cegeka

Automated Software Testing

A consultant journey

Henderickx, David
05/05/2025

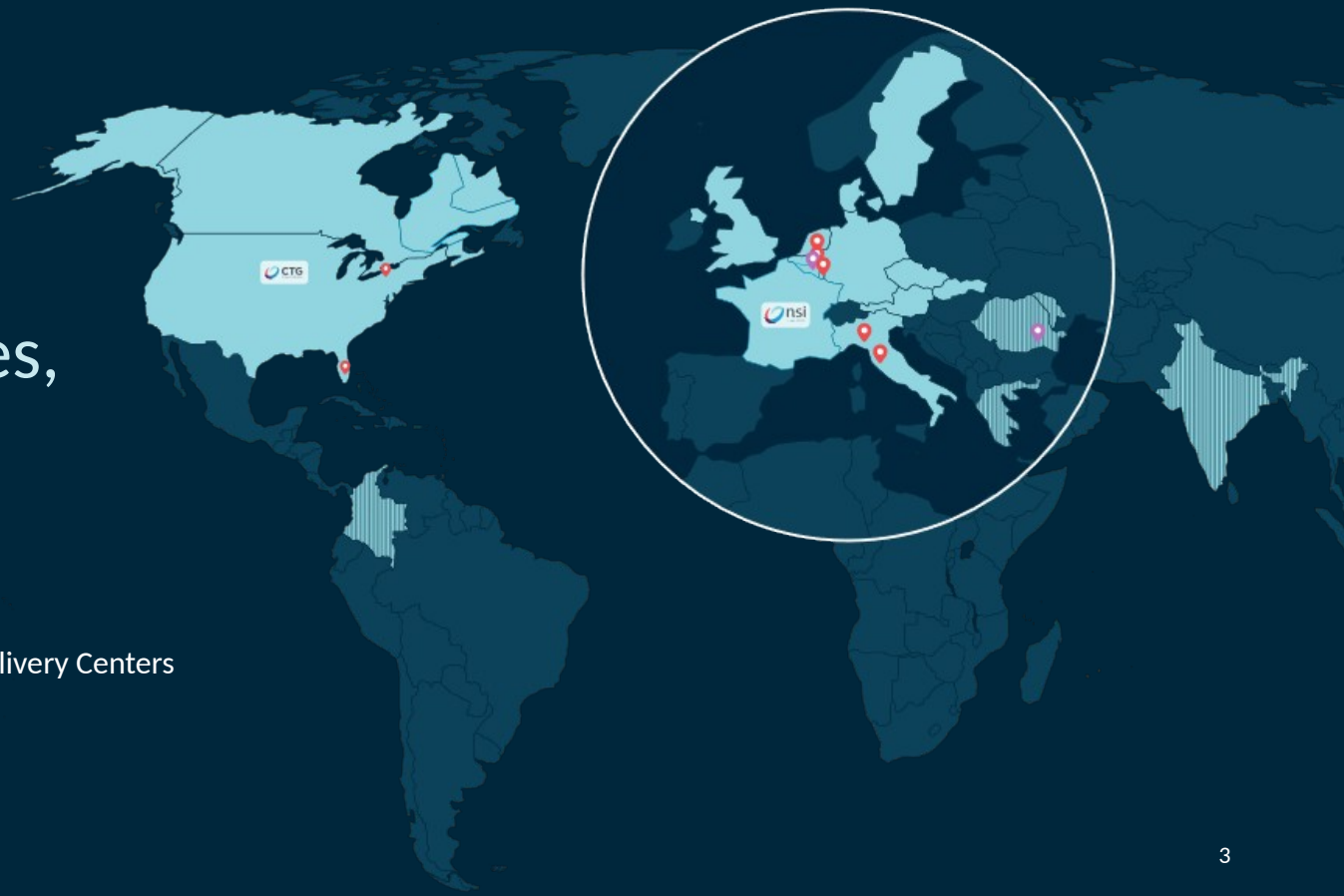
David Henderickx

David.Henderickx@cegeka.com -
https://www.linkedin.com/in/davidhenderickx/

Implementation Architect @ Cegeka

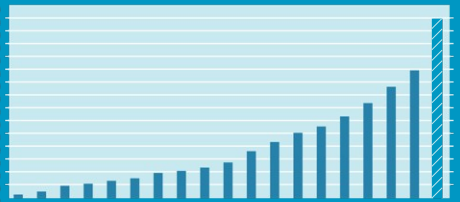
- Master in Industrial Engineering, Electronics ICT (2008)
- 17 years of software testing experience
- Passionate about test automation

Global capabilities, Local Ownership



- Commercially active
- Commercially active & Global Delivery Centers
- Data Centers
- Security Operations Centers

GROUP REVENUE



Revenue of **1.4** billion EUR

Forecast 2024)

