

D Appendix of the Bachelor's Thesis „Development and Evaluation of a Standardized SAP S/4HANA Performance Test Using Browser-based Multi-User Simulation” by Laurence Kamarasy

Transcription of the Audio Recording of the Focus Group on Evaluation of the created Performance Tests for SAP S/4HANA:

Host: Kleiner Verzögerung, herzlich willkommen und danke fürs Erscheinen zur zweiten Fokusgruppe für meine Bachelorarbeit. Diesmal geht es um die Evaluation der ersten DSR-Iteration. Dafür würde ich euch dann einmal abholen, worum geht es eigentlich nochmal, was man beim letzten Mal alles festgestellt hat. Das Ziel der Thesis ist das Erstellen eines Tests bzw. Prozesses, Frameworks für die Vergleiche von Systemen. Methodik ist Design Science Research. Es geht eben darum, erst einmal nur den Test zu generieren, um wenn sich etwas ändert, sei es Container statt Virtual Machines oder Power 27 Racks oder neue Virtualisierungstechnologie, dass man praktisch schon einen funktionierenden Test hat, den man jetzt einfach nur noch auf dem bestehenden System anwendet und auf dem neuen System anwendet, um dann direkt vergleichen zu können. In der letzten Focus Group ging es um Metrik und Requirements, um diesen Test zu erstellen bzw. worauf man achten muss bei der Testerstellung und das Ergebnis war, dass wir auf die Response-Zeit hauptsächlich achten wollen, weil wir haben die User-Perspektive als hauptsächlichlichen Scope dieser Tests und die Requirements, die wir herausgefunden haben, waren Reproducibility, also man sollte den Test erneut durchführen können, Automated, automatisierbar, reconfigurable, Modular, ich denke die meisten sind relativ selbsterklärend, Evaluable, also es sollte menschlich lesbar sein, die Ergebnisse des Tests, Scalable, Archivable, also die Ergebnisse sollten abspeicherbar sein, um sie zum späteren Zeitpunkt wieder aufzurufen, Disturbance-free, es sollten nur Funktionen verwendet werden bei dem Test, welche nur von dem System, das getestet werden soll, selbst auch wirklich ausgeführt werden, also sprich eine Excel-Mappe in SAP zu öffnen, da leistet Excel die Arbeit und nicht SAP sollte entsprechend nicht getestet werden.

Und genau, wir sollten mehrmals messen und ein arithmetisches Mittel nehmen, um temporäre Effekte auszuschließen beim Messen. Dann der Fortschritt seitdem, also die Entscheidung ist, dass wir die S&D-Case-Study verwenden [unverständlich]. Wir haben uns für Open [unverständlich] entschieden, um die Last zu erzeugen und zu generieren. Die Case-Study wurde gescrriptet mittels LoadRunner und die Tests wurden durchgeführt und Daten wurden gesammelt und die Daten wurden aufbereitet.

Ziel der Focus Group ist einmal eben diesen Entstehungsprozess zu evaluieren, dass ich da Feedback von euch bekomme. Der Prozess an sich scheint sinnvoll oder ihr hättet mehr Zeit hier und dort verwenden sollen oder darauf achten, Feedback zu den Tests selbst. Und ja, genau. Methode hätte ich jetzt Brainstorming und Diskussion vorgeschlagen. Es gab ja letztes Mal Feedback zu der Focus Gruppe, das versuche ich auch dieses Mal mit einzubauen. Das wir dann am Ende eine offene Aussprache und Brainstorming haben, einfach Sachen in den Raum geworfen werden.

Gut, ja genau, die Fokusgruppe wird aufgezeichnet und es wird ein Wortprotokoll erstellt. Ich hoffe, damit ist jeder, der hier anwesend, einverstanden.

[Ja von allen Teilnehmenden]

Dann würde ich euch den Prozess jetzt erst mal vorstellen, Stück für Stück die einzelnen Sachen, dass ihr da auch mal Feedback zu geben könnt. Einmal die ersten die S&D Case-Study. Ich weiß nicht, ob die jeder hier kennt.

[Ja von allen Teilnehmenden]

Dann wird dieser Teil sehr schnell gehen. Die verschiedenen Geschäftsprozesse, die da drin vorkommen, wurden alle entsprechend gescrriptet, dass sie von Loadrunner automatisch ausgeführt werden können. Dann kommen wir auch schon zu Loadrunner. Herausgeber ist OpenText. Das ganze ist in insgesamt drei zusammenarbeitende Programme aufgeteilt, eben sieht man das auch im Logo. Und zwar haben wir da einmal den sogenannten VuGen oder Virtual User Generator. In diesem Programm erstellt man die Scripts und man kann diese auch einmalig abspielen bzw. man kann sie öfters abspielen, aber nur eben um generelle Funktionalität zu testen. Man kann auch Abspieleinstellungen tätigen. Zum Beispiel soll zwischen jedem Schritt ein neuer Nutzer simuliert werden, was kontraproduktiv ist, weil es dann jedes Mal die Homepage neu lädt und SAP jedes Mal wollte, dass ich mich neu anmelde.

Dann gibt es den Controller. Der Controller erstellt und führt Szenarien, Testszenarien aus. Diese Testszenarien, so nennt sie Loadrunner, ist dann eben dass, wie viele Nutzer gleichzeitig arbeiten. Aber dazu komme ich noch genauer. Und das ist ermöglicht, gleichzeitig Nutzer zu simulieren. Und dann gibt es Analysis. Und auch wieder relativ selbst erklärt, man kann die Ergebnisse vom Controller auslesen. So, dann kommt die Scripterstellung. Das war der erste große Schritt, nachdem die anderen Sachen festgelegt wurden. Das ist in VuGen. Da wurde das Protokoll TruClient verwendet. TrueClient ist ein eigenes Protokoll von LoadRunner, was direkte Browser-Interaktionssimulationen ermöglicht. Also es schickt eben nicht nur HTML-Anfragen und liest die Antworten aus, sondern es rendert wirklich den Browser und sucht nach Buttons und klickt diese und tippt Sachen. Was halt für unsere Auffassung nach für den Scope User Experience und eben um die Simulation möglichst realistisch zu halten, recht wichtig war. Man erstellt durch Aufzeichnung von Browser-Interaktionen. Also ich klicke auf Aufzeichnen und gehe dann mit der Maus in ein Browserfenster, das sich zusätzlich öffnet. Da klicke ich dann auf einen Button oder klicke in ein Textfeld und tippe Text ein und VuGen zeichnet das alles auf und verwandelt das in ein Skript. Dazu gibt es eine eigene Benutzeroberfläche. Genau. Und es gibt kein direktes Coden, beziehungsweise Code ist teilweise nicht mehr sichtbar, sondern nur Pseudocode. Da habe ich auch gleich Bilder, ich gehe direkt zur nächsten Seite. Da seht ihr es etwas größer. Das ist einmal der Virtual User Generator. Links seht ihr die Skript-Solution-Explorer. Unten die Replay-Ergebnisse und da eben sozusagen das Code-Feld, wo ihr eben seht, das ist ein TruClient-Step. Click on More-Button, click on Sales and Distribution List Item, click on Manage Business Partners Link. Also das ist Pseudocode. Hier kann ich auch tatsächlich selbst drin gar nichts editieren. Ich muss oben, sieht man meine Maus noch? Ich muss hier auf Develop Script gehen. Dann öffnet sich ein neues Paar Fenster, das zeige ich auch gleich. Und da kann man dann entwickeln. Das hier würde eine Replay starten. Genau. Das nächste Fenster, wenn man dann auf Develop Script geht, sieht so aus. Rechts hat man eben den Browser, wo man die Interaktionen ausführen kann. Links ist eben das Development-Fenster, wo ihr diese einzelnen Schritte seht. 24, 29, click on Copy. 34, wait until Customer Reference Textbox exists. 35, type Customer Reference. Das ist in dem Fall ein Parameter. Man kann eben in VuGen auch Parameterlisten hinzufügen. Eben Customer Reference Box und dann kann man diese Schritte einzeln anpassen. Das ist das Argument, das Argument dieses Schritts ist eben ein Parameter das wählt man hier rechts aus und typed dann den Parameternamen. In Object kann man identifizieren, wo ist denn diese Customer Reference Box genau? Was ist der XPath oder was ist der JavaScript Path und HTML, um die zu erreichen? Transactions bezieht sich auf die Messung von Response Time und Step, da kann man zum Beispiel Timeout-Einstellungen treffen. Also wie lange soll ein VuGen oder Controller versuchen, diesen Schritt auszuführen, bis er es aufgibt, wenn er die Textbox nicht findet oder keinen Zugriff auf den Parameter hat. Genau. Das generelle Test-Framework kann man ähnlich wie Dr. Bögl sack, den habe ich ja auch gelesen, für die letzte Fokusgruppe in zwei Sachen unterteilen. Einmal die direkte Performance-Evaluierung pro Nutzer. Da haben wir uns ja für die Response-Time als Metrik entschieden und inklusive Browser-Renderzeit, weil das eben um die User-Experience geht. Also ich habe bei SAP festgestellt, dass oft vom Netzwerk die Daten längst übermittelt waren in meinem PC, aber ich locker nochmal 4-5 Sekunden da saß, bis das Fenster überhaupt sichtbar und für einen Nutzer interagierbar war. Und es wurden eben drei Messungen durchgeführt, um Durchschnittswerte zu erzeugen, um temporäre Effekte möglichst auszuschließen. Für die Skalierung haben wir dann verschiedene Test-Szenarien aufgestellt mit einer steigenden Anzahl an Nutzern, um genau zu sein immer verdoppelnd, 1, 2, 4, 8, 16, um zu schauen, wie verhält sich das System dann eben bei größerer Last. Genau.

