

BIENVENUE



TECHNIFUTUR®
CENTRE DE COMPETENCES



L'UNION EUROPEENNE ET LA WALLONIE
INVESTISSENT DANS VOTRE AVENIR

AUTOMATISATION DES PROCESSUS MÉTIER LIÉS AU DIGITAL

Le Citoyen Développeur et l'automatisation des processus

PRÉSENTATION

Gauthier Pierard



Maurizio Napoli

- ▶ Donner les outils pour aux personnes pour qu'ils puissent faire évoluer d'eux-mêmes leur métier lié au digital.

CATALYSTRPA

Contact@catalystrpa.com

AGENDA

1. Introduction: le travail répétitif et ses conséquences

- ▶ Pourquoi il est si commun
- ▶ Pourquoi s'en soucier
- ▶ Notre formation

2. La formation

- ▶ Création de scripts simples en R
- ▶ La RPA et Création de Robots basiques avec UIPath
- ▶ Robots Web avec Python et Selenium
- ▶ Développer des applications Web axées RPA avec Shiny

AGENDA

1. Introduction: le travail répétitif et ses conséquences

▶ Pourquoi il est si commun

- ▶ Pourquoi s'en soucier
- ▶ Notre formation

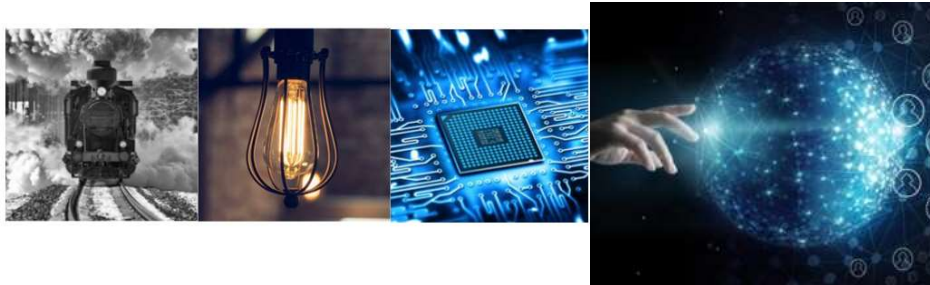
2. La formation

- ▶ Création de scripts simples en R
- ▶ La RPA et Création de Robots basiques avec UIPath
- ▶ Robots Web avec Python et Selenium
- ▶ Développer des applications Web axées RPA avec Shiny



POURQUOI LE TRAVAIL RÉPÉTITIF EST-IL SI COMMUN?

- ▶ Comment en est-on arrivé à laisser dans toutes les industries un si grand nombre de personnes perdre du temps à exécuter des procédures sans grande valeur ajoutée, et qui restent stagnantes pendant parfois des décennies?
- ▶ Nous sommes pourtant au cœur de la 4^e révolution industrielle



Quatrième Révolution Industrielle

La quatrième révolution industrielle est issue de la numérisation de l'information et de l'interconnectivité des appareils mobiles connectés. Bien que déjà visible dans ses réalisations comme l'intelligence artificielle avec les voitures autonomes, l'impression 3D, la nanotechnologie ou l'information quantique, nous en sommes qu'au début. Mais en comparaison avec les précédentes, elle évolue à un rythme exponentiel et non plus linéaire. Elle transformera tous nos systèmes de production, de management et de gouvernance. Tous les travaux intellectuels mais répétitifs peuvent être remplacés par la robotisation des processus du travail et ils le seront car ils représentent un gain considérable de productivité mais également d'amélioration de condition de travail et l'émergence de nouveaux métiers.



POURQUOI LE TRAVAIL RÉPÉTITIF EST-IL SI COMMUN?

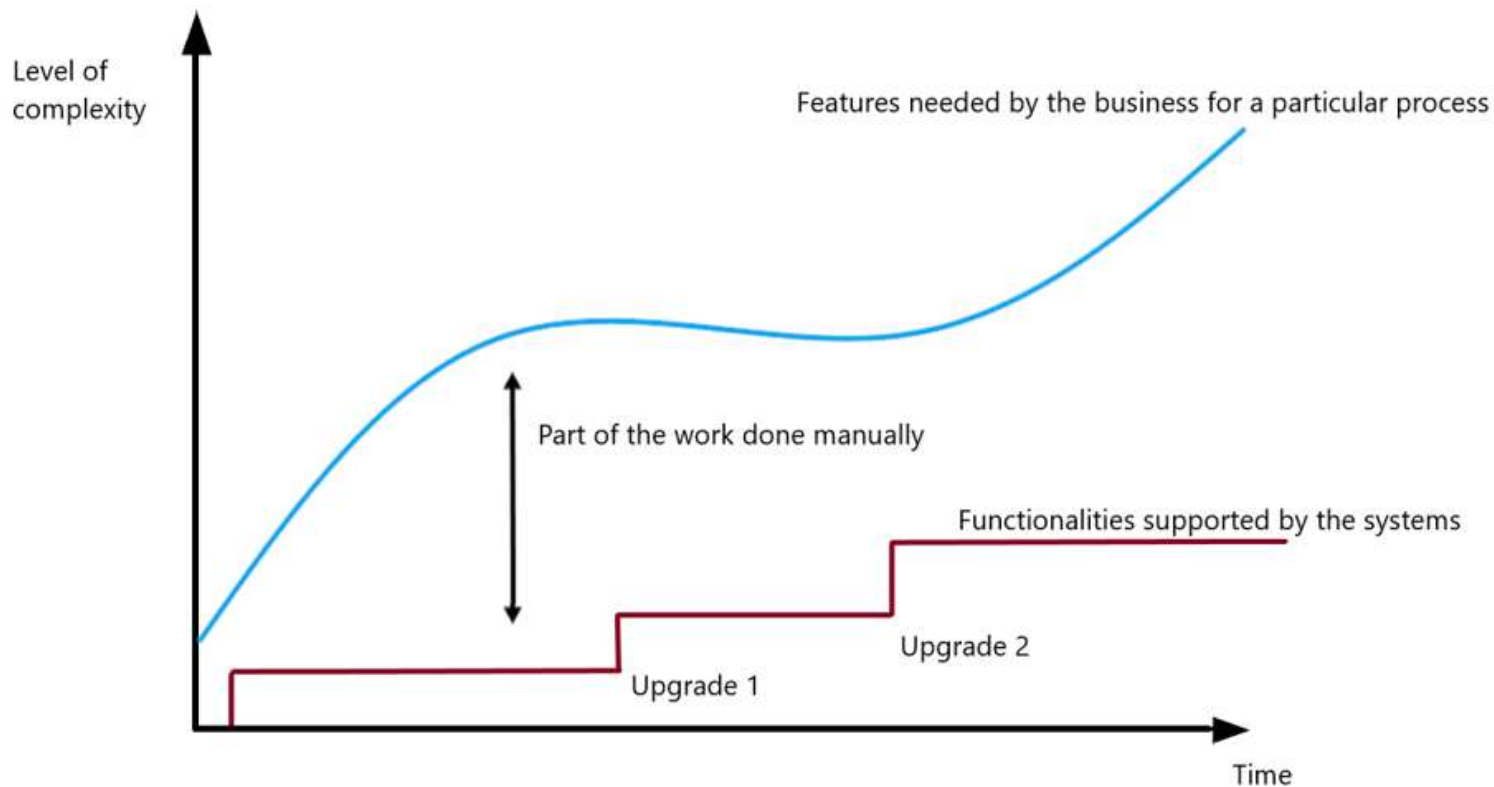
Beaucoup d'entreprises sont en retard par rapport à ce que cette 4^e révolution peut offrir.

- ▶ Utilisation de progiciels.
- ▶ Spécificité du mode de fonctionnement de chaque entreprise.
- ▶ Evolution permanente des besoins.
- ▶ La connaissance des besoins spécifiques est détenue par le métier.
- ▶ D'un côté on a des progiciels figés et lents et de l'autre des processus qui évoluent en permanence, donc inévitablement, il y a création d'un gap qui augmente au fil du temps.