CAPACITY FOR GROWTH

6
High Tech Datacenters

Offices in **19**
Countries **+**

SUSTAINABLE IMPACT

- Carbon neutral by **2030**
- **30% female** workforce by **2030** and leadership by **2026**
- Sustainable technology solutions for a **better world**

FAMILY OWNED

STRONG BALANCE SHEET

FINANCIAL STRENGTH

HUMAN CAPITAL

900
people in 2024

0+

DRIVEN BY CRAFTSMANSHIP

520+ Cloud

600+ Architects

1100 MS Certified

230+ Security

1700 Developers **+**

400+ Data

BUSINESS LINES

Data & AI

Hybrid Cloud

Cyber Security & Networking

Application Services

Digital Workspace

5G

Business Solutions

Products & Platforms

Quality Engineering

Who we work for

LOOKING FOR A FIRST JOB?

DO YOU WANT TO WORK FOR A GLOBAL TECH COMPANY?

OUR OFFER FOR YOUNG GRADUATES

- Continuous learning
- Tailor-made guidance
- Your workplace is wherever your computer is
- Fun = part of the job
- Opportunity to grow
- Competitve salary package
- Electric company car

EXPLORE OUR CAREERS

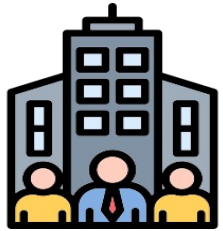
CUSTOMER

cegeka

Customer

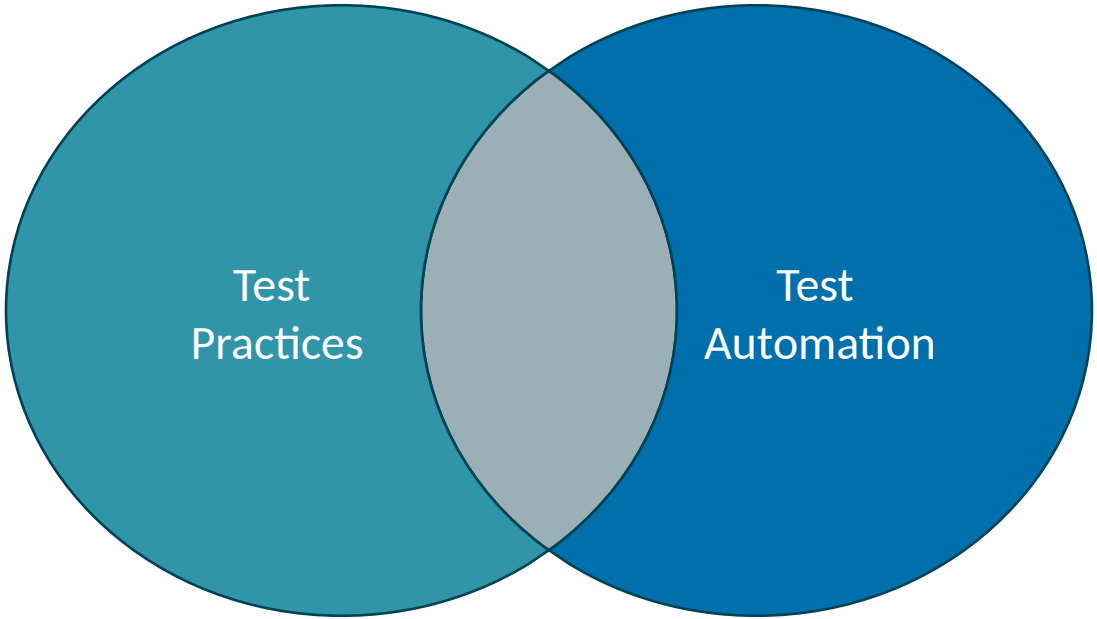
- Social service provider
  - +1500 employees
  - 29 offices in Belgium
  - Revenue +150.000.000 €

- Main services:
  - Entrepreneur helpdesk: Assistance for starting a company
  - Social Secretariat: Assistence for payroll, personal and administration
  - Social insurance funds: Social security for self-employed
  - Legal and HR expertise: For every phase in the company life cycle



We want to speed up the time to market, by reducing the testing time, while keeping the same level of trust in the release of the product.

