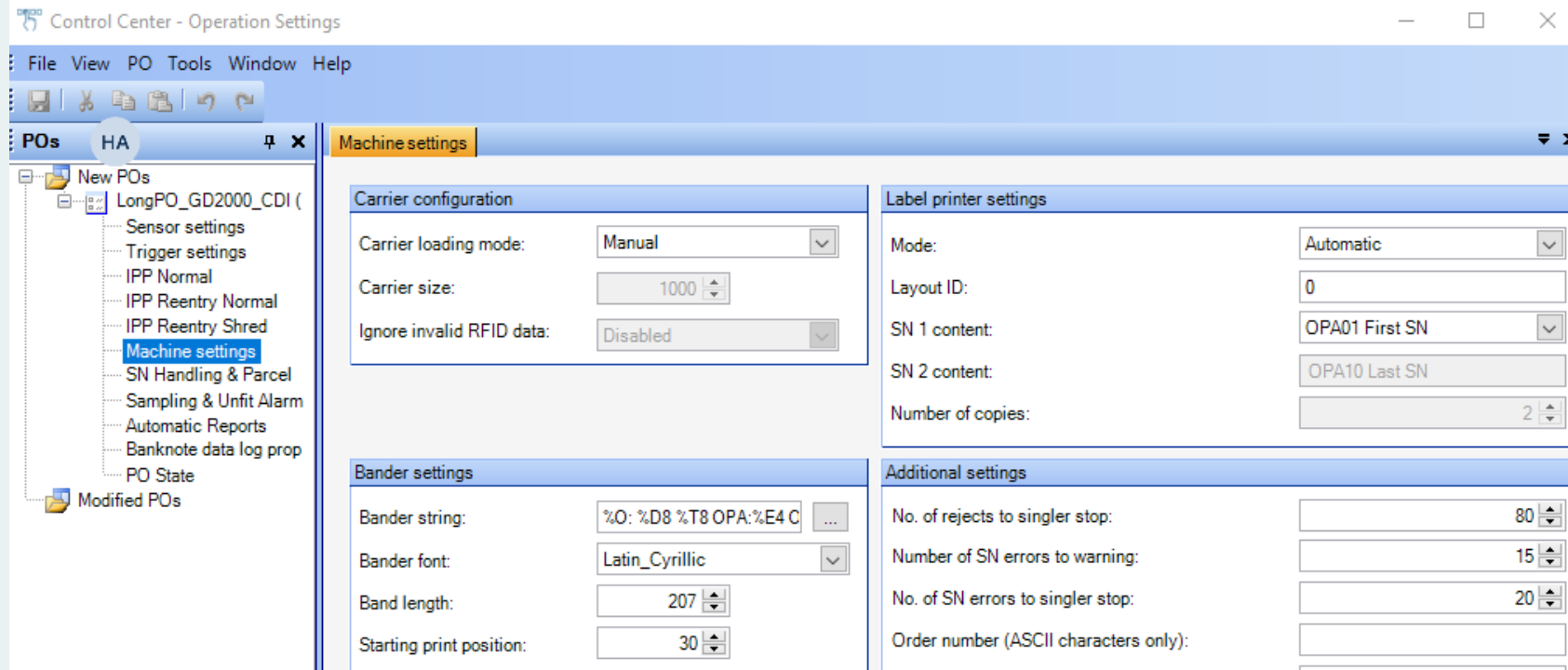




Human Machine Interface (HMI) Test Automatisierung mit dem CTCS SW Application Testing Framework (ATF)

K. Hechenblaikner
S. Schulz
V. Navratil

Einführung in HMI Test Automatisierung

Motivation

- Heute keine automatisierte Interaktionen bzw. Tests für G+D legacy tools wie den Control Center und ähnliche wie EcoConfigurator etc.
- Wenn überhaupt manuelle Tests
- Trend weg von kommerziellen Testautomatisierungslösungen

