

GERMAN TESTING MAGAZIN

Das unabhängige Magazin zu Software-Qualität



TESTAUTOMATISIERUNG ES GEHT UM MEHR ALS TESTDURCHFÜHRUNG

AUTOMATISIERTE
QUALITÄTSSICHERUNG
KI-BASIERTER SYSTEME

GETEILTE TEAMVERANT-
WORTUNG IN DER
(TEST-)AUTOMATISIERUNG

WIE RPA-BOTS DAS LEBEN
DES FACHBEREICHS
ERLEICHTERN



„SOFTWARE-TESTING IST ERWACHSEN GEWORDEN“ - INTERVIEW MIT HELMUT PICHLER, PRÄSIDENT DES ATB

»SOFTWARE-TESTING IST ERWACHSEN GEWORDEN«

„Software-Testing“ gilt manchmal noch als notwendiges Übel, als langweilig und unproduktiv. Dass dieser Eindruck täuscht, wird im Gespräch mit Helmut Pichler deutlich. Der Präsident des „Austrian Testing Board“ (ATB) ist Softwaretester aus Leidenschaft und engagiert sich daher stark für die Aufklärung über die Rolle, Tätigkeiten und vor allem die Erfolge des Softwaretestens. Denn letztlich lässt sich nur qualitativ hochwertige Software verkaufen – sei es an externe Kunden, sei es an interne Fachabteilungen im Unternehmen.

Allerdings muss auch das Software-Testing effizient und effektiv sein, betont Pichler im Interview. Ihm geht es dabei nicht nur um die eigentliche Testdurchführung, sondern darüber hinaus auch um die Automatisierung von Testimplementierung, -entwurf, -reporting und weiterer Testaktivitäten. Natürlich sprachen wir mit ihm auch über die Rolle des ATB, über Eigenheiten des österreichischen Marktes und insbesondere auch über internationale Standards sowie die Zusammenarbeit des ATB mit den Kollegen in Deutschland und der Schweiz.

Herr Pichler, seit mehr als 15 Jahren sind Sie Präsident des „Austrian Testing Board“. Worin sehen Sie die Hauptaufgaben des ATB?

Helmut Pichler: Wir stellen primär sicher, dass alle, die es wollen, in Österreich die weltweit erfolgreichsten und anerkannten Zertifikate im QA-Bereich erwerben können. Genau deshalb ist das ATB die offizielle Vertretung des „International Software Testing Qualifications Board“ – kurz ISTQB.

Im Jahr 2002 wurde das ATB gegründet und war dann auch Gründungsmitglied des ISTQB. Das ISTQB hat mit dem Certified-Tester-Schema einen weltweit etablierten Ausbildungsstandard für das Software-Testing geschaffen. Wir haben dafür gesorgt, dass man österreichweit über unseren geprüften Exam-Provider diese Zertifikate erlangen kann – dass es also Trainingsanbieter gibt, die mit vom Board geprüften und akkreditierten Unterlagen Ausbildungen zum umfangreichen ISTQB-Portfolio anbieten. Und wir kümmern uns darum, dass die angebotenen Zertifizierungsprüfungen qualitativ hochwertig sind.

Wir fungieren außerdem als unabhängige Anlaufstelle für Tester, all die Menschen, die es noch werden wollen und nicht zuletzt

natürlich auch für die gesamte Testing-Community in unserem Land. Die möchte ja auch auf dem Laufenden gehalten werden, was in Österreich beziehungsweise im DACH-Raum so abgeht.

„Die Unternehmen haben gelernt, dass eine Konformitätsnote 1 für das Befolgen der Agil-Bibel nicht reicht, wenn man professionelle Software mit guter Qualität erhalten will.“

Welche dieser Aufgaben können – und müssen Sie vielleicht sogar – national und sehr spezifisch adressieren? Und welche der ATB-Aufgaben sind eher länderübergreifend zu lösen, zum Beispiel im Rahmen des ISTQB oder mit ihren Nachbarn vom German und vom Swiss Testing Board, kurz GTB und STB?

Pichler: National gilt es einerseits, hiesige Trainingsanbieter zu finden und auch zu unterstützen, andererseits aber auch erfahrene Exam-Provider mit qualitativ hochwertigen Fragen auszustatten. Länderübergreifend fließt unsere Expertise in die Arbeitsgruppen des ISTQB ein, die Begriffe definieren und die Lehrpläne für die verschiedenen Module erstellen und weiterentwickeln.

Hinzu kommt das große Thema der Lokalisierung aller ISTQB-Materialien. Da bin ich sehr dankbar, dass wir das gemeinsam mit unseren Nachbar-Boards in Deutschland und der Schweiz – GTB und STB – umsetzen können, mit denen wir intensiv, produktiv und vor allem freundschaftlich zusammenarbeiten. Dass es beispielsweise eine deutsche Übersetzung der Lehrpläne und Begriffe gibt, ist unserer guten Zusammenarbeit zu verdanken. Alleine könnten wir das gar nicht schaffen.

Wenn Sie einen Vergleich wagen: Wie hat sich das Berufsbild des Softwaretesters in Österreich in den letzten 15 Jahren verändert?

Pichler: War anfangs oft noch Überzeugungsarbeit nötig, dass es gschiet ist, in die Ausbildung von Testern zu investieren, wurde der Softwaretest später völlig selbstverständlich. Unternehmen begannen sogar schon, ISTQB-Zertifikate als Voraussetzung für den Job als Tester zu verlangen.

Dann aber wurde die Softwareentwicklung „agil“, was zunächst für einen kleinen „Rückschlag“ sorgte. Die Ära der lupenreinen Agile-Berater brach an, die zwar keinerlei Ahnung von Software-Engineering hatten, dafür aber die „agilen Prinzipien“ rauf und runter rezitieren konnten. Und während jeder Punkt und jedes Komma im „Manifesto for Agile Software Development“ verpflichtend umzusetzen war (sonst war man ja nicht agil), wurde beim Test gerne geschludert. Der Grund ist so einfach wie banal: Im Manifesto kommen in der Team-Definition gar kein Tester vor.

„Mittlerweile sehe ich erfreulicherweise, dass das Thema Testen in den Unternehmen angekommen ist. Es werden am Markt händeringend Tester gesucht – und unser Zertifikat ISTQB CT FL ist nahezu überall ein Must-have bei den Stellenausschreibungen.“

Diese heikle Phase ist Gott sei Dank vorbei. Die Unternehmen haben gelernt, dass eine Konformitätsnote 1 für das Befolgen der Agil-Bibel nicht reicht, wenn man professionelle Software mit guter Qualität erhalten will. Und so sind nach und nach die Tester wieder ins Rennen gekommen. Aktuell gibt

es kaum noch Unternehmen, in denen die Tester nicht zu den Erfolgsfaktoren zählen; hier sag ich nur Testautomatisierung.

„Der Fokus verlagert sich jedoch vom rein manuellen Test hin zu technisch versierterem Testen. Das hat wohl vor allem mit dem Setting – Stichwort ‚Agile Teams‘ – zu tun.“

Wie ist es insgesamt heute in Österreich um die Disziplin des Softwaretestens bestellt?

Pichler: Mittlerweile sehe ich erfreulicherweise, dass das Thema Testen in den Unternehmen angekommen ist. Es werden am Markt händierend Tester gesucht – und unser Zertifikat ISTQB CT FL ist nahezu überall ein Must-have bei den Stellenausschreibungen.

Der Fokus verlagert sich jedoch vom rein manuellen Test hin zu technisch versierterem Testen. Das hat wohl vor allem mit dem Setting – Stichwort „Agile Teams“ – zu tun, weil hier die Tests direkt in die Tool-Frameworks

eingebunden sind. Eines ist doch klar: Continuous Integration & Development bedeutet eben auch Continuous Testing. Falls die Tester dann auch noch Automatisierungs-Skills mitbringen, sind sie sowieso die Helden (schmunzelt). Für erfahrene technische (Agile) Tester oder Test-Automation-Engineers stehen in der deutschsprachigen Welt die Türen der Unternehmen sperrangelweit offen.