Approach

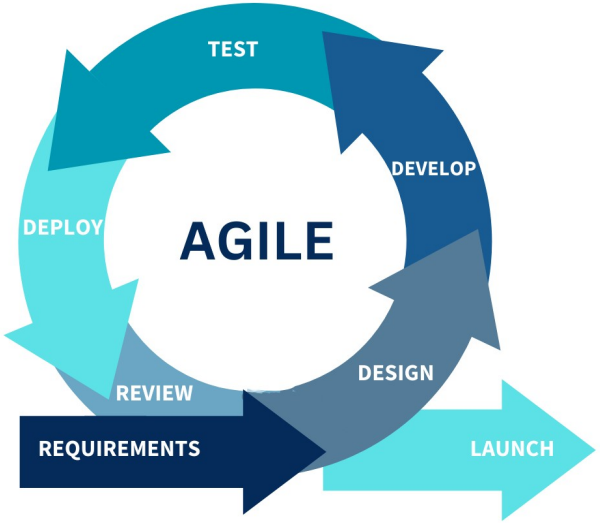


AS IS

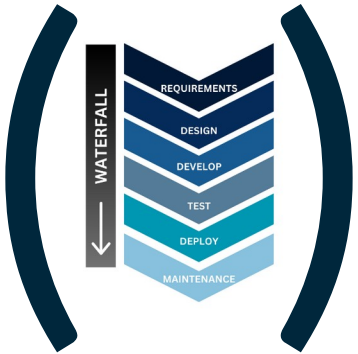
cegeka

AS IS

Project Methodology



Sprints of 2-4 Weeks



Sprints of +2 Months

AS IS

Project Management

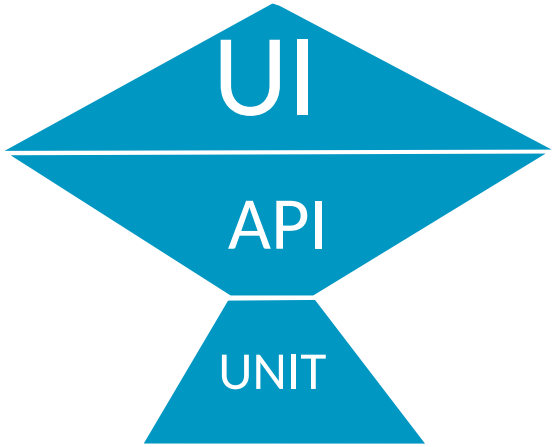
AS IS

Test Case Management

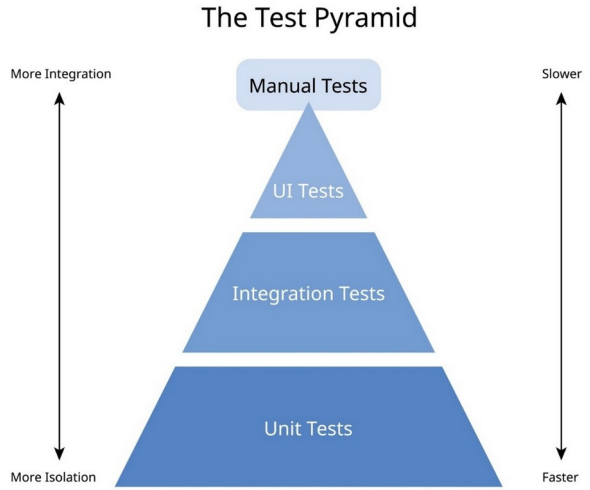
AS IS

Test Types

- **Manual Tests:**
  - UI
  - API
- **Automated Tests:**
  - Unit



V
S



START UP

cegeka

START UP

Timeline

START UP

Test Automation Tool Selection

OPEN- SOURCE **VS** COMMERCIAL

CODE **VS** USER INTERFACE

SINGLE TECHNOLOGY SUPPORT **VS** MULTIPLE TECHNOLOGY SUPPORT

ON PREMISE **VS** CLOUD

INTEGRATED **VS** TO BE INTEGRATED

HOT AND NEW **VS** STABLE STANDARD

...

START UP

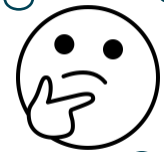
Test Automation Tool Selection

People

- Existing team members:
  - Functional Testers without technical skills
  - Developers (Limited amount and availability)

Technology

- Client Applications: Winform, Powerbuilder,Delphi
- Web Applications: Angular, React
- Mobile: Android, IOS
- Mainframe
- Citrix
- Database: SQL, Oracle
- Services: SOAP, REST
- File Handling: CSV, XML , TXT, PDF, Excel



