

Université de la Manouba
Ecole Nationale des Sciences de l'Informatique



Rapport de Mémoire de Fin d'Études
Présenté en vue de l'obtention du titre
D'INGÉNIEUR EN INFORMATIQUE

Sujet

MISE EN PLACE D'UNE PLATEFORME D'INTÉGRATION CONTINUE
AUTOUR DU BUS APPLICATIF TIBCO

Réalisé par
Mohamed Manssour GHORBEL



Organisme : TALAN TUNISIE CONSULTING

Nom du responsable : M. Bahjet BOUSSOFARA

Encadré par : Mme. Yasmine Mokaddem et M. Anis Barkaoui

Supervisé par : Dr. Imtiaz Fliss

Adresse : 10 rue de l'énergie solaire, impasse N 1, Cedex 2035 Charguia 1 Tunis

Année Universitaire : 2017-2018

Signature

Mme. Yasmine Mokaddem

Dédicaces

Je dédie mon travail à mes chers parents pour leurs soutien, patience et amour qui me pousse toujours vers l'avant.

À mes chères sœurs pour leur présence auprès de moi, leurs support et amour inconditionnel.

À tous mes amis et toute ma famille qui n'ont pas cessé de m'encourager.

À tous mes professeurs qui sans eux je ne pourrai pas atteindre ce stade.

À ma très chère fiancée Donia Hammami, pour tout l'amour et le soutien qu'elle m'a offert.

Remerciements


C'est avec un grand plaisir que je réserve cette page en signe de gratitude et de profonde reconnaissance à tous ceux qui m'ont aidé de près ou de loin à la réalisation de ce travail plus particulièrement :

Mes encadrants de stage Mme. Yasmine Mokaddem et M. Anis Barkaoui, qui ont proposé et dirigé ce travail. Je leur exprime mes profondes reconnaissances pour la confiance qu'ils m'ont accordée et leurs aide précieux, leurs encouragement et leurs conseils continus durant ce stage.

Tous les membres de l'équipe pour m'avoir accepté comme membre du groupe et pour leurs précieux soutiens.

Ma superviseuse à l'École Nationale des Sciences de l'Informatique, Mme Imtiaz Fliss pour sa disponibilité, son soutien et les conseils précieux qu'elle m'a prodigué autant sur le plan professionnel que sur le plan personnel tout au long du stage.

Table des matières

Introduction générale	1
1 Présentation générale	3
1.1 Cadre général 	3
1.1.1 Présentation de l'organisme d'accueil	3
1.1.2 Présentation de l'équipe	4
1.2 Présentation du projet	6
1.2.1 Motivations et problématique	6
1.2.2 Travail demandé	7
1.3 Cycle de développement	7
Conclusion	8
2 Étude préalable	9
2.1 Concepts clés	9
2.1.1 ESB	9
2.1.2 TIBCO Software	10
2.2 Étude de l'existant	12
2.2.1 Solution utilisée par l'entreprise	12
2.2.2 Solution existante sur le marché	13
2.3 Solution proposée	14
Conclusion	15
3 Sprint 0 : Avant démarrage	16
3.1 Capture des besoins	16
3.1.1 Définition des acteurs	16
3.1.2 Backlog du produit	17

3.1.3	Spécification des besoins	17
3.2	Conception globale	18
3.2.1	Diagramme de paquets	18
3.2.2	Diagramme d'activités	19
3.3	Environnement de travail	20
3.3.1	Environnement matériel	20
3.3.2	Environnement logiciel	21
3.4	Planification des sprints	22
3.5	Chronogramme	22
	Conclusion	23
4	Sprint 1 : Génération des projets TIBCO	24
4.1	Sprint Backlog	24
4.2	Spécification des besoins	25
4.2.1	Diagramme de cas d'utilisation relatif à l'utilisateur	25
4.2.2	Description de quelques scénarii	25
4.3	Conception	27
4.3.1	Diagramme de paquets	27
4.3.2	Diagramme d'activités de la création des projets TIBCO	28
4.4	Réalisation du sprint 1	29
	Conclusion	31
5	Sprint 2 : Invocation JMS	32
5.1	Sprint Backlog	32
5.2	Spécification des besoins	33
5.2.1	Diagramme de cas d'utilisation relatif à l'utilisateur	33
5.2.2	Diagramme de séquence du cas d'utilisation : Création d'opération d'in- vocation JMS	34
5.3	Conception	35
5.3.1	Diagramme de paquets	35
5.3.2	Diagramme d'activités	35
5.4	Réalisation du sprint 2	37
	Conclusion	38

6	Sprint 3 : Exposition SOAP	39
6.1	Sprint Backlog	39
6.2	Spécification des besoins	40
6.2.1	Diagramme de cas d'utilisation relatif à l'utilisateur	40
6.2.2	Diagramme de séquence du cas d'utilisation : Création d'opération d'ex- position SOAP	41
6.3	Conception	41
6.3.1	Diagramme de paquets	41
6.3.2	Diagramme d'activités	42
6.4	Réalisation du sprint 3	44
	Conclusion	45
7	Sprint 4 : Intégration Continue	46
7.1	Sprint Backlog	46
7.2	Spécification des besoins	47
7.3	Conception	48
7.3.1	Diagramme de déploiement	48
7.3.2	Diagramme d'activités	48
7.4	Réalisation du Job Jenkins	49
	Conclusion	53
	Conclusion générale	54
	Bibliographie	55
	Netographie	56

Table des figures

1.1	« Organisation de l'équipe Talan TIBCO Factory »	5
2.1	« Composants du TIBCO Business works »	10
2.2	« Exemple de composition d'un projet TIBCO »	12
2.3	« Cycle de développement des projets TIBCO »	13
2.4	« Windev » [5]	14
3.1	« Diagramme des cas d'utilisation global »	18
3.2	« Diagramme de paquets »	19
3.3	« Diagramme d'activités global »	20
3.4	« Diagramme de gantt réel »	23
4.1	« Diagramme de cas d'utilisation de la génération des projets »	25
4.2	« Diagramme de séquence système de création d'un projet TIBCO »	26
4.3	« Diagramme de séquence système de création d'une opération »	27
4.4	« Diagramme de paquets relatif à la génération des projets »	27
4.5	« Diagramme d'activités de la création d'un nouveau projet »	28
4.6	« Création d'un nouveau projet TIBCO »	29
4.7	« Liste des projets TIBCO »	30
4.8	« Liste des actions relatives à chaque projet »	30
4.9	« Création des opérations »	31
5.1	« Diagramme des cas d'utilisation de l'invocation JMS »	33
5.2	« Diagramme de séquence de l'invocation JMS »	34
5.3	« Diagramme de paquets invocation JMS »	35
5.4	« Diagramme d'activités du processus d'invocation JMS »	36

5.5	« Liste des projets à invoquer »	37
5.6	« Invocation JMS »	38
6.1	« Diagramme des cas d'utilisation de l'exposition SOAP »	40
6.2	« Diagramme de séquence de l'exposition SOAP »	41
6.3	« Diagramme de paquet de l'exposition SOAP »	42
6.4	« Diagramme d'activités détaillé de l'exposition SOAP »	43
6.5	« Liste des actions »	44
6.6	« Exposition SOAP »	44
7.1	« Diagramme des cas d'utilisation relatif à l'intégrateur »	47
7.2	« Diagramme de déploiement relatif à l'intégration continue »	48
7.3	« Diagramme d'activité relatif à l'intégration continue »	49
7.4	« Dashbord Jenkins »	50
7.5	« Job d'intégration continue »	50
7.6	« Sortie de la phase de Checkout »	51
7.7	« Sortie de la phase de la construction »	51
7.8	« Sortie de la phase de déploiement »	51
7.9	« Sortie de la phase d'archivage et de notification »	52
7.10	« Dashbord TIBCO Administrator »	52
7.11	« Notification Email »	52

Liste des tableaux

3.1	Backlog du produit	17
3.2	Partitionnement des user stories par sprint	22
4.1	Backlog Sprint 1	24
5.1	Backlog Sprint 2	32
6.1	Backlog Sprint 3	39
7.1	Backlog Sprint 4	46

Introduction générale

L'un des défis auxquels se confronte une entreprise moderne est la nécessité de maintenir une vue d'ensemble sur les nombreux changements dans son système d'information, quelles que soient leurs natures. Ceci est l'un des piliers garantissant sa compétitivité et sa pérennité.

En effet, les entreprises du secteur de développement logiciel concentrent toujours leurs recherches sur les astuces et les bonnes idées qui rendent leurs solutions plus performantes et plus fiables tout en répondant aux attentes et aux besoins du client qui ne cessent d'augmenter.

La prise de conscience des risques liés à l'absence d'un système qui automatise le processus de développement des projets TIBCO ainsi qu'à l'absence d'un système qui assure la non-régression du code grandit de plus en plus.

A cet égard, l'équipe TIBCO factory de Talan fait face à certains problèmes qui peuvent conduire à l'échec le développement des nouveaux projets ainsi que la maintenance de ceux qui existent déjà tels que : l'augmentation des délais de la mise en production, la redondance des tâches et du code, la difficulté de maintenance ainsi que des problèmes liés à un manque de processus garantissant l'intégration continue des projets.

C'est dans ce cadre, que s'inscrit notre projet de fin d'études du cycle des ingénieurs à l'École Nationale des Sciences de l'Informatique, réalisé à Talan Consulting. Notre projet consiste à concevoir et à développer un framework qui automatise le processus de développement des projets TIBCO. Ce framework présente une nécessité majeure qui vise à avoir un système performant résolvant les problèmes actuels de l'équipe TIBCO Factory.

Le présent rapport décrit les différentes étapes de notre travail et il s'articule autour de sept chapitres :

Le premier chapitre contient une présentation de l'organisme d'accueil ainsi que du cadre général de ce projet. Enfin, il expose le choix de la méthodologie de gestion de projet suivi tout au long la réalisation de notre travail.

Le deuxième chapitre expose dans un premier temps les concepts de base. En second temps, il présente une étude de l'existant suivie de la solution proposée.

Le troisième chapitre présente, en premier lieu, les différents acteurs de notre solution suivis du backlog du produit. En second lieu, il spécifie les besoins globaux ainsi que la conception générale de notre framework. Nous finissons par présenter nos choix technologiques.

Le quatrième, cinquième, sixième et septième chapitres décrivent les cycles de développement des différents sprints. Pour chaque sprint nous détaillons la spécification des besoins, la conception ainsi que la réalisation.

Nous récapitulons par une conclusion générale qui résume le travail que nous avons réalisé et dégage quelques perspectives.

Chapitre 1

Présentation générale

Avant d'entamer les détails de la réalisation de l'application, une présentation générale du cadre de notre projet est nécessaire. Ainsi, dans ce chapitre, nous nous intéressons à mettre notre projet dans son contexte en présentant l'organisme d'accueil Talan Tunisie Consulting. Ensuite, nous enchaînons par une explication des motivations qui nous ont amené à envisager ce projet. Nous finissons par donner un aperçu sur le cycle de développement suivi pour la réalisation de notre projet.

1.1 Cadre générale



Le sujet introduit par le présent rapport est intitulé Mise en place d'une plateforme d'intégration continue autour du bus applicatif Tibco. Il a été élaboré dans le cadre du projet de fin d'études en vue d'obtenir le diplôme National d'Ingénieur en Informatique à l'École Nationale des Sciences de l'Informatique pour l'année universitaire 2017/2018, réalisé au sein de l'entreprise Talan Tunisie Consulting.

1.1.1 Présentation de l'organisme d'accueil

Afin de situer notre travail dans son environnement de réalisation, nous présentons dans cette section l'organisme d'accueil en exposant ses secteurs d'activités.

1.1.1.1 Talan Tunisie Consulting

Talan Consulting [1] est une société de conseil spécialisée dans l'intégration des NTIC (Nouvelles Technologies de l'Information et de la Communication) dans le domaine de la relation

client. Son expertise sectorielle se concentre sur les opérateurs de services à savoir les services Finance et Assurance, Télécoms et Médias, Énergie et Services Publics et le service Transport et Logistique.

Talan est fondée en 2002 par Mehdi Houas, Eric Benamou et Philippe Cassoulat. Elle délivre ses savoir-faire métiers, fonctionnels et technologiques à l'échelle internationale avec plus de 2000 collaborateurs, en France, à New York, à Hong Kong, au Royaume-Uni, à Montréal. Pour accélérer son développement, Talan ouvre un centre de développement « nearshore » en Tunisie intitulé « Talan Tunisie Consulting », mobilisant à ce jour environ 180 ingénieurs de développement nouvelles technologies et travaillant avec de grands clients européens.

1.1.1.2 Secteurs d'activités

Les secteurs d'activités de Talan Consulting couvre essentiellement :

- Le secteur de finance à travers une collaboration avec des banques d'investissement et des assurances.
- Le secteur des télécoms à travers des projets destinés aux opérateurs télécoms et un ensemble de fournisseurs d'accès à internet.
- Le secteur de transport et logistique.
- Le secteur d'énergie à travers des projets de développement visant des opérateurs de service d'électricité, gaz, eau, etc.

Ces différents secteurs se caractérisent par la mise en disposition d'un ensemble de prestations :

- Conseil et assistance à la maîtrise d'ouvrage ;
- Refonte et optimisation des processus métiers ;
- Support aux grands projets de transformation ;
- Alignement des systèmes d'information aux changements d'organisation et à l'accompagnement au changement.

1.1.2 Présentation de l'équipe

Le projet est réalisé au sein de l'équipe Talan Tibco Factory. En effet, l'organisation de Talan, pour la réalisation des projets Tibco, se base sur le concept de Factory : un centre de compétence et de production dédié à la technologie Tibco. La Factory Tibco réalise l'ensemble du cycle de

développement, en mode agile, des différents besoins clients. Cette organisation est présentée dans la figure 1.1

- Un directeur de Factory : Responsable de l'équipe et de la gestion de la Factory.
- Un Manager de livraison : Responsable de l'architecture, des livraisons et de la garantie de la qualité des livraisons clients.
- Deux Stream Leaders : Chacun responsable d'un scope fonctionnel différent, ils réalisent la conception et assurent le bon fonctionnement et déroulement du projet au sein de leurs équipes.
- Trois Designers : Responsables du design des différents vues pour les projets Tibco.
- Trois développeurs : Responsables du développement.
- Un responsable Recette : Responsable de la qualité fonctionnelle, il effectue les tests de non régression sur les projets TIBCO.



FIGURE 1.1 – « Organisation de l'équipe Talan TIBCO Factory »

1.2 Présentation du projet

Dans cette partie, nous exposerons la problématique ainsi que le travail demandé.

1.2.1 Motivations et problématique

Comme présenté auparavant, notre projet de fin d'études est réalisé au sein de l'équipe TIBCO Factory. Cette équipe travaille sur un grand nombre de projets d'intégration et d'orchestration des services utilisant TIBCO Business Works (appelés projets TIBCO). Pour traiter ces projets, les membres de l'équipe se trouvent appelés à travailler manuellement et répéter un bon nombre d'étapes. La première étape consiste à créer un projet à partir d'un ancien template utilisée pour tous les projets, créer les ressources du projet (HTTP, JMS, JDBC, FTP) ainsi que les opérations qui lui sont nécessaires. L'étape suivante consiste à faire des invocations JMS (Java Message Service) et à exposer les opérations déjà créées. Enfin la dernière étape comporte l'exécution des tests métiers ainsi que la construction (build) et le déploiement. Cela a engendré de graves conséquences, entre autres, la non-conformité aux termes et aux conditions prédéfinies ainsi que le non-respect des délais de livraison.

En effet, la modification d'un projet existant ou l'ajout d'une nouvelle fonctionnalité concernant toutes les parties techniques du projet, cela engendre une validation du code et un redéploiement de la totalité de l'application. Vu ce grand nombre de modifications, la solution actuelle souffre de manque de flexibilité qui est exigée par toutes ces conditions. D'un autre côté, de nouveaux développeurs rejoignent l'équipe et d'autres plus expérimentés quittent l'entreprise. Néanmoins, cela engendre un effort supplémentaire de montée en compétence sur les projets TIBCO.

De plus, la façon traditionnelle d'organiser le travail ainsi que les nouvelles fonctionnalités augmentent considérablement le temps consacré pour la mise en production. Éventuellement, cela cause l'augmentation du temps d'indisponibilité pour l'application, une perte du temps ainsi qu'une perte d'argent.

Afin de remédier à ce problème, l'équipe tibco voulait se doter d'un moyen performant permettant l'automatisation maximale du processus de développement des projets TIBCO. D'où le besoin de concevoir et développer un framework interne capable de répondre aux besoins de l'équipe TIBCO Factory.

1.2.2 Travail demandé

En vue de ces exigences, notre mission consiste à concevoir et développer une solution interne pour Talan Consulting ayant comme but d'automatiser le processus de travail chez l'équipe TIBCO. Cette solution remplacera les étapes manuelles répétitives que les membres de l'équipe TIBCO avaient l'habitude de faire pour chaque projet.

