

# Mener à bien son projet de RPA

---

**Objet.** Cette note vise à accompagner les chefs de projets de la sphère publique dans leurs initiatives d'automatisation des processus via les technologies de *robotic process automation* (RPA). Elle détaille les opportunités, les limites et les risques de cette technologie.

**Contact.** Antoine Michon, Chef de projet, Direction interministérielle du numérique (DINUM)  
[antoine.michon@modernisation.gouv.fr](mailto:antoine.michon@modernisation.gouv.fr)

Introduction .....	2
1. Qu'est-ce que le RPA ? .....	2
2. Identification des processus à automatiser .....	4
3. Choix du logiciel de RPA .....	6
4. Paramétrage des robots .....	7
5. Vision long terme.....	8
6. Risques et écueils principaux.....	9
Conclusion.....	11

# Introduction

La technologie de *Robotic process automation* (RPA) vise à automatiser les opérations informatiques répétitives des agents *via* le déploiement de scripts. Elle cible la réduction du temps passé à l'exécution de tâches à faible valeur ajoutée, dégageant ainsi de la productivité tout en augmentant la satisfaction des agents.

Cette technologie connaît actuellement un succès commercial important, tiré par des ventes croissantes dans les domaines de la banque-assurance et de l'industrie manufacturière. *UiPath*, un des leaders du domaine, a achevé en avril 2019 une levée de fonds de 400M\$, alors qu'*Automation Anywhere* recevait quelques mois auparavant 300M\$ de ses investisseurs. Ce segment est considéré comme l'un des plus dynamiques de l'informatique professionnelle<sup>1</sup>.

Cette note, à destination des chefs de projets de RPA de la sphère publique, vise à préciser les contours de cette technologie, à en détailler les cas d'usage intéressants ainsi qu'à en dessiner les limites et les risques. Elle détaille comment mener à bien un projet de RPA.

## 1. Qu'est-ce que le RPA ?

Le RPA consiste en l'exécution de scripts (ou robots) qui mimiquent l'action d'un agent sur son poste de travail. Dans les versions les plus basiques, cela correspond donc à un algorithme qui réalise des actions souris et clavier dans l'interface utilisateur d'un ordinateur personnel. Des systèmes plus élaborés peuvent réaliser des opérations complexes sur des infrastructures de bureau virtuelles (*virtual desktop infrastructure*) situées sur serveur. Quoi qu'il en soit, le RPA permet d'automatiser les actions des agents répétitives et à faible valeur ajoutée.

En règle générale, ces opérations sont constituées d'interactions avec différentes applications non intégrées les unes avec les autres. Le RPA réalise alors ce lien inter-applicatif « manuellement » en imitant les actions des usagers sur ces différentes applications. **On peut ainsi définir le RPA comme une technologie permettant de faire de l'intégration entre applications au niveau de l'interface utilisateur (front-end)**, par opposition à des intégrations classiques réalisées directement dans les bases de données (back-end).

## Proposition de valeur

Le RPA vise à améliorer la productivité globale de l'organisation en satisfaisant des besoins d'automatisation provenant des métiers. Plus en détail, le RPA cible les améliorations suivantes :

- **Productivité** : les robots doivent permettre aux agents d'accorder plus de leur temps pour l'exécution des tâches à forte valeur ajoutée, en les déchargeant significativement de leurs tâches à faible valeur ajoutée.
- **Satisfaction des agents** : les tâches répétitives étant souvent les plus irritantes et rébarbatives pour les agents, le RPA vise à améliorer leur quotidien et à donner plus de sens à leur travail.

---

<sup>1</sup> Selon Gartner, le RPA est le segment de l'informatique professionnelle ayant la croissance la plus forte. Source : <https://www.gartner.com/en/newsroom/press-releases/2019-06-24-gartner-says-worldwide-robotic-process-automation-sof>

- **Satisfaction des utilisateurs** : en réduisant les erreurs dans la gestion des dossiers, en accélérant le traitement de ceux-ci et en augmentant la disponibilité des agents pour les cas complexes, le RPA doit améliorer la satisfaction des utilisateurs.

