

VERSION 1.7

SEPTEMBER 2, 2016

Projet MADERA

Projet Fil Rouge

Livrable 2



REDIGÉ PAR : CLEMENT SEBILLET

MAXIME LOUAZE

EDDY MASSON

SOMMAIRE

1.	Méthode de développement	4
1.1	Les trois piliers de scrum.....	4
1.2	Les caractéristiques de scrum	4
1.3	Les avantages de scrum	5
1.4	Les risques et inconvénients de scrum	6
1.5	Les bonnes pratiques a utiliser.....	6
1.5.1	Bonnes pratiques liées au développement.....	6
2.	Définition de l'environnement de développement	8
2.1	Environnement de développement	8
2.1.1	Application web	8
2.1.2	Application client mobile	8
2.2	Environnement de test.....	8
2.2.1	Application web	8
2.2.2	Application client mobile	8
3.	Argumentation des choix d'outils de developpement et de base de données associes	9
3.1	MySQL	9
3.1.1	Critères.....	9
3.1.2	Coûts	9
3.1.3	Ressources.....	9
3.2	Cloud9	9
3.2.1	Critères.....	9
3.2.2	Coûts	9
3.2.3	Ressources.....	10
3.3	LAMP	10
3.3.1	Critères.....	10
3.3.2	Coûts	10
3.3.3	Ressources.....	10
3.4	Node.js	10
3.4.1	Critères.....	10
3.4.2	Coûts	10
3.4.3	Ressources.....	10
3.5	React.js.....	11

3.5.1	Critères	11
3.5.2	Coûts	11
3.5.3	Ressources.....	11
3.6	Git (avec BITBUCKET)	11
3.6.1	Critères	11
3.6.2	Coûts	11
3.6.3	Ressources.....	11
3.7	Symfony.....	12
3.7.1	Critères	12
3.7.2	Coûts	12
3.7.3	Ressources.....	12
3.8	squash tm.....	12
3.8.1	Critères	12
3.8.2	Coûts	12
3.8.3	Ressources.....	12
3.9	phpDoc (avec phpdocumentor)	13
3.9.1	Critères	13
3.9.2	Coûts	13
3.9.3	Ressources.....	13
4.	modélisation UML et merise avec l'architecture de la base de données supportant l'application....	14
4.1	Modelisation Uml.....	14
4.2	modelisation merise.....	15
4.3	Structure de la base de données	16
4.4	Diagramme d'activité	17
5.	Description des modules de l'application et des traitements de l'informations associés.....	20
5.1	Diagramme de cas d'utilisation.....	20
5.2	Module « configuration ».....	21
5.2.1	Configuration des gammes	21
5.2.2	Configuration des modules	22
5.3	Module « conception des devis »	22
5.3.1	Identification du projet	22
5.3.2	Ressources produits	23
5.3.3	Modules à réaliser pour le projet.....	23

5.3.4	Edition du dossier technique et du devis	23
5.3	module « modalités de paiement »	24
5.3.1	Calcul de l'évolution des paiements échelonnés en fonction des lots à réaliser.....	24
5.3.2	Etat du devis réalisé	25
6.	Contrôles de saisies/données	26
6.1	Module « Gestion de stock »	26
6.1.1	Création gamme	26
6.1.2	Création composant	26
6.1.3	Création famille de composant	26
6.1.4	Création sous famille composant.....	26
6.1.5	Création fournisseur.....	26
6.2	Module « configuration ».....	26
6.2.1	Configuration des gammes	26
6.2.2	Edition de la gamme.....	27
6.2.3	Création module.....	27
6.2.4	Paramètre pourcentage	27
6.3	Module « Conception des devis »	27
6.3.1	Identification du projet ou chantier	27
6.3.2	Conception des produits	28
6.3.3	Création des modules à réaliser pour le projet.....	28
6.4	Module « modalités de paiements »	28
7.	Définition de l'ergonomie et de l'IHM (Mock-Up).....	29
8.	Préalables à la réalisation des séquences et les dépendances externes	30
9.	Glossaire.....	33
10.	Sources	37

1. METHODE DE DEVELOPPEMENT

Afin de développer de manière efficace et la plus rapide qui soit, nous avons décidé d'utiliser la méthode Scrum¹ pour organiser notre développement.

1.1 LES TROIS PILIERS DE SCRUM

Scrum est un processus empirique s'appuyant sur trois piliers étant :

- La transparence

Scrum met l'accent sur le fait d'avoir un langage commun entre l'équipe et le management. Ce langage commun doit permettre à tout observateur d'obtenir rapidement une bonne compréhension du projet.

- L'inspection

À intervalle régulier, Scrum propose de faire le point sur les différents artéfacts produits, afin de détecter toute variation indésirable.

- Adaptation

Si une dérive est constatée pendant l'inspection, le processus doit alors être adapté. Scrum fournit des outils durant lesquels cette adaptation est possible. Il s'agit de « la réunion de planification de sprint² », « la mêlée quotidienne³ », « la revue de sprint⁴ » ainsi que de la « rétrospective de sprint⁵ ».

1.2 LES CARACTERISTIQUES DE SCRUM

Cette méthode suit les principes des méthodes agiles⁶ et est composée de plusieurs caractéristiques adaptée à la typologie et la taille du projet que nous menons tel que :

- Le développement incrémental :

Le développement incrémental consiste à réaliser successivement des éléments fonctionnels utilisables, plutôt que des composants techniques.

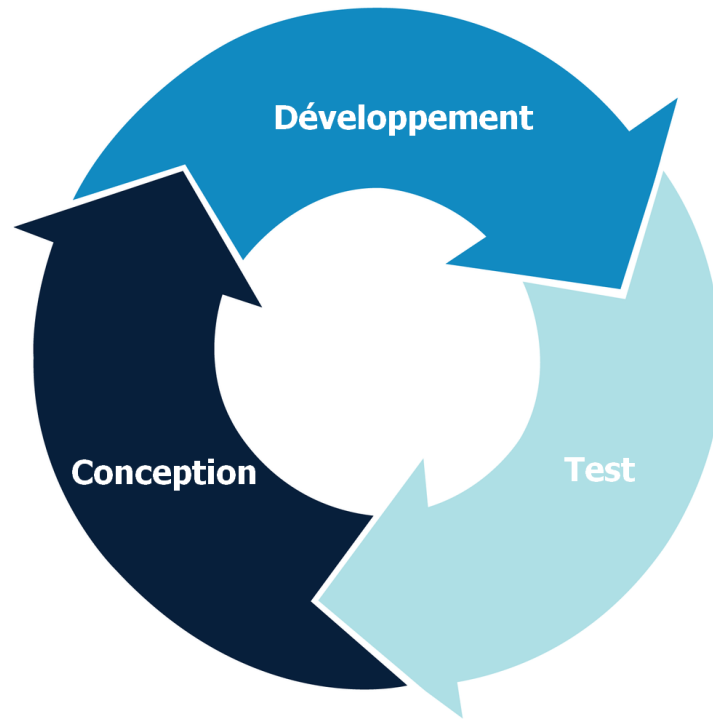
Un découpage en incréments est dit "vertical", en référence à l'imagerie habituelle qui présente les composants techniques d'une architecture logicielle comme les couches empilées d'un gâteau. Un incrément est une fonctionnalité complète, métaphoriquement une tranche verticale du gâteau.

Cela convient pour notre projet car nous avons deux développeurs qui pourront travailler en même temps sur le projet sans se gêner, en prenant chacun une part verticale de l'application à développer).

- Le développement itératif :

Une approche incrémentale implique nécessairement d'adopter également une approche itérative.

Ainsi, on considère comme itératif un cycle dans lequel on prévoirait, à l'issue d'une phase de spécifications et d'analyse, de répéter 3 fois une itération au cours de laquelle on réaliserait successivement la conception, le développement et le test. (Voir schéma ci-dessous)



De plus, la notion d'itération couvre l'adaptabilité au quotidien, ainsi avec ce type de méthode de développement nous pourrions assurer la continuité du service et des mises à jour de l'application qui seront demandées au fur et à mesure des changements du métier.

- La notion de processus empirique (imprévisible) :

Scrum et le développement empirique sont recommandés pour favoriser le développement de nouveaux logiciels ou des logiciels qui sont déjà orientés-objet ou qui ont des « interfaces propres ». Les objets et sous-systèmes de l'ouvrage qui sont cohérents et faiblement couplés avec les autres sont regroupés par paquets. Chaque équipe peut ainsi travailler sur un ou plusieurs paquets en toute indépendance.

1.3 LES AVANTAGES DE SCRUM

La méthode Scrum possède plusieurs avantages tel que :

- Chaque équipe a son lot de responsabilité
- Amélioration de la communication au sein de l'équipe projet
- Augmentation de la productivité
- Simplicité des processus
- La livraison d'un produit utilisable est plus rapide

1.4 LES RISQUES ET INCONVENIENTS DE SCRUM

Scrum possède néanmoins plusieurs inconvénients, en effet malgré son lot d'avantages non négligeable cette méthode n'est pas parfaite. Nous pouvons citer plusieurs inconvénients tel que :

- Peu, voire pas de documentation écrite (Comme la méthode privilégie le fonctionnel, nous pouvons rapidement nous retrouver avec un projet non-documentés)
- Les équipes doivent être dynamique et réactive
- Le « Flaccid scrum⁷ », ce terme est utilisé pour identifier une pratique erronée de scrum dans laquelle la qualité logicielle est négligée et le produit développé accumule de la dette technique⁸.

Afin de minimiser ces risques liés à l'utilisation de la méthode Scrum, nous allons utiliser un ensemble de bonnes pratiques que nous allons détailler ci-dessous.

1.5 LES BONNES PRATIQUES A UTILISER

Connaissant les risques de la méthode Scrum, nous allons utiliser un ensemble de bonnes pratiques afin de minimiser ces derniers. Ces bonnes pratiques proviendront soit de la norme NF logiciel (NF ISO/CEI 25051⁹) soit de notre propre expérience professionnelle. Nous utiliserons seulement celles qui nous semblent utiles et qui n'alourdiront pas trop les processus de notre projet.

