

Anwendungsautomatisierung

mateo unterstützt die Automatisierung unterschiedlicher Anwendungen und Technologien. Anwendungsübergreifend können Daten ausgetauscht und überprüft werden. Jede Technologie besitzt einen eigenen mateo-Treiber, welcher unkompliziert um neue Anforderungen und Features erweitert werden kann.



Web-Anwendungen (auch Headless ohne grafische Oberfläche)

mateo bietet einen Treiber auf Basis von Selenium, der auch mit Shadow-DOMs, StaleElements und mehreren Browserfenstern funktioniert.



Windows-Anwendungen (Win32, .NET, WPF ...)

mateo bietet unterschiedliche Treiber für unterschiedliche Windows 'Fat-Client' Anwendungen. Je nach Alter und Zugänglichkeit bietet mateo für die Windows Welt Treiber, die sowohl Win32 Anwendungen, .NET und WPF unter Kontrolle bringen können.



Java-Anwendungen

mateo bietet einen Java-Treiber, der Java Elemente identifizieren und steuern kann.



Mobile Anwendungen (Android / iOS)

Auf Basis von Appium bietet mateo eine Schnittstelle, um automatisiert echte Mobilgeräte zu steuern oder Emulatoren / Simulatoren zur Qualitätssicherung einzusetzen.



SAP

mateo bietet einen SAP-Treiber, der das SAP Gui Scripting nahtlos mit allen anderen Treibern von mateo nicht nur verfügbar macht, sondern ebenfalls eine Überprüfung von Laufzeiten und Einzelkommandoausführung ergänzt.



Host (IBM Mainframe)

mateo hilft bei der Transformation von modernen Datentypen wie xml und json in Host eigene Datentypen und umgekehrt auf Basis von cobol2xml.



Progressive Web-Apps

Progressive Web-Apps können mit dem Web-Treiber von mateo analog zu normalen Web-Seiten getestet werden. Selenium bietet mateo die Basis für die notwendige Stabilität und Flexibilität, die zur effektiven Qualitätssicherung benötigt wird.



Bildbasierte Steuerung

Anwendungen, die keine eigene Automationschnittstelle bieten, lassen sich teilweise nur über Bildvergleiche effizient steuern. Dabei werden Screenshots der Anwendung erstellt, die mateo sagen, welche Elemente geklickt oder befüllt, sowie welche Ergebnisse überprüft werden sollen. mateo bietet Treiber auf Basis von AutoHotkey, Autolt und

Sikuli, die auf unterschiedlichen Plattformen die Bilderkennung effizient ausführen. Ferner gibt es einen weiteren Treiber zur 'Optical Character Recognition', also der Texterkennung in Bildern, um sowohl Text auf dem Bildschirm zu suchen und zu klicken, sowie Text zu erkennen, um ihn zu überprüfen.

Continuous Integration

Regelmäßige automatische Durchführung von Codeänderungen in einem zentralen Repository, die vor der Auslieferung durch das mateo-Team getestet werden.



Upload und Start von Testfällen über HTTP-Webservices)

mateo bietet eine REST-Schnittstelle, die es ermöglicht, Skripte über HTTP(S)-Anfragen hochzuladen und zu starten. So kann mateo auch ohne Zugriff auf Web-Interface oder Dateisystem der Maschine, auf der mateo installiert wurde, genutzt werden. Dies ermöglicht auch die einfache Einbindung in Continuous-Integration-Lösungen wie Jenkins oder GitLab.



Starten über Aufgabensteuerung (wie Scheduled Tasks, Cron Jobs)

mateo kann über die Kommandozeile gestartet werden. Dies gilt auch für den Start einzelner Skripte. So lässt sich mateo flexibel in Aufgabensteuerungsanwendungen wie Cron (Linux) oder die Aufgabenplanung (Windows) einbinden, um die terminierte und regelmäßige Ausführung von Skripten zu ermöglichen.

