## Refonte du module de procédure

### Définition de la mission

#### Contexte général

L’entreprise Hutchinson est un groupe d’industriels filiale de Total Energies, spécialisé dans la fabrication de solutions multi-matériaux. Leader mondial dans la transformation d'élastomères, ils fournissent des secteurs comme l’automobile, l’aéronautique ou l’énergie[[1]](#footnote-1). Leurs produits incluent des systèmes antivibratoires, des solutions d’étanchéité et des systèmes de gestion des fluides[[2]](#footnote-2).

Dans la continuité de leur stratégie d’amélioration continue, Hutchinson cherche à harmoniser ses processus internes. Leur objectif est donc de créer un référentiel de procédures standardisées tout en permettant des adaptations spécifiques dans leurs sites de production.

PICC étant en discussion avec l’entreprise Hutchinson pour lui vendre notre solution, cette dernière souhaite développer certaines parties du logiciel pour répondre à ses besoins. Ainsi, le module de gestion des procédures sera remanié pour qu’il soit à la fois standardisé et adaptable. Cela permettra de faciliter la diffusion et l’application des meilleures pratiques au sein du groupe. Les enjeux de cette mission seront donc :

* D’Assurer la cohérence des procédures au niveau du groupe tout en respectant les particularités locales
* Permette une exécution collaborative des procédures avec une traçabilité et un suivi en temps réel
* Faciliter la mise à jour et l’évolution des procédures en fonction des retours d’expérience

#### Objectifs de la mission

Après avoir analysé le besoin de notre futur client Hutchinson, nous avons défini une suite d'objectifs pour répondre à ces attentes. Bien que le module de procédures soit déjà fonctionnel, il incombe d’apporter les modifications suivantes :

* Permettre de créer un référentiel commun à l’ensemble des sites de production
* Créer un mécanisme d’héritage au sein des procédures permettant la modification des procédures locales et la mise à jour par rapport aux procédures globales
* Améliorer la gestion des procédures en offrant une vue d’ensemble de toutes celles exécutées dans un contexte donné
* Ajouter un mécanisme de synchronisation en temps réel
* Permettre de visualiser en un coup d’œil l’avancement, les blocages
* Ajouter un système pour définir le rôle et la responsabilité de chacun vis-à-vis de chaque étape d’une procédure
* Permettre l’exécution partagée des procédures
* Ajouter un mécanisme de validation contrôlant les moments critiques de l’exécution

Ces modifications nous permettrons d’apporter clarté et uniformisation dans l’exécution des procédures non seulement pour l’entreprise Hutchinson mais aussi pour tous nos clients qui bénéficieront de ces nouvelles fonctionnalités.

#### Organisation du projet

Pour réaliser ce projet nous avons privilégié le cadre Agile qui nous permettait de rester au plus proche de l’attente de nos futurs clients. Par le biais de Constant qui tenait le rôle de Product Owner, nous avons pu ajuster le développement des fonctionnalités au fur et à mesure du projet.  La définition de nos rôles face au cadre Agile était :

* Constant **Product Owner** : donne la vision globale du projet ; est en contact direct avec le client ; réalise des tests de mise en situation réel ; propose des critiques sur le côté fonctionnel.

* Simon **Scrum Master et Manager Opérationnel** : définit la liste des tâches ; traduit les besoin en fonctionnalités ; valide les choix techniques ; aide le développeur en cas de blocage.

* Etienne **Développer et Manager Opérationnel** : définit la liste des tâches ; traduit les besoins en fonctionnalités ; propose des solutions.

Les outils ainsi que la méthodologie de pilotage utilisée sont définis dans la partie Organisation Interne de la présentation de l’entreprise (3.2.1 & 3.2.2).

### Choix techniques et justifications

#### Intégration de la matrice RASCI

Nous avons ajouté la matrice RASCI dans la définition de chaque étape d’une procédure pour affiner la gestion des rôles attribués. Son objectif est de :

* Clarifier la responsabilité au plus petit niveau
* Garantir qu’une personne soit désigné comme responsable de chaque étape
* Permettre une exécution partagée mais contrôlée des procédures

La matrice est appliquée à chaque étape de la procédure, une étape ne peut être complétée que par un utilisateur désigné comme responsable. Les autres rôles sont moins exploités mais pourront à l’avenir avoir des droits spécifiques.

