

# Kit de démarrage

Révision V1.3

Date 28 Février 2014

Contact Louis BAYLE

lbayle.work@gmail.com Tel. 04 42 604 734



## Historique des révisions

Révision	Date	Modification (Nature et chapitres concernés)	Auteur
1.0	27/11/2012	Version initiale	Louis BAYLE
1.1	30/11/2012	Ajout chapitre 'Suivi de l'avancement des projets'	Louis BAYLE
1.2	26/09/2013	Description des Commandes & Contrat de service	Louis BAYLE
1.3	28/02/2014	Maj Commandes et retrait du logo Atos.	Louis BAYLE

## Références

Élément	Description
http://codevtt.org	Site officiel de CodevTT (news, doc, forum, download,)
http://codevtt.org/codev	Démonstrateur en ligne
http://www.mantisbt.org	Site officiel de MantisBT
https://github.com/lbayle/codev/blob/master/doc/ fr/CodevTT Kit demarrage.pdf?raw=true	Dernière mise à jour de ce document

CodevTT.org Date: 28/02/14



## **Sommaire**

1 Glossaire	4
2 À propos de ce document	5
3 Présentation de CodevTT	5
4 Concepts de base	6
4.1 Mantis & CodevTT	6
4.2 Champs spécifiques	6
4.3 Types de tâches	7
4.3.1 Bugs et évolutions	7
4.3.2 Tâches annexes	7
4.3.3 Tâches externes	8
4.4 Rôles utilisateurs	8
5 Paramétrage	9
5.1 Création d'une équipe	9
5.1.1 Description de l'équipe	9
5.1.2 Projet de Tâches Annexes	10
5.2 Configuration de l'équipe	10
5.2.1 Général	11
5.2.2 Utilisateurs	11
5.2.3 Projets	12
5.2.4 Astreintes	13
6 Suivi d'activité	14
6.1 Saisie de CRA	15
6.2 Suivi d'activité	
6.2.1 Activité hebdomadaire	16
6.2.2 Activité Mensuelle	
6.2.3 Activité projets	18
7 Suivi de l'avancement des projets	18
7.1 Charges détaillées	18
7.2 Tâches en dérive	19
8 Commandes & Contrat de Service	
8.1 Contrat de service & tâches annexes	
8.1.1 Les commandes sont gérées de manière indépendante	
8.1.2 Les commandes s'inscrivent dans une continuité	20
8.2 Provisions	71



## 1 Glossaire

Désignation	Origine	Définition
Tâche	CodevTT	Une tâche représente une action sur laquelle du temps va être consommé. Les tâches sont regroupées par projets et possèdent des attributs tels que : description, priorité, criticité, charge initiale, reste à faire, jalon, etc. Les tâches respectent le cycle de vie du projet, et sont matérialisées par des fiches Mantis.
Fiche	Mantis	Une Fiche est le pendant de la Tâche CodevTT dans Mantis.
Projet	Mantis	Conteneur de fiches ayant des caractéristiques communes et un même cycle de vie.
CRA		Compte Rendu d'Activité
Manager	CodevTT	Rôle utilisateur : Chef de projet ou directeur de projet.
Développeur	CodevTT	Rôle utilisateur : Membre de l'équipe sans responsabilité de management.
Observateur	CodevTT	Rôle utilisateur : Personne externe ayant un droit de regard sur les activités de l'équipe.
RAF		Reste À Faire (backlog)

CodevTT.org Date: 28/02/14



## 2 À propos de ce document

Ce document est à l'attention des personnes n'ayant aucune expérience de CodevTT et désirant mettre en place l'outil dans leur équipe. On ne traitera ni de l'installation, ni des fonctionnalités avancées ou des indicateurs que propose l'outil.

La configuration de CodevTT pour une nouvelle équipe prend **environ 15 minutes**.

Une connaissance de Mantis est indispensable pour une bonne compréhension du document.

### 3 Présentation de CodevTT

CodevTT est un outil de gestion de projet, permettant un suivi détaillé de l'avancement des projets et des activités de l'équipe.

Sa caractéristique principale est son lien direct avec MantisBT - un système de suivi d'anomalies - dont on étendra le scope d'activité.

En puisant des informations dans la base de données de MantisBT et en simplifiant au maximum la saisie des comptes rendus d'activité des utilisateurs, CodevTT réduit considérablement le nombre d'opérations manuelles à effectuer pour générer des rapports, statistiques, alertes, diagramme de Gantt et autres indicateurs de production et de suivi.

