

## **A 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 Metric or Metrics and Requirements for the Creation of standardized SAP S4/HANA Performance Test:**

**Host:** Gibt es hier von irgendjemandem noch Fragen oder jetzt schon irgendetwas, was gerne geteilt werden wollen würde?

**A:** Also ich hätte eine Frage. Die Arbeiten von Jehle, Bögelsack und Co., die sind ja jetzt schon ein paar Jährchen her. Die sind ja 10 Jahre oder noch älter.

**Host:** 9, 11 und 12 glaube ich.

**A:** Genau, also so ungefähr 10 Jahre ungefähr so. Hast du dich mit der Frage beschäftigt, inwieweit sich vielleicht deren Erkenntnisse, also wie weit die vielleicht überholt sind oder inwieweit vielleicht andere Literatur zu dem Thema auch noch relevant sein könnte? Weil also seit 10 Jahren hat sich auch bei SAP viel geändert. Das waren damals noch R3-ECC-Systeme, heute haben wir S4HANA 2022 mit einer HANA-Datenbank. Ich weiß nicht, ob die damals auch schon auf einer HANA-Datenbank waren.

**Host:** Nein

**A:** Die haben noch ECC auf einer DB2 gemacht. Genau, also das heißt, ähm genau, hast du dich damit beschäftigt, inwieweit das Unterschiede macht.

**Host:** Noch nicht direkt, das ist auch etwas, das man gerne hier besprechen kann beziehungsweise ausdiskutieren. Ich habe nach anderer Literatur gesucht, ich habe relativ, überraschend wenig zu explizit SAP-Stresstests gefunden, hauptsächlich zu Datenbanken-Stresstests direkt, was allerdings natürlich komplett die Application-Layer außer Acht lassen würde bei den Tests. Auch etwas, was die drei Doktoren angesprochen haben, dass man eben bei Reinen Dat..., dass die Application-Layer durchaus großen Einfluss auf die Performance im SAP-System hat und man nicht nur die Datenbank testen darf. Ich habe noch theoretisch gesehen, die Literatur ausstehen, die ich noch lesen kann, um mir Gedanken zu fassen, aber es geht ja darum, praktisch ersten Input aus diesen Arbeiten der Doktoren zu sammeln, also das war der Ansatz, diesen hier einzubringen und dann eben mit der Expertenmeinung vor Ort herauszufiltern, was machen wir jetzt diesmal besser oder anders.

**A:** Jetzt ist das Problem, dass keiner von uns, der hier anwesend ist, glaube ich, gilt, also ihr dürft mich korrigieren, aber ich glaube keiner, der hier anwesend ist, hat die drei Arbeiten gelesen.

**Host:** Das stimmt, würde ich mir schon durchaus vorstellen.

**A:** Genau, deswegen kannst du, also ich erlebe es gerade so, du kommst hin, sagst ich hab die Arbeiten gelesen und mich schon hier mit dem Thema beschäftigt und jetzt möchte ich gerne euer

Expertenwissen haben. Ich habe immer noch nicht ganz verstanden, welches Expertenwissen du von uns haben möchtest.

**Host:** Input zum Thema, wie kann man solche Tests oder wie soll, was für Requirements sollte man bei der Erstellung solcher Tests beachten?

**A:** Okay, also aus SAP-Nutzerspektive.

**Host:** Aus Menschen, aus der Perspektive mit Leuten, die jeden Tag mit SAP arbeiten oder allgemein durchaus auch sich mit SAP-Performance auseinandersetzen. Diese Requirements müssen ja nicht für super, also da werde ich Beispiele, ein Beispiel für ein Requirement, ist zum Beispiel Reproduzierbarkeit. Ich denke, sollten normalerweise die meisten zustimmen, ja es macht Sinn, dass Tests reproduzierbar sind und das ist ein Requirement, um solche Tests aufzubauen. Ich würde halt die Requirements, die ich herausfiltern konnte, aus diesen drei Arbeiten hier dann vorstellen der Reihe nach und gerne Diskussionen und Meinungen an so, okay dass ergibt Sinn oder nee, das müsste man ändern oder das ist unnötig oder hey, mir fällt jetzt da noch etwas ein, was ich denke wichtig wäre.