Infrastructure

- Azure Devops: Git Repo, Pipeline, Project and Test Management
- Windows Server

Custom Request

START UP

Test Automation Tool Proposition

	CRITERIA		SELENIUM		RANOREX		UFT One		TOSCA	
			SUPP	COMMENT	SUPP	COMMENT	SUPP	COMMENT	SUPP	COMMENT
41	Javascript Programming Language		FC	Javascript, Java, C#, Ruby, Python	NC	Visual Basic, C#	NC	VBScript	NC	VBScript, C#
42	**Test Objects**									
43	Object Repository		FC	Have a shared object repository by using the 'page object' design pattern.	FC	Shared object repository.	FC	Shared object repository.	FC	Shared object repository.
44	Object Recognition		PC	Plugins available (POM Builder) or by looking at the HTML.	FC	Object Spy	FC	Object Spy	FC	Object spy and the possibility to easely add multiple objects at once.
45	Object Identification		FC	ID, Name, Link text, Tagname, Xpath, Class Name, CSS Selector	FC	Ranorex Xpath + visualize to select the properties of the object.	FC	Visualize to select the properties of the object.	FC	Visualize to select the properties of the object and hidden Xpath feature.
46	Image Based object identification		PC	Not build in, but can be done by the Sikuli library.	FC		FC		FC	
47	**Validation**									
48	Object Validation (properties)		FC		FC		FC		FC	
49	Data Validation (text)		FC		FC	Dynamic data needs to be created through code.	FC	Dynamic data needs to be created through code.	FC	Build in module to validate a combination of a variable and math expression, number, multiple combinations of variables, ...
50	Image Validation		PC	Not build in, but can be done by adding extra libraries to the project (example: applitools).	FC		FC		FC	
51	Data et False Positives		NC		NC		NC		NC	
52	External Validation (DB)	High	PC	Not build in, but can be done by adding extra libraries to the project.	PC	Not build in, but can be done by adding extra libraries to project and working with code based modules.	FC	Build in feature.	FC	Build in modules.
53	Report	High	PC	The unit test framework a provides a basic report. Fancy customizable reports are created by using 3th part libraries.	FC	Customizable Report.	FC	Customizable Report.	FC	Customizable report available in Tricentis. Tricentis Dll's you create external reports.
54	Centralized Dashboard		NC	When integrating the automated tests with Azure Devops, Azure Devops could be used to act like a "dashboard".	NC	When integrating the automated tests with Azure Devops, Azure Devops could be used to act like a "dashboard".	NC	When linked with Micro Focus ALM or Octane, you can view the detail in a dashboard. When integrating the automated tests with Azure Devops, Azure Devops could be used to act like a "dashboard".	NC	When integrating the automated tests with Azure Devops, Azure Devops could be used to act like a "dashboard".
55	**Maintainability**									
56	Field changes from position	High	FC	All dependent on the level of quality from the AUT. All test objects should be uniquely identifiable.	FC	All dependent on the level of quality from the AUT. All test objects should be uniquely identifiable.	FC	All dependent on the level of quality from the AUT. All test objects should be uniquely identifiable.	FC	All dependent on the level of quality from the AUT. All test objects should be uniquely identifiable.
57	Extra field added	High	FC	Test should be compared out of small reusable module.	FC	Test should be compared out of small reusable module.	FC	Test should be compared out of small reusable module.	FC	Test should be compared out of small reusable module.
	Functionality remains, technology changes		FC	As long as the identifiable properties of the test objects and the functionality does not change, there is no big refactor needed.	FC	As long as the identifiable properties of the test objects and the functionality does not change, there is no big refactor needed.	FC	As long as the identifiable properties of the test objects and the functionality does not change, there is no big refactor needed.	FC	As long as the identifiable properties of the test objects and the functionality does not change, there is no big refactor needed.