A: Also eine Frage, drücken die User immer gleichzeitig die Knöpfe oder sind die zeitversetzt leicht?

Host: Also das ist eine sehr gute Frage. Da ist etwas, wo ich noch drauf eingehen würde später, also warum es zu Versatz kommt. Es kommt zu Versatz. Im Verlaufe des Scripts bei 16 Nutzern hat man sehen können, wie die Messungen mehr und mehr auseinandergegangen sind. Im Sinne von, dass es auf einen Schlag 16 Messungen angezeigt hat, kamen erst 2, dann 8, dann nochmal 2, dann 4. Die sind auseinander gedriftet. Genau gleichzeitig Aktionen sind potenziell umsetzbar. Warum ich das nicht tun konnte, komme ich später nochmal in der Präsentation dazu.

A: Willst du noch weiter vortragen oder bist du direkt dazu Feedback?

Host: Gerne.

A: Also ist es aus dem SAP-Sicht überhaupt nicht klug, Sachen gleichzeitig zu machen, damit Transaktionen sperrst und die im Overhead lock im HANA-Log generierst. Also tatsächlich hatten wir schon SAP-Systeme, die dadurch abgeschmiert sind, dass 500 Leute gleichzeitig den Refreshbutton gedrückt haben. Also das heißt für ein SAP-System ist nicht darauf gebaut, dass 500 Leute gleichzeitig den gleichen Knopf drücken, sondern dass jeder halt so sein Ding macht. Deswegen hätte ich gesagt, eigentlich führt der Script, würde ich jetzt zusätzliche Requirement eigentlich sagen, dass die Bots leicht versetzt klicken, weil du sonst eine Überlastung des Systems provoziert, die eigentlich nicht objektiv ist. Also das heißt, du könntest quasi dein System schon mit deutlich weniger Nutzern in kritische Situationen bringen, als das System eigentlich verkraften kann.

Host: Okay, verstehe. Das ist ein sehr gutes Feedback. Perfekt.

A: Also von daher würde ich eher sagen, es ist eher gut, dass die leicht versetzt klicken.

Host: Also bei 4 und 8 Nutzern waren die Klicks meistens relativ zeitgleich. Es kam immer zum leichten Versatz, aber der war marginal bei 8 Nutzern. Also es haben 6 und nochmal 2 geklickt mit einer halben Sekunde Unterschied.

A: Ja und ich glaube 4, 8, 16 ist kein Problem, weil wenn du das Ding auf 100 skalierst, dann beginnst du so ein System in kritische Punkte zu bringen, sobald er nämlich die ganzen Zugriffe nicht mehr locken kann. Also kann es sein, dass er bei der HANA-Q dann sagt, ich komme nicht mehr mit und dann schickt die HANA eine Nachricht an den Applikationsserver und dann sagt der [unverständlich]-error und bleibt hängen.

Host: Ich hätte mir auch vorstellen können, dass ein versetztes Nutzen des Systems A auch realistischer ist

A: Ist es auch.

Host: Selbst wenn vielleicht alle Zeit gleich anfangen, weil sie in einem Schulungsraum sitzen und der Prüfer sagt oder der Lehrer sagt, so und jetzt führt ihr die Case Study aus, klicken zwar vielleicht alle gleichzeitig auf Login, aber danach hat jeder sein eigenes Tempo. Gleichzeitig ändert es ja theoretisch gesehen auch nichts an der Last auf dem, korrigiert mich da bitte, am Applikationsserver und an der Datenbank, ob alle in der gleichen Maske arbeiten oder überhaupt nur 16 Leute gerade an dem System arbeiten.

A: Idealerweise ja, also wenn alle gleichzeitig irgendwelche krassen Workprozesse releasen, dann ändert es schon was, aber im Kern nicht mehr.

Host: Ich meine, man kann das System auch auslasten durch 16 Nutzer, wenn theoretisch gesehen jeder gerade in einer anderen App arbeitet, weil das ja anders ist.

A: Ja, natürlich, ja, ja, klar.

Host: Genau, also das ist aber sehr gut zur Wissen. Genau, die Skriptausführung war dann eben im Controller. Das heißt, da stellt man auch ein, welchen Skript man denn bitte ausgeführt haben möchte. Man kann scheinbar auch externe Skripts importieren, aber damit habe ich jetzt nicht rum experimentiert. Man kann einstellen, wie viele V-User, eben virtuelle Nutzer der Controller simulieren soll und Einstellungen bezüglich [unverständlich], Initialisierung, Start und Ende, sprich sollen alle

User gleichzeitig Initialisieren, sollen alle gleichzeitig starten. Was ist das Ende? Ist es ein Dauerschleifenskript, den man fünf Minuten lang laufen lässt und dann alle 30 Sekunden einen V-User beendet oder ist es ein Skript, das einen Anfang und ein Ende hat wie in unserem Fall, das soll laufen bis es zu Ende ist. Genau, und es speichert die gesammelten Ergebnisse in Logs und für Analysis in einem eigenen Dateiformat von LoadRunner. Genau, und so sieht die Oberfläche von Controller aus. Ich hoffe, man kann es halbwegs erkennen. Also hier, wo die Maus gerade rumschwebt, sieht man, dass da gerade aktuell 16 Nutzer am Laufen sind. Links der Skriptname, da kann man hier immer schauen, welcher ist down, welcher ist pending, Initialisiert, bereit, am Laufen, wartet auf Rendezvous, dazu komme ich noch, Passed, also hat den Skript erfolgreich abgeschlossen, Failed, hat einen Error, aber ist potenziell noch am Laufen, das waren die gruseligsten. Gradually Exiting, das heißt, er führt jetzt noch einzelne Transaktionen oder Steps, wo er gerade mitten dabei ist, noch zu Ende aus, Exiting, er versucht, das Ende auszuführen. Also es gibt in jedem Skript von Virtual User Generator eine sogenannte Initialisierungsaction, da kann man dann so etwas wie ein Login oder so etwas reinmachen, also etwas, was idealerweise genau einmal am Anfang von jeder Iteration stattfinden soll, und dann gibt es das eine Endaction, wo ich jetzt zum Beispiel den Logout reingeschrieben habe, also etwas, was halt genau einmal am Ende vom Skript passieren soll. Gradually Exiting wäre sie würde, Exiting ist, sie machen jetzt eben diesen Logout-Step und Stopped ist, welche haben aufgehört. Wir sehen hier rechts, wie viele Transaktionen durchgegangen sind, Transaktionen sind selbst eingefügte Schritte in dem Skript, welche die Response Time messen. Also mit der Transaktion wird gemessen von Transaktionsstart bis Transaktionsende, was ist die Zeit. Transaktionsstart habe ich dann immer so gemacht, dass es ein Button-Click ist, und Transaktionsende, weil ich ja eben das Rendern des Browsers eben mitberücksichtigen wollte, ist das Warten auf die Existenz von irgendwas, einen Button, ein Textfeld oder sonst irgendwas. Und beide sind genau Warten auf Netzwerke. Und hier sieht man die Response Time-Messungen von der Transaktion. Und da siehst du eben auch, dass es am Ende deutlich auseinandergegangen ist. Jede Farbe symbolisiert eine Transaktion und wir haben da sehr viele Messpunkte am Ende. Und hier eine Liste von allen Transaktionen, welche bereits durchgegangen sind, wie viele davon. Wir sehen hier Apply New Address Details ist genau 16 Mal Passed. Open BP Roles 2 ist 7 Mal Passed, also spricht 8 sind an der Stelle, 9 sind an der Stelle noch am Arbeiten oder sind gerade dabei, haben noch gar nicht angefangen, weil sie zurückliegen oder sind noch nicht fertig. Und 2 haben sogar schon eine neue Sales Area created. Dieses Apply New Address Details und Open BP Roles 2 war eine der fehleranfälligen Schritte. Die haben sehr lange gebraucht, bis die flüssig durchgelaufen sind.