Il existe deux grands types de RPA : le RPA interactif et le RPA non interactif.

## RPA non interactif

Il s'agit de robots effectuant des tâches sans interaction avec l'agent ou l'usager. Il peut s'agir par exemple du transfert de données d'un système d'information (SI) à un autre, ou du croisement de données entre plusieurs SI. Imaginons le cas suivant : pour le traitement d'un dossier d'inscription, un agent reçoit les données du dossier dans une application web, doit ensuite les comparer avec des données disponibles dans une autre application web, puis finalement inscrire ces données dans une application Windows en client lourd. Si toutes ces opérations sont déterministes et sans valeur ajoutée, c'est-à-dire si on peut les synthétiser par des règles simples, alors un script de RPA peut répliquer dans l'interface utilisateur les actions de l'agent, sans risque d'erreur et avec une rapidité démultipliée.

## RPA interactif

Les robots de RPA interactifs nécessitent une action de la part de l'usager. Celle peut consister en la vérification d'éléments qui sont présentés à l'écran, ou bien en une prise de décision à un moment du processus. Un exemple : imaginons que pour traiter une demande d'un usager, un agent doive réunir des informations situées dans plusieurs SI avant de pouvoir prendre une décision. On peut imaginer un robot qui récupère ces données dans les différentes SI, puis lui présente une vision synthétique du dossier, et enfin demande à l'agent de prendre une décision sur la suite à donner à la demande. Finalement, le robot prépare un mail automatique contenant la décision de l'agent, et l'envoie à l'usager.

## Technologie

Les solutions de RPA sont généralement constituées de trois briques logicielles intégrées, déployées simultanément :

<b>Brique de développement</b> Environnement de développement des robots	<b>Brique de contrôle et d'orchestration</b> Programme sur serveur, en charge de lancer et de surveiller la bonne exécution des robots	<b>Brique d'exécution</b> Programme exécutant le robot, installé sur serveur ou sur ordinateur personnel
---	---	---

Les principaux logiciels du marché reposent sur le *framework* .NET et impliquent ses langages de programmation C# et VB.NET. Plus récemment, certains acteurs ont migré leur plateforme sur Java. Alors que les scripts d'automatisation de tâches sur ordinateur personnel existent depuis une vingtaine d'années déjà, la notoriété soudaine du RPA peut interroger. L'essor récent du RPA – alors qu'il utilise des technologies anciennes – semble être le fruit d'une conjonction de facteurs :

- des développements réalisés pour rendre les environnements de développement plus intuitifs, permettant une programmation très haut niveau adaptée à des développeurs non experts
- des investissements récents des éditeurs de logiciel de RPA dans des technologies permettant d'industrialiser à l'échelle d'une organisation entière l'automatisation des processus (notamment la sécurité et la traçabilité des scripts), tandis que les scripts étaient jusqu'alors cantonnés à des déploiements très locaux

- des progrès importants dans le domaine de l'intelligence artificielle, en particulier dans l'analyse sémantique et la reconnaissance de documents
- un marketing nouveau et agressif des éditeurs de logiciel de RPA, à travers duquel les scripts sont renommés « robots » et présentés comme véritable « main d'œuvre digitale »

## RPA, API et évolutions applicatives

La particularité du RPA est qu'il réalise des intégrations entre applications au niveau de l'interface utilisateur, ce qui est loin d'être idéal pour de nombreuses raisons. Cela augmente en règle générale la dépendance de l'organisation envers ses vieux SI, et accentue donc la dette technique. L'inflation du nombre de scripts en production peut également devenir un véritable casse-tête de gouvernance et d'entremêlement de processus. Sans compter la gestion de la sécurité informatique, qui est rendue plus complexe par les interactions multi-application des robots. Ces points seront tous approfondis par la suite. Pour toutes ces raisons, il faut être vigilant à utiliser la technologie de RPA en « dernier recours » et comme solution de transition. En particulier, il convient d'être attentif à :

- Toujours privilégier une intégration entre applications au niveau du back-end au RPA. Cela signifie qu'il est préférable, dès que c'est possible, d'effectuer le développement des API nécessaires à l'automatisation du processus, ou d'utiliser ces API si elles existent, plutôt que de déployer des robots de RPA.
- Favoriser impérativement les évolutions applicatives au développement de robots de RPA mono-applicatifs (n'opérant que sur une seule application).