1.3 Cycle de développement

Afin de réaliser notre projet de fin d'études, un bon nombre de modèles de cycle de développement sont possibles. Les plus adaptés pour notre cas de figure sont les méthodes itératives. Parmi ces méthodes, nous pouvons distinguer les méthodes AGILE [B1]. Une méthode AGILE assure une meilleure communication avec le client et une meilleure visibilité du produit livrable. Il existe plusieurs méthodes AGILE à savoir : Scrum, Extreme Programming (XP), Agile Unified Process (AUP), Feature Driven Development (FDD), ect..

Pour la réalisation du projet, nous devons choisir la méthode la plus adéquate au projet et non pas la meilleure méthode d'entre eux. En effet, nous avons opté pour la méthode Scrum [B2].

Scrum consiste à faire travailler l'équipe de développement d'une manière collective, sou-
dée vers un objectif commun. Scrum sert à développer des produits généralement en quelques mois. Les fonctionnalités souhaitées sont collectées dans ce que nous appelons le backlog du produit où elles sont classées par priorité. L'équipe travaille en mode itératif (une itération de deux à quatre semaines), appelé aussi sprint, pour produire à la fin de chaque itération une version potentiellement livrable. Le Scrum Master contrôle le déroulement des sprints et intervient pour appliquer des ajustements pour assurer le succès du sprint. La version produite à la fin de chaque sprint est un incrément du produit et son évaluation permet de modifier le backlog pour le sprint suivant.

- Le backlog complet du produit est défini par le client. Il rassemble les fonctionnalités que doivent être présent dans produit. Dans notre cas de figure, le client est Talan Consulting puisque le projet est interne.
- Le backlog du sprint et la planification des tâches de sprint se font par l'équipe.

- La réunion quotidienne se fait entre les membres d'équipe et ne dépasse pas les 15 minutes. Cette réunion n'est pas destinée à la résolution des problèmes, elle donne aux membres de l'équipe une vision sur la progression de travail. Chaque membre est amené à répondre à trois questions : "Ce qu'il a fait hier ? , ce qu'il fera aujourd'hui ?, Quels sont les obstacles rencontrés ?".

Conclusion

Au niveau de ce chapitre, nous avons présenté le cadre de notre travail, tout en spécifiant le contexte du sujet, la problématique ainsi que le travail demandé. Nous avons clôturé par donner un bref aperçu sur le cycle de développement suivi pour la réalisation de notre travail.

Chapitre 2

Étude préalable

Dans le but d'une bonne compréhension de notre projet, une étude préalable est essentielle et indispensable. Ainsi, nous consacrons le présent chapitre pour aborder de plus près le contexte du projet en détaillant, dans un premier temps, les concepts clés et l'étude de l'existant, dans un second temps.

2.1 Concepts clés

Dans cette section, nous exposons les concepts de base liés à notre projet.

2.1.1 ESB

Un « Enterprise Service Bus » [2] est une solution d'intégration implémentant une architecture totalement distribuée. C'est une infrastructure qui optimise les échanges entre consommateurs et fournisseurs de services. Elle permet la mise en place de l'architecture orientée service (SOA) [3]. L'ESB donne un API qui peut être utilisé pour :

- Le développement des services
- La transformation des données.
- La transformation du format des messages.
- Rendre les services interactifs les uns avec les autres de manière fiable.
- Conversion des protocoles.
- L'orchestration des services.
- Intégration des services.

2.1.2 TIBCO Software

Tibco Software [4] est une société américaine qui fournit des logiciels d'infrastructure pour les entreprises dans un contexte de cloud computing. Elle propose de gérer les informations, les prises de décisions, les process et les applications en temps réel pour plus de 4000 clients.

Tibco a été fondée en 1997 par Vivek Ranadivé, le siège est situé en Palo Alto en Californie. Ses principaux concurrents sont IBM et Oracle.

2.1.2.1 TIBCO BusinessWorks

TIBCO BusinessWorks est l'une des solutions de TIBCO SOFTWARE destinée pour l'orchestration et l'intégration des services. Elle est basée sur une architecture orientée service. Elle permet d'héberger plusieurs applications simultanément et d'exécuter les processus métier en mode test et au moment de l'exécution.

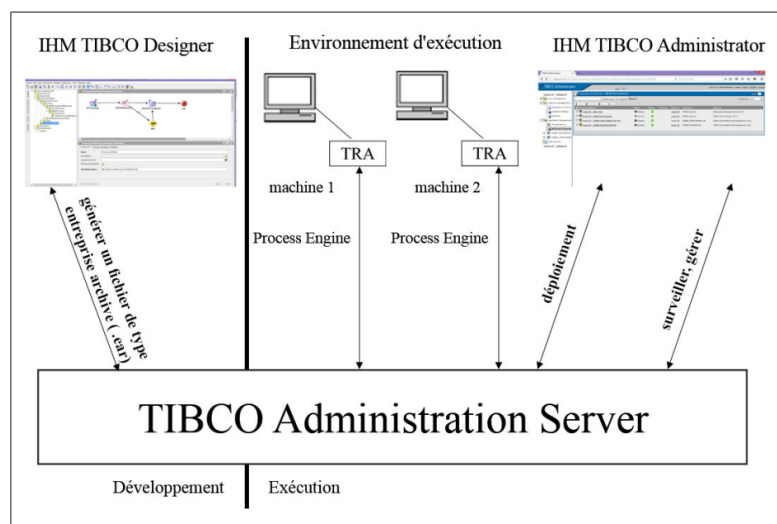


FIGURE 2.1 – « Composants du TIBCO Business works »

La figure 2.1 ci-dessus représente les différents composants du TIBCO Business works, nous citons :

- **TIBCO Designer :** est une interface graphique facile à utiliser pour créer des projets d'intégration. TIBCO Designer vous permet de faire glisser et déposer des composants dans un projet, puis de spécifier les informations de configuration pour chaque composant.
- **TIBCO Runtime Agent (TRA) :** est un ensemble de logiciels TIBCO et de logiciels tiers. Il est un pré-requis pour les autres produits logiciels TIBCO. Il fournit les produits et

les composants suivants : TIBCO Designer, TIBCO Domain Utility, TIBCO Hawk. Il est responsable du démarrage et de l'arrêt des processus exécutés. Il fournit l'environnement d'exécution, c'est-à-dire toutes les bibliothèques partagées, y compris les bibliothèques tierces.

- **TIBCO Administrator** : Inclut tout ce qui concerne le déploiement, la sécurité de l'administration, le monitoring et la gestion des processus et des machines. Il représente le serveur d'administration TIBCO ainsi que l'interface web que fournit TIBCO aux utilisateurs.

2.1.2.2 Apports de TIBCO Business Works

TIBCO Business Works offre à ses utilisateurs la possibilité de :

- réduire le temps et l'effort fournis pour développer et déployer des activités métier.
- accélérer le cycle de développement et de déploiement des applications.
- améliorer la consistance et la performance.
- réutiliser les fonctions et les données sous forme de services dans les différents processus métiers à grande complexité.

2.1.2.3 Structure du projet TIBCO

Un projet développé sous l'environnement TIBCO suit une structure spécifique composée de plusieurs dossiers comme l'indique la figure 2.2

Parmi les différents dossiers qui forment le projet, il y a le dossier "Process Definition" qui, comme son nom l'indique, contient les définitions des processus contenus au sein du projet, couramment appelés opérations.

Le dossier "Resources" contient plusieurs formes de ressources partagées au sein du même projet :

- **Connexions** : Pour assurer une interactivité avec les composants externes au projet, TIBCO permet d'établir plusieurs types de connexions : HTTP, JDBC, JMS, FTP, etc.
- **Définition des schémas** : Ce dossier contient les fichiers XSD qui sont nécessaires lors de l'appel du service web. Ces fichiers permettent de définir la structure de l'opération en indiquant les différents paramètres qui la composent.

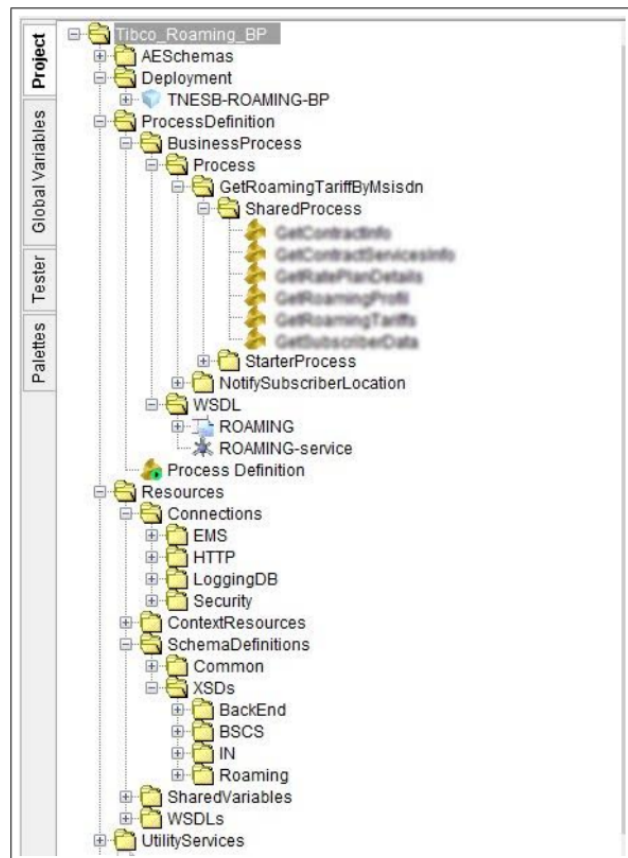


FIGURE 2.2 – « Exemple de composition d'un projet TIBCO »

2.2 Étude de l'existant

Après avoir défini tous les concepts qui vont nous faciliter la compréhension du contexte, nous passons maintenant à l'étude de la solution actuelle et la solution existante sur le marché afin de dégager les anomalies et proposer notre nouvelle solution.

2.2.1 Solution utilisée par l'entreprise

La figure 2.3 ci-dessous illustre le cycle de développement actuel des projets TIBCO. En effet, l'équipe Talan Tibco Factory utilise une template d'un autre projet existant pour la création du nouveau projet. Afin de faire adapter l'ancien template au nouveau projet et satisfaire les besoins du client, l'équipe doit apporter des changements sur le projet.

Ces changements se modélisent par la suppression des ressources (HTTP, FTP, JDBC, JMS) et des schémas xsd. Les changements peuvent être aussi des modifications des variables globales et des configurations des processus et peuvent être notamment un ajout de nouvelles

opérations, des ressources, des schémas xsd et des services SOAP.

Néanmoins, la solution actuelle dégage plusieurs anomalies à savoir la répétition des tâches d'une façon manuelle qui engendre une perte de temps, de ressources et d'argent. De plus, l'équipe Talan TIBCO factory n'adopte pas les pratiques de l'intégration continue, d'où le manque d'agilité et de l'efficacité opérationnelle dans l'entreprise.

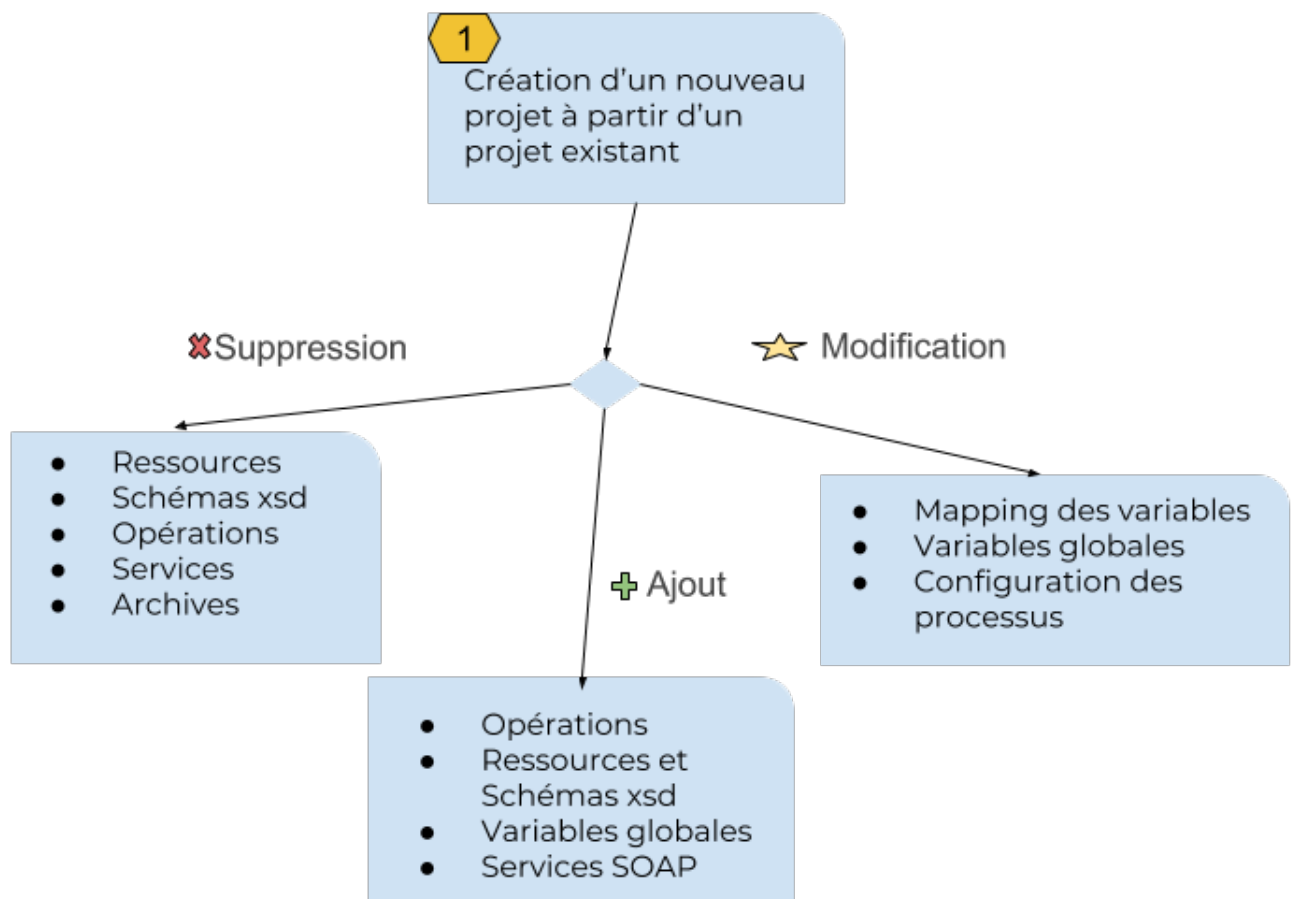


FIGURE 2.3 – « Cycle de développement des projets TIBCO »

2.2.2 Solution existante sur le marché

Avant de proposer notre solution, il est crucial d'étudier les solutions les plus utilisées sur le marché. La seule solution qui permet l'automatisation d'un nombre de projets est WinDev. [5].

WINDEV est un logiciel développé par PC SOFT, il permet par son intégration totale, sa

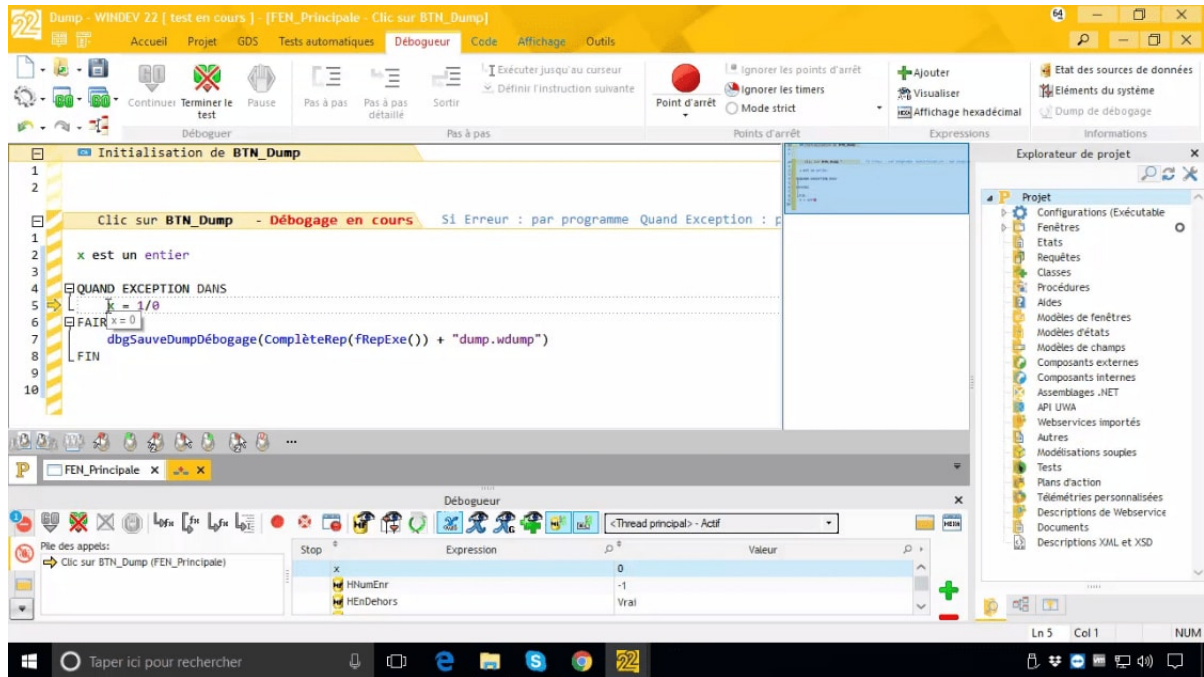


FIGURE 2.4 – « Windev » [5]

facilité légendaire, sa technologie de pointe, de réussir plus vite les projets d'envergure sous Windows, Linux, .NET, Java et plus.