Les données de MantisBT étant tenues à jour en permanence par les développeurs, le chef de projet peut avoir une vue en temps réel de l'avancement du projet, sans créer de surcharge de travail pour l'équipe.

Les informations remontées par CodevTT permettent au chef de projet d'identifier plus rapidement les points durs du projet. La réduction d'un grand nombre de tâches récurrentes lui permet de se concentrer d'avantage sur les parties nécessitant le plus d'attention et d'analyse.

Les statistiques aident à identifier les actions à entreprendre pour améliorer la productivité de l'équipe et permettront d'en mesurer l'efficacité à court/moyen/long terme.

La section 'Contrats et Commandes' permet d'avoir une vue 'client' de l'avancement, et propose des indicateurs qui pourront lui être remontés.

CodevTT est donc un outil de gestion de projet réactif, en lien direct avec le développement, et se fixe comme objectif la maîtrise du suivi et la réduction des coûts de management par la simplification et l'automatisation des processus.



## 4 Concepts de base

### 4.1 Mantis & CodevTT

La première fonctionnalité de CodevTT est de fournir une interface simple et efficace pour remplir les CRA. On aura donc la possibilité de saisir des imputations sur des tâches aussi variées que « correction de l'anomalie X », « développement de la fonctionnalité Y », « réunion », « mise à jour de wiki », « absence », etc.

Mantis a initialement été conçu pour gérer des anomalies, mais il peut parfaitement être utilisé pour gérer l'ensemble des tâches de développement, ou toute autre activité de la journée.

CodevTT va donc s'appuyer sur Mantis pour gérer l'ensemble des tâches quotidiennes. Cela nécessite de rajouter un certain nombre de champs supplémentaires dans les fiches, mais Mantis le permet grâce aux « champs personnalisés ».

Il est important de souligner que CodevTT utilise la même base de donnée que Mantis et y ajoute un certain nombre de tables. En revanche, CodevTT se veut non intrusif, et ne modifie aucune des tables initiales. Ce point est important, car il assure que Mantis pourra être mis à jour indépendamment, et que l'on puisse facilement « débrancher » CodevTT si nécessaire.

## 4.2 Champs spécifiques

Pour faire du suivi d'activité, un certain nombre de champs personnalisés doivent être ajoutés aux fiches.

### Charge manager

C'est la charge (en jours/homme) estimée pour la réalisation de la tâche. Ce champ n'est pas montré aux développeurs. Il peut être à zéro si la tâche n'a pas été prévue lors du chiffrage initial.

### Charge initiale

C'est la charge (en jours/homme) mise à disposition du développeur pour effectuer la tâche.

#### RAF

Le reste à faire (en jours/homme) est estimé et mis à jour par les développeurs.

### • **Deadline** (Date jalon)

Date pour laquelle la tâche doit être réalisée. Si elle n'est pas précisée, la date de la 'Version ciblée' sera utilisée. Si la version ciblée n'a pas de date associée, alors cette tâche n'a pas de deadline.

Note : Il existe d'autres champs définis par CodevTT, seuls ceux nécessaire à la compréhension des concepts de base sont exposés ici.



## 4.3 Types de tâches

Lors de la conception de l'outil, quatre types de tâches ont été identifiées : Les bugs, les évolutions, les tâches annexes et les tâches externes.

## 4.3.1 Bugs et évolutions

Il n'y a aucune différence entre un bug et une évolution. Ce sont des tâches de développement qui font parti d'un même projet et partagent le même cycle de vie Mantis. Pour les besoins des indicateurs, on utilisera tout de même le champ 'CodevTT\_Type' qui propose les valeurs [Bug / Task] pour les identifier.

### 4.3.2 Tâches annexes

Si bugs et évolutions sont des tâches de travail effectif prises en compte dans l'avancement du projet, il existe bon nombre d'activités qui ne rentrent pas dans cette catégorie mais qu'il faut néanmoins prendre en compte dans le consommé.

Les tâches annexes ne possèdent pas de charge initiale, ce sont des tâches persistantes (sans cycle de vie) et on essaiera d'en limiter leur nombre en leur donnant un descriptif suffisamment générique. Il est bien entendu possible de créer une tâche annexe dédiée à une activité précise pour la tracer et en évaluer le coût (ex : Audit qualité sur le projet).