(A, E): Konntest du feststellen, was genau die Probleme dieser Schritte waren?

Host: Ja. Ganz am Anfang war es wahrscheinlich die automatische Objekterkennung von LoadRunner.

A: Also, dass sie die Buttons erkennt.

Host: Ja, oder hat Buttons verwechselt, weil zum Beispiel bei der Create Business Partner gibt es sehr viele Create Buttons. Wenn man die mit einem Deskriptor im Sinne von Button heißt Create an Loadrunner übergeben hat. Und durch irgendwelche komischen Jumps, da habe ich sehr irreguläres Verhalten auch von SAP teilweise beobachtet, wo ich zum Beispiel auf Show More Details geklickt habe und aus irgendeinem Grund die App auf den kompletten Boden der Seite gescrollt ist und ich dann erstmal wieder 10 Minuten hochscrollen musste, bis ich da war, wo ich etwas eintippen wollte. Wenn so etwas beim Replay passieren sollte und er dann halt gerade im Bild einen anderen Create Button sieht, als der den er eigentlich drücken sollte, kam es zu Problemen. Oder, was ein ganz witziges Problem war, ist, dass eine Seite, und zwar war es Bestellung anlegen, zwei verschiedene Layouts hatte, wo einmal entweder das Menüband permanent sichtbar oben war und es ein Go-To, Header, Conditions gab, oder aber, wo das Menüband extrem eingeklappt war und man erst auf Menü, Go-To, Header, Conditions klicken musste.

A: Ich habe dazu eine Frage, wie funktioniert LoadRunner, dass er quasi eine Windows-Session simuliert, wo er dann tatsächlich auf einem GUI rumklickt, also lässt er sich tatsächlich eine grafische Information schicken, auf der er rumklickt, oder wie genau funktioniert das?

Host: Also, so wie ich das verstanden habe, läuft er im Gegensatz zu anderen HTML-Coding, wo man nur HTML responses schickt, simuliert er wirklich Browser-Interaktionen.

A: Okay, das heißt, er tut quasi wirklich so, er lässt sich quasi tatsächlich auch Zeit,

Host: um was einzutippen.

A: Okay, und er lässt sich quasi auch wirklich das Browser, also das Anzeigen quasi, also die Information des Browsers anzeigen und klickt da drauf quasi rum.

Host: Ja. Und das Problem war bei diesem Menüband, dass, wenn ich gesagt habe, am Anfang geskriptet habe ich es, als wäre das gesamte Menüband da gezeigt worden, habe gesagt, ja, pass, klickst auf Go-To, Header, Conditions, so dann ist das Skript gefailt, habe Go-To nicht gefunden. Ich sag, wie du hast Go-To nicht gefunden? Bin wieder rein, habe alle Schritte bis da eine abgespielt und ich habe das Menüband wieder gesehen. Ich sage, ich verstehe es nicht, habe Feierabend gemacht, am nächsten Tag reingegangen und den Skript abgespielt mit offenem Development, dann sehe ich auch, wie der Browser sich verändert. Und dann plötzlich komme ich in dieses Feld und warte und so, ja, und jetzt wird er Go-To finden und ich verstehe wieder nicht, was das Problem ist. So, nee, er findet es nicht, Moment mal. Und dann habe ich gesehen, dass das Layout anders war. Dann habe ich es umprogrammiert und am Ende musste ich dann meine If-Claus einbauen nach dem Motto If Button Menu Exists, then Go to Menu, Go-To, Header, Conditions, Else Go to Go-To, Header, Conditions. Ich habe einige Runs mit 16 Nutzern, wo tatsächlich ich sehen kann in dieser Transaktionshistorie, dass tatsächlich beide Versionen ausgeführt wurden. Oft war die eine Option nur einmal vertreten, aber es waren oft beide Optionen vertreten.

D: Aber was bedeutet das, dass SAP sein Layout spontan ändert?

Host: Keine Ahnung.

D: Okay

A: Du klickst in Fiori rum, ne?

Host: Ja.

A: Zur Fehlerfindung wird es vielleicht nicht richtig gecached.

B: Es gibt mehrere Optionen. Das Browserfenster kann unterschiedlich groß sein, dass der oder durchläuft. Dann wird automatisch das Collapsed. Es kann länger dauern, etwas abzufragen vom System. Dann wird halt zuerst dieses kleinere Version geliefert. Genau. Das sind eigentlich beiden Optionen, die mir jetzt spontan eingefallen sind.

Host: Also Browserfenster war immer gleich groß beim Development und beim Abspielen. Cache kann sein. Ich habe Screenshots von beiden Layouts, die sind in der Bachelor-Arbeit. Ich habe jetzt nicht hier mit dabei. Das wäre dann vielleicht interessant für dich, die werde ich da einfügen würde. Also das waren Probleme. Solche Probleme habe ich dann immer wieder, hab ich finden und lösen können. Bei diesem Button-Click im OpenBP_Rolls_2 Apply_New_Sales_Area war es am Anfang ein Timeout. Bei dem einen, beim anderen war es, dass es eben diesen Button nicht gefunden hat. Create. Dann habe ich statt Descriptors Xpath verwendet. Das heißt, ich habe die HTML-Seite durchgeklickt und den eindeutigen Pfad von diesem Button eingefügt. Dann hat das Problemlos jederzeit gefunden. Genau.

E: A, für dich zum Verständnis nochmal kurz. Wir messen halt auch bis zum Ende der Darstellung, weil du gefragt hast, wo wir uns befinden. Wir messen nicht bis zum Ende und sagen, wir haben jetzt Datenempfang, sondern er hat jetzt gewartet, bis die Daten noch dargestellt wurden.

A: Alles klar.

Host: Also es gibt da verschiedene Steps, die man in Virt, in Vugen einfügen kann, die heißen, sowas wie Generic Browser Action und dann kann man diese anpassen. Zum Beispiel kann man dann eben sagen Generic Browser Action mit irgendeinem Objekt. Dann wähle ich das Objekt aus Button, dann stelle ich ein, was soll es mit dem Button machen. Soll es den Text kopieren, soll es den Text einfach nur lesen, soll es verifizieren, dass dieser Text dort steht oder soll es einfach nur auf die Existenz von

diesem Button warten und gibt auch den Step Wait for Text Being Visible. Das kann man da alles einfügen. Genau, dann geht es in Analysis. Da kann man eben die vom Controller gesammelten Daten auslesen. Man kann sie sich in verschiedenen Grafen darstellen lassen und man kann sie auch in verschiedene Formate exportieren. CSV, XLSX, Texte .txt. Analysis sieht dann erstmal so aus. Da oben seht ihr einen Reiter mit den verschiedenen Graphen. Bei der Maus, die man sich erstellen lassen kann, das ist der Transaktionsübersicht. Der zeigt einfach nur an, welche Transaktion ist wie oft erfolgreich durchgegangen. Da alles grün ist, ist es keine gefailt. Logisch, wir wollten ja, dass alle Nutzer durchlaufen können. Also war alles, wo es ein Fail gab, abgebrochen und neu gestartet oder versucht zu fixen, noch mal zu starten. Genau und hier oben kann man sich auch Raw-Daten erstellen lassen, mit denen ich sehr viel gearbeitet habe oder eben direkt in andere Formate exportieren. Gut, dann würde ich euch ein paar Ergebnisse vorstellen jetzt von diesen ersten Messungen. Es ist sehr interessant, aber scheinbar haben 16 Nutzer im Durchschnitt doch nicht gereicht, diese Maschinen an ihre Grenze zu bringen.

A: Hoffentlich.