**A:** Genau, das ist gut. Also es ist gut, wenn du sagst, ich habe was rausgefiltert, ich würde gerne auch Meinungen zu hören, weil ich glaube, wenn du jetzt einfach das öffnest und sagst, das sind eure Vorstellungen von Requirements, dann wirst du glaube ich viel bekommen, was dir nicht weiterhilft, weil viele von uns hier in der Anwenderperspektive sind und die können halt sagen, ja ich hätte gerne eine flüssige Performance der Fiori Oberfläche oder ich hätte gerne eine flüssige Performance der SubGui Verbindung. Das Problem ist aber das, also diese flüssige Verbindung der SubGui in harte KPIs umzusetzen, ist sau schwer, weil das an 20 Sachen liegen kann und einer der großen Probleme, zum Beispiel bei uns häufig die Internetverbindung ist. Zum Beispiel viele unserer Kunden sagen, sie haben Performance Probleme und in der Wirklichkeit ist ihr Internet so schlecht. Und das war für mich als [unverständlich 04:12] wichtig. Cool, dann freue ich mich.

**E:** Also um das vorwegzugreifen war von uns die Vorgabe, dass die Tests entweder über eine VM von der RBG gemacht werden oder lokal von einem Computer hier, das werden wir dann halt auf die Arbeit aufbauen, eben genau diese Problematik ausschließen können, dass wir sagen können, okay wir haben ein Offset, aber der Offset ist bei allen Messungen gleich und somit können wir diesen Offset da mehr oder weniger vernachlässigen.

**A:** Sehr gut.

**Host:** Sonst irgendwelche Fragen oder Gedanken?

**Host:** Genau, dann würde ich als erstes gerne auf die KPI eingehen bzw. die KPIs. Die Arbeiten, die drei davor genannten Arbeiten, haben sich ausschließlich entweder auf Throughput oder Response Time konzentriert, um entsprechend die Performance zu messen sozusagen. Wobei allerdings zweimal ging es auf Response Time und zwar von Jehle und Gradl, Bögelsack war der einzige der auf Throughput gegangen ist. Die Begründung von Jehle und Gradl war, dass eben Throughput eher transaktionslastig sei und damit für das Testen von Datenbanken und deren Performance geeigneter, aber sobald es um eine Applikation gibt, wo Anwendereingaben stattfinden und Anwenderinteraktionen, Response Time wichtiger sei, vor allem auch aus einem unternehmerischen Kontext, du hast selber gesagt, ihr seid ja auch Anwender, ist auch Response Time für die User Experience deutlich angenehmer und interessanter. Und da würde mich jetzt eben eure Meinung interessieren, ok, ja, das klingt ja ganz sinnvoll, was Gradl und Jehle gesagt haben, Response Time ist sinnvoll oder ne Throughput ist das Wahre oder eben mach beides. Also das wäre da jetzt eure Einschätzung zu der KPI, welche man berücksichtigen sollte.

**B:** Was ist der Unterschied?

**Host:** Response Time ist wie schnell es, wie lange es dauert von Funktionsstart eines Anwenders bis er eine Antwort bekommt und Throughput ist theoretisch gesehen wie viele Transaktionen gleichzeitig pro gegebener Zeiteinheit durchgeführt werden können. Hängen indirekt zusammen, aber auch nicht unbedingt. Aus einer Userperspektive wäre Response Time, was A angesprochen hatte bezüglich flüssiges Fiori, wenn man klickt und man hat einen Response Time von einer Millisekunde, dann sollte alles sehr, sehr flüssig wirken.