CodevTT classe ces tâches annexes en plusieurs catégories :

### Management

Deux tâches suffisent généralement dans cette catégorie : « Gestion de projet » et « Réunion » (Il serait en effet contre-productif de créer une nouvelle tâche pour chaque réunion). On pourra en créer d'autres en fonction des besoins du chef de projet. (« Avant vente », « Administration et logistique », « Réunion Hebdo », « Réunion Client », …)

### Outillage

ex : « Maj des scripts de compilation », « mise en place poste de dev », « Administration Mantis/CodevTT »

### Capitalisation

Regroupe les activités internes à l'équipe, n'ayant pas de rapport direct avec les tâches de production.

ex : « Maj Wiki », « Formation outils de production », « LEAN », « Montée en compétence »

### Incident

Permet de gérer tous les imprévus qui ralentissent la production. Des indicateurs permettront d'en mesurer l'impact afin de décider des actions à entreprendre.



Ex : « Indisponibilité réseau », « Blocage compte de production », « Environnement de compilation non opérationnel », « intervention plate-forme »

Toutes ces tâches sont regroupées dans un « projet de tâches annexes » **spécifique à chaque équipe** et qui sera mis en place par CodevTT lors de la création d'une nouvelle équipe.

### 4.3.3 Tâches externes

Les tâches externes représentent les tâches non rattachées aux activités de l'équipe, et imputés sur d'autres OTPs.

On y trouve deux catégories :

#### Absences

Généralement une seule tâche « absence » est nécessaire, mais pour des besoins de logistique on peut vouloir en créer d'autres comme « maladie », ...

#### Autre activité

Contrairement aux absences, ces activités sont effectuées dans le cadre du travail. Cela permet d'effectuer des imputations sur des projets non gérés avec CodevTT comme les formations individuelles.

Là aussi, sauf besoin spécifique, une seule tâche est nécessaire : « Autre tâche externe ».

Ces tâches sont regroupées dans un « projet de tâches externes» commun à toutes les équipes.

### 4.4 Rôles utilisateurs

Afin de gérer les droits d'accès aux informations, trois profils utilisateurs ont été définis au sein d'une équipe :

### Manager

Le manager à une vue complète de l'équipe, il est le seul à avoir accès à l'ensemble des indicateurs, à la 'charge manager' et au 'contrat de service' (OTP). Il peut imputer sur les projets de son équipe et créer/modifier contrats et commandes.

### Développeur

Le développeur n'a qu'une vue partielle : la partie management lui est cachée. Il impute sur les projets de son équipe.

#### Observateur

L'observateur à une vue complète de l'activité et des contrats de l'équipe, mais ne pourra ni faire d'imputations, ni modifier les paramètres de l'équipe.

Chaque équipe a un 'team Administrator' celui-ci est généralement un des managers de l'équipe, il est le seul à avoir les droits d'administration sur l'équipe.



Une équipe « CodevTT Admin » permet d'identifier les personnes ayant un droit d'administration sur l'ensemble des paramètres de CodevTT ainsi que l'ensemble des équipes.

## 5 Paramétrage

La première chose a faire est de créer une équipe et d'y affecter utilisateurs et projets Mantis. Cette action simple se fait **en 15 minutes** et donne la possibilité aux membres de l'équipe de mettre à jour leur CRA et au manager de faire du suivi d'activité.

Le reste du paramétrage (création des commandes, contrats de service, indicateurs) peut se faire dans un deuxième temps, et ne sera décrit que sommairement dans ce document.

## 5.1 Création d'une équipe

Pour créer une nouvelle équipe, aller dans le menu « Admin/Créer équipe »



La page de création d'équipe comporte deux sections :

## 5.1.1 Description de l'équipe

Description de l'équipe				
Nom de l'équipe	UNA STELLA *			
Description	Equipe en charge de	*		
Team Leader	Ibayle     ★			

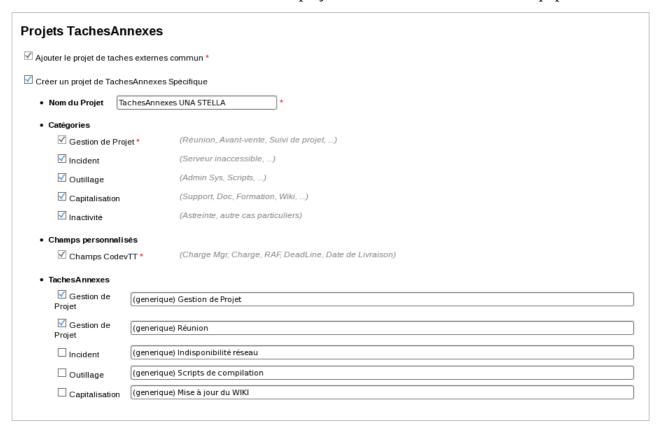
Tous les champs sont obligatoires, le « Team Administrator» aura les droits d'administration sur cette équipe.