Fully Compliant

Partially Compliant

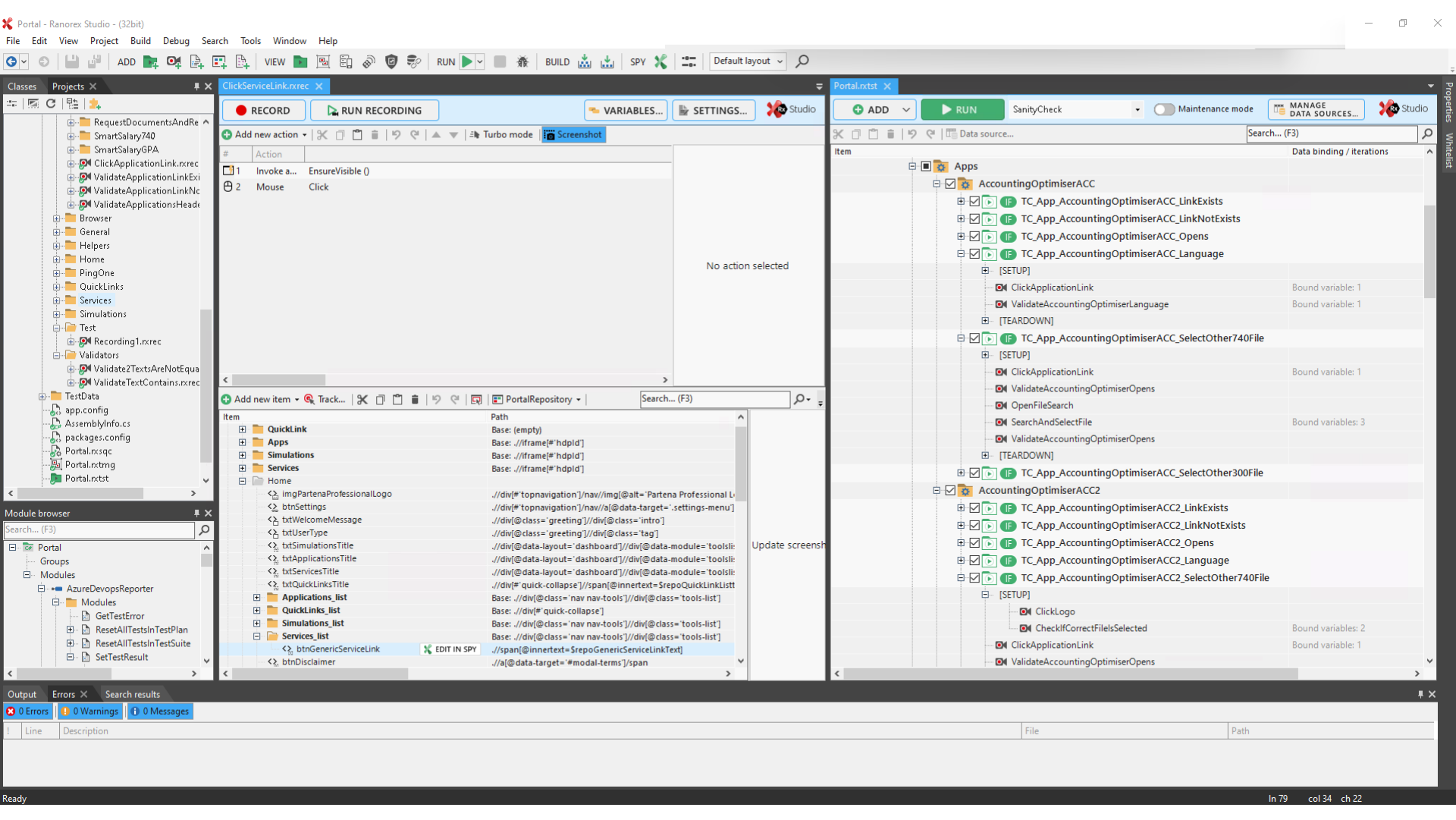
Not Compliant

START UP

Selected Tool: Ranorex

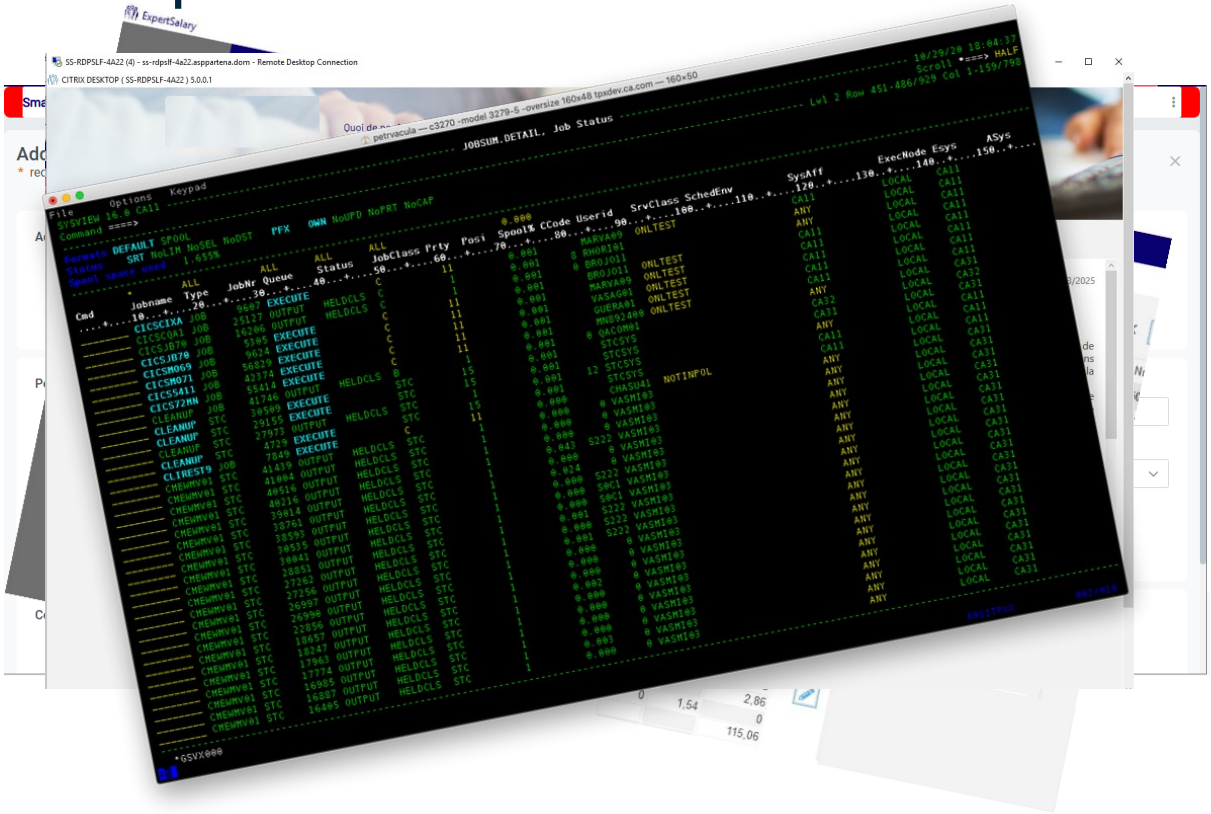
- Commercial test automation tool
- Easy-to-use automated testing software, as a beginner or expert test automator
- Create automated testing projects on any desktop, web or mobile application.
- Test “any” technology
- Use of an easy to use graphical user interface
- Code and code-less test automation
- Recording functionality