Host: Beim Login sieht man allerdings recht deutlich eine Änderung. Unten die Szenarios, das sind eben 1, 2, 4, 8 und 16 Nutzer. Szenario 5, 16 Nutzer, 4, 8, 3, 4, 2, 2, 1, 1.

F: Sorry, ich wollte es schon, vielleicht hättest du es eh gleich erklärt, aber was bedeutet das durchschnittlich oder gesamt?

Host: Das ist der Durchschnitt von den drei Messungen.

F: Heißt es das, also heißt dann diese Grafik, dass ein Nutzer braucht?

Host: Der Durchschnitt von allen 16 Nutzer in Szenario 5 und der Durchschnitt von dem Durchschnitt aller 16. Also ich habe von den 16 Nutzer den Durchschnitt gebildet, diese Messungen dreimal und dann eben von diesen drei Durchschnitten noch mal den Durchschnitt gebildet.

F: Okay, und das bedeutet aber quasi jetzt im übertragenden Sinne, dass wenn ich 16 User habe, der Login,

Host: wenn er nicht gecached ist.

F: Wenn er nicht gecached ist für einen einzelnen User im Durchschnitt

Host: 18 Sekunden

F: dauert, währenddessen wenn ich alleine auf dem System bleibe, sind es 9 Sekunden.

[unverständlich]: Das ist ja ganz schön lang.

F: Das ist schon interessant.

Host: Wir sind bei 9, ungefähr noch 8,5 bis 9 Sekunden bei einem Nutzer. Wenn er gecached ist, ist er selbst bei 16 Nutzer, liegt er bei 1,5 Sekunden im Durchweg. Also wenn ich einen Test abgebrochen habe und bei den Parametern nicht die Nutzer geändert habe, die verwendet werden sollen, war der Login plötzlich immer bei 1,5.

A: Das ist echt spannend, es zeigt nämlich wie relevant cache ist. Das ist ja was wir zwischendurch immer als Beobachtungswert hatten, aber jetzt ein bisschen auf einer Evidenzbasis.

Host: Also gecached war da immer sehr praktisch. Dann ist aber oft meistens sehr linear gewesen. Wir haben zwar immer eine leichte Tendenz zu sehen, ok 16 brauchen ein wenig mehr als einer, aber generell ist alles dann doch relativ linear gewesen bis 16 Nutzer. Unten, das sind die Namen der Transaktionen. Also da habe ich jetzt, ich habe alles in größeren Steps geclustert. Also eben diese Schritte aus der Case Study, Kunde anlegen, Kontaktpersonen anlegen, Inquiry anlegen und das ist jetzt der Step Create Customer. Da wird eben erstmal die App Manage Business Partners geöffnet, es wird eine neue Organisation erstellt, es werden die Roles geöffnet, New Company Code eingetragen.

Alles Tendenz steigend bei 16 Nutzern, aber nicht sehr krass, sage ich mal, von der Verlängerung. Dann gab es ganz Fälle, wo sich innerhalb einer Aktion, sage ich mal, waren einzelne Funktionen von SAP, die sehr linear geblieben sind, also scheinbar wirklich unglaublich leichtgewichtig sind und auch bei 16 Nutzern nicht lange brauchen und wiederum eben einzelne Funktionen, die doch durch die 16 Nutzer beeinflusst wurden. In dem Fall ist es eben die App Manage Billing Document überhaupt erst mal zu öffnen, hat bei 16 Nutzern drastisch länger gedauert als bei eben einem. Also drastisch, wir gehen von zwei Sekunden auf 2,8, aber während wiederum Search Billing Document, Post Billing Document sehr linear eben, also keine wirklicher Einfluss und dann gab es sehr witzige Fälle, sehr interessante Fälle, sage ich mal lieber, wo tatsächlich fünf Nutzer schneller waren als, Gesundheit, als 4 und 8. Das war ein Reload der Homepage, nachdem wir die Sales Order angelegt haben. Also einfach nur Reload zu drücken. Um das auszubügeln, ich kann natürlich mit drei Messungen immer noch nicht die effekte komplett ausschließen, aber ich wäre auch interessant zu wissen, woran das liegen könnte.

B: Vielleicht hast du ja auch ein internen Cache-System, dass du öfter die Page abfragst, dass es die gleiche Information zu mehreren Leuten schickt.

Host: Gut, aber dann müsste ja trotzdem auch bei acht Leuten schneller gehen.

B: Stimmt, ja.

[unverständlich]: [unverständlich]

A: Man sollte mit 32 und 64 noch messen und gucken, ob das gerade noch weiter nach oben geht oder weiter nach unten geht.

E: Ich weiß nicht, ob das gerade gesagt hast, weil wir es beim letzten Mal im Gespräch hatten, sind das jetzt die Daten vom Durchschnitt oder vom Median?

Host: Durchschnitt.

E: Okay

Host: Median war im Durchschnitt etwas niedriger als der Durchschnitt, aber

F: Okay

A: Heißt dann Outlier nach oben, ok.

Host: keine große Änderung. Ja vor allem bei den 16 Nutzer hat man gesehen, dass immer vier bis sechs sehr schnell mit den Sachen fertig waren und dann die anderen haben sich bitten lassen. Gut, zusätzliche Erkenntnisse, da kommen wir jetzt zu den Sachen mit 32 und 64 Nutzer-Tests. Extrem starke CPU-Auslastung auf Test der Maschine mit hohen vusern. Also ich habe einen relativ leistungsfähigen PC. Wir reden von 16 Kernen mit 96 MB V3-Cache und 4,2 GHz Taktung. War bei 16 Nutzern am permanenten Ausschlag von 99 Prozent. Und ab, das war der Moment.

A: Ist es die Maximalnutzung oder war das auch durchgehend?

Host: Nicht durchgehend, sondern die Maximalnutzung.

A: Ja, aber diese SAP ist zwar mal funktionieren, also man darf auf diese 99 Prozent.

Host: Nicht SAP.

F: Auf dem privaten.

Host: Der der Controller ausgeführt hat.

A: Okay, das ist interessant.

Host: Der Controller ausgeführt hat und es waren immer diese Spikes an CPU-Auslastung, wo dann Fehler geschehen sind. Wo ein vuser versagt hat, einen Button zu finden, 15 andere ihn aber gefunden haben. Also an dem Scripting kann es nicht liegen. Er scheint ja zu funktionieren, sonst würde keiner es finden. Und das waren dann immer genau die Fälle, wo Fehler passiert sind. Ich kann mir auch nur vorstellen, dass eben aufgrund der Browsersimulation die CPU-Nutzung so drastisch höher ist. Oder so allgemein ob es so hoch ist. Aber es war mir physikalisch unmöglich 32 Nutzer zu testen. Das war ursprünglich angedacht, aber mit meinem eigenen PC nicht möglich. Zeitlich könnte es jetzt etwas kritisch werden, noch eine Virtual Maschine vom LRZ oder so mit 64 Kern oder 32 Kern anzufragen, um diese Tests durchzuführen. Aber für mehr als 16 Nutzer braucht es definitiv mehr als 16 Kerne CPU. Im besten Fall, also im allerbesten Fall wahrscheinlich ein Threadripper mit 64. Oder Xeon, wenn man lieber Intel mag. Genau, das war, das war teilweise schwierig. Das hat mich auch sehr viel Zeit gekostet, bis ich das realisiert habe. Und dauernd dachte, versucht habe zu verstehen, warum manche Schritte im Script bei einem Nutzer fehlschlagen. Und bei anderen nicht, wo ich dachte, ist vielleicht, ist es ähnlich wie Automatic? Ist es ein Glückspiel, ob er es findet oder nicht? Hat der Seite gescrollt? Muss ich es irgendwie robuster machen? Mit mehr XPath, mehr JavaScript etc. Am Ende war es dann eben gerade die Erkenntnis, dass das Ding bei 8 Nutzern absolut fehlerfrei durchläuft, bei 15 Nutzern fehlerfrei durchläuft. Aber bei Punkt 16 so die Grenze ist, wo immer wieder einer gefailt ist. Ich habe dennoch es geschafft, meine drei oder insgesamt sogar vier fehlerfreie Messungen mit 16 Nutzern durchführen zu können. Aber es hat sehr lange gedauert. Also ich glaube, weil es war, am Ende konnte ich wirklich im Script nichts mehr ändern. Es war wirklich wie im Lotto. Ich drück auf Starten vom Replay, vom Controller und hoffe, ob es fehlerfrei durchgeht oder nicht.

