클라우독 테스트 자동화(with RPA)

품질보증팀 배재우

목차

A table of Contents

Part 1,

What is RPA

What is RPA

- **인간을 대신하여 수행할 수 있도록 단순 반복적인 업무를 알고리즘화하고 Software적으로 자동화하는 기술**
  - ✓물리적 로봇이 아닌 Software Program으로 사람이 하는 Rule Based 업무를 기존의 IT 환경에서 동일하게 할 수 있도록 구현한 것
  - ✓S/W Robot이 하는 일은 시스템 로그인, 문서 생성/쓰기, 화면조회, 특정 셀의 데이터를 읽고 쓰고 계산하기, 이메일 보내기 등 매우 단순하나, 이러한 단순 업무가 사무직 업무의 30~50% 이상 차지

What is RPA 장점

- 저렴하고 신속한 구축비용과 빠른 ROI
  - 기간계 시스템을 건드리는 기존 IT Project와 달리 기존 IT 인프라상에서 S/W적으로 구현하여 수주~수개월의 짧은 기간이 소요되며 비교적 명확한 ROI 산출 가능
  - ERP 구축 대비 상대적 ROI는 10배 이상 높고, 구현의 난이도는 50분의 1 수준 (Ernest&Young, ’17)
- 업무 생산성 제고 및 근무 만족도 제고
  - 24h x 365d(無휴일, No Extra Cost)로 생산성이 높고, Biz Volume 증가감소에 따른 인력운영의 유연성 확보 용이
  - 단순 반복업무 비중 감소에 따른 직원 만족도 제고 및 고부가가치 업무 전환
- 업무 품질 및 투명성 제고, Compliance 대응 용이
  - 사람에 의한 입력 실수 등 Human Error 방지
  - 업무 투명성 제고 및 프로세스 모니터링 강화
  - Compliance 대응, 업무 가이드라인 변경 등에 따른 변화관리 용이

Part 2,

RPA 액티비티

액티비티_UiPath.Core.Activities

- UiPath.Core.Activities.Click
- UiPath.Core.Activities.TypeInto
- UiPath.Core.Activities.SendHotkey
- UiPath.Core.Activities.ReadTextFile
- UiPath.Core.Activities.WriteTextFile
- UiPath.Core.Activities.AppendLine
- UiPath.Core.Activities.GetPassword
- UiPath.Core.Activities.SetValue
- UiPath.Core.Activities.GetEnvironmentFolder
- UiPath.Core.Activities.GetFromClipboard /.SetToClipboard

액티비티_UiPath.Core.Activities

- UiPath.Core.Activities.StartProcess
- UiPath.Core.Activities.CloseWindow
- UiPath.Core.Activities.CreateDirectory
- UiPath.Core.Activities.OpenBrowser
- UiPath.Core.Activities.CloseTab
- UiPath.Core.Activities.MaximizeWindow
- UiPath.Core.Activities.WaitUiElementAppear
- UiPath.Core.Activities.PathExists
- UiPath.Core.Activities.InvokeCode(C#)
- UiPath.Core.Activities.Delete

액티비티_System.Activities.Statements

- System.Activities.Statements.If
- System.Activities.Statements.Switch
- System.Activities.Statements.TryCatch
- System.Activities.Statements.Assign

액티비티_Excel, Word, 기타

- UiPath.Excel.Activities.ExcelApplicationScope
- UiPath.Excel.Activities.ExcelReadCell
- UiPath.Excel.Activities.ExcelWriteCell
- UiPath.Word.Activities.WordApplicationScope
- UiPath.Word.Activities.WordReadText
- UiPath.Word.Activities.WordAppendText
- UiPathTeam.File.Activities
- BalaReva.DirFileSize.Activities

Part 3,

테스트 설계

자동화 테스트 목적

- 테스트 시나리오 중 RPA로 검증 가능한 테스트 케이스를 자동화 하여 리소스를 절감하고 병렬 처리로 테스트 속도 향상
- 자동화로 Human error 방지하며 일관성 있는 테스트 결과 보장
- 자동화 스크립트 관리/유지보수로 인한 비효율을 방지하기 위해 ‘기본 기능 검증’ 위주로 최소화된 스크립트(Small TC) 설계
- RPA로 늘어난 QA 엔지니어의 리소스는 테스트 시나리오 고도화 등의 다른 가치 있는 작업에 사용

자동화 테스트 계획

- 자동화 도구 : Uipath(RPA), Selenium, C#, Excel
- 테스트 범위 : BVT, 통합테스트, 회귀테스트
- 테스트 예외 : RPA로 검증이 불가능한 케이스 ex) disklock+ …
- 테스트 설계
  ① 테스트 동작 수행
  ② 수행된 동작의 결과가 정상인지 아닌지 검증
  ③ 검증된 결과는 엑셀 파일에 Pass/Fail 형태로 입력
- 테스트 목표 : Tedious 작업 회피 및 High Risk 결함 조기 검출
- 테스트 통과 기준 : 자동화한 전체 테스트 케이스 Pass