„Der Schlüssel zum Erfolg ist die seriöse Betrachtung der nicht-funktionalen Qualitätsmerkmale gemäß ISO 25010 – also Security, Portabilität, Zuverlässigkeit oder Performance, um nur ein paar zu nennen.“

Im Zuge der Modernisierung sollen selbst Legacy-Anwendungen zum Beispiel auch „Emerging Technologies“ wie KI/Machine Learning, IoT, Drohnen oder „Augmented Reality“ nutzen. Wie können die so entstehenden Softwaresysteme mit bewährten Methoden getestet werden? Oder sind dann in jedem Fall neue Methoden gefragt? Und wenn das der Fall ist: Welche?

Pichler: Das ist natürlich *die* Frage, die uns aktuell alle beschäftigt in der Szene. Welche Methoden eignen sich für KI/ML – und wie können wir die starke Abhängigkeit der KI- und ML-Systeme von den Hardware-Devices beim Testen sinnvollerweise berücksichtigen?

Bei Nagarro, wo ich arbeite, haben wir bereits einige Erfahrungen gesammelt. Meine „Best Practice“: Mit den altbewährten Test-Vorgehensmodellen und -Entwurfstechniken aus dem ISTQB Foundation Level sozusagen „im kleinen Finger“, verstärkt um erfahrungsbasierte-, explorative Testansätze und *viel Geduld und Kreativität* kann man am besten für die erforderliche Flexibilität sorgen. Außerdem werden die teilweise unberechenbaren und überraschenden Besonderheiten der „Things“ und Devices etwas planbarer.

Der Schlüssel zum Erfolg ist die seriöse Betrachtung der nichtfunktionalen Qualitätsmerkmale gemäß ISO 25010, also Security, Portabilität, Zuverlässigkeit oder Performance, um nur ein paar zu nennen – und das sowohl in der konstruktiven als in auch analytischen Qualitätssicherung. Sonst ist der aus Asterix & Obelix bekannte Gallier-Ef-



fekt bei den Piraten zu befürchten: „Sobald Gallier auftauchen, versenken die Piraten ihr Schiff“. Wir mit unseren Apps wären die Piraten mit ihrem vorseilenden Gehorsam – und das Nicht-Berücksichtigen des De-vice-Verhaltens die Gallier.

„Automatisieren ist wie das Gehen über Wasser: Beides funktioniert umso besser, je mehr die Grundlage eingefroren ist.“

Inwieweit lässt sich Softwaretesten automatisieren – und wo bleiben auch in Zukunft die Kreativität, das handwerkliche Geschick und die Intuition des Menschen gefragt?

Pichler: Moderne, erfolgreiche Testautomatisierung funktioniert nicht mehr wie früher nach der Devise: „Wir kaufen ein Tool und legen los“. Angefangen hat es mit intelligenten Testautomations-Frameworks, die die Redundanzen minimieren. Diese wurden mehr und mehr professionalisiert, sind vielseitiger und toolunabhängig geworden.

Nun werden in die Automation mehr und mehr KI-Ansätze integriert, zum Beispiel zur Fehlererkennung oder zur Klassifizierung bei Automatisierungsläufen. Auch hier gilt dasselbe wie beim klassischen Test: Auf dem über Jahre gelernten und weiterentwickelten Tool- und Methoden-Set aufsetzen und „neugierig und kreativ“, also mit PI („People Intelligence“), weiterentwickeln.

Ach ja, eines noch: Automation ist mittlerweile reif geworden, quasi vom Teen (reine Testautomation) zum Erwachsenen (eigenes,

integriertes Projekt im Projekt). Und das heißt auch: Für das Testen sind Vollprofis gefragt.

Wie bewerten Sie Testautomatisierung im agilen Kontext? Eigentlich schreiben Automatismen ja bestimmte Abläufe fest und gelten als Gift für agile Projekte ...

Pichler: Stimmt! Automatisieren ist wie das Gehen über Wasser: Beides funktioniert umso besser, je mehr die Grundlage eingefroren ist. Bei agilen Methoden haben wir inzwischen gelernt, dass ohne Testautomation nichts geht. Allein schon die wachsende Funktionalität bringt ein Team rasch an die Grenzen der Testkapazitäten. Hier gilt es – und das haben die „Mädls und Jungs“ aus der Testing-Community mittlerweile genial hinbekommen –, mit innovativen Ansätzen ein „Gegengift“ zu entwickeln und wirksam gegenzusteuern.

„Für den Testerfolg ist außerdem eine Form des Reporting notwendig, das möglichst kurz und bündig eine Status-Interpretation zulässt, zum Beispiel nach dem KISS-Prinzip ‚Keep It Simple & Straightforward‘ oder wie der berühmt-berühmte Elevator-Talk.“

Sie gelten als Pragmatiker, der Zielorientierung in Projekten höher bewertet als die Konformität der Entwicklungsprozesse zu den grundlegenden Methoden. Wie wägen Sie ab, wie viel Methode zwingend notwendig ist – und wie viel kreativen oder handwerklichen Spielraum die Entwickler und Tester haben sollten?

Pichler: Oh, hat sich das schon herumgesprochen?! Ja, manch einer ist davon über-rascht. Viele erwarten: Jetzt kommt Mr. ISTQB und setzt alles aus dem Lehrplan bis auf das i-Tüpfelchen genau um. Doch dazu bin ich schon zu lange im Geschäft und habe bereits sehr früh die „Brille des Endkunden“ entdeckt, mit der ich primär an Testaufgaben und -projekte herangehe. Hier spielte mir Agil stark „in die Karten“: Mach nur das, was wirklich Wert hat oder stiftet, also wirklich wichtig ist für den Kunden und dessen kritische Geschäftsprozesse. Für den Testerfolg ist außerdem eine Form des Reporting notwendig, das möglichst kurz und bündig eine Status-Interpretation zulässt, zum Beispiel nach dem KISS-Prinzip „Keep It Simple & Straightforward“ oder wie der berühmt-berühmte Elevator-Talk.

Was ist das Wichtigste für den Projekterfolg?

Pichler: Das Wichtigste ist: Jede/r im Team lebt „für das Team“ und fokussiert sich zu 100 Prozent auf das gemeinsame Ziel, dem Kunden so rasch als möglich die Software zu liefern. Für das Management gilt die Devise: Vertraue ins Team! Und für alle: transparent, ehrlich (auch gegenüber dem Auftraggeber!), auf Augenhöhe, ziel- und lösungsorientiert kommunizieren! Daraus leiten sich dann auch die entscheidenden Soft Skills ab: Team-Player, Vertrauen und Kompromissbereitschaft. Wenn alle auf sich selber schauen, hilft das dem Team. Und wie heißt es zu schön: „Geht's dem Team gut, dann geht's auch dem Kunden und dem Projekt gut!“

Herr Pichler, vielen Dank für das Interview!

ZUR PERSON

Ing. Helmut Pichler unterstützt als Tester und Testmanager „aus Leidenschaft“ seit vielen Jahren Kunden und Projekte aus verschiedenen Branchen als Berater und Manager bei den Themen Softwaretest und -qualität. Der Aufbau und die Etablierung des Berufsbildes „Softwaretester“ ist eines seiner Ziele. Ein wesentlicher Bestandteil des Berufsbildes ist für den Co-Autor des Buchs „Agile Testing“ der strukturierte Wissensnachweis, idealerweise in Form international anerkannter Zertifizierungen. Als ehrenamtlicher Präsident des „Austrian Testing Board“ hat Helmut Pichler deshalb dafür gesorgt, dass das ATB die offizielle Vertretung des „International Software Testing Qualifications Board“ (ISTQB) in Österreich ist. Als Präsident des ATB ist er in mehreren internationalen Arbeitsgruppen aktiv, wo er gemeinsam mit anderen Top-Experten den internationalen Teststandard weiterentwickelt. Im Hauptberuf arbeitet Pichler aktuell als QA-Engineer in Emerging-Technology-Projekten und ist verantwortlich für das gesamte Schulungsprogramm bei Nagarro, einem weltweit führenden Unternehmen für digitale Produktentwicklung.