**A:** Genau, also ich sage es mal so, die Frage, die du stellst, ist, was ist besser und was soll ich beide machen und die kurze Antwort ist, beides ist immer besser, weil beide KPIs haben ihre Berechtigungen. Ich glaube, das worüber die Kunden am meisten maulen, ist die Response Zeit. Das ist so das Ding. Throughput ist aber bei manchen unserer Systeme auch auf jeden Fall relevant, vor allem bei dem, wobei wir haben so Großkunden, die in Asien sitzen, wo tausende Leute gleichzeitig auf ein SAP-System zugreifen. Da haben wir auch ein Throughput-Thema. Also von daher, wenn du sagst, ich kann nur eins schon mal machen, würde ich dir empfehlen, mach die Response Zeit. Aber die Frage ist, was hätte ich davon beides zu machen, wenn du sagst, das ist ein riesen Aufwand, das Projekt wird dadurch riesengroß. Dann würde ich sagen, mach nur eins und mach Response Zeit. Wenn du sagst, das kann man eigentlich total easy machen, würde ich beides machen.

**Host:** Weitere Meinungen?

**C:** Ich würde dem A in gewisser Weise zustimmen, dass beide simpler machen, weil ich finde, die Zwei KPIs hängen halt irgendwie zusammen, aber ich finde, die sind trotzdem sehr unterschiedlich und zeigen jeweils einen anderen Aspekt. Also ich hätte jetzt keine Begründung, einen wegzulassen.

**Host:** Okay, die Begründung von Jehle und Gradl war eben hauptsächlich, dass der Throughput nicht wirklich die Gesamtleistung des SAP-Systems abbilden würde, sondern eben nur der unterliegende Datenbank im Endeffekt, weil die User-Interaktionen und die Application eher zum Teil nicht wirklich da kreditiert werden. Bei der KPI, was haltet ihr von dieser Aussage?

**D:** [unverständlich]

**Host:** Ich glaube, das hängt von dem Tool ab. Jehle hatte ein eigenes Tool programmiert für die Datenerhebung, und das war in der Application selbst integriert. Ich glaube, die hat dann halt nur an der Datenbankstelle gemessen, aber es hieß, dass vor allem die Datenbank es ist, was Throughput praktisch begrenzt und einschränkt.

**A:** Ja, aber das ist in Zeiten von HANA nicht mehr der Falle.

**Host:** Ist nicht mehr?

**A:** Weil die HANA so leistungsstark ist. Ich glaube, dass ist grad so ein Ding. Das ist, glaube ich, was ganz wichtiges, worüber du dir jetzt, bevor du anfängst zu messen, Gedanken machen solltest. Und zwar, bei SAP hast du so ein Problem, und zwar SAP ist ein riesenkomplizierter Stack, mit der aus ganz vielen Diensten besteht, die alle miteinander kommunizieren. Dadurch ist es total schwer, so reduktionistisch hinzugehen und zu sagen, ja, dieser eine Parameter ist es, der dazu führt, dass das System entläuft. So nach dem Motto. Sondern das ist immer ein ganz großes Zusammenspiel, und es ist manchmal auch auf den ersten Blick auch oft schwer zu erkennen, was jetzt eigentlich der Grund dafür ist, dass eine bestimmte Verhaltung zu beobachten ist. Das heißt, SAP-Nutzung und das, was du in der SAP-Nutzung erlebst, und das, was du richtig an richtigen Hardware KPIs messen kannst.