## 2. Identification des processus à automatiser

Un projet de RPA commence systématiquement par l'identification des processus qui vont être automatisés. La meilleure manière de procéder consiste à entamer une démarche centrée sur l'amélioration des processus. Pour ce faire, on s'attachera à identifier en premier lieu les processus de l'organisation méritant d'être améliorés. Dans un second temps, on triera et priorisera les processus proposés selon des critères d'éligibilité et de rentabilité.

### Obtenir des propositions de processus améliorables

Il existe deux stratégies principales afin d'obtenir des suggestions d'automatisation.

- 1) **Réunir une sélection d'experts métier, puis leur demander de soumettre leurs idées d'automatisation ou d'amélioration de processus.** Quelques conseils :
  - Si l'on fait une présentation du RPA, on privilégiera toujours une séance centrée autour d'une démonstration. Cette démonstration est en effet capitale pour faire comprendre à l'utilisateur ce que peut réaliser un robot de RPA, et fonctionne pour cela bien mieux qu'une projection de diapositives PowerPoint.
  - Il est possible de faire émerger des propositions « à chaud » en organisant un atelier créatif en fin de présentation avec les experts métiers présents.
  - Une alternative, qui peut être complémentaire à l'atelier de créativité, consiste à demander aux experts métiers de soumettre leurs idées « à froid », quelques jours ou semaines après l'atelier.

- 2) **Réaliser une immersion dans les fonctions métiers pour identifier des processus automatisables**, en sélectionnant auparavant les zones de l'organisation soumises aux plus fortes tensions de charge de travail ou réalisant le plus de tâches de gestion de dossiers. On peut également cibler les services connaissant une augmentation conjoncturelle forte de leur volume d'activité (en cas de déploiement d'une nouvelle politique publique par exemple). Une bonne connaissance des critères d'éligibilité détaillés ci-après aide afin de choisir les services dans lesquels réaliser cette immersion. L'avantage principal de cette seconde technique est qu'elle mobilise moins de temps des métiers. Son approche moins systématique la rend toutefois moins performante. Notons que ces deux approches peuvent aller de pair.

## Trier et prioriser les processus proposés

Après avoir constitué une liste de processus candidats à l'automatisation, il faut établir ceux qui présentent un retour sur investissement significatif, et les classer selon une liste de priorités. Trois facteurs principaux doivent être pris en compte afin d'évaluer l'opportunité de déployer du RPA sur un processus :

- 1) **Le gain conféré par l'automatisation du processus en question.** Pour cela, les questions à se poser sont les suivantes :
  - Est-ce un processus à forte charge de travail ? Peut-on quantifier précisément le temps annuel dédié à l'exécution de ce processus ?
  - Est-ce un processus qui peut engendrer des erreurs humaines (fautes de frappe ou d'inattention) ?
  - Est-ce un processus dont l'automatisation réduirait significativement le délai de traitement pour la partie tierce concernée (citoyen, autre administration, agent, etc.) ?
- 2) **La difficulté technique de l'automatisation de ce processus via le RPA.** On se demandera ainsi :
  - Le processus repose-t-il sur des règles simples, déterministes, sans décision humaine ?
  - Le processus utilise-t-il uniquement des données structurées (pas de champ libre de texte par exemple) ? Quelle est la fiabilité des données en question ?
  - Le processus est-il standardisé dans l'organisation et stable dans le temps ?
  - Combien d'applications différentes sont mobilisées dans ce processus ?
  - Quelles sont les différentes étapes du processus ?
  - Dispose-t-on d'un environnement de test commun à toutes les applications impliquées dans le processus, répliquant exactement les applications en production ? Dispose-t-on de données de test répliquant les caractéristiques des données réelles ?
- 3) **La pertinence de l'automatisation de ce processus dans la stratégie globale SI de l'organisation.** Il conviendra de se poser les questions :
  - Est-ce que l'automatisation de ce processus peut se réaliser dans le back-end via des APIs ? Si non, peut-on développer ces APIs ?
  - S'il s'agit d'un processus mono-applicatif (ne se déroulant que dans une seule application), peut-on faire évoluer l'application ?
  - Quelles sont les *roadmaps* à moyen terme pour les applications concernées par le processus (horizon 3 à 4 ans) ?