»QUALITÄTSSICHERUNG EINER INDOOR-NAVIGATION«

Jeder, der bereits einmal gezwungen war, sich in einer Großstadt und einem dichten Verkehrsnetz zurechtzufinden, weiß, wie schwer es sein kann, sich zu orientieren. Im Regelfall verlassen sich Reisende auf Wegweiser, Informationsschalter oder Kartenmaterial, offline oder online, um ans Ziel zu gelangen. Und nun stelle man sich vor, man ist blind und hat daher keinen Zugriff auf die üblichen visuellen Navigationshilfen, was dann?!

Zur Lösung dieser Problemstellung haben sich zwei Projektpartner zusammengetan: *Upstream Mobility*, ein Tochterunternehmen der Wiener Linien und Wiener Stadtwerke. Ihre Mission: „Wir stellen den Menschen in den Mittelpunkt und schaffen Mobilität mit Mehrwert. Stets einen Schritt voraus, bauen wir mit zukunftsweisenden Projekten unsere Innovationsführerschaft in Sachen digitaler Mobilität aus.“

Und *Nagarro*, ein weltweit führendes Unternehmen für digitale Produktentwicklung. Das Full-Service-Portfolio umfasst Digital Product Engineering, E-Commerce und Customer Experience Services, Managed Services, ERP-Beratung und Technologie-Dienstleistungen.

Die gemeinsame Vision: Indoor-Navigation über Smart Glasses zur Unterstützung von Blinden und Sehbehinderten sowie sehenden Menschen im ÖV.

Konkret wurde das Wissen beider Unternehmen gebündelt um herausfinden, wie man blinden- und sehbehinderten Menschen, durch die Nutzung von modernster Technologie in der Indoor-Navigation ein Stückchen Freiheit zurückgeben kann – Codewort „Mobi“. Ich bekam die einmalige Chance, als Innovation Quality Manager bei Nagarro Teil dieser Mission zu sein, und möchte praxisnahe Insights aus diesem spannenden Projekt mit Ihnen teilen.

Der Projektstart

Zwei ambitionierte Teams standen in den Startlöchern. Nach einigen Brainstormings zur Aufgabenstellung haben wir mit den Upstream Technology Scouts erste Ideen, Wünsche und Vorstellungen geboren und in eine griffigere und anschaulichere Form gebracht, bevor wir uns mit konkreten Technologien beschäftigten.

Persona1 : Gustav / Blinde Person	Persona2 : Ina / Sehende Person
User Journey – PERSONA Gustav - Blinde Person VLB/2020-06-30 Gustav, von Geburt an blind, trifft sich mit seinem Freund nahe der U-Bahn-Station Wien Landstraße. Er kennt diesen Weg nicht besonders gut, ist nur 2 oder vielleicht 3 Mal dort gewesen. Gustav steht in der U-Bahn in der Nähe des Ausgangs. Er kennt auf jeden Fall die genaue Abfolge der einzelnen U-Bahn-Stationen auf dieser Linie. Bei der U-Bahn-Station Landstraße macht er sich bereit zum Aussteigen und startet per Sprachbefehl die Applikation. ... Gustav gibt die Details an, zu welchem Ausgang er geleitet werden und welche Hilfsmittel, wie Lift oder Rolltreppe, er heute verwenden möchte. Die bevorzugten Hilfsmittel hat er in der Applikation generell hinterlegt, kann die ...	User Journey PERSONA Ina – Sehend VLB/2020-06-30 Ina arbeitet als selbstständige Therapeutin und nutzt für Heimbeseuche von Patienten regelmäßig die U-Bahn. Ina ist sportlich und leidenschaftliche Kaffee-Trinkerin, ihre Lieblings-Kaffee-Kette ist „Kaffee-kuchl“ und sie schaut immer, dass sie sich morgens auf dem Weg zum ersten Patienten einen Kaffee holt. Sobald Ina in der U-Bahn-Station angekommen ist, startet sie die Navigations-App auf ihrer Smart Glass und gibt ihr Navigationsziel (z. B. „Kaffee-kuchl“) mittels Sprache oder Touchpad ein. So kann sie sicher sein, immer auf dem schnellstmöglichen Weg durch die U-Bahn-Station bzw. zu ihrem Kaffee zu kommen ...

Tabelle 1: Zwei Personas

Schritt 1: Recherche

Die Außenwelt ist kartiert. Die Genauigkeit/ Abweichung liegt je nach Tool-/App-Lösung (via GPS) bei rund 10 bis 20 Metern. Bei Navigation in Innenräumen oder sogar im „Untergrund“ (z. B. U-Bahn-Stationen) liegt die Genauigkeit sogar darunter. Theoretisch haben sich bereits Indoor-Positioning-Systeme (IPS) etabliert, die Abweichungen von weniger als 2 cm Genauigkeit versprechen.

Wir wollten diese Theorien in der Praxis prüfen! Das war der Ausgangspunkt unseres Projekts. Unsere initiale Komponenten-Recherche beinhaltete:

- Selektion eines *PDR-Algorithmus* (Pedestrian Dead Reckoning).
- Die Identifikation von passenden *Wegfindungs-, sowie Positionsfindungs-Algorithmen*:

men: BLE-Beacon-Positionierung, Magnetfeldmethodik sowie Sensor-basierte Positionierung.

Die für uns erfolgversprechendste Lösung: Eine Kombination aus *Smart Glass*, *Smartphone & BLE Beacons* – eine integrierte Datenbrillen-Lösung mit Sprachausgabe, Suche und Navigation zum Point-of-Interest mithilfe von Echtzeit-Standorterkennung über Bluetooth Beacons und dynamische Objekterkennung.

Schritt 2: Eine erste Standortbestimmung

Die erste Recherche bot schon einen guten Einblick, was wir am Ende in Händen halten sollten. Aber wie man weiß, liegt der Teufel im Detail. Da die Entwicklung einer solchen Lösung für uns alle „Neuland“ war, setzten wir auf ein agiles Vorgehen bei der Produktentwicklung.

