



PIE SD-004 :

Optimisation des plannings de maintenance de l'armée de l'air

Auteurs : Charlotte CRAMPON, Lhoussine EL ALAOUI, Arthur LELIEVRE, Théo LOUISET, Robin MISSIRIAN, Marion ROBERT

Contacts entreprise : Pierre BAZOT, Nicolas DUPIN, Rémy PARIZE

Référent ISAE : Olga BATTAIA

Coach projet : Francis DOURNES

Rapport V1
13/03/2017

Mots clés : planning, maintenance, algorithme glouton, démarche agile

Résumé :

L'objectif de ce PIE est la création d'un outil de planification pour une flotte de Mirages 2000. Ce planning doit tenir compte d'exigences très spécifiques liées aux particularités des cycles de maintenances et de missions d'une flotte d'avions de chasse. Si l'outil sera ultimement utilisé par l'Armée de l'Air, notre client était la Direction Générale de l'Armement, qui servait donc d'intermédiaire pour définir et transmettre les besoins à satisfaire.

Le projet s'est articulé autour de deux axes de travail complémentaires : la gestion de projet et la création de l'outil. Le livrable final est un outil VBA/Python, qui réalise un planning optimisé de l'utilisation de la flotte en fonction de contraintes initiales. Cet outil sera accompagné d'une documentation demandée par le client.

Remerciements :

Nous souhaiterions remercier la Direction Générale de l'Armement ainsi que l'Armée de l'Air pour leur implication dans ce projet.

Nous remercions tout d'abord la DGA, qui s'est déplacée à Toulouse de nombreuses fois pour des réunions avec nous. Vous avez su être très présents et réactifs tout au long du projet, nous permettant d'avancer sereinement dans le projet. Travailler avec vous fût un réel plaisir.

Nous remercions également l'Armée de l'Air, sans qui ce projet n'aurait pas eu lieu. Prendre part à la création d'un outil que vous allez effectivement utiliser a donné un sens à notre travail. Nous avons fortement apprécié votre accueil de deux d'entre nous à Mérignac, ainsi que votre déplacement à Toulouse pour la réunion de clôture. Cette réunion a mis un beau point final à notre collaboration.

Charlotte CRAMPON, Lhoussine EL ALAOUI,
Arthur LELIEVRE, Théo LOUISET,
Robin MISSIRIAN, Marion ROBERT

Je, soussigné Marion ROBERT, certifie sur l'honneur :

1. Que les résultats décrits dans ce rapport sont l'aboutissement de notre travail.
2. Que nous sommes l'auteur de ce rapport.
3. Que nous n'avons pas utilisé des sources ou résultats tiers sans clairement les citer et les référencer selon les règles bibliographiques préconisées.

A handwritten signature in blue ink, consisting of a stylized loop followed by a long horizontal stroke.

Sommaire

I.	Introduction	7
II.	Gestion de projet	8
1.	Organisation de l'équipe	8
2.	Communication	8
2.1.	Communication au sein de l'équipe.....	8
2.2.	Spécificité de la communication avec le client.....	9
3.	Un processus de développement simple.....	10
4.	Répartition des tâches.....	10
5.	Suivi du projet	13
5.1.	Une méthode simple	13
5.2.	Utilisation de l'outil Microsoft Project	13
6.	Analyse des risques et retour sur expérience	15
III.	Contenu technique.....	16
1.	Contexte	16
2.	Livrables.....	16
3.	Règles métier.....	16
3.1.	Règles générales.....	17
3.2.	L'OPEX et les missions métropole.....	17
3.3.	Les maintenances	17
3.4.	Stockage, Blocage.....	18
3.5.	Vols d'entraînement en métropole.....	18
3.6.	Avions hors d'usage.....	19
4.	Exigences	19
4.1.	Pièces fournies	19
4.2.	Exigences utilisateurs.....	20
4.3.	Exigences Système.....	23
5.	Algorithme	23
4.	Structure	27
5.1.	Structure des différents fichiers	27
5.2.	Structure du code Python	27
5.	Gestion des modifications	31
6.1.	Intégration progressive des différentes affectations.....	31

6.2. Détail des modifications	31
6. Outil final et utilisation	33
7.1. Forme de l'outil pour l'utilisateur.....	33
7.2. Première étape : modification des paramètres	33
7.3. Deuxième étape : Lancement de l'élaboration du planning.....	34
7.4. Troisième étape : Visualisation du planning.....	35
7.5. Quatrième étape : Rebouclage	36
IV. Conclusion	38

I. Introduction

Le Projet Ingénierie et Entrepreneuriat sur lequel nous avons œuvré visait à la création d'un outil de planification pour l'utilisation d'une flotte d'avions de chasse. Il a, six mois durant, nécessité un travail autonome et soutenu. Le présent rapport rend compte du déroulement du projet, de sa gestion théorique et effective, des liens avec notre client et des difficultés rencontrées.

L'Armée de l'Air utilise actuellement une méthode simplifiée pour planifier l'utilisation de sa flotte de Mirages 2000. Ainsi, les maintenances, missions à l'étranger (OPEX) et autres entraînements résultent d'un traitement manuel par un opérateur. Ce mode de fonctionnement, en plus d'être chronophage, se révèle non-optimal.

C'est dans ce cadre que la Direction Générale de l'Armement (DGA) nous a proposé de produire un outil facilitant le procédé sus-décrit. La difficulté du projet est dès lors double. Il s'agit d'abord de comprendre avec exactitude les besoins de l'Armée de l'Air afin de fournir un outil répondant aux contraintes opérationnelles réelles. Par ailleurs, ces besoins doivent être définis par l'intermédiaire de la DGA, c'est-à-dire au travers d'une double traduction.

Cette définition des besoins est une première étape fondamentale et continue ; elle ne constitue néanmoins qu'une condition nécessaire à la bonne réussite du projet.

La particularité de ce projet réside dans le fait que nous avons eu plusieurs livrables à rendre au cours du projet. Le premier livrable correspondait à une version dite d'avant-projet qui a joué le rôle de support de discussion avec l'Armée de l'Air. Puis, les réitérations avec le client ont conduit à d'autres versions jusqu'à la version finale.

Le développement du projet s'est articulé autour de deux principaux axes de travail.

Le premier axe regroupe tous les éléments relatifs à la gestion de projet. Elle nous a permis de percevoir la difficulté que rencontre les sous-traitants confrontés aux attentes d'un client. En outre, les attentes vis-à-vis de cette partie nous ont poussés à mettre en place un système d'information, de communication et d'organisation interne efficace et systématisé. Elle nous a également permis d'utiliser des outils comme Microsoft Project dans le cadre d'un projet réel. Le second axe est relatif à la création de l'outil en lui-même, *id est* à la définition des algorithmes à utiliser et à l'écriture du programme en lui-même. Ce système tient compte de contraintes initiales fixées par un fichier Excel, réalise le planning optimisé via un code en Python et le communique à l'opérateur via le même fichier Excel. Un algorithme glouton, qui détermine le planning pas de temps par pas de temps, a été utilisé.

La contribution à ce projet s'est révélée, collectivement et individuellement, très gratifiante. Prendre part à la création d'un outil qui sera probablement effectivement utilisé par l'Armée de l'Air, et pourra être adapté à d'autres types d'avions, a donné un sens à notre travail.

II. Gestion de projet

1. Organisation de l'équipe

Dans un premier temps, nous avons déterminé au sein de l'équipe des rôles indispensables au bon déroulement du projet :

Chef de projet et relation client	Marion ROBERT
Secrétaire	Charlotte CRAMPON
Gestionnaire du Système d'Information	Robin MISSIRIAN
Codeurs	Lhoussine EL ALAOUI Arthur LELIEVRE Théo LOUISET

La chef de projet s'est chargée de conduire le projet de bout en bout. Elle a assumé la responsabilité des différentes phases, depuis la définition des exigences, la traduction des besoins utilisateurs en spécifications fonctionnelles et techniques, jusqu'au développement de l'outil. Elle a animé l'équipe afin que les livrables répondent à la demande et soient délivrés à temps au client. Par ailleurs, elle s'occupait des relations avec le client, veillant à maintenir un contact régulier.

La secrétaire avait pour rôle de mettre en commun les notes prises par les membres de l'équipe pendant les réunions et d'en fournir un compte-rendu complet. Ces compte-rendus sont disponibles sur le Système d'Information et cela permet à un membre absent à la réunion de prendre connaissance de ce qui a été dit, mais surtout de tracer chaque réunion. La secrétaire s'est également chargée de rédiger et mettre à jour les versions successives des exigences, ainsi que le document de *Gestion des modifications*, document qui a permis de tracer les modifications apportées dans chaque version.

Le Gestionnaire du Système d'Information a mis en place le Système d'Information et s'est assuré de son maintien à jour tout au long du projet.

Les codeurs sont les membres de l'équipe qui ont travaillé au code Python ou à l'Excel. Chaque personne ayant sa propre manière de coder, il a paru nécessaire de faire le choix de trois personnes pour ces rôles, deux sur le Python, une sur l'Excel. En effet, si nous avions été six à travailler sur le code, cela aurait pu pour conséquence de complexifier l'élaboration du code.

Un des élèves codeur a également établi un contact direct avec le client, permettant de traiter de détails techniques directement.

2. Communication

2.1. Communication au sein de l'équipe

La communication au sein de l'équipe s'est faite par plusieurs moyens. Chaque lundi, nous nous rencontrions durant 10 minutes afin de faire un point sur l'avancement du projet. Cette régularité nous a permis d'anticiper les problèmes, et de les traiter à la source. Par

ailleurs, nous avons tenu des réunions de 2 heures toutes les trois semaines, ou sur demande d'un des membres de l'équipe si le besoin se faisait sentir. Ces réunions ont permis de rentrer plus en profondeur que lors des points hebdomadaires. Ces deux types de réunions ont été très complémentaires.