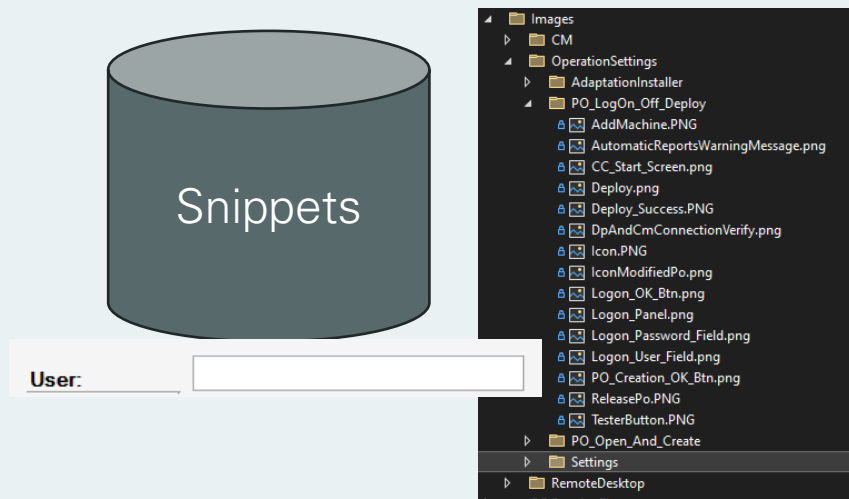
Idee

- Nutzung von Open Source basierter Bilderkennung (OpenCvSharp)
 - Einsatz von Snippets für Image Matching
- Automatisierung von User Interface Interaktionen
 - Klicks, Texteingaben, Dropdown-Auswahl, Scrollen, Checkboxes
 - Steuerung per Maus- und Tastatur-Simulation (inkl. hotkeys, mouse wheel)
- Unterstützung von unterschiedlichen Einsatzumgebungen
 - Auf lokalem Rechner als auch auf einer Virtual Machine ausführbar
- Unterstützung von Robotic Process Automation (RPA) sowie Tests

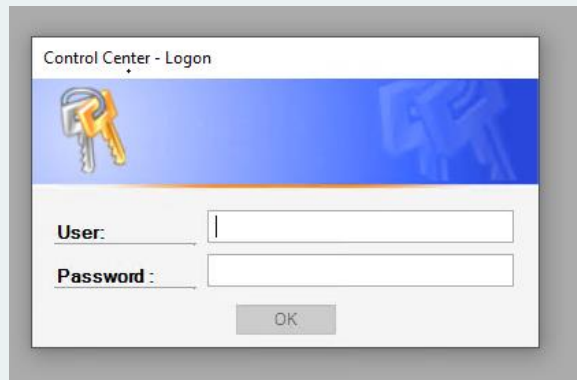
Inhalt

1. Motivation
2. G+D HMI Test Framework Idee und Struktur
3. Alternative Werkzeuge
4. Herausforderungen & Lösungsansätze
5. Zusammenfassung