WINDEV crée des applications destinées à fonctionner sous différents systèmes d'exploitation. WEBDEV les recompile pour fonctionner sous Internet, Intranet. WINDEV Mobile les recompile pour fonctionner sur Tablette ou smartphone.

Toutes ces applications peuvent fonctionner dans le cloud. Les développeurs utilisent le même projet, le même code, les mêmes objets et les mêmes éléments sur toutes les plateformes.

Parmi les caractéristiques de WINDEV nous pouvons citer sa rapidité, sa puissance et sa simplicité. De plus, l'architecture de l'éditeur supporte toutes les bases de données et s'interface avec plusieurs langages et technologies.

Cependant, cette solution est payante et exige beaucoup d'espace mémoire pour les bibliothèques dynamiques (Dynamic Link Library). De plus, WINDEV n'est pas applicable sur des projets TIBCO.

2.3 Solution proposée

Après une étude approfondie de la solution existante et dans le but de surmonter les limites déjà citées, il est crucial d'automatiser le processus de travail actuel. Le but de cette automa-

tisation est d'améliorer la performance, gagner en temps de production, gagner de l'argent et avoir une meilleure satisfaction client.

Dans ce contexte, notre solution consiste à concevoir et développer un framework interne permettant d'une part, d'automatiser le processus de développement des projets TIBCO. D'autre part, il assure la non-régression du code via l'intégration continue.

Conclusion

Dans ce chapitre, nous avons commencé, tout d'abord, par mettre l'accent sur les concepts clés afin de mieux comprendre notre projet. Nous avons exposé par la suite une étude de l'existant suivie de la solution proposée. Le prochain chapitre sera consacré au sprint de démarrage.

Chapitre 3

Sprint 0 : Avant démarrage

Le présent chapitre représente notre sprint de démarrage. D'abord, nous commençons par présenter les acteurs principaux de notre solution, suivi du backlog du produit. Ensuite, nous détaillons les besoins globaux ainsi que la conception globale de notre solution. Enfin, nous fixons nos choix technologiques qui vont servir pour définir notre framework TTFF et nous présentons le chronogramme des tâches.

3.1 Capture des besoins

Cette section est consacrée à l'analyse et la spécification des besoins. Dans un premier lieu, nous identifions les acteurs ainsi que les besoins de notre application. En second lieu, nous détaillons la spécification des besoins globaux via un diagramme de cas d'utilisation.

3.1.1 Définition des acteurs

Dans cette partie, nous allons identifier nos acteurs à savoir : L'utilisateur de TTFF (Talan Tibco Factory Framework) et l'intégrateur.

- **Utilisateur** : est un acteur principal, il interagit directement avec notre application pour gérer les projets TIBCO.
- **Inégrateur** : est un acteur principal, il est responsable de l'intégration continue de chaque projet TIBCO.

3.1.2 Backlog du produit

Dans cette section nous présentons notre backlog de produit. En effet, le tableau 3.1 présenté ci-dessous illustre nos besoins (User Stories).

TABLE 3.1 – Backlog du produit

ID	User stories	Priorité
1	En tant qu'utilisateur, je veux pouvoir créer un nouveau projet TIBCO.	Elevée
2	En tant qu'utilisateur, je veux pouvoir créer des opérations (Process).	Elevée
3	En tant qu'utilisateur, je veux pouvoir créer des opérations d'invocation JMS.	Moyenne
4	En tant qu'utilisateur, je veux pouvoir créer des opérations d'exposition SOAP.	Moyenne
5	En tant qu'utilisateur, je veux pouvoir faire un checkout du svn vers " Espace de travail partagé du framework ".	Elevée
6	En tant qu'intégrateur, je veux pouvoir lancer un job Jenkins pour l'intégration continue qui doit couvrir : Checkout, Build, Validation, Analyse de code, Déploiement.	Faible

3.1.3 Spécification des besoins

Dans cette section, nous allons modéliser toutes les fonctionnalités offertes par notre application en se basant sur le diagramme de cas d'utilisation présenté par la figure 3.1 ci-dessous. En effet, l'utilisateur peut, à partir de notre application, créer de nouveaux projets TIBCO, créer des opérations élémentaires et faire un checkout pour les projets existants. Il peut également créer des opérations d'invocation JMS ainsi que des opérations d'exposition SOAP. L'intégrateur peut configurer un job jenkins et peut aussi le lancer au cas du besoin.

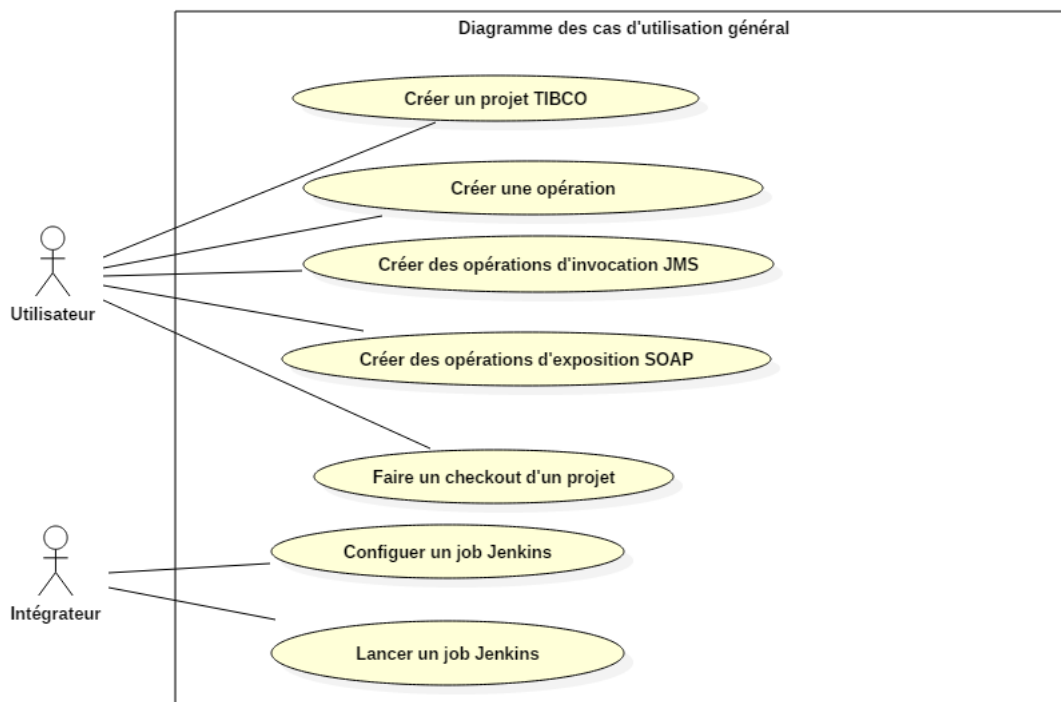


FIGURE 3.1 – « Diagramme des cas d'utilisation global »

3.2 Conception globale

Dans cette partie nous allons détailler la phase de conception de notre solution en s'appuyant sur le diagramme de paquets ainsi que le diagramme d'activités

3.2.1 Diagramme de paquets

La figure 3.2 ci-dessous représente le diagramme de paquets global de notre solution.

- **Generate project** : C'est le service responsable de la création de nouveaux projets TIBCO et de la création des opérations.
- **Exposition SOAP** : C'est le service responsable de la création d'un service SOAP à partir des opérations.
- **Invocation JMS** : C'est le service responsable de la création des processus d'invocation JMS.
- **CheckOut** : C'est le service responsable d'assurer le checkout depuis le SVN.

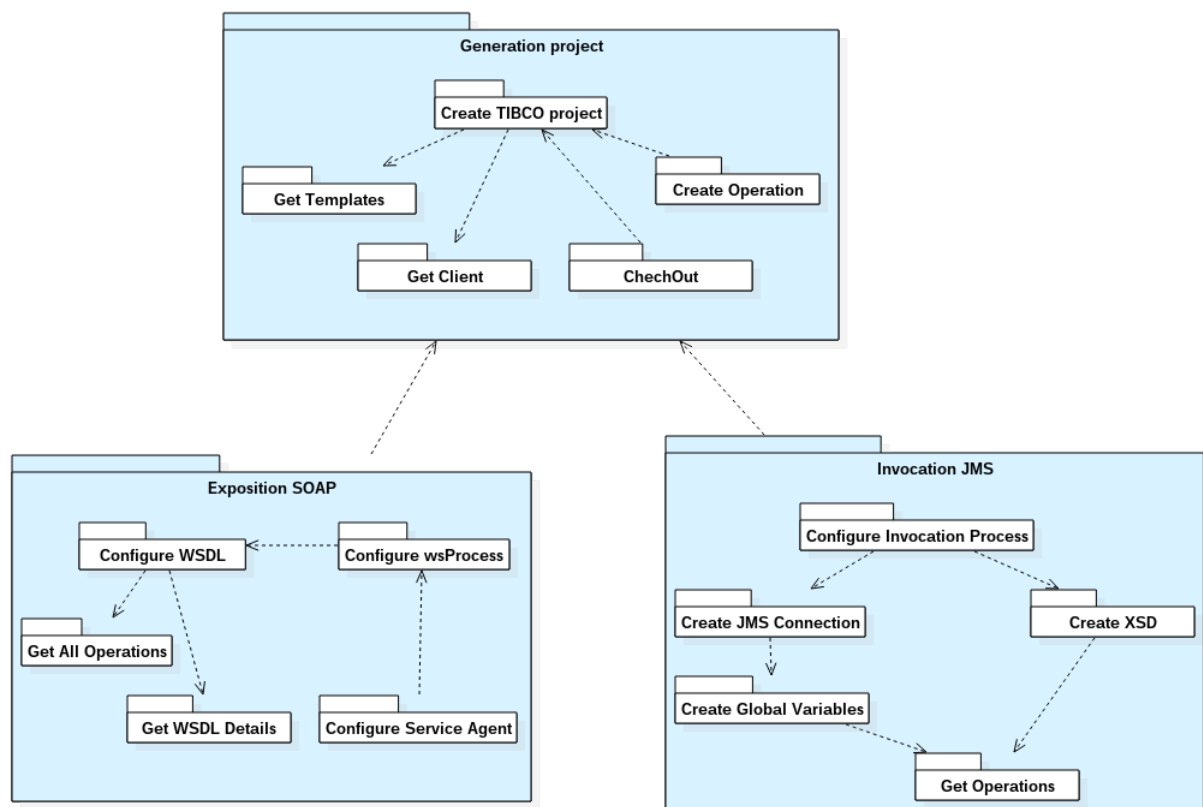


FIGURE 3.2 – « Diagramme de paquets »

3.2.2 Diagramme d'activités

La figure 3.3 ci-dessous représente le diagramme d'activités global de notre solution. En effet, La première étape de notre projet consiste à créer un nouveau projet TIBCO ainsi que les opérations qui lui sont nécessaires. Une fois cette étape est achevée, nous pouvons soit créer un processus qui invoque une opération JMS distante, soit exposer nos opérations dans un service SOAP. Après avoir créer le projet et commencer à développer les fonctionnalités nécessaires, l'intégration continue du projet s'impose afin d'assurer la non-régression du code.

Pour ce faire, le cycle d'intégration continue doit être exécuté après chaque commit des développeurs, il doit couvrir les étapes suivantes : Checkout, Construction et Validation, Création et archivage d'artifacts et Déploiement.

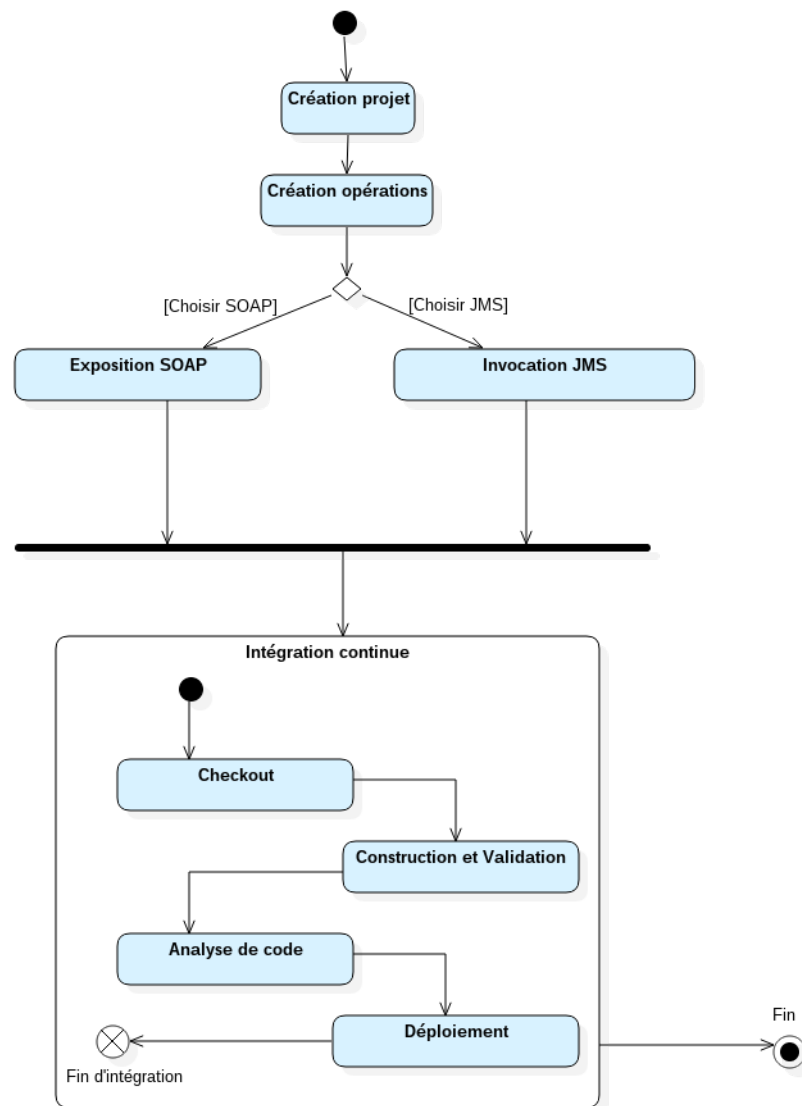


FIGURE 3.3 – « Diagramme d'activités global »

3.3 Environnement de travail

Dans cette section nous présentons l'environnement matériel et logiciel utilisés pour la réalisation de notre projet.

3.3.1 Environnement matériel

Pour mener à bien la réalisation, nous avons utilisé comme environnement matériel, un poste de travail ayant les caractéristiques suivantes :

- Système d'exploitation : Windows 10

- Disque dur : 1 To
- Ram : 8 Go
- Processeur : Intel(R) Core(TM)i5-6500U CPU 3.20GHz

3.3.2 Environnement logiciel

Dans cette partie, nous nous intéressons aux langages, aux bibliothèques et aux techniques de programmation utilisées tout au long de la réalisation de notre application.

3.3.2.1 TIBCO Business Works

TIBCO Business Works [4] est une des solutions de TIBCO SOFTWARE destinée pour l'orchestration et l'intégration des services.

3.3.2.2 Spring Boot

Spring Boot [6] est un framework qui permet de créer facilement des applications Spring basées sur la production. Avec Spring Boot, nous pouvons créer des applications Spring autonomes. Il offre des conteneurs Tomcat intégrés, des fonctionnalités prêtes pour la production et des dépendances « starter » orientées pour simplifier la configuration de notre build.

3.3.2.3 Angular 5

Angular [7] est une plateforme qui facilite la création d'applications sur le Web. Angular combine des modèles déclaratifs, des injections de dépendances, des outils de bout en bout et des meilleures pratiques intégrées pour résoudre les problèmes de développement. Angular permet aux développeurs de construire des applications qui vivent sur le web, le mobile ou le bureau.

3.3.2.4 Base de données : Oracle

La base de données Oracle (Oracle DB) est un système de gestion de base des données relationnelles entièrement évolutives et est souvent utilisée par les entreprises mondiales. Il est l'un des moteurs de base des données relationnelles les plus fiables et les plus utilisés.

3.3.2.5 Jenkins

Jenkins [8] est un serveur d'intégration continue et d'automatisation, Open Source et autonome qui peut être utilisé pour automatiser toutes sortes de tâches liées au checkout, construction, au test et à la livraison ou au déploiement de logiciels.

3.3.2.6 Apache Subversion (SVN)

Subversion [9] est un système de contrôle de version open source. Fondé en 2000 par CollabNet. Il s'appuie sur le principe de dépôt centralisé.

3.4 Planification des sprints

Le tableau 3.2 représente la planification de nos sprints en s'appuyant sur le backlog de produit cité précédemment.