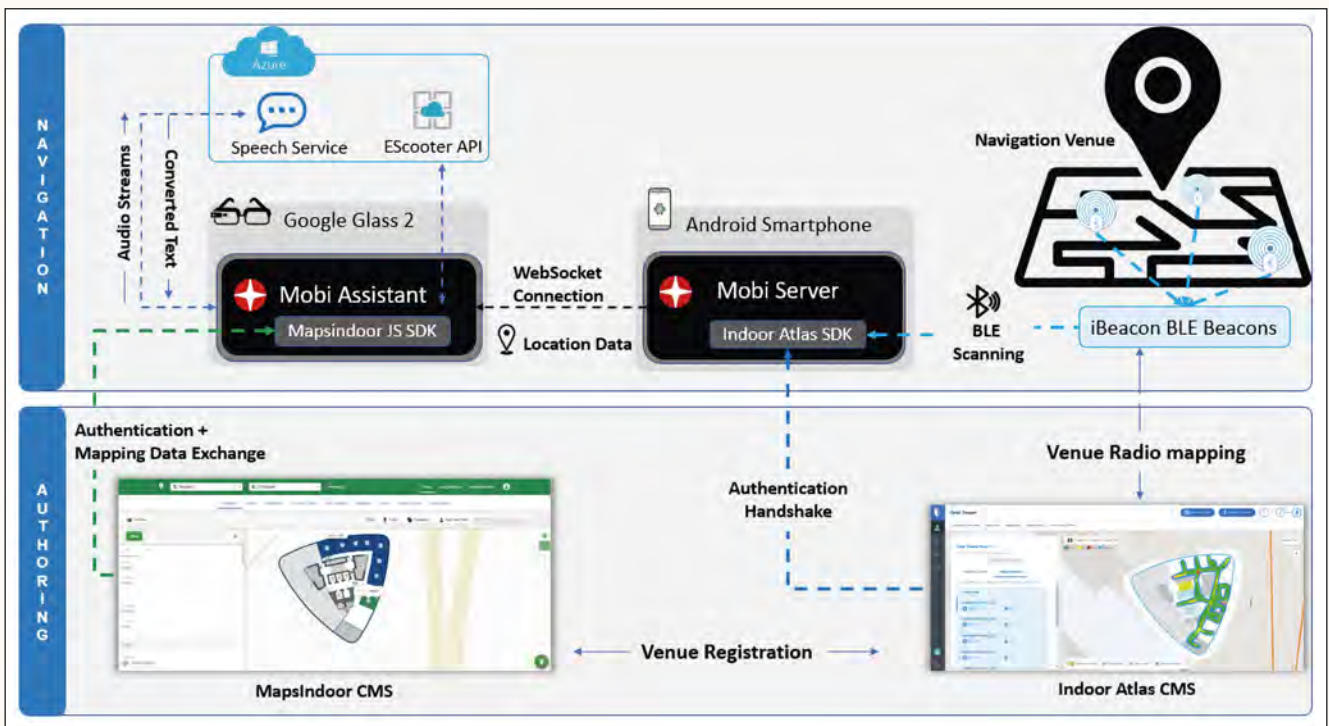


Abb. 1: Systemarchitektur Mobi

Neben einer technischen Evaluierung galt es aber vor allem, gewünschte Funktionalitäten für unsere Zielgruppe zu konkretisieren. Welche Bedürfnisse, Wünsche und vor allem reale Hilfen könnten hier wirklich Erleichterung bringen? *Unser Ansatz:* Wir definierten Personas inklusive User-Journeys.

Schritt 3: Verfeinerung der gewünschten Funktionalität

In weiteren Workshops wurden, aus Sicht der beiden Personas „Gustav“ und „Ina“ (siehe Tabelle 1), die skizzierten Features auf konkrete Machbarkeit und Nutzen geprüft, konkretisiert und in Epics und weiter in User-Stories (US) heruntergebrochen:

- › Neben den Navigationselementen wurden KI-unterstützte Bilderkennungs-Features eingebracht, z. B. zur Erkennung temporärer Warnhinweise, wie „Vorsicht Rutschgefahr-“ oder „Außer Betrieb“-Schilder.
- › Wir ergänzten Epics & US mit Akzeptanzkriterien (AK). Das Kern-Planungsteam fungierte als erweiterter Product Owner, welche Akzeptanz & Priorität abstimmten.
- › Zu den AK definierten wir entsprechende Testfälle, denn was wäre die beste App ohne vernünftige Qualität?

- › Weiters setzten wir „Prüfen auf Automatisierbarkeit der Testfälle“ auf unsere To-do-Liste.

Schritt 4: Architektur

Aus technischer Sicht sind die aktuelle Verfügbarkeit, verlässliche Partner sowie Technologien wesentliche Erfolgsfaktoren. Daraus leitet sich die in **Abbildung 1** skizzierte Systemarchitektur für uns ab.

Aufgrund der Einschränkungen (Corona-Lockdown & Co.) mussten während der Test-

phase drei Testlocations aufgesetzt werden – zwei physische Testlocations in den Büros von Nagarro und Upstream in Wien sowie eine virtuelle Testlocation für unsere Nagarro-Projektkollegen in Indien –, bevor wir schlussendlich den Einsatz in einer realen U-Bahn-Station prüfen konnten. Dadurch gewannen wir wichtige Erkenntnisse.

Aus früheren Projekten wissen wir, wie wichtig eine klare Menü-Struktur, welche den Nutzer sowohl akustisch (Sprachbefehle) als auch visuell (Ansicht in Smart Glass) durch die Anwendung leitet, für die Nutzer-Ak-

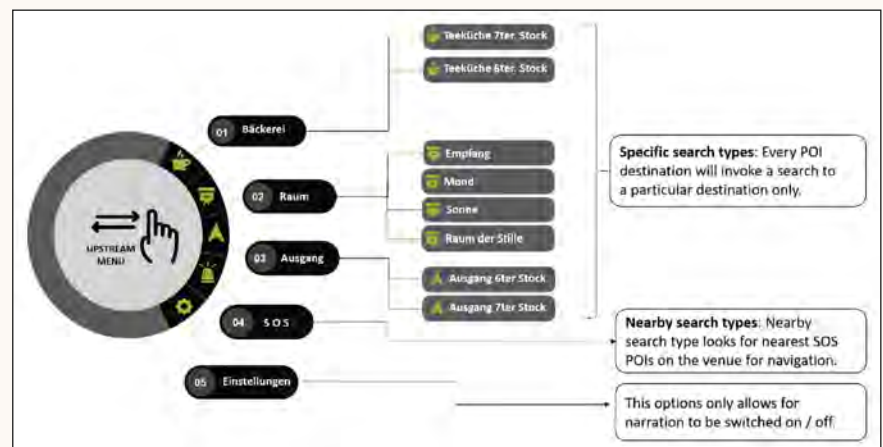


Abb. 2: Menüstruktur Mobi (Testsystem)

zeptanz ist. Darum legten wir ein Hauptaugenmerk darauf. Unsere Menüstruktur zeigt **Abbildung 2**.

Die Implementierung

Neben dem Aufbau der Architektur startete unser Entwicklungsteam mit Hochdruck die Implementierung der Basis-Funktionalität und die Integration der System-Komponenten:

- › **Digitalisierung der Testlocation:** Digitalisierung der Location-Grundrisse, Definition der möglichen Verbindungspfade sowie aller verwendeten Kategorien & Typen (Ausgänge, Sitzbänke, Aufzüge, Rolltreppen usw., siehe **Abbildung 3**).
- › **Beacon Coverage:** Mittels der App eines weiteren Partners wurde die Abdeckung der montierten Beacons aufgezeichnet – eine der wesentlichsten Voraussetzungen für eine funktionstüchtige Navigation (**siehe Abbildung 4**).
- › **Smart Glass Set-up:** Das Aufsetzen/Aktualisieren der Google Glass bedurfte mehr Geduld und Kreativität, als ursprünglich angenommen, letztlich waren wir aber erfolgreich (**siehe Abbildung 5**).

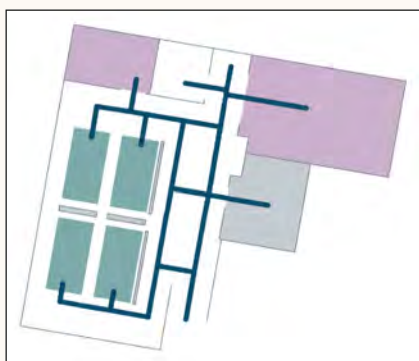


Abb. 3: Digitalisierte Testlocation 1



Abb. 4: Beacon Coverage

Der Testbetrieb

„OK Mobi“, unser „Wake Up“-Kommando für die Lösung, wurde im Testbetrieb unser Alltag. In zahlreichen Testläufen wurden Routen, Screen-Anzeigen und Sprachausgaben verifiziert, korrigiert und optimiert, bis uns „OK Mobi“ im Traum erschien und wir uns mit „OK Mobi“ begrüßten.

Durch die effiziente Zusammenarbeit mit unserem außerordentlich motivierten Entwicklungsteam konnten initiale Fehler stets rasch behoben werden. Neben ersten Erfolgen gab es jedoch auch einige Herausforderungen, die uns einiges an Kopfzerbrechen bereiteten. Ein Dauerbrenner war die akkurate Lokalisierung der aktuellen Position der Testperson („Blue Dot“) auf dem Weg von A nach B. Wir investierten viele Teststunden, um das Zusammenspiel von berechneter Navigation und Identifikation der aktuellen Position zu verfeinern.