Attention : Si vous sélectionnez une autre personne que vous-même en temps que team Administrator, et que vous ne faites pas partie de l'équipe 'CodevTT Admin' vous n'aurez pas la possibilité d'accéder à la section « Configuration de l'équipe ».



## 5.1.2 Projet de Tâches Annexes

La deuxième section concerne la création du projet de tâches annexes associé à l'équipe.



### Catégories

Les différentes catégories ont été décrites à la section 4.3.2 à l'exception de la catégorie « inactivité » qui peut être utilisée dans de rares cas d'absences spécifiques à l'équipe.

### Tâches Annexes

Par commodité, CodevTT propose ici de créer quelques tâches annexes. À vous de définir celles dont vous avez besoin, vous pourrez en ajouter d'autres en utilisant l'interface « Rapporter un bogue » du menu Mantis.

## 5.2 Configuration de l'équipe

Une fois l'équipe créée, on bascule automatiquement sur la page d'édition de l'équipe.



### 5.2.1 Général

Equipe UNA STELLA				
Team Leader	Ibayle   💠	Mise à jour		
Date de création	2012-11-23	Mise à jour		
Activé	<b>✓</b>	Mise à jour		
Note : Un TeamLeader doit également être déclaré comme TeamMember pour être pris en compte dans les indicateurs de production.				

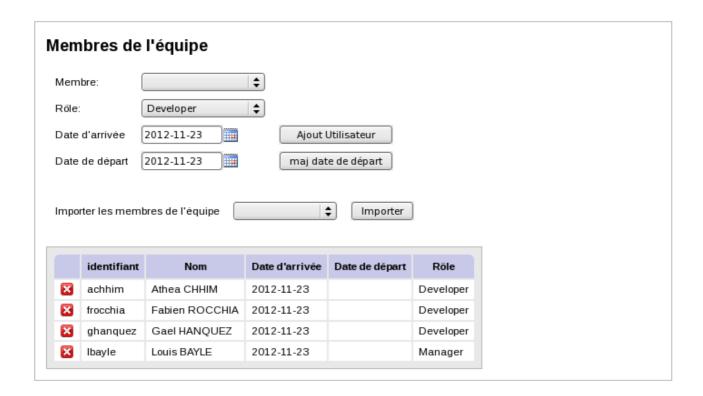
Le Team Administrator possède des droits d'administration, il est le seul à pouvoir modifier la liste des membres et des projets de l'équipe. Il n'occupe en revanche aucun rôle particulier au sein de l'équipe, donc si le Team Administrator est un membre actif (qu'il impute sur les projets de l'équipe) alors il doit également être ajouté dans la liste des membres.

L'attribut 'Activé' permet de ne plus afficher l'équipe dans les pages de CodevTT. Décochez cette case si l'équipe est dissoute, mais que vous voulez garder la possibilité d'afficher ses statistiques.

### 5.2.2 Utilisateurs

CodevTT.org Date: 28/02/14





La liste des rôles à été décrite au paragraphe 4.4.

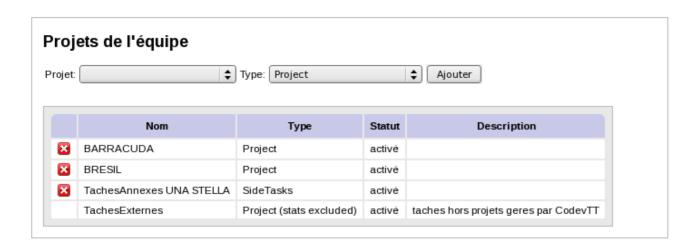
Pour ajouter un utilisateur, sélectionnez son login dans la liste déroulante, donnez sa date d'entrée dans l'équipe et validez avec le bouton « Ajout Utilisateur ».