1.5.1 Bonnes pratiques liées au développement

Nous avons regroupé ici toutes les bonnes pratiques liées au développement en lui-même et que nous allons utiliser.

1.5.1.1 Les standards de codage

Une règle de nommage dans la programmation informatique est un ensemble de règles destinées à choisir les identifiants logiciels (noms des éléments du programme) dans le code source et la documentation.

Nos standards de codage sont disponibles dans l'annexe « Standards de codage.pdf »

1.5.1.2 Gestion des tests fonctionnels

Afin de gérer correctement les tests d'acceptations et d'avoir un aperçu global de ce qui est fonctionnel à tout moment du projet, nous allons utiliser un gestionnaire de référentiel de test.

De plus, celui-ci nous permettra d'être certains de la qualité de l'application qui a été produite.

(Voir document « Workflow de test et création documentation utilisateur.pdf » en annexe)

1.5.1.3 Création de documentation utilisateur

A chaque fin de sprint, nous allons créer la documentation utilisateur liée à chaque fonctionnalité développée. (Voir document « Workflow de test et création documentation utilisateur.pdf » en annexe)

1.5.1.4 Création de documentation technique

A chaque fin de sprint, nous allons créer la documentation technique liée à chaque fonctionnalité développée. (Voir document « Workflow de test et création documentation utilisateur.pdf » en annexe)

1.5.1.5 Gestion du code source

Afin de gérer notre code source de manière effective et faciliter le travail collaboratif, nous allons utiliser un logiciel de gestion de versions décentralisé nommé Git¹⁰. Nous utiliserons donc Bitbucket qui est un dépôt distant Git.

2. DEFINITION DE L'ENVIRONNEMENT DE DEVELOPPEMENT

2.1 ENVIRONNEMENT DE DEVELOPPEMENT

2.1.1 Application web

Pour le développement de l'application web en PHP, nous avons choisi Cloud9¹¹ comme IDE¹², car il permet de développer en ligne facilement. Il permet d'installer un serveur LAMP¹³ rapidement, ce qui va nous permettre de faire fonctionner l'application PHP sur notre serveur en ligne, ainsi que d'accéder aux bases MySQL grâce à phpMyAdmin.

2.1.2 Application client mobile

Pour le développement de l'application client mobile web en html/js grâce au plugin React.js¹⁴, nous allons aussi utiliser Cloud9 comme IDE, car il permet d'installer facilement la plateforme logicielle libre Node.js¹⁵ indispensable à l'utilisation de React.js.

2.2 ENVIRONNEMENT DE TEST

Les développeurs utiliseront leurs environnements de développement en ligne Cloud9 afin de faire leurs tests durant leurs développements. Quand le développement d'une fonctionnalité est terminé, le code est envoyé sur le dépôt distant BitBucket afin d'être mis sur le serveur de test et être validé.

Nous allons avoir deux serveurs de test, un pour l'application web et un autre pour l'application client mobile. Nous aurons accès à ces serveurs à tout moment, ce qui n'est pas le cas des serveurs Cloud9 des développeurs qui ne sont disponible que lorsque les développeurs sont dessus en train de travailler.

2.2.1 Application web

Pour le serveur de test de l'application web, nous allons avoir une VPS¹⁶ avec une architecture LAMP. Ainsi nous aurons un serveur ayant la configuration de production pour faire les tests.

2.2.2 Application client mobile

Pour le serveur de test de l'application client mobile, nous allons avoir une VPS avec la plateforme logicielle Node.js d'installée. Ainsi nous aurons un serveur ayant la configuration de production pour faire les tests.

3. ARGUMENTATION DES CHOIX D'OUTILS DE DEVELOPPEMENT ET DE BASE DE DONNEES ASSOCIES

Note : Pour tous nos outils, nous avons décidé de choisir une version stable et non une version de développement.

3.1 MYSQL

Utilisation de MySQL version 5.7.9, système de gestion de base de données.

3.1.1 Critères

- Solution très courante en hébergement public
- Très bonne intégration dans l'environnement Apache/PHP
- Facilité de déploiement et de prise en main
- Plusieurs moteurs de stockage adapté aux différentes problématiques, configurable au niveau table
- Open Source
- Version cluster depuis la version 4
- Ordonnanceur dès la version 5.1
- Partitionnement dès la version 5.1

3.1.2 Coûts

- Aucun coût lié à MySQL car ce SGBDR est gratuit pour la réalisation d'application non commerciale

3.1.3 Ressources

- Hébergement en ligne via Cloud9 pour le développement
- Hébergement sur les VPS pour les serveurs de test

3.2 CLOUD9

Utilisation de Cloud9, IDE en ligne.

3.2.1 Critères

- IDE hébergé en ligne gratuitement disposant d'une communauté active et réactive
- Hébergeur fiable et disponible, utilisé par Harvard lors de leurs cours
- Permet l'installation rapide d'architecture tel que LAMP pour PHP ou la plateforme logicielle Node.js pour React.js
- 2 serveurs avec 512MB de mémoire et 2GB d'espace disque
- Bande passante des serveurs stable (latence faible)

3.2.2 Coûts

- Aucun coût lié à Cloud9 car cet IDE hébergé en ligne possède un plan gratuit permettant la création de plusieurs espaces de travail personnel

3.2.3 Ressources

- Aucune

3.3 LAMP

Utilisation de l'architecture LAMP avec les versions suivantes :

Linux : Debian 7

Apache : version 2.4.17

Mysql : version 5.7.9

PHP : version 7

3.3.1 Critères

- Libre
- Gratuite
- Robuste
- Incontournable en termes de développement d'applications Web
- Facilité de déploiement

3.3.2 Coûts

- Aucun coût car LAMP est un ensemble de logiciels libres.

3.3.3 Ressources

- Utilisation du serveur LAMP hébergé par Cloud9
- Utilisation de la VPS de test de l'application web

3.4 NODE.JS

Utilisation de Node.js version 4.5.0.

3.4.1 Critères

- Node.js est rapide
- C'est une technologie Open Source
- Productivité des développeurs augmentée
- Possède de multiples modules grâce à npm

3.4.2 Coûts

- Aucun coût car Node.js est une plateforme logicielle libre

3.4.3 Ressources

- Utilisation du serveur Node.js hébergé par Cloud9
- Utilisation de la VPS de test de l'application client mobile

3.5 REACT.JS

Utilisation de React.js version 15.3.1.

3.5.1 Critères

- Facilite la lisibilité du code
- Facilite la maintenabilité du code
- Facile à tester et de savoir comment vont être rendu les composants
- React est flexible
- Ce type de librairie est l'avenir du développement web

3.5.2 Coûts

- Aucun coût car React.js est une bibliothèque JavaScript libre

3.5.3 Ressources

- Utilisation du serveur Node.js hébergé par Cloud9
- Utilisation de la VPS de test de l'application client mobile

3.6 GIT (AVEC BITBUCKET)

Utilisation de Git version 2.9.3 avec le service web d'hébergement et de gestion de développement logiciel par Bitbucket.

3.6.1 Critères

- Gestion des branches
- Rapidité
- Possibilité de merge
- Travail en équipe
- Gestion des versions du code source
- BitBucket permet d'avoir une interface en ligne pour parcourir le code source présent sur le dépôt distant

3.6.2 Coûts

- Aucun coût car Git est distribué gratuitement et Bitbucket nous permet de créer des dépôts distants pour une équipe de développement de 5 personnes gratuitement

3.6.3 Ressources

- Installation de git sur les serveurs Cloud9 ainsi que sur les VPS de test

3.7 SYMFONY

Utilisation de Symfony version 2.8.9.

3.7.1 Critères

- Open Source
- Evolutivité et maintenance garantie
- Communauté active et support auprès de Sensiolabs
- Gain de temps et qualité du code optimisée
- Interopérabilité
- Réputation et références dans le monde auprès des professionnels

3.7.2 Coûts

- Aucun coût car Symfony 2 est un framework libre

3.7.3 Ressources

- Utilisation du serveur LAMP hébergé par Cloud9
- Utilisation de la VPS de test de l'application web

3.8 SQUASH TM

Utilisation de Squash TM version 1.14.1.

3.8.1 Critères

- Open Source
- Gestion du référentiel de test
- Gestion des exigences
- Gestion des cas de test
- Gestions des campagnes, exécution des tests, anomalies
- Edition de rapports
- Travail collaboratif
- Facile de prise en main

3.8.2 Coûts

- Aucun coût car Squash TM est un gestionnaire de référentiel de test Open Source gratuit

3.8.3 Ressources

- Utilisation de la VPS de test de ayant l'architecture LAMP installée pour installer Squash TM

3.9 PHPDOC (AVEC PHPDOCUMENTOR)

Utilisation de PhpDocumentor version 2.7.0.