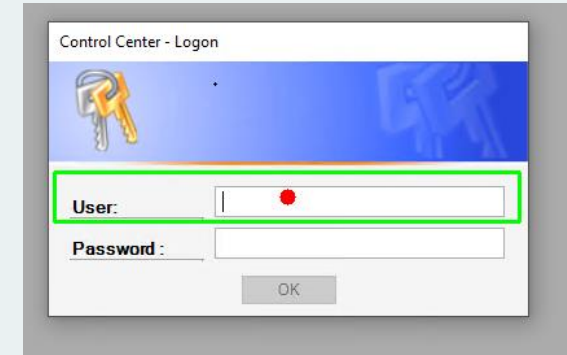
Image Matching Grundidee



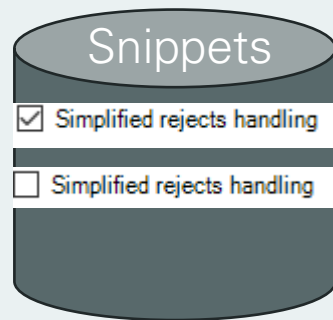
Screenshot



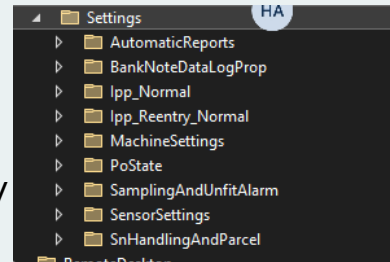
Finde Snippet im Screenshot



Beispiel: Checkbox toggeln

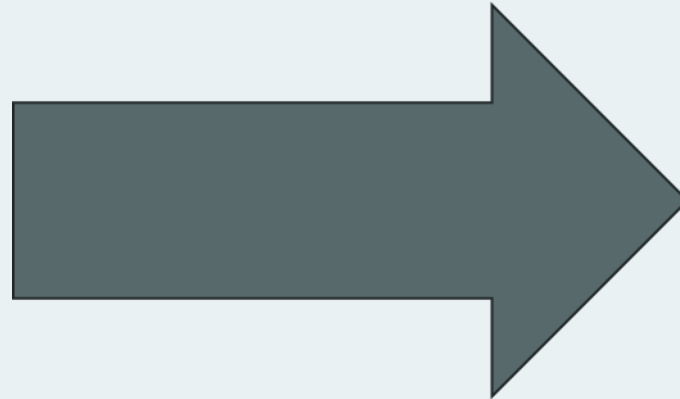


Verify
Click



Screenshot

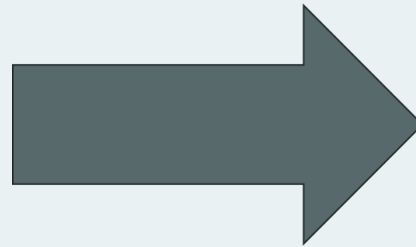
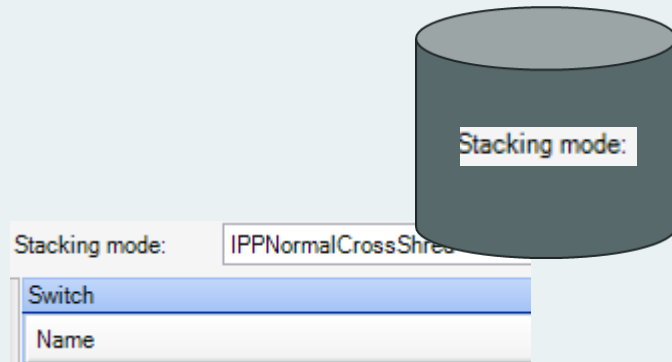
- ☒ Fast Jam Recovery
- ☒ Allow repeated input of reject BNs
- ☐ Simplified rejects handling
- ☐ IPP Normal Repass handling
- ☒ Postpone IPP reconciliation



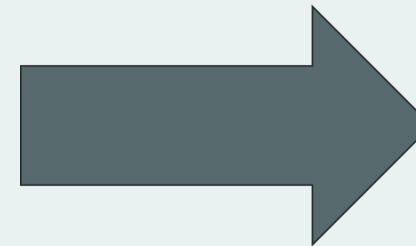
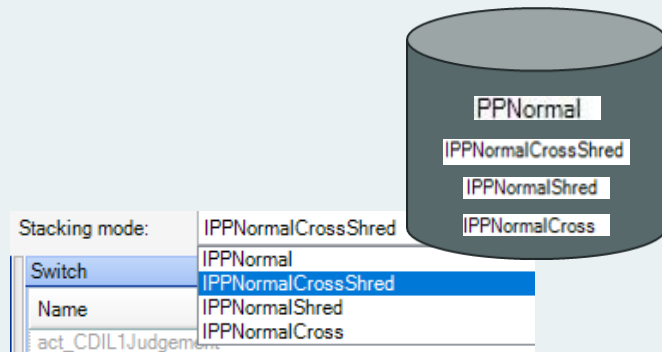
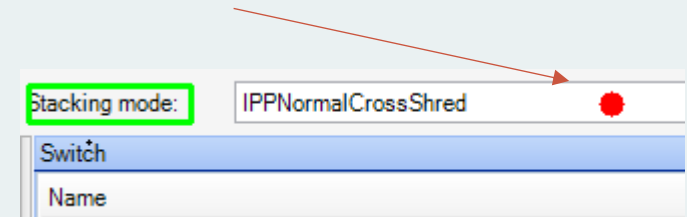
Finde Snippet im Screenshot

- ☒ Fast Jam Recovery
- ☒ Allow repeated input of reject BNs
- ☒ **Simplified rejects handling**
- ☐ IPP Normal Repass handling
- ☒ Postpone IPP reconciliation