Cloud Image zur Cloud-Automatisierung von Web-Anwendungen



Container- und Kubernetes-fähig

mateo kann auch als Container gestartet werden, der sowohl bei den gängigen Cloud-Anbietern (Google Cloud Platform, Azure, AWS), als auch in einer Private Cloud (z. B. OpenShift) funktioniert. Hiermit können Ressourcen gespart und Hintergrundautomatisierung realisiert werden.



Horizontales wie auch automatisches Skalieren möglich

Zur Skalierung von durchzuführenden Automatisierungsaufgaben können in einer Container- oder Cloud-Umgebung durch das bereitgestellte Docker Image schnell und unkompliziert beliebig viele Instanzen von mateo bereitgestellt werden.

Dokumentautomatisierung



Texterkennung (OCR)

mateo kann Texte aus übergebenem Bildmaterial extrahieren, darin im Anschluss z. B. ein Schlüsselwort identifizieren und es darauffolgend anklicken. Dabei werden je nach Wunsch und Bedarf etablierte Lösungen wie Tesseract (offline nutzbar) oder auch Cloud-Vision-Dienste von Google (online) genutzt.



PDF-Analyse

Textinhalte können aus PDF-Dateien extrahiert werden. Im Anschluss können diese, z. B. mit Hilfe von regulären Ausdrücken, auf gewünschte Muster hin untersucht werden, um bspw. die Korrektheit von Angaben in Druckstücken zu prüfen.



Lesen/Schreiben (JSON, CSV, XLSX, PDF, ...)

mateo unterstützt eine Vielzahl von Ein- und Ausgabeformaten. So können Sie z. B.:

- Ergebnisse aus Datenbankabfragen (SQL) als JSON oder CSV speichern
- Excel-Dateien (z. B. XLSX) als Datengrundlage für Testfälle und als Ergebnisspeicher verwenden
- PDF-Reporte in Abhängigkeit der zu betrachtenden Vorgänge erzeugen

Einfache Testfall- und Automationserstellung und Wartung

Die Erstellung und Wartung von Testfällen und Automationen ist die zentrale Kernaufgabe einer Automationssoftware. Je nach Wissensstand und Automationsanforderungen können unterschiedliche Bedarfe an Form, Modularisierung und Arbeitsmodus vorliegen. mateo bietet ein breites Set an Möglichkeiten, die sie für ihre Bedürfnisse nach Bedarf verwenden können.



Prozessbeschreibung in Excel

Excel bietet für Neueinsteiger der Automatisierung und Fachbereichsmitarbeiter ein bekanntes Tooling zur Definition von Testfällen und Automatisierungen. Ferner können Sie fachliche Definitionen und Testdaten in Excel festlegen und die technische Abwicklung z.B. in mateoScript (s.U.) durchführen.



Prozessbeschreibung in Visual Studio Code mit mateoScript

mateoScript ist eine an Python angelehnte Scriptsprache zur Definition von Automatisierungen und Testfällen. Für Visual Studio Code ist ein Plugin verfügbar, das die gewohnten Funktionen von modernen Entwicklungsumgebungen (Code Completion, Syntax Highlighting, ...) für mateoScript verfügbar macht. Visual Studio Code ist kostenlos und leichtgewichtig und bietet so für mateo eine optimale technische Entwicklungsplattform für PowerUser.



Prozessbeschreibung in Blockly

Blockly ist eine von Google unterstützte freie Bibliothek zur graphischen Entwicklung von Programmen. mateo unterstützt einen Blockly Editor, der eine transparente Übersetzung in echtes mateoScript ermöglicht und so auch Mitarbeitenden mit wenig bis gar keinen Programmierkenntnissen die Definition von Testfällen und Automationen ermöglicht. Gleichzeitig wird ein Verständnis für mateoScript aufgebaut, und somit die Lesbarkeit der technischen Implementierung erhöht.



Einzelausführung von Kommandos

Die Entwicklung und Wartung von Testfällen und Automationen kann zeitintensiv und schwierig sein. Insbesondere bei langen Ausführungsketten ist es hinderlich, wenn die Automation immer wieder initial gestartet werden muss. mateo unterstützt bei den meisten Treibern die Einzelausführung von Kommandos, so dass einzelne Kommandos überprüft werden können und sukzessiv einzelne Schritte weiterentwickelt werden können.