테스트 설계_공통

- Uipath 설계 파일 경로 : C:\Users\{User Name}\Documents\Uipath
- 테스트 케이스 경로 : C:\RPA_TEST\RPA_Client_TC.xlsx
- 테스트 순서 : A01(B01~B12) >윈도우 재시작 > A02 > A03
- 네트워크락 패키징은 A04(B13~14)까지 진행
- 테스트 전 최초 1회 설정
  - Uipath 실행>홈>도구>Chrome Extension 설치
  - Uipath\ClouDoc\batch\network_rule_add.bat 실행
  - 휴지통 속성 > " 삭제 확인 대화 상자 표시 " 설정
- ~~“B.12”, “A.02” 설계 내 Get Password 액티비티 비밀번호 입력~~

테스트 설계_B.01_레지스트리

1. Input Dialog로 dwMajorVersion에 대해 사용자 입력 받기
2. Input Dialog로 dwMiorVersion에 대해 사용자 입력 받기
3. Invoke code(C#)로 dwMajorVersion 레지스트리 검증
4. Invoke code(C#)로 dwMinorVersion 레지스트리 검증
5. Invoke code(C#)로 InstallPath 레지스트리 검증
6. Invoke code(C#)로 ProductName 레지스트리 검증
7. Invoke code(C#)로 npStartup 레지스트리 검증

테스트 설계_B.02_office

1. Invoke code(C#)으로 개인문서함 마운트 체크
2. 개인문서함 내 엑셀, 워드 파일 쓰기
3. 저장된 파일 내용 검사
4. 개인문서함 내 엑셀, 워드 파일 편집 저장
5. 저장된 파일 내용 검사
6. 1~5 스텝을 부서문서함에서 진행

테스트 설계_B.03_move

1. 개인문서함 루트 경로 -> 1depth 폴더 경로 파일 이동
2. 개인문서함 루트 경로 -> 1depth 폴더 경로 파일 덮어쓰기
3. 개인문서함 -> 부서문서함 파일 이동
4. 개인문서함 -> 부서문서함 파일 덮어쓰기
5. 1~4 스텝에 대해 각각 Path Exists 액티비티로 결과 검증
6. 1depth에 저장된 파일을 delete 키로 삭제 및 검증
7. 부서문서함에 저장된 파일을 서버 삭제 및 검증
8. 개인문서함, 부서문서함 휴지통 비우기 검증

테스트 설계_B.04_copy

1. 개인문서함 루트 경로 -> 1depth 폴더 경로 파일 복사
2. 개인문서함 루트 경로 -> 1depth 폴더 경로 파일 덮어쓰기
3. 개인문서함 -> 부서문서함 파일 복사
4. 개인문서함 -> 부서문서함 파일 덮어쓰기
5. 1~4 스텝에 대해 각각 Path Exists, 파일 바이너리 비교(C#)
6. 개인문서함에 저장된 파일을 Shift+서버 삭제 후 검증
7. Tear Down(파일 복사 테스트 중 생성된 파일 삭제)

테스트 설계_B.05_폴더

1. 개인문서함 내 폴더 생성
2. 폴더 내 office 파일 쓰기, 편집 저장 검증(“B.01”과 동일)
3. 폴더 이동 복사 검증(“B.02~3”과 동일)
4. Tear Down(폴더 테스트 중 생성된 파일 삭제)

테스트 설계_B.06_공유폴더

1. Invoke code(C#)으로 공유문서함 마운트 체크
2. 모든 권한 허용된 공유 폴더 권한 검사
3. 읽기 권한만 허용된 공유 폴더 권한 검사
   - Try Catch : AggregateException 체크
   - Try Catch : UnauthorizedAccessException 체크
4. 읽기, 쓰기 권한만 허용된 공유 폴더 권한 검사
   - Try Catch : UnauthorizedAccessException 체크
5. 목록보기 권한만 허용된 공유 폴더 권한 검사
   - Try Catch : AggregateException 체크
   - Try Catch : UnauthorizedAccessException 체크

테스트 설계_B.07_웹 링크

1. 웹링크 복사 실행(키보드 입력, 마우스 클릭 자동화)
2. GetFromClipboard 액티비티로 복사된 URL 주소 검증
3. 크롬 브라우저로 웹 링크 파일 다운로드
4. 다운로드 여부, 다운로드된 파일 바이너리 검사

테스트 설계_B.08_확장 업로드, 다운로드

1. “B.07”에서 다운로드한 파일을 개인문서함으로 확장 업로드
2. Path Exists, 원본, 업로드 파일 간 바이너리 비교(C#)
3. 업로드된 파일을 “/Desktop” 경로로 확장 다운로드
4. Path Exists, 원본, 다운로드 파일 간 바이너리 비교(C#)