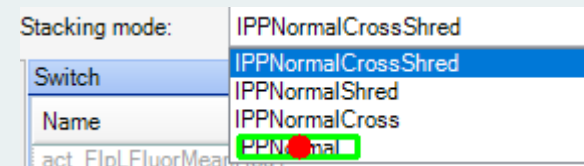
Beispiel: Dropdown Menü Selektion



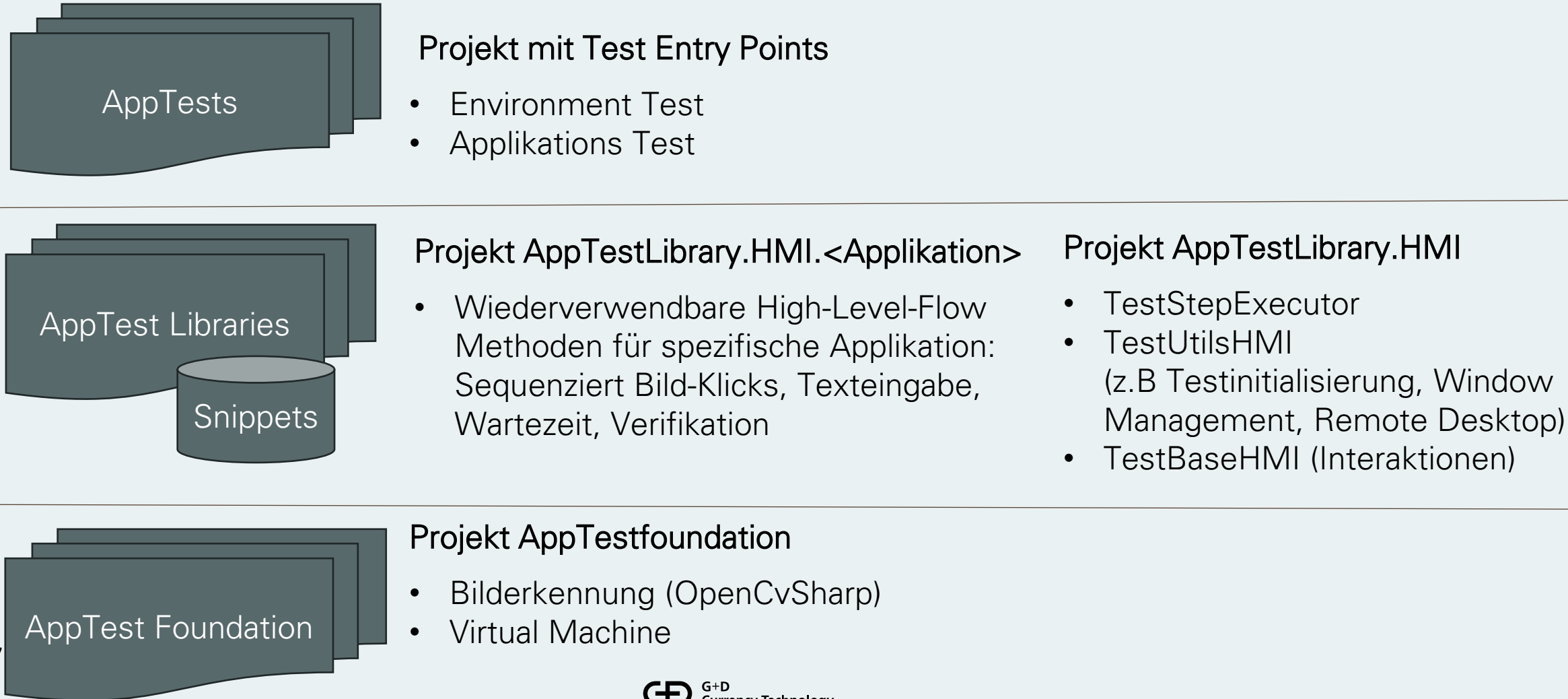
Nutze relative X-Offset



Beispiel: PPNormal



HMI Testautomation Struktur



AppTests

Projekt mit Test Entry Points

- Environment Test
- Applikations Test

AppTest Libraries

Snippets

Projekt AppTestLibrary.HMI.<Applikation>

- Wiederverwendbare High-Level-Flow Methoden für spezifische Applikation: Sequenziert Bild-Klicks, Texteingabe, Wartezeit, Verifikation

Projekt AppTestLibrary.HMI

- TestStepExecutor
- TestUtilsHMI (z.B Testinitialisierung, Window Management, Remote Desktop)
- TestBaseHMI (Interaktionen)

AppTest Foundation

Projekt AppTestfoundation

- Bilderkennung (OpenCvSharp)
- Virtual Machine

Beispiel: BPSX9 PO Creation via Control Center App

The screenshot displays the 'Control Center - Operation Settings' application window. The 'Machine settings' tab is active, showing various configuration options for the BPSX9 PO creation process. The interface includes a menu bar (File, View, PO, Tools, Window, Help), a toolbar, and a left-hand navigation pane with a tree view. The tree view shows a hierarchy starting with 'New POs', followed by 'LongPO_GD2000_CDI', and then several sub-items including 'Machine settings', which is currently selected. The main area is divided into four sections: 'Carrier configuration', 'Label printer settings', 'Bander settings', and 'Additional settings'. Each section contains specific settings with dropdown menus, text boxes, and checkboxes. At the bottom, there are 'Reset', 'Save', and 'Cancel' buttons. The status bar at the very bottom provides information about the user (ADMIN1), the current project (New POs\LongPO_GD2000_CDI), and the date/time (5/12/2025 4:45 PM).

Control Center - Operation Settings

File View PO Tools Window Help

POs HA Machine settings

New POs

- LongPO_GD2000_CDI
 - Sensor settings
 - Trigger settings
 - IPP Normal
 - IPP Reentry Normal
 - IPP Reentry Shred
 - Machine settings
 - SN Handling & Parcel
 - Sampling & Unfit Alarm
 - Automatic Reports
 - Banknote data log prop
 - PO State
- Modified POs

Carrier configuration

Carrier loading mode: Manual

Carrier size: 1000

Ignore invalid RFID data: Disabled

Label printer settings

Mode: Automatic

Layout ID: 0

SN 1 content: OPA01 First SN

SN 2 content: OPA10 Last SN

Number of copies: 2

Bander settings

Bander string: %O: %D8 %T8 OPA: %E4 C

Bander font: Latin_Cyrillic

Band length: 207

Starting print position: 30

Date format (%D)

December 31st 1999: Separator:

☒ 31 12 99 ☒ .

