

**Power
Week**

Université IBM i 2019

22 et 23 mai

IBM Client Center Paris



**S44 – Développement IBM i et Agilité :
compatible ou pas ?**



Étienne PETIT
ID INFO
etienne.petit@idinfo.eu

Société à taille humaine,
avec comme objectif
d'assister les entreprises
à construire et se
projeter sur un SI pour
les 15 prochaines années

ID INFO, en deux mots

LA société qui est dédiée à la plateforme POWER

- Profils techniques / formation
- Smart'Infra Logiciels / Matériels
- Services managés

Une approche unique :

- Une équipe entièrement orientée projet (prise en compte besoin métiers, organisation, contexte technique)
- Assistance et accompagnement

Une association pour
mieux répondre aux
besoins des directions
générales et directions
informatiques

ID INFO, en deux mots

L'agilité passe par une analyse de son marché et nécessite de s'adapter à l'évolution des besoins du marché.

Modernisation, ressources jeunes, approches agiles, etc... sont les souhaits des directions pour continuer leur engagement sur la plateforme IBM i.

Pour plus de proximité, plus d'assistances techniques, plus de formation, plus de développements, plus de sécurité, plus de plus !!



S'ASSOCIENT !

Comment l'agilité est devenue incontournable

Au 20^{ème} siècle les utilisateurs de l'informatique étaient internes et n'avaient pas comparaison en terme de fonctionnalité et délai.

Au 21^{ème} siècle : les progrès incessants : internet, bande passante mobile, smartphone, font que chaque user est plus exigeant et peut comparer à tout instant avec le SI interne.

Au siècle dernier, les directions fonctionnelles (MOA) demandaient des développements avec une attente en terme de délai assez longue (plusieurs mois ou années).

Souvent la communication était sporadique et les besoins initiaux évoluaient sans que la MOA ne le signale à la MOE.

Lors du développement classique du cycle en V (analyse, spécifications, conception, codage, test) les premiers livrables ne correspondaient plus à la MOA.

Le BtoC avec son besoin de réactivité a changé les attentes, surtout celles des Directions (utilisatrices des moyens technologiques modernes).

Malgré les efforts d'IBM pour véhiculer une image plus moderne :

Les services de stratégie en communication d'IBM l'ont renommé à plusieurs reprises au début du 21^{ème} siècle pour essayer de le rendre plus attractif.

Son nom aujourd'hui : l'IBM i (pour mieux représenter son incomparable base de données intégrée)

On change de nom mais est-ce plus AGILE ?

L'AS/400 aux yeux de bon nombre de Directions est un vieux système centralisé, solide, géré par des spécialistes peu ouverts aux nouvelles technologies, et représentant le 20^{ème} siècle.

Et pourtant l'IBM i du 21^{ème} siècle est un système performant, agile et très ouvert à toutes les dernières technologies.

Cependant il ne faut pas détourner sa fonction primaire qui est la gestion de données transactionnelles.

Pourquoi est-il perçu à risque par la majorité des décideurs au lieu de le voir comme une opportunité ?

Les individus et leurs interactions

Préjugés sur l' « AS400 » :

- Plateforme vieillissante
- Mauvaise image des DG/DSI
- Peu d'innovation
- Ressources techniques rares
- Développements isolés
- 1 PGM = 1 développeur
- « Effet tunnel »

L'agilité a gagné en popularité pour sortir de l' « effet tunnel »

Il augmente considérablement le risque de dérapages au niveau des jalons temporels et du budget.

Langage legacy : les programmes étaient auparavant développés en **silos** et de façon **séquentielle**

Donc, estimée comme n'étant pas compatible avec la notion d'AGILITE

Justifié ou non ?



IBM i : AGILE avec l'adaptation au changement

Plate-forme accessible aux langages de programmation de chaque époque

Tout en gardant les grands principes de sa base de données DB2

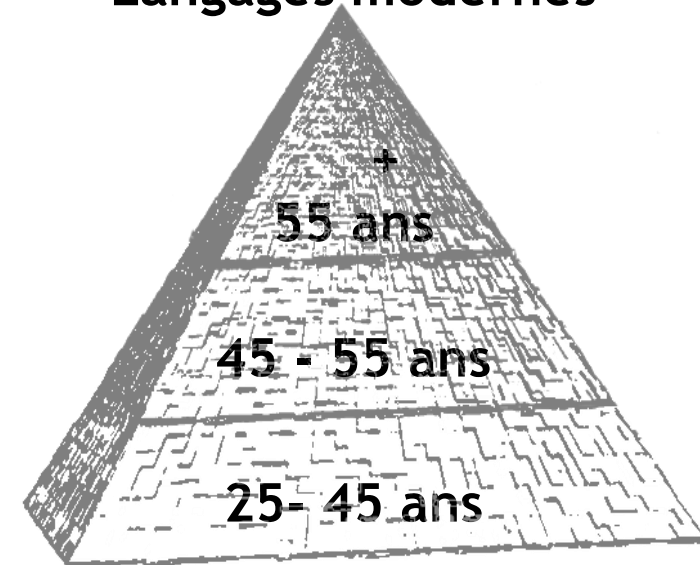
Des langages de programmation plus modernes sont sans cesse proposés sur la plate-forme