3.9.1 Critères

- Logiciel libre
- Permet de créer la documentation technique PHP
- Génère la documentation au format HTML, PDF, CHM ou Docbook
- Standard formalisé pour commenter le code PHP

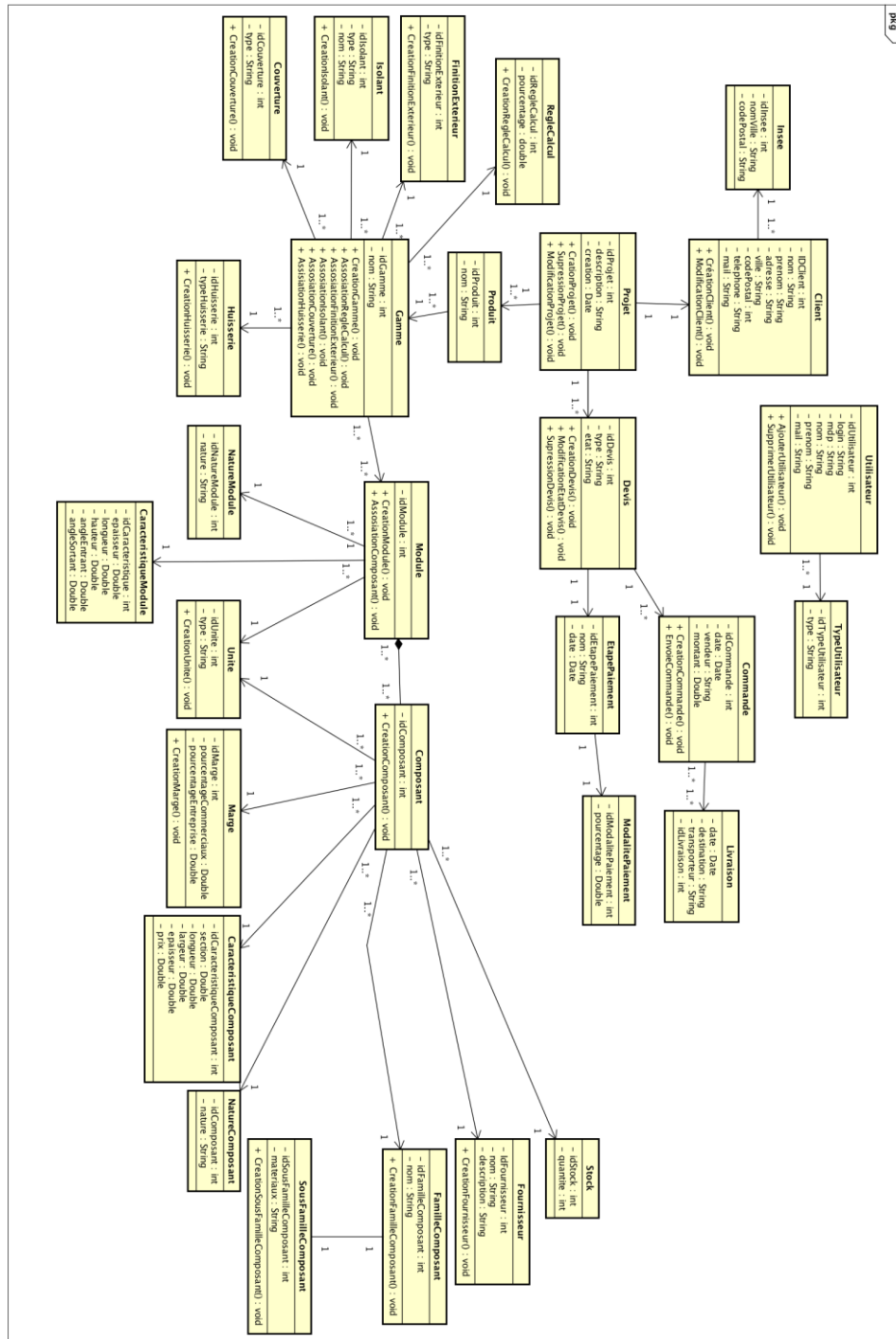
3.9.2 Coûts

- Aucun coût car PhpDocumentor est un logiciel libre gratuit

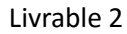
3.9.3 Ressources

- Installation de PhpDocumentor sur les serveurs Cloud9 des développeurs

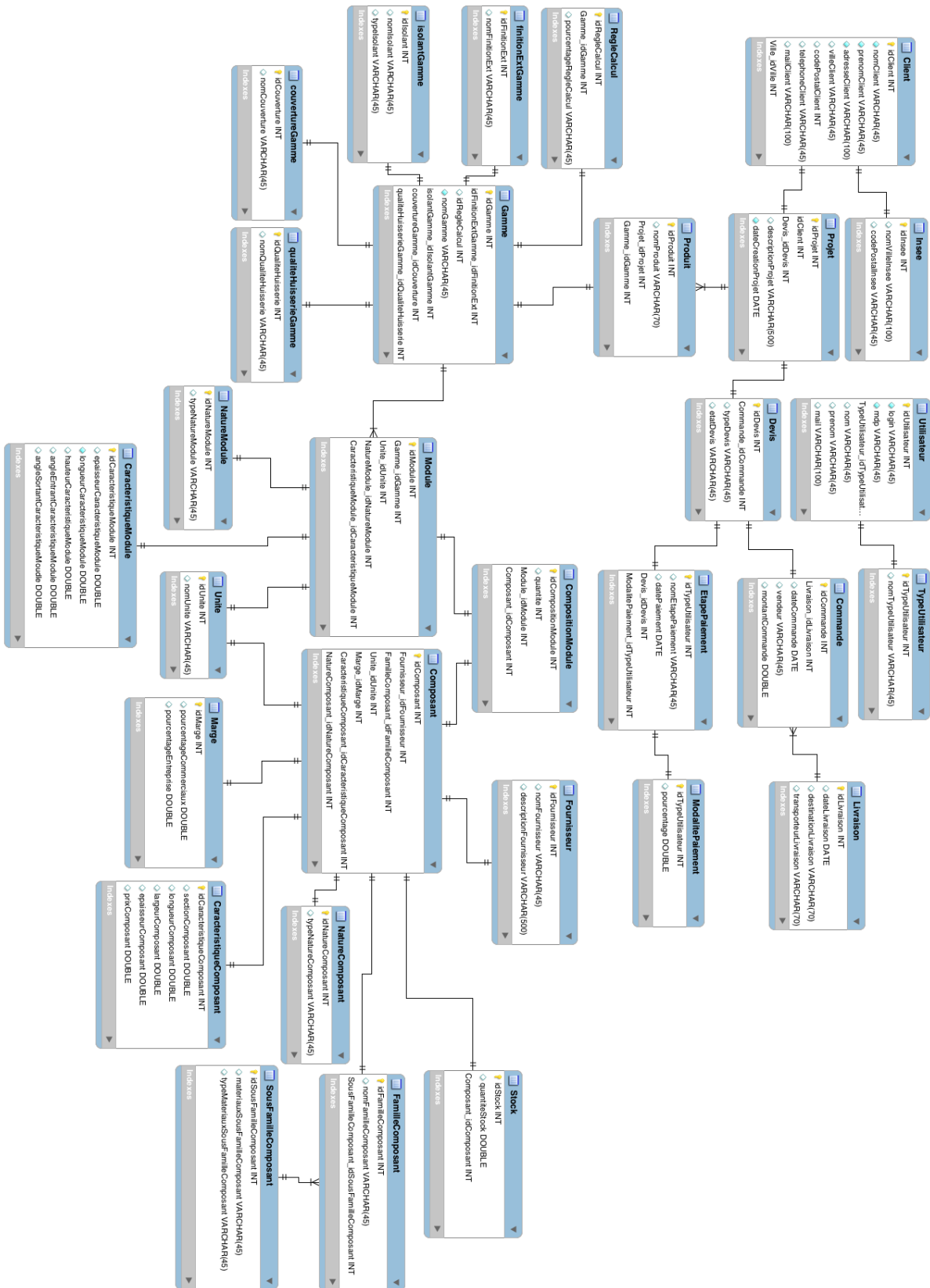
4.1 MODELISATION UML



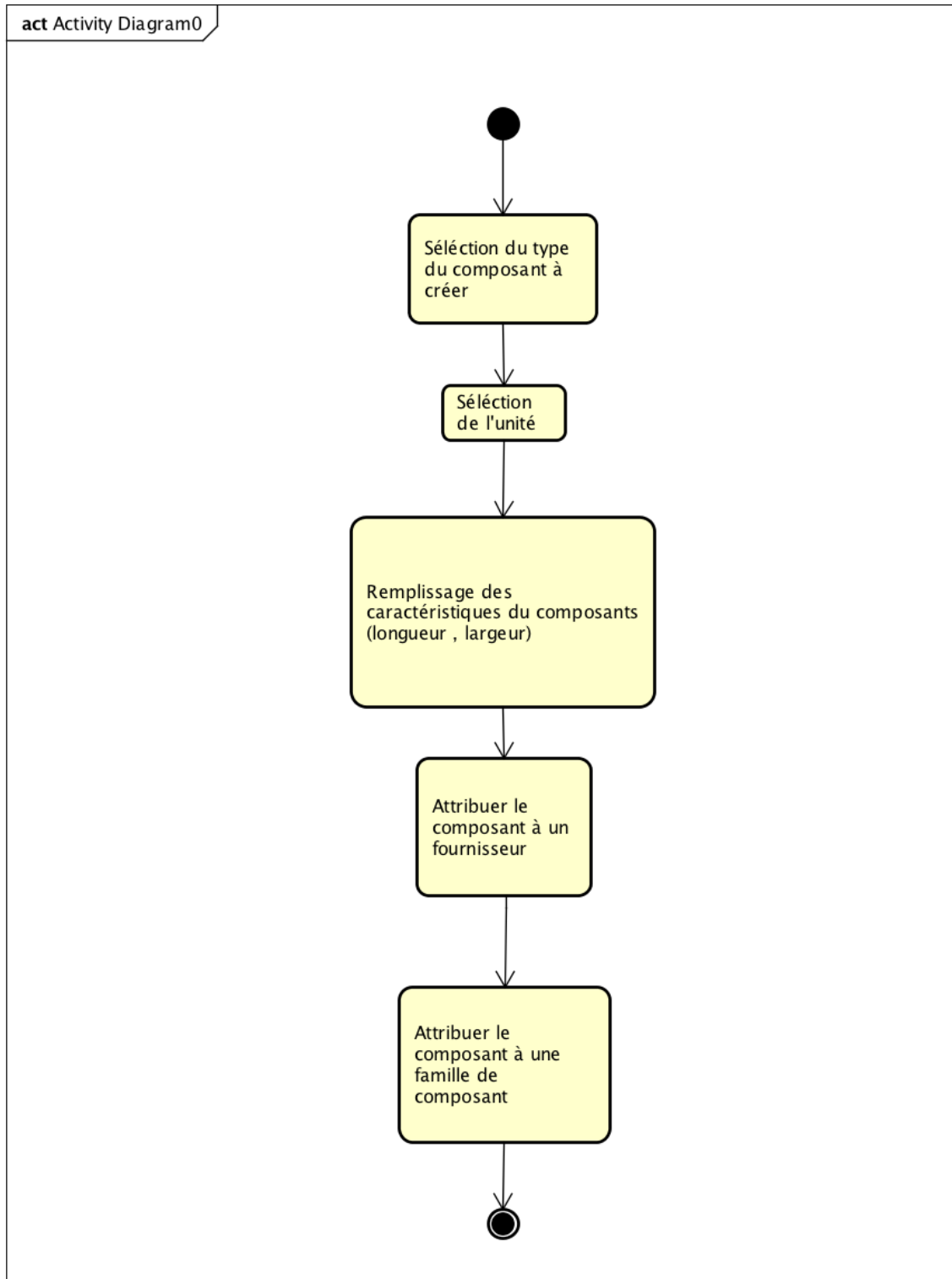
9/2/2016



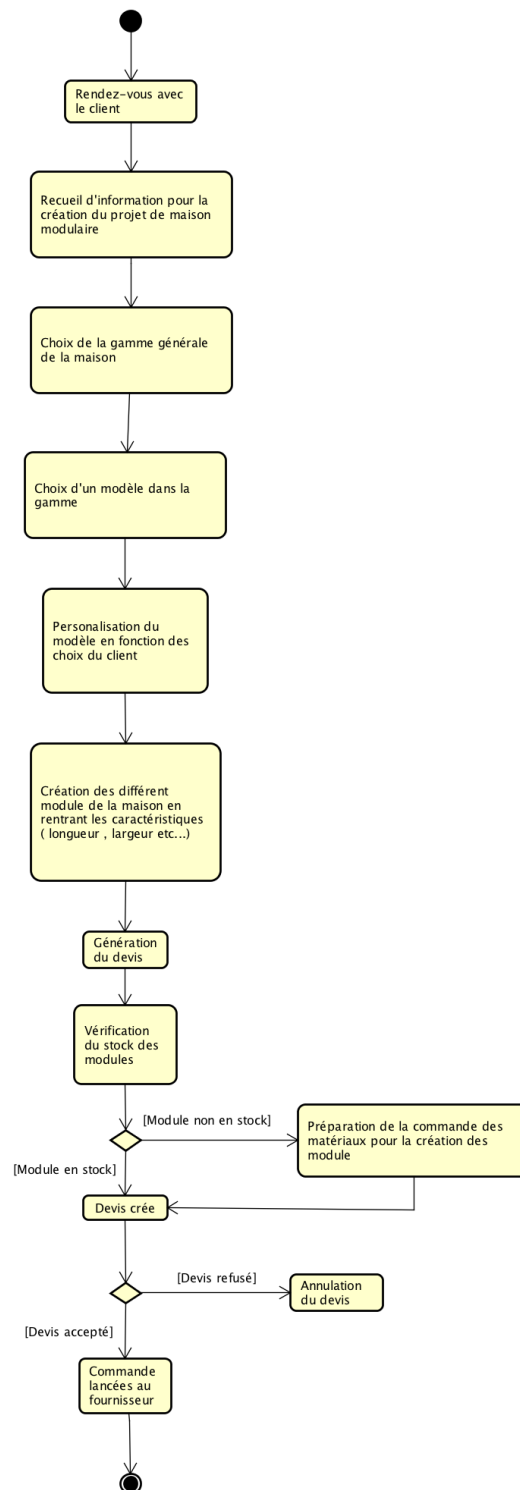
16