TABLE 3.2 – Partitionnement des user stories par sprint

	Sprint 1	Sprint 2	Sprint 3	Sprint 4
User Story ID	1,2,5	3	4	6

Nous commençons par entamer le premier sprint intitulé " Génération des projets TIBCO " qui englobe les fonctionnalités de la priorité élevée : Création des projets TIBCO, Création des opérations élémentaires et assurer le checkout des projets depuis le SVN. Par la suite, nous enchaînons avec le deuxième et le troisième sprint intitulés respectivement " Invocation JMS " et " Exposition SOAP " qui représentent les fonctionnalités de la priorité moyenne : Création des opérations d'invocation JMS et d'exposition SOAP. Nous finissons par présenter le quatrième sprint qui consiste à développer les fonctionnalités de la priorité faible : Intégration continue des projets TIBCO.

3.5 Chronogramme

Ce travail a été réalisé durant une période de six semaines. La répartition des tâches durant notre stage est illustrée par le diagramme de Gantt réel de la figure 3.4 ci-dessous :

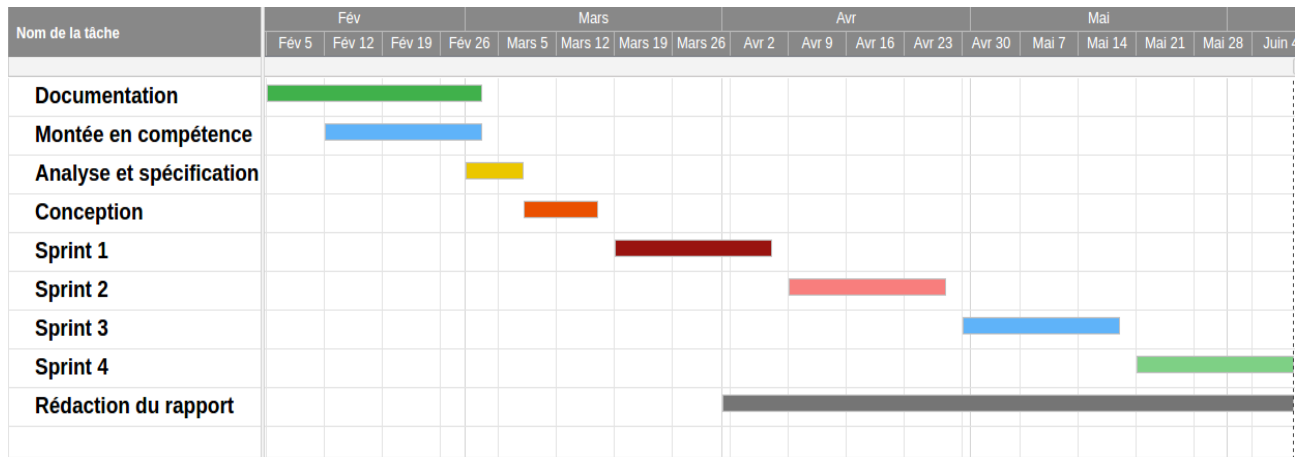


FIGURE 3.4 – « Diagramme de gantt réel »

Conclusion

Dans ce chapitre, nous avons abordé, tout d'abord, la spécification des besoins en présentant nos acteurs ainsi que notre backlog du produit. Après, nous avons présenté la conception globale de notre solution suivie de nos choix technologiques et du chronogramme des tâches. Dans le chapitre suivant, nous abordons notre premier sprint.

Chapitre 4

Sprint 1 : Génération des projets TIBCO

Dans ce chapitre qui représente le premier sprint, nous allons détailler les différentes étapes ayant pour objectif la génération des projets TIBCO. Ce sprint permet de créer de nouveaux projets TIBCO, créer aussi des opérations (Opérations élémentaires) et faire un checkout des projets depuis le SVN.

Nous commençons, ce chapitre, par présenter le sprint backlog. Nous présentons par la suite la partie spécification des besoins et la conception. Nous finissons par présenter l'interface Homme-Machine dans la partie réalisation.

4.1 Sprint Backlog

Le tableau 4.1 présente notre sprint backlog.

TABLE 4.1 – Backlog Sprint 1

Exigence	Sous tâche
1.1	Créer un nouveau projet TIBCO (BP ou SP).
1.2	Créer des opérations (Opérations élémentaires).
1.3	Faire un checkout du projet depuis SVN.

4.2 Spécification des besoins

Afin de spécifier nos besoins, nous allons utiliser le diagramme de cas d'utilisation et le diagramme de séquences.

4.2.1 Diagramme de cas d'utilisation relatif à l'utilisateur

La figure 4.1 représente le diagramme de cas d'utilisation principal. Ce diagramme représente les différentes fonctionnalités de ce sprint que l'utilisateur peut les effectuer. En effet, l'utilisateur peut créer un nouveau projet TIBCO. Il peut créer des opérations (Opérations élémentaires). Il peut aussi effectuer un checkout d'un projet à partir du SVN.

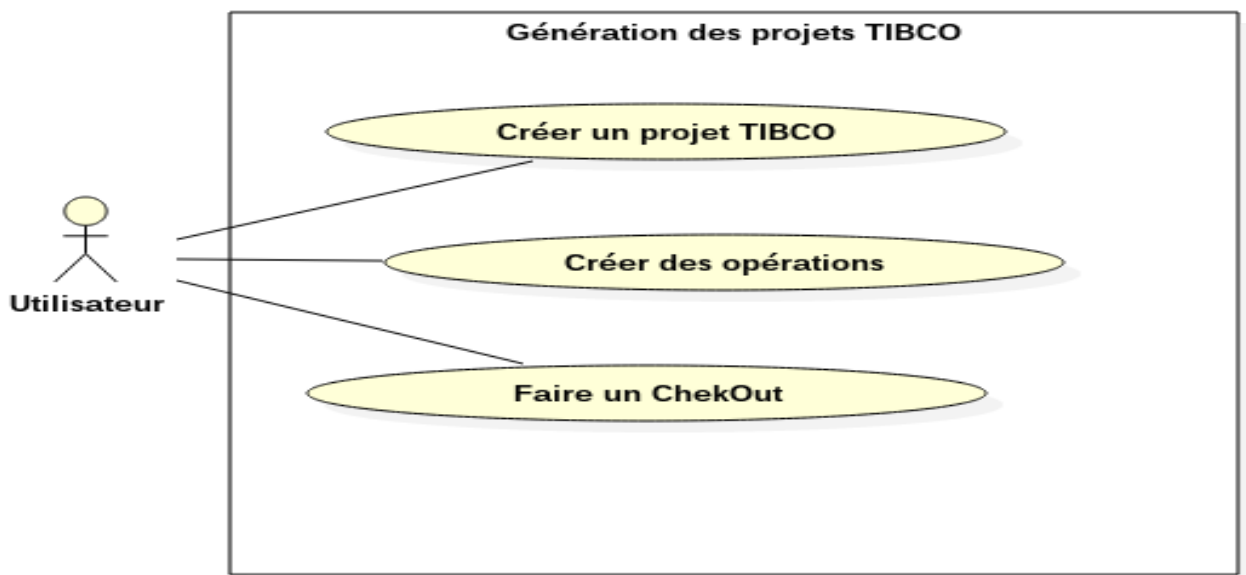


FIGURE 4.1 – « Diagramme de cas d'utilisation de la génération des projets »

4.2.2 Description de quelques scénarii

Dans cette partie, nous étudions quelques scénarii relatifs à la génération des projets TIBCO.

4.2.2.1 Scénario du cas d'utilisation " Créer un projet TIBCO "

La figure 4.2 ci-dessous représente le diagramme de séquence système du scénario de création d'un projet TIBCO.

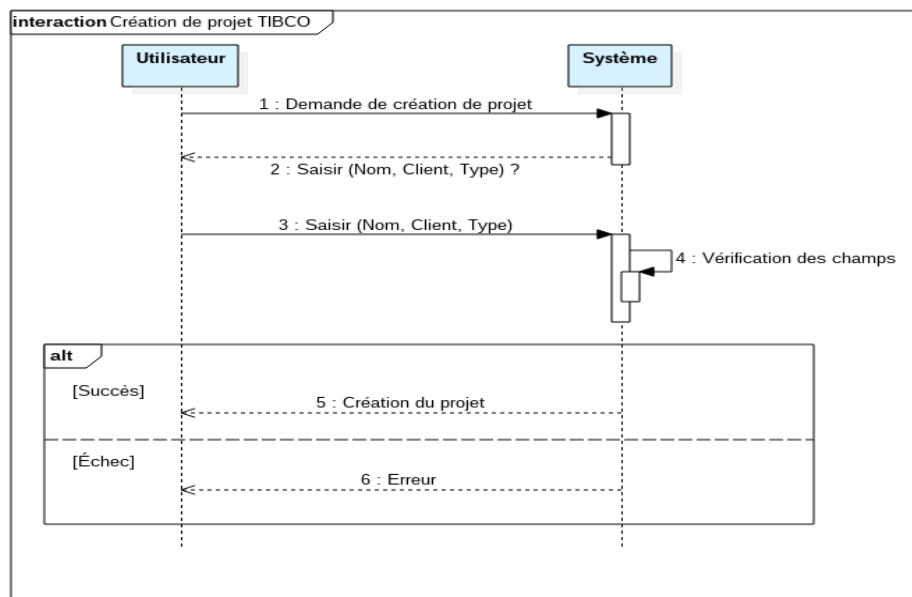


FIGURE 4.2 – « Diagramme de séquence système de création d'un projet TIBCO »

Pour créer un nouveau projet TIBCO, l'utilisateur doit saisir le nom du projet, choisir le client et le type du projet. Si les données saisies sont correctes alors le projet sera créé. Si les données sont erronées alors un message d'erreur apparaîtra.

4.2.2.2 Scénario du cas d'utilisation " Créer des opérations "

La figure 4.3 ci-dessous représente le diagramme de séquence système du scénario de création d'une opération.

Pour créer une opération, l'utilisateur doit saisir le nom de l'opération. Si les données saisies sont correctes alors l'opération sera créée. Si les données sont erronées alors un message d'erreur apparaîtra.

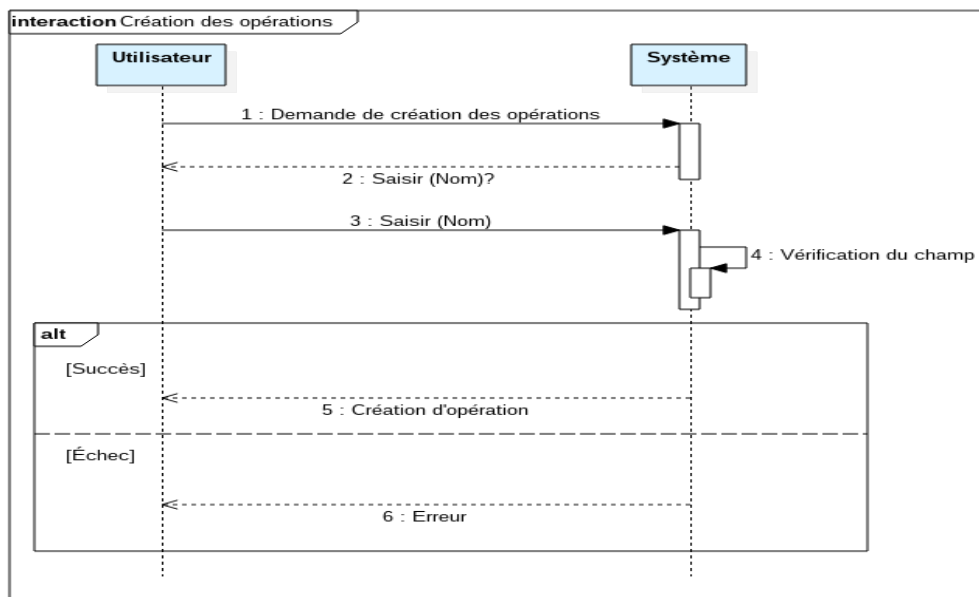


FIGURE 4.3 – « Diagramme de séquence système de création d’une opération »

4.3 Conception

Cette section présente la phase de conception du sprint Génération des projets TIBCO. Nous présentons le diagramme de paquets ainsi que le diagramme d’activités.

4.3.1 Diagramme de paquets

La figure 4.4 ci-dessous illustre le diagramme de paquets du premier sprint.

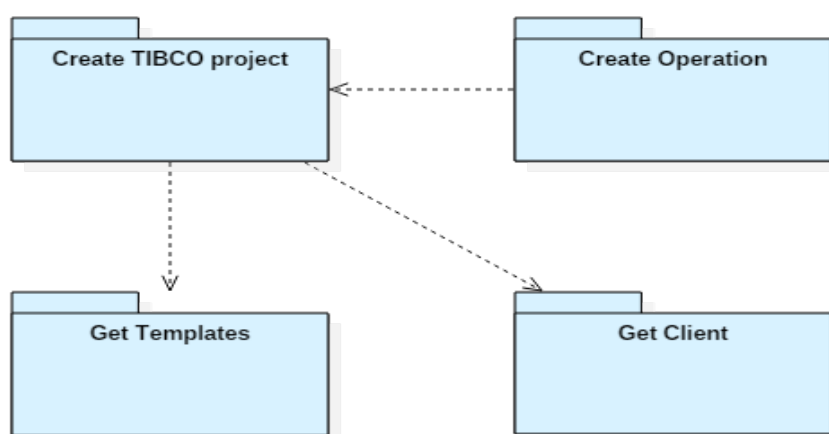


FIGURE 4.4 – « Diagramme de paquets relatif à la génération des projets »

- **Create TIBCO project** : Contient notre logique métier pour la création des projets TIBCO.
- **Create Operation** : Contient notre logique métier pour la création des opérations.
- **Get Templates** : C'est le service responsable de la récupération des templates des projets.
- **Get Client** : C'est le service responsable de la récupération des clients.

4.3.2 Diagramme d'activités de la création des projets TIBCO

La figure 4.5 ci-dessous illustre le processus de création d'un nouveau projet TIBCO. En effet, l'utilisateur commence par saisir le nom du projet, il choisit le client ainsi que le type de projet. Après la validation, si le projet existe alors une exception se produit, sinon le projet sera créé et configuré.

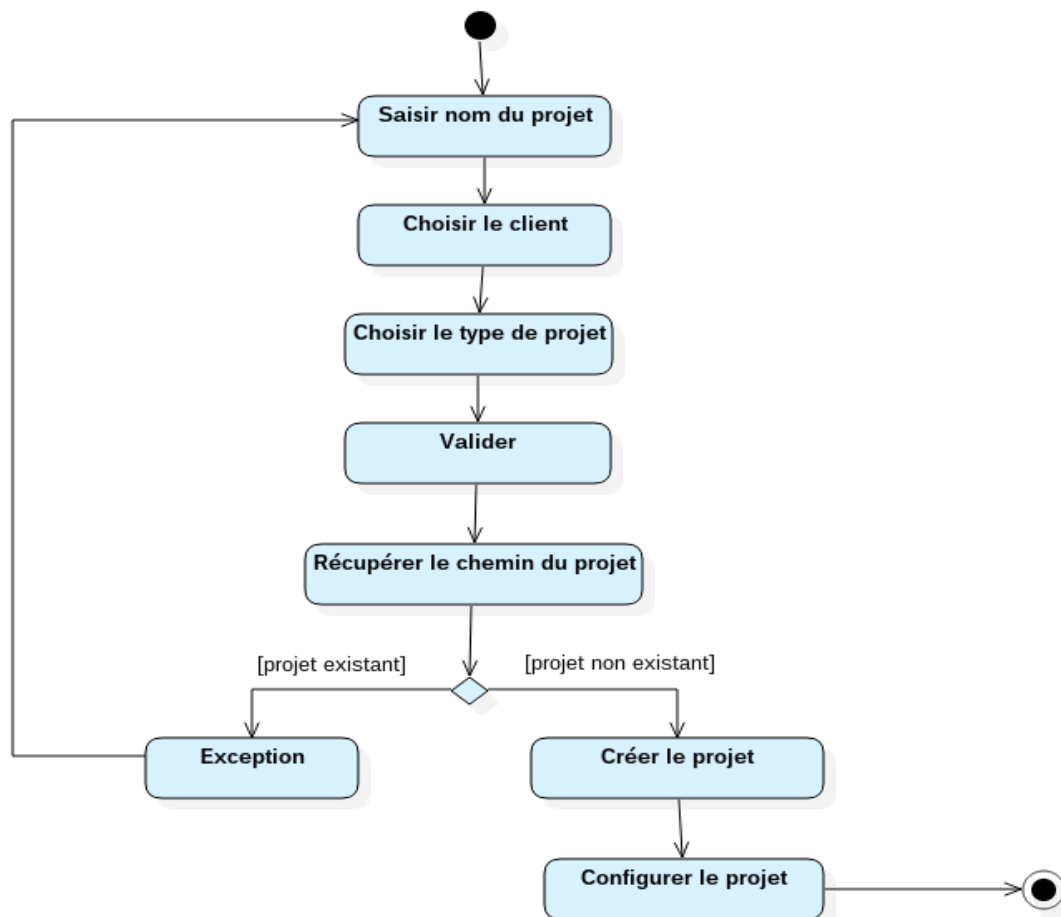


FIGURE 4.5 – « Diagramme d'activités de la création d'un nouveau projet »

4.4 Réalisation du sprint 1

La figure 4.6 ci-dessous représente l'interface de la création d'un nouveau projet TIBCO. En effet, l'utilisateur doit saisir le nom du projet, choisir le nom du client ainsi que le type du projet et cliquer sur "Submit".

