

État des lieux fonctionnels de notre méthodologie de travail

Contributeurs :

Soufien Techini : PDG de Sastec France

Amina Abdelkafi : Chef de projet technique DevFactory

Olivier Zuratas : Directeur développement DevFactory

octobre 2021

Introduction

DevFactory est une plateforme de micro-services qui s'adresse aux porteurs de projets ayant comme impératif de développer plus rapidement tout en maintenant la qualité et la performance de leurs applications web.

La plateforme permet aux entreprises souhaitant créer ou construire des applications informatiques ; de recueillir les informations nécessaires dans un premier temps ; de les interpréter et de les transformer en solution exploitable sur leur marché dans un second temps.

La plateforme est basée sur des micro services et des algorithmes automatisés permettant un gain considérable en termes de coûts et de temps.

L'objectif affiché est de réduire la charge de travail des équipes en proposant des outils et processus permettant de créer plusieurs projets de manière simultanée.

Une planification inappropriée, le non-respect des délais, les dépassements de budget et un manque de coordination peuvent compromettre un projet.

Depuis deux ans, **SASTEC** a développé des outils d'optimisation et de réalisation de projets, principalement pour un de ses clients éditeurs de logiciels.

La réalisation d'un projet peut être effectuée en suivant un processus divisé en quatre étapes :

- **Planification** : réfléchissez à votre application avant de commencer à la créer
- **Données** : définissez les données que vous souhaitez collecter, afficher, traiter et rapporter
- **Design** : pensez aux parcours utilisateurs, comment voulez-vous que les autres interagissent avec votre application ?
- **Développement** : quelles sont les fonctionnalités nécessaires pour faire tourner votre application ?

Ce document présente nos outils et notre méthodologie pour chacune de ces 4 étapes.



Planification

Dans cette partie nous présentons les principes de base de la gestion de projets, les étapes de la planification de la gestion d'un projet, ainsi que les techniques spécifiques que nous utilisons au quotidien.

La méthode classique pour préparer un plan de projet est de rédiger un document (Word ou feuilles de calcul Excel) qui décrit le périmètre, les objectifs et les livrables du projet. Faire un plan de projet permet entre autres de définir le travail à fournir et de préciser qui effectuera quelles tâches et quand.

La phase de planification est sans doute la partie la plus importante de la gestion d'un projet. L'utilisation d'un format de planning projet visuel permet :

- D'analyser clairement les projets et de définir tous les livrables.
- De diviser n'importe quel projet en parties plus petites et donc plus faciles à gérer.
- D'estimer et de détailler le temps, les coûts et les ressources nécessaires pour l'exécution d'un projet.

En transformant les idées non structurées en maps visuelles organisées, nous pouvons donner à toutes les personnes impliquées dans un projet une compréhension plus claire de leurs objectifs et une meilleure maîtrise de leur temps. Nous utilisons le mindMap (miro.com) qui permet de subdiviser les principales activités ou sous-activités du projet en activités plus petites et plus gérables jusqu'à ce que les activités soient définies de manière suffisamment détaillée pour soutenir la gestion et le développement des travaux du projet.

Avec DevFactory, l'objectif est de développer notre propre mindMap et permettre de :

- générer automatiquement des tâches après la phase de planification par un mindMap.
- travailler avec un référentiel d'informations unique et centralisé pendant toute la durée du cycle de vie du projet.
- capturer et partager en temps réel les nouvelles idées, informations et suggestions de toutes les personnes impliquées.

Données

Après la phase de planification et de préparation des prérequis, il est temps de créer le modèle de données. Nous allons créer une ou plusieurs tables avec des champs dessus, éventuellement les charger avec des données.

On utilise actuellement des méthodes classiques qui permettent de transformer les données de feuille de calculs en modèle JSON et des tables.

Il y a six champs créés automatiquement pour chaque table. Les six champs contiennent des informations renseignées automatiquement sur la table, telles que la date de sa création, la date de sa dernière mise à jour et la personne qui l'a mise à jour. Les champs contiendront également un identifiant unique pour la table. Ces champs ne peuvent pas être manipulés.

Nom	champ de table	La description
Créé par	created_by	Utilisateur qui a créé l'enregistrement.
Créé le	created_on	Date/heure de création de l'enregistrement.
Mis à jour par	updated_by	Utilisateur qui a mis à jour l'enregistrement pour la dernière fois.
Mis à jour	updated_on	Heure à laquelle le dernier enregistrement a été mis à jour
ID système	id	Identifiant unique de l'enregistrement. Il est attribué automatiquement et unique dans toute l'instance.
Mises à jour	mod_count	Champ numérique qui compte le nombre de mises à jour de l'enregistrement depuis la création de l'enregistrement.

Cette étape nous permet de préparer des fake API basées sur une feuille de calcul.

- Chaque feuille de calcul doit correspondre à un modèle (Entity) dans le projet.
- Chaque colonne peut devenir un champ dans la table
- Chaque ligne peut devenir un enregistrement de données dans la table

Notre objectif avec **DevFactory** est d'automatiser la création et le déploiement des fake API à partir d'un document Excel. Pour cela on prépare un outil d'import de données et cet outil va déclencher un worker qui reste toujours en écoute d'une nouvelle demande. Le rôle du worker est de créer une instance docker qui permet d'afficher un studio d'API. Le rôle du studio est d'afficher la liste des tables et les API de base pour chaque table.

Cet outil permet de :

- documenter l'API
- tester la conception de l'API
- préparer des services d'intégration de l'API pour l'équipe front et backend
- Synchroniser les tâches entre l'équipe front et backend.

Design

Le Wireframe et le prototypage font partie de notre processus de conception. Ce sont les deux livrables de conception les plus souvent associés à la conception UX.