IOPS, Throughput, CPU-Auslastung, was es da alles so gibt, das ist so ein bisschen disjunkt, das passt nicht so gut zusammen. Und dadurch ist jede SAP-Performancearbeit immer so ein bisschen an dem Problem, dass sie entweder sich viel Gedanken über den Zusammenhang machen muss, aber wenn du das über den Zusammenhang, wenn du sagst, mein Thema ist der Zusammenhang, dann kannst du das wahrscheinlich mit mehr als zwei KPI's nicht schaffen. Also dann musst du dich wirklich sagen, okay, ich stürze mich auf diese KPI, und jetzt untersuche ich den Zusammenhang zwischen dem, was ich auf der Hardware messe und dem, was ich auf der Oberfläche erlebe. Und versuche da irgendwie die Beziehung klarer zu machen. Das ist zum Beispiel das, was die Annie auch in ihrer Arbeit eigentlich im Endeffekt macht. Oder du musst hingehen und sagen, ich messe einfach nur das User-Verhalten, dann ist es aber häufig schwer, daraus technische Rückschlüsse zu ziehen. Oder du sagst, ich messe technische Sachen, dann ist es aber schwer, von den technischen Werten auf die Performance, auf das User-Erlebnis Rückschlüsse zu machen. Also deswegen solltest du dir im Vorhinein fragen, was von den drei Dingen will ich machen. Mehr als eins von diesen drei Sachen wird, glaube ich, den Scope deiner Arbeit massiv reißen. Und je nachdem, welchen dieser drei Scopes du machst, solltest du eben auf andere Sachen achten. Wenn du sagst, ich will den Zusammenhang machen, dann machst du es nur für eine KPI, und alle anderen KPIs sind unwichtig. Wenn du sagst, ich will User-Performance machen, dann musst du dich um KPIs, dann musst du quasi, mit welchen KPIs kann ich die User-Performance messen? Das sind dann eben zum Beispiel sowas wie, dass es nicht nur um Hardware technisch messbar, weil du dann irgendwas auf der SubGUI machen musst, oder im FIORI Launchpad. Und das Dritte wäre eben, wenn du sagst, ich will eher auch so auf Hardware-Ebene messen, also wenn ich so throughput, oder ich schreibe Geschwindigkeiten der Datenbank, oder CPU-Auslastung oder sowas, dann ist da halt aber immer die Frage, was hat das jetzt mit dem User-Erlebnis zu tun? Und da musst du auch eben aufpassen, weil von uns wirst du untersch, wir haben unterschiedliche Perspektiven, und von uns wirst du unterschiedliche Sachen hören. Manche werden eher die Benutzer KPIs beschreiben, andere werden eher die technischen KPI's beschreiben. Und ich glaube, bei deinem Thema hast du die Gefahr, so zwischen den Stühlen zu sitzen, das und das will ich machen, aber das ist zu viel.

**B:** Also wenn ich dich jetzt richtig verstanden habe, dann willst du nicht, also wir setzen das Vorsystem auf, und wollen das dann irgendwie messen, ob das richtig aufgesetzt wurde?

**Host:** Nicht, ob es richtig aufgesetzt wurde. Wir wollen den aktuellen Status Quo messen, d.h. wir werden dann Load mit Loadrunner darauf generieren, entsprechend dann sammeln, was Loadrunner sammeln kann, da werden viele Werte mit dabei sein. Und es geht darum, nicht die aktuellen Performance oder Menge oder was auch immer festzustellen, es geht nur darum, ein Testsystem aufzubauen, das dann, wenn eben neue Technologien vorgestellt werden sollten, wie Container oder Power-11, 12, 14, 50 Racks, man dann schauen kann, so wir haben diesen einen Test X, den haben wir auf Version A angewendet, da ist dann ein Response Time, oder was weiß ich, 2 Millisekunden rausgekommen, jetzt haben wir die Container Introduced mit genau der gleichen Installation, und der gleiche Test haut jetzt aber eine Response Time von 60 Millisekunden raus, also haben wir hier eine massive Verlangsamung, dann kann man schauen, was ist der Grund, warum ist das.

**B:** Und die KPIs sollen User, also aus User Sicht sein?

**Host:** Aktuell ist die Idee, dass man auf die User-Perspektive geht.

**B:** Okay, dann müssen wir, das sind beide wichtige [unverständlich 13:25], wie ich vorhin gesagt hab, Respons Time, ich würde es mal so verstehen, dass wenn du eine Anfrage schickst, und wartest bis es zurückkommt, und der Throughput, ist es dann eher so, wenn du ganz viele Anfragen schickst und schaust wie viele davon bearbeitet werden zu gegebener Zeit.