Lorsqu'un membre quitte l'équipe, sélectionnez son login dans la liste déroulante, donnez sa date de départ et validez avec le bouton « maj date de départ ».

ATTENTION : supprimer un membre de l'équipe modifie tout les indicateurs !

## 5.2.3 Projets





Sélectionnez les projets Mantis sur lesquels l'équipe va travailler.

Il existe plusieurs types de projets, ceux-ci sont liés aux types de tâches décrits au paragraphe 4.3 mais il en existe d'autres :

### Project

Projet normal, contenant bugs et évolutions

#### SideTasks

Projet de tâches annexes. (défini à la création de l'équipe.)

### Project (no common jobs)

Projet normal, contenant bugs et évolutions, mais ayant sa propre liste de 'Postes' (Se référer au manuel d'utilisateurs).

### Project (stats excluded)

Les imputations sur les tâches de ce projet ne seront pas pris en compte dans les statistiques. Cela peut être utile si l'on utilise des projets conteneurs dans mantis, dont les tâches ne sont pas à prendre en compte. (Ex : Fiches de livaisons regroupant des tâches filles, etc.)

### 5.2.4 Astreintes

À ce stade de CodevTT (v0.99.19), les astreintes ne sont autres que des absences, identifiées par une couleur particulière dans la vue des absences.



### Ajouter des Astreintes

Note: une astreinte est très semblable à une tâche 'Absence'. Elle est spécifique à l'équipe.

Aucune tâche de catégorie 'inactivité' définie dans les projets de tâches annexes.

Pour ajouter une tâche de type astreinte, créez une fiche Mantis de catégorie « inactivité » dans le projet de tâches annexes. Rechargez la page.



Sélectionnez la fiche et validez avec le bouton « Ajouter ».

### 6 Suivi d'activité

La configuration est terminée, il est maintenant possible d'utiliser CodevTT pour faire du suivi d'activité.

Cette fonctionnalité permet de remplacer les traditionnels 'CRA Excel' en proposant un formulaire de saisie très similaire. La particularité principale est que les imputations se font exclusivement sur des fiches Mantis.

Cela permet une meilleure maîtrise de l'activité pour le manager (pas d'imputations 'sauvages' sur des tâches non identifiées) et d'autre part, - le périmètre d'une fiche Mantis étant bien défini - d'obtenir un RAF plus précis de la part des développeurs.

Cette saisie des CRA a été très bien acceptée par les développeurs parce qu'elle simplifie considérablement la saisie des imputations. et comme tout est fait pour inciter à saisir les imputations quotidiennement, le manager est informé en temps réel de l'avancement des projets.

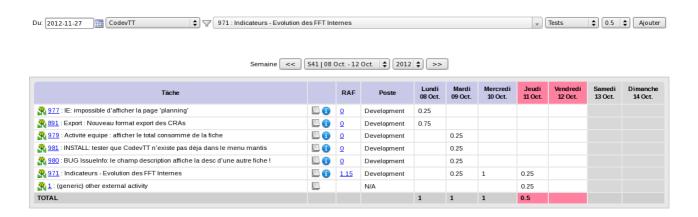


### 6.1 Saisie de CRA

Pour saisir des imputations, ouvrez le menu « Activité / Saisie des CRA »



Si vous êtes manager, une liste déroulante s'affiche et permet de sélectionner un des membres de l'équipe. Sinon, la page de saisie des CRA s'affiche directement.



Le CRA est composée de deux parties : un formulaire de saisie, et un tableau récapitulatif des imputations de la semaine.

Le formulaire est composé de 5 champs :

Date

Date de l'imputation

Projet

Le projet Mantis permet de réduire la liste des tâches candidates.

Tâche

Liste des fiches Mantis imputables.

Poste

Définit le type d'activité qui a été pratiquée sur cette tâche (étude, analyse, développement,



tests, documentation, ...). Cette liste est paramétrable dans le menu « Admin / éditer postes ».

### • Durée (jour)

L'unité est le jour, '0.5' représente une demi-journée. Il à été délibérément choisi de ne pas utiliser des heures comme unité car le nombre d'heures travaillés dans une journée peut varier en fonction des contrats des utilisateurs.

Les jours comportant des erreurs de saisies sont affichés en rouge. Après chaque saisie, s'ouvre automatiquement une boite de dialogue de saisie du RAF. En effet celui-ci doit être réévalué après chaque imputation.