Die aus den Akzeptanzkriterien abgeleiteten Testfälle, welche sich auf die Menü-Abfolge bezogen, konnten rasch erfolgreich abgewickelt werden. Dabei bewährte sich unser agiler Ansatz, durch welchen wir im PD-CA-Zyklus aus Plan – Do – Check – Act die Screen-Inhalte und -Abläufe laufend verbessern und optimieren konnten.

Routen-Kommandos, welche uns zum gewünschten Ziel führen sollten, hielten häufig Überraschungen für uns bereit. Manchmal führte uns Mobi sehr präzise zum gewünschten Ziel, beim „Regressionstestlauf“ blieben wir allerdings „auf der Strecke“ liegen. Nach vielen, vielen (Test-) Stunden kamen wir zur Erkenntnis, dass die auf der Website des Po-

sitionsbestimmungs-Tool-Herstellers kolportierten 1 bis 3 Meter Abweichung bei der Positionserkennung wohl eher ein theoretischer Marketingspruch war als erprobte Praxis.

Neu war für uns auch, dass die Größe der Location einen wesentlichen Einfluss auf die Präzision der gelieferten (Beacon-)Signale hatte, welche die Basis für die Positionsbeziehung bildeten. Konkret: Bei kleinen Locations (ca. 100 m²) ist hier mit Abweichungen von 5 m und mehr zu kalkulieren. Bis zuletzt teilten wir an den Nachjustierungen und setzten auf Aussagen des Herstellers, wonach die Lösung in größeren Räumlichkeiten deutlich bessere Ergebnisse liefern sollte.

Mit dieser Hoffnung im Gepäck starteten wir den Vorbereitungstest für den UAT im Upstream Office (ca. 1.000 m²).

Bereits erste Testläufe zeigten deutlich verbesserte Navigationsergebnisse, was unsere Hoffnungen – zum Glück – bestätigte. Mit ein Erfolgsfaktor war die konstruktive und flexible Art des Upstream-Projektleiters und seines Teams, durch deren Engagement und Erfahrung auch noch kleinere technische Hürden erfolgreich genommen wurden. So stand der Abnahme am Ende dieser R&D Projekt-Phase (UAT) nichts im Wege.

Unseren Plan, hier auch gleich mit automatisierten Tests eine Basis für die nachhaltige Qualitätssicherung zu schaffen, mussten wir aufgrund der Volatilität der Ergebnisse, mehr und mehr reduzieren. Aktuell beließen wir es bei der konzeptionellen Planung und konzentrierten uns auf die Erfüllung der zugesagten Grundfunktionalität.

Was wir gelernt haben

Dieses Projekt zeigt, dass heute vieles bereits machbar ist, allerdings noch einige Schritte zu einer veritablen Produktreife erforderlich sind. Ein Auszug aus meinem Test-Stammbuch:

Hohe Flexibilität und Kreativität ist Grundvoraussetzung bei Emerging Technology (R&D)-Projekten, da die eingebundene Hardware sehr volatile Ergebnisse verursacht/liefert.

Grobe statt detaillierte Testfall-Beschreibungen sind unabdingbar, wenn Testfälle mit vernünftigem Aufwand aktuell gehalten werden sollen.



Abb. 5: „Gustav“ in Action

Agile in Reinkultur: „Inspect & Adapt“ gilt hier in allen Bereichen – bei der Definition der Epics, US, AK, bis hin zu den Testfällen.

Testautomatisierung setzt stabile, ident ablaufende Abläufe voraus, um hier zielführend auf- und umgesetzt werden zu können:

- › In *Basisfunktionen*, wo stabil laufende Abläufe sichergestellt sind, wie beim Starten und Auswählen der Basissettings (Blind/ Sehend Modus, Aufzug/Rolltreppe/Stiegen-Präferenz), grundsätzlich denk- und machbar, zumal das auch via Simulatoren realistisch erfolgen kann.
- › Auch in *Routen-Funktionen* sind Tests automatisierbar, allerdings ausschließlich, solange die Ergebnisse stabil via Simulatoren geliefert werden.
- › Zur *Automatisierung realer Teststellungen*, also wenn „echte Devices“ im Live-Einsatz verwendet werden, sind wir aktuell auf-

grund der volatilen Ergebnisse noch weit davon entfernt.

Support: Dank der aktiven Mitarbeit des Upstream-Teams sowie der unbürokratischen Unterstützung seitens Wiener Linien (U-Bahnbetreiber) konnten wir so manche Hürde rasch überwinden.

Live-Einsatz in Vorbereitung

War da nicht die Rede von einer U-Bahn-Station? Richtig! Der letzte Schritt unseres Pro-

jekts, das Installieren unserer Lösung in einer ersten U-Bahn-Station in Wien, ist aktuell in Vorbereitung und wir können es kaum erwarten, „Mobi“ in der realen Welt zu erleben, wenn es zahlreichen Blinden und sehbehinderten Menschen ein Stück Freiheit wiedergibt, und auch für sehende Menschen einen Mehrwert bringt (Stichwort „hands-free“).

Bis dahin werden wir noch viele weitere wichtige Erfahrungen gewinnen, die für weitere Innovationsprojekte dieser Art sehr wertvoll sein werden. The future is bright!



Helmut Pichler

helmut.pichler@nagarro.com

unterstützt seit vielen Jahren Kunden und Projekte aus verschiedenen Branchen bei den Themen Softwaretest und -qualität. Der Aufbau und die Etablierung des Berufsbildes „Softwaretester“ ist eines seiner Ziele. Als ehrenamtlicher Präsident des „Austrian Testing Board“ hat Helmut Pichler dafür gesorgt, dass das ATB die offizielle Vertretung des „International Software Testing Qualifications Board“ (ISTQB) in Österreich ist. Im Hauptberuf arbeitet er als QA-Engineer in Emerging-Technology-Projekten und ist verantwortlich für das gesamte Schulungsprogramm bei Nagarro.

SAP, ASAP.

Tricentis ist jetzt die empfohlene Testlösung von SAP

10X

schnelleres
Testen

90%

Risiko-
minderung

50%

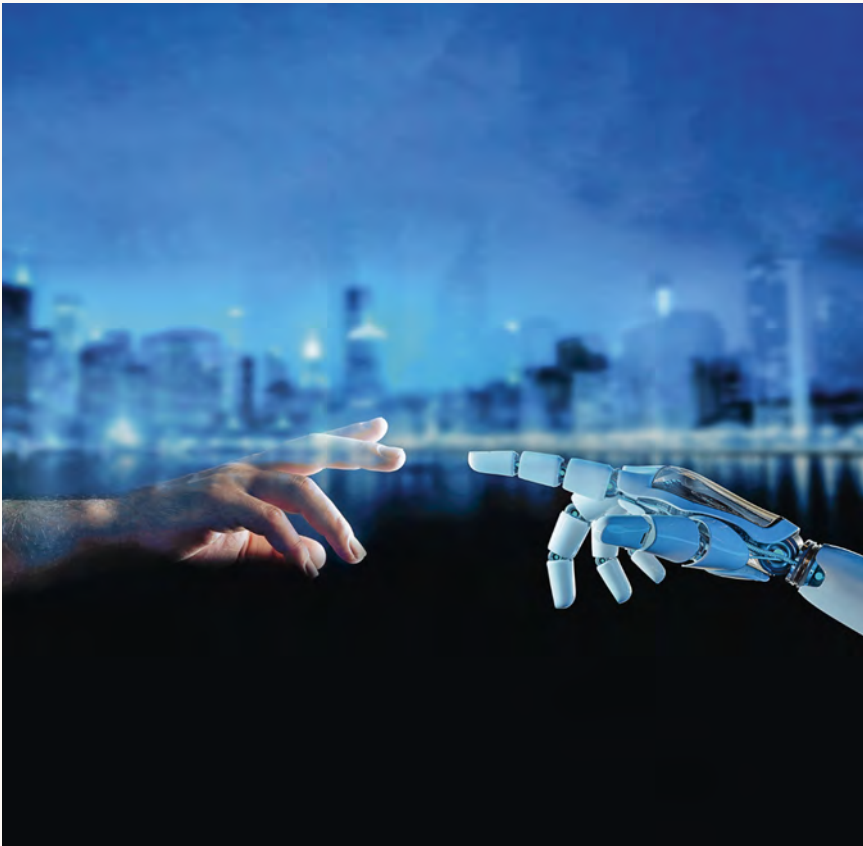
niedrigere
Kosten

tricentis.com/de/sap

 **Tricentis**

»VON DER FORSCHUNG IN DIE REALITÄT«

Gefördert von der Österreichischen Forschungsförderungsgesellschaft (FFG) hat ein Team von Automatisierungs- und KI-Experten an der Vision eines hochautomatisierten Test Life Cycle geforscht.