Les rôles disponibles dans cette matrice sont :

* R – Responsable : exécution de l’étape
* A – Accountable : validateur final ou garant du résultat
* S – Support : fournit des informations ou ressources
* C – Consulted : expert à consulter avant l’action
* I – Informed : reçoit des notifications liées à l’étape

En intégrant un outil du Lean management au cœur de notre module nous nous attendons à réduire les ambiguïtés dans la gestion des connaissances pratiques et à aligner les standards de qualité et de conformité de chaque site. C’est aussi un moyen efficace d'encadrer un workflow qui peut être complexe une fois que l’exécution des procédures est autorisée.

#### Utilisation de socket

A l’origine, l’actualisation des états des procédures se faisait via une requête envoyée au serveur avec un intervalle régulier. Ce mécanisme nous a causé certains problèmes notamment en test avec les clients qui ne voyaient pas apparaître immédiatement une procédure à laquelle il devrait pourtant avoir accès. Avec un besoin croissant dans ce module d’interconnectivité nous avons envisagé d’actualiser l’exécution des procédures via le socket client. Son objectif est :

* D’émettre des évènements pour notifier les changements d’états des étapes de procédure
* De diffuser des mises à jour à tous les clients pour s’assurer que l’information affichée est cohérente

Cette technologie permet d’améliorer l’expérience utilisateur grâce à des mises à jour instantanée sans rafraîchir la page ou attendre la fin de l’intervalle entre deux requêtes. C’est aussi une réduction de la charge pour le serveur qui subissait des requêtes répétitives de la part de tous les utilisateurs, même lorsqu’aucun avancement n’avait été fait dans l’exécution d’une procédure. Enfin, cela facilite la collaboration entre utilisateurs puisque cette fonctionnalité nous a permis d’intégrer une vue dédiée à l’avancement de toutes les procédures en cours.

### Méthodologie mise en œuvre

#### 4.2.4.1 Procédure désordonnée et mécanisme de validation

Avant la refonte, les procédures s'exécutaient toujours de manière séquentielle. L’utilisateur devait compléter chaque étape dans l’ordre de la première à la dernière sans pouvoir en laisser une seule pour plus tard. L’idée était d'apporter plus de souplesse dans l’exécution des procédures en permettant de compléter chaque étape dans n’importe quel ordre.

           La modélisation des procédures se fait déjà sous forme de graphiques orientés où chaque nœud est une étape et chaque arc la transition possible vers un autre nœud. Cette modélisation permet de générer des procédures complexes avec plusieurs embranchements et incluant des branches conditionnelles. La validation est alors simple puisqu’il suffit d’avoir un chemin reliant la première étape à la dernière. De plus cette validation n’a pas lieu d’être codée explicitement puisque les contraintes d’une procédure exécutée séquentiellement, c’est-à-dire ne pas pouvoir faire de sauts ??? entre les étapes et être obligé de valider une étape pour passer à la suivante, permettent à elles seules d’être sûr de la validité d’une procédure une fois arrivée sur l’étape de fin.

         L’ajout des procédures désordonnées complique cette logique puisqu’il est possible de compléter l’étape de fin à n’importe quel moment de l’exécution. Il est aussi possible de valider une étape dans une branche conditionnelle avant d’avoir passé le nœud représentant la condition.

           Nous avons donc utilisé l’algorithme de Parcours en Profondeur pour trouver tous les chemins possibles entre le point de départ et le point d'arrivée. Nous avons aussi modifié cet algorithme pour qu’il exclut d’office tous les chemins ne comprenant pas l’étape courante, ce qui nous permet de réduire nos recherches au fur et à mesure de l’avancement de l’exécution.

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Le contenu généré par l’IA peut être incorrect.

Exemple d’exécution d’une procédure

Une fois la liste de chemins possibles trouvée, il nous faut chercher le plus court comprenant un maximum d’étapes valides. En cherchant le chemin le plus court, nous nous assurons que le chemin prédéfini se rapproche le plus de celui que l’utilisateur emprunte actuellement. Si l’étape courante n’est plus dans liste des chemins possibles, c’est que l’utilisateur a changé d'embranchement, il faut alors recalculer les chemins possibles. A voir /identique à la page 21

Ce mécanisme permet à chaque étape de la procédure désordonnée de définir un chemin allant de la première à la dernière étape. Il est alors possible de valider une procédure une fois que toutes les étapes de ce chemin sont validées. Cela permet également d’ajouter de nouvelles fonctionnalités comme un fil d'Ariane, une liste des étapes et un pourcentage de complétion. Ces fonctionnalités sont fortement liées au chemin qu'emprunte l’utilisateur et sont amenées à être modifiées au fil de l’exécution de la procédure. En effet, en choisissant un chemin conditionnel le pourcentage de complétion diminue et la liste des étapes s’allonge.