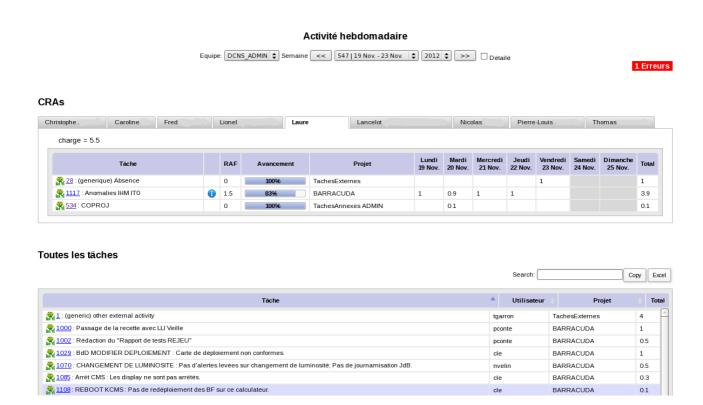
### 6.2 Suivi d'activité

Les CRAs étant saisis, le manager peut les consulter de trois manières différentes : « l'activité hebdomadaire », « l'activité mensuelle » et « l'activité projet ».

### 6.2.1 Activité hebdomadaire

Cette page comporte deux sections. La première regroupe l'ensemble des CRA de l'équipe, tels que saisis par les utilisateurs. La deuxième montre une liste exhaustive des tâches effectuées dans la semaine avec leur consommé.





### 6.2.2 Activité Mensuelle

Cette page permet d'afficher la liste des tâches par utilisateur, sur une période précise.



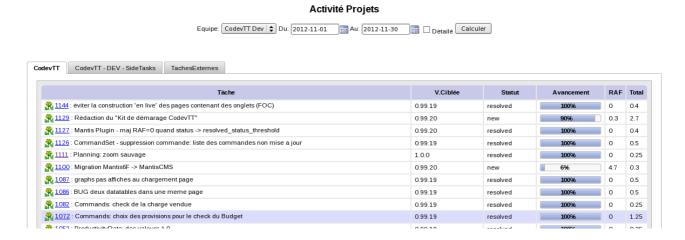
La zone cliquable 60 Erreurs ouvre une boite de dialogue affichant les erreurs/omissions d'imputations sur la période.

CodevTT.org Date: 28/02/14



## 6.2.3 Activité projets

Cette page permet d'afficher la liste des tâches par projets, sur une période précise.



## 7 Suivi de l'avancement des projets

CodevTT propose plusieurs vues pour suivre l'avancement. Dans un premier temps, nous ferons abstraction du suivi des Commandes pour rester au niveau Mantis et suivre l'activité d'une équipe sur ses projets.

Dans le menu principal, « Projet » permet d'afficher une page contenant deux sections.

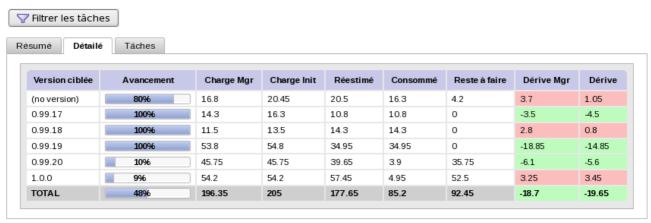
## 7.1 Charges détaillées

La première section donne les indicateurs standards de suivi d'avancement : % Avancement, Charge, Ré-estimé, Consommé, RAF et Dérive.

Un ensemble de filtres sont disponibles. Dans la capture ci-dessous, seul le filtre « Version ciblée » à été appliqué.



#### Info Projet

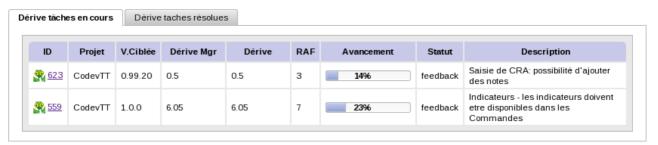


Note : Pour plus de détail sur les indicateurs, se référer au guide de l'utilisateur.

### 7.2 Tâches en dérive

La deuxième section donne le détail des tâches en dérive de charge. Ces fiches sont à surveiller tout particulièrement.

#### Tâches en dérive



### 8 Commandes & Contrat de Service

Pour gérer les contrats, CodevTT propose trois niveaux d'abstractions : Contrat de service, Macro-Commande et Commandes.