der Tests 10 Stunden. Somit weiß Bernd erst am nächsten Tag, ob alles fehlerfrei funktioniert. Am darauffolgenden Morgen stellt er fest, dass einige der Tests über Nacht fehlgeschlagen sind. Die Analyse gestaltet sich kompliziert und dauert diesmal den ganzen Vormittag. Das Ergebnis ist eine bisher nicht entdeckte Fehlfunktion der Legacy-Backend-Komponente.

Warum ist das Problem ausgerechnet im Legacy-Code?

Das Problem betrifft jene Legacy-Komponente, die kaum dokumentiert und getestet ist, und deren Entwickler natürlich längst nicht mehr verfügbar sind. Man hätte die Komponente schon lange optimieren müssen. Dies wird nun eine sehr langwierige Aufgabe.

Die geschilderte Situation ist für die meisten Softwareentwickler nicht bloß graue Theorie. Viele Prozesse rund um die Testautomatisierung sind immer noch manuell und aufwendig (vgl. **Abbildung 1**). Ziel ist es, möglichst viele wiederkehrende Aufgaben im Test Life Cycle zu automatisieren.

Im Softwareengineering ist die automatisierte Testdurchführung inzwischen fester Bestandteil der Entwicklungsprozesse. Viele Firmen integrieren automatisierte Tests bereits in ihren Entwicklungspipelines. Dadurch fallen Fehler schneller auf und finden im besten Fall gar nicht erst den Weg ins gemeinsame Code-Repository. Aus Sicht der Qualitätssicherung könnte man meinen, damit ein zufriedenstellendes Qualitätsniveau erreicht zu haben. Die Realität sieht leider doch oft anders aus.

Aus der täglichen Praxis

Da aktuell häufig nur die Testdurchführung automatisiert ist, sind immer noch zahlreiche Herausforderungen offen, wie folgendes Beispiel-Szenario veranschaulicht:

Tom hat die neuen Designs an unserem Log-in für unsere Webanwendung umgesetzt. Nach

seinem Commit schlagen die Check-in-Tests fehl. Da Tom den Task vor Sprintende noch abschließen möchte, akzeptiert er den Umstand und checkt den Code ein. Am nächsten Morgen stellt Bernd fest, dass im Nightly Testrun ca. 40 Testfälle nicht erfolgreich durchgelaufen sind.

Mal eben schnell die Log-Dateien analysieren ...

Nach zwei Stunden zeitraubender Analyse der Log-Dateien, um den wirklichen Fehler herauszufinden, glaubt Bernd, alle relevanten Testfälle aktualisiert zu haben. Um sicherzugehen, startet er die Regressions- und Systemtests.

Nun heißt es, auf den Testdurchlauf zu warten ...

Da bereits ein sehr umfangreiches Testset implementiert wurde, benötigt der Durchlauf

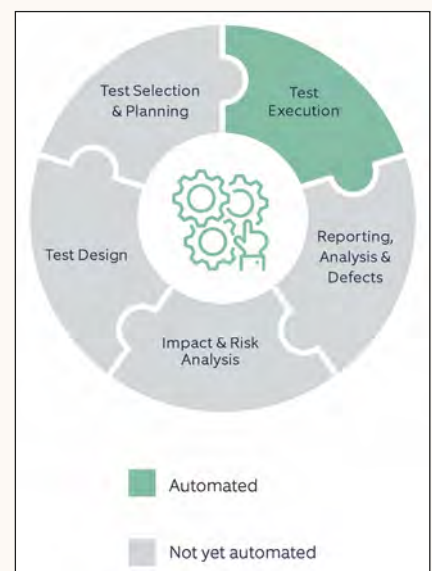


Abb. 1: Die automatische Testdurchführung ist lediglich ein kleiner Bestandteil im Test Life Cycle, die verbleibenden Schritte sind meist noch manuell und wiederkehrende Aufgaben

Die Vision: Ein hochautomatisierter Test Life Cycle

Gefördert von der Forschungsförderungsgesellschaft Österreich (FFG) hat das Technologieunternehmen Nagarro, gemeinsam mit der Fachhochschule Technikum Wien, Ende 2019 ein Forschungsprojekt gestartet: *AI4T – Advanced Intelligence for Testing* (siehe **Kasten 1**).

Im Folgenden stellen wir die Ergebnisse aus dem ersten Forschungsjahr vor.

Smart Test Selection

Mit wachsender Komplexität von Anwendungen und Systemen steigen die zu durchlaufenden automatisierten Testfälle. Ein vollständiger Testdurchlauf des kompletten Testsets benötigt meist einige Stunden. Diese Tests bei jeder kleinen Änderung auszuführen, ist sehr zeitaufwendig, selbst wenn deren Durchführung automatisiert ist. Dennoch möchte man nach jedem Commit

möglichst umfangreich testen. Viele Unternehmen verwenden dafür eine fixe Teilmenge der Tests als Smoke-Tests.

Smart Test Selection beschäftigt sich mit der Frage, wie solch eine Teilmenge dynamisch ausgewählt werden kann. Das Ziel ist, die Gesamtlaufzeit zu reduzieren und trotzdem möglichst alle relevanten Fehler zu finden.

Der entwickelte Prototyp greift auf erfasste Daten aus vergangenen Testdurchläufen zu,

um relevante Testfälle für die Ausführung vorzuschlagen. Hierbei wird iterativ der relevanteste, noch verbleibende Test ausgewählt und durchgeführt, bis eine konfigurierbare Durchlaufzeit erreicht wurde. Testfälle werden häufiger ausgewählt, wenn sie:

- › häufig fehlschlagen,
- › lange nicht mehr ausgeführt wurden und
- › eine kurze Laufzeit haben.

Durch den Einsatz von Machine Learning und Artificial Intelligence will man der Vision eines hochautomatisierten Test Life Cycles ein Stück näher kommen.

Das von der Forschungsförderungsgesellschaft Österreich geförderte Forschungsprojekt AI4T ist für insgesamt drei Jahre ausgelegt (bis 2022). Dieser Artikel präsentiert die Ergebnisse des ersten Forschungsjahres: vier Prototypen, basierend auf anonymisierten Testdaten aus praxisnahen Szenarien bei Nagarro-Kunden. Manche davon sind bereits im Echt-Einsatz implementiert und bewähren sich.

Mehr zum Forschungsprojekt AI4T finden Sie auf www.nagarro.com/AI4T

Kasten 1: AI4T – Advanced Intelligence for Testing



AI4T – Advanced Intelligence for Testing

Das Ziel von AI4T ist ein hochautomatisierter Test-Lifecycle. Durch den Einsatz von Künstlicher Intelligenz und Machine Learning wird der Testaufwand drastisch reduziert und QA-Prozesse werden schneller, skalierbarer und umfassender.