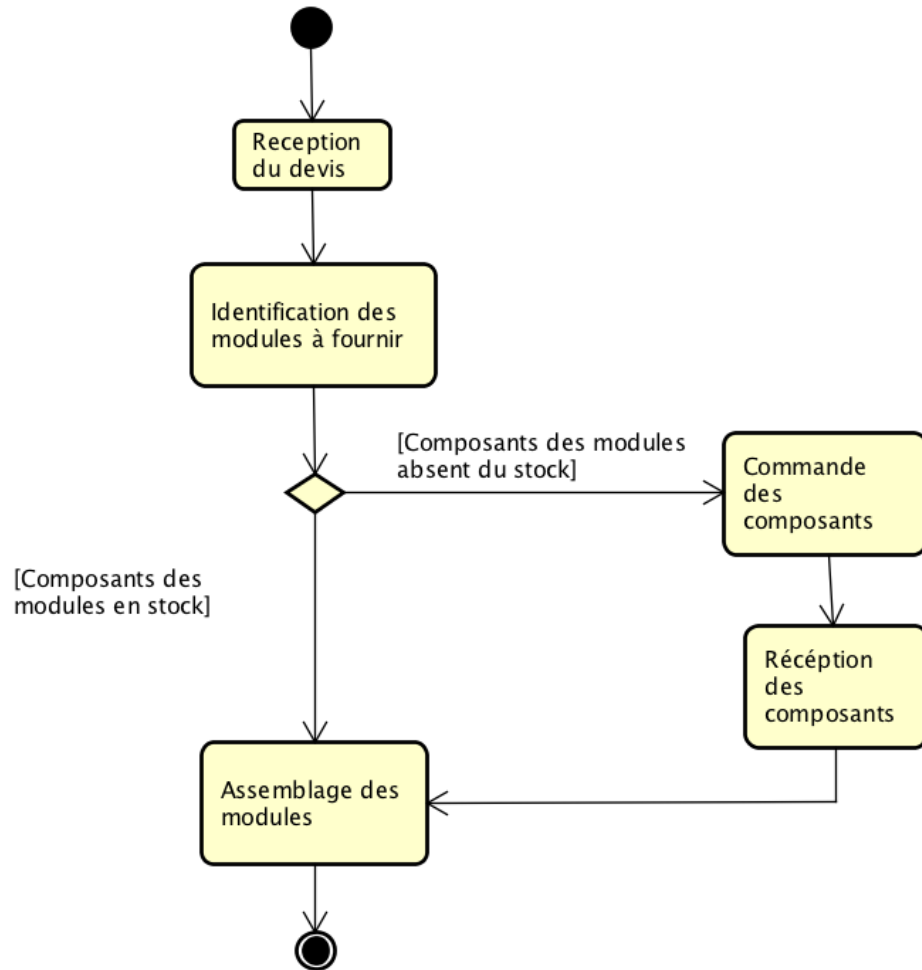


4.4 DIAGRAMME D'ACTIVITE



powered by Astah

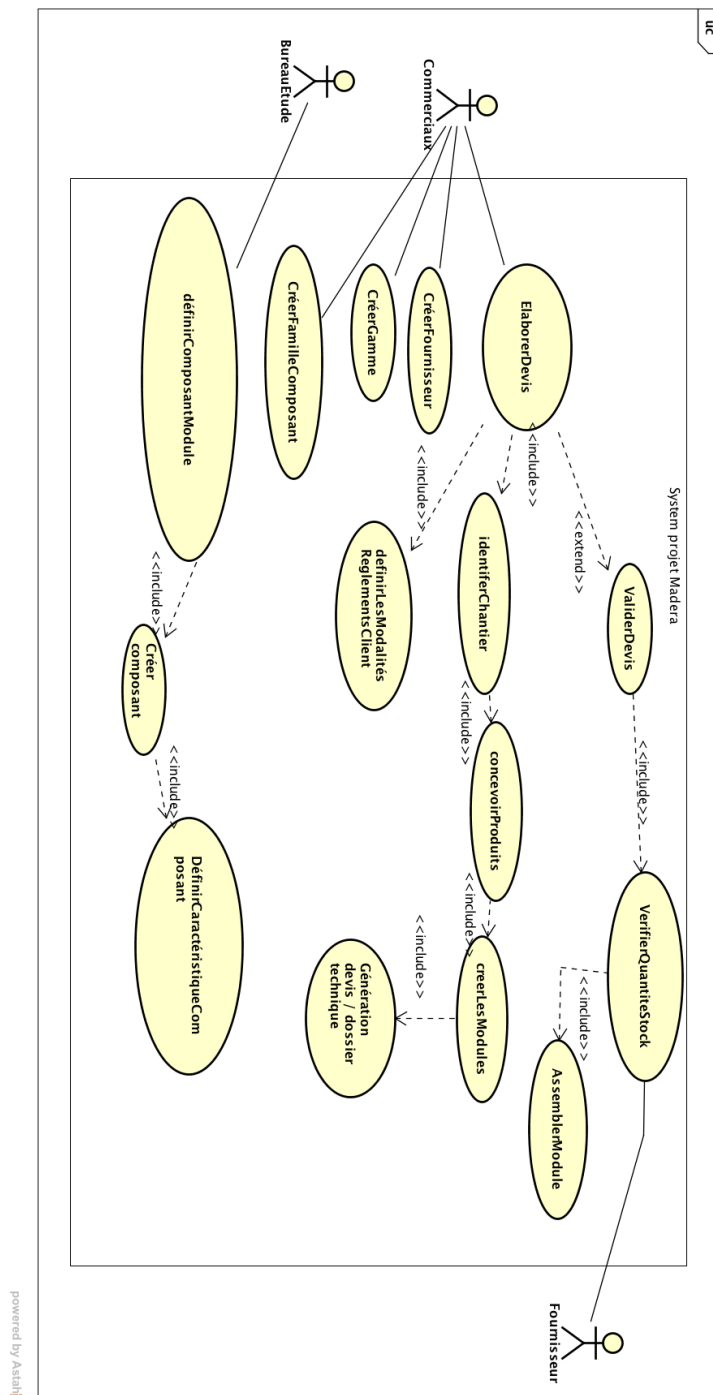




5. DESCRIPTION DES MODULES DE L'APPLICATION ET DES TRAITEMENTS DE L'INFORMATIONS ASSOCIES

5.1 DIAGRAMME DE CAS D'UTILISATION

Pour mieux comprendre le fonctionnement et les interactions entre les modules, voici un diagramme d'utilisation qui peut aider à la compréhension.



5.2 MODULE « CONFIGURATION »

En fonction des articles saisis dans la gestion de stock, ce module permet la conception des modules de chaque gamme et permet de faciliter la création de devis.