Ausführliche Dokumentation

mateo bietet eine umfängliche Dokumentation von der Installation der Software bis hin zur Anwendung spezifischer Features. Jeder Treiber, der mit mateo ausgeliefert wird, beinhaltet eine Dokumentation der mitgelieferten Kommandos. Die Dokumentation kann schnell auf Stichworte hin untersucht werden und wird mit jeder Version aktualisiert.



Credential Management mit KeePass

In nahezu jedem Unternehmen gibt es Software mit Credentials (Benutzername/Kennwort). Damit die Zugangsdaten nicht für jedermann zugänglich sind oder im Log auftauchen unterstützt mateo KeePass. Hier können Kennwörter auf Basis eines zentralen Kennworts abgelegt werden, die innerhalb der Automation benötigt werden.



Separierte Exception Handler

Testfälle und Automationen laufen auf Fehler! Es ist bereits im Vorfeld wichtig, sich Gedanken zu machen, was dann passieren soll. mateo bietet sogenannte Exception Handler, um im Fehlerfall das System auf einen gewünschten Zielzustand zu bringen. Z.B, wird die Anwendung neugestartet, der Browser beendet oder Testdaten aufgeräumt. Auch hier sind der Kreativität keine Grenzen gesetzt.



Recording-Möglichkeit

mateo bietet die Möglichkeit, laufende Automationen aufzunehmen. Dies kann notwendig sein, um die Fehleranalyse von laufenden Testautomatisierungen zu vereinfachen, oder um Tutorial-Videos der aktuellen Oberfläche zu erstellen. Die erzeugten Videos können einfach an einem

beliebigen Ort abgelegt werden und sind somit unabhängig von dem grundsätzlichen Reporting von mateo.

Erweiterbarkeit und Modularisierung

mateo ist als Framework konzipiert. D.h. Sie können das Werkzeug um Ihre individuellen Anforderungen erweitern oder Veränderungen an bestehender Funktionalität vornehmen. Damit sind Sie in der Lage, das für Ihre Anwendungsfälle optimale Werkzeug zu schaffen. Workarounds gehören damit der Vergangenheit an.



Entwicklung eigener Kommandos

Neben der Vielzahl bestehender Kommandos, können Sie für spezielle Fälle eigenständig oder mit Unterstützung unseres Support-Teams, schlanke und effiziente Kommandos mittels Java-Programmierung entwickeln.



Wiederverwendbare Module (Repositories)

Schritte für wiederkehrende Prozesse (z. B. Eingaben von Benutzername und Passwort, Klick auf einen Button, Auswahl einer Umgebung, etc.) können Sie an einer Stelle definieren und an vielen Stellen wieder verwenden. Dabei ist es auch möglich, diesen Modulen Parameter wie Benutzernamen zu übergeben, so dass unterschiedliche User sich anmelden können.



Trennung von Ablaufbeschreibung und Prozessdaten (CSV, XLS)

Stellen Sie sich vor, Sie möchten über eine Maske sehr viele Datensätze eingeben. So haben wir mit mateo z.B. schon oft Testdatenbestände aufgebaut. Dafür war es nötig, viele Personen- und Vertragsdaten über eine Oberfläche einzugeben. Die Testdatenkombinationen kennen Ihre Fachabteilungen in der Regel sehr gut und die Fachabteilungen sind auch in der Lage, diese in Form von Exceldateien zeilenweise zu beschreiben. Genau diese Exceldateien können Sie mateo geben, um die Daten in die Masken einzugeben. Dazu ist lediglich die Beschreibung des Ablaufs nötig. In der Ablaufbeschreibung wird auf die Spalten in der Exceldatei verwiesen, die dann zeilenweise abgearbeitet wird.



Groovy-Integration (Embedded-Code-Funktionalität)

Groovy ist eine weit verbreitete Scriptsprache. Um komplexe Berechnungen z.B. vorzunehmen, haben wir in mateo die Möglichkeit integriert, groovy-Scripte auszuführen. Damit können Sie die Anzahl der automatisierbaren Anwendungsfälle deutlich erweitern.