En plus de ces réunions, nous avons mis en place une discussion avec tous les membres de l'équipe sur la plateforme collaborative Facebook. Cette discussion a été le lieu d'échanges rapides d'informations, permettant de débloquer les petits problèmes et de ne pas ralentir le travail. Par exemple, la secrétaire, qui s'occupait de mettre à jour la gestion des modifications, n'était pas certaine qu'une modification particulière ait été apportée pour la V2.1., elle a donc demandé confirmation sur la discussion, et a eu une réponse très rapidement, lui permettant de continuer son travail sans attendre le point du lundi.

Le Gestionnaire du Système d'Information a mis en parallèle un Google Drive où sont disponibles les documents suivants :

- Compte-rendus de réunion
- Exigences, spécifications et risques
- Plan de développement
- Plan de management
- Charte relationnelle
- Contacts
- Structure du code et notice explicative
- Divers fichiers envoyés par le client

Ces documents sont maintenus à jour par le Gestionnaire du Système d'Information, et sont disponibles pour consultation par tout membre de l'équipe.

2.2. Spécificité de la communication avec le client

Le client pour lequel nous devons élaborer l'outil est l'Armée de l'Air. Néanmoins, nous n'avons initialement pas été en contact direct avec l'Armée de l'Air. La spécificité de notre projet réside dans le fait que la DGA a joué le rôle d'interlocuteur entre l'Armée de l'Air et nous-mêmes. Ainsi, dans la première phase du projet (lors de la définition des exigences et des spécifications), notre interlocuteur était la DGA et non l'Armée de l'Air.

Néanmoins, une fois les exigences et les spécifications définies pour la première version, un contact direct a été établi avec l'Armée de l'Air. La première version V1.1 a été livrée à l'Armée de l'Air à Mérignac lors d'un déplacement de la DGA avec deux membres de l'équipe. Cela a permis aux élèves de montrer comment utiliser l'outil, et d'avoir les retours de l'Armée de l'Air pour un premier rebouclage. Par la suite, un des élèves codeurs est resté en contact direct avec l'Armée de l'Air, mais la DGA est restée prête à intervenir si le besoin s'était fait sentir.

Par ailleurs, au sein de la DGA nous avons également plusieurs interlocuteurs. Ces différents interlocuteurs se sont relayés pour venir aux réunions à Toulouse. Nous craignons

que cela pose un problème mais finalement, ce ne fut pas un frein. Ils communiquaient entre eux, ce qui a assuré une bonne continuité et cohérence dans le projet.

3. Un processus de développement simple

Le projet présente une autre particularité, celle de demander des livrables intermédiaires. En effet, il ne nous était pas demandé seulement une version finale livrable à la fin du projet, mais plusieurs versions intermédiaires. Nous avons ainsi adopté une démarche agile.

Le projet est constitué d'une première partie avant-projet, à l'issue de laquelle est fournie la version V1.1. Il s'agit d'une première solution simple, et qui fonctionne. Cette première version a servi de support à la discussion avec l'Armée de l'Air, afin de déterminer si les exigences prises en compte étaient les bonnes. Cela a permis un premier rebouclage qui a conduit à la version V1.2. Il y a de nouveau eu réitération avec le client, ce qui a conduit à la V2.1 puis à la version finale V2.2. Nous avons géré les versions à l'aide du document *Gestion des modifications*. Cela nous a permis de tracer les modifications, mais cela pourrait également être utile ultérieurement si le client souhaite finalement revenir à une version précédente.

Les échéances sont les suivantes :

V1.1	14/12/2016
V1.2	20/01/2017
V2.1	08/02/2017
V2.2	24/03/2017

4. Répartition des tâches

Nous avons découpé le projet en différentes tâches, avec un nombre d'heures associées, et une date limite.

Tâche	Nombre d'heures allouées	Date limite
Définition du besoin avec le client	20	02/11/2016
Définition des spécifications	20	02/11/2016
Définition des entrées / sorties	20	15/11/2016
Compréhension/Adaptation de l'outil Excel / VBA	20	15/11/2016
Programmation initiale	40	31/11/2016
Déplacement à Mérignac et livraison de la V1.1		14/12/2016
Modifications sur le Python 1.1	20	20/01/2017
Modifications sur l'Excel 1	15	20/01/2017
Livraison de la V1.2		20/01/2017
Modifications sur le Python 1.2	20	31/01/2017
Modifications sur l'Excel 1.2	15	31/01/2017
Livraison de la V2.1		10/02/2017
Modifications sur le Python 2.1	25	24/03/2017
Modifications sur l'Excel 2.1	15	24/03/2017
Rédaction d'un tutoriel	15	24/03/2017
Développement d'indicateurs	15	24/03/2017
Travail sur l'ajout de graphes pour l'analyse de données	15	24/03/2017
Livraison de la V2.2		24/03/2017
Gestion de projet	25	
Rédaction du rapport et préparation de la soutenance	20	13/03/2016
Soutenance		13/03/2016

Figure 1.4.1 : Tâches, nombre d'heures allouées et échéance

Une fois les tâches définies, nous avons affecté les différentes tâches entre les personnes de l'équipe, en cherchant à avoir une répartition équitable.

Le tableau ci-dessous est une matrice RACI, elle indique les rôles et les responsabilités des membres de l'équipe et du client pour chaque tâche. La matrice est remplie par les lettres « R », « A », « C » ou « I », qui signifient :

R (Responsible) : le ou les R réalisent l'action

A (Accountable) : le A est celui qui doit rendre des comptes sur l'avancement de l'action

C (Consulted) : les C sont les entités qui doivent être consultées

I (Informed) : les I sont les entités qui doivent être informés

	Charlotte	Théo	Lhoussine	Robin	Arthur	Marion	Client
Définition du besoin avec le client	I	I	I		I	R,A	C
Définition des exigences et spécifications	R,A	I	I		I	C	C
Définition des entrées / sorties		R,A	C		I		
Adaptation de l'outil Excel / VBA		R,A	C		I		
Programmation initiale		C	R,A		R		
Modifications sur le Python et sur l'Excel	I	R,A	R,A		R		C
Rédaction d'un tutoriel	R	C	C	R		R,A	
Développement d'indicateurs	R,A	C					C
Travail sur l'ajout de graphes				R,A			
Gestion de projet (préparation des revues projet, etc)	R			R		R,A	

Figure 1.4.2 : Matrice RACI

Lors de nos revues hebdomadaires, nous faisons le point sur le nombre d'heures travaillées dans la semaine précédente, et les écarts éventuels qu'il y avait avec les heures de travail estimées. Nous avons plutôt bien prévu le nombre d'heures de travail, les codeurs ayant cependant dû fournir un travail plus important dues aux itérations sur les versions. Nous obtenons le volume horaire par étudiant suivant :

Lhoussine	135,5
Charlotte	105,5
Théo	133
Marion	110,5
Robin	100,5
Arthur	108

Figure 1.4.3 : Volume horaire total de travail en heures par personne

Les créneaux alloués pour le PIE dans l'emploi du temps correspondant à 80h, nous avons travaillé en plus afin de réaliser le travail.

5. Suivi du projet

5.1. Une méthode simple

Lors de la première revue de projet, nous avons proposé un tableau de bord pour effectuer le suivi de l'avancement du projet. Cependant, par la suite ce tableau de bord envisagé initialement s'est révélé trop complexe dans notre cas. La particularité évoquée ci-dessus (un avancement par itérations) nous a conduits à faire le choix d'une méthode plus simple.

En effet, pour chaque version, nous avons estimé le temps de travail nécessaire, puis nous avons convenu avec le client d'une échéance réaliste pour laquelle envoyer le livrable. Lors de nos revues hebdomadaires, nous faisons le point sur l'avancée du travail. Puisque nous avons correctement estimé le temps de travail nécessaire, nous n'avons pas eu de mauvaise surprise et les livrables ont toujours été rendus dans les temps.

5.2. Utilisation de l'outil Microsoft Project

Pour avoir une vision rétrospective et prospective des tâches réalisées ou à réaliser, nous avons utilisé l'outil Microsoft Project. Néanmoins, nous estimons que nous aurions pu l'utiliser encore plus tout au long du projet. Nous avons compris l'utilité de cet outil pour la planification que trop tardivement et l'avons essentiellement perçu comme un indicateur de suivi. Cela s'explique par le fait qu'une méthode de suivi d'avancement simple suffisait, comme nous l'avons vu dans la partie précédente.

Le diagramme de Gantt permet un aperçu global de ces deux visions. Il nous a notamment servi pour la gestion du projet en tant que telle, fournissant un diagnostic efficace de notre situation. Le diagramme met en évidence certains caractères propres à notre projet. Par exemple, sa forme en escaliers relativement pentus résulte directement de la nature itérative des tâches. Chaque rencontre avec la DGA est suivie d'objectifs avec une deadline, ce qui se traduit par des marches. L'avancement de tout ce qui n'est pas l'écriture du programme est alors mis en suspens, conditionnel à la complétion des tâches d'écriture.

Pour une utilisation future de l'outil, nous tirerions meilleur parti de la vision prospective qu'il apporte. Comme nous l'avons vu précédemment, pour chaque version nous avons mené une évaluation horaire du travail confié, vérifié que ce volume est dans les limites de nos capacités puis fixé l'échéance avec le client. Nous avons fait ce travail, à la main, et nous aurions pu pour cela utiliser le logiciel.

L'étape cruciale est ici l'évaluation du volume horaire et la compatibilité avec les autres contraintes du projet. Dans le cas où une ressource (homme ou matériel) est affectée à plusieurs tâches, il s'agit de vérifier que les attentes du client pour la prochaine deadline ne mettent pas en péril d'autres tâches, ou font prendre un retard trop conséquent.