FIGURE 4.6 – « Création d'un nouveau projet TIBCO »

La figure 4.7 représente la liste des projets. À partir du bouton "Choose", l'utilisateur peut faire un checkout (Update) du projet. Il peut également le modifier en cliquant sur "Modify".

La figure 4.8 représente les différentes actions qu'un utilisateur peut les faire après la création du projet. Il peut créer des opérations, créer des ressources, invoquer une opération JMS et exposer des opérations SOAP.

La figure 4.9 représente l'interface de la création des opérations. Après avoir cliqué sur "Add Operations", l'utilisateur est mené à cette interface, il peut ajouter une opération et peut également la supprimer.

Manage Your Projects..

PROJECT Generate New Project Upload New Project

LIST OF PROJECTS

Show 10 entries

Project Name	Type	Client	Creation Date	Status	Action
Tibco_OrderManagement_BP	BP	Client	today	starting	Choose
Tibco_OrderManagement_BP_Test	BP	Client	today	starting	<div> Modify Update Download </div>
Tibco_OrderManagement_BP	BP	Client	today	starting	Choose
Tibco_OrderManagement_BP1	BP	Client	today	starting	Choose
Tibco_OrderManagement_BP	BP	Client	today	starting	Choose
Tibco_OrderManagement_BP	BP	Client	today	starting	Choose
Tibco_OrderManagement_BP	BP	Client	today	starting	Choose
Tibco_OrderManagement_BP	BP	Client	today	starting	Choose

FIGURE 4.7 – « Liste des projets TIBCO »

PROJECT ACTION

Tibco_OrderManagement_BP

Add Operations	Generate Test Project
Add Ressources	Expose SOAP Operation
Add JMS	Generate Documentation
WSDL Upload	

FIGURE 4.8 – « Liste des actions relatives à chaque projet »

The screenshot shows a web application titled 'PROJECT ACTION'. The main content area is titled 'Tibco_OrderManagement_BP'. It contains a table with two columns: 'Operation Name' and 'Action'. The table has three rows: 'Operation1' with a trash icon, 'Opearion2' (note the typo) with a trash icon, and an empty row with a plus icon. Below the table is a 'Submit' button.

Operation Name	Action
Operation1	
Opearion2	
<input type="text"/>	

FIGURE 4.9 – « Création des opérations »

Conclusion

Dans ce chapitre nous avons présenté le cycle de développement de notre premier sprint. Nous avons commencé, dans un premier temps, par exposer la spécification fonctionnelle via le diagramme de cas d'utilisation et de séquences système. Dans un second temps, nous avons présenté la conception de notre sprint par le diagramme de paquets ainsi que le diagramme d'activités. Enfin, nous avons présenté les principaux IHM qui composent la génération de nouveaux projets TIBCO. Le chapitre suivant est consacré à notre deuxième sprint.

Chapitre 5

Sprint 2 : Invocation JMS

Dans ce chapitre qui représente le deuxième sprint, nous allons détailler les différentes étapes ayant pour objectif l'invocation JMS (Java Message Service). Ce sprint permet de créer et configurer un processus TIBCO qui doit invoquer une opération JMS distante, d'ajouter des logs au processus (LogInfo, LogException). Il permet aussi de créer une connexion JMS et Schema XSD et assurer le mapping des variables.

Nous commençons tout d'abord par présenter le sprint backlog. Nous entamons par la suite la partie spécification des besoins et conception. Enfin, nous présentons l'interface Homme-Machine dans la partie réalisation.

5.1 Sprint Backlog

Le tableau 5.1 présente notre sprint backlog.

TABLE 5.1 – Backlog Sprint 2

Exigence	Sous tâche
2.1	Afficher tous les projets de l'opérateur.
2.2	Afficher les opérations qui sont exposées JMS.
2.3	Créer et configurer la connexion JMS.
2.4	Créer les variables globales de la connexion JMS.

2.5	Créer le schéma XSD qui correspond à l'opération à invoquer.
2.6	Création du processus responsable à l'invocation (Synchrone ou Asynchrone).
2.7	Ajouter les logs nécessaires (Log Exception ou Log Info).
2.8	Configurer le processus et faire le mapping des variables.

5.2 Spécification des besoins

Afin de spécifier nos besoins, nous allons utiliser le diagramme de cas d'utilisation et le diagramme de séquence.

5.2.1 Diagramme de cas d'utilisation relatif à l'utilisateur

Dans cette partie, nous présentons le diagramme de cas d'utilisation principal. Cette phase représente la vue fonctionnelle de notre sprint. Ce dernier est représenté par la figure 5.1 ci-dessous.

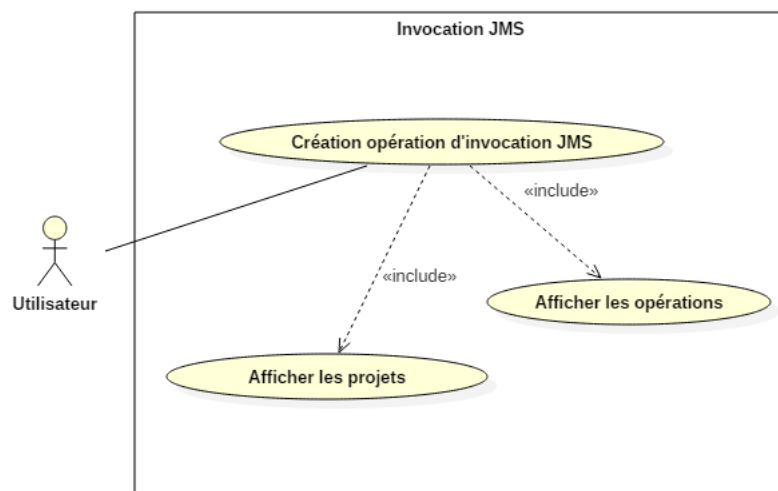


FIGURE 5.1 – « Diagramme des cas d'utilisation de l'invocation JMS »

Ce diagramme de cas d'utilisation présente les différentes fonctionnalités de ce sprint que l'utilisateur de notre application peut faire. En effet, l'utilisateur peut afficher les projets à partir desquels il peut invoquer. Il peut aussi afficher des opérations dans chaque projet ainsi que créer des opérations d'invocation JMS.

5.2.2 Diagramme de séquence du cas d'utilisation : Création d'opération d'invocation JMS

La figure 5.2 ci-dessous représente le diagramme de séquence système de l'invocation JMS. En effet, pour créer une opération d'invocation JMS, l'utilisateur doit afficher les projets à partir desquels il peut invoquer. Puis, l'utilisateur doit afficher les opérations du projet choisi. Une fois les opérations sont affichées, l'utilisateur doit choisir une opération et choisir le type d'invocation et valider.

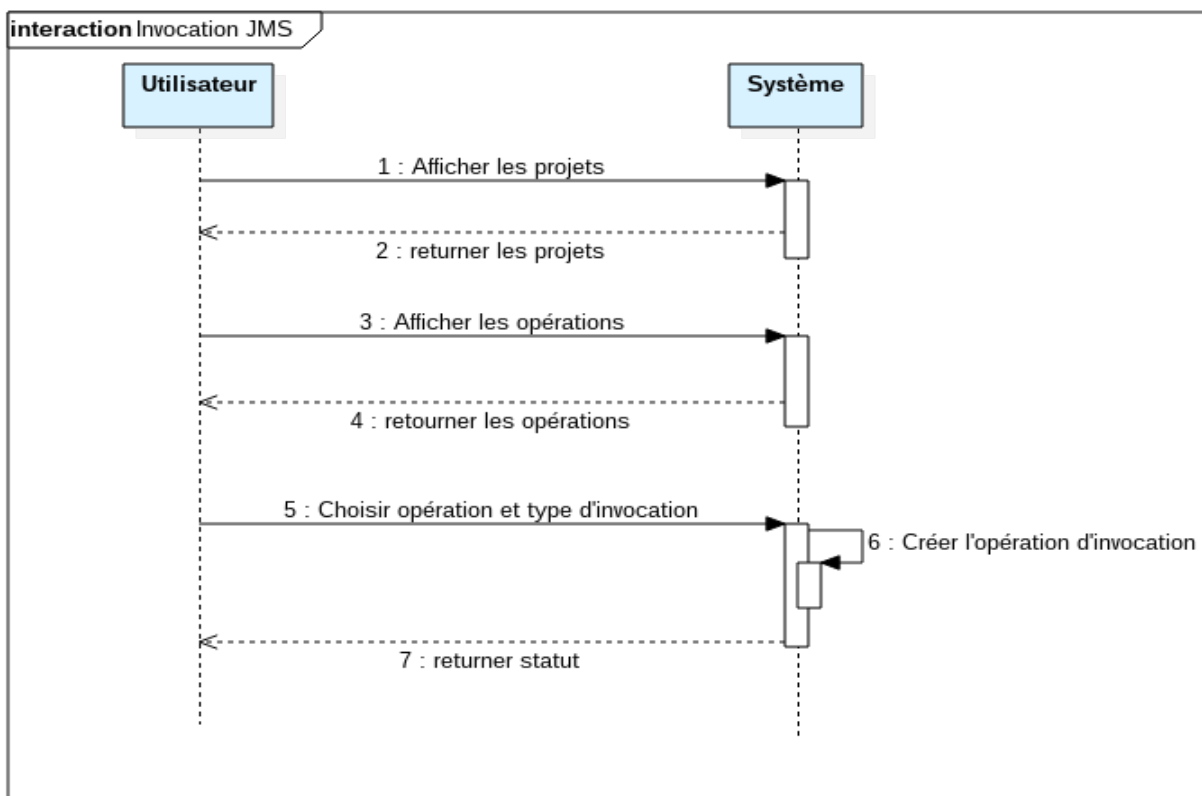


FIGURE 5.2 – «Diagramme de séquence de l'invocation JMS»

5.3 Conception

Dans cette partie nous allons détailler la phase de conception du service invocation JMS en s'appuyant sur le diagramme de paquets ainsi que le diagramme d'activités

5.3.1 Diagramme de paquets

La figure 5.3 ci-dessous illustre le diagramme de paquets du service d'invocation JMS.

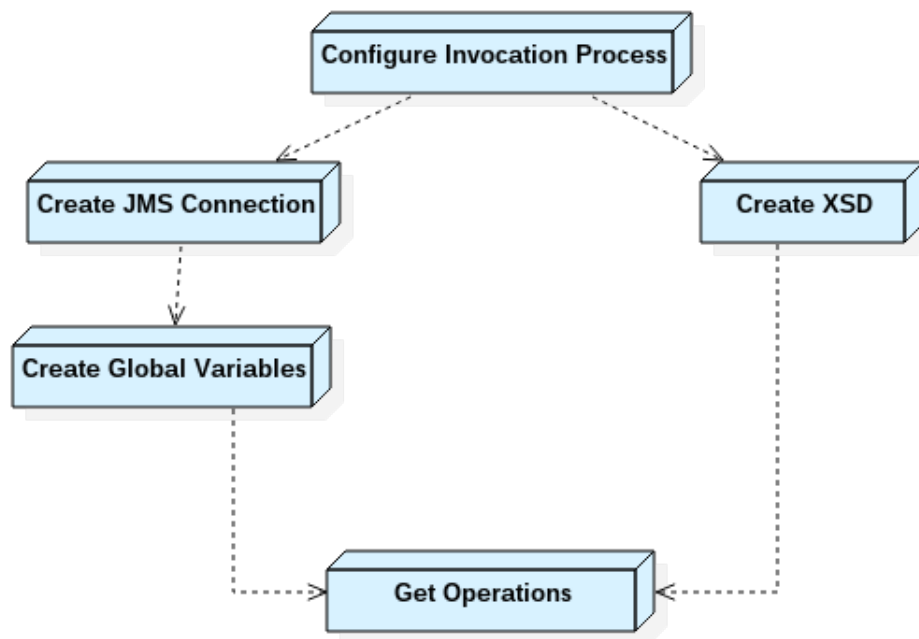


FIGURE 5.3 – « Diagramme de paquets invocation JMS »

- **Get Operations** : Assure la récupération des opérations qui sont exposées en JMS.
- **Create Global Variables** : Assure la création des variables globales qui vont être utilisées dans la configuration de la connexion JMS.
- **Create JMS connection** : Responsable de la création de la connexion JMS.
- **Create XSD** : Assure la création des schémas XSD de chaque opération.
- **Configure Invocation Process** : Assure la configuration du processus d'invocation.

5.3.2 Diagramme d'activités

La figure 5.4 représente le diagramme d'activités relatif à l'invocation JMS.

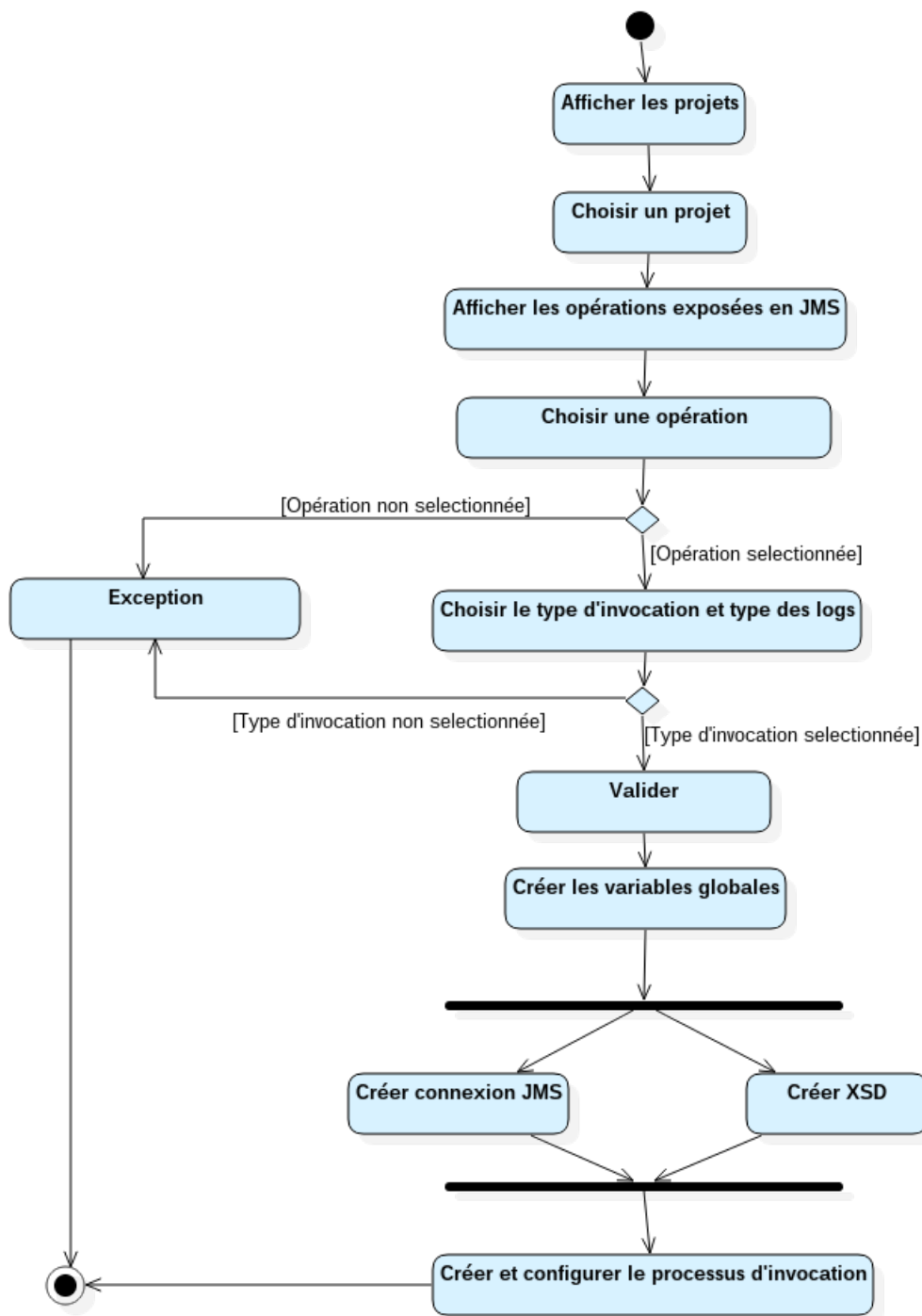


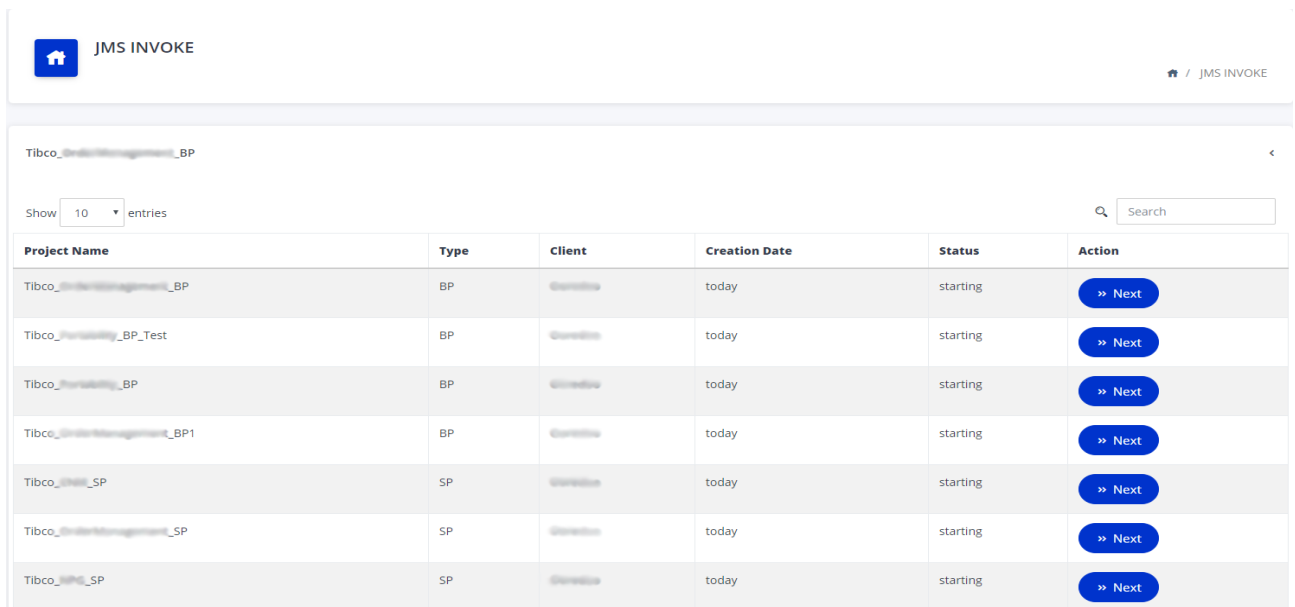
FIGURE 5.4 – « Diagramme d'activités du processus d'invocation JMS »