Externes Reporting

Wenn Oberflächen automatisch bedient werden, werden Ergebnisse produziert. Im Kontext von Testautomatisierung möchten Sie schnell wissen, wo Fehler festgestellt wurden. Auch im Kontext von RPA wird Ihr Informationsbedarf hoch sein, weil im Zweifel viele Daten verarbeitet werden, die oft auch hohe Relevanz haben. Ihr Sicherheitsbedürfnis wird hoch sein.

mateo hilft Ihnen dabei, die benötigten Informationen schnell bereitzustellen. Schnittstellen zu diversen Systemen sind implementiert. Darüber hinaus bieten wir eine generische Schnittstelle an, so dass Sie schnell Ihre eigenen (individuellen) Berichtssysteme anschließen können. Zu folgenden Systemen bestehen bereits Schnittstellen:



Jira

mateo bietet eine Jira-Schnittstelle. So können im Fehlerfall Tickets so intelligent angelegt werden, dass sie nur einmal erzeugt werden, auch wenn z.B. Testfälle mehrfach in der Nacht ausgeführt werden.



XRay

XRay ist die populärste Testmanagement-Erweiterung von Jira. Auch nach XRay können automatisierte Testlauf-Beschreibungen exportiert werden.



Micro Focus ALM (vorm. HP QualityCenter)

Noch immer ist HP QC sehr weit verbreitet. Inzwischen wird das revisionssichere Werkzeug aber von Microfokus vertrieben und weiterentwickelt. mateo-Testfälle aus diesem Werkzeug können von dort gestartet werden und die Läufe werden natürlich auch im Anschluss dokumentiert. Weitere Testmanagement-Werkzeuge Auch für TestRail und TestLink bietet mateo Schnittstellen an.



Custom-Reporter-Schnittstelle

mateo bietet eine anpassbare Schnittstelle nach außen, die sog. Custom-Reporter-Schnittstelle. Auf diesem Weg können also weitere Systeme einfach angebunden werden.

Orchestrierung

Mittels des [mateo Orchestrators](#) können mehrere Instanzen von mateo kontrolliert und parallele Ausführungen von Testfällen über eine zentrale Applikationsinstanz gestartet werden.

Performance-Messung

War Ihre Anwendung am Wochenende langsamer als noch am Dienstag? Wie viel Zeit benötigte die automatisierte Anlage eines Vertrages im Mittel über die letzten 10 Läufe?

Solche und ähnliche Fragen können Sie mit unserem Performance-Logging beantworten. Die Laufzeiten eines jeden Tests / jedes Prozesses werden granular erfasst und stehen im Anschluss zur Auswertung bereit.



Visualisierung der Ergebnisse

Wir bieten ein Standard-Dashboard an. Wesentliche Zahlen sind damit adhoc sichtbar. Da mateo Ihnen aber auch die Datenbasis zur Verfügung stellt, können Sie auch Ihre ganz eigenen Berichts- und Dashboard-Systeme daran anschließen.



Speicherung von Laufzeiten in Datenbank

Zu jedem Kommando (bspw. ein Klick auf ein Element, das Auslesen eines Wertes aus einer Oberfläche oder das Schreiben einer Datei) speichert mateo auf Wunsch Start- und Endzeitpunkt. Zusammen mit Informationen über Start und Ende des gesamten Tests erhalten Sie so eine mächtige Datengrundlage zur Auswertung Ihrer Applikations- und Backend-Performance. Die Speicherung der Daten erfolgt dabei in einer Datenbank, wie z. B. MariaDB.

Prozessbasierte und anwendungsübergreifende Tests und Prozesse

Unsere betriebliche Welt ist sehr komplex. Wir alle bedienen unterschiedlich Anwendungen an unseren Rechnern, die gemeinsam verschiedene Prozesse abbilden.