F: Und dementsprechend, wenn die Maschine auf der SAP läuft, runtergesized wird, dann hast du quasi dasselbe Problem, oder?

Host: Weil es immer noch mein PC ist, der den Rest auslöst.

F: Also du könntest dann nur sagen, also es wäre also rein hypothetisch, wäre das Einzige, was möglich wäre, eine skurrile Lösung, wo man sagt, ich mache es so klein. Aber dann startet vielleicht HANA gar nicht mehr. Wenn du sagst, ich gebe dir jetzt irgendwie 60 Gigabyte RAM, damit ich halt den Unterschied vielleicht mehr sehe zwischen einen und acht Usern. Aber dann hat man wahrscheinlich die Gefahr, oder, dass du sagst, irgendwann würde SAP gar nicht mehr starten.

B: Warum verwendet ihr keine LPAR von uns für das?

F: Dann müsste man es wahrscheinlich nicht, also da ist ja die Frage...

A: Läuft LoadRunner auf Power?

Host: Das ist nicht bekannt.

F: Gute Frage

Host: Also weiß ich nicht.

A: Du hast es jetzt auf Windows laufen, oder?

Host: Ja.

[unverständlich]: Man könnte natürlich einen x86 LPAR, wie viel RAM braucht das Ding?

Host: Ich hab jetzt auf meinen 16er-Test 32, ungefähr 30 bis 32 verwendet.

[unverständlich]: Das ist schon viel. Das ist zu viel für ne RBG.

F: Ja, das ist das Problem.

E: Das war jetzt eben unser Problem. Deswegen haben wir jetzt gesagt, okay, das ist eine Problematik, die komplett neu ist. Und die muss man halt neu erfassen, sage ich mal.

A: Ich glaube, langfristig würde ich sagen, sollten wir das sowieso bei uns deployen, weil das wollen wir ja auch als internes Testing benutzen. Da sind wir ja Holger und ich gerade dran, ob wir die RBG dazu motiviert kriegen, uns größere LPARS zu bauen. Aber das ist einfach das, das ist grad noch nicht fix.

Host: Also entweder könnte man sich überlegen, wenn man es irgendwie argumentieren kann mit dem Budget, sich eine dedizierte Workstation eben mit einem Threadripper oder sowas hier für das UCC anzuschaffen, auf der dann die Tests ausgeführt werden können. Oder sich über Amazon Web Services oder das LRZ, je nachdem, was geht oder flexibler ist, einfach für die 1, 2... Weil Skripten kann man ja, man kann es auch mit 8 Nutzern oder sowas testen. Aber für die großen Ausführungen, wenn man weiß, ok, das Skript funktioniert, dass man dann einfach sich für eine Stunde eine Amazon Web Service mit 64 Kernen oder so eine VM holt, einmal die Tests durchlaufen lässt und sagt so, passt, danke, ich habe meine Ergebnisse.

B: Klassischer Anwendungsfall für Cloud.

A: Ich glaube auch, das ist wirklich ein klassischer Anwendungsfall.

Host: Ja, eben, einfach irgendwie, irgendeine Leistung, irgendwoher. Sei es LRZ, sei es Amazon, sei es Google. Genau, Debugging teils sehr aufwendig, weil wenn ich nur auf Replay gedrückt habe, waren die Logs riesig. Also ich konnte einstellen, was es loggen soll und im Endeffekt waren 99% davon entweder für mich, mit meinem Wissenstand unverständlich, und irrelevant. Und ich habe teilweise durch 17.000 Zeilen an Logdaten gescrollt, bis ich endlich die Info gefunden habe, was nach 20 Sekunden Skriptausführung schiefgelaufen ist. Oder ich habe gar keine Infos, weil mir es dann einfach immer nur sagt, gefailt in diesem Schritt. Warum? Ob er kontinuierlich identifiziert werden konnte. Warum nicht? Dann heißt es entweder rumprobieren, ich habe das meist mit Trial and Error gemacht, im Endeffekt einfach in das Programm eingesetzt und drauf los gestartet. Es hat am Ende zu einem funktionierenden Skript geführt. Und das Debugging war eben dann immer wieder nur zu überlegen, zu testen, immer wieder zu testen. Problem, man kann das Skript ausführen, mit einem Browser angezeigt daneben, was gerade passiert, nur in VuGen mit genau eben einem Nutzer. Das hat zwar geholfen, um solche Sachen wie dieses Menu Loadout zu identifizieren. Oder ein ganz skurriler Fehler war bei der Einbuchung der Rechnung, der eingehende Zahlung, hat der Skript gesagt, tip Kundennummer in das Feld und dann drück auf Propose. Das würde mir jetzt vorschlagen, welche offene Rechnung mit diesem Zahlungseingang getilgt werden könnte. Und das hat alles funktioniert, wenn ich es manuell gemacht habe. Und super, Skript ist fehlgeschlagen. Aber wirklich 16 von 16 Nutzern gefailt. Also das ist jetzt nicht nur CPU-Lust, das ist wirklich ein Fehler. Und habe es mir dann einmal abspielen lassen, mit wenig Hoffnung, dass ich etwas herausfinde mit einem Nutzer. Da habe ich festgestellt, dass wenn ich die Kundennummer eintippe und auf Propose drücke, ein Fehlerfenster aufploppt mit der Info, ich muss erst einen Kunden festlegen, bevor ich etwas vorschlagen kann. Was hat gefehlt? Ein Schritt, wo ich Enter drücke auf die Kundennummer. Da ich das scheinbar immer abseits des Recordings gemacht habe, ist mir der Fehler nie wiederfahren. Das ist so ein Debugging, wo man sehr viel mit Trial and Error, einfach mal reinschauen, überlegen, versuchen nachzudenken, was könnte an dieser Stelle schiefgegangen sein? Kann es zu weit gescrollt haben? Gibt es mehrere Buttons, die so heißen? Was ist die Objektidentifizierungsmethode? Ist es automatisch? Ist es Skript oder ist es XPath? Sind es dynamische Werte? Also es gibt ja, ähm, das habe ich immer im HTML gesehen, wenn ich irgendwas bei Descriptor hatte oder JavaScript und es gefailt ist, dass dann die ID immer was mit Clone 2, 4, 7, 8 drinstehen hatte. Also es ist irgendein dynamischer Wert, etwas, was von SAP jedes Mal neu generiert, neu erstellt wird, wenn man die Seite lädt. Entsprechend, wenn ich versucht habe, das mit JavaScript zu öffnen, ist natürlich fehlgeschlagen, weil diesen spezifischen Clone hat es nicht mehr gefunden. XPath war da meistens zuverlässiger und stabiler. Ähm, und... Irgendwie einen Punkt hatte ich gerade noch auf meiner Zunge. Ah ja, IFrames. IFrames war eine große Herausforderung. Weiß jeder, was IFrames sind? Nein, okay. IFrame ist praktisch ein HTML-Dokument innerhalb eines HTML-Dokuments. Also eine Encapsulation. Und VuGen hat prinzipiell erstmal nur auf das prädominante HTML-Dokument zugegriffen. Ein Problem war das, als ich im Zuge der Case Study die Kunden-Nummer von einem neu erstellten Kunden, die Bestellnummer oder die Angebotsnummer oder die Inquirynummer von einem dieser drei kopieren wollte, weil diese Nummer, die angezeigt wurde, war innerhalb eines IFrames. Und dann musste ich einen JavaScript innerhalb von dem VuGen-Skript erstellen, der aktiv diesen IFrame erstmal auswählt und anvisiert. Und dann konnte er innerhalb des IFrames nach der Nummer suchen und diese kopieren. Was wiederum zu einem neuen Problem geführt hatte, dass

wenn ich dann im nächsten Schritt einen neuen Browser-Tab geöffnet habe, weil das Anlegen von einer Bestellung hat automatisch einen neuen Tab geöffnet, wo dann das generierte PDF angezeigt werden sollte, dass dadurch LoadRunner komplett den Fokus verloren hat, weil es auf diesen IFrame fokussiert war, im neuen Tab aber diesen IFrame nicht mehr fokussiert bekommt, dass sämtliche Interaktionen danach viel geschlagen sind, so als wäre gar kein Browser mehr offen. Das wurde gelöst, indem ich nach dem Kopieren der Nummer, einen Reload der Homepage, der [unverständlich] Script drin hatte, weil das den Fokus vom IFrame wieder weggenommen hat auf die gesamte Page. Und relativ steile Lernkurve, also die ersten paar Wochen mit LoadRunner, keinen Spaß gemacht. Aber wenn man erstmal ein bisschen drin ist und sieht, wie das funktioniert, wie man diese Steps verwenden kann, was man für Einstellungsoptionen hat, XPath, Descriptors für Objektidentifizierung, Wait-steps, If-Clauses, Catch-Error-Clauses, ist das ganze relativ intuitiv am Ende und tatsächlich auch manchmal sogar recht angenehm zu verwenden. Genau, dass man die zusätzlichen Erkenntnisse. Und nach dem kleinen Vortrag danke für eure Aufmerksamkeit und genau, gerne euer Input und da es ja Brainstorming ist, hätte ich hier so ein kleines Brainstorming Canvas. Man sieht es nicht. Fantastisch. Okay, ich teile scheinbar nur die Powerpoint und nicht nur ein Bildschirm.