Ci-dessous le diagramme de Gantt tel qu'il était le 26 janvier :

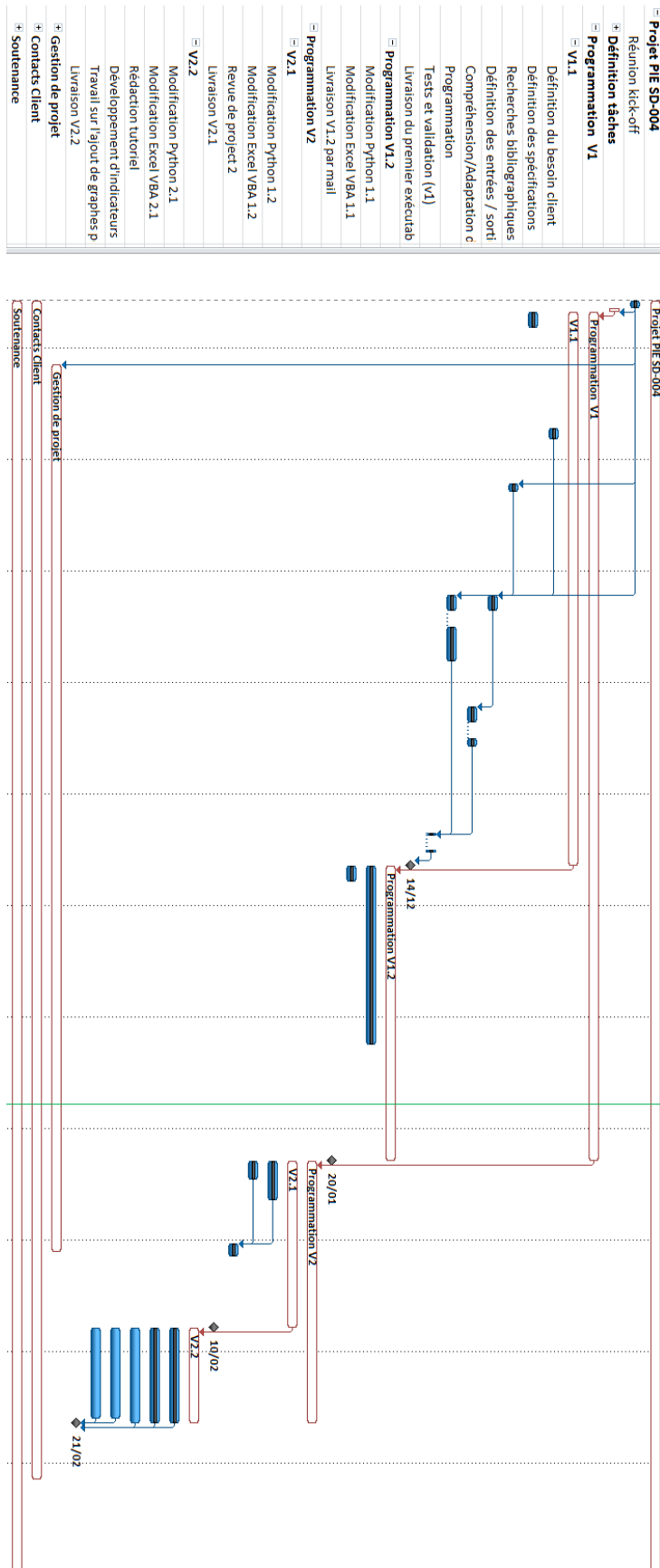


Figure 1.5.1 : Diagramme de Gantt au 26/01/17

6. Analyse des risques et retour sur expérience

Dans la première phase du projet, nous avons procédé à une analyse de risques. Les risques sont associés à la particularité de notre projet qui est de fournir un premier livrable dans une phase d'avant-projet, puis de réitérer avec le client pour fournir d'autres livrables jusqu'à aboutir à la version finale. Le tableau ci-dessous rend compte des risques identifiés :

Identification		Probabilité	Impact
RIS01 : Besoin horaire	Les heures de travail nécessaires sont mal évaluées	Faible	Moyen
RIS02 : Exigences utilisateurs	Le système ne peut pas remplir chacune des exigences utilisateurs exprimées au fur et à mesure du projet	Moyenne	Fort
RIS03 : Infaisabilité	Aucune solution ne satisfait toutes les exigences	Faible	Fort

Figure 1.6.1 : Analyse de risques

Premièrement, la présence d'une première phase avant-projet avant les réitérations implique une mauvaise estimation du nombre nécessaires d'heures durant les phases de réitérations. Nous avons géré ce risque en prenant en compte une marge sur les heures de travail nécessaires. Puis, à chaque version nous avons affiné l'estimation du nombre d'heures de travail. Finalement, nous avons eu assez d'heures pour rendre chaque livrable à temps (en cumulant les créneaux PIE et des heures de travail personnel).

Par ailleurs, l'absence d'une liste initiale exhaustive des exigences utilisateurs et le fait que les exigences évoluent au fur et à mesure du projet impliquent deux risques : le risque que le système ne remplisse pas chaque exigence, et le risque que aucune solution finale ne satisfasse toutes les exigences. En effet, lors de la phase d'avant-projet et de l'élaboration de la première version V1.1, les codeurs ont du faire des choix déterminants pour la suite. Nous avons donc essayé de concevoir un outil le plus adaptable possible, et nous avons consacré beaucoup de temps à la définition des exigences / spécifications avec le client. Ce fût l'objet de plusieurs réunions ainsi que de nombreux appels téléphoniques avec la DGA. A l'issue de cela, nous avons formalisé et rédigé les exigences comme nous les avons comprises à l'écrit, puis nous avons envoyé ce document à la DGA pour validation. Ceci nous a permis de nous assurer d'une bonne compréhension, et de limiter au maximum les risques associés. Ce travail a été bénéfique, car la solution finale satisfait les exigences utilisateurs.

III. Contenu technique

1. Contexte

Comme nous l'avons vu précédemment, le projet consiste à proposer un outil de planification pour la flotte d'avions de chasse de l'Armée de l'Air (Mirage 2000). A l'issue d'un précédent stage, une solution a déjà été proposée mais elle utilise un logiciel à licence payante. L'objectif du projet était alors de proposer une solution simple, sur un logiciel à licence gratuite.

Rappelons le contexte du projet. Le client final et utilisateur de l'outil est l'Armée de l'Air et l'outil doit donc répondre à leurs besoins. Néanmoins, ces besoins sont définis par l'intermédiaire de la DGA, notre client. Pour cela, nous avons livré une première version d'avant-projet V1.1, qui a été le support de discussions avec l'Armée de l'Air. Dans cette version 1.1 est définie la structure globale du code. Puis, s'en suivent plusieurs réitérations aboutissant à de nouvelles versions, jusqu'à aboutir à la version finale V2.2.

L'élaboration de l'outil été guidée par les méthodes utilisées en gestion de projet. En effet, la première phase du projet a consisté à passer en revue les règles métier afin de définir les exigences et les spécifications avec le client. Cette étape nécessaire et fondamentale a été le point de départ de l'écriture du code de la première version V1.1. Cette version demandait l'élaboration d'un code adaptable. Puis, des modifications ont été apportées pour aboutir aux versions V1.2, V2.1 et V2.2. Les outils de communication mis en place ont alors été indispensables pour que les codeurs soient en adéquation avec la demande du client.

2. Livrables

Les livrables demandés pour le 23/03 sont les suivants :

- la version finale de l'outil, soit la V2.2
- le code commenté : le code doit être clair et comporter des commentaires complets. Si le code est repris à l'avenir, il doit être compris facilement
- la documentation ingénieur : elle comprend les algorithmes détaillés qui ont abouti au code. L'outil doit pouvoir être modifié facilement.
- la documentation utilisateur : elle est à destination des utilisateurs et décrit comment utiliser l'outil pas à pas. Elle doit être détaillée et est illustrée de captures d'écran.
- des indicateurs : un travail de réflexion est attendu sur la définition d'indicateurs pertinents pour mesurer une qualité de planning

3. Règles métier

Comme nous l'avons vu, la première phase du projet a consisté à définir les exigences et les spécifications de l'outil. Le but de l'outil étant de fournir une aide à la planification des activités, il a donc été indispensable de connaître les Règles Métier. L'élaboration de la liste des règles a été le sujet de nombreuses discussions avec le client.

Les règles sont nombreuses et parfois en opposition. Ce paragraphe rappelle les règles à respecter.

3.1. Règles générales

Pour un avion, sept affectations sont possibles :

- une mission OPEX (à l'étranger)
- une mission métropole
- une maintenance
- un stockage
- un blocage
- un « HS » : l'avion est hors d'usage
- les vols d'entraînement

La priorité est donnée selon l'ordre suivant :

0. (les choix imposés par l'utilisateur lors du rebouclage)
1. le besoin opérationnel (OPEX et métropole)
2. les visites de maintenance
3. le stockage, blocage et HS (qui, dans notre cas, sont imposés par l'utilisateur)
4. les vols entraînement

3.2. L'OPEX et les missions métropole

- Une mission est définie par :
 - Une date de début
 - Une date de fin
 - Un type d'avion, avec des options
 - Un nombre d'avion
 - Un potentiel en heure de vol par avion par mois
- Un avion reste au moins 2 mois en mission, au plus 4 mois.
- Le choix des avions à envoyer en mission se fait selon les critères suivants :
 - 1^{er} critère : ratio POT HORAIRE / POT CALENDRAIRE (cravate)
 - 2^{ème} critère : potentiel horaire max
 - 3^{ème} critère : celui qui a le numéro le plus petit

3.3. Les maintenances

- On considère qu'un avion possède une liste de butée : B1 = calendaire, B2 = horaire, etc...
- Les visites liées aux butées calendaires et horaires :
 - V5 : à 5 ans. Durée : 140 jours calendaires, chez l'industriel
 - V10 : à 10 ans. Durée : 220 jours calendaires, chez l'industriel
 - V15 : à 15 ans. Durée : 185 jours calendaires, chez l'industriel

- V20 : à 20 ans. Durée : 220 jours calendaires, chez l'industriel
- V25 : à 25 ans. Durée : 140 jours calendaires, chez l'industriel
- V30 : à 30 ans. Durée : 220 jours calendaires, chez l'industriel
- On déclenche une maintenance quand on atteint pile la butée horaire / calendaire. On essaie par ailleurs de les mettre en stockage pour allonger la butée calendaire.
- Pour la dernière version finale, nous souhaiterions que les maintenances soient « lissées » : pour éviter des fluctuations trop importantes du nombre d'avions envoyés en maintenance, un avion pourra être envoyé en maintenance « préventivement », avant qu'il ait atteint sa butée.
- Le fretting (pose de bagues sur les cellules pour mieux suivre leurs vieillissements) dure 2 mois.