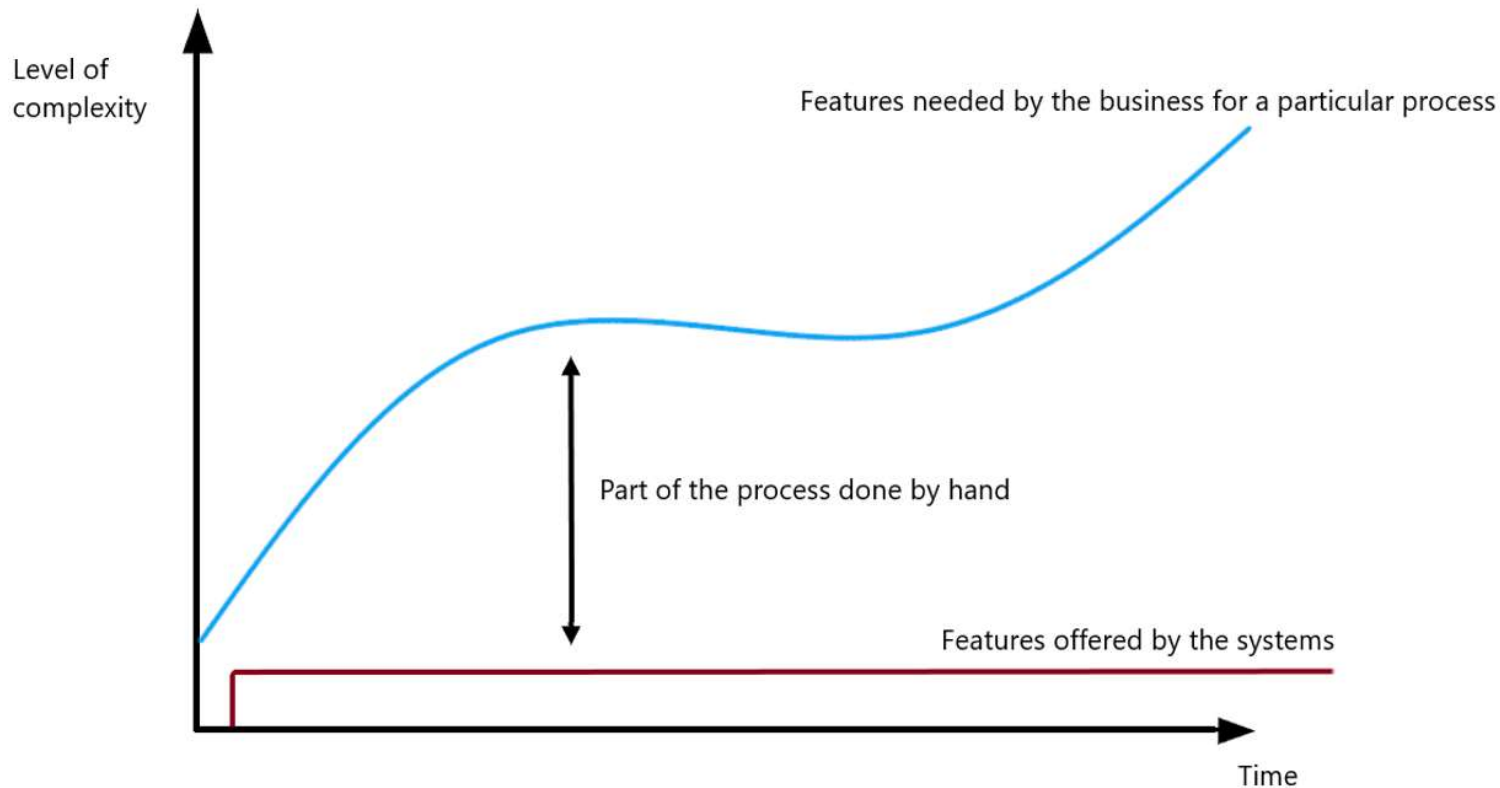
POURQUOI LE TRAVAIL RÉPÉTITIF EST-IL SI COMMUN?

- ▶ Il y a toujours un écart entre ce qui est fait en pratique et ce pour quoi les systèmes de l'entreprise sont conçus.
- ▶ Progiciels vs. vie réelle: Entreprises matures



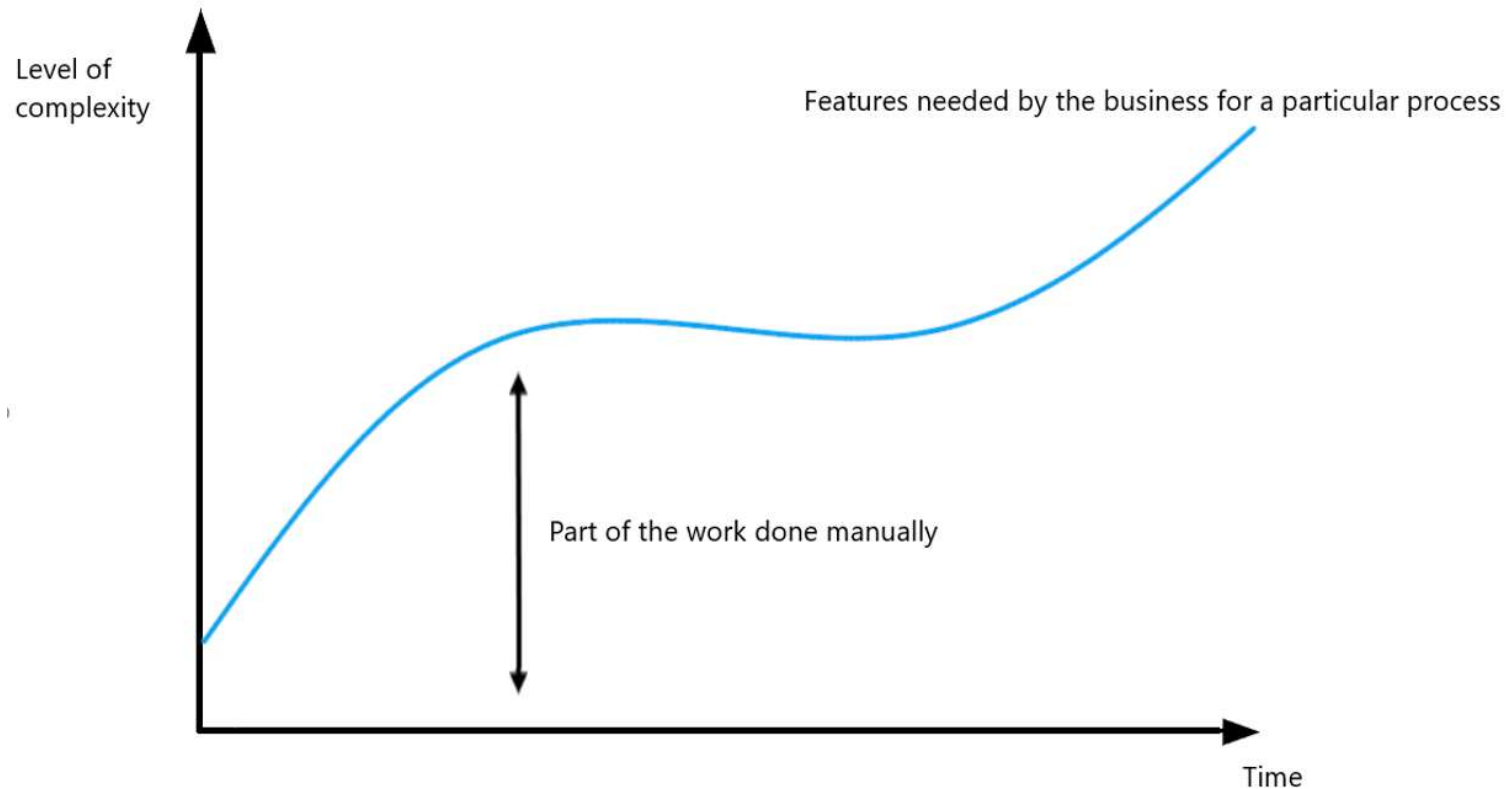
POURQUOI LE TRAVAIL RÉPÉTITIF EST-IL SI COMMUN?

- ▶ Il y a toujours un écart entre ce qui est fait en pratique et ce pour quoi les systèmes de l'entreprise sont conçus.
- ▶ Progiciels vs. vie réelle: scénario 2



POURQUOI LE TRAVAIL RÉPÉTITIF EST-IL SI COMMUN?

- ▶ Il y a toujours un écart entre ce qui est fait en pratique et ce pour quoi les systèmes de l'entreprise sont conçus.
- ▶ Progiciels vs. vie réelle: scénario 3



INTRODUCTION

Les tâches répétitives en entreprise.

► Pourquoi existent-elles?

► Pourquoi s'en soucier?

► Quelles alternatives?

► Evolution constante des processus métier

► L'environnement externe est changeant

► Spécificité de ces processus métier

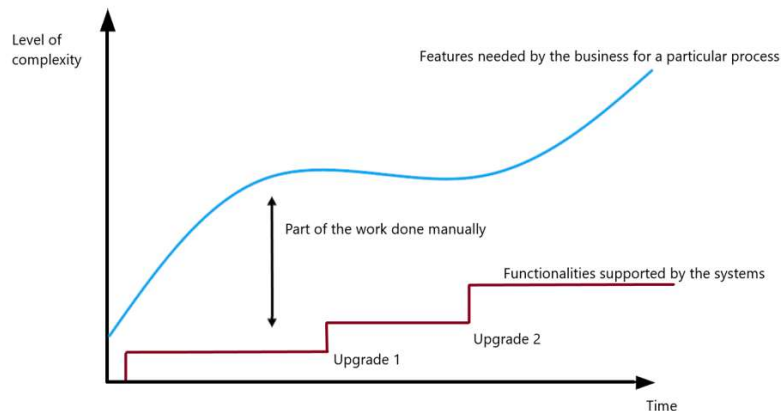
► Propre à chaque compagnie, chaque département, chaque client

► Durée et coût des mises à jour.