Le rôle d'un wireframe est de créer une base pour la conception finale. Les concepteurs créent des wireframes et les valident pour s'assurer que le wireframe contient tous les éléments essentiels. Un nombre limité d'éléments visuels permet à l'équipe de se concentrer uniquement sur les décisions de conception de base avant de plonger dans les détails.

Les concepteurs créent généralement des wireframes au début du processus de conception, avant que l'équipe ne commence à travailler sur les détails visuels.

On utilise actuellement Figma pour présenter le wireframe qui facilite le partage des wireframes et le partage des formations. Mais bien évidemment, il est plus facile de concevoir des wireframes lorsque l'on a des références visuelles.

Pour cela nous avons créé des collections utiles de modèles de Wireframing qui nous aident à démarrer le projet. Nous avons mis au point un système qui aide les concepteurs à créer des conceptions esthétiques, conviviales et évolutives très rapidement.

Pour optimiser la durée de préparation d'une maquette et un prototype pour nos clients, on prépare un fichier de conception avec tous les composants : icônes, styles de couleurs, styles de texte et styles de grille pour commencer immédiatement le travail de conception.

Ce kit de conception est utile pour comprendre comment créer, organiser et nommer des composants dans Figma. Il garantit ainsi le respect des meilleures pratiques en termes de conception.

L'objectif avec **DevFactory** est de préparer des plugins Figma qui facilitent les partages de collections de wireframe et de kit de conceptions. Nous développerons par ailleurs un plugin qui facilite la synchronisation entre les designers et les développeurs.

Ainsi, les développeurs peuvent rester dans le contexte et avoir la garantie de prendre toujours comme référence les dernières conceptions.

Développement

Après la validation de prototype, le plan établi dans la charte du projet est mis en application. Les tâches sont affectées à chacun des membres de l'équipe et les choses sérieuses commencent.

Développer plus rapidement un produit, sans en altérer la qualité, est un challenge pour tout porteur de projet.

Pour répondre à cet enjeu crucial des stratégies de développement existent.

1 – Création des projets Boiler plate

L'objectif principal de Boiler plate est de mettre en place des projets prêts à l'emploi. Le Boiler plate essaie donc de suivre les meilleures pratiques partout où il le peut, savoir :

- Une structure de répertoires cohérente, significative pour les nouveaux membres de l'équipe et soutenant les points ci-dessus.
- Bien documenté, pour montrer aux nouveaux développeurs pourquoi les choses sont configurées comme elles sont.
- Le système de construction doit fonctionner automatiquement, sans implication du développeur. Les projets doivent être testés, lissés, compilés et minimisés, prêts à être utilisés dans un environnement de production.

2 – Utilisation des fake API

L'utilisation de fake API peut s'avérer pertinente pour assurer les fonctionnalités backend des applications, que ce soit dans sa phase de développement ou dans sa phase de test. Parfois, l'API réelle qui s'intègre à l'application peut ne pas être prête à être testée, ce qui peut être une cause de blocage. Pour surmonter ce genre de situations, nous suggérons de développer notre propre fake API et la déployer. Ainsi, elle peut être facile à utiliser par n'importe qui et de n'importe où pour avancer rapidement.

3 – Socle technologique (Technology Stack)

Nous n'utilisons pas une approche mono stack. Notre large stack nous donne la possibilité d'utiliser les meilleurs outils pour chaque cas d'utilisation, anticiper les évolutions technologiques et éviter le verrouillage technologique en provenance de certains fournisseurs.

La meilleure stratégie est de choisir les bons outils en fonction du projet que vous réalisez et d'éviter d'utiliser une multitude d'outils puissants, souvent lourds à mettre en place et qui nuisent à la productivité.

4 – Disposer d'un environnement de développement similaire à la production

La mise en production d'un projet informatique est une étape très sensible. Elle donne souvent lieu à des erreurs inattendues.

Ces erreurs sont issues généralement d'un module manquant, d'une version incompatible, d'une bibliothèque non installée.

Ces problèmes sont généralement dus au fait d'avoir un environnement de production différent à celui du développement. Par exemple, lorsque vous faites le développement de votre projet sur Windows et puis le déploiement sur Ubuntu Server, vous rencontrez des problèmes d'encodage, de chemin, de variables d'environnement.

Disposer d'un environnement de développement identique à celui de la production est un élément essentiel pour garantir une production sans problème.

Pour cela, nous suggérons la mise en place d'un système qui automatise la création des environnements de travail par devSpace.io. Bien évidemment on utilise les Dockers, c'est un conteneur semblable à une machine virtuelle dans lequel on peut installer tout l'environnement et par la suite profiter d'un environnement 100 % identique et quelle que soit la machine que vous utiliserez, le code s'exécutera de la même manière.

5 – Don't repeat yourself

Comme son nom l'indique, la technique du "Don't Repeat Yourself", souvent abrégée DRY suppose que ses utilisateurs évitent au maximum de se répéter. Il s'agit d'une des bonnes pratiques de développement. Notre approche consiste à développer, puis à utiliser des "*Library*" qui permettent de gagner du temps dans le développement des logiciels.

Notre objectif est de préparer une collection de fonctions fournies en développement, que ce soient des classes ou un ensemble de fichiers objets. Les bibliothèques de programmes comprennent toujours des sous-programmes qui facilitent le travail avec des outils d'aide à la programmation.

La plateforme DevFactory permettra de partager ces collections de Library avec l'IDE (environnement de développement) et d'intégrer un assistant AI qui proposera des codes pour optimiser la durée de développement.

L'IDE sera synchronisé dans le Cloud pour faciliter la tâche de développement et utiliser un environnement de travail similaire aux productions.