Eine Beschreibungssprache für alle Arten von Anwendungen

Um diese Anwendungen zu automatisieren im Kontext von Testautomatisierung oder RPA wird damit ein Werkzeug wie mateo benötigt, das für alle Arten von Anwendungen eine Möglichkeit zur Automatisierung bietet. Da die Technologien sehr unterschiedlich sein können, kapselt mateo diese Komplexität weg und bietet eine Art von Beschreibungssprache an. Damit bleiben die Prinzipien über alle Anwendungen gleich und die Komplexität wird für Sie beherrschbar.

Revisionssichere Dokumentation

Alle ausgeführten Skripte von mateo werden revisionssicher in unterschiedlichen Formaten abgelegt. Dabei können sowohl das Layout, als auch die Inhalte konfiguriert und auf die gewünschten Bedürfnisse abgestimmt werden. Screenshots und/oder Videos können die Dokumentation abrunden und für vollständige Nachvollziehbarkeit ergänzen.

Schnittstellenautomatisierung

Automatisierung bedarf nicht nur das Bedienen eines einzelnen Systems, sondern benötigt ferner auch die Konfiguration der Umwelt, der Umgebungen oder das Simulieren bestimmter Schnittstellen. Hierfür bietet mateo unterschiedliche Treiber zum Ansprechen von Schnittstellen, die einen einfachen und vollständigen Automationsfluss ermöglichen.



Datenbanken (SQL, NoSQL)

Sämtliche JDBC-Treiber (Datenbanktreiber der Java-Welt) können verwendet werden, um unterschiedlichste Datenbanken anzusprechen.



HTTP-Webservices (REST, SOAP ...)

Webservice Kommandos ermöglichen die direkte Kommunikation mit HTTP, REST oder SOAP-Schnittstellen, welche ebenfalls über einfache Authentifizierung abgesichert sein können.



APIs (Microsoft Graph, Google Cloud ...)

Gängige APIs mit HTTP-Schnittstelle können über die Webservice Kommandos ebenfalls angesprochen werden, so dass eine nahtlose Integration z.B. von Cloud Services möglich wird.



Logik- und Daten-Output zu externen Process Engines wie Camunda

mateo kann interne Variablen nach außen reichen, um sie Prozess Engines oder sonstigen steuernden Applikationen zur Verfügung stellen, um diese dort vor- oder weiterverarbeiten zu können.



Asynchrone Inter-Bot-Kommunikation sowie Kommunikation mit Messaging-Systemen

Einzelne mateo Instanzen können sich Ressourcen teilen, um gegenseitig Informationen auszutauschen (z.B. welche Mails wurden bereits bearbeitet).

Sicherheit, Logging und Nachvollziehbarkeit

Automatisierung hat unterschiedliche Anforderungen an Sicherheit, Logging und allgemeine Nachvollziehbarkeit. Abhängig von den für Sie relevanten Use-Cases können sie mateo nach ihren Vorstellungen arbeiten lassen.



Detailliertes Logging

mateo schreibt ein detailliertes Log zur Fehleranalyse. Dabei wird einerseits ein Report mit Screenshots und Ausführungsparametern geschrieben, sowie ein tatsächliches Log der Java-Anwendung mit detaillierten Ausführungsinformationen bis hin zur programmatischen Ebene.



Debugging

Setzen von Breakpoints, Schritt-für-Schritt-Ausführung beim Aufbau von Testfällen und Prozessbeschreibungen



Sicherheit

- Sicherer Umgang mit Passwörtern
- Verschlüsselter Datenverkehr (SSL-Support)
- Logging deaktivierbar für sensible Umgebungen
- Access Log konfigurierbar zum Tracken der Anwendungszugriffe

Vollständige Automationsprache (in Excel)

Excel ist ein allgegenwärtiges Tool in der Enterprise-Welt mit den unterschiedlichsten Anwendungsfällen. Auf Grund des hohen Bekanntheitsgrads fühlen sich insbesondere Personen aus Fach- oder Testbereichen von Anfang an wohl. Mateos einfache Beschreibungssprache führt schrittweise an die logische Beschreibung einer Automatisierung bzw. Testfallbeschreibung und bietet ein funktionales Excel Template zum einfachen Handling der korrekten Syntax.