La gestion des stocks permettra la réalisation :

- Des Gammes : En référence à la table gamme de la base de données, la création d'une gamme comportera un identifiant (idGamme), une finition extérieur (idFinitionExtGamme), une règle de calcul (idRegleCalcul), un nom de gamme (nomGamme) et isolant (idIsolantGamme), une couverture (idCouvertureGamme) et la qualité des huisseries (idQualitéHuisserie)
- Des Familles de composants : En référence à la table FamilleComposant comportant un identifiant (idFamilleComposant) et un nom (nomFamilleComposant)
- Des fournisseurs : En lien avec la table fournisseur, composé d'un identifiant (idFournisseur) d'un nom (nomFournisseur) et d'une description (descriptionFournisseur)
- Des composants : En référence à la table composant, composé d'un identifiant (idComposé) d'une Famille de composant (idFamillecomposant) d'une nature de composant (natureComposant) d'une valeur de composant (valeurComposant) d'une hauteur et d'une largeur (hauteurComposant et largeurComposant), il est également lié à la table Module, Unité, Fournisseur et Famille composant, afin de récupérer l'idFournisseur, l'idModule, l'idUnite et l'idFamilleComposant

5.2.1 Configuration des gammes

Chaque maison modulaire est une gamme différente, elles varient en fonction de :

- La finition extérieure de la table finitionExtGamme, composé d'un identifiant (idFinitionExt) et d'un nom (nomFinitionExt)
- Le type d'isolant, en référence à la table IsolantGamme, composé d'un identifiant (idIsolant) et d'un nom (nomIsolant)
- Le type de couverture, en référence à la table couvertureGamme, composé d'un identifiant (idCouverture) et d'un nom (nomCouverture)
- La qualité des huisseries, en référence à la table qualiteHuisserieGamme, composé d'un identifiant(idQualitéHuisserie) et d'un nom (nomQualiteHuisserie)

La configuration des gammes permettra la création des règles de calcul de chaque gamme en fonction du mode de conception de l'ossature bois : cette règle fait référence à la table RegleCalcul composé d'un identifiant (idRegleCalcul) et lié à une gamme(idGamme).

Cette règle de calcul dépendra du mode de conception de l'ossature en bois en référence à la table Module (angleEntrant et angleSortant).

5.2.2 Configuration des modules

La conception des éléments de module s'effectuera en identifiant le module :

- Attribuer le module à une gamme (idGamme en clé étrangère dans la table Module) et en le nommant (natureModule)
- Sélectionner la coupe de principe défini par angleSortant et angleEntrant dans la table module
- Sélectionner les caractéristiques de la dalle de béton ou des plots de bétons, pour cela, chaque composant est lié à un module (idModule dans la table composant), la nature, la longueur, l'épaisseur et la section sont également définis dans la table composant

La conception des éléments du module s'effectuera ensuite en sélectionnant les composants du module et en précisant les quantités par unité de bases à savoir :

- Les Sections
- Montants
- Remplissage

Ces trois éléments sont des types de composants, en référence à la table composant, ils sont définis par un identifiant, une nature, une section, une longueur, une largeur et une épaisseur

La conception des éléments du module comporte également des Huisseries en fonction de la gamme (Table QualitéHuisserieGamme) et sont définis par un nom et un identifiant.

Les derniers éléments de la conception des éléments du module sont les paramètres qui permettent de définir le pourcentage appliqué au prix de base pour définir le prix de vente (le prix du composant est dans la table composant).

Le calcul se fait ensuite via la table Marge défini par un identifiant, une marge pour les commerciaux (pourcentageCommerciaux et pourcentageEntreprise).

5.3 MODULE « CONCEPTION DES DEVIS »

Ce module permet la réalisation de devis par les commerciaux en présence du client sur son ordinateur portable ou une tablette.

La réalisation de ses devis se base sur des modules et règles de calepinage définies tel que ci-dessous.

5.3.1 Identification du projet

Chaque projet ou chantier sont identifiés par les critères suivants :

- Le nom du projet défini par le commercial (En référence à descriptionProjet de la table Projet)
- La référence du projet, renseigné automatiquement (idprojet de la table projet)
- Les références clients (idclient de la table client)
- Date de réalisation du devis, incrémenté automatiquement lors de la création en référence à DateCreation de la table projet

5.3.2 Ressources produits

Le commercial sélectionné avec le client les différents composant, définis par :

- La gamme de produit (idgamme de la table Gamme)
- Le modèle de la gamme du produit (nomGamme de la table Gamme)

Le modèle est ensuite redéfini en fonction de la demande du client :

- Le Type de remplissage est redéfini en référence à natureComposant de la table Composant
- La finition intérieur et extérieur est modifiée, la finition intérieure est un composant (idComposant de la table Composant) la finition extérieure fait référence à (nomFinition de la table finitionExtGamme)

Si le modèle est modifié, la coupe de principe est mise à jour. (La coupe est un empilage de composant qui seront modifiés automatiquement dans la table Composant).

Le nombre de produit peut être modulable pour chaque projet.

5.3.3 Modules à réaliser pour le projet

Plusieurs étapes sont à réaliser pour chaque type de projet

- Identifier et nommer le module "mur nord -1" premier module à saisir à la création du projet
- Pour chaque section, entrer sa longueur et définir entre chaque section la présence d'un angle entrant ou sortant, associé à la table Composant (largeurComposant et longueurComposant de la table Composant)

5.3.4 Edition du dossier technique et du devis

5.3.4.1 Devis

Chaque devis devra être quantitatif et estimatif :

La première page comportera :

- La référence client, référence à idClient de la table Client
- La référence projet, référence à idProjet de la table Projet
- Le CCTP par produit (chaque composant associé à la création sera renseigné et détaillé)

Les unités, remontés automatiquement via le champ largeurComposant, longueurComposant, epaisseurComposant.

5.3.4.2 Dossier technique

Un état reprenant tous les composants générés

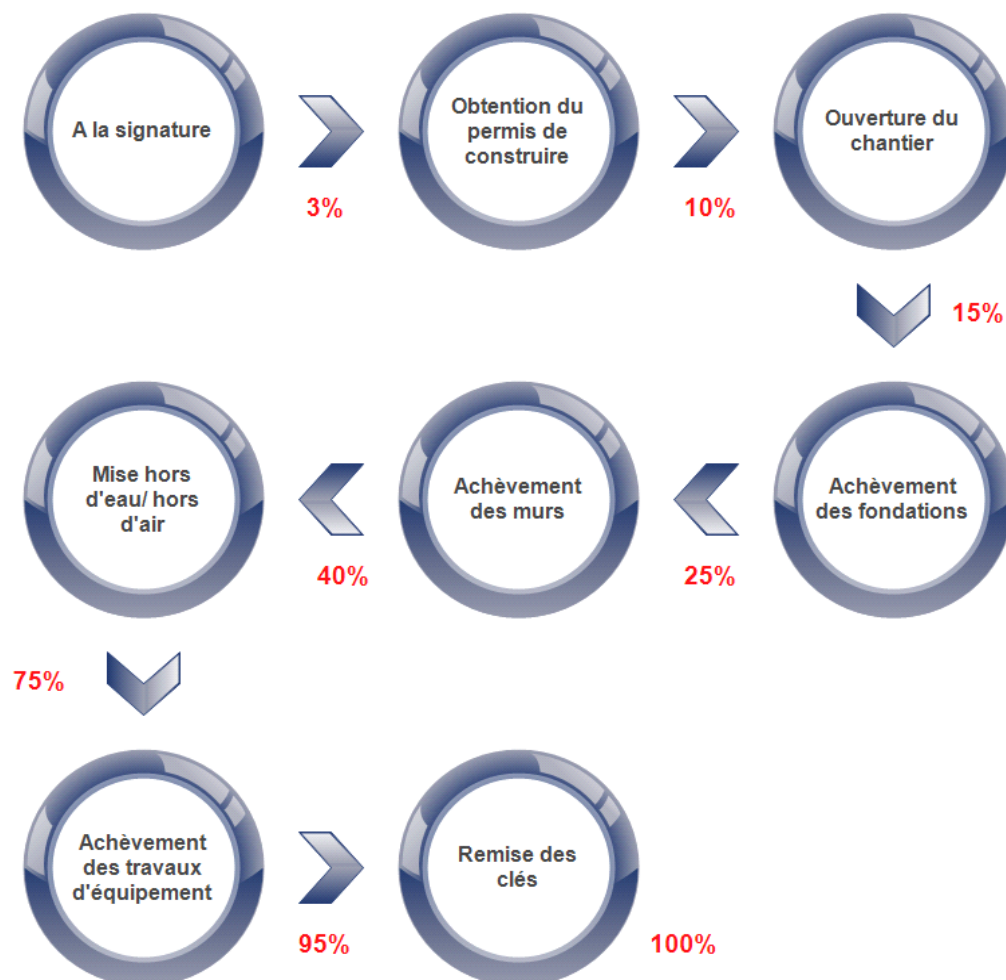
- Produits par produits
- Modules par modules

5.3 MODULE « MODALITES DE PAIEMENT »

Ce module permettra de calculer les échelonnements des paiements et suivre l'état d'avancement du devis, les calculs seront automatisés et calculés en fonction de montantCommande de la table Commande.