► Transcription des besoins

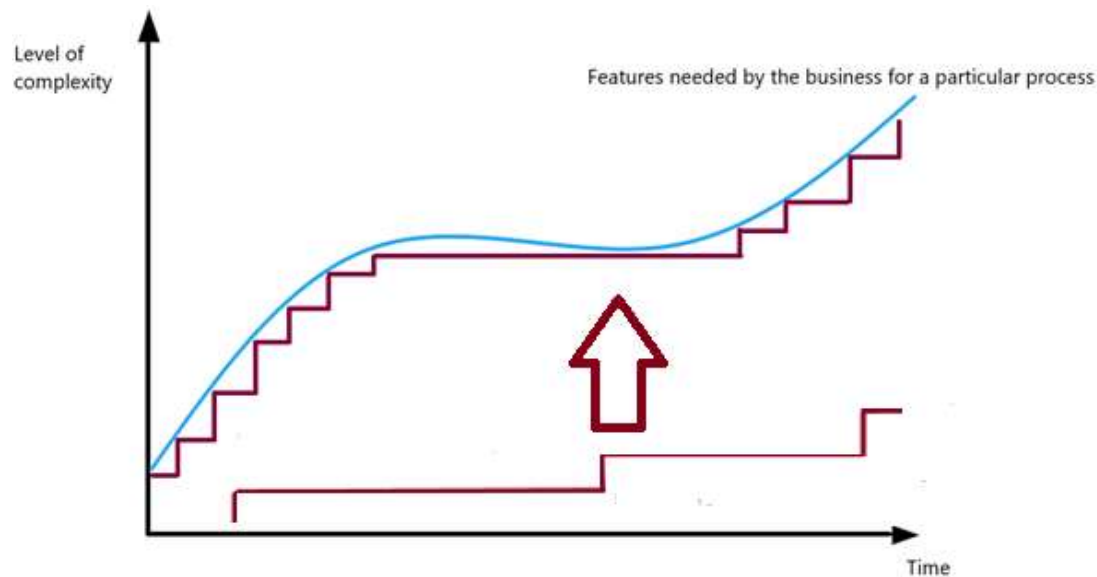
► Phase de test

► Déploiement



NOTRE PROPOSITION: LE CITOYEN DÉVELOPPEUR

- ▶ Permettre aux utilisateurs d'effectuer de petits développements simples permettant de pallier quand c'est nécessaire aux évolutions du processus
- ▶ De manière accompagnée et transparente
- ▶ Permet de réduire le travail manuel et de permettre aux employés de se consacrer à d'autres tâches, tout en améliorant le degré de contrôle.



AGENDA

1. Introduction: le travail répétitif et ses conséquences

- ▶ Pourquoi il est si commun
- ▶ Pourquoi s'en soucier
- ▶ Notre formation

2. La formation

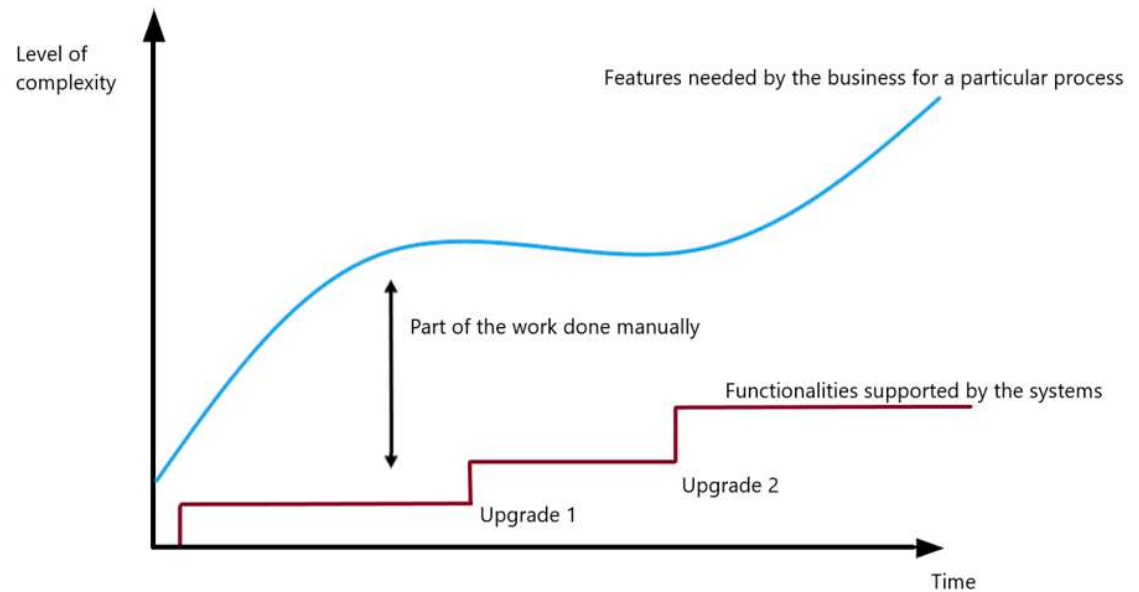
- ▶ Création de scripts simples en R
- ▶ La RPA et Création de Robots basiques avec UIPath
- ▶ Robots Web avec Python et Selenium
- ▶ Développer des applications Web axées RPA avec Shiny



CONSÉQUENCES

Pour l'entreprise

- Rentabilité sur le temps de travail
- Risques d'erreur
- Manque d'efficacité



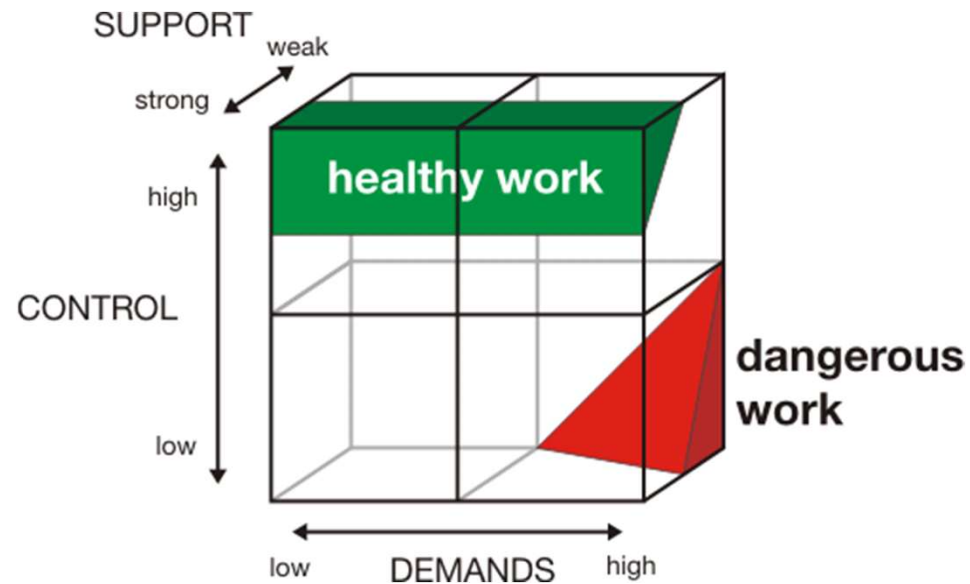
Pour l'employé



CONSÉQUENCES

Pour l'entreprise

Pour l'employé

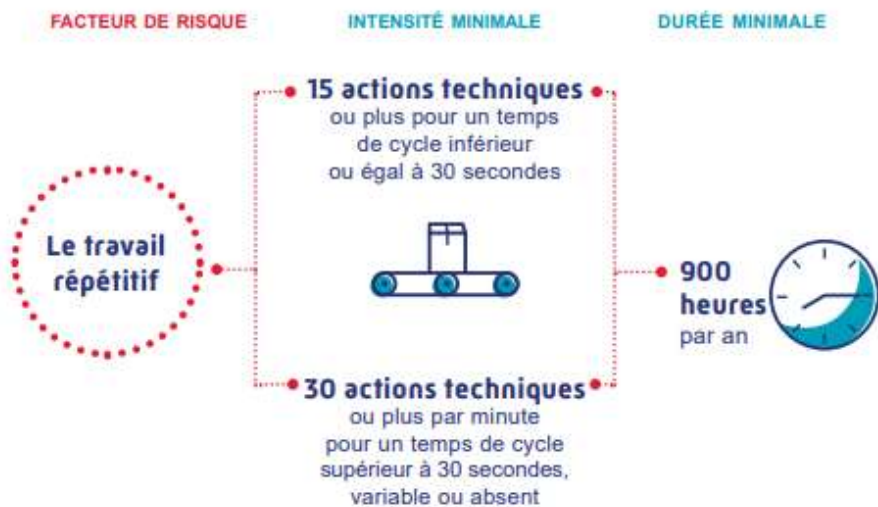


INTRODUCTION

- ▶ En France et en Belgique, le travail répétitif est identifié comme risque professionnel
- ▶ En France, des mesures sont prises pour permettre à l'employé de diminuer ce risque
- ▶ Toutefois, cela concerne quasi-uniquement le travail manuel