[verschiedene]: {Diskussion, wie die Freigabe korrigiert werden kann}

Host: Also wenn ihr Fragen habt zu dem System oder wie alles abgelaufen ist oder wie es funktioniert, wie A ja auch schon Fragen gestellt hat, dann die sehr gerne. Andernfalls einfach euer Feedback. War mein Vorgehen diesen Test zu erstellen sinnvoll? Habe ich vielleicht doch noch Requirements übersehen, jetzt wo ihr gehört habt, wozu das alles praktisch geführt hat, wie es weiter abgelaufen ist, würden euch dann vielleicht jetzt noch zusätzliche Requirements einfallen? Etwas, was man beachten muss, ähm, Feedback zu dem Test selbst, da haben wir ja auch schon gehört, 32 und 64 Nutzer im besten Fall. Bei 64 bräuchte man eine Lizenz von LoadRunner, die kostenlose geht nur bis 50.

A: Was kostet die?

Host: 24.000 pro Semester

A: Was?

Host: Ja

A: 24.000

Host: Mein letzter Info-Standard.

A: Die haben keine andere Lizenz, die haben nur 0 und 24.000. Was ist denn das für ein Lizenzmodell?

Host: Also man kann ja sicher mal als Institut bei Ihnen noch mal anfragen. Ich glaube, es wird immer, wenn man anfragt.

A: Also ich finde, das klingt nach einem super whacken Lizenzmodell.

F: Also anscheinend ist es auch das, was SAP tatsächlich selbst benutzt. Aber die können das halt auch leisten.

A: Ich sag's mal so, das ist eine große Firma 24.000 zahlt. Kein Problem. Aber was ist denn das für ein beknacktes Konzept mit 24.000? Überleg mal, wie viel Leute du da raus schmeißt. Eigentlich ein viel besseres System wäre das so was wie 10 pro Nutzer oder so.

Host: Also ich habe jetzt auch nicht zu tief in das Lizenzmodelle recherchiert. Ich habe herausgefunden, es gibt eine kostenlose Version.

A: Also weil ich habe jetzt gerade gesagt, also aus unserer Erfahrung aus dem UCC heraus hatten wir eigentlich, also echte Load-Probleme. Also nicht Probleme, die durch schlechte Hardware, merkwürdiges Verhalten des S4/HANA -Systems oder Fehlbenutzung erzeugt wurden. Also das wir jemals Probleme hatten mit Leuten im Kurs. Das waren immer Kurse mit über 100 Teilnehmern. Also

das heißt, ich glaube, also ich fand diese Messe und das war alles nett und schön und das zeigt, dass Cache wichtig ist und alles. Aber wenn du wirklich sagst, wir wollen loadtesten als UCC, also wir wollen irgendwie das benutzen, LoadRunner nutzen, irgendwie um unsere Systeme krisenfest zu machen, dann brauchen wir halt minimum 100 Kurs.

F: Ich hätte eine Anmerkung dazu. Also wer da war, das hatten ja E und ich oder E gleich hatte das auch schon mal erwähnt, wäre eigentlich einfach eine ganz stupide, simple Lösung, sich fünf E-Mail-Accounts zu machen, sich fünf PCs nebeneinander zu stellen und halt fünf Mal quasi,

Host: Und jede benutzt dann den gleichen Script

F: Weil ich meine, am Ende des Tages wird es ja auf die gleiche Maschine bei uns zugegriffen, also bevor man halt 24.000 Euro ausgibt.

A: Ja natürlich, sowas könnte man natürlich auch machen, aber ich finde, für mich klingt das irgendwie nach einem so whacken Geschäftsmodell.

Host: Also ich würde nochmal rein recherchieren oder auch einfach anfragen, hey was habt ihr noch für Lizenzen?

F: Vielleicht kannst du das echt machen.

Host: Das ist ja, das ist, kann man ja fragen, ich hab nur geschaut, wie kann ich LoadRunner nutzen, hab dann irgendwo diesen Preistag gesehen und dann habe ich mit dem nächsten Link auf die auf Google schon gesehen, dass es hier eine kostenlose Testversion für bis zu 50 simulierte Nutzer gibt, ja passt für mich. Machen wir 32 und 50 und können ja dann...

A: Also klar kannst du 5 LPARS mit 32 Nutzer machen, aber das ist, also ich finde, irgendwie kann ich mir das noch nicht vorstellen, dass die nur 0 oder 24.000 Euro haben.

Host: Das ist jetzt nicht in Stein gemeißelt, dass war nur meine Info.

A: Genau, genau, also ich zweifel das gar nicht, was du sagst, ich glaube nur, dass man sich da vielleicht mit dem Konzern in Kontakt setzt. Also einfach von mir als Requirement, ich würde halt sagen, wenn du wirklich sagen willst, also wenn wir wirklich sagen wollen, wir wollen das nutzen, um mal die UCC-Systeme krisenfest zu testen, dann brauchen wir halt mal minimum 100, eher 150.

Host: Die grundlegende Idee, aber dafür kann das Script natürlich auch, sage ich mal, erweitert oder genutzt werden für die Krisensicherheit, war eher der Vergleich zwischen Technologien und diese Skalierung war gedacht, um zu schauen, okay, weil angenommen, man würde jetzt bei 16 Nutzern schon was sehen, zum Beispiel über Login, jetzt würden wir zum Beispiel Container einführen, dann könnte man jetzt bei der Skalierung sehen, okay bei einem Nutzer ist der Container 2 Sekunden langsamer, aber bevor man es verwirft, schaut man sich die Skalierung an, aber bei 16 Nutzern 8 Sekunden schneller, zum Beispiel, dafür wäre es ja auch schon.

F: Dafür muss man sagen, dass das ja aus der Idee von dem OpenShift-Cluster kam, dass man eigentlich sagte, so der Wolfgang, der Holger auch, dass man erstmal sagte, man müsste eigentlich mal einen Performance-Test machen, da kam die Idee, dass man sagt, funktioniert zum Beispiel SAP S4/HANA auf den Containern, also wäre der nächste Schritt auch, das Cluster eigentlich wieder aufzubauen und genau dann eben diese Red Hat Containerlandschaft damit zu testen und genau für dieses Pad will ich auf jeden Fall das, was A da vorgesagt hat, aufschreiben, dass man sagt, Test soll nicht punktgleich sein und was ich denke, was auch wichtig ist insgesamt für diese Bachelorarbeit, ist das, was mir irgendwann im Laufe der Besprechungen so aufgefallen ist, dass die Ergebnisse oder die Probleme müssen irgendwie generalisiert oder die Lösungen sollten generalisierbar sein, weil du hast ja quasi gemerkt, du bist jetzt bei dieser Fallstudie auf verschiedene, sei es iFrames oder sonst irgendwas, auf verschiedene Probleme gestoßen und wahrscheinlich wirst du viele davon in anderen Fallstudien, würde ich jetzt einmal vermuten, auch treffen und dass du sagst, also dass man jetzt, ja genau, dass du sagst, so iFrame sollte man das machen, das sollte man das machen und wahrscheinlich wird es auch neue Probleme geben, aber dadurch, dass du schon, ich will mal sagen,

ein breites Set an Lösungen herausgefunden hast, vielleicht kann man da auch irgendwie dann die neuen Probleme irgendwie ähnlich lösen oder so. Genau. Sonst stimme ich dem A auch zu.