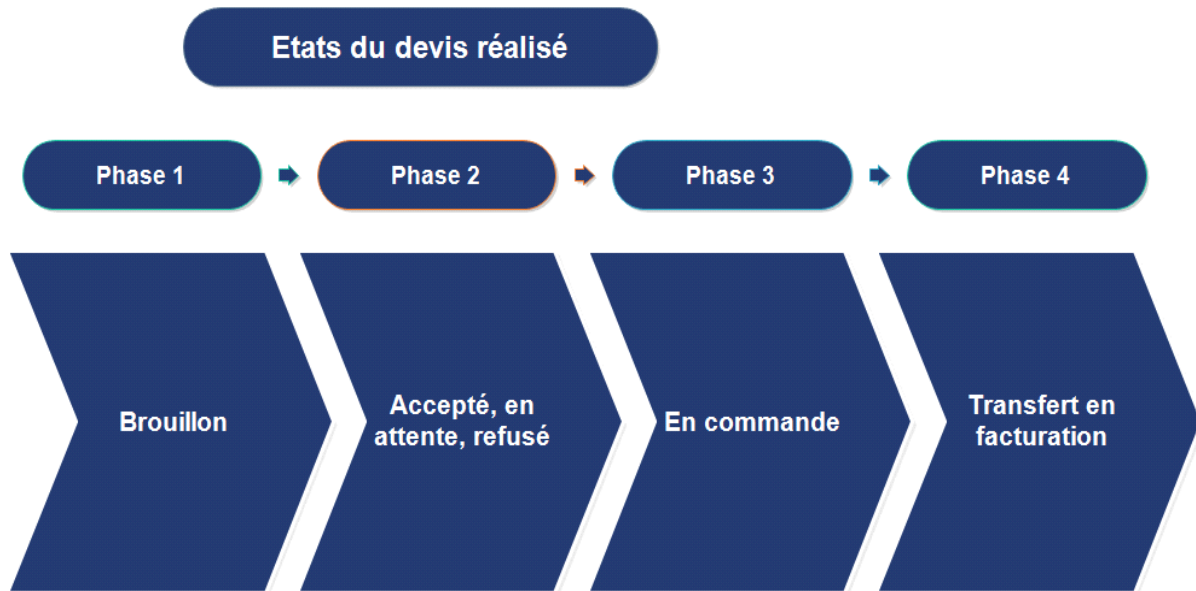
5.3.1 Calcul de l'évolution des paiements échelonnés en fonction des lots à réaliser

Les montants seront répartis comme suivant : Un pourcentage de la somme est à débloquent à chaque étape comme suivant :



5.3.2 Etat du devis réalisé

Le devis passe par différentes phases avant d'être validé, à savoir :



L'Etat du devis peut être modifié à tout moment, pour cela, on met à jour le champ `etatDevis` de la table `Devis`.

6. CONTROLES DE SAISIES/DONNEES

6.1 MODULE « GESTION DE STOCK »

6.1.1 Création gamme

- Nom de la gamme : Zone d'édition, caractères de a à z, chiffres de 0 à 9
- Nom finition : Zone d'édition, caractères de a à z, chiffres de 0 à 9
- Nom isolant : Zone d'édition, caractères de a à z, chiffres de 0 à 9
- Nom couverture : Zone d'édition, caractères de a à z, chiffres de 0 à 9
- Type ossature : Zone d'édition, caractères de a à z, chiffres de 0 à 9
- Règle de calcul : Zone d'édition, caractères de a à z, chiffres de 0 à 9
- Type huisserie : Zone d'édition, caractères de a à z, chiffres de 0 à 9

6.1.2 Création composant

- Nature composant : Menu déroulant, caractères de a à z, chiffres de 0 à 9
- Section composant : Zone d'édition, chiffres de 0 à 9
- Longueur : Zone d'édition, chiffres de 0 à 9
- Largeur : Zone d'édition, chiffres de 0 à 9
- Epaisseur : Zone d'édition, chiffres de 0 à 9
- Unité de mesure : Menu déroulant, caractères de a à z

6.1.3 Création famille de composant

- Nom : Zone d'édition, caractères de a à z, chiffres de 0 à 9

6.1.4 Création sous famille composant

- Type Matériaux : Zone d'édition, caractères de a à z, chiffres de 0 à 9
- Matériaux : Zone d'édition, caractères de a à z, chiffres de 0 à 9

6.1.5 Création fournisseur

- Nom : Zone d'édition, caractères de a à z, chiffres de 0 à 9
- Description : Zone d'édition, caractères de a à z, chiffres de 0 à 9

6.2 MODULE « CONFIGURATION »

6.2.1 Configuration des gammes

- Nom de la gamme à éditer : Menu déroulant, caractères de a à z, chiffres de 0 à 9

6.2.2 Edition de la gamme

- Nom de la gamme : Zone d'édition, caractères de a à z, chiffres de 0 à 9
- Nom finition : Zone d'édition, caractères de a à z, chiffres de 0 à 9
- Nom isolant : Zone d'édition, caractères de a à z, chiffres de 0 à 9
- Nom couverture : Zone d'édition, caractères de a à z, chiffres de 0 à 9
- Type ossature : Zone d'édition, caractères de a à z, chiffres de 0 à 9
- Règle de calcul : Zone d'édition, caractères de a à z, chiffres de 0 à 9
- Type huisserie : Zone d'édition, caractères de a à z, chiffres de 0 à 9

6.2.3 Création module

- Type : Menu déroulant, caractères de a à z
- Nom : Zone d'édition, caractères de a à z, chiffres de 0 à 9
- Gamme : Menu déroulant, caractères de a à z
- Coupe de principe : (Image avec les différentes parties du module à configurer), choix des matériaux via un menu déroulant, caractères de a à z
- Épaisseur CCTP : (si disponible en fonction du type), Zone d'édition, chiffres de 0 à 9
- Unité de mesure CCTP : Menu déroulant, caractères de a à z
- Composant du module :
 - Caractéristique des composants : Menu déroulant, caractères de a à z, chiffres de 0 à 9
 - Quantité des composants en fonction du module, zone d'édition, chiffres de 0 à 9

6.2.4 Paramètre pourcentage

- Marge entreprise : Zone d'édition, chiffres de 0 à 9
- Marge commerciaux : Zone d'édition, chiffres de 0 à 9

6.3 MODULE « CONCEPTION DES DEVIS »

6.3.1 Identification du projet ou chantier

- Nom du projet : Zone d'édition, caractères de a à z, chiffres de 0 à 9
- Référence projet : Génération automatique dans la base
- Référence client :
 - Nom : Zone d'édition, caractères de a à z
 - Prénom : Zone d'édition, caractères de a à z
 - Adresse : Zone d'édition, caractères de a à z, chiffres de 0 à 9
 - Ville : Zone d'édition, caractères de a à z
 - Code postale : Zone d'édition, chiffres de 0 à 9
 - Téléphone Zone d'édition, chiffres de 0 à 9
 - Mail : Zone d'édition, caractères de a à z, chiffres de 0 à 9, caractère spéciaux
- Date : Généré automatiquement

6.3.2 Conception des produits

- Sélection gamme : Menu déroulant, caractères de a à z
 - Sélection modèle de la gamme : Menu déroulant, caractères de a à z
 - Mise à jour du modèle si besoin :
 - Type remplissage : Menu déroulant, caractères de a à z
 - Finition intérieure : Menu déroulant, caractères de a à z
 - Finition Extérieur : Menu déroulant, caractères de a à z
-
- Modification de la coupe de principe :
 - Sélection des différents matériaux de la coupe de principe : Menu déroulant

6.3.3 Création des modules à réaliser pour le projet

- Type module : Menu déroulant, caractères de a à z
- Nom module : Zone d'édition, caractères de a à z
- Hauteur : Zone d'édition, chiffres de 0 à 9
- Longueur : Zone d'édition, chiffres de 0 à 9
- Angle (si besoin) : Zone d'édition, chiffres de 0 à 9

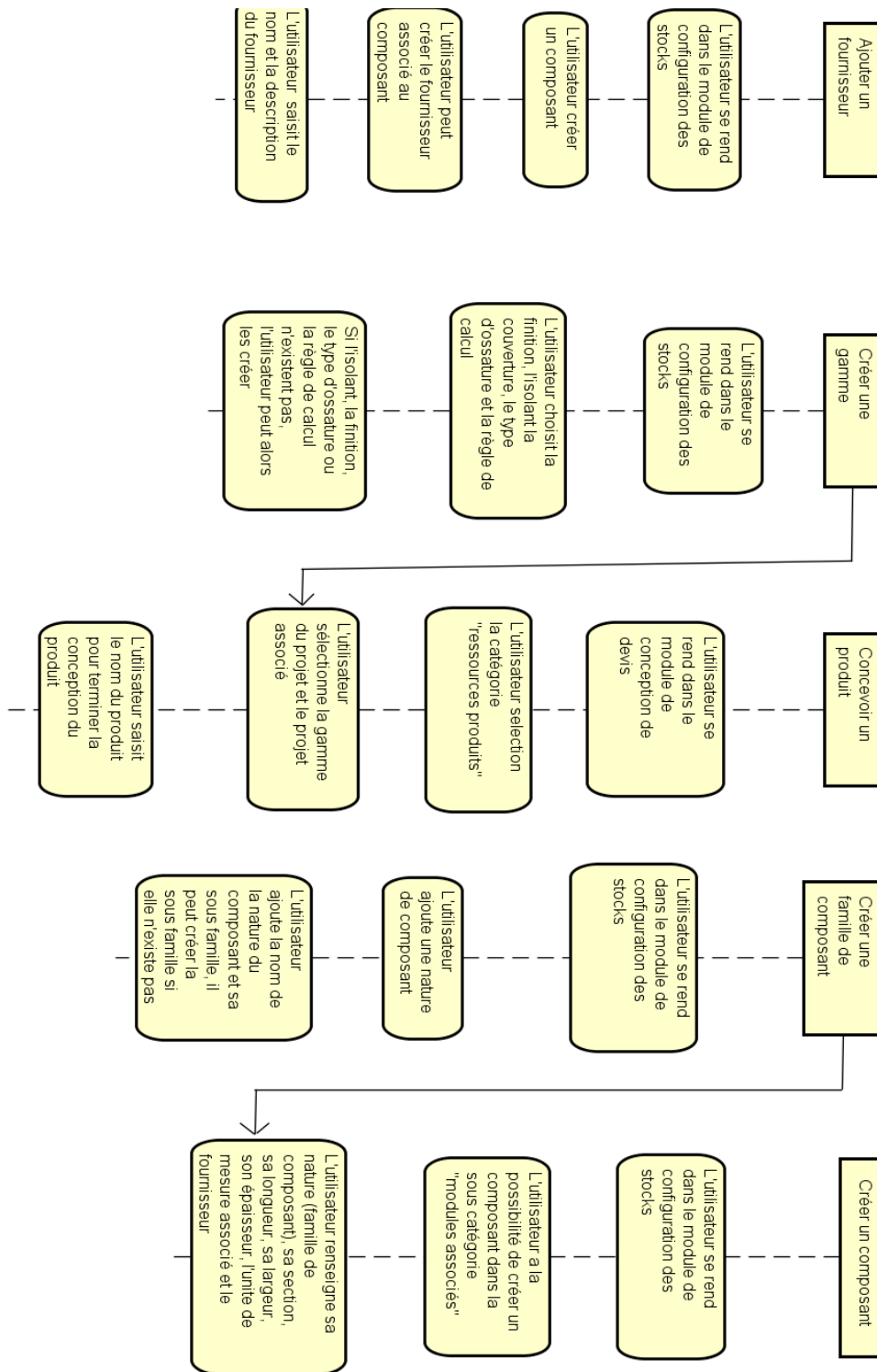
6.4 MODULE « MODALITES DE PAIEMENTS »