☐ 12 31 99 ☐ -

☐ 99 12 31 ☐ /

☐ 31121999 ☐ blank space

☐ 12311999 ☐ none

☐ 19991231

Additional settings

No. of rejects to singler stop: 80

Number of SN errors to warning: 15

No. of SN errors to singler stop: 20

Order number (ASCII characters only):

GTIN of Denomination:

Stop after OPP

☐ Unlimited

☒ Stop after amount of OPPs: 101111

☐ Stop after OPP number: 0

☒ Fast Jam Recovery

☐ Allow repeated input of reject BNs

☒ Simplified rejects handling

☐ IPP Normal Repass handling

Reset Save Cancel

logged on: ADMIN1 (Admin/Production) @ EcoCenter New POs\LongPO_GD2000_CDI (DM2_GD_2K_G2_O4_STQ_2024_11_05_V1_17.0.2445.0)\Machine settings 5/12/2025 4:45 PM

Verschiedene Test Entry Points

1. Environmenttest

- Führen Applikationstests in einer anderen Umgebung durch
zB einer Virtuellen Machine (VM)

2. Applikationstest

- Sind für Ausführung der Applikation auf dem host bzw desktop designed aber können auch in anderen Umgebungen wieder verwendet werden.
- Rufen Flows (= Testschrittsequenzen) auf - Daten sollten parameterisiert werden!
- Beide Testtypen können am Ende über NUnit aufgerufen werden.

```
/// <summary>
/// (based on SNC1 adaptation)
/// Name of PO : PO_Dev
/// </summary>
1 reference | 0/1 passing | Hechenblaikner Kristian Aruna, 5 days ago | 1 author, 1 change
public static void TE_0101_Create_LongPO_RD(TestContext testContext)
{
    RunPoCreationOnRemoteDesktop(testContext, PO_DEV, DM2_SNC1_BLK_0_G2_01_STQ, TC_0101_Create_LongPO);
}
```

```
#region Application Tests
2 references | 0/1 passing | Hechenblaikner Kristian, 5 days ago | 2 authors, 2 changes
public static void TC_0101_Create_LongPO(TestContext testContext)
{
    POCreationFlows.NewPOCreationFlow(TestStepsPoSettings.Po_Dev_DM2_Default(), PO_DEV, DM2_SNC1_BLK_0_G2_01_STQ, "PO_Dev");
}
```



Beispiel: Test Daten Design für Flows

Die TestStepsPoSettings Datenklasse fasst zB alle Daten des Machine Settings Tabs zusammen

Design Ideen:

- 1 Daten Klasse pro Tab
- Enthält nur Felder, die benutzt werden!

```
namespace AppTestLibrary.HMI.ControlCenter
{
    16 references | Hechenblaikner Kristian, 7 days ago | 2 authors, 3 changes
    public static class POCreationFlows
    {
        #region Application Flow
        11 references | Hechenblaikner Kristian, 7 days ago | 1 author, 2 changes
        public static void NewPOCreationFlow(TestStepsPoSettings testStepsPoSettings, string aroFileName, string adaptationPrefix, string poName)
        {
            TestStepsPoSetup.CopyOperationsForPOCreations(adaptationPrefix, aroFileName);
            TestStepsLogOn.PRE_LogonToControlCenter();
            TestStepsOperationSettings.InstallAdaptation();
            TestStepsOperationSettings.OpenAndCreatePO(poName);
            TestStepsOperationSettings.Updated_Po_Settings(testStepsPoSettings);
            TestStepsOperationSettings.Deploy_PO(testStepsPoSettings, adaptationPrefix, poName);
        }
    }
}
```

```
/// <summary>
/// (based on SNC1 adaptation)
/// </summary>
/// <returns></returns>
2 references | Hechenblaikner Kristian Aruna, 46 days ago | 1 author, 4 changes
public static TestStepsPoSettings Po_Dev_DM2_Default()
{
    return new TestStepsPoSettings(machineSettings: new MachineSettings { RejectsToSinglerStop = 80, oppSizeValue: 2,
        samplingAndUnfitAlarmSettings: new SamplingAndUnfitAlarmSettings { UnfitAlarmActive = true, UnfitAlarmNumberOfBlocks = 10 });
}
```

```
public class MachineSettings
{
    11 references | Hechenblaikner Kristian Aruna, 53 days ago | 1 author, 1 change
    public int? RejectsToSinglerStop { get; set; } = null;
    3 references | Hechenblaikner Kristian Aruna, 53 days ago | 1 author, 1 change
    public string OrderNumber { get; set; } = "";
    3 references | Hechenblaikner Kristian Aruna, 53 days ago | 1 author, 1 change
    public string DenominationGTIN { get; set; } = "";
    9 references | Hechenblaikner Kristian Aruna, 53 days ago | 1 author, 1 change
    public int? StopAfterOPPNNumber { get; set; } = null;
    6 references | Hechenblaikner Kristian Aruna, 42 days ago | 1 author, 1 change
    public int? StopAfterNumberOfOPPs { get; set; } = null;
    3 references | Hechenblaikner Kristian Aruna, 53 days ago | 1 author, 1 change
    public bool FastJamRecovery { get; set; } = true;
    3 references | Hechenblaikner Kristian Aruna, 53 days ago | 1 author, 1 change
    public bool AllowRepeatedRejectBNS { get; set; } = true;
    12 references | Hechenblaikner Kristian Aruna, 53 days ago | 1 author, 1 change
    public bool SimplifiedRejectsHandling { get; set; } = true;
    9 references | Hechenblaikner Kristian Aruna, 53 days ago | 1 author, 1 change
    public bool PostponeIPPReconciliation { get; set; } = true;
    4 references | Hechenblaikner Kristian Aruna, 53 days ago | 1 author, 1 change
    public bool IppNormalRepassHandling { get; set; } = false;
    6 references | Hechenblaikner Kristian Aruna, 53 days ago | 1 author, 1 change
    public LabelPrinterSettings? LabelPrinterSetting { get; set; } = LabelPrinterSettings.Automatic;
    9 references | Hechenblaikner Kristian Aruna, 53 days ago | 1 author, 1 change
    public CarrierLoadingModes? CarrierLoadingMode { get; set; } = CarrierLoadingModes.Manual;
}
```