**Host:** Genau

**B:** dann musst du beides auf User-Ebene, also im UI testen, dass es, dann kannst du ja nicht auf Daten mal gehen, dann machst du beide KPI. Aber die Response Time ist wahrscheinlich die erst mal wichtigere, weil wir haben, wie A gesagt hat, nur wenig, was wirklich ein Toolkit gibt.

**E:** Die Ursprungsintentionen, um das mal kurz für das Team zusammenzufassen, war, dass der Alex Läubel dieses Paper vom Johannes gerne halt getestet haben wollte.

Ich weiß nicht, wie er es genau gesagt hat, aber da Leistung, wie leistungsfähig es halt ist.

Jetzt kannst du natürlich einfach die Container testen, dann hast du random einfach irgendwelche Werte. Ja, schön. Um das halt vergleichbar zu machen, wird jetzt die Basis gesetzt. Okay, wir haben jetzt einfach ein System, das ist einfach oldschool, nenne ich es mal, klassisch aufgesetzt, VM, SAP-System, Pipapo. Wie ist denn da jetzt eigentlich die User-Performance? Wie schnell ist das? Und was für Daten kriegen wir? Und das soll dann dafür die Basis darstellen.

**Host:** Der Fokus ist mehr auf den Prozess als die tatsächlichen Ergebnisse oder der unterliegenden [unverständlich].

**A:** Ich finde, es hat immer noch nicht ganz die Frage beantwortet, ob du jetzt eher auf die technische oder auf die, ich nenne es mal, Erleb-Metriken stürzen willst.

**Host:** Tendenz bisher, aber das ist noch nicht in Stein gemeißelt, dass die Erlebmeterik.

**A:** Okay

**Host:** Wie können die User das wahrnehmen.

**A:** Das ist gut. Dass ist sinnvoll, aber dann würde ich sagen, geh auf jeden Fall auf die Response-Zeiten. Weil der Throughput ist viel weniger erlebbar als die Response-Zeit.

**Host:** Und eben mit HANA scheint da auch weniger ein Issue oder weniger ein Bottleneck.

**A:** Ja, deswegen kam die HANA-Datenbank ja. Weil man ja noch gemerkt hat, vor allem für große SAP-Systeme sind die damals ja sogar noch HDD-Festplatten basierten Datenbanken einfach nicht schnell genug, um die Anfragen abzuarbeiten. Das war halt damals das große Ding. Und dann kam ja die HANA-Datenbank, die gesagt hat, hey, komm, dann schreiben wir die Daten in den RAM rein, der ist ja viel, viel, viel, viel schneller als die HDD. Und damit hat man so ein bisschen dieses Problem gelöst. Es gibt da noch ein Log-Problem, also das heißt quasi auf HDD zu schreiben. Für die HANA-Datenbank ist es immer noch nicht so schnell, die HANA auf SSD zu betreiben, weil die HANA so viel Logs schreibt. Aber, genau, also dieses ganz klassische Throughput-Problem, das die alten DB2-Datenbanken auf HDD-Festplatten hatten, das gibt es so in der HANA nicht mehr.

**Host:** Okay, dann auf jeden Fall vielen Dank für den Input.

**D:** Noch eine kurze Frage. Wenn du das Erlebnis messen willst, hast du dir irgendwelche Gedanken gemacht, wie man das Erlebnis für den Nutzer in der Fiori UI messen kann?

**B:** Respons time

**D:** Ja, aber wo ist denn die Response-Time? Bis der Server es an den Client gesendet hat, oder bis zur die UI.

**Host:** Anfrage bis Ergebnis. Ich möchte Transaktionen öffnen, ist die Response-Time, bis mir die Transaktion angezeigt wird.