테스트 설계_B.09_온라인 보안디스크

1. Invoke code(C#)으로 온라인 보안 디스크 마운트 체크
2. 보안디스크 내 엑셀, 워드 파일 쓰기, 편집 저장 검증
3. 저장한 파일을 delete 키로 파일 삭제 [예, 아니오] 검증
4. 파일 삭제 후 휴지통 내 파일 복원 진행
5. Path Exists 액티비티로 파일 복원 여부 확인

테스트 설계_B.10_오프라인 보안디스크

1. Start Process 액티비티로 네트워크 차단 배치 실행
2. Invoke code(C#)으로 오프라인 보안 디스크 마운트 검증
3. 보안디스크 내 엑셀, 워드 파일 쓰기, 편집 저장 검증
4. Start Process 액티비티로 네트워크 차단 해제 배치 실행
5. 네트워크 연결 후 자동 파일 업로드 진행
6. Path Exists 액티비티로 파일 업로드 여부 확인
7. Invoke code(C#)으로 오프라인 보안 디스크 마운트 해제 검증

테스트 설계_B.11_문서반출 일반 후결재

1. 현재 날짜, 시간을 Today 변수에 저장
2. 일반 문서반출 후결재 신청
3. DOC_EXPORT 내 최근 수정된 폴더 이름 저장(C#)
4. “Today < 반출 폴더 날짜”로 반출 여부 체크
5. DOC_EXPORT, 원본 파일 간 바이너리 비교(C#, FC)
6. DOC_EXPORT 내 폴더 우클릭 후 바탕화면으로 반출
7. DOC_EXPORT, 바탕화면 파일 간 바이너리 비교(C#, FC)

테스트 설계_B.12_문서반출 보안 선결재

1. 현재 날짜, 시간을 Today 변수에 저장
2. 보안 문서반출 선결재 신청
3. 크롬 브라우저로 문서반출 승인
4. DOC_EXPORT 내 최근 수정된 폴더 이름 저장(C#)
5. “Today < 반출 폴더 날짜”로 반출 여부 체크
6. 반출 목록에 로컬 디스크가 표시되지 않음 : Try Catch로 “SelectNotFoundException” 발생 검증
7. Invoke code(C#)으로 반출디스크 마운트 체크
8. 반출디스크로 문서 반출 후 바이너리 비교(C#, FC)

테스트 설계_B.13_네트워크락 내부망

1. 허용된 IP 대역 http request 요청 후 OK 응답 체크
   - http://192.168.1.100
   - https://www.google.com
2. 차단된 IP 대역 http request 요청 후 WebException 체크
   - https://www.naver.com
   - https://www.daum.net
3. 외부망 전환 시 프로세스 종료 창 on element appear 체크
4. 외부망 전환 후 get clipboard 데이터 체크

테스트 설계_B.14_네트워크락 외부망

1. 클라우독 문서함 언마운트, 반출 디스크 마운트 검증
2. 반출 디스크 내 office 파일 쓰기 검증
3. 차단된 IP 대역 http request 요청 후 WebException 체크
   - http://192.168.1.100
   - https://www.google.com
4. 허용된 IP 대역 http request 요청 후 OK 응답 체크
   - https://www.naver.com
   - https://www.daum.net
5. 내부망 전환(클라우독 문서함 마운트 검증)

테스트 설계_A.02_반출디스크_권한

1. “B.12” 수행 후 윈도우 재시작
2. 보안디스크 로그인
3. Invoke code(C#)으로 반출디스크 마운트 검증
4. 반출 디스크 내 파일 읽기O, 쓰기O, 편집O, 삭제O 권한 검증
5. 클라우독 로그인
6. 반출 디스크 내 파일 읽기O, 쓰기X, 편집X, 삭제O 권한 검증

테스트 설계_A.03_WhiteList 정책

1. WhiteList 정책 수동 설정
2. 개인 문서함 내 Read Only 카테고리 파일 저장
3. 개인 문서함 내 Read Only 카테고리 파일 편집 저장
4. 로컬 디스크 내 Read Only 카테고리 파일 저장 거부 확인
5. 로컬 디스크 내 미분류 카테고리 파일 저장 확인
6. 개인 문서함 내 미분류 카테고리 파일 저장 거부 확인

테스트 설계_참고자료

- C# 네임스페이스 참조 : https://docs.uipath.com/studio/docs/importing-new-namespaces
- C# 레지스트리 : https://docs.microsoft.com/ko-kr/dotnet/api/microsoft.win32.registryvaluekind?view=dotnet-plat-ext-3.1
- C# 파일 비교 : https://docs.microsoft.com/ko-kr/troubleshoot/dotnet/csharp/create-file-compare
- C# CMD 입출력 : https://infodbbase.tistory.com/90
- C# httpWebRequest : https://docs.microsoft.com/ko-kr/dotnet/api/system.net.httpstatuscode?view=netcore-3.1

감사합니다.