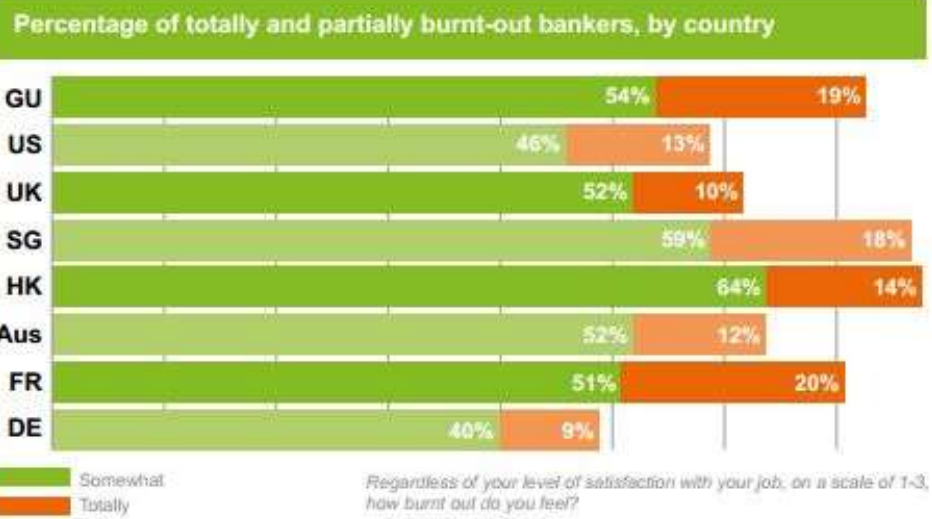
Le Compte personnel prévention pour les salariés

L'exposition à un ou plusieurs facteurs de risques professionnels est susceptible de laisser des traces durables, identifiables et irréversibles sur la santé.



INTRODUCTION

► Pour autant, les conséquences du travail répétitif sont-elles insignifiantes pour le travail de bureau?



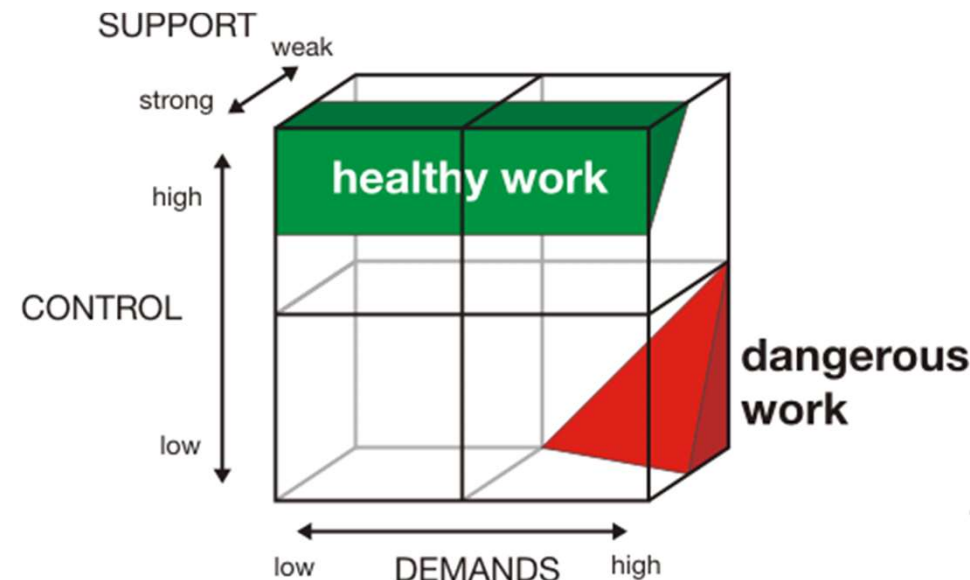
INTRODUCTION

- Pour autant, les conséquences du travail répétitif sont-elles insignifiantes pour le travail de bureau?

Job strain model (1990)

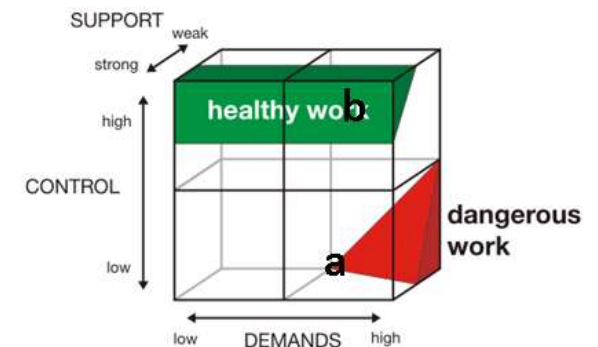
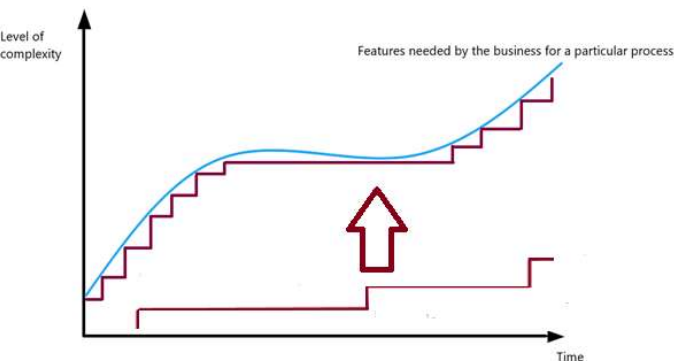
- Niveau d'exigence du travail
 - Pas forcément un mauvais facteur
- **Degré de contrôle de l'individu**
 - Variété
 - Apprentissage continu
 - Créativité
 - Latitude de décision
 - Possibilité de s'exprimer
- Degré de soutien social
 - Du manager ou des collègues

"Job strain model" de Karasek



CONCLUSION

- ▶ Augmentation de compétitivité par pour l'entreprise
 - ▶ Meilleure automatisation et allocation du travail
 - ▶ Réduction du taux d'erreurs
 - ▶ Idéation
 - ▶ Satisfaction au travail
- ▶ Amélioration de la motivation et des compétences de l'employé
 - ▶ Augmente le degré de contrôle du travailleur sur ses processus
 - ▶ Augmente le degré de support
 - ▶ Augmente l'exigence sur le travail dans un premier temps, pour la réduire ensuite



AGENDA

1. Introduction: le travail répétitif et ses conséquences

- ▶ Pourquoi il est si commun
- ▶ Pourquoi s'en soucier
- ▶ Notre formation

2. La formation

- ▶ Introduction a la programmation
- ▶ Création de scripts simples en R
- ▶ La RPA et Création de Robots basiques avec UIPath
- ▶ Robots Web avec Python et Selenium
- ▶ Développer des applications Web axées RPA avec Shiny



AGENDA

1. Introduction: le travail répétitif et ses conséquences
2. La formation

▶ Introduction a la programmation

- ▶ Création de scripts simples en R
- ▶ La RPA et Création de Robots basiques avec UIPath
- ▶ Robots Web avec Python et Selenium
- ▶ Développer des applications Web axées RPA avec Shiny



INTRODUCTION À LA PROGRAMMATION

Pourquoi:

- ▶ L'idée de programmation a pour but de « **transformer** » quelque chose (Input) en autre chose (output). => données d'entrée et résultat
- ▶ Le programme est la somme d'un code (algorithme) appliqué à des données.
- ▶ Cette transformation est constituée d'une décomposition logique et séquencée d'opérations moins complexes.



INTRODUCTION À LA PROGRAMMATION

Définitions:

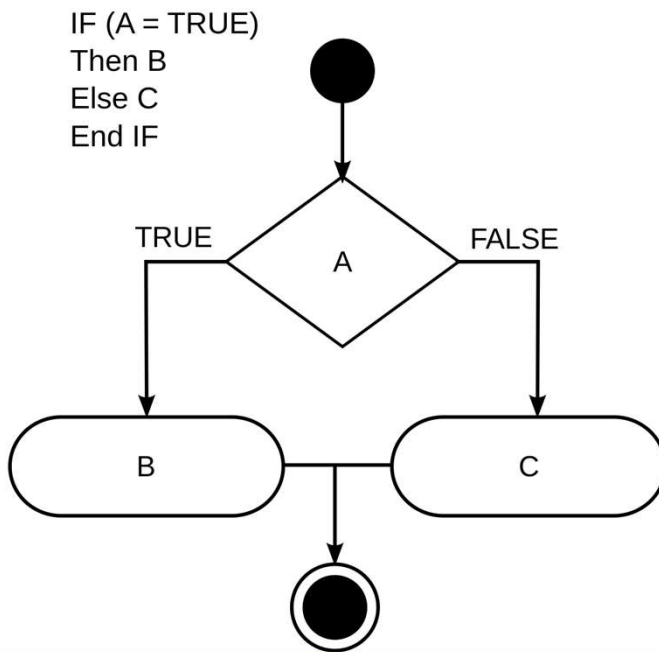
- ▶ **PROGRAMMATION:** Capacité de pouvoir faire exécuter une séquence d'opérations à une entité (machine, être vivant...)
- ▶ **OPERATION:** action unitaire produisant un effet
 - ▶ Opération mathématique (addition, soustraction, multiplication, division...)
 - ▶ Opération de comparaison ($=$, $<>$, $>$, $<$)
 - ▶ Opération logique (ET, OU)
- ▶ Ces opérations ne forment une séquence de programmation que si elles ont un **lien**, un **ordre** pertinent permettant d'accomplir une tâche complexe.
- ▶ Il y a plusieurs types d'enchaînement entre ces actions