Ces quelques questions constituent une grille d'éligibilité. En fonction des réponses à celles-ci, on peut établir une liste de priorités. On notera que de nombreux irritants identifiés qui sont inadaptés au déploiement de RPA peuvent toutefois être réorientés vers d'autres solutions technologiques.

## Optimisation des processus

Il faut être vigilant à ne pas automatiser un processus non optimisé. Quelquefois, le gain de productivité le plus important se trouve dans la remise à plat de processus trop complexes – souvent issus d'accumulations successives de nouvelles règles au fil des années – plutôt que dans leur automatisation. Il convient donc impérativement de s'assurer de la pertinence des processus avant d'entamer les développements de robots. Pour ce faire, un regard extérieur est souvent appréciable.

### 3. Choix du logiciel de RPA

Le choix de l'éditeur de logiciel de RPA est structurant pour le reste du projet, dans la mesure où les solutions présentes sur le marché ont des caractéristiques relativement distinctes et que l'interopérabilité entre logiciels est quasi-nulle. On dresse une liste de critères à prendre en compte dans ce choix :

- **RPA interactif vs. non interactif.** Les acteurs sont en général plutôt spécialisés dans un des deux aspects. Si l'identification des processus à automatiser permet de déceler des besoins prépondérants plutôt orientés vers un type de RPA (interactif ou non interactif), on favorisera un outil spécialisé dans ce domaine.
- **Ergonomie et niveau d'abstraction de l'interface de développement.** Les outils ne présentent pas tous la même simplicité de développement des robots. Certains ne demandent que très peu de connaissances informatiques et peuvent donc être utilisés par des utilisateurs de type « macro excel », tandis que d'autres nécessitent des compétences de niveau quasi-développeur. Bien entendu, les processus complexes sont ceux qui en général seront les moins accessibles aux outils « *no code* ».
- **Prix et modèle de facturation.** Au-delà du prix des redevances qui varie d'un éditeur à l'autre, il faut également noter que les modèles de facturation diffèrent entre éditeurs. Certains proposent des solutions dont les licences se paient annuellement, tandis que d'autres vendent des licences perpétuelles et ne facturent que le support et la maintenance les années suivantes.
- **Disponibilité avec marchés existants.** Un élément clef dans le choix du logiciel de RPA peut être la possibilité d'acheter des licences sans nécessiter une procédure d'appel d'offre. Plusieurs véhicules contractuels peuvent permettre ceci : le recours au marché multi-éditeur de l'UGAP, dans lequel plusieurs solutions de RPA sont référencées ; ou bien l'utilisation d'un éventuel marché préexistant avec un intégrateur (entreprise de services numériques) qui pourrait les cas échéant revendre des licences de logiciels de RPA, si tant est que le périmètre du marché préétabli est bien respecté.
- **Scalabilité et sécurité** : Alors qu'il est relativement aisé de déployer quelques scripts locaux sur une poignée de postes de travail, le passage à l'échelle est bien plus complexe et nécessite une attention particulière sur les sujets d'orchestration et de sécurité, notamment. Les outils de RPA ne peuvent pas tous être aussi facilement déployés à grande échelle, et selon le périmètre envisagé pour le déploiement (local et temporaire, ou long-terme et généralisé), il conviendra de prendre en compte ce critère.