Pour lancer le processus d'invocation JMS, nous commençons tout d'abord par afficher la liste des projets et choisir le projet concerné. Ensuite, nous choisissons une opération exposée en JMS. Si aucune opération n'est sélectionnée, cela nous amène à une exception. Sinon, nous

choisissons le type d'invocation ainsi que le type des logs. De même, si aucun type d'invocation n'est sélectionné, alors une exception se produit. Dans le cas contraire et après validation, nous passons à la création des variables globales. Une fois les variables créées, la connexion JMS ainsi que le schéma XSD sont créés. Enfin, le processus d'invocation JMS est créé et configuré.

5.4 Réalisation du sprint 2

La figure 5.5 représente l'interface d'invocation JMS. L'utilisateur doit choisir le projet à partir duquel il va invoquer puis il doit cliquer sur "Next".



The screenshot shows the 'JMS INVOKE' interface. At the top, there is a header with a home icon and the text 'JMS INVOKE'. Below the header, there is a breadcrumb trail 'Tibco_..._BP' and a search bar. A table lists several projects with columns for Project Name, Type, Client, Creation Date, Status, and Action. Each row has a '» Next' button in the Action column.

Project Name	Type	Client	Creation Date	Status	Action
Tibco_..._BP	BP	...	today	starting	» Next
Tibco_..._BP_Test	BP	...	today	starting	» Next
Tibco_..._BP	BP	...	today	starting	» Next
Tibco_..._BP1	BP	...	today	starting	» Next
Tibco_..._SP	SP	...	today	starting	» Next
Tibco_..._SP	SP	...	today	starting	» Next
Tibco_..._SP	SP	...	today	starting	» Next

FIGURE 5.5 – « Liste des projets à invoquer »

Après que l'utilisateur clique sur "Next", l'interface représentée par la figure 5.6 ci-dessous s'affiche. L'utilisateur doit choisir alors une opération et clique sur "Next". Puis, il choisit le type d'invocation et le type de journal (Log) et valider.

The screenshot shows a web application titled "JMS INVOKE". At the top, there is a header bar with a home icon and the text "JMS INVOKE". Below the header, there is a breadcrumb trail showing "Tibco_" followed by a blurred path. The main content area is divided into two sections. The first section, titled "Operation List", contains a dropdown menu with "AddNumbersToPool" selected and a blue button labeled "» Next". The second section, titled "Choose Log Type and Invocation Type", contains two columns of options. The "Add Log" column has two checkboxes: "Log Info" and "Log Exception". The "Type Of Invocation" column has two radio buttons: "Call Synchron" and "Call Asynchron". A blue button labeled "Submit" is at the bottom of this section.

FIGURE 5.6 – « Invocation JMS »

Conclusion

Dans ce chapitre nous avons présenté le cycle de développement de notre deuxième sprint ayant pour objectif la réalisation du service d'invocation JMS. Nous avons commencé par une spécification fonctionnelle grâce au diagramme de cas d'utilisation et au diagramme de séquences système. Ensuite, nous avons présenté notre conception en utilisant le diagramme de paquets ainsi que le diagramme d'activités. Enfin, nous avons présenté les principaux IHM qui composent la partie invocation JMS. Le chapitre suivant est consacré à notre troisième sprint.

Chapitre 6

Sprint 3 : Exposition SOAP

Ce chapitre qui illustre le cycle de développement du troisième sprint, nous allons détailler les différentes étapes ayant pour objectif l'exposition SOAP. Ce sprint permet de créer et configurer des processus (Web Service) pour chaque opération. Il permet aussi de créer le WSDL (WSDL Messages et WSDL PortType) ainsi que le service agent (Service SOAP).

Nous commençons tout d'abord par présenter le sprint backlog. Nous entamons par la suite la partie spécification des besoins et la conception. Enfin, nous présentons l'interface Homme-Machine dans la partie réalisation.

6.1 Sprint Backlog

Le tableau 6.1 présente notre sprint backlog.

TABLE 6.1 – Backlog Sprint 3

Exigence	Sous tâche
2.1	Afficher tous les projets.
2.2	Afficher les opérations du projet.
2.3	Créer le WSDL (Messages et PotType).
2.4	Créer des processus (Web Service) pour chaque opération.
2.5	Créer le service agent.

6.2 Spécification des besoins

Afin de spécifier nos besoins, nous allons utiliser le diagramme de cas d'utilisation et le diagramme de séquence.

6.2.1 Diagramme de cas d'utilisation relatif à l'utilisateur

Dans cette partie, nous présentons le diagramme de cas d'utilisation relatif à l'exposition SOAP. Cette phase représente la vue fonctionnelle de notre sprint. Ce dernier est représenté par la figure 6.1 ci-dessous.

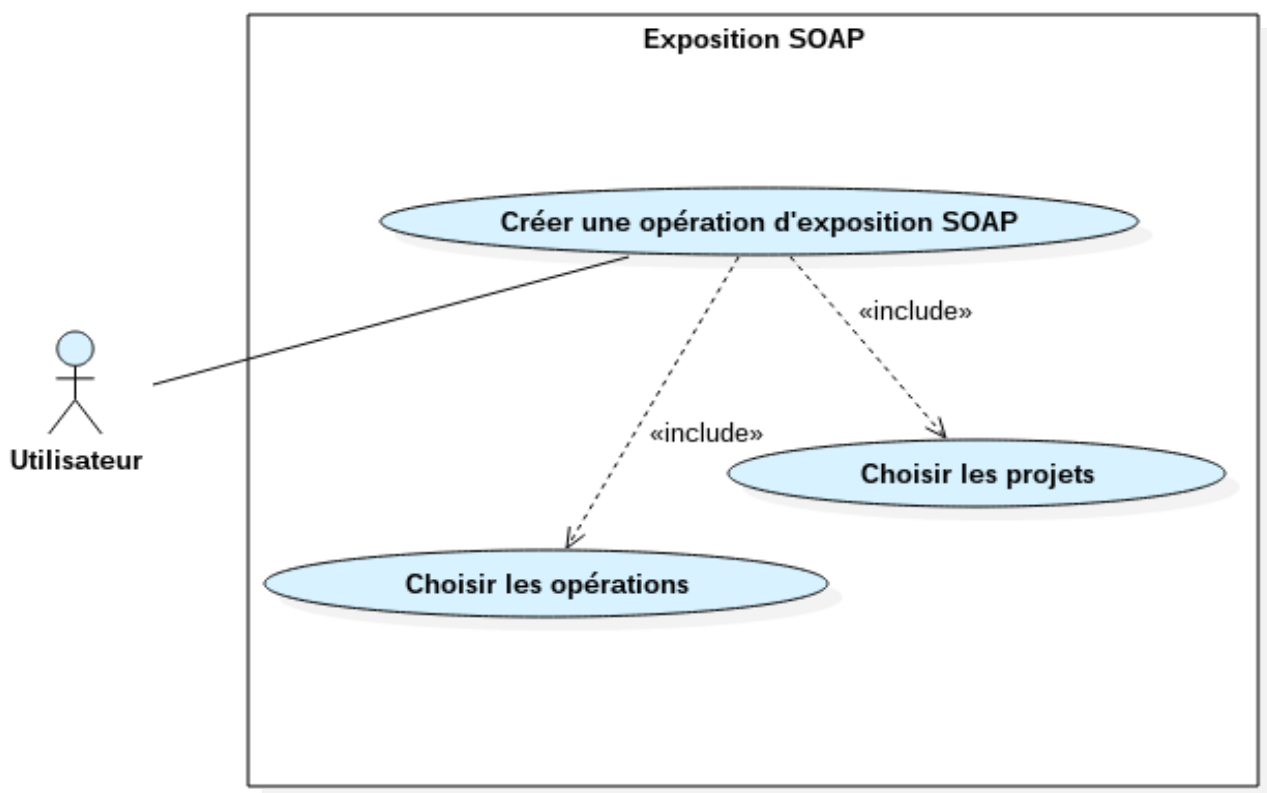


FIGURE 6.1 – « Diagramme des cas d'utilisation de l'exposition SOAP »

Ce diagramme de cas d'utilisation présente les différentes fonctionnalités de ce sprint. Afin de créer une opération d'exposition SOAP, l'utilisateur doit choisir le projet concerné ainsi que ses opérations.

6.2.2 Diagramme de séquence du cas d'utilisation : Création d'opération d'exposition SOAP

La figure 6.2 ci-dessous représente le diagramme de séquence système. En effet, l'utilisateur commence par afficher les projets existants. Il choisit un projet et affiche ses opérations. Il choisit ensuite les opérations à exposer. Après la création des opérations d'exposition, un statut sera affiché.

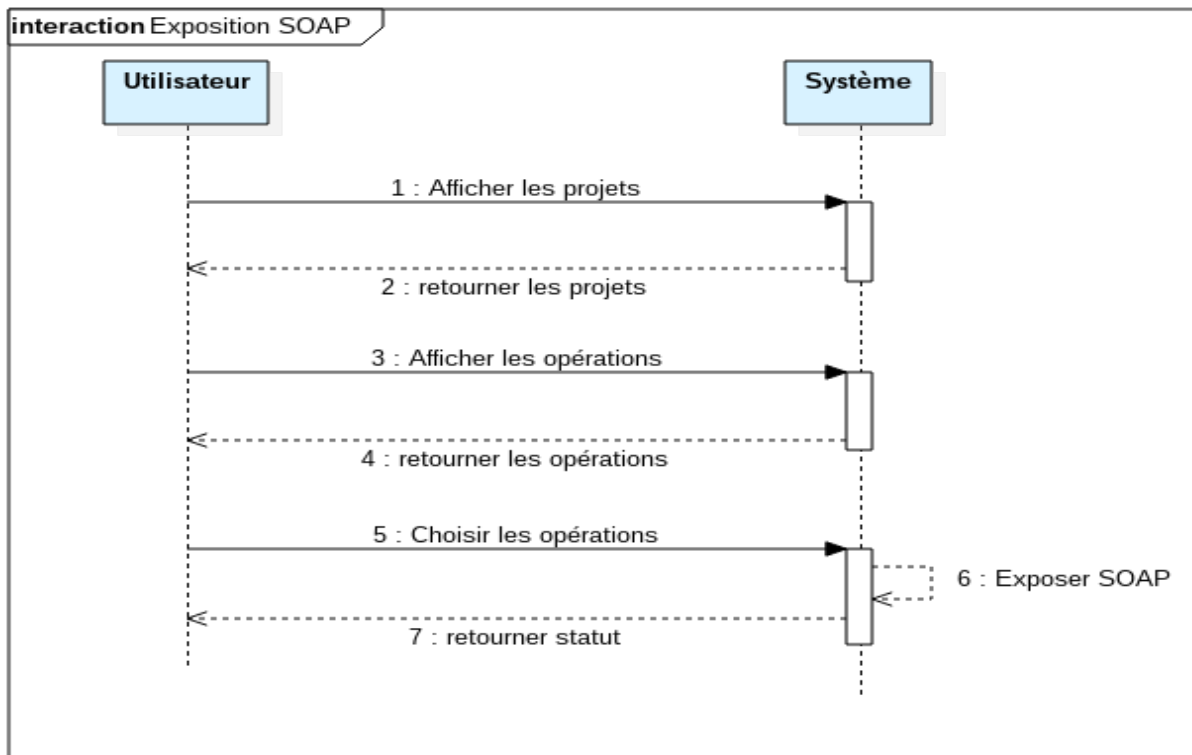


FIGURE 6.2 – « Diagramme de séquence de l'exposition SOAP »

6.3 Conception

Cette section présente la phase de conception du processus d'exposition SOAP. Nous présentons le diagramme de paquets ainsi que le diagramme d'activités.

6.3.1 Diagramme de paquets

La figure 6.3 ci-dessous illustre le diagramme de paquets de l'exposition SOAP.

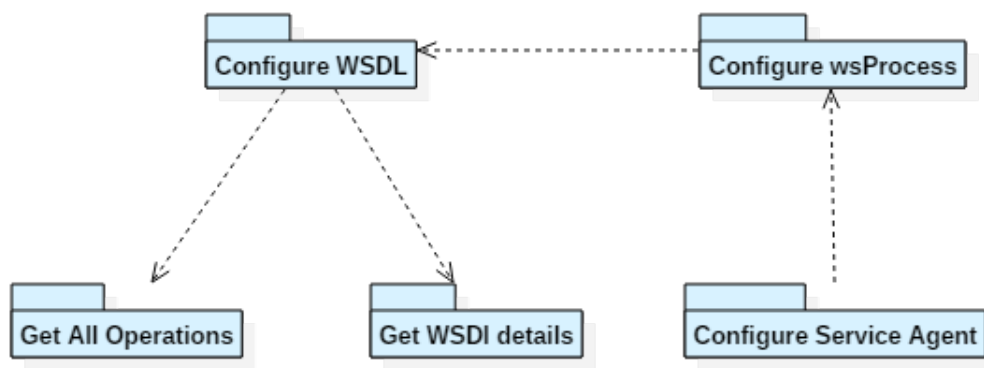


FIGURE 6.3 – « Diagramme de paquet de l'exposition SOAP »

- **Get All Operations** : Assure la récupération des opérations d'un projet.
- **Get WSDL Details** : Assure la récupération des données nécessaires (Namespace, Schema location, ect..) pour la création du WSDL.
- **Configure WSDL** : Assure la création du WSDL (Messages et PortType).
- **Configure wsProcess** : Assure la création des processus Web Service.
- **Configure Service Agent** : Assure la création et la configuration du service agent qui va contenir tous les Web services nécessaires.

6.3.2 Diagramme d'activités

La figure 6.4 ci-dessous représente le diagramme d'activités détaillé de l'exposition SOAP. En effet, l'utilisateur choisit un projet parmi les projets affichés. Ensuite, il affiche toutes les opérations existantes de ce projet. Si aucune opération n'est sélectionnée alors une exception se produit. Si au moins une opération est sélectionnée alors le WSDL est créé et configuré (WSDL Message et PortType). Enfin le web service ainsi que le service agent sont créés et configurés.