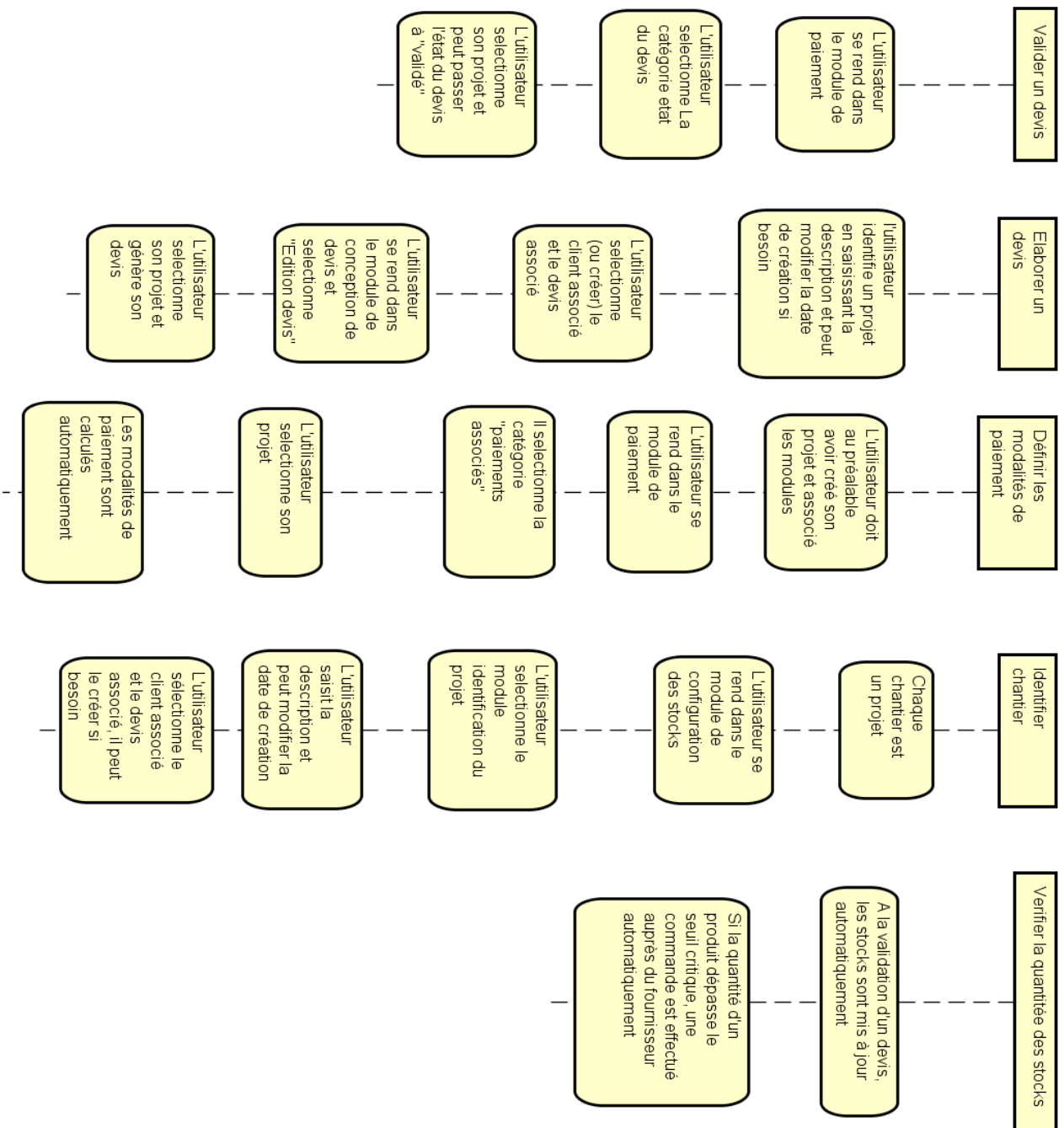
- Etat du devis : Menu déroulant, caractères de a à z

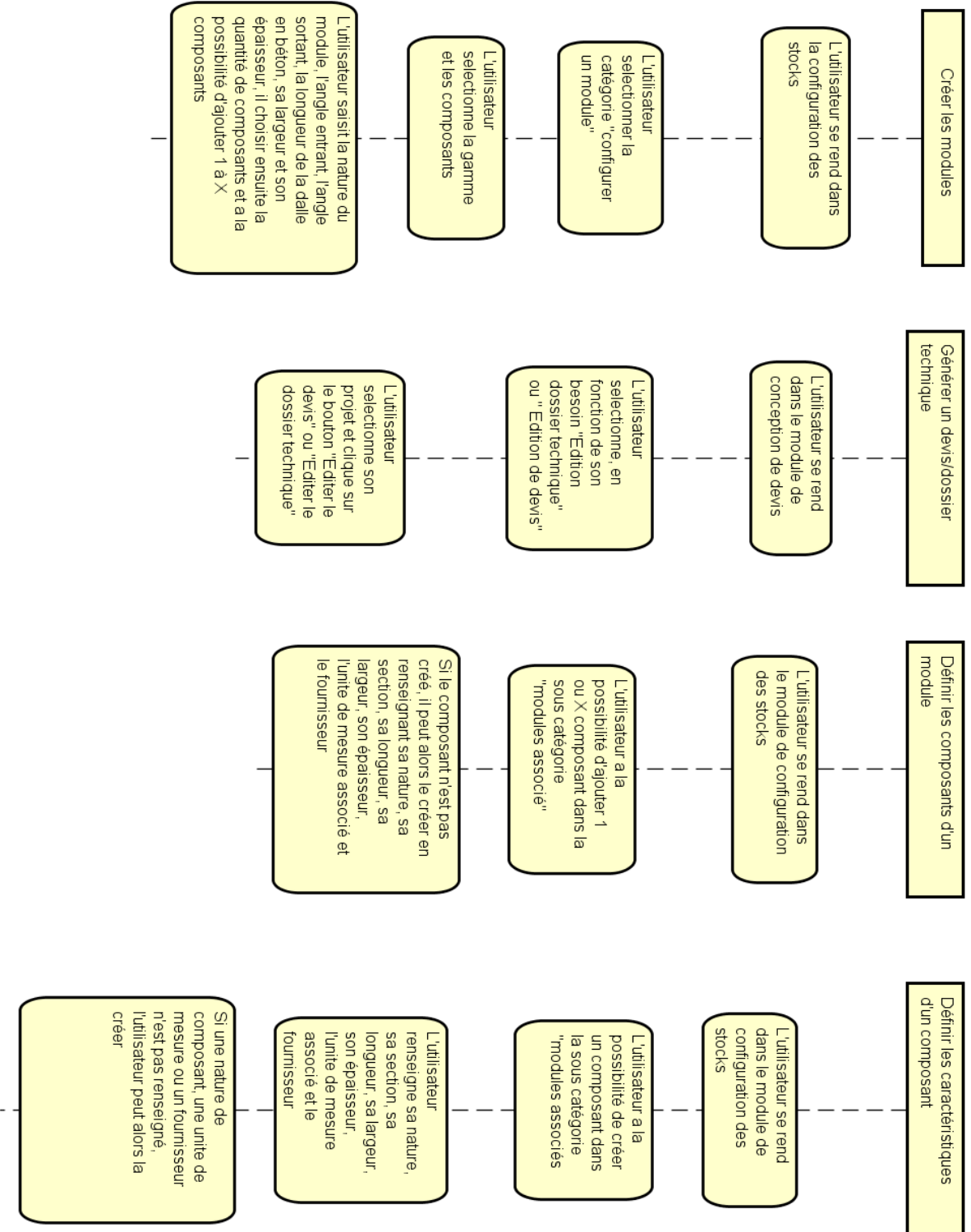
7. DEFINITION DE L'ERGONOMIE ET DE L'IHM (MOCK-UP)

Pour la définition de l'ergonomie et de l'interface homme-machine, voir annexe « Définition de l'ergonomie et de l'IHM (Mock-Up).pdf ».

8. PREALABLES A LA REALISATION DES SEQUENCES ET LES DEPENDANCES EXTERNES







9. GLOSSAIRE

9.2.1 Scrum

Définition : Scrum est un schéma d'organisation de développement de produits complexes. Il est défini par ses créateurs comme un « cadre de travail permettant de répondre à des problèmes complexes et changeants, tout en livrant de manière productive et créative des produits de la plus grande valeur possible ». Scrum est considéré comme une méthode agile.

La méthode s'appuie sur le découpage d'un projet en boîtes de temps, nommées « sprints ». Les sprints peuvent durer entre quelques heures et un mois (avec une préférence pour deux semaines). Chaque sprint commence par une estimation suivie d'une planification opérationnelle. Le sprint se termine par une démonstration de ce qui a été achevé. Avant de démarrer un nouveau sprint, l'équipe réalise une rétrospective. Cette technique analyse le déroulement du sprint achevé, afin d'améliorer ses pratiques. L'adaptation et la réactivité de l'équipe de développement est facilitée par son auto-organisation.

9.2.2 La réunion de planification de sprint

Définition : Toute l'équipe scrum est présente à cette réunion, qui ne doit pas durer plus de 8 heures pour un sprint d'un mois. Pour un sprint plus court, la durée est réduite proportionnellement. À l'issue de cette réunion, l'équipe a décidé des éléments du carnet du produit qu'elle traitera dans le cadre de la prochaine itération, et comment elle s'organisera pour y parvenir.