#### 4.2.4.2 Amélioration des outils de gestions des procédures

Avant la refonte, les procédures en cours d’exécution étaient affichées dans le panneau descriptif de l’entité procédure ainsi que dans la page d’accueil de l’utilisateur. Ce mécanisme ne permettait pas une gestion des exécutions par projet et pouvait même entraîner des surcharges d’éléments dans l’interface.

Pour pallier ces problèmes nous avons créé une nouvelle entité dédiée à la gestion des procédures. Celle-ci permet d'affilier une liste de procédure pour ainsi suivre l'exécution de toutes les procédures lancées depuis cette entité.

|  |  |
| --- | --- |
| Une image contenant texte, capture d’écran, logiciel, Page web  Le contenu généré par l’IA peut être incorrect. | Une image contenant texte, Appareils électroniques, capture d’écran, logiciel  Le contenu généré par l’IA peut être incorrect. |

Gestion des instantes de procédures avec et sans gestionnaire de procédure

Le gestionnaire de procédures permet de segmenter les différentes instances de chaque procédures. En effet, dans l’onglet procédures en cours, toutes les instances de cette procédure qui sont lancées simultanément sont affichées. Bien que chaque instance dispose d’un nom, il devient vite compliqué de s’y retrouver. En revanche, dans les processus associés, seules les instances créées depuis cette interface apparaissent.

De plus, la mise à jour par socket permet de synchroniser l’état des procédures en temps réel assurant la cohérence des informations et une vue d’ensemble plus sécurisée.

Enfin des statuts ont été ajoutés, ils permettent d’identifier en un coup d’œil les éventuels blocages sur une étape. Couplés à la barre de progression de la procédure, ils permettent aux managers d’être plus réactifs vis-à-vis de leurs équipes.

#### Intégration de l’héritage dans les procédures

La mise en place d’héritage dans les procédures permet leur standardisation et leur adaptation. L’entreprise peut définir des procédures modèles et autoriser leur adaptation dans un contexte local. C’est aussi un moyen de diminuer le nombre de doublons facilitant ainsi la maintenance de ces procédés standardisés.

En pratique une procédure modèle ne diffère en rien d’une procédure classique. L’utilisateur devra créer une suite d'étapes et de liens pour les relier. La relation parent/enfant se crée lorsque cette procédure est copiée à partir d’un modèle. L’entité qui sera créé à la suite de cet évènement sera légèrement différente puisqu’elle bénéficie :

* D’éléments graphiques rappelant la provenance de chaque étape, locale ou modèle
* D’affichages fantomatiques de la procédure modèle lorsque des étapes sont supprimées
* De boutons supplémentaires permettant le retour à la version d’origine

Une image contenant texte, capture d’écran, diagramme, logiciel

Le contenu généré par l’IA peut être incorrect.

Affichage d’une procédure locale avec héritage

En ce qui concerne la propagation des modifications, elles se font sans action de l’utilisateur lorsqu’une étape n’est pas modifiée. De même lorsqu’une étape est ajoutée dans la procédure modèle, elle sera ajoutée automatiquement si la procédure locale n’a pas modifié les étapes qui l'entourent. En revanche, si ces étapes ont été modifiées, la nouvelle étape apparaîtra de manière fantomatique dans la procédure locale. Enfin, lorsqu'une étape est modifiée dans la procédure locale et modèle, l'utilisateur peut choisir d’intégrer la nouvelle version de l’étape.

           Cette gestion augmente la flexibilité dans la mesure où les sites de production peuvent apporter leurs modifications sans compromettre la standardisation globale. Le coup en maintenance est aussi moindre puisque les mises à jour sont, autant que possible, propagées automatiquement.

#### Problèmes rencontrés

Dans la première phase de développement, nous nous sommes heurtés à des problèmes de complexité dans la gestion des types de procédures. En effet, nous utilisons 4 types de procédures ayant chacun des règles de navigation et de validation d'étapes différentes. A ce stade l’organisation du code ne permettait pas de gérer cette diversité, ce qui entrainait des incohérences dans l’exécution et la validation. Les 4 types de procédures utilisés sont :

* Ordonnée sans statut
* Ordonnée avec statut
* Désordonnée sans statut
* Désordonnée avec statut

Les étapes ont alors fait face à des problèmes de validation. Les mécanismes de navigation et de validation étaient tout d’abord partagés et s'adaptaient mal aux différents types de procédures. Les sauvegardes et les statuts ne s'appliquaient pas au même moment, d’autant plus qu’en fonction du type de procédure ces mécanismes sont parfois automatiques. Ainsi, chaque ajout demandait trop d’attention pour ne pas enrayer le mécanisme d’un autre type de procédure.