- https://www.ranorex.com/


Portal - Ranorex Studio - (32bit)
File Edit View Project Build Debug Search Tools Window Help
ADD VIEW RUN BUILD SPY Default layout
Classes Projects
RequestDocumentsAndRe
SmartSalary740
SmartSalaryGPA
ClickApplicationLink.rxrec
ValidateApplicationLinkExi
ValidateApplicationLinkNc
ValidateApplicationsHeade
Browser
General
Helpers
Home
PingOne
QuickLinks
Services
Simulations
Test
Recording1.rxrec
Validators
Validate2TextsAreNotEqua
ValidateTextContains.rxrec
TestData
app.config
AssemblyInfo.cs
packages.config
Portal.rxsqc
Portal.rxtmg
Portal.rxtst
Module browser
Search... (F3)
Portal
Groups
Modules
AzureDevopsReporter
Modules
GetTestError
ResetAllTestsInTestPlan
ResetAllTestsInTestSuite
SetTestResult
ClickServiceLink.rxrec
RECORD
RUN RECORDING
VARIABLES...
SETTINGS...
Studio
Add new action
Turbo mode
Screenshot
Action
1 Invoke a... EnsureVisible ()
2 Mouse Click
No action selected
Add new item
Track...
PortalRepository
Search... (F3)
Item Path
QuickLink Base: (empty)
Apps Base: .//iframe[#'hdpId']
Simulations Base: .//iframe[#'hdpId']
Services Base: .//iframe[#'hdpId']
Home
imgPartenaProfessionalLogo .//div[#'topnavigation']/nav//img[@alt='Partena Professional L
btnSettings .//div[#'topnavigation']/nav//a[@data-target='.settings-menu']
txtWelcomeMessage .//div[@class='greeting']//div[@class='intro']
txtUserType .//div[@class='greeting']//div[@class='tag']
txtSimulationsTitle .//div[@data-layout='dashboard']//div[@data-module='toolsli
txtApplicationsTitle .//div[@data-layout='dashboard']//div[@data-module='toolsli
txtServicesTitle .//div[@data-layout='dashboard']//div[@data-module='toolsli
txtQuickLinksTitle .//div[#'quick-collapse']//span[@innertext=$repoQuickLinkListT
Applications_list Base: .//div[@class='nav nav-tools']//div[@class='tools-list']
QuickLinks_list Base: .//div[#'quick-collapse']
Simulations_list Base: .//div[@class='nav nav-tools']//div[@class='tools-list']
Services_list Base: .//div[@class='nav nav-tools']//div[@class='tools-list']
btnGenericServiceLink EDIT IN SPY .//span[@innertext=$repoGenericServiceLinkText]
btnDisclaimer .//a[@data-target='#modal-terms']/span
Update screensh
Portal.rxtst
ADD RUN SanityCheck Maintenance mode MANAGE DATA SOURCES... Studio
Data source...
Search... (F3)
Item Data binding / iterations
Apps
AccountingOptimiserACC
TC_App_AccountingOptimiserACC_LinkExists
TC_App_AccountingOptimiserACC_LinkNotExists
TC_App_AccountingOptimiserACC_Opens
TC_App_AccountingOptimiserACC_Language
[SETUP]
ClickApplicationLink Bound variable: 1
ValidateAccountingOptimiserLanguage Bound variable: 1
[TEARDOWN]
TC_App_AccountingOptimiserACC_SelectOther740File
[SETUP]
ClickApplicationLink Bound variable: 1
ValidateAccountingOptimiserOpens
OpenFileSearch
SearchAndSelectFile Bound variables: 3
ValidateAccountingOptimiserOpens
[TEARDOWN]
TC_App_AccountingOptimiserACC_SelectOther300File
AccountingOptimiserACC2
TC_App_AccountingOptimiserACC2_LinkExists
TC_App_AccountingOptimiserACC2_LinkNotExists
TC_App_AccountingOptimiserACC2_Opens
TC_App_AccountingOptimiserACC2_Language
TC_App_AccountingOptimiserACC2_SelectOther740File
[SETUP]
ClickLogo
CheckIfCorrectFileIsSelected Bound variables: 2
ClickApplicationLink Bound variable: 1
ValidateAccountingOptimiserOpens
Output Errors Search results
0 Errors 0 Warnings 0 Messages
Line Description File Path
Properties Whitelist
Ready
ln 79 col 34 ch 22