**D:** In der Fiori UI?

**Host:** Ja. Das heißt, theoretisch gesehen, IO bis zum Server, Verarbeitung am Server, IO zurück.

**D:** Und dann wärst du die Person, die das dann auch selber manuell misst?

**Host:** Es wird automatisiert über Loadrunner. Soweit die Idee.

**D:** Und der misst dann, wann das in der UI komplett angezeigt wird.

**Host:** Ja.

**E:** Das kannst du so ein bisschen mit diesen Google-Webseiten-Tests vergleichen.

**D:** Das ist wie so eine UI-Click-Tool.

**(E, Host):** Ja, genau.

**Host:** Man könnte es auch hard coden, aber ich würde es wahrscheinlich mit UI-Click-Skripten lieber. Gut. Dann... Vielen Dank schonmal für den Input bisher. Dann würde ich zu den Requirements kommen, die so weit von den drei Doktoren vorgeschlagen wurden, beziehungsweise hauptsächlich beachtet wurden. Und da würde ich mich dann eure Meinung so weit mit den Batches vorstellen, also vielleicht öffnen für Meinungen, nach Motto, ja, klingt sinnvoll, oder eben, boah, ist vielleicht to much, oder da fehlt definitiv noch was. Und zwar das eine der Requirements, die von Jehle und Gradl hauptsächlich genannt wurden, war die von Reproduzierbarkeit, was ich persönlich finde, schon sehr Sinn ergibt, wenn man einen Test auf verschiedenen Konfigurationen ausführen möchte.

**B:** Soll ich dazu was sagen?

**Host:** Gerne, bitte.

**B:** Ja, auf jeden Fall. Es ist, glaube ich, nichts wie... Es ist sehr wichtig.

**C:** Ich müsste nicht was dagegensprechen sollte. Sehr wichtig, ja.

**Host:** Danke. Dann die nächsten zwei, die Sinn ergeben, wo ich finde, dass man für unsere oder meine, also E's, F's und meine Anwendung, eins von beiden auf jeden Fall auch, ist entweder Referenz Hardware oder Referenz Installation, weil wenn man jetzt den Test durchführt auf einer anderen Version von SAP und auf einer anderen Hardware, ist das, denke ich, recht schwer vergleichbar dann. Wenn man allerdings halt eine Experiment-Variable, also entweder Hardware oder Version, ändert, denke ich, wäre das wieder sinnvoll, oder übersehe ich da etwas und man sollte definitiv beides beachten?

**B:** Also kannst beim Experiment sowieso nur eine Variable [unverständlich] ändern. Das Andere sollte gleich bleiben, sonst kannst [unverständlich]. Also mit Installation meinst du jetzt die Software-Version.

**Host:** Ja. Also in dem Sinne von S/4HANA und nicht, dass ich dann das nicht persönlich... Man sollte jetzt nicht auf den aktuellen VMs S4HANA testen und dann später auf den Containern R3. Und das dann versuchen zu vergleichen.

**B:** Auf jeden Fall auch die gleiche Version von S4HANA, die gleiche Version der verschiedenen Komponenten innerhalb von S4HANA und also beide so gleich wie Möglichkeiten.

**Host:** Genau. Also das ist praktisch Referenz, so war das diskutiert in den Thesis', dass man sagen muss, okay, das muss immer eben... Ceteris Paribus, alles andere gleich bleiben.

**B:** Ja.

**Host:** Dann...

**E:** Oder du machst halt Zwischenmessungen, dass du halt sagst, es ist wieder ein bisschen Zeit vergangen und es sind jetzt halt schon zwei neue Versionen von SAP raus. Okay, dann muss ich halt in den sauren Apfel beißen auf der Anführungszeichen alten Hardware halt erstmal die neue Version installieren. Ändere damit diese eine Variable, was ja B gesagt hat, mach ne Messung, schau mir das an und dann kann ich halt auf die neue Hardware mit der neuen Version gehen. Oder umgedreht ich mach die alte Version auf die neue Hardware. Oder man macht beides je nachdem wie wichtig einem das dann erscheint oder was die ersten Messergebnisse sind.