Le projet a donc souffert de l’apparition de nombreux bugs dûs à l’incohérence dans la gestion des différents types de procédures, ce qui a affecté la fiabilité du module.  Ces problèmes ont causé des retards au regard de la complexité de développement et de la résolution de bugs. Ces problèmes ont aussi impacté la satisfaction et la productivité des utilisateurs finaux en phase de test. Pour répondre à ce problème, nous avons revu l’architecture de notre module en veillant à bien séparer le code liée à chaque type de procédure. Le mécanisme de validation d’une étape a lui aussi été revu pour être en harmonie avec la gestion des statuts. Enfin, nous nous sommes concentrés sur la formation des utilisateurs en interne qui devait interagir avec un système bien plus complexe et l’expliquer au client.

### Analyse critique et réflexive

#### 4.2.4.1 Points forts de la mission

Cette mission a un intérêt stratégique important puisqu’elle consiste à adapter un module clef du logiciel dans le cadre d’une négociation commerciale avec un client. Apport qui pourra s’étendre à l’ensemble de nos clients et créer un avantage concurrentiel durable. C’est aussi l’opportunité d’aligner notre application sur les standards industriels.

Du fait de l’introduction de concept avancé, avec la théorie des graphiques pour modéliser les procédures et la mise en place d’un algorithme de parcours en profondeur pour détecter les chemins valides en temps réel, ainsi nous avons apporté une détection dynamique des embranchements conditionnels améliorant la robustesse du système.

Nous avons aussi amélioré l’UX et rendu les exécutions plus traçables via des statuts d’étape, indicateur de progression et fil d’Arianne. Nous avons aussi facilité la gestion des exécutions permettant à un manager de suivre l’avancement de ses équipes en un coup d’œil. Avec cela la synchronisation en temps réel apporte un réel plus, indispensable pour une expérience utilisateur fluide. Enfin le système de visualisation fantôme fonctionne bien pour permettre de toujours visualiser l’héritage dans l’éditeur de procédure.

#### 4.3.4.2 Axes d’amélioration et leçon apprises

Nous avons fait face à une complexité grandissante du module (à cause)en raison des différents types de procédures et des logiques de validation historiquement partagées. Ces problèmes sont dûs à un manque d’anticipation vis-à-vis de l’architecture du module, celui-ci n’était/n’étant pas conçu pour gérer différents modes de navigation et de validation. Ainsi, la nécessité de restructurer en partie le module, à entraîner un retard non négligeable dans le développement.

A l’avenir, il nous faudra porter plus d’attention à la structure du code, notamment lorsque plusieurs workflow se croisent. Mais aussi isoler les responsabilités, c’est-à-dire définir clairement les parties de code qui seront utilisées par tous les types de procédures et celles qui seront spécifiques à un type.

D’un autre côté, les feedbacks réguliers via les tests de notre Product Owner nous ont montré l’importance des mises au point permis par la méthode Agile. Ceux-ci ont permis de faire des ajustements dans la manière dont on avait d’envisager la solution.

### Liens avec la formation

|  |  |
| --- | --- |
| Liens avec la formation M2I | |
| Référentiel | **Compétence éprouvé** |
| Bloc 1 – Analyse stratégique du SI | Identification des faiblesses structurelles et incohérences fonctionnelles |
| Intégration des pratiques du Lean management |
| Bloc 2 – Conception et Déploiement | Compréhension du besoin du client |
| Utilisation de la théorie des graphs |
| Développement de fonctionnalité (héritage, synchronisation) |
| Réorganisation structurelle du module |
| Bloc 3 – Pilotage de projets complexes | Collaboration avec plusieurs acteurs, PO et testeurs |
| Projet Agile, interaction et adaptation des livrables |
| Bloc 4 – Intégration de technologies complexes | Utilisation des sockets pour la mise à jour en temps réel |
| Utilisation d’algorithme de parcours de graph modifié |
| Gestion de l’héritage des procédures |
| Bloc 5 – Veille, documentation, communication de projets | Accompagnement et formation des équipes aux nouvelles fonctionnalités |

1. Wikipédia. (n.d.). *Hutchinson (entreprise)*. Wikipédia <https://fr.wikipedia.org/wiki/Hutchinson_(entreprise)> [↑](#footnote-ref-1)
2. France Hydrogène. (n.d.). *Hutchinson – Annuaire des acteurs*. <https://vighy.france-hydrogene.org/annuaire-des-acteurs/hutchinson> [↑](#footnote-ref-2)