Open Source sur IBM i :

- Multitude d'outils et de communautés, évolutions permanentes, compétences facile à trouver ...
- Ouverture de l'IBM i avec l'extérieur plus facile, nouvelles possibilités (applications Web, mobile)
- Délivré par IBM (mise à jour par PTFs)

- RPG
 - full free, SQL, RDi
- Développement ILE
 - Interfaçage entre tous les langages selon leur utilité propre
- Open Source (5733-OPS)
 - Node.js
 - Python
 - GCC/chroot
 - Git
 - Orion
 - Cloud-init
 - Nginx
 - Perl
 - Curl
 - ...

Langages modernes



1. Équipe

2. Outils

3. Process

Stratégie = être capable de réagir aux évolutions du marché

Le besoin ne peut être figé : il faut s'adapter aux changements de ce dernier avec un minimum de règles

1. Une équipe => implication de A à Z de tous les collaborateurs
2. Des outils adaptés au contexte => portfolio important
3. Un process itératif et incrémental => PDCA, capitalisation

Comment inverser les tendances

L'IBM i aujourd'hui est la « colonne vertébrale » des activités dans de nombreuses entreprises.

Il faut que les DSI s'inscrivent auprès des Directions Générales et fonctionnelles en et les accompagnant sur :

- Revue de l'existant : atouts et faiblesses de l'IT en place, organisation existante,
- Compréhension des enjeux métiers et besoins utilisateurs,
- Projections d'évolutions de l'entreprise : métiers, organisation interne - externe, décisionnel, IA, etc...
- Définition de la stratégie avec la cible à atteindre,
- Présentation du projet global avec une méthodologie impliquant les acteurs majeurs de l'entreprise.

Il semble plus opportun :
*de penser aux changements,
que de changer le pansement !*

Utilisez les atouts de votre IBM i grâce à la transformation et la modernisation **étape par étape** du code source, de la base de données et de l'interface utilisateur.

L'approche AGILE

Le mouvement AGILE que nous préconisons repose sur 4 phases élémentaires permettant de moderniser le SI basé sur du code RPG pour un SI plus moderne, plus « digital », plus adapté aux attentes des utilisateurs



L'approche AGILE

L'analyse du code est le préalable a toute démarche de modernisation.

Elle permet d'analyser la qualité de la documentation et de nettoyer les programmes non utilisés.

C'est une action à réaliser que ce soit pour un projet bigbang ou un projet de modernisation.

Première étape indispensable : analyser le code et les programmes pour mieux les moduler.

Une cartographie précise permet d'isoler les domaines fonctionnels et d'arbitrer sur un nouveau développement sur une autre plateforme ou de transformer le code existant.

C'est le point de départ indispensable !

Est il possible de construire un immeuble sans plans d'architecte ?

Comprendre le
SI central
Cartographie/
reverse ingeniering

Cette phase permet de
poser des jalons sur un
développement AGILE.

L'approche AGILE

Si le choix est de construire sur le socle IBM i, il est important de mettre en place des outils et processus dans l'analyse du code.

La mise en place de modes opératoires pour une TMA est souhaitée.

Les vieux codes doivent être analysés pour bien identifier la possibilité de les transformer en langage plus moderne ou si un manque de source nécessite de tout réécrire.



L'approche AGILE

Les programmes listés comme pilote pour être gardés doivent être transformés en langage plus moderne.

Le but est de pouvoir se projeter pour les 15 prochaines années et d'être gérés par des développeurs plus jeunes pour assurer la continuité.



Cette phase ne peut être totalement automatisée, l'utilisation d'outil permet de gagner du temps mais l'intervention humaine est obligatoire car les moulinettes de transformation excèdent rarement plus de 80% du code transformé.

L'approche AGILE

La gestion du versioning est aussi importante pour les développements que pour l'exploitation du SI

La culture AGILE prend en compte la notion de traçabilité des développements, mais également la mise en production.

Toute intervention doit être saisie et documentée pour effectuer rapidement un diagnostic.

