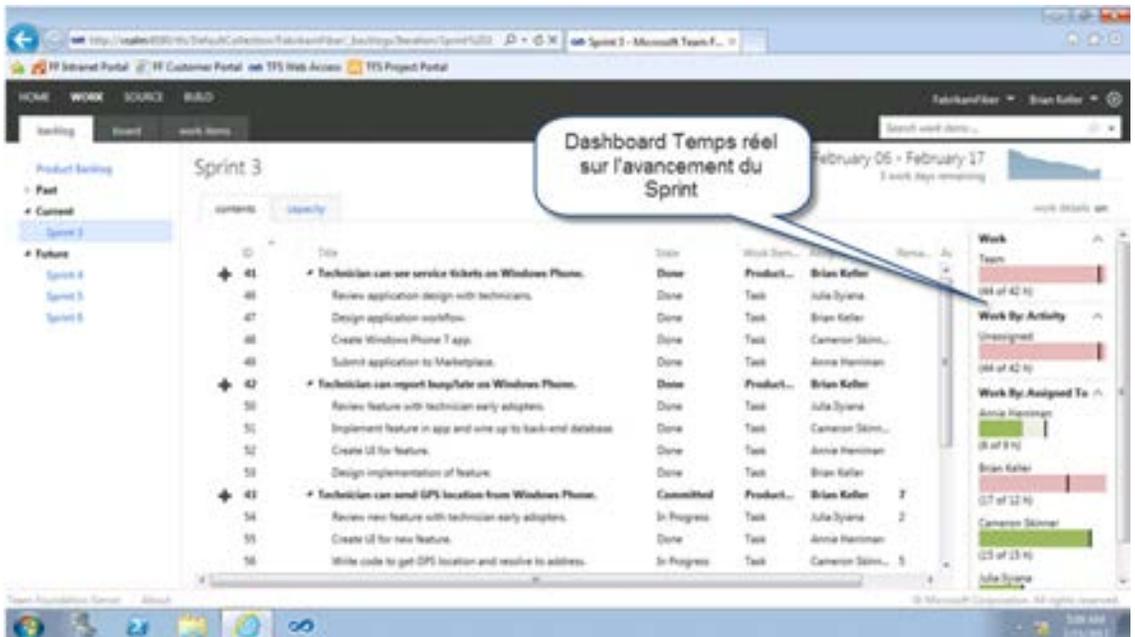


Figure 60 : Fonctionnalité de Planning Agile interactif

6.2. TFS 2012 Extensibilité et ouverture

Parmi les points particulièrement appréciables de cette plateforme, on notera les possibilités d'extensibilité, permettant d'étendre son périmètre fonctionnel. On notera quelques exemples intéressants :

La feuille de saisie des temps Avanade TFS Timesheet développée par la société **Avanade**, apportant la possibilité d'un suivi très précis des heures sous la forme d'un CRA conventionnel, cet outil développé en Silverlight permet une saisie des temps multi projets et multi collection avec également des fonctionnalités intégrées de Reporting et une synchronisation temps réel avec TFS. La TimeSheet dans cette nouvelle version permet l'édition des fiches de Work Items pour par exemple passer un statut à résolu directement à partir de l'outil. A noter que ce module vient combler un manque dans la suite TFS concernant une saisie des temps précise avec traçabilité jour par jour.



Figure 61: Avanade TFS Timesheet

Coté ouverture, on note Visual Studio Team Explorer Everywhere, permettant de connecter des équipes de développement Java / Eclipse à TFS et de bénéficier de l'ensemble des fonctionnalités de la plateforme, ci-dessous, la matrice de compatibilité :

Apporter de la constance à travers les équipes

	eclipse	Visual Studio
CMME Agile and Custom Process Templates	•	•
Iteration Planning	•	•
Project reporting	•	•
Atomic check-in	•	•
Check-in Folders	•	•
Work item linking	•	•
Work item hierarchy	•	•
Synchronize in Eclipse	•	
Branch Visualization		•
Shelve / Unshelve	•	•
Next Build	Next builds	All builds
Continuous Integration & Gated Check-in	•	•

Figure 62 : Matrice de compatibilité TFS / Visual Studio / Eclipse

7. Conclusion

L'ALM constitue indubitablement un facteur clé de succès des projets informatiques et Microsoft a apporté sa pierre à l'édifice avec son offre autour de la plateforme TFS 2012.