3.4. Stockage, Blocage

- Le stockage est de 2 ans maximum entre (V0,V10), entre (V10,V20) et (V20,V30), à répartir comme on le souhaite.
- En théorie, pour déterminer le stockage :
 1. On considère le besoin pilote mensuel.
 2. On le compare aux heures fournies par les avions (18h/mois environ par avion)
 3. On obtient le nombre d'avions en trop : c'est le nombre d'avions qui sont à mettre en stockage
 4. Pour choisir quels avions on met en stockage :
 - le système choisit ceux qui ont un ratio $\frac{\text{butée horaire}}{\text{butée calendaire}}$ faible (un tel avion a trop consommé d'heures par rapport à sa butée calendaire, on cherche donc à reculer son envoi en maintenance).
 - On regarde également leurs situations précédentes : on essaie de minimiser les allers retours inutiles vers le stockage
- Dans notre cas, la décision de mettre un avion en stockage sera uniquement prise par l'utilisateur.
- Un blocage est un stockage qui n'empêche pas le vieillissement de l'avion : ses potentiels horaires et calendaires continueront d'être consommés.

3.5. Vols d'entraînement en métropole

- Il s'agit d'un nombre d'heure de vol à faire par mois pour maintenir les pilotes entraînés
- Un avion affecté nulle part vole : il est en vol d'entraînement en métropole un nombre d'heures par mois. Cette valeur est constante dans les premières versions puis est entrée en paramètre par l'utilisateur.

3.6. Avions hors d'usage

L'utilisateur peut décider de mettre un avion « HS » hors d'usage, pour des cas de pannes par exemple.

4. Exigences

Le paragraphe suivant expose les exigences utilisateurs et exigences systèmes (spécifications) que nous avons construites au cours du projet, et ce notamment à partir des Règles Métier explicitées dans le paragraphe précédent. La version présentée correspond à la version V2.1 de l'outil.

Similairement à la construction de notre outil, les exigences se sont construites par itération. A chaque nouvelle version du code demandée par le client, les exigences étaient mises à jour. Chaque nouvelle version était en effet due à :

- une évolution des exigences et des spécifications due à la meilleure compréhension des Règles Métier
- l'apparition de nouveaux besoins exprimés par le client

Il y a ainsi 4 versions successives des exigences, une pour chaque version de l'outil. Chacune de ces versions a par ailleurs subi des révisions successives, après des discussions avec le client. Le but de ces révisions était de se mettre d'accord sur les attentes, pour finalement figer les exigences lors de l'élaboration de chaque version de l'outil.

Les changements entre les différentes versions ne concernent pas tant le nombre d'exigences, mais plutôt leur contenu. Certaines ont été ajoutées, mais la plupart ont été modifiées. Par exemple, les critères de décision ainsi que les types d'affectation possibles ont été étoffés à chaque itération. Certaines exigences ont été abandonnées au cours du projet, toujours après discussion avec le client. Par exemple, il avait été envisagé de rendre le stockage automatique, ce qui a été abandonné par la suite, par soucis de temps, de priorité, et de faisabilité.

Les exigences utilisateurs concernent principalement l'affichage du planning, les critères de décisions pour affecter les avions, ainsi que les performances de l'outils (rapidité, clarté, transposabilité). Les exigences systèmes sont les spécifications que nous avons construites pour répondre aux exigences utilisateurs, en terme de structure de l'outil.

4.1. Pièces fournies

PF01 : Excel actuellement utilisé par l'Armée de l'Air et permettant de traduire le fichier de sortie .csv en fichier excel lisible et compréhensible pour l'utilisateur. Cet Excel présente le planning solution.

PF02 : Excel servant de base de données pour l'armée de l'air qui répertorie l'ensemble de la flotte de l'armée de l'air ainsi que leurs missions, capacités, indices d'usure et tout autre

paramètre utilisé pour la création du planning de la maintenance. Ce fichier permet d'exporter ces informations vers l'algorithme.

4.2. Exigences utilisateurs

EXUT01 Précision : le planning a une précision de 1 mois et une étendue d'au moins 5 ans.

EXUT02 Performance : le système a des performances semblables à celle de l'Excel PF01.

EXUT03 Temps de calcul : le temps de calcul du système en réglage normal ne dépasse pas 5 minutes.

EXUT04 Bibliothèque : l'exécutable est bâti à partir de bibliothèques libres et fonctionne sur un ordinateur PC classique non connecté au réseau.

EXUT05 Transposabilité : le système pourra être transposé des M2000 aux Rafale à l'avenir.

EXUT06 Qualité logiciel : le code utilisé pour bâtir le système est lisible, commenté et modulaire. Pour cela :

- Le code est segmenté en fonctions
- Une description de chaque fonction est présente.
- Chaque fonction ne dépassera pas une dizaine de lignes
- Les paramètres modifiables seront concentrés. Ainsi, le paramétrage implique de ne modifier qu'une partie du code uniquement.

EXUT07 Entrées : le système prend en entrée les données suivantes pour construire un planning solution :

- La situation actuelle de la flotte de l'Armée de l'Air, précisant l'état et les caractéristiques de chaque avion (sous la forme de l'Excel PF02)
- Un planning réunissant les choix imposés par l'utilisateur (EXUT10), où l'utilisateur aura placé des affectations.

EXUT08 Catégorie avion : chaque avion a pour caractéristique la catégorie avion. Cette catégorie est le doublon (type avion ; configuration avion). Le code intègre cette caractéristique sous la forme d'un doublon. Deux avions de même catégorie sont considérés équivalents.

EXUT09 Paramétrage : le système est paramétrable, c'est-à-dire qu'un utilisateur de l'Armée de l'Air pourra effectuer des modifications sur le planning proposé, relancer le programme et obtenir un nouveau planning qui aura pris en compte (sauf impossibilité) les modifications.

EXUT10 Modifications utilisateurs : l'utilisateur peut réaliser les modifications suivantes sur le planning solution :

- mise en maintenance
- affectation à une mission OPEX
- stockage
- hors d'usage (HS)
- vols d'entraînement
- affectation à une mission métropole
- blocage (stockage avec vieillissement)

EXUT11 Décision : le système prend la décision de placer chaque avion dans une des catégories suivantes :

- mise en maintenance
- affectation à une mission OPEX
- stockage
- hors d'usage (HS)
- vols d'entraînement
- affectation à une mission métropole
- blocage (stockage avec vieillissement)

L'affectation à ces différentes activités se fait selon l'ordre de priorité suivant :

- 1) Stockage, HS, BL (imposés par l'utilisateur)
- 2) Missions OPEX et missions métropole (automatisé)
- 3) Maintenance (automatisé)
- 4) Vol métropole (automatisé)

EXUT12 Contraintes sur les activités : le système construit le planning en respectant les contraintes sur les activités suivantes :

- **EXUT12a Besoin opérationnel** :
 - Chaque mission est caractérisée par le vecteur (nombre d'avions nécessaires, type d'avions, options avions nécessaires, nombre d'heures de vol par mois, date de début, date de fin). La caractéristique « options avions nécessaires » se présente sous la forme d'un nombre, chaque nombre représentant une configuration précise.
 - Le nombre minimum d'avions envoyés simultanément vers une mission est de 2.
 - Le temps passé en mission pour un avion est au minimum de 2 mois, au maximum de 4 mois.
 - Un avion doit passer au minimum deux mois en métropole
 - Un avion envoyé en mission a un fort ratio $\frac{\text{butée horaire}}{\text{butée calendaire}}$ et un potentiel horaire supérieur à $\text{nombre de mois en mission} * \text{potentiel horaire par mois} + 100$ (un tel avion n'a pas assez consommé d'heures par rapport à sa butée calendaire).
- **EXUT12b Maintenances** :

- chaque avion subit 6 maintenances différentes et successives qui imposent :
 - une butée calendaire
 - une durée d'immobilisation
 - une capacité (le nombre d'avions envoyés en maintenance a un minimum et un maximum)
- un avion est envoyé en maintenance lorsqu'il atteint exactement sa butée horaire ou calendaire
- **EXUT12c HS, Stockage ou BL** : un avion est mis en HS, Stockage ou BL si l'utilisateur a imposé ce choix lors du rebouclage.
- **EXUT12d Vols entraînement** :
 - Les avions sélectionnés pour être envoyés en vols entraînement sont ceux qui n'ont pas été affectés aux trois autres activités.
 - Un avion en vol entraînement consomme 18h de vol par mois.

EXUT13 Rebouclage utilisateur : lorsque l'utilisateur réalise une modification sur le planning (EXUT10) et relance le système afin d'obtenir un planning solution, on considère que tous les choix réalisés avant la modification sont approuvés et donc fixés.