Schleifen

Eine Schleife ist eine mehrfache Ausführung derselben Schritte innerhalb einer Automation. Dabei werden in der Regel für jeden Durchlauf unterschiedliche Werte verwendet, z.B. der zu verwendende Kunde. Schleifen können in Excel durch die Verwendung von einfachen Schlüsselwörtern realisiert werden. Bei einer fehlerhaften Verwendung gibt mateo aussagekräftige Fehlermeldungen.



Bedingungen

Bedingungen sind, wenn-dann-sonst Entscheidungen innerhalb einer Automatisierung. Da nicht alle Abläufe eines Prozesses die exakt gleichen Schritte beinhalten, kann es notwendig sein auf Basis von Bedingungen entweder die eine oder die andere Aktion durchzuführen. Z.B. muss beim Versand eines Briefes an eine Firma das Feld "Firma" befüllt werden, für eine Natürliche Person das Feld "Vorname" und "Nachname". Bedingungen werden in mateo über einfache Schlüsselwörter dargestellt. Bei einer fehlerhaften Verwendung gibt mateo aussagekräftige Fehlermeldungen.

Weboberfläche

Die Weboberfläche ermöglicht es, dass Sie jederzeit auch von einem entfernten Rechner sehen können, in welchem Zustand sich mateo gerade befindet und was mateo gerade tut. Und natürlich können Sie so Testfälle oder automatisierte Prozesse auf einem anderen Rechner laufen lassen als Ihrem eigenen. Gleichzeitig liefert die Weboberfläche viele Informationen und bietet die Möglichkeit der Erstellung von Testfällen (Testautomatisierung) oder Prozessbeschreibungen (RPA).



Testfälle oder Prozessbeschreibungen anlegen

Über die Weboberfläche von mateo können Sie Testfälle oder Prozessbeschreibungen als Excel-Dateien oder mateo-Script-Dateien anlegen, bearbeiten oder löschen. mateo-Script-Datei können Sie sogar online, also über die Weboberfläche, bearbeiten. Damit sind Sie dann völlig autark von weiteren Installationen von Excel oder Visual Studio Code, denn mateo bringt alles mit.



Testfälle oder Prozesse durchführen

Sie können Excel- oder mateo-Script-Datei direkt über die Weboberfläche ausführen. Aktuell durchgeführte Kommandos werden am oberen Rand des Bildschirms angezeigt. Ein Hinweis in der Menübar zeigt Ihnen an, dass ein Lauf beendet wurde. Somit sehen Sie auch während mateo läuft, was er gerade tut. Hier können Sie Läufe auch beenden und Sie behalten stets das Heft in der Hand.



Reportanalyse

Die Report-Seite zeigt Ihnen die Berichte zu allen Läufen. Hier können Sie sehr detailliert sehen, was mateo getan hat, wie lange er dafür gebraucht hat und was das Ergebnis war. Ergebnisse gibt es von der Lauf- bis runter zur Kommando-Ebene. Hier können Sie auch auf einzelne Reports filtern und auch innerhalb der Reports wieder filtern z.B. auf bestimmte Ergebnis-Arten. Damit gelangen Sie schnell an die Ergebnisse, die Sie suchen.

Betriebssysteme

mateo unterstützt Sie über alle gängigen Plattformen hinweg. Ob Windows, Linux oder Mac: Durch den Unterbau in der etablierten Programmiersprache Java ist mateo nahezu jedem Test- und Betriebskontext gewachsen.



Windows

- Windows 7/10/11 sowie Server 2008 R2/2012 (inkl. R2)/2016/2019/2022
- Mindestens 1 GB RAM, empfohlen ≥ 2 GB RAM
- Benötigter Festplattenspeicher: 1 GB



Linux

- Benötigt eine installierte Java-Laufzeitumgebung ab Version 11
- Mindestens 1 GB RAM, empfohlen ≥ 2 GB RAM
- Benötigter Festplattenspeicher: 1 GB



Mac

- Benötigt eine installierte Java-Laufzeitumgebung ab Version 11
- Mindestens 1 GB RAM, empfohlen ≥ 2 GB RAM
- Benötigter Festplattenspeicher: 1 GB