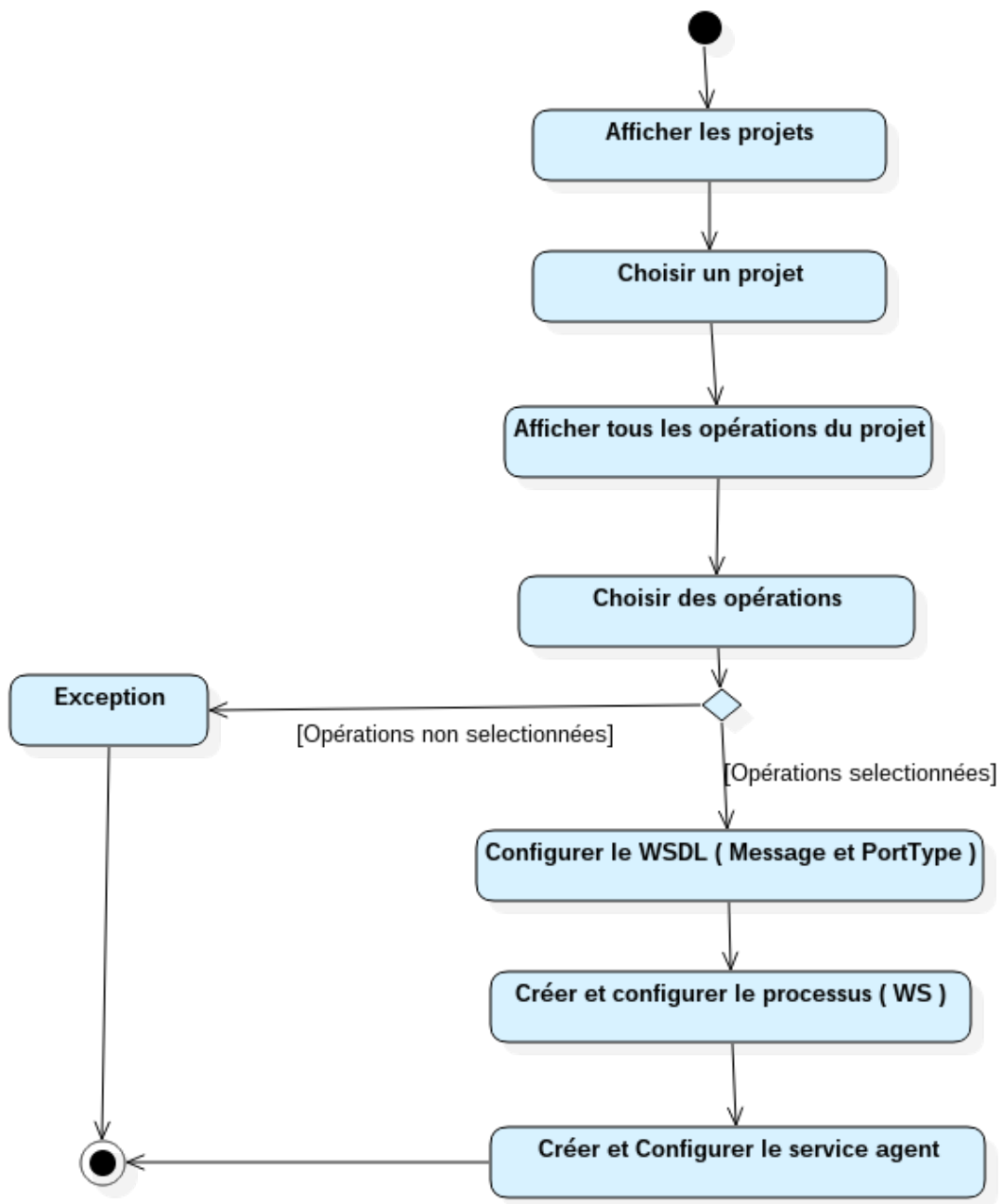


FIGURE 6.4 – « Diagramme d'activités détaillé de l'exposition SOAP »

6.4 Réalisation du sprint 3

La figure 6.5 représente la liste des actions relatives à un projet. Pour l'exposition SOAP, l'utilisateur doit choisir le projet et cliquer sur "Expose SOAP".

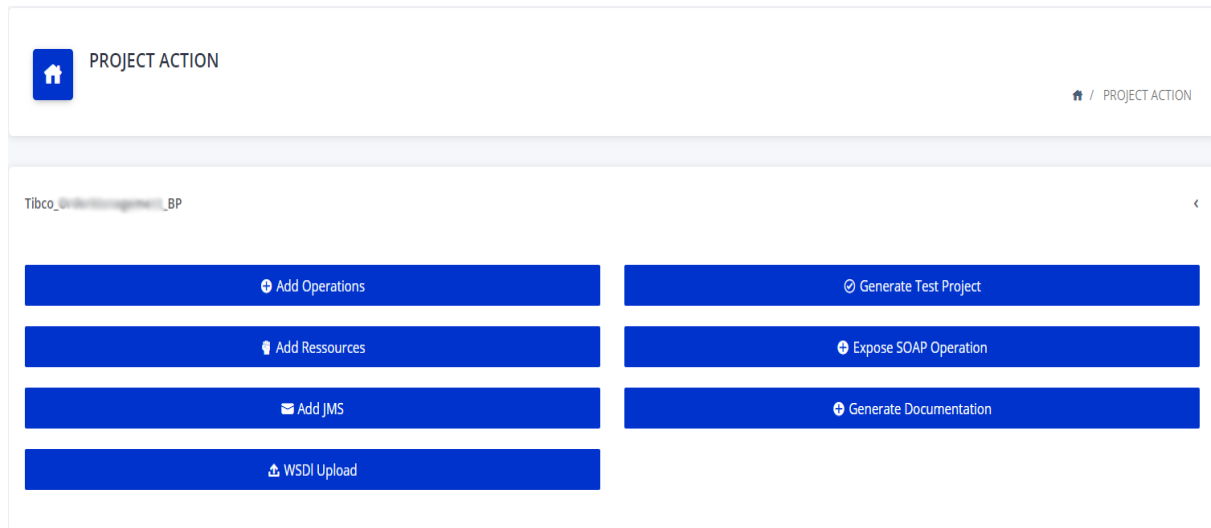


FIGURE 6.5 – « Liste des actions »

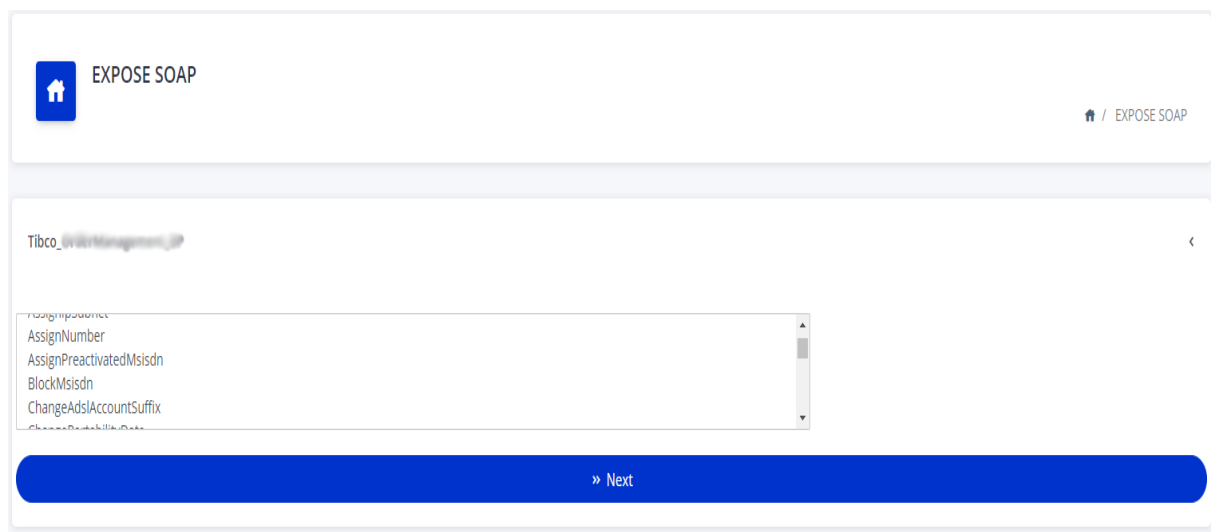


FIGURE 6.6 – « Exposition SOAP »

La figure 6.6 représente l'interface de l'exposition SOAP. L'utilisateur doit choisir les opérations à exposer et cliquer sur "Next".

Conclusion

Ce chapitre illustre le cycle de développement du troisième sprint. Nous avons commencé, dans un premier temps, par exposer la spécification fonctionnelle via les diagrammes de cas d'utilisation et de séquences système. Dans un second temps, nous avons présenté la conception de notre solution par le diagramme de paquets de l'exposition SOAP ainsi que le diagramme d'activités de ce processus. Enfin, nous avons présenté les principaux IHM qui composent cette partie. Le chapitre suivant est consacré à notre quatrième sprint.

Chapitre 7

Sprint 4 : Intégration Continue

Dans ce chapitre représentant le dernier sprint, nous allons détailler les différentes étapes ayant pour objectif l'intégration continue. En effet, ce sprint permet à l'intégrateur de configurer et de lancer un job Jenkins qui couvre les étapes suivantes : Checkout, Construction, Validation, Génération d'archive, Génération d'artefacts, Déploiement.

Tout d'abord, nous présentons le sprint backlog. Nous entamons par la suite la partie spécification des besoins ainsi que la conception de ce processus. Enfin, nous exposerons la partie réalisation.

7.1 Sprint Backlog

Le tableau 7.1 présente notre sprint backlog.

TABLE 7.1 – Backlog Sprint 4

Exigence	Sous tâche
2.1	Assurer le checkout.
2.2	Faire la construction (Build) accompagné de la validation.
2.3	Générer l'Enterprise Application Archive (Ear).
2.4	Assurer le déploiement de l'application sur TIBCO Administrator.
2.5	Créer les artefacts.
2.6	Notifier l'intégrateur par l'envoi d'un email accompagné d'un rapport.

7.2 Spécification des besoins

Afin de spécifier nos besoins, nous allons utiliser le diagramme de cas d'utilisation et le diagramme de séquence.

Diagramme de cas d'utilisation relatif à l'intégrateur

Dans cette partie, nous présentons le diagramme de cas d'utilisation relatif à l'intégrateur. Ce dernier est représenté par la figure 7.1 ci-dessous.

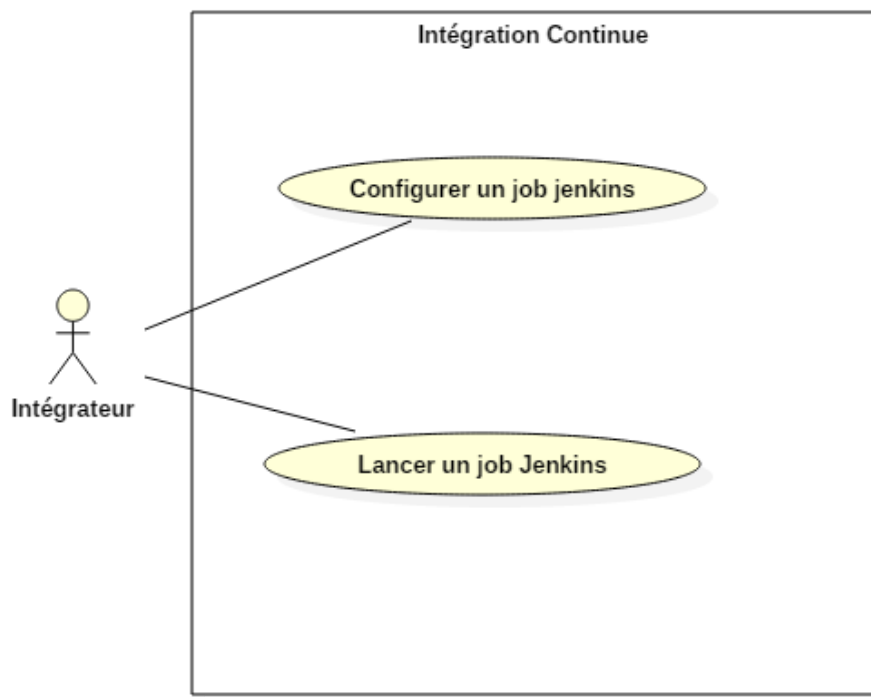


FIGURE 7.1 – « Diagramme des cas d'utilisation relatif à l'intégrateur »

Ce diagramme de cas d'utilisation présente les différentes fonctionnalités de ce sprint. En effet, l'intégrateur peut configurer un job Jenkins pour chaque projet TIBCO. Le Job est lancé automatiquement lors de la détection d'un changement dans le dépôt SVN. L'intégrateur peut aussi lancer le job manuellement en cas de besoin.

7.3 Conception

Cette section présente la phase de conception de la phase d'intégration continue. Nous présentons le diagramme de déploiement ainsi que le diagramme d'activités.

7.3.1 Diagramme de déploiement

La figure 7.2 ci-dessous représente le diagramme de déploiement relatif à l'intégration continue.

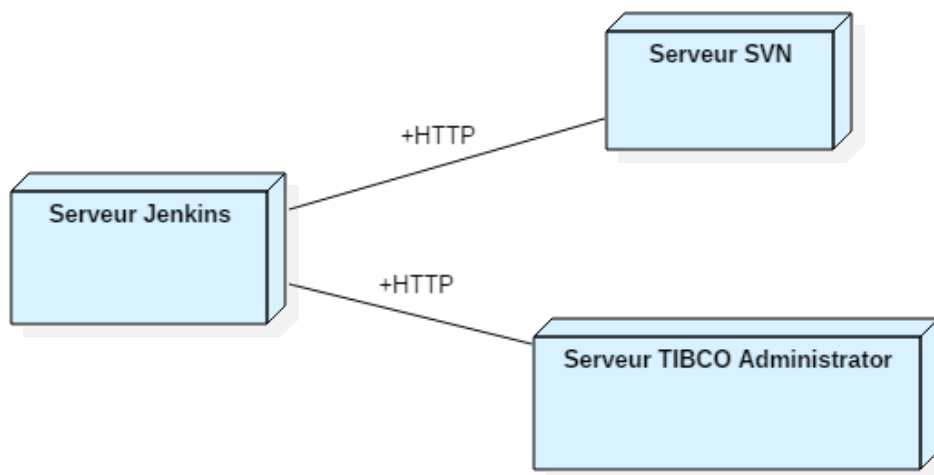


FIGURE 7.2 – « Diagramme de déploiement relatif à l'intégration continue »

- **Serveur Jenkins** : C'est le serveur responsable de l'intégration continue.
- **Serveur SVN** : Contient le dépôt des projets.
- **Serveur TIBCO Administrator** : C'est le serveur responsable du déploiement des applications TIBCO.

7.3.2 Diagramme d'activités

La figure 7.3 ci-dessous représente le diagramme d'activités relatif à l'intégration continue. En effet, le job jenkins se met à l'écoute de dépôt SVN. Chaque minute, il compare les deux dépôts, s'il n'y a aucune modification du code source alors le job se termine, s'il détecte une modification alors il lance le checkout de code source depuis le dépôt SVN. Par la suite, il assure la construction (Build) et la validation du code source. Si le code est non validé alors un email

d'échec sera envoyé vers l'intégrateur, sinon il enchaîne par la construction d'Entreprise Application Archive (EAR). Ensuite, il crée et archive les artefacts. Enfin, il assure le déploiement de l'EAR sur TIBCO Administrator et envoie un email de succès vers l'intégrateur.

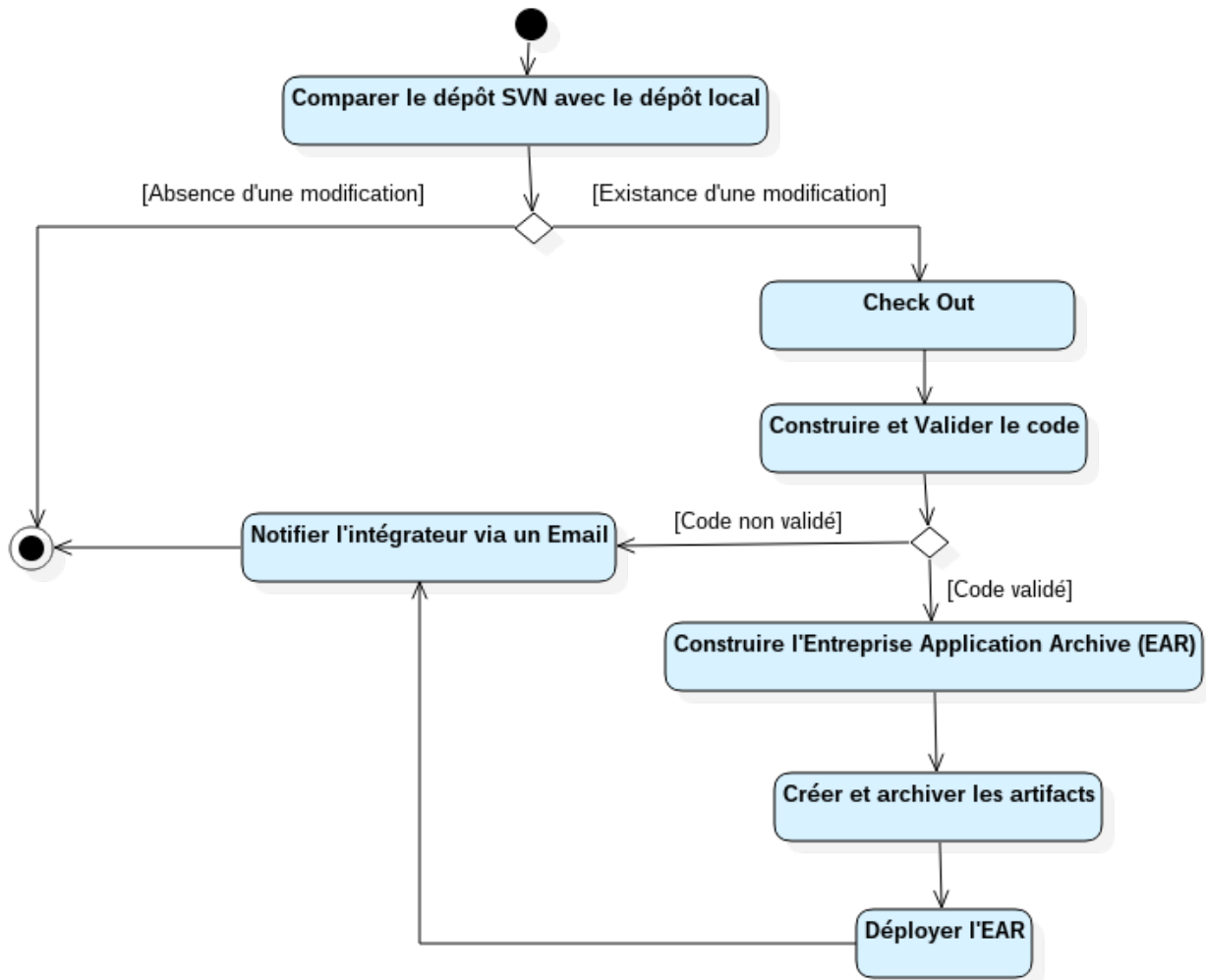


FIGURE 7.3 – « Diagramme d'activité relatif à l'intégration continue »

7.4 Réalisation du Job Jenkins

Dans cette partie, nous allons présenter la réalisation de notre job Jenkins.