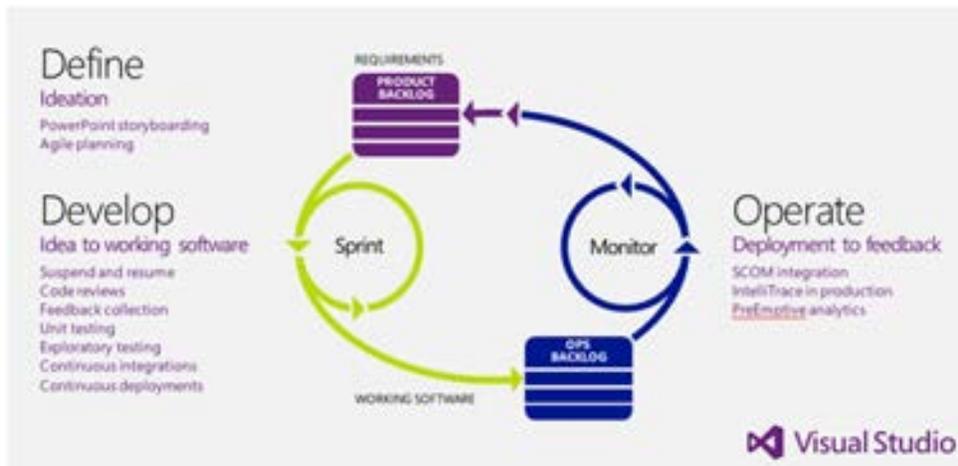


Figure 63 : Visual Studio 2012 de la définition des exigences jusqu'à la production et les opérations

L'approche de Microsoft pourrait en conclusion se résumer par :

- Combiner méthodes et outils au service des projets et des hommes
- Automatiser le processus et tout ce qui peut l'être
- Couvrir l'ensemble du cycle de vie
- S'adapter au rôle de chacun au travers d'outils dédiés.
- Fournir un workflow et des processus adaptables
- Piloter avec les métriques et les tableaux de bord de la fabrique logicielle
- Le rôle clé d'un référentiel unique

Avec Visual Studio 2012 et Team Foundation Server 2012, Microsoft propose une solution intégrée qui brise les silos en favorisant la collaboration entre les différents membres d'une équipe projet tout en assurant une qualité logicielle de bout en bout.

SHORTBREAD VISUAL STUDIO à la myrtille

Ingrédients

Pour le biscuit :

75g de farine
25g de sucre
50g de beurre

Pour le caramel :

50g de beurre
50g de vergeoise
400g de lait concentré
10cl de purée de myrtille
2 citrons verts

Couverture :

200g de couverture ivoire
colorant violet en poudre

Croquant pétillant :

100g de couverture ivoire
20g de sucre pétillant

Déco : myrtilles fraîches

Recette

LE BISCUIT :

Mélanger la farine le sucre et le beurre jusqu'à homogénéité.

Cuire à 180°C pendant 20 minutes. Laissez refroidir.

LE CAMEL :

Faire fondre le beurre avec le lait et le sucre en tournant continuellement, lorsque le sucre est dissout porter à ébullition.

Une fois à ébullition réduire la température tout en continuant de tourner et cuire environ 7 minutes ou jusqu'à ce que le caramel ait épaissi. Ajoutez la purée de myrtille et laissez reprendre la consistance désirée. Zestez les citrons verts.

Verser sur le biscuit dans le moule et laisser refroidir.

LA COUVERTURE :

Une fois refroidi, fondre le chocolat avec le colorant. Versez sur le caramel.

Taper le fond du moule sur le plan de travail pour rapidement répartir le chocolat cela assure une coque en chocolat extra lisse.

CROQUANT PETILLANT :

Faites fondre le chocolat et coulez le finement sur un silpat. Parsemez de sucre pétillant. Coupez des triangles à l'aide d'un couteau à la lame chaude. Plantez le triangle sur la couverture.

DRESSAGE :

Découpez le Shortbread Visual Studio en triangle. Piquez de petits triangles blancs de couverture pétillants et parsemez de quelques myrtilles fraîches.





Les défis Visual Studio



Relevez les Défis Visual Studio

Montrez ce dont vous êtes capables
et passez du simple commis au chef étoilé !

[lancez-vous sur visualstudio.fr/defis](http://visualstudio.fr/defis)



Installez l'extension
Défis Visual Studio
depuis la galerie Visual
Studio

**Téléchargez
l'extension VS**



Accumulez des points
dans l'environnement
de développement VS
2012

**Débloquez les haut
faits**



Des QR Codes sont
dissimulés dans tous les
événements Visual
Studio. A vous de les
découvrir !

**Participez aux
événements**

Notes



Avanade fournit des services métier, fonctionnels et technologiques qui allient vision, expertise et innovation sur les technologies Microsoft.

Les solutions d'Avanade améliorent la performance, la productivité et les ventes de sociétés opérant dans tous les secteurs d'activité. Grâce à son réseau mondial de consultants experts Microsoft, Avanade est à même de combiner de manière optimale la mobilisation de ses consultants onshore, offshore et nearshore et de délivrer un bon compromis entre coûts et risques. Avanade, dont l'actionnaire majoritaire est Accenture, a été créé en 2000 par Accenture et Microsoft Corporation. Avanade emploie 16 000 professionnels dans plus de 20 pays à travers le monde.

Pour plus d'informations, consultez : <http://www.avanade.com/fr/>