Pour rappel, le DEVOPS est une démarche de collaboration agile entre :

- Les études & développement
- La production, l'exploitation et l'infrastructure
- Les métiers, le business.

Gérer les
Versions des
développements
(ALM)

Transformer le RPG : compliqué ou pas ?

La méthodologie AGILE nécessite de bien décomposer chaque phase du développement d'un module.

Les objectifs sont :

- ✓ Respecter les délais et le budget
- ✓ Capitaliser sur chaque développement
- ✓ Automatiser le plus possible au travers d'outils
- ✓ Documenter pour la TMA et l'amélioration des processus

Un projet AGILE nécessite la définition d'une équipe type. C'est parfois complexe d'immobiliser les ressources internes.

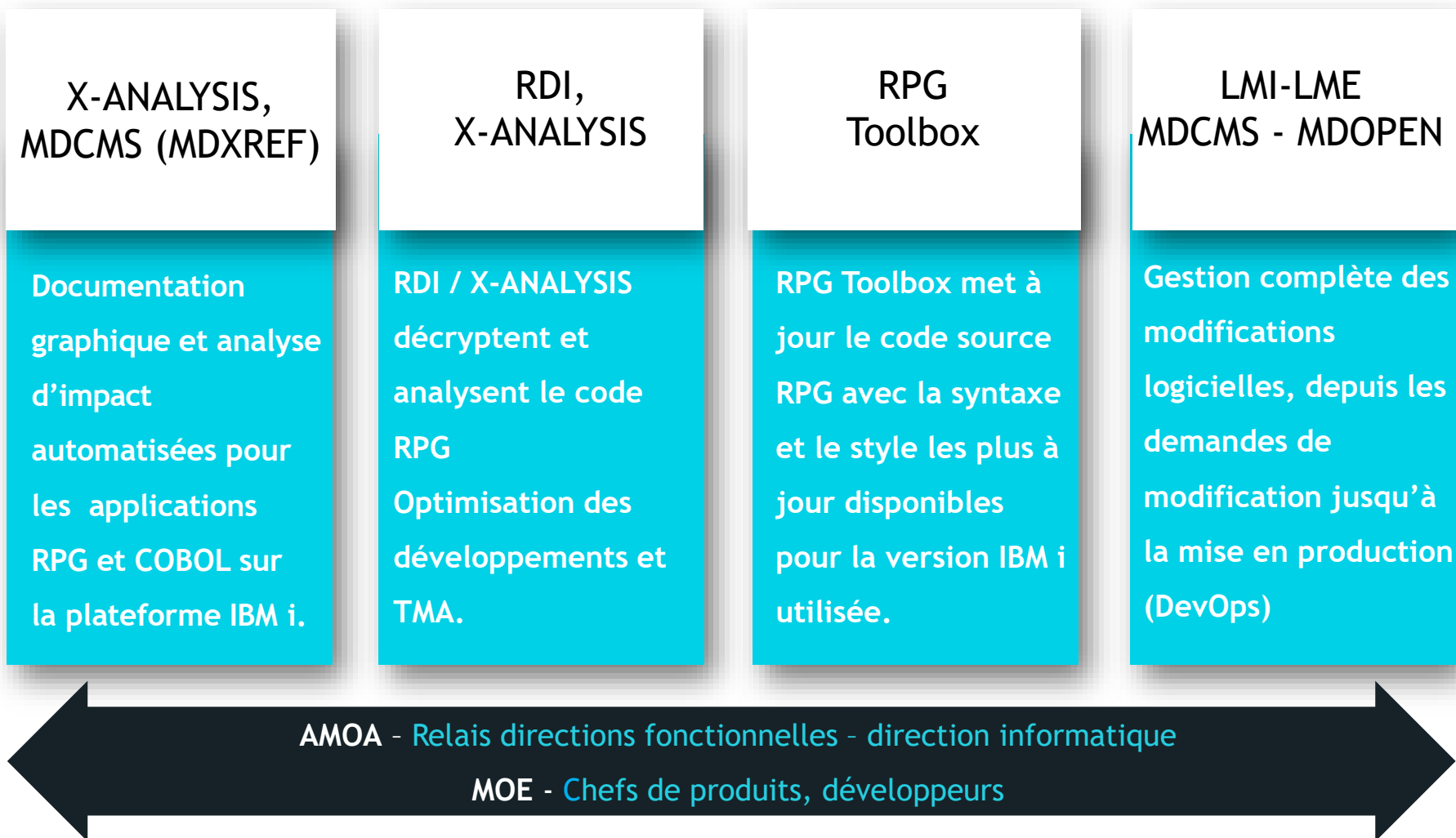
ID INFO / NOTOS dispose

- d'une cellule de sourcing dédiée à la recherche et sélection de candidats (CV-thèque de 200 profils IBM i, développeurs, exploitants, chefs de projets, consultants).
- d'une cellule de formation sur les méthodes et langages de programmation moderne.