### La Commande

Permet de regrouper un ensemble de tâches, y associer un budget, et définir des provisions (marge de risque, garantie, qualité, ...).

### La Macro-Commande

Une macro-commande est un ensemble de commandes.



• Le Contrat de Service (équiv. OTP)

C'est le niveau le plus élevé, il permet de regrouper l'ensemble des Commandes d'un OTP, les tâches annexes, les provisions et de présenter un avancement global sur l'activité.

### 8.1 Contrat de service & tâches annexes

Il y a deux visions sur la gestion des tâches annexes :

- Soit on veut les gérer chaque commande de manière indépendante.
- Soit on considère que les commandes s'inscrivent dans une continuité (en particulier pour une TMA), et que les tâches annexes qui ne sont pas des tâches visibles du client doivent être traitées à un niveau supérieur.

## 8.1.1 Les commandes sont gérées de manière indépendante

Cette méthode est a appliquer lorsque l'on veut obtenir un cloisonnement fort entre les commandes. Cela consiste à ne pas utiliser macro-commandes et contrat de service.

Dans ce cas, les tâches relatives au management, ainsi que les autres tâches annexes doivent être inclues dans la commande.

- L'avantage c'est qu'on aura la maîtrise du consommé sur les tâches annexes au niveau de la commande, et donc un rapport direct avec la charge vendue.
- L'inconvénient de cette méthode réside dans son côté fastidieux de mise en œuvre. Il y aura un processus de réplication des tâches annexes pour chaque nouvelle commande. (ex : 'gestion de projet Cmd1', 'gestion de projet Cmd2', 'maj Wiki Cmd1', 'maj Wiki Cmd2', ...).

Le fonctionnement est simple, mais la mise en œuvre relativement lourde et contraignante sur le long terme. On n'utilisera cette méthode si l'on sent qu'il y a un risque de dérive sur les tâches annexes. Ce mode est également choisi par les chefs de projets désirant voir l'ensemble de l'activité dans le WBS (Petits projets n'ayant qu'une seule commande).

### 8.1.2 Les commandes s'inscrivent dans une continuité

Dans ce modèle on considèrera les tâches annexes comme n'étant pas directement liées aux commandes : La mise à jour du Wiki, l'outillage, la mise en place d'une plateforme de tests, la montée en compétence, le support interne relèvent de la capitalisation pour l'équipe. Ces tâches n'ont pas de charge définie, ni de date jalon. En revanche, elles ont un coût, et celui-ci est « absorbé » par les provisions.

Les provisions sont vendues au niveau des commandes, certes, mais ne concernent généralement pas directement les tâches de production. Elles s'inscrivent dans un processus global.



Nous considérons le management de la même manière : on vend une provision de management lors de chaque commande, mais l'activité de gestion de projet dépasse le cadre de celle-ci et ne s'équilibre qu'au niveau le plus élevé : le contrat de service.

Pour résumer simplement : le contrat de service fait le lien entre les tâches de production (commandes), les tâches annexes et les provisions.

- Avantages : Il n'est plus nécessaire d'avoir des tâches annexes spécifiques à chaque commande, la saisie des CRA est simple et l'utilisation de CodevTT bien plus fluide.
- Inconvénient : Il faudra veiller à ce que les tâches annexes ne soient pas utilisées de manière abusives pour masquer une dérive sur une commande donnée.

Les abus étant rares et faciles à repérer, cette méthode est préférable dans la majeure partie des cas.

Note : il est impératif de lier le projet SideTasks de l'équipe au Contrat de service.

### 8.2 Provisions

Dans les propositions commerciales que nous faisons à nos clients, nous vendons en plus de la charge de développement, des provisions pour risque, management, documentation, qualité, etc.

CodevTT permet d'ajouter ces provisions au niveau de chaque commande. Celles-ci permettent d'absorber les tâches annexes et/ou les dérives sur les tâches de production.

Prenons l'exemple d'une commande dont la charge de travail est estimée à 100 jours homme. Nous Vendons comme coût de fonctionnement les charges suivantes :

- + 15 % Gestion de projet
- + 10 % Risque
- + 10 % Garantie
- + 5 % Support à l'intégration
- + 5 % Divers

Soit un total de 145 jours qui sera facturé au client.

Les 100 jours de développement seront réparties sur les tâches (charge manager), le reste sera défini en provisions sur la commande.

Les indicateurs de suivi du budget prendront en compte ces provisions pour le suivi.