Finden Sie heraus, wie AI4T Ihnen und Ihrem Team einen klaren Vorsprung verschaffen kann!

www.nagarro.com/AI4T



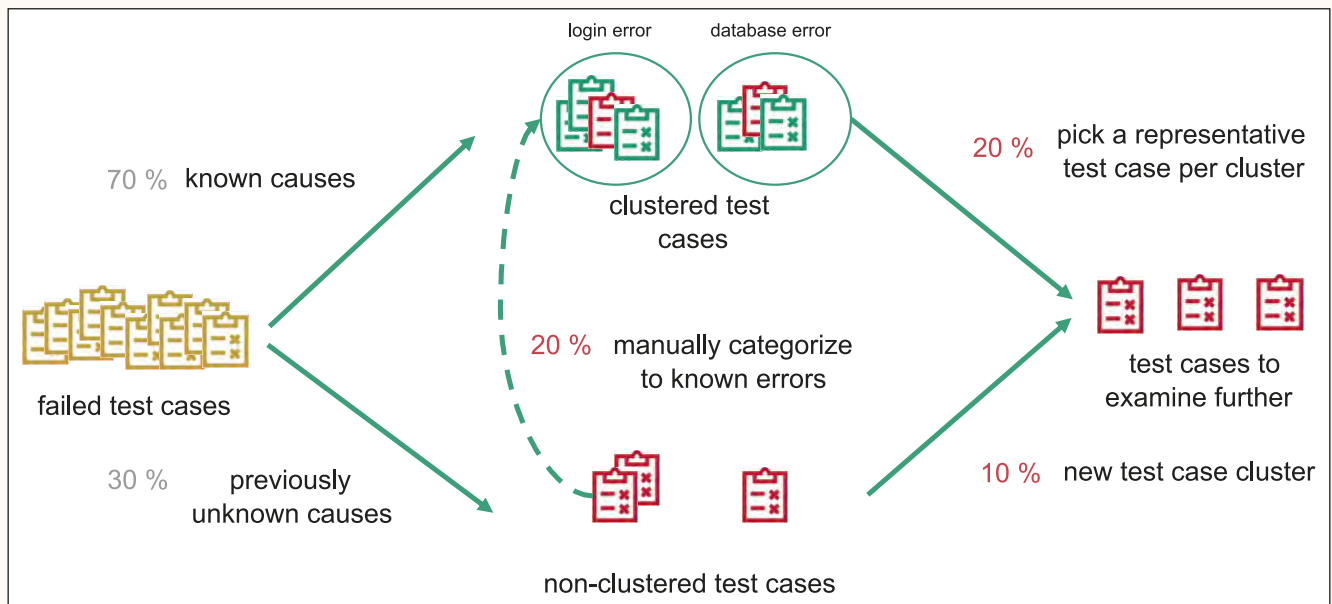


Abb. 2: Intelligent Failure Analysis Process: Aufgrund von intelligentem Clustering müssen ca. 50 Prozent weniger Testfälle untersucht werden

Nach Feedback von ersten Anwendern wird diese Komponente aktuell erweitert. Mit Hilfe von KI-basierten Algorithmen werden die Code-zu-Testfall-Verknüpfungen errechnet, auf deren Basis die Testfälle, welche für gerade geänderte Codefragmente relevant sind, wahrscheinlicher ausgewählt werden.

Self-Healing Test Automation

Ein vielfach unterschätztes Problem ist der Wartungs- und Pflegeaufwand, den automatisierte Testfälle mit sich bringen. So führt beispielsweise eine Umbenennung eines Buttons der GUI schnell zu einem scheinbaren Fehler. Erst im Nachhinein stellt sich dann heraus, dass es sich um einen nicht mehr zum Code passenden Testfall gehandelt hat.

Bei Self-Healing Test Automation beschäftigt man sich mit der Fragestellung, wie Testfälle bei Änderungen im Quellcode (semi-) automatisch angepasst werden können, um Wartung und Pflege der Testfälle zu automatisieren und den Aufwand der Anpassung der Testfälle zu reduzieren.

Der entwickelte Prototyp erkennt mithilfe von KI-basierten Algorithmen Änderungen in UI-Komponenten. Automatisierte Testfälle, die wegen dieser Änderung fehlschlagen, werden voll-automatisch korrigiert. Dazu werden Graph-Matching-Techniken angewendet, um Abweichungen der alten und neuen HTML-Struktur zu vergleichen und umbenannte Komponenten neu zuordnen zu können.

Der Prototyp überwacht den Testdurchlauf. Wenn ein Test fehlschlägt, wird analysiert, ob dies aufgrund eines fehlenden UI Identifier geschehen ist. Wenn ja, wird dieser korrigiert, die Korrektur dem Entwickler gemeldet und der Test läuft erfolgreich durch – alles automatisiert.

Im weiteren Verlauf des Forschungsprojekts werden wir diese Komponente auf andere Bereiche, wie zum Beispiel API-Management, erweitern. Außerdem wollen wir bei Änderungen, bei denen sich das System unsicher ist, ein semi-automatisches Feedback des Entwicklers erfragen.

Intelligent Fail Analysis

Große Softwareprojekte führen große Mengen an Testfällen aus. Oftmals ist man als Entwickler in der Situation, dass man schon morgens mit zahlreichen fehlgeschlagenen Testfällen konfrontiert ist. Die langwierige und zeitraubende Suche nach der eigentlichen Fehlerursache beginnt.

Die Intelligent Fail Analysis beschäftigt sich mit Methoden, um aus Log-Einträgen den wahren Fehler deutlich schneller zu extrahieren. Hierfür stehen zahlreiche Algorithmen und Verfahren im Bereich Machine Learning (ML) zur Verfügung.

Im ersten Jahr wurde ein auf Clustering basierender Ansatz entwickelt, welcher die fehlgeschlagenen Log-Dateien kategorisiert.

Abbildung 2 veranschaulicht die Funktionsweise unseres Prototyps. Wenn die Zuordnung eines Testfalls in ein Cluster durch das ML-Modell gelingt, muss von diesem Cluster nur noch ein repräsentativer Testfall näher inspiziert werden. In den seltenen Fällen, in denen das ML-Modell den Fehler keinem Cluster zuordnen kann, muss man diesen gesondert betrachten. Durch Nutzerfeedback verbessert sich das Modell kontinuierlich, sodass dieser Fall immer seltener auftreten wird.

Zusätzlich entstand im ersten Forschungsjahr das Framework *Sqeed*, welches Testdurchläufe zusammenfasst und übersichtlich visualisiert. Dieses Framework stellt die Testdurchläufe übersichtlich dar und aggregiert automatisch Informationen aus verschiedensten Log-Dateien in einer Quelle. **Abbildung 3** zeigt die Übersichtsansicht von *Sqeed* von Testfällen. Das Framework kann sowohl in zahlreiche Test-Frameworks integriert werden als auch direkt auf den Log-Dateien arbeiten. Aufgrund der einfachen Installation und stetig wachsenden Funktionalitäten, wie einem Coverage Graph, einer automatischen Testdurchführung via UI und vielem mehr, ist *Sqeed* bereits bei Kunden im Einsatz und wird dort laufend erweitert.