Host: Danke schon mal dafür. Das Ganze ist ja im Endeffekt DSR, das heißt, das ist die erste Iteration, es wurde ein Problem identifiziert, das Problem war ein fehlender, standardisierter Test für eben die Systeme, für Vergleichbarkeit, dann wurde der Requirements-Analyse durchgeführt, das war unsere erste Fokusgruppe, und jetzt wurde ein Artefakt entwickelt. Jetzt geht es praktisch um die Evaluation, okay, wie ist dieses Artefakt, taugt das für das Ziel? Inwiefern könnte man es verbessern? Was war vielleicht Overkill? Und wenn euch da irgendwas einfällt, von wegen eben z.B. 32 Nutzer könnte man testen. Man könnte eine zweite Case Study noch mit aufnehmen. Ein Vorschlag von mir wäre gewesen, die Rendezvous-Punkte, aber da haben wir jetzt schon das Feedback, das soll auf keinen Fall stattfinden, weil Rendezvous-Punkte sind etwas, was man in VuGen einfügen kann, um alle virtuellen Nutzer dazu zu zwingen, etwas gleichzeitig zu machen, sprich, wenn einer schneller ist, wird er an einem Rendezvous-Punkt warten, bis alle anderen an der gleichen Stelle sind. Das frisst auch noch mal substanziell mehr CPU.

F: Ich denke, dass vielleicht, was mich auch verwundert hat, vielleicht, lege ich da natürlich auch falsch, vielleicht müsste man tatsächlich noch mehr als drei Durchgänge machen, um den Mittelwert zu bilden, weil ob da jetzt wirklich, also ich kenne bei manchen Sachen, dass bei 16 zeitlich runtergeht, also für mich müsste das gleich oder mehr sein, es könnte gleich oder mehr sein, aber vielleicht könnte es schon sein, wenn die Zeit wieder runtergeht, dass es einfach drei glückliche Läufe waren, so in etwa. Um das halt wirklich verlässlich zu sagen, dass man eigentlich sagen müsste, keine Ahnung, dass man das dann noch mehr, aber wie gesagt, es ist schon, es war auf jeden Fall sehr gut, dass du das schon mehrmals gemacht hast, aber das wird mir auf jeden Fall noch einfallen.

B: Also, wenn man das jetzt nutzen möchte für die Test von neuen Systemen, dann müssten wir auf jeden Fall mehr Case Studies drin haben, mehr Nutzer logischerweise, gerade wenn man es an Unis ausgeliefert mit vielen Studenten, ansonsten fällt mir eigentlich nichts ein, außer die Reihen.

F: Und was mir auch noch gerade einfällt, ist, man müsste vielleicht so, weil man ist ja ein bisschen von deinen LRZ, Amazon hast du erwähnt oder privaten Rechnern dann quasi abhängig, dass man sozusagen so ein LoadRunner-as-a-Service hätte, dass man sagt, jetzt müssen jetzt nicht irgendwie in den Hiwi da hinsetzen, dann erst mal fünf Mal LoadRunner jedes Mal neu installiert auf jeder neuen Amazon Maschine und dann so, sondern dass man sagt, wie du vorhin im Gespräch gesagt hast, dass du so ein Image oder so eigentlich hast, dass da wo schon alles fest drin ist und das wir uns dann quasi aufspielen und dann klickt einer auf LoadRunner und geht los und zusätzlich das, was wir vorhin auch angesprochen haben, fand ich auch gut mit diesem, wenn es wirklich so eine perfekte Maschine mit so vielen Kernen gäbe, dann bräuchte man noch eine zusätzlichen privaten PC, der als Testmaschine dient vielleicht, wo du dann vier Nutzer hast, wo du aber schon das erste Skript funktioniert.

Host: Das könnte wahrscheinlich jedes dieser Laptops.

F: Genau, genau, aber ich meine, das ist ja trotzdem wichtig, dass man irgendwo diesen Blick dafür hat, dass du wahrscheinlich eine Testmaschine brauchst, dann musst du diese Tests hoch importieren in eine Operativ. Genau, und dann

Host: Den VuGen-Skript, das ist ein Ordner, einfach Copy-Paste auf die andere Maschine und dann den Controller importieren lassen. Das ist gar kein Problem. Ich wüsste nicht, wo da ein Problem wäre. Getestet habe ich das natürlich noch nicht, dass ich hier irgendwie LoadRunner noch auf dem Laptop installiert und Skript zugeschickt habe und kann nicht schauen, ob ich das, das tun wir doch gleich auf, Portierbarkeit von Skripts zwischen Maschinen. Ich meine, es würde mich stark wundern, wenn es nicht geht.

F: Ja, ich glaube auch, dass es nicht schwierig ist, aber das wurde es jetzt noch nicht.

B: Wenn man es jetzt produktiv nehmen würde, würde natürlich auch dieses ganze Pricing-Thema eine Rolle spielen. Also ich weiß nicht, ob es auch teilt, wenn der Bachelorarbeit bei dir ist, aber so eine Produktiv-Umgebung bräuchte dann diesen Runner und diese Test-Umgebung und sowas.

Host: Und ansonsten gerne zu Allem-Brainstorming, was euch einfällt zu dem ganzen raus. Also auch wenn ihr sagt, ja 1, 2, 4 waren vielleicht zu kleine Schritte, das kann man zukünftig weglassen. Man geht direkt auf 1, 8, 64 und eskalierung hoch oder sonst irgendwas. Oder meint ihr, das macht auch schon Sinn?

E: Ich glaube, da muss man den Use-Case unterscheiden. Also wie jetzt da B zum Beispiel gesagt hat mit den Mehr Case Studies, wäre halt zum Beispiel der Use-Case, wenn wir sagen, okay, wir wollen testen, wie in Anführungszeichen gut die Maschine oder die LPAR läuft. Wenn wir jetzt den Use-Case nehmen, wir wollen unsere Ursprungssysteme mit zum Beispiel einem Container vergleichen, dann hast du natürlich ganz andere Voraussetzungen. Dann musst du darauf achten, was für SAP-Versionen habe ich, welche Maschinen habe ich. Dass du halt da die Messwerte möglichst vergleichbar machst und nicht sagst, ich messe jetzt auf einer Power 9 mit einem 1809-SAP und einer Stein-Alten-Case-Studie und jetzt switche ich plötzlich auf Container auf einer Power 10 mit der neuesten Version. Also da muss man, glaube ich, unterscheiden, was dann der Use-Case ist von der eigentlichen Messung. Genauso mit dem Umfang der Messungen musst du dann natürlich auch ganz neu beurteilen. Also klar, wenn ich jetzt richtig evaluieren will, ob mein System gut ist oder nicht, sind drei Messungen natürlich nicht repräsentativ. Aber grundsätzlich zeigt deine Arbeit, dass das das richtige Tool ist dafür und das ist für mich die Erkenntnis schlechthin, dass der Test-PC gut ausgewählt werden muss.

Host: Prinzipiell Thema Use-Case des Tests. Ich meine, der ganze Prozess des Developments des Tests und auch der Test selbst so bisher wurden ja eben auf diese Vergleichbarkeit ausgelegt. Wenn man wirklich versuchen möchte, die Maschinen krisensicher zu machen oder einen absoluten Stresstest durchzuführen, braucht man natürlich zusätzlich auch noch extensives Hardware-Monitoring auf den LPARS, also was ist die Auslastung, zu welcher Sekunde, mit wieviel Prozent, CPU-Zeit, Rabennutzung etc. Das ist ja bisher auch alles noch gar nicht integriert. Aber es ist natürlich möglich, dann diesen Test praktisch zu expandieren, eben auch auf diese anderen Use-Cases, indem man noch was hinzudichtet, würde ich sagen.

B: Kann man im Script Screenshots machen von dem aktuellen Bild auf dem Browser?

Host: Wo genau meinst du, beim Replay im Development, wo die zwei Fenster offen haben?

B: Also automatisch Screenshots machen.

Host: Achso, vom Script Screenshots machen lassen.

F: Also der statt quasi einem Klick, meinst du, ein Skinshot bauen?

Host: Interessante Frage. Ich weiß es nicht.

B: So könnte man theoretisch die Erstellung von Case Studies wesentlich vereinfachen.

Host: Über JavaScript. Wenn man mit JavaScript sich Screenshots machen lassen kann, dann geht es auf jeden Fall.

F: Aber gut, das ist ja jetzt wirklich spannend. Aber du musst ja trotzdem einmal durchklicken.