Beispiel: Parametrisierter Flow

```
#region Application Tests
```

```
2 references | 0/1 passing | Hechenblaikner Kristian, 5 days ago | 2 authors, 2 changes
```

```
public static void TC_0101_Create_LongPO(TestContext testContext)
{
    POCreationFlows.NewPOCreationFlow(TestStepsPoSettings.Po_Dev_DM2_Default(), PO_DEV, DM2_SNC1_BLK_0_G2_01_STQ, "PO_Dev");
}
```

```
namespace AppTestLibrary.HMI.ControlCenter
```

```
{
```

```
16 references | Hechenblaikner Kristian, 7 days ago | 2 authors, 3 changes
```

```
public static class POCreationFlows
```

```
{
```

```
    #region Application Flow
```

```
11 references | Hechenblaikner Kristian, 7 days ago | 1 author, 2 changes
```

```
public static void NewPOCreationFlow(TestStepsPoSettings testStepsPoSettings, string aroFileName, string adaptationPrefix, string poName)
```

```
{
```

```
    TestStepsPoSetUp.CopyOperationsForPOCreations(adaptationPrefix, aroFileName);
    TestStepsLogOn.PRE_LogonToControlCenter();
    TestStepsOperationSettings.InstallAdaptation();
    TestStepsOperationSettings.OpenAndCreatePO(poName);
    TestStepsOperationSettings.Updated_Po_Settings(testStepsPoSettings);
    TestStepsOperationSettings.Deploy_PO(testStepsPoSettings, adaptationPrefix, poName);
}
```

```
namespace AppTestLibrary.HMI.ControlCenter
```

```
{
```

```
1 reference | Whitespace, 28 days ago | 3 authors, 10 changes
```

```
public class TestStepsLogOn
```

```
{
```

```
1 reference | Hechenblaikner Kristian, 42 days ago | 2 authors, 7 changes
```

```
public static void PRE_LogonToControlCenter()
```

```
{
```

```
    TestStepsBaseHMI.ObjectFolder = @"AppTestLibrary.HMI.ControlCenter\Images\OperationSettings\PO_LogOn_Off_Deploy";
    TestStepsOperationSettings.ClickCancelTesterComponent();
    Executor.ClickObject(PRE_LogonToControlCenter.CC_LOGON_PANEL, failVerdict: TestVerdict.Verdict.Inconclusive);
    Executor.ClickObject(PRE_LogonToControlCenter.CC_LOGON_USER_TEXTBOX, failVerdict: TestVerdict.Verdict.Inconclusive);
    EnterTextString("ADMIN1");
    Executor.ClickObject(PRE_LogonToControlCenter.CC_LOGON_PASSWORD_TEXTBOX, failVerdict: TestVerdict.Verdict.Inconclusive);
    EnterTextString("ADMIN1");
    Executor.ClickObject(PRE_LogonToControlCenter.CC_LOGON_OK_BTN, failVerdict: TestVerdict.Verdict.Inconclusive);
    Executor.ClickObject(PRE_LogonToControlCenter.CC_START_SCREEN, failVerdict: TestVerdict.Verdict.Inconclusive);
}
```

OpenCV

- Open Source Computer Vision Library
- Ziel: Echtzeit-Bild- und Videoverarbeitung.