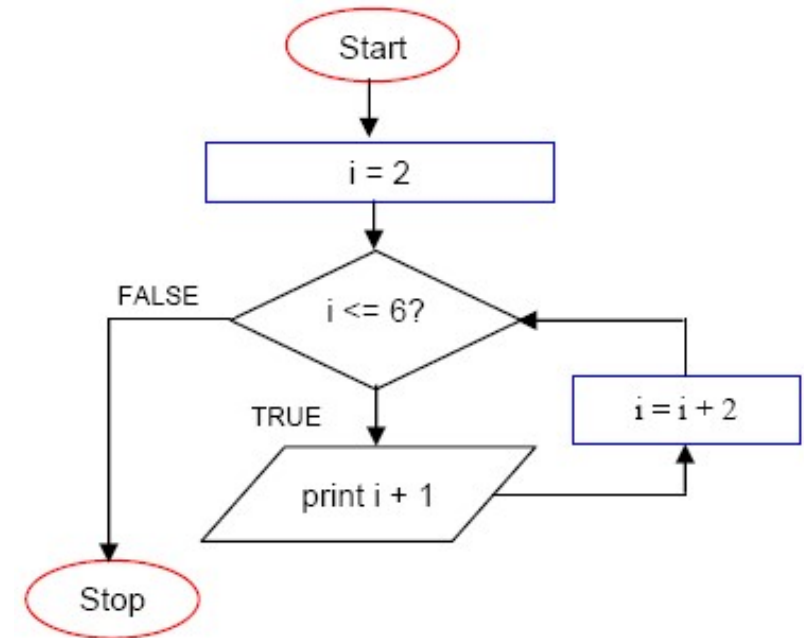
INTRODUCTION À LA PROGRAMMATION

Enchainements:

- **Condition:** C'est la capacité de poser une condition ayant 2 issues.
(positive ou négative) « **Si** »



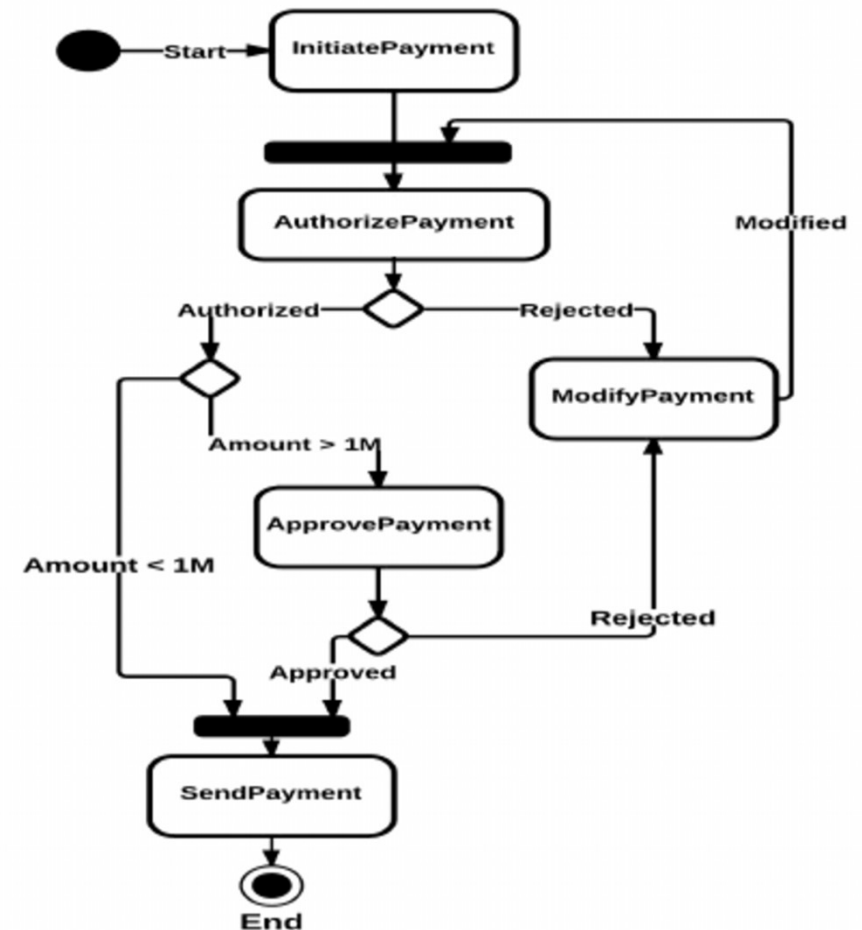
- **Boucle:** répétition d'une séquence d'opération sur un élément jusqu'à une condition de rupture.



INTRODUCTION À LA PROGRAMMATION

Enchainements d'enchainements:

- ▶ L'imbrication et le séquençement de ces opérations enchainées permettent de reproduire un processus aussi appelé **algorithme**.
- ▶ Cet algorithme va **mimer la pensée humaine** après traduction dans un langage « machine ».
- ▶ C'est donc ensuite la **machine** qui va **exécuter** ce **schéma de pensée** et produire le **résultat** attendu.

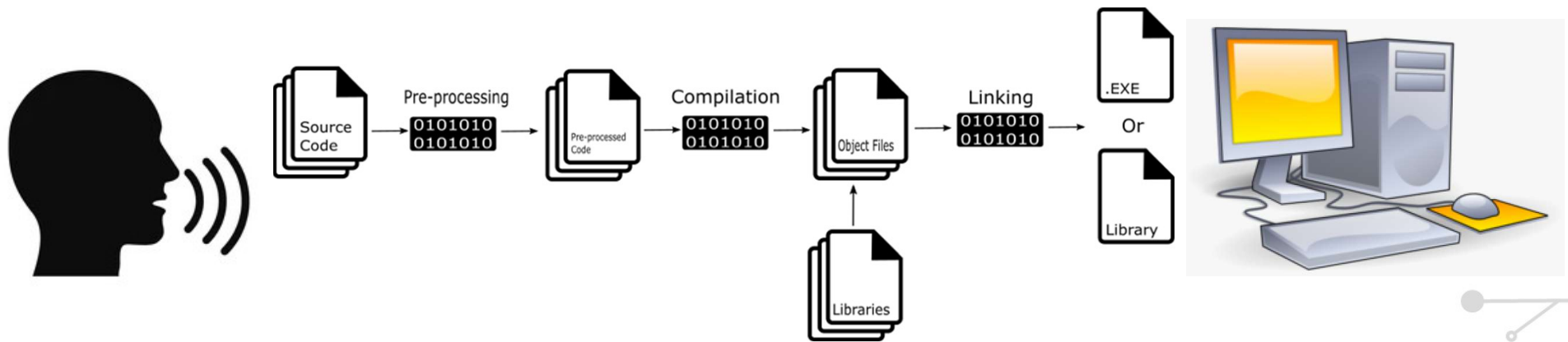


INTRODUCTION À LA PROGRAMMATION

Comment:

- ▶ Pour qu'une machine comprenne ce que l'on désire qu'elle face, il faut lui parler machine!!! 001101010011110
- ▶ Personne ne peut parler machine. Donc il faut trouver une langue qui puisse être traduite en langage machine => langage de programmation

| | | |
|----------------|---------|--------|
| (Visual) Basic | C / C++ | R / S |
| Python | C# | Golang |
| Java | SQL | ... |



INTRODUCTION À LA PROGRAMMATION

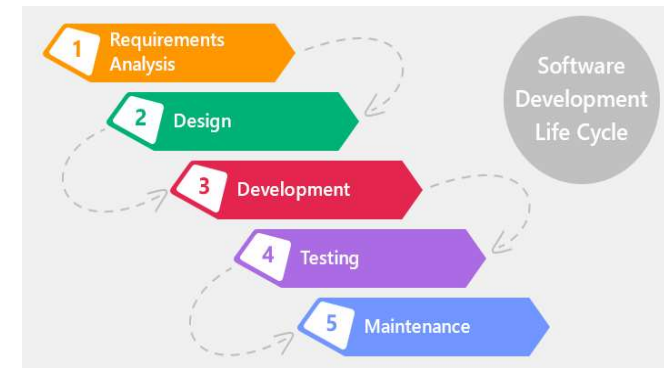
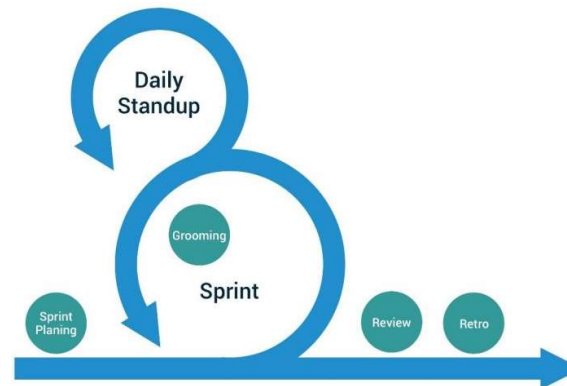
Processus de gestion:

Le processus de développement d'un programme suit des règles de bases:

- Analyse du problème
- Développement de la solution
- Test de la solution
- Exploitation de la solution

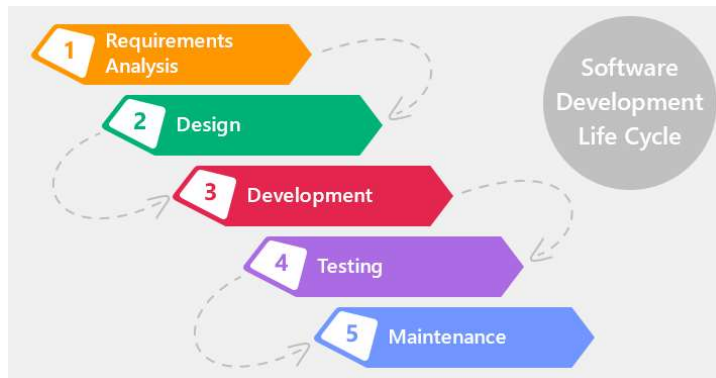
- Ces règles si elles sont appliquées avec des contraintes de temps différentes

- Waterfall
- Agile / Scrum
- Six Sigma



CONCLUSION - PROGRAMMATION

- ▶ Introduction à la programmation
 - ▶ En résumé, on va se concentrer dans cette formation sur quelques notions clés
 - ▶ Définition processus vs programme
 - ▶ Principes de modélisation de processus (analyse des besoins et design)
 - ▶ Séparation entre environnements (dev, testing, maintenance...)
 - ▶ Waterfall vs Agile (Scrum)

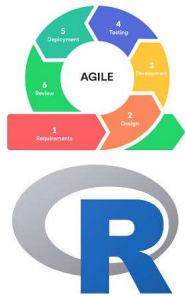


AGENDA

1. Introduction: le travail répétitif et ses conséquences
2. La formation
 - ▶ Introduction a la programmation

▶ Création de scripts simples en R

- ▶ La RPA et Création de Robots basiques avec UIPath
- ▶ Robots Web avec Python et Selenium
- ▶ Développer des applications Web axées RPA avec Shiny

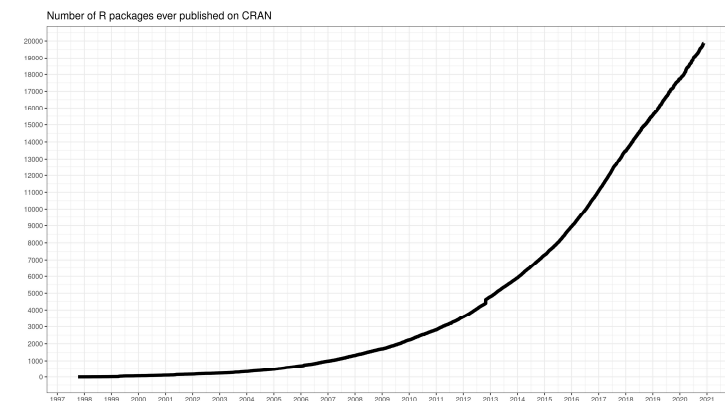


LE LANGAGE R

- ▶ Créé en 1993 par Ross Ihaka and Robert Gentleman à Auckland



- ▶ Totalement Opensource et libre d'utilisation dans le domaine commercial.
- ▶ Nombre de fonctionnalités (librairies) croissant
 - ▶ Outil multitâche



PARENTHÈSE: POURQUOI R?

- ▶ Fonctionnel
- ▶ Adapté au style procédural
- ▶ Expressif et concis, facile à apprendre
- ▶ En évolution constante, de manière rapide et stable



EXCEL COMME PORTE D'ENTRÉE VERS LA PROGRAMMATION EN R

- Exemple: La fonction « Somme.si » dans Excel et son équivalent dans R.

| | Value | Type |
|----|---------------------------|------|
| 6 | | |
| 7 | 1 | A |
| 8 | 2 | B |
| 9 | 3 | C |
| 10 | 4 | A |
| 11 | 5 | B |
| 12 | | |
| 13 | =SUMIF(C7:C11,"A",B7:B11) | |

```
1 value <- 1:5
2 Type <- c('A','B','C','A','B')
3 sum(value[Type=="A"])
4
```

- R est-il vraiment plus compliqué que les formules d'Excel ?

EXEMPLE

Imaginons un exemple simplifié de processus métier:

- ▶ Les commandes arrivent via e-mail

Hello, I need to create an order for 2589 USD, thanks

- ▶ En différentes dénominations (EUR, USD, JPY...)
- ▶ Un certain nombre de règles existent
 - ▶ Le montant doit représenter plus de 100 EUR et être multiple de 5 cent
- ▶ Un employé doit
 - ▶ Collecter ces e-mails
 - ▶ Convertir chaque montant en euro
 - ▶ Vérifier que les tests sont valides
 - ▶ En faire un résumé dans Excel et l'envoyer au management de manière périodique.

Gauthier Pierard
Order 5
Hello, I need to create an

Gauthier Pierard
Order 4
Hello, I need to create an

Gauthier Pierard
Order 3
Hello, I need to create an

Gauthier Pierard
Order 2
Hello, I need to create an

Gauthier Pierard
Order 1
Hello, I need to create an

EXEMPLE

- ▶ Rassembler les e-mails concernés
- ▶ Les commandes sont lues depuis Outlook

```
orders <- list()
for (i in 1:nbemails) {
  subject <- emails(i)$subject(1)
  if(grep("order", subject)) {
    print(subject)
    orders[i] <- emails(i)$Body() # adding email body to list
  }
}
```

```
> orders
[[1]]
[1] " Hello, I need to create an order for 456000 JPY, thanks \r\n"

[[2]]
[1] " Hello, I need to create an order for 987 EUR, thanks \r\n"

[[3]]
[1] " Hello, I need to create an order for 2589 USD, thanks \r\n"
```

```
> df
      amounts currencies
1 456000.00      JPY
2   987.00      EUR
3  2589.00      USD
4   523.00      CAD
```

EXEMPLE

Les commandes sont converties en euro

```
> df
  amounts currencies
1 456000.00      JPY
2   987.00       EUR
3  2589.00      USD
4   523.00      CAD
```

```
> fxrates
  currency  rate
1      JPY 0.0075
2      EUR 1.0000
3      USD 0.8200
4      CAD 0.6800
```

```
# vlookup function -----
fxrates <- read.xlsx('fxrates.xlsx')
vlookup <- function(lookupvalue, tablearray, col_index) {  
df$fxrates <- vlookup(df$currencies, fxrates, 2)  
df$amounts.EUR <- df$fxrates*df$amounts
```

```
> df
  amounts currencies fxrates amounts.EUR
1 456000.00      JPY  0.0075    3420.00
2   987.00       EUR  1.0000     987.00
3  2589.00      USD  0.8200    2122.98
4   523.00      CAD  0.6800     355.64
5   456.25       EUR  1.0000     456.25
6    23.00      USD  0.8200      18.86
```

EXEMPLE

- ▶ On vérifie que les règles sont respectées
 - ▶ Règle 1: Le montant en Euro doit être multiple de 5 cent
 - ▶ Règle 2: Le montant doit être supérieur a 100 EUR

```
df$ismultiple <- (df$amounts.EUR/0.05)%%1==0
df$abovelimit <- df$amounts.EUR >= 100
df$tests_passed <- df$ismultiple&df$abovelimit
```

```
> df
  amounts currencies fxrates amounts.EUR ismultiple abovelimit tests_passed
1 456000.00      JPY  0.0075      3420.00      TRUE      TRUE      TRUE
2   987.00      EUR  1.0000      987.00      TRUE      TRUE      TRUE
3  2589.00      USD  0.8200     2122.98     FALSE      TRUE     FALSE
4   523.00      CAD  0.6800      355.64     FALSE      TRUE     FALSE
5   456.25      EUR  1.0000      456.25      TRUE      TRUE      TRUE
6    23.00      USD  0.8200       18.86     FALSE     FALSE     FALSE
```


EXAMPLE

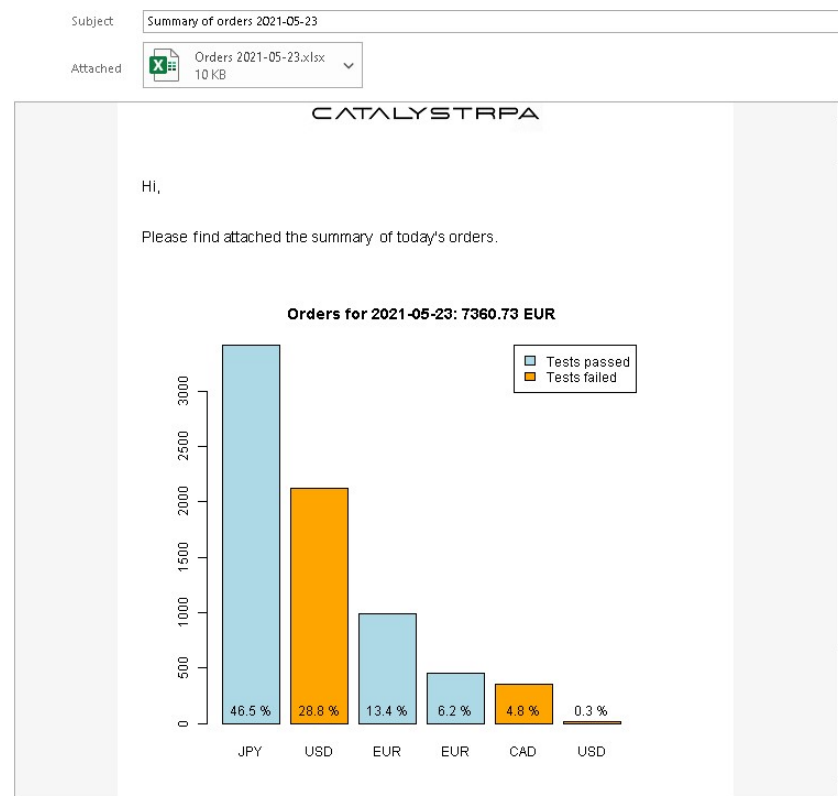
Parenthèse: defensive programming: la clé pour avoir un code robuste.