EXUT14 Format de sortie : le planning a un format proche de l'Excel PF01.

- Le temps est représenté sur l'axe horizontal, la flotte d'avions sur l'axe vertical.
- Plusieurs onglets sont présents : le planning global, puis un planning pour chaque type d'avion.
- Les cases sont remplies de la façon suivante :
 - Maintenance : « VXX »
 - Stockage : « STK »
 - Vol métropole : « Pt » avec Pt le potentiel horaire consommé
 - Mission : « Pt » avec Pt le potentiel horaire consommé, et en commentaire le nom de la mission

EXUT15 Sauvegarde : lors du calcul d'un planning solution N à partir d'un planning N-1, le planning N-1 est sauvegardé.

EXUT16 Impossibilité contraintes : le système envoie un message d'erreur dans le cas où aucune solution vérifiant toutes les contraintes ne serait trouvée. Ce message d'erreur permet à l'utilisateur d'identifier le point bloquant.

EXUT17 Nouvelles exigences : lors de la phase d'itération du projet, l'Armée de l'Air pourra exprimer de nouvelles exigences. Ces exigences seront soumises à l'approbation de la DGA pour être intégrées au système.

EXUT18 Compte-rendu : à la suite d'une réunion entre le groupe de travail et la DGA, un compte-rendu est fourni à la DGA

EXUT19 (additionnelle) Qualité : la qualité du planning sera mesurée à l'aide d'indicateurs. Ces indicateurs peuvent par exemple concerner la distribution des heures de potentiel avion perdues (moyenne, écart type).

EXUT20 (additionnelle) Paramétrage temps de calcul : En plus du paramétrage de EXUT10, l'utilisateur peut influencer sur le paramètre « Temps de calcul » : L'utilisateur peut choisir d'affiner le planning, au détriment du temps d'exécution. L'utilisateur indiquera cet ordre par la modification d'un paramètre, et le temps de calcul ne sera alors pas supérieur à 5 minutes. Cette affinage pourra se faire par la prise en compte de préférences additionnelles.

4.3. Exigences Système

EXSYS01 Structure : le système comprend 3 sous-systèmes :

- Un module VBA de conversion des entrées en plusieurs fichiers .csv : un fichier de paramètres et un fichier de situation initiale
- Un programme de création du planning qui fournit un fichier .csv
- Un module VBA de conversion du fichier .csv en un Excel lisible par l'utilisateur

EXSYS02 Fichiers : le système est segmenté en deux parties pour l'utilisateur : une interface Python/Java et deux fichiers Excel (qui rassemblent les deux modules VBA).

EXSYS03 Structure du code : le code est segmenté en 4 modules, contenant 6 fichiers :

- 1^{er} module : Un fichier « Lecture » qui extrait les informations des fichiers .csv d'entrée
- 2^{ème} module « Entrée » : Deux fichiers « Constantes » et « Objets qui définissent les différents objets et constantes utilisés dans module « Calcul »
- 3^{ème} module « Calcul » : Constitué des fichiers « Algorithme » et « Main », il construit le planning solution.
- 4^{ème} module : Un fichier « Ecriture » qui génère un fichier .csv à partir du planning solution.

EXSYS04 Langage : Le programme d'optimisation est codé sous Python.

Nous venons de voir les exigences prises en compte. Il s'agira maintenant de montrer quelques algorithmes ont permis d'écrire le code, puis la structure même des différents fichiers.

5. Algorithme

Afin d'écrire le code pour créer l'outil, il nous a fallu définir un certain nombre d'algorithmes. Ci-dessous en sont détaillés quelques-uns.

- **Algorithme principal :**

MainProgram(CSV)

lecture donnees du CSV

t0 : pas de temps de départ

Pour t dans [domaine temporel [t0; tlim], t++]

algorithme_affectation_en_mission(mission opex)

algorithme_affectation_en_mission(mission metropole)

algorithme_affectation_en_maintenance

algorithme_affectation_autres //stockage, bockage..

Fin pour

ecriture donnees en CSV

- **Algorithme_affectation_en_mission :**

MainProgram(données)

liste A : liste Avions

liste M : liste Missions

t0 : pas de temps de départ

Pour M dans (liste M)

liste a = besoinEnOpex(M, liste A)

liste fl = choixAvions(liste a)

liste c = affectationAvions(liste fl)

Fin pour

Fonction besoinEnOpex(M, liste A)

Pour chaque A dans (liste A)

Si A vérifie type(M) et potentiel(M)

liste 2 < - ajouter A

Fin si

Fin pour

Retourner liste 2

Fonction choixAvions(liste 2)

n = longueur(liste 2)

Pour chaque A dans (liste 2)

calculer cravate(A)

Fin pour

liste 3 <- liste 2 ordonn e en fonction de cravate(A)

Retourner liste 3

Fonction AffectationAvion(M, liste 3)

nb = nombreAvionsNecessaires(M)

liste 4 = sousliste(liste 3), tronqu e   nb

Pour chaque A dans (liste 4)

Ecrire (Mission M)

Fin pour

- **Algorithme_affectation_en_maintenance :**

MainProgram(donn es)

liste A : liste Avions

liste Mt : liste Maintenances

n1 = nombre d'avions en maintenances

n2 = 0 // nombre des entr es par pas de temps

Pour chaque A dans (liste A)

Si n1 < limite nombre d'avions en maintenances :

si n2 < limite nombre d'entr es en maintenances :

```

        AffectationMaint(A,liste Mt)
    Fin si
Fin si
Fin pour
Si strategie = 2 // lissage
    liste A2= liste avions classÈs par potentiel calendaire
    Pour chaque A dans (liste A)
        Si n1 < limite nombre d'avions en maintenances :
            Si n2 < limite nombre d'entrÈes en maintenances :
                AffectationMaint(A,liste Mt)
            Fin si
        Fin si
    Fin pour

```

```

Fonction AffectationMaint(A,Mt)
    Si potentiel(A) <= potentielMinNecessaire(strategie)
        Pour chaque Mt dans (liste Mt)
            Si Mt = prochaineMaintenance(avion A)
                ecrire(Maintenance Mt)
            Fin si
        Fin pour
    Fin si

```

- **Algorithme_affectation_autres :**

```

MainProgram(données)
    sit_Init = données(matrice_situation_initiale)
    Si sit_Init non vide
        parcourir sit_Init(t) // colonne qui correspond à l'instant t
    Fin si

```

Fin si

Fonction parcourir()

Pour chaque A dans (liste A)

Si sit_Init(t,A)=stockage(STK)

Ecrire (stockage STK)

Si sit_Init(t,A)=blockage(BL)

Écrire (blockage BL)

...

4. Structure

5.1. Structure des différents fichiers

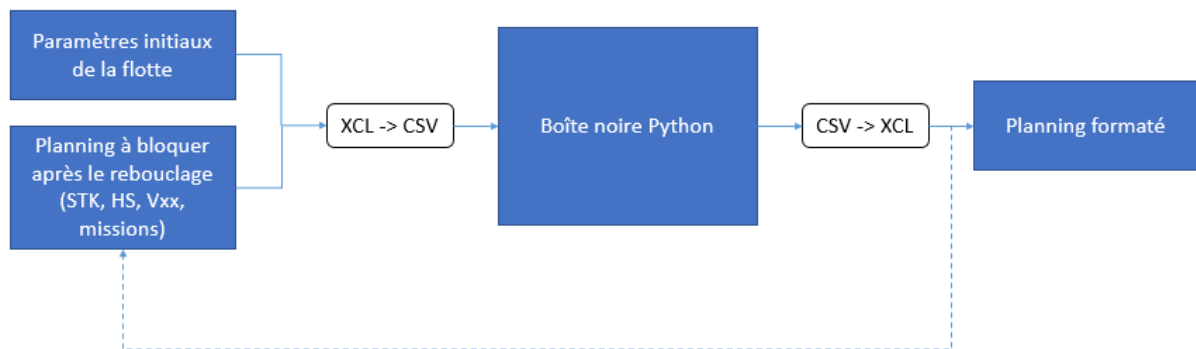


Figure 2.5.1 : Structure globale du code Python

5.2. Structure du code Python

Le code Python est composé de 6 fichiers divisés en quatre modules.

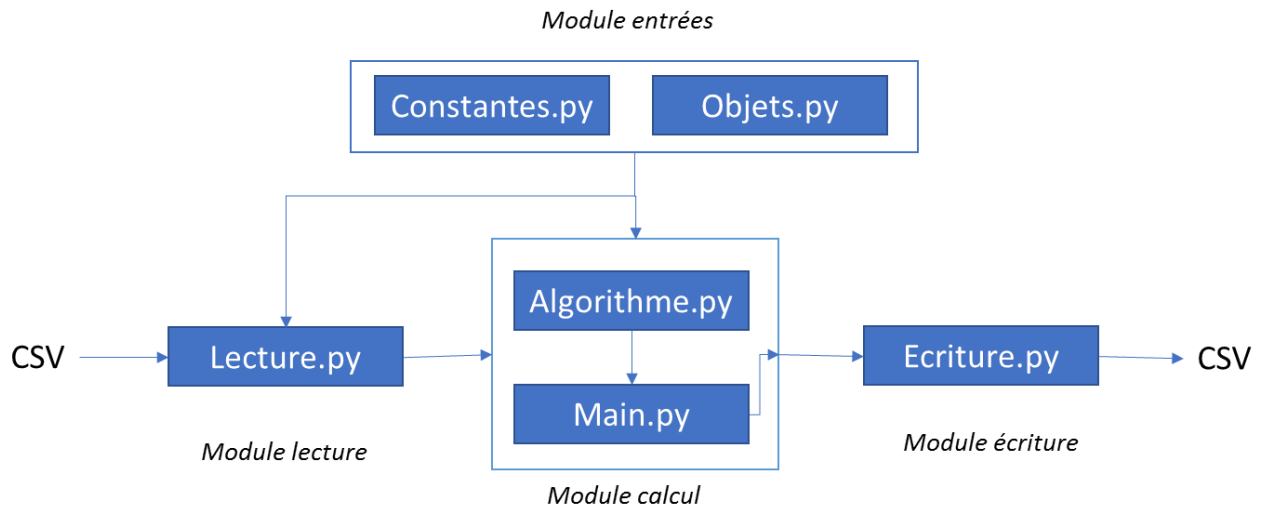


Figure 2.5.2.1 : Structure globale du code Python

Chaque fichier contient des éléments détaillés ci-dessous.