**B:** Ja.

**Host:** Dann... Die Verbindung wird er aufgebaut, spannend. Wird noch aufgezeichnet, oder?

**E:** Ja.

**Host:** Das nächste, da würde ich ein bisschen noch was dazu erklären. Die requirement war störungsfrei. Das haben die Doktor Jehle dazu argumentiert, dass es verschiedene Arten von Funktionen gibt bei der Software. Er hatte nicht SAP ERP getestet, sondern SAP Portal, ein SAP Portalsystem, [unverständlich]. Aber das trifft auch auf ERP zu meines Wissensstandes nach. Und zwar gab es innere und integrierte Funktionen. Integrierte wäre eben, wenn man ein Excel spreadsheet in SAP zum Beispiel hoch lädt als Anhang, oder es auch in SAP bearbeiten kann. Die Funktion stellt aber im Endeffekt Excel dann zur Verfügung. Während innere Funktionen, die von der Software selbst ohne Fremdkomponenten sind. Er hat argumentiert, dass man ausschließlich für eine wirkliche Performance-Darstellung ausschließlich innere Funktionen testen sollte und keine integrierten. Genau, das meinte er mit störungsfrei und auch nur Funktionen, die weitestgehend lokal verarbeitbar sind. Das war die Aussage, die Argumentation mit störungsfrei sollten die Tests sein. Denkt ihr soweit...

**B:** [unverständlich]

**Host:** Dann automatisch und rekonfigurierbar und modular, wäre noch ein Batch an drei requirements. [unverständlich]

**A:** soll dein Test-Tool sein oder soll SAP sein?

**Host:** Soll der Test-Prozess sein. So, da rekonfigurierbar, wie ich ja vorhin gesagt habe, im weiteren Vorgehen, ist ja auch die Idee, dass man nochmal eine Fokusgruppe hält mit einem Review und Feedback zum gesamten Prozess. Wenn jetzt ein Feedback wäre, ok, du hast 20 Tests zu wenig ausgeführt, damit das ein repräsentatives Ergebnis darstellt, dann wäre natürlich rekonfigurierbar praktisch und auch modular, dass man dann weitere Module an Tests hinzufügen kann in den Prozess. Der nächste Punkt wäre auswertbar. Ich wollte es einfach der Formel halber mit aufnehmen.

**B:** Aber das könnte tatsächlich spannend werden. Das könnte [unverständlich]. Also was ist dein Zieldatenformat oder was sind deine Ziel Dinge, die du zum Schluss hast ob du jetzt eine Response-Time hast oder? Gibts wahrscheinlich verschiedene Datenformat, gibts verschiedene Darstellungen von Response-Time. Und dann kommt noch an welches Tool du zur Auswertung verwenden wirst. Und dass man dann zusammenfasst.

**Host:** Das ist ein sehr guter Input, aber tatsächlich war beim Auswertbar in den Doktorarbeiten wirklich nur damit beschrieben, dass man einen quantifizierbaren Wert zurückbekommt. Also, dass es eine Zahl gibt. Aber trotzdem auch ein guter Input. Danke, dafür sind wir hier. Hast du den? Absolut recht. Das nächste ist skalierbar. Hauptsächlich von Bögelsack und Gradl. Wenn man feststellen soll, mit einer 10-Nutzer-Simulation kommt man nicht dazu, die WMs auszulasten. Sollten die Tests auch in der Lage sein, dann entsprechend auch 30, 40, 50 Nutzer ausgewertet werden zu können.

**B:** [unverständlich].

**Host:** Das ist auch ein bisschen ähnlich, finde ich, zu Modular. Die Auswerte, oder halt man die Tests modular bauen kann. Oder könnte man das gemeinsam betrachten?

**B:** Das kriegst du jetzt nicht automatisch, [viele unverständlich] wenn man es macht. Man kann es schlicht und schlecht hoch skalieren. Aber schon, ich habe es.

**Host:** Und der letzte Punkt, größere Punkt, den ich ausfiltern konnte, war ausschließend temporäre Effekte. Im Endeffekt, dass man einfach Tests mehrmals durchführt und ein arithmetisches Mittel bildet, um halt kurze Störungen, Frequenzabfälle in der CDU entsprechend ausfiltern zu können.