9.2.3 La mêlée quotidienne

Définition : La mêlée quotidienne (Daily scrum) est une réunion de planification « juste à temps » et permet aux développeurs de faire un point de coordination sur les tâches en cours et sur les difficultés rencontrées. Cette réunion dure 15 minutes au maximum. Le scrum master s'assure que la réunion ait lieu à heure fixe.

À tour de rôle, chaque membre aborde trois sujets :

- Ce qu'il a réalisé la veille,
- Ce qu'il compte réaliser aujourd'hui pour atteindre l'objet du sprint,
- Les obstacles qui empêchent l'équipe d'atteindre le but du sprint.

9.2.4 La revue de sprint

À la fin du sprint, l'équipe scrum et les parties-prenantes invitées se réunissent pour effectuer la revue de sprint, qui dure au maximum quatre heures. L'objectif de la revue de sprint est de valider l'incrément de produit qui a été réalisé pendant le sprint. L'équipe énonce les éléments du carnet de produit sélectionnés en début de sprint. L'équipe présente les éléments finis (complètement réalisés). Les éléments non finis (partiellement réalisés) ne sont pas présentés.

Une fois le bilan du sprint réalisé, l'équipe de développement et le propriétaire du produit mettent à jour le carnet du produit en fonction de ce qui a été réalisé (fini). Ils discutent avec les parties-prenantes de l'état courant du projet (budget, financement, conditions du marché), pour ajuster les éléments de carnet de produit et la planification selon les opportunités découvertes.

9.2.5 La rétrospective du sprint

La rétrospective du sprint est faite en interne à l'équipe scrum (équipe de réalisation, propriétaire du produit et scrum master). Elle dure trois heures pour un sprint d'un mois, et réduit selon la durée du

sprint. Elle a pour but l'adaptation aux changements qui surviennent au cours du projet et l'amélioration continue du processus de réalisation.

L'objectif est d'inspecter l'itération précédente, afin de déterminer quels sont les éléments du processus de développement qui ont bien fonctionné et ceux qui sont à améliorer. L'équipe de développement déduit un plan d'actions d'amélioration qu'elle mettra en place lors de l'itération suivante.

9.2.6 Les méthodes agiles

Les méthodes agiles sont des groupes de pratiques de pilotage et de réalisation de projets. Initialement destinées au développement en informatique (conception de logiciel), leur champ d'application s'élargit régulièrement. Elles ont pour origine l'Agile manifesto. Rédigé en 2001, celui-ci consacre le terme d'« agile » pour référencer de multiples méthodes existantes. Les méthodes agiles se veulent plus pragmatiques que les méthodes traditionnelles. Elles impliquent au maximum le demandeur (client) et permettent une grande réactivité à ses demandes. Elles visent la satisfaction réelle du client en priorité aux termes d'un contrat de développement.

Les méthodes agiles reposent sur une structure (cycle de développement) itérative, incrémentale et adaptative. Elles doivent respecter quatre valeurs fondamentales déclinées en douze principes desquels découlent une base de pratiques, soit communes, soit complémentaires.

9.2.7 Le flaccid scrum

Le terme flaccid scrum est utilisé par Martin Fowler en 2009 pour identifier une pratique erronée de scrum dans laquelle la qualité logicielle est négligée et le produit développé accumule de la dette technique¹⁶. Fowler indique que même si scrum ne décrit pas quelles pratiques techniques mettre en place, la méthode n'encourage pas l'absence de telles pratiques et que les scrummers de premier plan ont toujours insisté sur l'utilisation de pratiques techniques efficaces. La négligence de l'aspect technique de l'activité de développement et l'augmentation de la dette technique qui en résulte sont des signes d'une mauvaise utilisation de scrum.

Fowler mentionne que les pratiques techniques de l'extreme programming constituent un bon point de départ pour les équipes scrum et valent mieux que l'absence de pratiques.

En 2014 Fowler considère que le problème persiste et encourage les utilisateurs de scrum et des méthodes agiles en général à prendre garde à l'utilisation de pratiques techniques efficaces.

9.2.8 Dette technique

La dette technique est une métaphore du développement logiciel inventée par Ward Cunningham.

Il s'inspire du concept existant de dette dans le contexte du financement des entreprises et l'applique au domaine du développement logiciel.

Une dette technique peut être intentionnelle ou pas.

Une dette technique non intentionnelle est due à des malfaçons : non-respect de la conception, non-respect des règles de codage, etc. C'est une mauvaise dette car il n'y a aucun bénéfice à retirer de cette dette.

Une dette technique peut être contractée de manière intentionnelle. Dans un projet, la qualité augmente la charge de travail, ce qui peut avoir un impact sur le délai immédiat. Ainsi, lors de la survenue imminente d'une nouvelle version du logiciel, respecter la conception idéale peut mettre en péril la livraison d'une nouvelle version du logiciel. À ce moment précis, ne pas respecter la conception idéale peut permettre d'atteindre l'objectif prioritaire à court terme (sortir une nouvelle version). C'est

une dette intentionnelle car on sacrifie la qualité à long terme pour le bien du projet. Il est alors conseillé de rembourser cette dette immédiatement après la livraison de la nouvelle version du logiciel pour qu'elle ne devienne pas une dette à long terme. Ainsi on contracte une dette à court terme pour en retirer un bénéfice immédiat.

9.2.9 NF ISO/CEI 25051

Norme NF définissant les exigences qualité et documentation de tests : des évolutions dans la prise en compte d'exigence de confort en relation avec les travaux de normalisation (ISO25051)

Libellé de la norme : « Ingénierie du logiciel - Exigences de qualité pour le logiciel et évaluation (SQuaRE) - Exigences de qualité pour les logiciels et instructions d'essai »

9.2.10 Git

Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2. En 2016, il s'agit du logiciel de gestion de versions le plus populaire qui est utilisé par plus de douze millions de personnes.

9.2.11 Cloud9

Cloud9 IDE est un environnement de développement en ligne open source à partir de la version 3.0. Il supporte plusieurs centaines de langages de programmation, dont PHP, Ruby, Perl, Python, JavaScript avec Node.js. Cela permet aux développeurs de commencer à coder immédiatement avec un espace de travail préconfiguré. Il permet de travailler en équipe, avoir une prévisualisation de son développement et de tester la compatibilité avec les navigateurs.

9.2.12 IDE

En programmation informatique, un environnement de développement est un ensemble d'outils pour augmenter la productivité des programmeurs qui développent des logiciels¹. Il comporte un éditeur de texte destiné à la programmation, des fonctions qui permettent, par pression sur un bouton, de démarrer le compilateur ou l'éditeur de liens ainsi qu'un débogueur en ligne, qui permet d'exécuter ligne par ligne le programme en cours de construction. Certains environnements sont dédiés à un langage de programmation en particulier

9.2.13 LAMP

LAMP est un acronyme désignant un ensemble de logiciels libres permettant de construire des serveurs de sites web. L'acronyme original se réfère aux logiciels suivants :

« Linux », le système d'exploitation (GNU/Linux) ;

« Apache », le serveur Web ;

« MySQL ou MariaDB », le serveur de base de données ;

À l'origine, « PHP », « Perl » ou « Python », les langages de script.

Même si les auteurs de chacun de ces programmes ne se sont pas coordonnés pour construire des plates-formes LAMP, cette combinaison de logiciels s'est popularisée du fait du faible coût de l'ensemble et de la présence de tous ces composants dans la plupart des distributions GNU/Linux.

9.2.14 React.js

React (aussi appelé React.js ou ReactJS) est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page (ou portion) HTML à chaque changement d'état.

React est une bibliothèque qui ne gère que l'interface de l'application, considéré comme la vue dans le modèle MVC. Elle peut ainsi être utilisée avec une autre bibliothèque ou un Framework MVC comme AngularJS. La bibliothèque se démarque de ses concurrents par sa flexibilité et ses performances, en travaillant avec un DOM virtuel et en ne mettant à jour le rendu dans le navigateur qu'en cas de nécessité.

La bibliothèque est utilisée notamment par Netflix, Yahoo, Airbnb, Sony, Atlassian ainsi que par les équipes de Facebook, appliquant le dogfooding sur le réseau social éponyme, Instagram ou encore WhatsApp.

9.2.15 Node.js

Node.js est une plateforme logicielle libre et événementielle en JavaScript orientée vers les applications réseau qui doivent pouvoir monter en charge. Elle utilise la machine virtuelle V8 et implémente sous licence MIT les spécifications CommonJS. Node.js contient une bibliothèque de serveur HTTP intégrée, ce qui rend possible de faire tourner un serveur web sans avoir besoin d'un logiciel externe comme Apache ou lighttpd, et permettant de mieux contrôler la façon dont le serveur web fonctionne.

Concrètement, node.js est un environnement d'assez bas niveau permettant d'exécuter du JavaScript non plus dans le navigateur web mais sur le serveur.

Node.js est de plus en plus populaire comme plateforme serveur, elle est utilisée par Groupon, SAP, LinkedIn, Microsoft, Yahoo!, Walmart, Rakuten et PayPal.

9.2.16 VPS

Un serveur dédié virtuel (également appelé serveur virtuel), en anglais virtual private server (VPS) ou virtual dedicated server (VDS) est une méthode de partitionnement d'un serveur en plusieurs serveurs virtuels indépendants qui ont chacun les caractéristiques d'un serveur dédié, en utilisant des techniques de virtualisation. Chaque serveur peut fonctionner avec un système d'exploitation différent et redémarrer indépendamment.

Dans le domaine de l'hébergement web, plusieurs dénominations recoupent le même type d'offres et donc de services. Les acronymes VPS (Virtual Private Server) et VDS (Virtual Dedicated Server) désignent le même concept, et leur usage est parfois confus.

10. SOURCES

Wikipedia : <https://fr.wikipedia.org>

Symfony : <https://symfony.com/>

React: <https://facebook.github.io/react/>

Referentiel institute agile: <http://referentiel.institut-agile.fr/>

Webopedia: <http://www.webopedia.com/>

Doc.ubuntu: <https://doc.ubuntu-fr.org/lamp>

Blog.modulus : <http://blog.modulus.io/>

Phpdoc : <https://www.phpdoc.org/>