Les outils d'assistances

Le SI est documenté, les programmes maîtrisés et transformés en langage plus moderne → facilités pour trouver des ressources techniques.

Possibilités de construire un système moderne et mettre en place une stratégie API et/ou webservices



ID INFO/NOTOS est une organisation de type agile, utilisant les « bonnes pratiques » dans le développement de ses services.

Le référentiel ITIL est devenu un incontournable international dans le domaine de la gestion des services informatiques, mais il est également souvent considéré comme lourd et susceptible de scléroser une organisation en lui faisant perdre de son agilité.

Le cadre de référence PRINCE2 couvre une large gamme de concepts agiles, incluant SCRUM, Kanban.

ID INFO/NOTOS combinent les deux mondes, pour donner « l'Agile Service Management ».

C'est plus simple de regarder et de suivre un cap tout préparé avec des bonnes pratiques et avec une liste d'erreurs prélistées à ne pas appliquer.

SCRUM comment cela fonctionne

La modélisation d'une organisation permet à tous les acteurs de comprendre son positionnement et les attentes de chacun.

3 ROLES



Product Owner
Utilisateurs Métier



Equipe
Développeurs



Scrum Master
Veille au respect des processus et à la non perturbation de l'équipe

3 LIVRABLES



Product Backlog
Liste priorisant les fonctionnalités à réaliser, rédigée par le Product Owner



Sprint Backlog
Liste des fonctionnalités à réaliser dans un Sprint, rédigée en accord avec le Sprint Owner

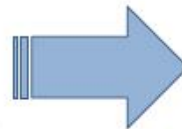


Impediment List
Liste des obstacles nuisant à la qualité et à la productivité du produit, rédigée par le Scrum Master

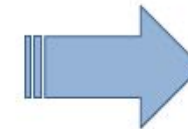
3 CEREMONIES



Sprint Planning
Réalisé avant chaque début de Sprint, permet de lister le Sprint Backlog en accord avec le Product Owner



Sprint Review
Réalisée en fin de Sprint, l'Equipe fait une démonstration du produit avec les fonctionnalités réalisées



Rétrospective
Réalisée après chaque Sprint, l'Equipe et le Scrum Master se réunissent pour comprendre les erreurs du Sprint et s'améliorer



Daily Scrum Meetings
Point de synchronisation et d'avancement réalisé quotidiennement durant 15 min avec l'Equipe et le Scrum Master (suivi et remontée des éventuels problèmes)

La modélisation d'une organisation permet à tous les acteurs de comprendre son positionnement et les attentes de chacun.

Comment ça marche ?

Pour comprendre Scrum, regardons ses principales caractéristiques :

- **Priorisé** : les utilisateurs métier sont représentés par le **Product Owner** qui liste et priorise les fonctionnalités à réaliser. Cette liste s'appelle le **Product backlog**. Les fonctionnalités les plus importantes seront réalisées en premier.
- **Itératif** : l'**équipe** développe ces fonctionnalités par itérations courtes et de durée fixe (généralement entre 2 et 4 semaines), appelées les **sprints**. Les fonctionnalités réalisées dans un sprint, listées dans le **sprint backlog**, sont sélectionnées en accord avec le Product Owner lors du **sprint planning** avant le début de chaque sprint.
- **Incrémental** : chaque sprint aboutit à un incrément du produit opérationnel (potentiellement livrable). A la fin du sprint, lors du **Sprint review**, l'équipe fait une **démonstration** du produit avec les fonctionnalités réalisées.