Une possibilité afin d'affiner le choix de logiciel consiste à lancer des pilotes avec deux ou trois logiciels en parallèle, afin d'en sélectionner un par la suite. La DINUM se tient à votre disposition pour vous éclairer dans votre choix d'éditeur de logiciels de RPA.

## 4. Paramétrage des robots

Une fois décidé quel processus sera automatisé, on entame la phase de développement. On commence tout d'abord par documenter très précisément le processus. Ensuite démarre le paramétrage à proprement parler, que l'on souhaitera le plus modulaire et agile possible.

### Documentation précise du processus

Il est très difficile et hasardeux de développer les robots sans une description détaillée du processus. Par ailleurs, cette description est très précieuse pour maintenir les robots et les faire évoluer. En conséquence, à l'exception des preuves de concept très expérimentales, il est fortement recommandé de s'abstenir de lancer le développement de robots de RPA sans une documentation précise du processus. Cette documentation doit contenir les deux éléments suivants :

- 1) Une cinématique détaillée, sous la forme d'une succession de capture d'écrans. Cette cinématique doit contenir toutes les étapes du processus, et indiquer clairement les champs affectés par chaque étape, grâce à une capture d'écran « avant/après ».
- 2) Une capture d'écran vidéo du processus entier.

En règle générale, la phase de développement amène de légères modifications du processus (souvent, des optimisations). Il est important d'amender la documentation du processus au fur et à mesure de ces modifications, afin que celle-ci reste une image conforme au robot que l'on développe : cela facilitera grandement le maintien en conditions opérationnelles du robot, et d'éventuelles améliorations de celui-ci. Une erreur classique consiste à penser que le développement agile implique une absence de documentation.

### Développement modulaire

Une fois la description précise du processus établie, on peut entamer la phase de développement à proprement parler. Un point préliminaire important : il convient de s'assurer avant d'entamer les développements que l'on dispose d'un environnement de test commun à toutes les applications impliquées dans le processus. A défaut, on prend un risque important au moment de la mise en production du robot, risque qu'il convient d'évaluer précisément selon le processus (un processus en lecture simple est significativement moins risqué qu'un processus en lecture et écriture, par exemple). Une manière simple de remédier à ce risque est de développer sur des machines virtuelles correspondant exactement à l'environnement de production final.

Pour la phase de développement, on donne les quelques conseils suivants :

- Séparer le processus en modules (ou « tâches ») pouvant être testés et utilisés indépendamment les uns des autres. Ces modules seront ensuite intégrés au fur et à mesure dans le processus complet. Des robots complexes peuvent nécessiter le travail parallèle de plusieurs développeurs : dans ce cas, on s'assurera d'autant plus d'une forte modularité des développements.
- Réaliser des sprints d'une semaine environ, ponctués par un point hebdomadaire devant une poignée d'utilisateurs métiers au cours duquel on fait la démonstration des modules ayant été développés au cours de la semaine précédente. En règle générale, le premier sprint dure plutôt 2 semaines, afin d'avoir matière à discussion lors de la première réunion avec les métiers.

- Mettre à disposition de la cellule de développement un utilisateur métier capable de répondre très rapidement à leurs interrogations (idéalement, situé physiquement à proximité). En tout, l'équipe de développement est généralement constituée de 2 ETP (développement et architecture du robot) + 0,5 ETP (métier). Le développement ne dure que quelques semaines, selon la complexité du processus.

## Maintien en conditions opérationnelles

Un robot, une fois paramétré et mis en production, doit être régulièrement mis à jour et adapté, notamment pour suivre les modifications des applications sur lesquelles il s'appuie. Il est impératif de prévoir la charge nécessaire à ce maintien en conditions opérationnelles, afin de ne pas avoir de mauvaises surprises au cours de la durée de vie du robot. Un ordre de grandeur indicatif du temps de travail annuel à prévoir (et donc des budgets) pendant la durée de maintien en conditions opérationnelles est de 20% de la charge de travail totale de paramétrage initial du robot.