- **Objets.py :**

Ce fichier contient trois classes : avion, mission et maintenance. Chaque classe est composée d'un constructeur objet avec plusieurs instances en paramètre.

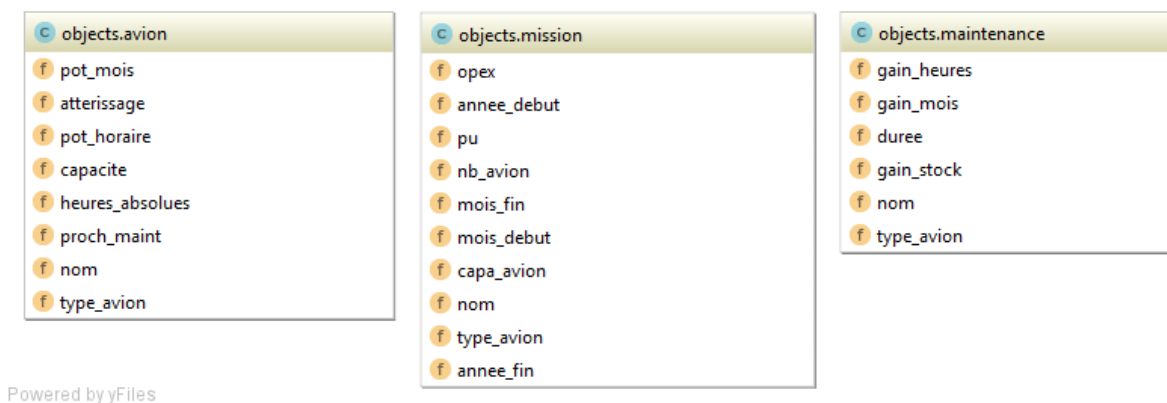


Figure 2.5.2.2 : Eléments contenus dans le fichier Objects.py

- **Constantes.py :**

Ce fichier contient tous les instances qui ne varient pas lors de la simulation, et qui ne nécessitent pas la création d'une structure objet.

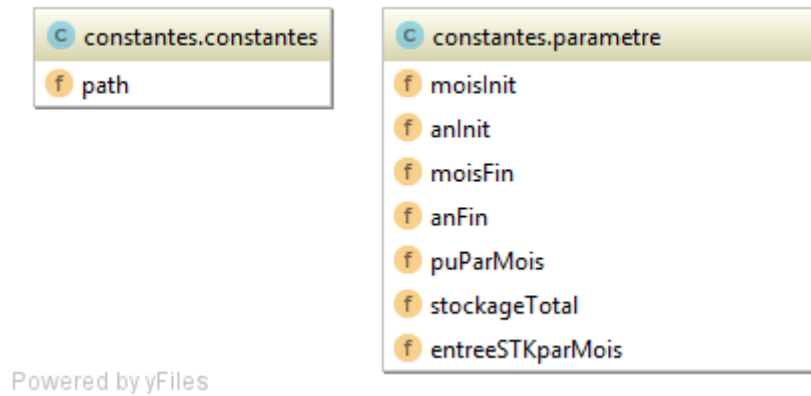


Figure 2.5.2.3 : Eléments contenus dans le fichier Constantes.py

- `Lecture.py` :

Ce fichier contient l'ensemble des méthodes utilisées pour la lecture des différents fichiers CSV.

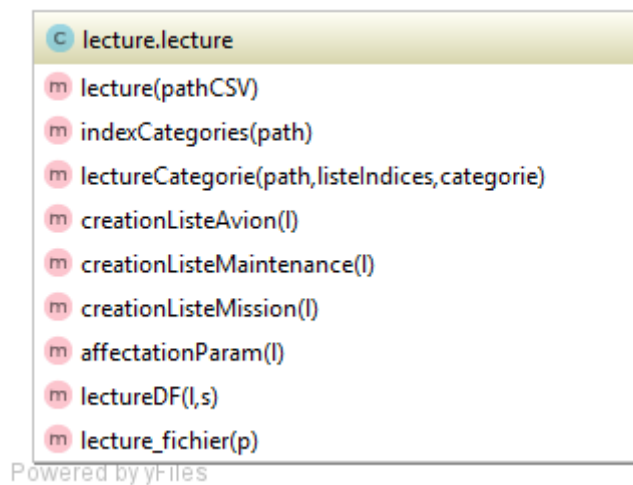


Figure 2.5.2.4 : Eléments contenus dans le fichier Lecture.py

- `Ecriture.py` :

Ce fichier contient deux méthodes : export de la solution finale et écriture d'un fichier indicateur sous format CSV.

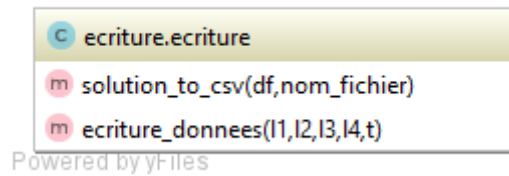


Figure 2.5.2.5 : Eléments contenus dans le fichier Ecriture.py

- Algorithme.py :

Ce fichier contient les méthodes d'affectations et de choix des avions pour les missions, maintenances, etc.

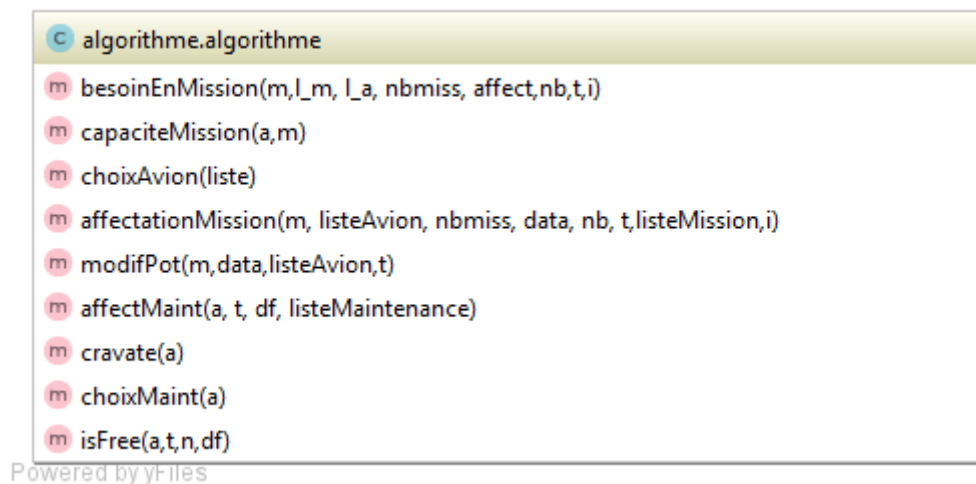


Figure 2.5.2.6 : Eléments contenus dans le fichier Algorithme.py

- Main.py :

C'est le fichier principal qui regroupe et fait appel à toutes les méthodes et classes citées auparavant. C'est le seul fichier à exécuter pour lancer le programme.

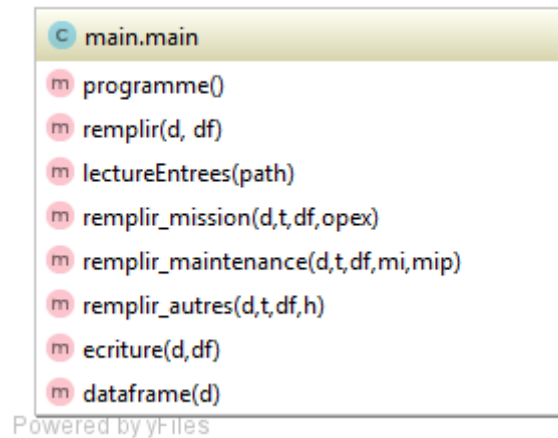


Figure 2.5.2.6 : Eléments contenus dans le fichier Main.py

5. Gestion des modifications

6.1. Intégration progressive des différentes affectations

Suite aux règles métier que nous avons déterminées, nous avons identifié sept affectations qui ont été intégré progressivement à l'outil : stockage, maintenance, missions OPEX, vol entrainement, HS, Missions métropole, BL. Le rebouclage correspond aux entrées manuelles par l'utilisateur (*id est* : l'utilisateur impose un choix). L'automatisation signifie que le programme lui-même décide (tout en respectant les choix de l'utilisateur, bien sûr).

	Rebouclage	Automatisation
V1.1	Stockage	Maintenances Missions OPEX Vols entrainement
V1.2	Stockage Maintenance Missions OPEX Vols entrainement HS Missions métropole	Maintenances Missions OPEX Vols entrainement
V2.1	Stockage Maintenance Missions OPEX Vols entrainement HS Missions métropole BL	Maintenances Missions OPEX Vols entrainement Missions métropoles

6.2. Détail des modifications

Le tableau ci-dessous détaille les modifications apportées à chaque version.