START UP

Proof Of Concepts



START UP

Verify Test Tool Selection

- Proof of concept (POC) done on a selection of priority 1 applications:
  - Web Application
  - Legacy Client Application
  - Citrix
  - Mainframe

- The goal of a POC is to verify:
  - If we can automate the application with the test automation tool
  - How difficult is it to automate the application:
    - *Object recognition*
    - *Special elements*
    - *Dynamic pages*

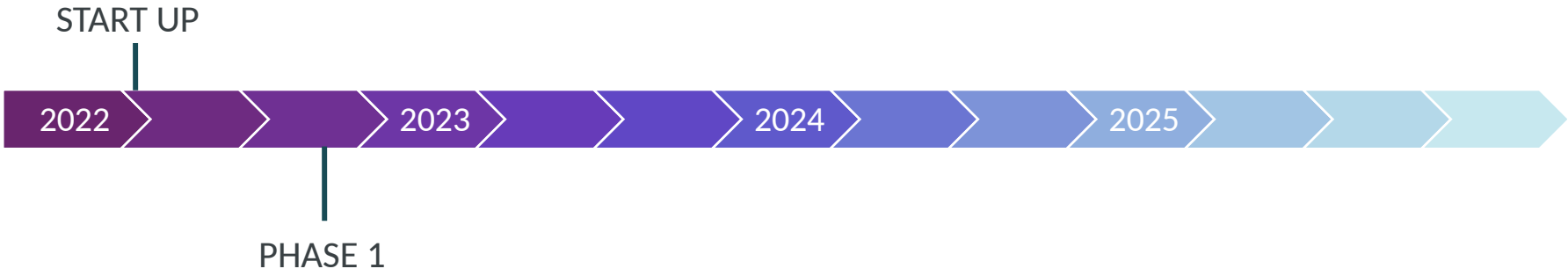
- Results:
  - Web Application: Automatable but elements need more custom attributes to improve object identifications
  - Legacy Client Application: Automatable but slow object identification
  - Citrix: “Not” automatable. Screen seen as an image. Image based testing is possible, but not advisable
  - Mainframe: Automatable with OCR (Optical Character Recognition) and position on the terminal

PHASE 1

cegeka

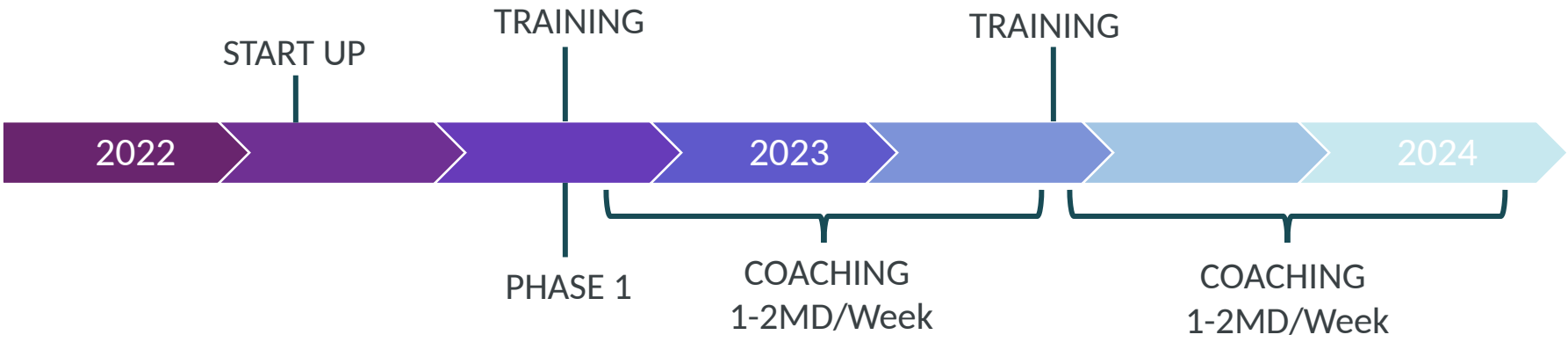
PHASE 1

Timeline



PHASE 1

Approach



Test Automation Scope: Prioritized regression test cases

PHASE 1

Outcome

	Team 1 (ES)	Team 2 (CTS)	Team 3 (CP)	Team 4 (EDM)				
App type	Client	Web	Web	Web				
PHASE 1 (09/2023 - 12/2024)								
Trained People	3	4	3	2				
Automated Regression cases	1%	50%	5%	0%				