## 5. Vision long terme

Il arrive que l'on mobilise des scripts de RPA pour des déploiements très ponctuels, par exemple dans le cas de migration de base de données entre deux SI. En règle générale, les scripts sont toutefois plutôt déployés pour une durée de quelques années, dans l'attente d'évolutions applicatives permettant d'exposer les API nécessaires à une automatisation en back-end. Il convient donc de mettre en place les bons outils de gouvernance, de gestion de projet et de maintien des compétences afin d'assurer la pérennité du déploiement.

## Gouvernance

En règle générale, il est recommandé de créer un pôle de compétences RPA (parfois appelé « centre d'excellence »), ayant pour mission :

- d'assurer la publicité interne des avantages et inconvénients de la technologie ;
- de sélectionner les cas d'usage pertinents ;
- de paramétriser les robots et d'assurer leur déploiement ;
- de gérer la relation avec les éventuels prestataires travaillant sur le sujet RPA ;
- d'assurer le maintien en conditions opérationnelles des scripts ;
- de maintenir une cartographie des SI sur lesquels vont opérer les RPA, afin de veiller aux conséquences éventuelles des mises à jour de ces SI.

Il est important que ce pôle de compétence fasse le trait d'union entre les besoins des métiers et les compétences de la DSI. Un pilotage DSI avec une implication forte des métiers semble donc recommandé. Ce pôle doit bénéficier de l'appui et des orientations du management de la DSI, ce qui signifie en pratique une réunion de pilotage régulière dirigée par un sponsor DSI.

## **Internalisation des compétences**

Il convient de planifier, dès le début du projet de RPA, l'internalisation des compétences de paramétrage et de maintenance des robots, même si ceci est rendu difficile dans certains cas de figure à cause de contraintes de ressources disponibles. La tactique classique d'internalisation consiste à :

- Faire développer les premiers robots par un prestataire extérieur, éventuellement accompagné par des ressources internes entamant leur montée en compétence
- Se faire former, via les ressources disponibles en ligne et éventuellement *via* une prestation extérieure, délivrée par l'éditeur de logiciel ou bien l'intégrateur

A long terme, la cible devrait être une internalisation complète des compétences, avec un recours à des prestations externes limité aux pics ponctuels de charge.

## **Roadmap d'automatisation**

Comme déjà souligné, l'automation des processus *via* le RPA n'est pas une solution idéale de long-terme. On devrait lui privilégier une automatisation reposant sur une urbanisation de l'infrastructure logicielle, l'usage d'APIs et de micro-services. Il est impératif d'intégrer, dès la genèse du projet de RPA, la nécessité d'avoir une *roadmap* d'automatisation de long terme.

## **6. Risques et écueils principaux**

Il est important de connaître les écueils classiques des projets de RPA pour les anticiper, et limiter leur impact en termes de dépassement du délai et du coût du projet.

### **Risques SI**

**Urbanisation** : les scripts de RPA utilisent généralement l'interface utilisateur des applications. Toute modification de celles-ci implique donc une adaptation des scripts de RPA. Le RPA augmente en conséquence la dette technique plutôt qu'il ne la réduit, et freine les évolutions des applications anciennes. Il faut donc être vigilant à ne pas déployer du RPA sur des applications évoluant très rapidement. Par ailleurs, l'inflation du nombre de robots opérant sur de nombreuses applications peut rendre leur gestion quotidienne très complexe. Il faut éviter tout phénomène de « sac de noeuds » et garder une vision et une cartographie claire de l'architecture globale des scripts.

**Charge supplémentaire sur les applications** : les robots effectuent un nombre élevé de requêtes sur les interfaces d'application, à des fréquences souvent inhabituelles. Il faut être très vigilant à ne pas rendre cette charge insupportable pour ne pas provoquer des pannes d'applications qui pourraient être critiques. Il faut donc estimer en amont les charges acceptables et prendre des marges de sécurité conservatrices.