▶ En particulier, le principe de la vérification constante.

```
# Document, and comment -----  
# Verify everything -----  
file.exists('fxrates.xlsx')  
assertthat::assert_that(!anyNA(result), msg="Missing values in the result table")  
assertthat::assert_that(is.numeric(result$amounts), msg="Non-numeric values in the amounts")  
  
# keep functions short -----  
vlookup <- function(lookupvalue, tablearray, col_index) {  
  first_matching_row <- match(lookupvalue, tablearray[,1])  
  result <- tablearray[first_matching_row, col_index]  
  return(result)  
}  
  
# use a consistent style -----  
# refer to external functions explicitly -----  
# library() > require() -----  
# manage dependencies -----  
# design everything as a package -----  
# use meaningful names -----  
# know the rules and when to break them -----
```

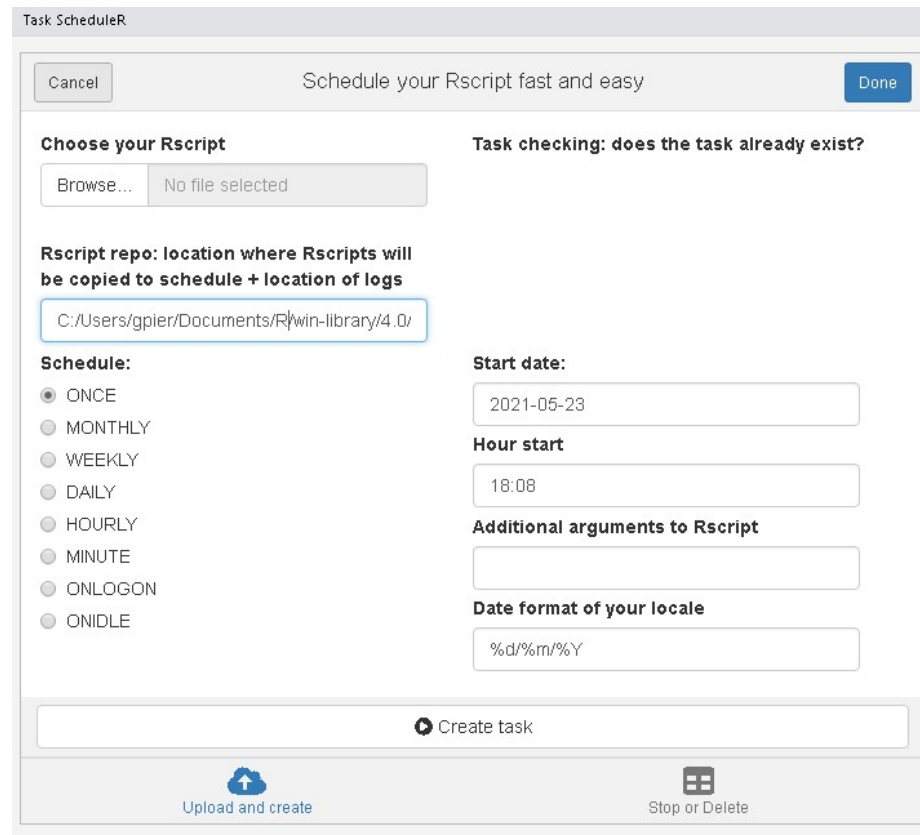
EXEMPLE

On fait un résumé dans Excel et on l'envoie.



EXAMPLE

Comme tout programme, on peut en planifier l'exécution.



The screenshot shows the 'Task SchedulerR' dialog box with the title 'Schedule your Rscript fast and easy'. It includes a 'Cancel' button and a 'Done' button. The 'Choose your Rscript' section has a 'Browse...' button and a text field showing 'No file selected'. The 'Rscript repo: location where Rscripts will be copied to schedule + location of logs' section has a text field with the path 'C:/Users/gpier/Documents/R/win-library/4.0/'. The 'Schedule:' section has radio buttons for 'ONCE', 'MONTHLY', 'WEEKLY', 'DAILY', 'HOURLY', 'MINUTE', 'ONLOGON', and 'ONIDLE'. The 'Task checking: does the task already exist?' section is empty. The 'Start date:' section has a text field with '2021-05-23'. The 'Hour start' section has a text field with '18:08'. The 'Additional arguments to Rscript' section has a text field. The 'Date format of your locale' section has a text field with '%d/%m/%Y'. At the bottom, there is a 'Create task' button and two icons: 'Upload and create' and 'Stop or Delete'.

AGENDA

1. Introduction: le travail répétitif et ses conséquences

2. La formation

▶ Introduction a la programmation

▶ Création de scripts simples en R

▶ La RPA et Création de Robots basiques avec UiPath

▶ Robots Web avec Python et Selenium

▶ Développer des applications Web axées RPA avec Shiny



INTRODUCTION UIPATH

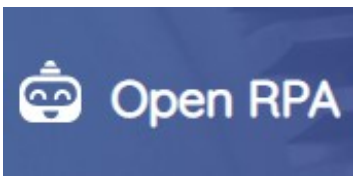
- ▶ Compagnie Roumaine Fondée en 2005 à Bucharest par Daniel Dines and Marius Tîrcă



- ▶ Logiciel closed-source payant (pour les entreprises commerciales) qui ne nécessite pas de connaissance en programmation.
- ▶ Les programmes sont créés en glissant des activités et les agençant les unes aux autres
- ▶ L'outil typique de ce qu'on appelle la « RPA »
- ▶ La seule chose qui est différente entre les outils de RPA et de programmation classiques est la capacité à « espionner » et à manipuler des Interfaces utilisateur sous Windows.

INTRODUCTION RPA

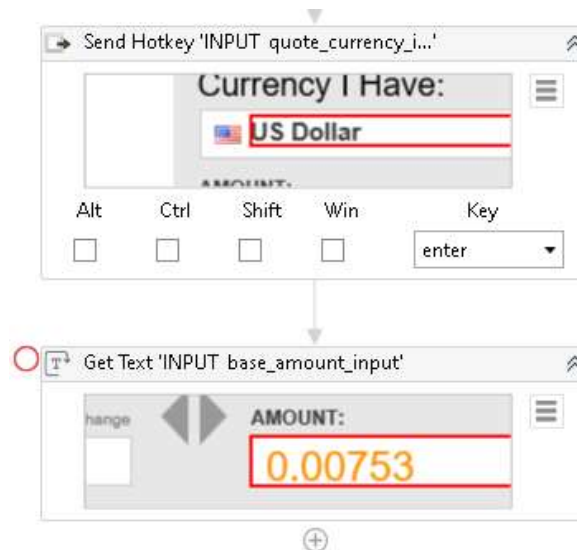
- ▶ Les outils de RPA comme UiPath, BluePrism, OpenRPA... permettent la création de programmes sans utilisation de code.
- ▶ A part cette caractéristique « no-Code », la seule chose qui est véritablement différente entre les outils de RPA et de programmation classiques est leur capacité à « espionner » et à manipuler des Interfaces utilisateur sous Windows.
- ▶ Ceci permet de créer des « Robots » qui vont mimer les actions d'un utilisateur.



EXEMPLE

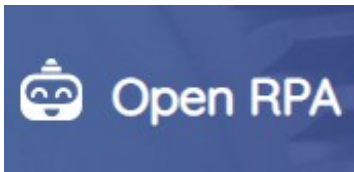
Cas 2: avec R et UiPath

- ▶ Imaginons qu'au lieu d'avoir les taux de change dans un fichier Excel, il faille utiliser une application avec une Interface Utilisateur (UI) comme Chrome.
- ▶ Pour chaque devise, UiPath va rechercher le taux correspondant sur le web



CONCLUSION RPA

- ▶ UiPath et les autres outils de RPA sont très utiles pour naviguer dans des UI's grâce à leur capacité à « espionner » les différents éléments de ces applications.
- ▶ Toutefois ils sont plus lents qu'un langage de programmation classique, et leurs fonctionnalités sont plus limitées.
- ▶ Ils sont également plus consommateurs en ressources
- ▶ UiPath est un programme relativement coûteux (Minimum 2000 EUR/an pour un user et un poste), mais qui propose une édition Community gratuite.
- ▶ Des solutions Opensource sont en développement comme OpenRPA, toutefois les possibilités en sont plus limitées.