Version	Livrable pour le	Détail des modifications
V1.1	14/12/2016	<p>Point de départ avant-projet</p> <ul style="list-style-type: none"> - Automatisation de : <ul style="list-style-type: none"> o Missions OPEX o Maintenances o Vols d'entraînement - Rebouclage sur : <ul style="list-style-type: none"> o Stockage
V1.2	20/01/2017	<p>Python :</p> <ul style="list-style-type: none"> - Prise en compte du rebouclage sur : <ul style="list-style-type: none"> o les missions OPEX o les maintenances o les vols d'entraînement - Consommation du besoin entraînement « vol métropole » : consomme 18h/mois - Prise en compte sous forme de rebouclage des besoins : <ul style="list-style-type: none"> o HS o Missions métropole <p>Excel :</p> <ul style="list-style-type: none"> - Formatage conditionnel similaire au précédent planning excel utilisé par l'armée de l'air : <ul style="list-style-type: none"> o Affichage des plannings par catégorie avion o Affichage du potentiel horaire mensuel consommé o En commentaire le nom de la mission OPEX
V2.1	08/02/2017	<p>Python :</p> <ul style="list-style-type: none"> - Automatisation des missions métropole - Apparition du besoin « Blocage » (BL), un stockage avec vieillissement : rebouclage - Mis en place de tampons : <ul style="list-style-type: none"> o Tampon d'immobilisation de deux mois entre mission et maintenance o Tampon d'immobilisation de deux mois entre deux missions pour un avion - Mise en place d'un nombre max d'avions en maintenance par mois - Rédaction de commentaires dans le code - Code plus modulaire - Modification de la structure des fichiers .csv de situation initiale et des paramètres. <p>Excel :</p> <ul style="list-style-type: none"> - Accélération de l'affichage de la solution - Adaptation de la lecture .csv suites aux modifications du Python.
V2.2	En cours 24/03/17	<p>Python :</p> <ul style="list-style-type: none"> - Améliorer la généricité du code

		<ul style="list-style-type: none"> - Commenter le code - Lisser les maintenances - Ajouter des fonctions de préférences <p>Excel :</p> <ul style="list-style-type: none"> - Indiquer la progression de la fenêtre de commande Python sur l'Excel <p>Documentation :</p> <ul style="list-style-type: none"> - Test du rebouclage - Rédiger une documentation ingénieur pour que le code puisse être repris - Rédiger un manuel utilisateur - Définir des indicateurs évaluant la performance du planning.
--	--	--

6. Outil final et utilisation

7.1. Forme de l'outil pour l'utilisateur

Comme nous l'avons vu dans les paragraphes précédents, l'outil d'élaboration du planning est composé de deux modules principaux : le code Python et l'interface Excel. Par soucis de simplicité pour l'utilisateur, celui-ci n'interagit qu'avec l'interface Excel, puisque l'interface Excel est capable de lancer le code Python. Il doit cependant avoir téléchargé sur ordinateur un logiciel regroupant Python v3.5 accompagné des packages Pandas et Numpy (par exemple Anaconda 4.3).

Deux fichiers Excel sont à utiliser :

- *parametres_DGA* : fichier qui réunit les informations sur l'état actuel de la flotte (heures de vol des avions, missions à réaliser, maintenances, ...)
- *planning_final_DGA* : fichier qui permet de lancer le code Python, d'afficher le planning solution, et éventuellement de reboucler avec des choix imposés par l'utilisateur.

Les paragraphes suivant résument comment utiliser l'outil.

7.2. Première étape : modification des paramètres

La première étape pour l'utilisateur est de mettre à jour l'état de la flotte des avions, à partir du fichier *parametres_DGA*.

L'utilisateur met à jour les informations des onglets et fait des choix de paramètres (en bleu sur la figure suivante), puis génère le fichier .csv de paramètres (en rouge sur la figure suivante).

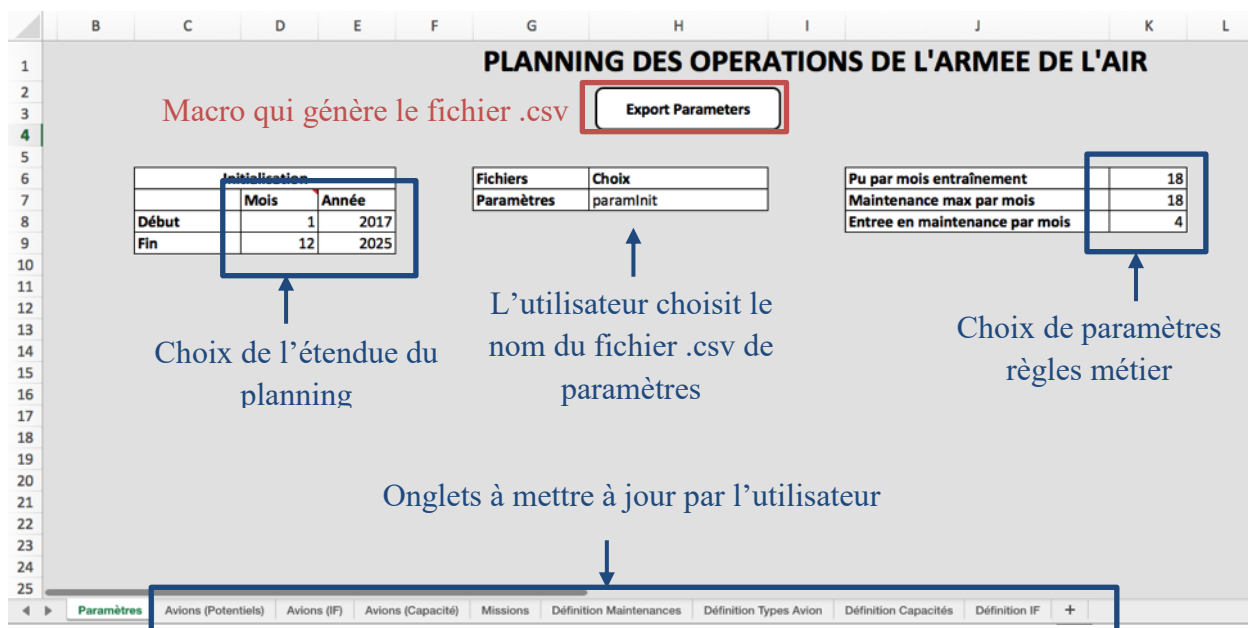


Figure 2.7.2.1 : Utilisation de l'Excel parametres_DGA

Un exemple d'onglet à mettre à jour est présenté ci-dessous : il s'agit des missions à remplir, qu'elles soient OPEX ou en métropole. Les missions ainsi que leurs caractéristiques (dates, configuration souhaitée,...) sont à renseigner par l'utilisateur.

	A	B	C	D	E	F	G	H	I	J	K
1	Id Mission	Matricule Mission	OPEX?	Mois de début	Année de début	Mois de fin	Année de fin	MaxPu/avion/mois	MinPu/avion/mois	ConsolF_1	ConsolF_2
2	O1	CHAMMAL_D	1	9	2017	12	2022	100	100	0.63	0.63
3	O2	NDJAMENA_D	1	10	2016	12	2022	100	100	0.63	0.63
4	O3	NIAMEY_C	1	7	2016	12	2022	50	50	0.68	0.68
5	O4	NIAMEY_D	1	7	2016	12	2022	50	50	0.36	0.36
6	O5	DIJIBOUTI_5F	0	1	2016	8	2028	23	10	0.99	1.08
7	O6	LUXEUIL_5F	0	1	2016	8	2028	23	10	0.62	0.67
8	O7	MARSAN_C	0	1	2016	8	2019	23	10	0.68	0.68
9	O8	MARSAN_D	0	1	2016	8	2025	23	10	0.23	0.23
10	O9	MARSAN_5F	0	1	2016	8	2028	23	10	0.62	0.67
11	O10	NANCY_D	0	1	2016	8	2030	23	10	0.32	0.3
12	O11	ORANGE_B	0	1	2016	8	2025	23	10	0.6	0.6
13	O12	ORANGE_C	0	1	2016	8	2019	23	10	0.68	0.68
14											
15											
16											
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											
27											
28											
29											
30											
31											
32											
33											
34											
35											
36											
37											
38											
39											
40											
41											
42											
43											
44											
45											
46											
47											
48											
49											
50											
51											
52											
53											
54											
55											
56											
57											
58											
59											
60											
61											
62											
63											
64											
65											
66											
67											
68											
69											
70											
71											
72											
73											
74											
75											
76											
77											
78											
79											
80											
81											
82											
83											
84											
85											
86											
87											
88											
89											
90											
91											
92											
93											
94											
95											
96											
97											
98											
99											
100											
101											
102											
103											
104											
105											
106											
107											
108											
109											
110											
111											
112											
113											
114											
115											
116											
117											
118											
119											
120											
121											
122											
123											
124											
125											
126											
127											
128											
129											
130											
131											
132											
133											
134											
135											
136											
137											
138											
139											
140											
141											
142											
143											
144											
145											
146											
147											
148											
149											
150											
151											
152											
153											
154											
155											
156											
157											
158											
159											
160											
161											
162											
163											
164											
165											
166											
167											
168											
169											
170											
171											
172											
173											
174											
175											
176											
177											
178											
179											
180											
181											
182											
183											
184											
185											
186											
187											
188											
189											
190											
191											
192											
193											
194											
195											
196											
197											
198											
199											
200											
201											
202											
203											
204											
205											
206											
207											
208											
209											
210											

L'utilisateur doit commencer par remplir un certain nombre de paramètres :