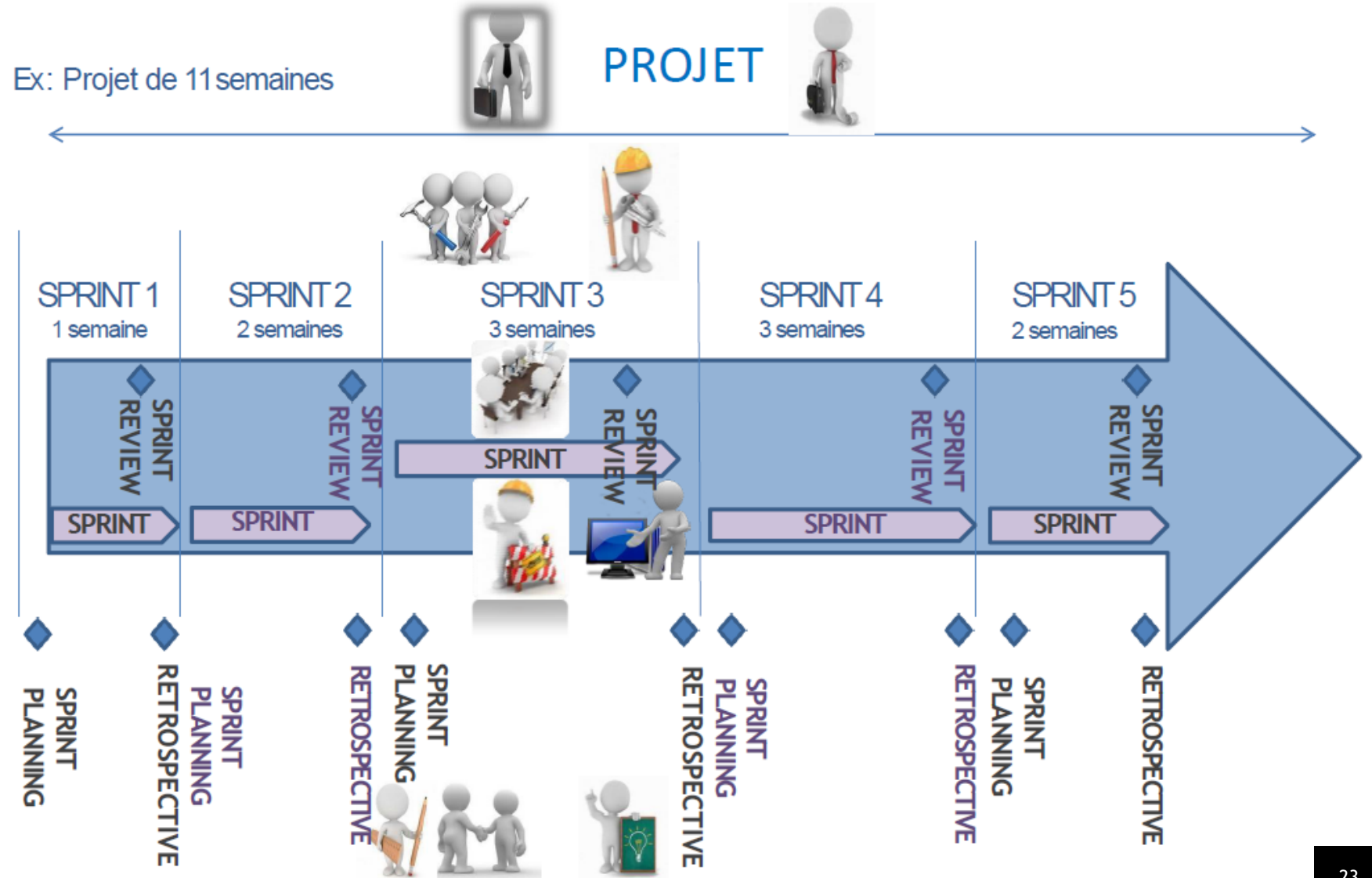
Rôles des acteurs

La modélisation d'une organisation permet à tous les acteurs de comprendre son positionnement et les attentes de chacun.

- **Adaptatif** (réponse au changement) : entre chaque sprint, les priorités métier peuvent changer et les évolutions sont les bienvenues. Il suffit de mettre à jour le product backlog.
- **Productif** : une fois qu'un sprint est commencé, l'équipe se concentre sur le développement des fonctionnalités sélectionnées et ne doit pas être dérangée. Le **ScrumMaster**, qui veille au respect du processus, s'assure notamment que l'équipe n'est pas perturbée par des événements extérieurs. Sauf cas exceptionnel, les évolutions attendront le sprint suivant.
- **Amélioration continue** : tous les jours, l'équipe fait un point de synchronisation et d'avancement (15 min), le **daily scrum** (mêlée quotidienne), pendant lequel l'équipe remonte les éventuels problèmes au Scrum Master qui est chargé de les résoudre. Le Scrum Master trace tous les obstacles qui nuisent à la productivité et à la qualité du produit dans l'**impediment list**.
Après chaque sprint aussi, l'équipe et le Scrum Master se réunissent pour faire une **rétrospective** du sprint afin de comprendre leurs erreurs et s'améliorer.

Projection simplifiée projet

La représentation schématique permet de se projeter dans des plannings macroscopique qui sont souvent relayés des GANT's et des rétroplannings.



Questions





📍 12 rue de la tombe d'Issoire, 75014 Paris

@ guy.routier@idinfo.eu

☎ 01 88 32 12 34