**Captivité technologique** : il n'y a pas d'interopérabilité entre les solutions de RPA, ce qui signifie qu'un passage d'un logiciel à l'autre nécessite le redéveloppement de tous les robots, ce qui peut être coûteux et long. Afin de réduire ce risque, il convient de s'attacher à bien documenter les processus automatisés, et de garantir que les scripts s'inscrivent dans une *roadmap* d'automatisation plus globale, afin de s'assurer que leur durée de vie soit limitée dans le temps.

## Sécurité

**Habilitations des robots :** créer des habilitations pour les scripts peut poser problème dans les organisations habituées à ne délivrer des habilitations qu'à des agents. Afin de pouvoir analyser en détail le comportement des scripts, il est recommandé d'attribuer à chaque robot un compte utilisateur par application.

**Environnement de développement et test :** pour tester le bon fonctionnement des robots, il est pratique de disposer d'un environnement de test commun à toutes les applications du processus automatisé. A défaut, on utilise un environnement de test distinct pour chaque application utilisée, et on réalise la jonction du robot en production. Quoi qu'il en soit, les environnements de test doivent présenter une copie exacte des interfaces des applications et des échantillons représentatifs des données de la production. Par ailleurs, certaines briques des logiciels de RPA doivent être installées sur serveur ou sur machine virtuelle ce qui peut nécessiter du travail pour les personnes responsables de l'architecture à la DSI.

## Risque métier

**Processus mal défini :** un problème classique des démarches d'automatisation survient lorsque l'on change significativement, en cours de route, le processus que l'on souhaite automatiser. Ceci survient en général lorsque l'on ne s'est pas assuré de la stabilité dans le temps du processus en question, ou bien du fait qu'il soit standard dans l'organisation. Il ne faut pas hésiter à vérifier auprès de plusieurs experts métiers que la cinématique du processus est la bonne avant d'entamer les travaux de paramétrage. En sus, il faut s'assurer en amont que les optimisations éventuelles du processus ont bien été réalisées.

**Exceptions et erreurs :** il est crucial, au cours du paramétrage du robot, de s'assurer du bon traitement des cas particuliers. Cela signifie notamment de porter une attention particulière sur toutes les exceptions et erreurs éventuelles du robot, et de s'assurer que celles-ci sont correctement réorientées vers un traitement manuel. Cela implique de mettre en place une traçabilité des dossiers ou cas traités par le robot, souvent sous la forme d'un fichier tableur.

**Dépendance au robot :** une fois le script en production, les agents qui réalisaient le processus automatisé sont amenés à travailler sur d'autres tâches. En cas de dysfonctionnement ou de panne soudaine du robot, les équipes métier rencontrent un pic de charge difficilement surmontable, voire même ont perdu la connaissance exacte du processus. Afin de limiter ce risque, il convient de maintenir une équipe technique chargée de la maintenance des robots, adopter une démarche proactive vis-à-vis des *roadmaps* des applications impliquées dans les processus, et documenter précisément les scripts.

**Retour sur investissement :** une surestimation des économies générées par le robot, couplée à une sous-estimation des frais de développement de celui-ci peut plomber le retour sur investissement du projet. Afin de limiter ces risques, il convient de respecter scrupuleusement les critères d'éligibilité des processus et de simplifier le processus au maximum avant de l'automatiser. Par défaut, il est déconseillé de développer un robot permettant d'épargner moins de l'équivalent d'un ETP.

## Risque juridique

**Traitement des données personnelles :** Si des données personnelles sont impliquées, il convient de noter que le traitement automatique de celles-ci peut avoir un statut juridique distinct de leur traitement manuel et nécessiter une analyse d'impact à la protection des données. En conséquence, il est important de

prendre contact dès le début du projet avec le délégué à la protection des données référent. Il faut savoir qu'un dossier d'analyse d'impact peut être très chronophage et initier un processus de plusieurs mois.

## Conclusion

Le RPA est un outil permettant de combler des lacunes temporaires dans les *roadmap* d'automatisation des processus et de modernisation des SI. Déployé convenablement, il peut permettre de dégager d'importantes marges de productivité à moindre coût, et de réduire les irritants subis par les fonctions métiers.