B: Ja, aber du kannst einmal einen Script machen und wenn sich wieder die BTP-Version ändert, wo sich die Farben anders sind und die Studenten davon überfordert sind, dann kannst du einfach als Script durchlaufen, weil das wäre Script ist aber nicht überfordert, wenn sie die Farben ändern.

Host: Sollte es nicht sein.

E: Aber wenn sich der Button ändert, schon.

Host: Also es kommt drauf an, mit was man identifiziert, wenn sich die Farben ändern, aber es bleibt, wenn du so XPath's hast, an der gleichen Stelle im HTML-Code. Dann kein Problem. Wenn du Descriptor hast und aus irgendeinem Grund ist die Farbe des Buttons ein Teil des Descriptors, könnte es Probleme geben. Aber ansonsten sollte es die, also am stabilsten in meiner Erfahrung so bisher,

auch am aufwendigsten, aber am stabilsten ist für alles den XPath's raussuchen. F12 drücken und 50 Millionen Dropdowns öffnen und den X-Plates rauswenden.

B: Du kannst auch mit rechtsklicken untersuchen und den Browser direkt auf den XPath springen.

Host: Das hat teilweise bei mir nicht funktioniert.

F: Aber das ist schon interessant. Also ich weiß nicht, das geht wahrscheinlich ein bisschen von dem Scope von der Bachelorarbeit weg, aber damit könntest du, aber trotzdem ziemlich cool, Damit könntest du zum Beispiel, das kannst du mit einem User machen, dass heißt, du fährst da nicht in diese 16-User-Hürde. Du könntest damit auch quasi sagen, eine neue BTP-Version ist draußen und dann könntest du halt dir automatisch mal schauen, ich mach meinen Skript, im besten Fall mit Screenshots und guck einfach mal, ob es so durchläuft. Und wenn dein Skript bei der alten BTP-Version funktioniert hat und bei der neuen nicht, dann weißt du, da ist jetzt irgendwas anders. Wenn es durchläuft, kannst du die Screenshots einfach angucken und sagen, okay, hier sind meine neuen Screenshots.

B: Genau. Und jetzt können wir ganz weit spinnen, also weit, wo die Bachelor-Arbeit hinaus ist. Wir können das Case Study nehmen. Mit einem AI-Tool versuchen, dort draußen den Skript zu bauen. Wie meinst du, das habe ich nicht verstanden. Das ist die Case Study, wo er drin steht, natürlich im Text, was du machen musst. Und baut daraus in den Script für LoadRunner, lässt das durchlaufen und wir haben das Ganze.

F: Dass du gar nicht mehr aufklickst.

B: Aber das ist ein ganz normaler Use-Case. Aber ich meine, wenn du das so präsentiert hast, das könnten wir auch probieren.

F: Das sind quasi Input-Daten, damit hast du schon mit den bestehenden.

E: Witziger Outlook auf jeden Fall.

F: Das ist witzig. Das ist cool.

Host: Ansonsten würde ihr sagen, dass sonst alles für den Prozess und den Test an sich so erst mal passt? Dass es ein solides Artefakt ist?

B: Ich denke schon. Ich denke, wir müssen skalieren. Mehr Messungen, mehr Nutzer und mehr Case Studies. Und wenn das alles hinhaut, so wie du es jetzt hier skizziert hast, dann könnte das eine ganz tolle Ergänzung sein. Also vom Prozess her denke ich, passt das.

Host: Sehr cool.

F: [unverständlich] für verrückte Dinge, deswegen ein. Ich denke auch, wenn z.B. theoretisch in dem UCC-Kontext ein Kunde sagt, ich habe ein Fallstudienproblem, dann könntest du dem das als Self-Service begeben. Und es dann einfach mal für ihn bis dahin klicken lassen. Und dann quasi so schauen, ob es durchläuft. Oder ob er Recht hat und es dann wirklich abbricht. Genau, so ein Debugging-Tool für einen Kunden, der sagt, euer Fallstudium funktioniert nicht. Könntest du da natürlich auch einfach mal sagen, wir haben hier die Lösung von dem Fallstudium, lässt es mal laufen.

Host: Also, und schick uns das Fehlerprotokoll.

F: Genau, wenn es wirklich einen Fehler hat. Oder ob du halt irgendwas falsch gemacht hast.

E: Beziehungsweise kannst du es als Support-Manager dann fix selber machen. Dann änderst du einfach die Maschine quasi in deinem LoadRunner, auf der es ausgeführt wird. Dann klickst du auf Ausführen und dann führst du das Skript einfach auf der Maschine als Support-Manager aus. Das stimmt, ist auch eine geile Idee.

Host: Das Script, also das ich jetzt hatte für SD, nur um vielleicht für euch einen Scope für die Zeit, hat ungefähr 19 bis 21 Minuten Laufzeit gehabt.

B: Okay, das ist in Ordnung, denke ich.

Host: Einfach nur als Info.

B: Ich meine, wir antworten eh nicht schneller als 19 Minuten auf dem Ticket.

Host: Ich meine ja, also einfach nur als Info, dass selbst wenn man jetzt zum Beispiel mehr Case Studies noch einbringt, ich weiß nicht, wie groß die sind jeweils. Aber ich glaube, ihr meintet ja, SD sei so eher das Standard, die Standard-Case Studies, die so am meisten verwendet werden.

B, X: Ja, SD und MM.

Host: Okay.

F: Ja, das ist vielleicht auch gut, dass man sagt, wenn man skaliert, dann fährt man halt bei denen an, wo jetzt die Applikationsexperten auch sagen, dass die werden irgendwie öfter genutzt.

Host: Genau, passt dann. Dann Skalieren kann man festhalten, man kann sagen, okay, mehr als drei Messungen um noch mehr, die temporäre Effekte zukünftig ausgeht, aussondern zu können. Mehr Nutzer, mehr Case Studies, also eben MM vielleicht dann noch mal runterskripten. Ich werde eine klare Dokumentation unter Probleme erstellen, die beim Scripting. Ihr werdet sowieso eine Einleitung bekommen, eins zu eins, wie man das Ding aufbaut.

F: Das ist wichtig.

Host: Ausführung sollte nicht punktgenau gleich stattfinden. Das ist auch ein sehr gutes Feedback, weil da war ich auch in einem innerlichen Hin und Her gerissen sein die ganze Zeit. Okay, was ist jetzt das Realistische? Testet das System nur, wenn sie gleichzeitig drücken, oder reicht es schon nur, dass sie aktiv sind? Im Endeffekt sind alle jetzt nicht künstlich von mir auseinandergebracht worden. Also bei zwei Nutzern ist es schon meistens relativ parallel gelaufen. Man kann aber mit VuGen, oder besser gesagt mit Controller auch einstellen, dass zum Beispiel nur alle 10 Sekunden ein neuer Nutzer startet, wodurch man automatisch dann eine Zeitdifferenz drin hätte, um dieses, wenn ihr mal wirklich große Tests macht, 128 Nutzer, 256 Nutzer, um zu verhindern, dass die zufällig 256 den gleichen Button drücken, oder das System in Error stürzen. Einfach einstellen, Ramp-Up-Zeit, alle 12 Nutzer, alle 20 Sekunden, oder irgendwie sowas. Genau, Portierbarkeit von Scripts zwischen Maschinen kann ich versuchen zu überprüfen. Ich schaue, wie schwierig es ist, LoadRunner jetzt hier nochmal einzustellen. Referenzen festhalten. Referenzmaschinen meine ich damit, also die genaue Doku, welche LPARs habe ich euch schon angefragt. Ich werde auch festhalten, was die Specs von meiner Maschine sind. Was war die Testmaschine, die es nicht geschafft hat, mehr als 16 Nutzer zu simulieren. Mein PC ist echt gut. Nutzung zur Erstellung von ja, genau, dann noch eben als Ausblick, Überlegung. Man kann es nutzen, um Bilder für die Case Study zu erstellen, wenn Screenshots erstellt werden können, aber JavaScript kann glaube ich schon Screenshots erstellen. Also eine JavaScript-Code kann man ChatGPT fragen. Man kann, es muss reskaliert werden, und man kann es als Debugging-Tool benutzen, wenn jemand sagt, mit der Case Study stimmt etwas nicht. Und wenn es ansonsten nichts mehr gibt von jemandem?

B: Nö.

Host: Dann sage ich an dieser Stelle, vielen Dank für die Teilnahme. Vielen Dank.

[Gruppe]: Gerne.