Smart Refactoring

Von der ersten Idee hin zu einer Anwendung oder einem System entstehen nicht selten Prototypen und Provisorien die, als Opfer ihres eigenen Erfolgs, ungewollt ihren Weg in

	2020-11-23 15:05:32.000 web-40.12 21	2020-11-23 15:02:56.000 web-40.12 156	2020-11-23 15:00:05.000 web-40.12 174	2020-11-23 14:58:03.000 web-40.12 210
I can manually add a food to the diary [Diary] 100	3.77 s	3.96 s4.03 s3.07 s	3.76 s3.90 s2.92 s	4.68 s3.60 s2.97 s
I can add a food entry to my diary [Food] 54		33.29 s5.13 s34.00 s	6.97 s5.19 s16.76 s2	8.04 s7.49 s5.21 s
I can make a quick search for foods [Food] 90		4.63 s4.13 s3.53 s	4.55 s4.55 s4.07 s	4.69 s4.84 s4.03 s
I can open the food details view [Food] 90		3.35 s4.14 s3.33 s	1.13 s0.91 s0.43 s	3.48 s4.01 s3.37 s
I can search for foods [Food] 90		2.79 s2.84 s2.10 s	2.87 s2.30 s2.12 s	2.77 s2.83 s2.07 s
As a user, I can login [Login] 46	1 10.37 s	30.23 s31.56 s2.03 s	1 8.82 s 1 7.46 s 1 6.67 s	4.38 s2.74 s3.23 s
With invalid credentials, I cannot log in [Login] 91	4.28 s	33.14 s4.74 s3.75 s	6.15 s4.47 s3.61 s	4.29 s4.58 s6.17 s

Abb. 3: Darstellung der Testdurchläufe (Spalten) und Testfälle (Zeilen) sowie der einzelnen ausgeführten Tests (Zellen) in Sspeed (<https://demo.sspeed.nagarro.com/>)

das fertige Produkt finden. Der dabei entstandene unsaubere Code wird in Folge als „technische Schuld“ durch die Jahre mitgetragen.

Smart Refactoring verändert Softwarecode so, dass die externe Funktionsweise gleich bleibt, sich aber die interne Struktur verbessert. Durch diese Verbesserung der internen Struktur wird die Codequalität verbessert und somit die Wartbarkeit des Codes erhöht.

Im ersten Forschungsjahr entstand eine Smart-Refactoring-Komponente, welche Patterns automatisch erkennt und Refactorings durchführt. Häufig müssen hierfür neue Me-

thoden generiert werden. Für das Finden der neuen Methodennamen wird Machine Learning eingesetzt.

Es ist geplant, ML-Algorithmen anhand real durchgeführter Refactorings in zahlreichen Open-Source-Projekten lernen zu lassen, wann man ein solches Refactoring-Pattern durchführen sollte und wann nicht.

Erste Erfolge und vielversprechende Aussichten

Die Ergebnisse des Forschungsprojekts zeigen bereits nach dem ersten Jahr, dass

viele Aspekte um den Test Life Cycle automatisiert werden können. Wir werden den Menschen beim Testen niemals ersetzen können und dies ist auch gar nicht das Ziel von AI4T. Aber es zeigt sich, dass durch den Einsatz von intelligenten Technologien viele von den ständig wiederkehrenden Aufgaben (teil-)automatisiert werden können. So bleibt mehr Zeit, um sich auf die komplexeren Fragestellungen zu fokussieren, die mehr Spaß machen!



Dr. Jan Nößner

jan.noessner@nagarro.com

entwickelt bei Nagarro Systeme im Bereich KI und Machine Learning (ML). Sein theoretisches Wissen über ML und Datenintegration erwarb er während seiner Promotion an der Universität Mannheim, dem größten Institut in Deutschland für Data und Web Science. Er veröffentlichte zahlreiche Artikel auf hochkarätigen Konferenzen und gewann mehrere Best-Paper-Awards. Dr. Nößner leitete die Implementierung von ML in Produktsysteme für verschiedenste Unternehmen. Er entwickelte innovative und neue Datenmodellierungsstrategien und Big-Data-Lösungen mit SQL-, No-SQL- und Graph-Datenbanken.

»FÜR TEST AUTOMATION ENGINEERS MIT PRÄDIKAT«

Das Buch „Basiswissen Testautomatisierung“ schafft es, mehr zu sein als „nur“ ein Buch zur Aus- und Weiterbildung.

Warum ich dieser Meinung bin? Ich vergleiche es gern mit einem anderen technischen und künstlerischen Bereich, der Fotografie. Werden hier in einem Buch, Artikel oder Ähnliches die wichtigsten Aspekte wie Blende, Belichtungszeit, ISO und gestalterische Aspekte wie der Goldene Schnitt behandelt, heißt das noch lange nicht, dass man fotografieren kann. Es braucht Übung und Erfahrung, um das eigene Praxiswissen vertiefen zu können. Eine offene Blende für viel Licht und der beliebte Bokeh-Effekt klingt erst mal gut. Trotzdem ist es eine schlechte Idee, diese bei praller Mittagssonne ohne Filter für eine Langzeitbelichtung zu verwenden. Hier wäre es klasse, eine Abkürzung durch einen Tipp von erfahrenen Leuten aus der Praxis zu bekommen.

Genau dies geschieht in diesem Buch durch die über das ganze Buch verteilten „Tipps aus der Praxis“. Etwas weg von der reinen Theorie schaffen es die Autoren hier, den eben gelesenen Part einzuordnen, hilfreiche Tipps zu geben und auch durchaus kritisch mit der Portierbarkeit der Theorie in die Praxis umzugehen.

„Basiswissen Testautomatisierung“ vereint das Standard-Wording zum Thema Testautomatisierung, die Wissensvermittlung zur Vorbereitung auf die Prüfung und sinnvolle



Manfred Baumgartner, Stefan Gwihs, Richard Seidl, Thomas Steirer, Marc-Florian Wendland

**Basiswissen Testautomatisierung.
Aus- und Weiterbildung zum ISTQB®
Advanced Level Specialist –
Certified Test Automation Engineer**

Seiten: 398

Verlag: dpunkt.verlag,

3., aktualisierte und überarbeitete Auflage, 2021

ISBN Print: 978-3-86490-675-6

Sprache: Deutsch

Tipps aus der Praxis. Dies schaffen die Autoren, ohne dabei angestaubt zu wirken. Trotz des Umfangs des Buchs verstehen sie es, sinnvolle Grenzen zu ziehen. Oft wollen „Experten“ in Artikeln, Magazinen oder Büchern keine Lücke lassen und einen Themenbereich möglichst allumfänglich abdecken. Dadurch lassen sie sich zu Aussagen hinreißen, die schwer zu beurteilen sind oder einfach aufgrund ihrer Natur zu schnelllebig sind. Nicht so in diesem Buch. Geht es zum Beispiel um konkrete Tools im Bereich der Testautomatisierung, so lassen sich die Autoren nicht zu Empfehlungen hinreißen, die quasi sofort nach dem Niederschreiben schon wieder veraltet wären. Vereinzelt werden für Be-

reiche jedoch Vertreter genannt. Dies hätte für meinen Geschmack noch etwas häufiger vorkommen können, obwohl die „Platzhirsche“ in ihren jeweiligen Disziplinen ohnehin bekannt sein dürften.

Wir haben im Team mittlerweile den ersten zertifizierten TAE (Test Automation Engineer) und ich freue mich, den anderen Teammitgliedern ab jetzt dieses Buch ans Herz legen zu können. So werden sie gut auf die Prüfung vorbereitet und bekommen einen umfangreichen Einblick in das Thema. Diese Kombination stellt sowohl einen Mehrwert für das jeweilige Teammitglied als auch für das gesamte Team und die Firma dar.

ISTQB-Box

Das ISTQB bietet seit 2016 einen Kurs zum „Certified Tester Advanced Level Test Automation Engineer“ an, der seit September 2019 vom German Testing Board (GTB) für den deutschen Markt lokalisiert vorliegt. Der Lehrplan richtet sich vor allem an erfahrene Tester, die sich im Bereich der Testautomatisierung weiterentwickeln wollen. Er umfasst folgende Inhalte:

- › Einführung und Ziele der Testautomatisierung,
- › Vorbereitungen für die Testautomatisierung,
- › die generische Testautomatisierungsarchitektur,
- › Risiken und Eventualitäten bei der Softwareverteilung,
- › Berichte und Metriken bei der Testautomatisierung,
- › Überführung des manuellen Testens in eine automatisierte Umgebung und
- › fortlaufende Optimierung.



Jan Giesen

(j.giesen@gtb.de)

ist Leiter der

GTB-Arbeitsgruppe
„Test Automation Engineer“.



FÜR SIE GELESEN VON ...

Florian Spohn

(florian.spohn@gmail.com)

Teamleiter für Testautomatisierung
bei der EUROIMMUN AG.