Lizenzierung

- Apache 2.0 License: kostenlose kommerzielle und private Nutzung.

Kernmodule

- **core**: Grundlegende Datenstrukturen und Operationen.
- **imgproc**: Bildfilterung, Kantenerkennung, Morphologie. (Template Matching)
- **highgui**: Einfache GUI-Funktionen und Videoinput/-output.
- **video**: Bewegungsanalyse, Hintergrundsubtraktion.
- ...
- Große Community, kontinuierliche Weiterentwicklung.

Wir benutzen das Template Matching von OpenCv für das Image Matching

12 **Alternativen:** Feature-Detektoren & Deskriptoren, lernbasierte Verfahren

Alternative Image Matching Tools

Tool	Stärken	Lizenz
SikuliX	Screenshot-basiertes Matching und Steuerung, plattformunabhängig	Apache 2.0
Percy	Cloud-basierte Visual Regression mit Baseline-Management	Proprietär, kommerziell
BackstopJS	Open-Source Visual Regression für Web-Apps, hoch konfigurierbar	MIT
Resemble.js	JavaScript-Library für Bildvergleich und Diff-Visualisierung	MIT

Bekannte kommerzielle UI Automatisierungstools

Tool & Anbieter	Kernfunktion(en)
Eggplant Test (Keysight)	Rein bildbasiertes Framework mit AI-unterstütztem „model-based digital twin“, OCR-Kombi und Template-Verwaltung.
TestComplete (SmartBear)	GUI- und Web-Tests mit „Image-Based Actions“, einstellbaren Ähnlichkeits-Thresholds und integriertem Recorder.
Ranorex Studio	Low-Code Recorder plus image-basiertes Matching (SSD, NCC), Mobile- und Desktop-Support.
Squish GUI Tester	Plattformübergreifend (Qt, Java, Web, Windows) mit hybriden Lookup-Methoden und OCR-Integration.
Applitools Eyes	Visuelle Regression via „Visual AI“, adaptive Baseline-Verwaltung und intelligente Falsch-Positiv-Filter.

Alternative Ansätze zur UI-Testautomatisierung

1) Accessibility-APIs / UI-Automation-Frameworks

- **Beispiele:** Microsoft UIA, WinAppDriver, FlaUI, White Framework
- **Wie es funktioniert:** Greift per Accessibility-Tree auf Controls zu (Automation IDs, Properties)

2) DOM-/Struktur-basierte Web-Automation

- **Beispiele:** Selenium WebDriver, Playwright, Puppeteer
- **Wie es funktioniert:** Identifiziert HTML-Elemente per CSS-Selektor oder XPath und interagiert programmatisch

3) Deep-Learning-Objekterkennung

- **Beispiele:** YOLO, SSD, Faster R-CNN
- **Wie es funktioniert:** Trainierte CNN-Modelle erkennen & lokalisieren UI-Elemente direkt im Screenshot

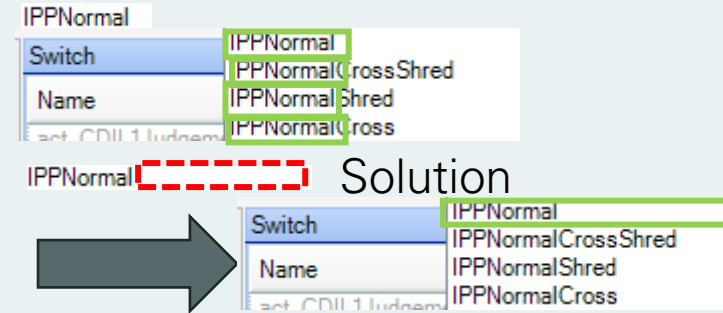
Warum Image Matching für Legacy ideal ist

- **Plattform-agnostisch & API-unabhängig**
Arbeitet rein auf Pixel-Ebene, ganz ohne spezielle Automation-Provider.
- **Breite Einsatzgebiete**
Funktioniert in 3270/5250-Emulatoren, RDP/Citrix-Sitzungen, VB6- und PowerBuilder-Anwendungen.
- **Stabile, unveränderte Layouts**
Legacy-GUIs laufen oft jahrelang mit identischem Design – Referenzbilder bleiben gültig.
- **Skalierbar & parallel ausführbar**
Tests lassen sich auf VMs oder Containern farmen, um große Testpools zeitgleich zu absolvieren.
- **Schneller Prototyp-Aufbau**
Kein Framework-Setup nötig, einfache Integration via OpenCV-Template Matching.(Proof of Concept)
- **Einfache CI/CD-Integration**
Nahtlose Einbindung in Jenkins, GitLab CI oder Azure DevOps für automatisierte Builds & Tests.
- **Geringer Wartungsaufwand**
Solange Auflösung und DPI-Factor konstant sind, muss man nur selten neue Snippets pflegen.
- **Signifikante Zeitersparnis**
Manuelle Testzyklen, die früher Stunden dauerten, lassen sich per Image Matching in wenigen Minuten ausführen – bis zu 90 % schneller.