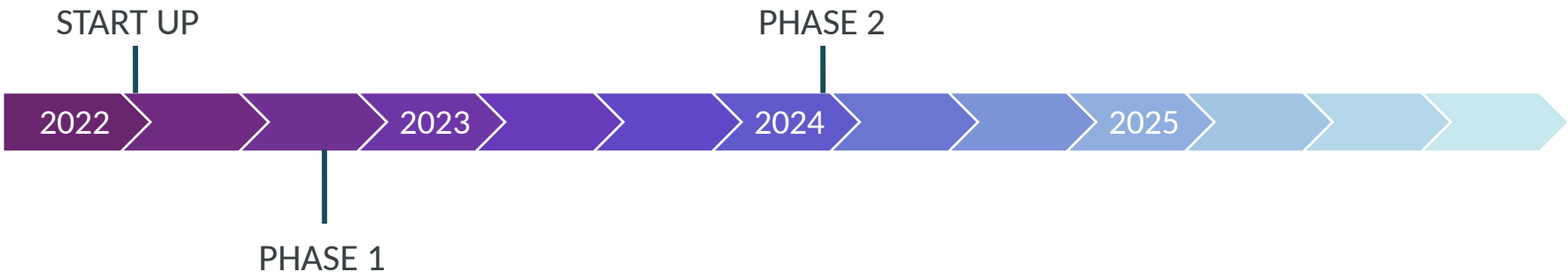
- Teams did not have the needed capacity
- “When I have some time” is not efficient
- Test automation remains a technical skill

PHASE 2

cegeka

PHASE 2

Timeline



PHASE 2

Accept and Adapt

- Change Test Automation Recourses:
  - Transversal team: Team of test automation experts (3) that implement and maintain the project team’s automation backlog.
  - Project team: Run the automated tests. Create and maintain automated tests if capacity and knowledge exist.

- Change Test Automation Tool suite:
  - Reduce license cost: Open Source
  - Speed up test execution: Parallel execution on scalable Docker infrastructure
  - Increase maintainability: Uniform implementation with best practices and peer review. Advantages of code-based scripts vs UI actions.
  - Improve reporting: Azure Devops integration and Allure Report Server (https://allurereport.org/)

PHASE 2

Workflow

	INITIATE	PREPARE	EXECUTE	CLOSE
PROJECT TEAM ACTIONS	Initiate new request via the service portal	Approve the estimation Arrange practicalities	Ensure the availability of a SPOC in case of questions	Review and validate the received deliverables
TRANSVERSAL TEAM ACTIONS	Intake of the service request Estimation (effort, planning)	Verify if everything is in place to start the service execution.	Create the automation script(s) and perform a verification run for each	Hand over deliverables Document knowledge to secure the continuity
TRANSVERSAL TEAM ACTIONS	Logged service request Estimation	Approved service request (scope & effort)	Automated test cases Verification run report	Retrospective

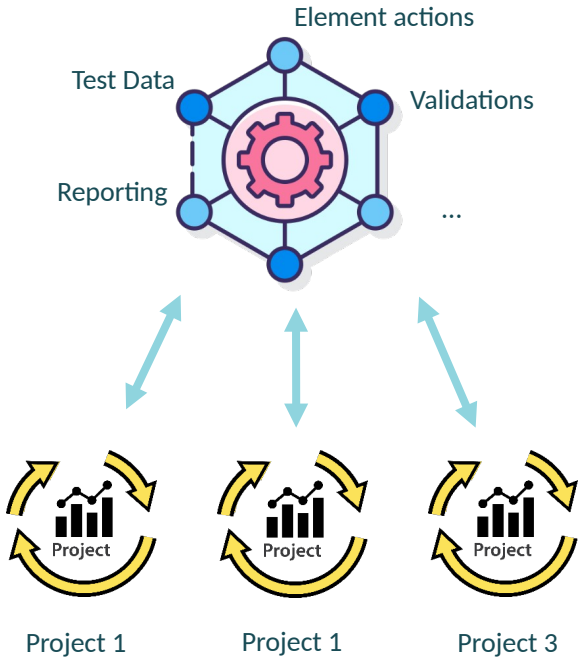
PHASE 2

C# Test Automation Framework

- The framework:
  - Common framework for all teams
  - Provides all functionality needed to be able to automate
  - Managed and distributed by the transversal team
  - Available as a NuGet package

- Team projects:
  - Contains the tests, test data, application elements, ... of that specific team
  - Uses the functionality of the framework for running the tests, interacting with the application, reporting, ...

- Technologies:
  - Web: Selenium
  - Client UI: No implementation
  - Mainframe: Keyboard actions
  - Mobile: Appium
  - API: RestAssured.Net



PHASE 2

Current Status

	Team 1 (ES)	Team 2 (CTS)	Team 3 (CP)	Team 4 (EDM)	Team 5 (ESS)	Team 6 (PPP1)	Team 7 (CR)	Team 8 (EP)
App type	Client	Web	Web	Web	Web	Web	Web	Web
PHASE 1 (09/2023 - 12/2024)								
Trained People	3	4	3	2				
Automated Regression cases	1%	50%	5%	0%				
PHASE 2 (01/2024 – Now)								
Implement workflow, Framework: Proposal + Implementation + Infrastructure								
Automated Regressions cases	90% (API)	80%	90%	0%	75%	10%	startup phase	startup phase
Performance Test	X	X			X			


cegeka