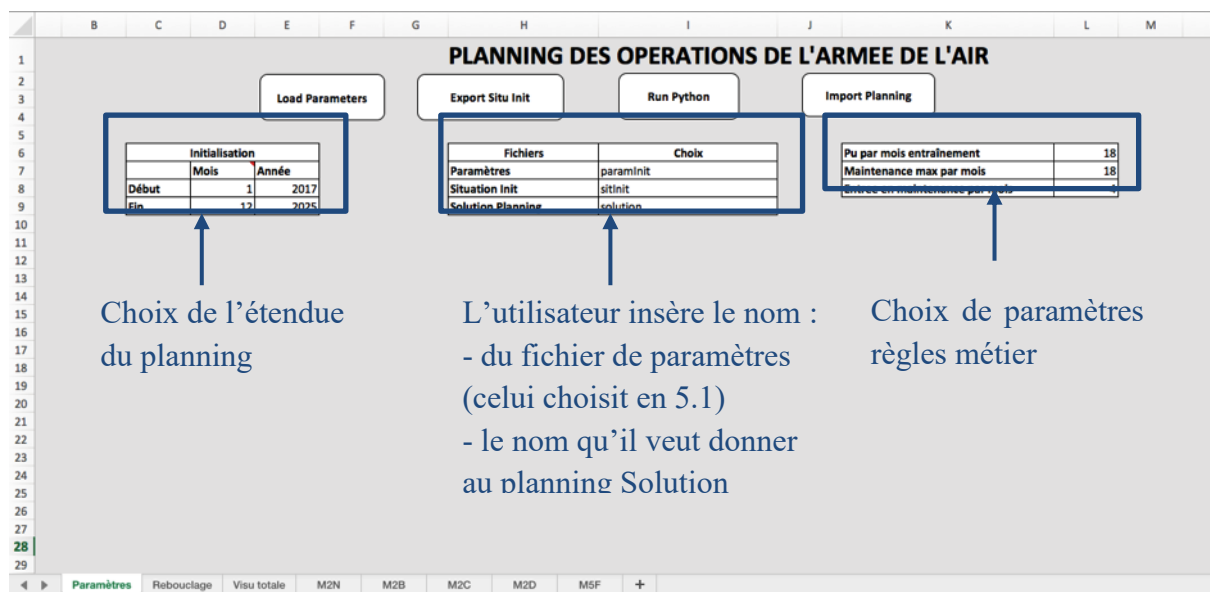


Figure 2.7.3.1 : Paramètres de l'interface du fichier planning_final_DGA

Par la suite, l'utilisateur lance trois Macro, qui lui permettent d'obtenir le planning.

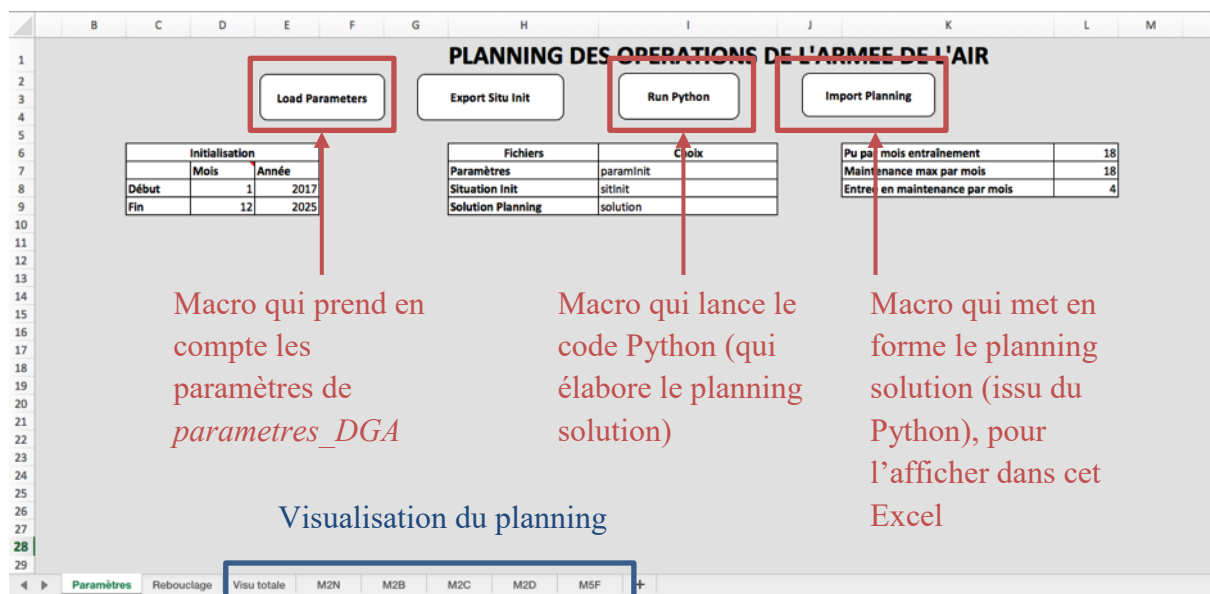


Figure 2.7.3.2 : Lancement des Macro du fichier planning_final_DGA

7.4. Troisième étape : Visualisation du planning

Six onglets de l'Excel *planning_final_DGA* permettent la visualisation du planning :

- “Visu totale”: onglet qui affiche le planning pour l'ensemble de la flotte

L'utilisateur peut imposer certaines affectations à partir de l'onglet "Rebouclage" du fichier *planning_final_DGA*. Il relance ensuite trois Macro pour obtenir le nouveau planning Solution, qui aura pris en compte ses choix.

Figure 2.7.5.1 : « Rebouclage » du fichier *planning_final_DGA*

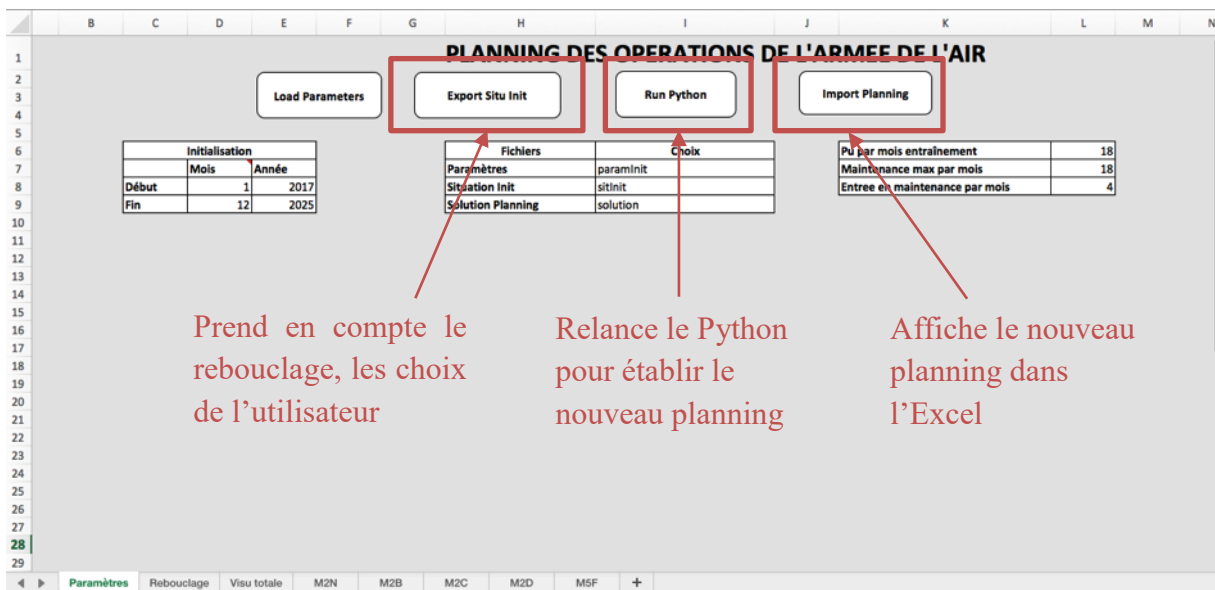


Figure 2.7.5.2 : Landement des macro pour le rebouclage dans le fichier *planning_final_DGA*

De même que précédemment, l'utilisateur peut alors observer le planning sur les différents onglets de visualisation. Les nouveaux plannings auront pris en compte ses choix.

IV. Conclusion

L'ambition de notre PIE était de créer un outil optimisant la planification de l'utilisation d'une flotte de Mirages 2000 pour l'Armée de l'Air. Pour cela, il a fallu comprendre en profondeur les besoins exacts de notre client et dans un second temps y répondre au mieux avec l'outil codé par nos soins. En organisant ingénieusement notre calendrier de travail, nous avons pu fournir rapidement des versions rudimentaires de l'outil, afin de l'améliorer par la suite en procédant par itérations. Les attentes finales n'étaient définies pour personne ; cette méthode a dès lors eu pour principal avantage de nous maintenir sur la bonne voie et d'éviter toute errance.

Mener ce projet n'a pas été chose aisée. Sa réussite a demandé un investissement intense et régulier de la part de chaque membre du groupe ; une organisation robuste pour coordonner les différents groupes de travail et l'avancement ; une forte résilience face à l'inconnu et aux difficultés. Le résultat final et la satisfaction manifeste de notre client valorisent ces efforts.

Ce rapport a essayé de dépeindre avec justesse un bilan de l'avancement de notre PIE pendant les six mois qu'il a occupés, de la première réunion à la soutenance prochaine. Si nous espérons avoir rendu compte de façon claire et exhaustive des éléments tangibles de notre progression, il est plus difficile de laisser transparaître la bonne entente qui régnait au sein du groupe, facteur non-négligeable de son bon fonctionnement.

Au-delà du résultat final, c'est probablement le chemin parcouru qui est source de notre plus grande satisfaction. Nous avons appris à travailler en groupe, à nous placer dans la position d'un sous-traitant face à un client exigeant, à utiliser et mettre en œuvre des méthodes de gestion de projet, à coder un algorithme glouton en python... autant de compétences qui seront mises en valeur dans les débuts (proches) de nos vies professionnelles.