AGENDA

1. Introduction: le travail répétitif et ses conséquences
2. La formation
 - ▶ Introduction a la programmation
 - ▶ Création de scripts simples en R
 - ▶ La RPA et Création de Robots basiques avec UiPath
 - ▶ Robots Web avec Python et Selenium
 - ▶ Développer des applications Web axées RPA avec Shiny



INTRODUCTION PYTHON

- ▶ Langage Orienté-objet créé par l'Hollandais Guido van Rossum en 1991



- ▶ Comme R, totalement Open-source et multi-plateformes
- ▶ A notre avis, moins simple que R et moins adapté a des programmes d'automatisation
- ▶ Par contre, il permet, comme les outils de RPA, la navigation dans des Web-browsers.

NAVIGATION WEB AVEC PYTHON ET SELENIUM

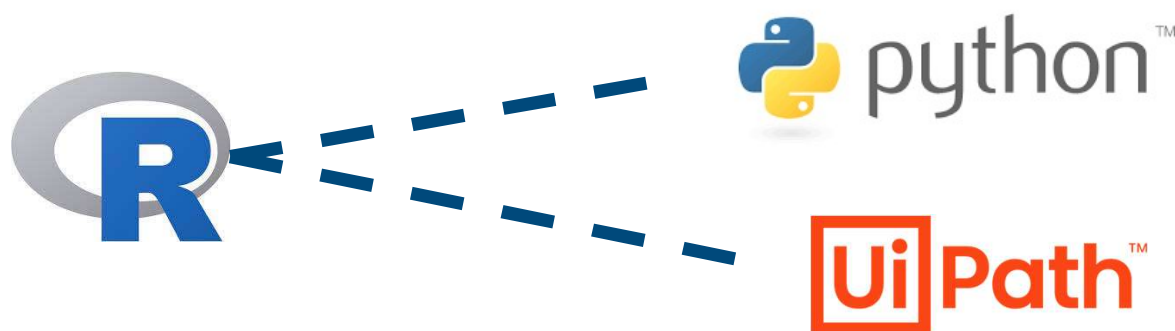
- ▶ Python permet également la navigation web grâce au package Selenium, ainsi que dans les UI's via pyautogui, win32gui etc.
- ▶ Fonctionne très bien pour le web, mais pour les UI's, c'est moins rapide et robuste que les solutions « pure » RPA, car on doit se baser sur l'analyse des images.
- ▶ La navigation est beaucoup plus rapide qu'avec UiPath bien que légèrement moins robuste.

```
inputcurr = driver.find_element_by_xpath('//*[@id="quote_currency_input"]')  
inputcurr.send_keys(othercurr['other_currencies'][i])
```



RÉSUMÉ

- ▶ Une solution pure R permet d'avoir des gains de temps sur 70% des processus communs en entreprise, mais elle ne permet pas de naviguer ni sur le web, ni dans des applications type SAP.
- ▶ R avec Python/Selenium permet de naviguer sur le web
- ▶ UiPath (et autres outil RPA) permettent de naviguer de manière robuste dans les UI's des programmes type SAP en plus du web.
- ▶ De plus en plus, les ERP's et autres progiciels disposent d'une interface web.
- ▶ R a la capacité à s'interfacer facilement avec d'autres plateformes



RÉSUMÉ

- ▶ Python permet également la navigation web grâce au package Selenium, ainsi que dans les UI's mais seulement via image
- ▶ Fonctionne très bien pour le web, mais pour les UI's, c'est moins rapide et robuste que les solutions « pure » RPA.

| | R | UiPath, BluePrism, OpenRPA | Python |
|---|-------------------------|----------------------------|-------------------------|
| Type | Programmation classique | Robotic Process Automation | Programmation classique |
| Simplicité | 8/10 | 5/10 | 6/10 |
| Navigation Web? | Instable | Oui | Oui |
| Navigation UI? | Pas Robuste (images) | Oui | Pas robuste (images) |
| Création d'UI's | Oui | Limité (Forms) | Oui |
| Nécessite l'utilisation de code? | Oui | Non (Optionnel) | Oui |

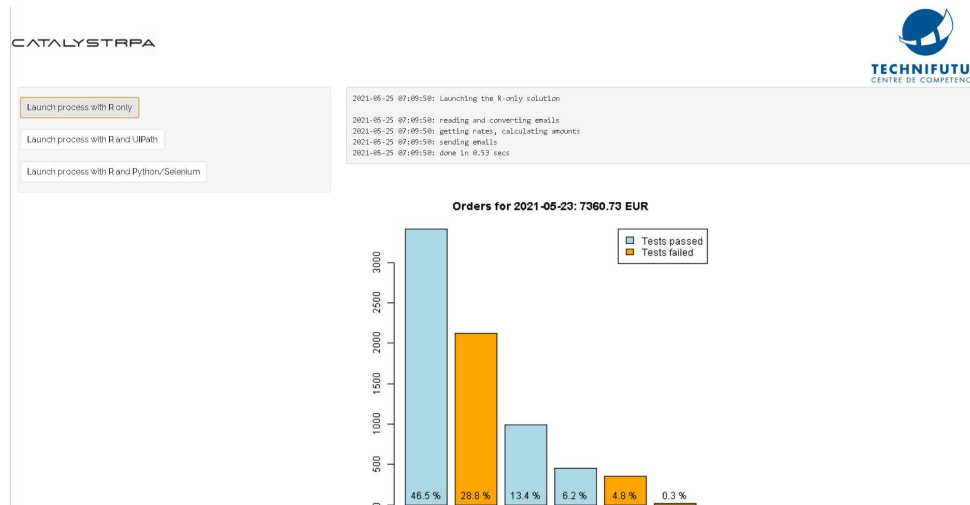
AGENDA

1. Introduction: le travail répétitif et ses conséquences
2. La formation
 - ▶ Introduction a la programmation
 - ▶ Création de scripts simples en R
 - ▶ La RPA et Création de Robots basiques avec UiPath
 - ▶ Robots Web avec Python et Selenium
 - ▶ Développer des applications Web axées RPA avec Shiny

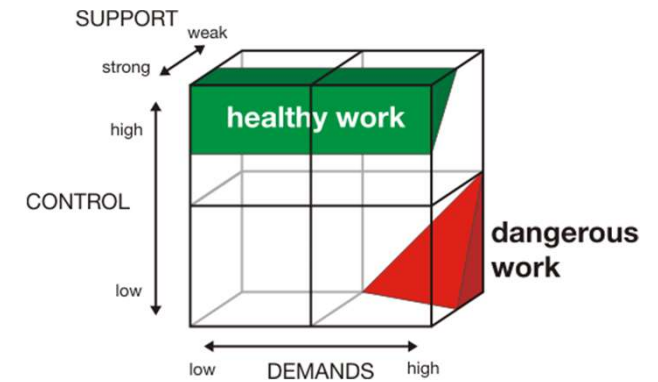
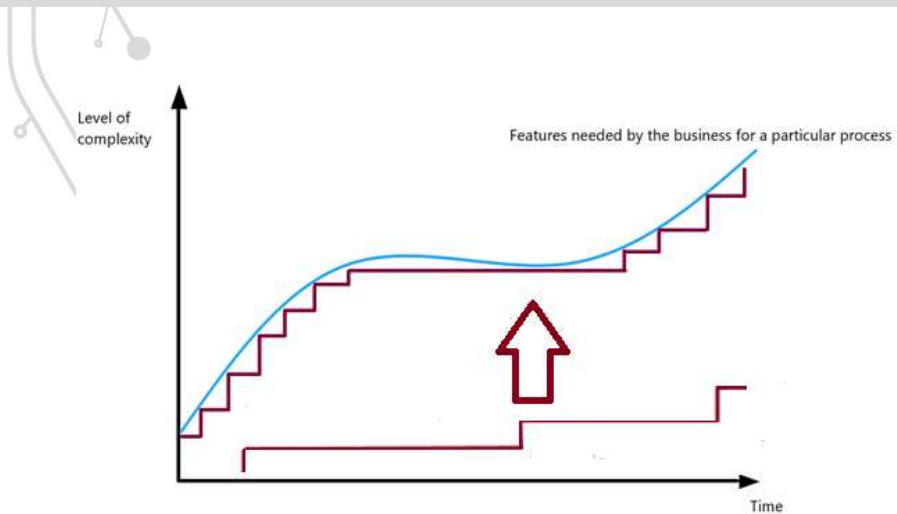


SHINY

- ▶ Grâce à R et UIPath, on arrive à automatiser quasiment tous les processus qui ont un minimum de constance.
- ▶ R seul peut très bien suffire, R et Python sont une autre possibilité.
- ▶ Une fois le programme en tant que tel réalisé, Shiny permet aux développeurs de mettre leurs scripts à disposition d'autres utilisateurs



CONCLUSION / Q&A





MERCI DE VOTRE ATTENTION



WWW.TECHNIFUTUR.BE

VOS CONTACTS

Contact@catalystrpa.com • catalystrpa.com