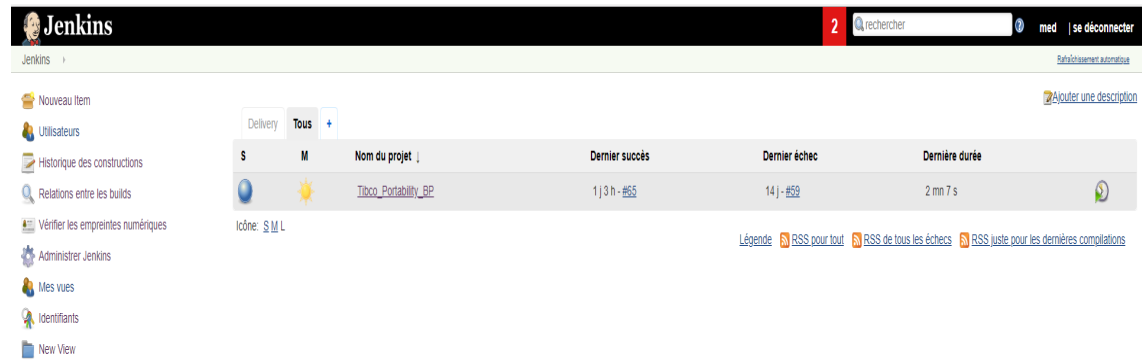


FIGURE 7.4 – « Dashbord Jenkins »

La figure 7.4 représente le dashbord Jenkins. Nous avons créé notre job d'intégration qui va couvrir les étapes citées dans le diagramme d'activités.

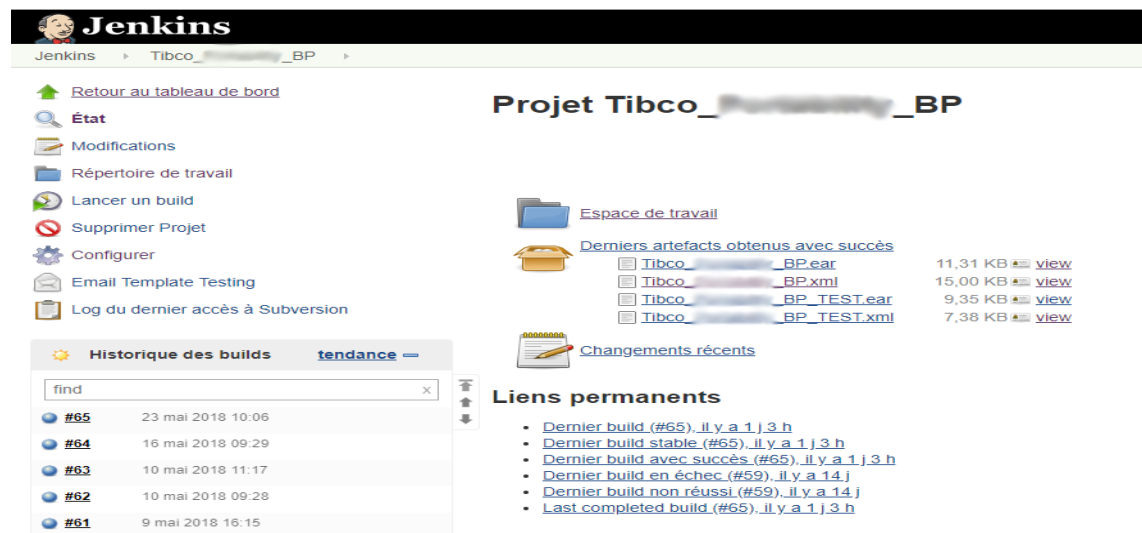


FIGURE 7.5 – « Job d'intégration continue »

La figure 7.5 représente le Job d'intégration. L'intégrateur peut, à partir de cette interface, lancer la construction (Build), configurer le job ou le supprimer. Il peut aussi consulter les derniers artefacts obtenus avec succès.

La figure 7.6 représente la sortie de la console relative au checkout. En effet, ce job est lancé lors un changement dans la base de code SVN.

Sortie de la console

```
Lancé par un changement dans la base de code
Building in workspace C:\Program Files (x86)\Jenkins\workspace\Tibco_Performance_BP
[WS-CLEANUP] Deleting project workspace...
[WS-CLEANUP] Done
Checking out a fresh workspace because C:\Program Files
(x86)\Jenkins\workspace\Tibco_Performance_BP\Tibco_Performance_BP doesn't exist
Cleaning local Directory .Tibco_Performance_BP
Checking out https://talanpfe-157.dom.tti/svn/Tibco\_Performance\_BP at revision '2018-05-25T11:19:05.888
+0100' --quiet
Using sole credentials ghorbel/***** (User And Password For SVN) in realm '<https://talanpfe-157.dom.tti:443> VisualSVN Server'
At revision 14

Checking out a fresh workspace because C:\Program Files
(x86)\Jenkins\workspace\Tibco_Performance_BP\Tibco_Performance_BP_TEST doesn't exist
Cleaning local Directory .Tibco_Performance_BP_TEST
Checking out https://talanpfe-157.dom.tti/svn/Tibco\_Performance\_BP\_TEST at revision '2018-05-25T11:19:05.888 +0100' --quiet
Using sole credentials ghorbel/***** (User And Password For SVN) in realm '<https://talanpfe-157.dom.tti:443> VisualSVN Server'
At revision 5
```

FIGURE 7.6 – « Sortie de la phase de Checkout »

```
C:\tibco\tra\5.7\bin>buildear.exe -x -s -ear /Deployment/Tibco_Performance_BP.archive -o "C:\Program
Files (x86)\Jenkins\workspace\Tibco_Performance_BP\Tibco_Performance_BP.ear" -p "C:\Program Files
(x86)\Jenkins\workspace\Tibco_Performance_BP\Tibco_Performance_BP"
Starting up...
Exception in DesignerResourceStore.loadResourceFrom() uri:.project using unknown file type resource.
Enterprise Archive File has built correctly.
Ear created in: C:\Program Files (x86)\Jenkins\workspace\Tibco_Performance_BP\Tibco_Performance_BP.ear
```

FIGURE 7.7 – « Sortie de la phase de la construction »

```
C:\tibco\tra\5.7\bin>AppManage -deploy -ear "C:\Program Files
(x86)\Jenkins\workspace\Tibco_Performance_BP\Tibco_Performance_BP.ear" -deployconfig "C:\Program Files
(x86)\Jenkins\workspace\Tibco_Performance_BP\Tibco_Performance_BP.xml" -app Tibco_Performance_BP -
domain TALAN -user ghorbel -pw ghorbel
Initializing ...
Finished initialization
Loading archive ...
Finished loading archive
Configuring application ...
Finished configuring application
Deploying application ...
Instance Tibco_Performance_BP created succesfully
Finished deploying application
Finished successfully in 46 seconds
```

FIGURE 7.8 – « Sortie de la phase de déploiement »

Les figures 7.7 et 7.8 représentent respectivement la sortie de la console relative à la construction (Build) et au déploiement de l'application.

```
C:\Program Files (x86)\Jenkins\workspace\Tibco_ghorbelmohamed37_BP>exit 0
Archiving artifacts
Email was triggered for: Always
Sending email for trigger: Always
Sending email to: ghorbelmohamed37@gmail.com
Finished: SUCCESS
```

FIGURE 7.9 – « Sortie de la phase d’archivage et de notification »

La figure 7.9 représente la sortie de la console relative à l’archivage des artefacts et l’envoi des notifications.

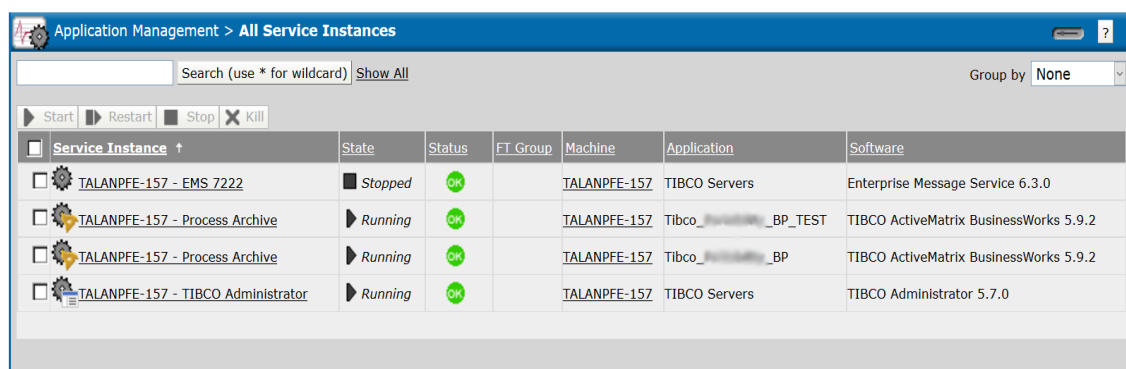


FIGURE 7.10 – « Dashbord TIBCO Administrator »

La figure 7.10 représente la dashbord de TIBCO Administrator. Nous pouvons constater que les deux applications Tibco_BP et Tibco_BP_TEST sont déployées avec succès.

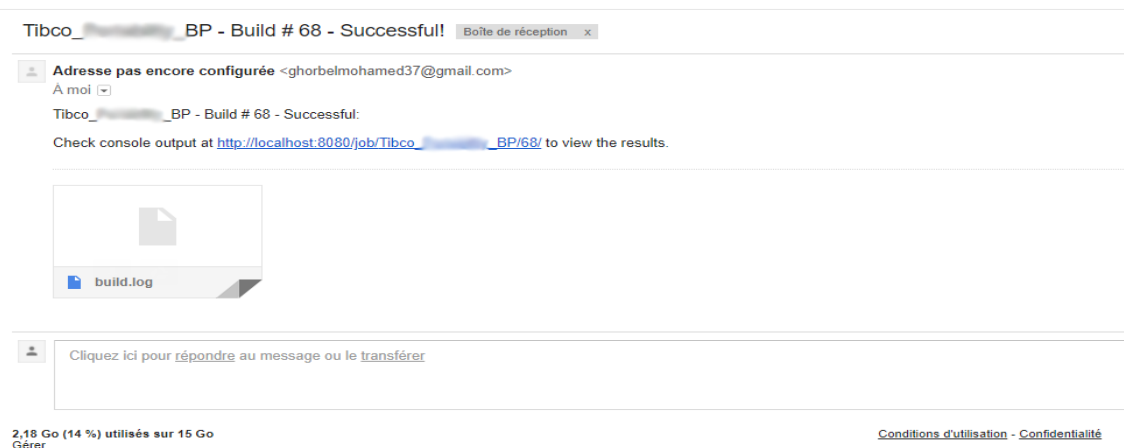



FIGURE 7.11 – « Notification Email »

La figure 7.11 représente l'email envoyé par Jenkins à l'intégrateur. En effet, cet email est envoyé dans les cas de succès ou d'échec. Son objet se compose par le nom du job, le numéro de la construction suivi de l'état du job. Il est accompagné par le journal du Job (log) en pièce-jointe.

Conclusion

Ce chapitre illustre le cycle de  du quatrième sprint. Nous avons commencé, dans un premier temps, par présenter la spécification fonctionnelle via un diagramme de cas d'utilisation. Dans un second temps, nous avons présenté la conception de notre solution via les diagrammes de déploiement et d'activités du processus d'intégration continue. Enfin, nous présentons la réalisation relative au job Jenkins.

Conclusion générale et perspectives

Le processus de développement des projets informatiques est une tâche stratégique délicate qui conditionne la performance des organisations. Bien mené, ses effets agissent directement sur la productivité, l'efficacité, l'image et le climat de travail. Donc l'automatisation de ce processus permet d'accroître la productivité de l'équipe TIBCO Factory.

Le projet de développement d'un framework interne de création de projet TIBCO réalisé par l'équipe TIBCO factory, renforce significativement l'agilité de l'entreprise. L'objectif ultime de Talan Consulting est de faire accroître son évolution et ses progrès technologiques afin d'apporter de nouveaux clients et de nouveaux projets plus performants et plus fiables.

Afin de mener à bien ce projet, nous avons procédé par un ensemble de tâches bien étudiées. Nous étions responsables de la mise en place de l'architecture et la conception de la nouvelle solution ainsi que la réalisation et le test des différents services créés.

Dans le présent rapport, nous avons exposé les différentes tâches primordiales pour réaliser notre solution et aboutir au résultat souhaité. D'abord, nous avons présenté le cadre général de notre travail ainsi que les concepts de base liés à notre projet. Ensuite, nous avons présenté les différents besoins relevés ainsi que le backlog de produit. Enfin, nous avons attaqué la spécification, la conception ainsi que la réalisation des différents sprints.

Bien que l'objectif de notre stage est atteint, nous proposons d'améliorer le framework TTF tenant compte que notre solution est extensible. Nous envisageons de mettre en place un système d'analyse des journaux (Logs) en utilisant le stack ELK (ElasticSearch, Logstash, Kibana) afin de mieux gérer nos applications.


Bibliographie

- [B1] Tabaka, Jean. Gestion de projet vers les méthodes agiles. Éditions Eyrolles, 2006, 268 pages.
- [B2] Mckenna, Dave. The Art of Scrum. Éditions Kindle, 2016, 222 pages.

Netographie

- [1] <http://www.talan.tn/> (dernière consultation le 21/03/2018)
- [2] <http://www.informationbuilders.fr/esb-bus-de-service-d-entreprise>
(dernière consultation le 21/03/2018)
- [3] https://www.service-architecture.com/articles/web-services/service-oriented_architecture_soa_definition.html (dernière consultation le 22/02/2018)
- [4] <https://docs.tibco.com/> (dernière consultation le 21/03/2018)
- [5] <https://www.pcsoft.fr/windev/index.html>
(dernière consultation le 21/03/2018)
- [6] <https://spring.io/>
(dernière consultation le 01/05/2018)
- [7] <https://angular.io/docs> (dernière consultation le 11/05/2018)
- [8] <https://jenkins.io/doc/> (dernière consultation le 11/05/2018)
- [9] <https://subversion.apache.org/> (dernière consultation le 11/05/2018)

ملخص

يعتبر التكامل المستمر والتشغيل الآلي مهام أساسية و مهمة في قطاعات تطوير البرمجيات. يدرك فريق "تبيكو" أهمية هذه العملية التي تتم حاليًا يدويًا. هذا يسمح للشركات بتحسين الأداء وتقليل المخاطر الكامنة. ومن هنا جاءت الحاجة إلى ية تتيح التشغيل الآلي لتطوير مشاريع تبيكو وتضمن سلامة الكود البرمجي

الكلمات المفتاحية : التشغيل الآلي، التكامل المستمر، تبيكو، جينكينز

Résumé

L'intégration continue ainsi que l'automatisation représentent des tâches indispensables et primordiales dans les secteurs de développement logiciel. L'équipe TIBCO reconnaît l'importance de ce processus qui se fait actuellement manuellement. Cela permet aux entreprises d'optimiser le rendement et de réduire les risques inhérents. D'où le besoin d'une solution qui permet l'automatisation de développement des projets TIBCO et assure la non-régression du code source.

Mots clés : Intégration continue, Automatisation, TIBCO, Jenkins

Abstract

Continuous integration and automation represent the primary and essential tasks in the software development sectors. The TIBCO team recognizes the importance of this process which is currently done manually. This allows companies to optimize performance and reduce inherent risks. Hence the need for a solution that automates the development of TIBCO projects and ensures the non-regression of the code source.

Key words : Continuous integration, Automation, TIBCO, Jenkins