Herausforderungen & Lösungsansätze

Bilder

- Vermeide Ambiguität im Bild
- Bei minimalen Unterschieden den Threshold strenger setzen



Dynamik & Timing

- **Nicht-deterministische Ladezeiten**
 - Variierende Antwortzeiten von UI-Elementen
 - Über Skripte oder sich wiederholdendes Image Matching möglich
- **Asynchrone Prozesse**
 - Hintergrund-Threads, Animationen, Lazy-Loading

Fenster- & Fokus-Management

- **Mehrere Fenster/Pop-ups**
 - Gezielt minimieren, in den Vordergrund holen
- **Multi-Monitor-Setups**
 - Fensterposition relativ zum Hauptbildschirm

Herausforderungen & Lösungsansätze

Störfaktor beim Image Matching	Lösungsansatz
Skalierung & Auflösung	Multi-Scale Matching (Bildpyramiden), automatische Größen-Anpassung
Rotation & Orientierung	Templates in mehreren Rotationen
Beleuchtungs-/Kontraständerungen	Histogramm-Equalization, adaptive Kontrast-/Helligkeitsanpassung,
Rauschen & Artefakte	Vorverarbeitung: Glättung (Median-Filter), Rauschreduzierung
Variierende DPI-/Zoom-Faktoren	Einheitliche DPI-Einstellung, Multi-Scale Matching (Bildpyramiden)
Kleine Templates oder große Suchbereiche	Passende Template-Größe wählen, Region-of-Interest einschränken
Ähnliche oder wiederkehrende Symbole	Kombination mehrerer Teil-Templates, Kontext-Prüfung

Best Practices für Image Matching

- **ROI einschränken**
 - Croppen des Screenshots auf den relevanten Bereich vermindert Rechenaufwand und Fehllalarme.
- **Vorverarbeitung**
 - Graustufen-Konvertierung + Histogramm-Equalization für gleichmäßigere Kontrastverhältnisse.
- **Multi-Scale Matching / DPI-Skalierungs-Handling**
 - Templates in mehreren Größen durchsuchen, falls UI-Elemente dynamisch skaliert werden.
- **Rotationstoleranz**
 - Entweder mehrere rotierte Templates parat halten oder Feature-Matching (ORB/SIFT) ergänzen.
- **Thresholding**
 - Vor dem Klick prüfen, ob $\text{maxVal} > 0.8$ (oder ein selbst definierter Wert) um Fehlklicks zu vermeiden.
- **Fallback-Strategien**
 - Wenn kein Match über dem Threshold, warte kurz, aktualisiere Screenshot, und versuche es nochmal – statt blind zu klicken.
- **Synchronisation**
 - Warte auf bestimmte Screens oder Ladeindikatoren (ebenfalls per Bild-Matching)

Zusammenfassung

Herausforderung

- Viele Legacy-Systeme bieten keinen API-Zugang – nur GUI-basierte Bedienung
- Manuelle Tests sind aufwendig, fehleranfällig und nicht skalierbar
- **Unser Ansatz:**
 - **Image Matching** mit OpenCV zur visuellen Erkennung von UI-Elementen
 - **Snippets** ermöglichen präzise Aktionen wie Klicks, Texteingaben, Dropdown-Auswahl
- Plattformunabhängig & non-intrusiv – kein Eingriff in den Anwendungscode nötig

Bildbasierte Automatisierung ist ein robuster und flexibler Weg, um Legacy-Systeme zuverlässig zu testen und in moderne CI/CD-Prozesse zu integrieren.

Technische Struktur

- Klare Trennung von Test Entry Points, Libraries und Snippets
- Wiederverwendbare Testflüsse & Datenstrukturen für hohe Wartbarkeit
- Automatisierung steuerbar auf lokalen Maschinen & virtuellen Umgebungen
- Auch auf Servern ausführbar über VMs

Vorteile

- Schnell erste Ergebnisse möglich (PoC)
- Ideal bei stabilen, langjährig genutzten Legacy-UIS
- Signifikante Zeitersparnis
- Erweiterbar durch OCR & Feature-Matching