**B:** Auf jeden Fall. Das ist was der A gesagt hatte über SAP-Systems, riesen Zusammenspiel mit mehreren Services. Und du weißt gar nicht, was wann welcher Service wie läuft und warum. Oder könnte man schon wissen, aber es ist aufwendig es rauszufinden. Und da sind solche [unverständlich] Temporäre-Effekte zählen lassen. Und dann kannst du auch, am besten Wochen hinweg bei allem Tests durchführen, die das dann herausfiltern. Und dann auch zu verschiedenen Tageszeiten und zu verschiedenen Wochentagen. Genau. Das ist halt die perfekte [unverständlich].

**Host:** Perfekt. Also nachher, das ist praktisch die Liste an basic, die auf meine Tests übertragbaren Requirements, die ich aus den drei Doktorarbeiten ausfiltern konnte. Manche wirklich sehr basic. Jetzt wird mich interessieren, jetzt mit dieser Vorlage, mit diesen Beispielen, was Requirements für Tests sind, würden euch noch irgendwelche Sachen einfallen, wo ihr sagt, ok, darauf muss man auch achten. Das ist wichtig.

**Host:** Gerne offene Diskussion, Brainstorming. Also wenn da irgendetwas ist, wo ihr denkt, ist das überhaupt sinnvoll, soll ich es jetzt sagen. Gerne raus.

**B:** Also ich denke, du kannst davon profitieren, wenn noch bisschen weiter in die Literatur reinschaust, weitere Experimente reinschaust, wie man sowas macht. Ich bin jetzt kein Experte, aber es gibt sehr, sehr umfangreiche Test-Aufbauten, die alles mögliche beachten. Könnte sein, dass da schon alles drin ist, aber könnte genau so gut sein, dass es nicht ist. Das hat nichts mit SAP, nichts mit HANA oder irgendwas zu tun. Sondern, dass sind halt Experimentesachen.

**A:** Ich denke, du solltest dich auf jeden Fall mit den Abschlussarbeiten von der Annie beschäftigen. Ich weiß nicht, wieviel das sind. Also wir haben eine Kollegin, die jetzt gerade nicht da ist, die macht bei uns im Endeffekt Strommessungen, also die schaut quasi inwieweit, führen bestimmte Programmabläufe in dem SAP Applikationsserver zu Stromverbrauch auf der Maschine darunter, mit der Idee, das vielleicht eines Tages in Green-Coding umzusetzen. Aber die hat sich viel mit solchen



Themen beschäftigt. Wie komme ich überhaupt von dem, was im Applikationsserver passiert, zu dem, was auf dem CPU passiert? Wie kann man das messen? Was kann man da so machen? Das ist nicht genau das, was du machst, aber das ist das, was am ehesten dem gleich kommt, was wir hier am UCC machen.

**Host:** Gut. Danke für den Input.

**A:** Was noch wichtig methodisch wäre, ich vermute, du machst Design Science Research, oder?

**Host:** Ja

**A:** Genau. Jetzt hast du ein Set an Requirements erarbeitet. Was ganz wichtig am Ende deiner Arbeit ist, dass du in deiner Evaluation darauf eingehst, ob du deine Requirements überhaupt erreicht hast. So.

**Host:** Okay.

**B:** Und dafür müssten sie definiert sein.

**A:** Genau. Also das heißt, am besten machst du die Science Science Research in deinem Fall so, dass du sagst, mein Tool so, tak, tak, tak, tak, tak, am besten 8 oder 5 oder was auch immer, am besten in der Liste oder der Tabelle. Und jetzt fange ich an zu bauen, dann baust du deinen Tool und am Ende sagst du, jetzt habe ich ein Tool, und dann guckst du, ist es wirklich, also misst es die Response Zeit oder habe ich das nicht geschafft? Ist es modularisierbar oder habe ich das nicht geschafft? Gibt es vielleicht auch etwas, was ich gelernt habe? Also habe ich zum Beispiel gemerkt, dieses Requirement ist eigentlich unschaffbar. Oder es ist total doof, das so zu fragen, man sollte die Frage eigentlich lieber so und so stellen. Das sind all die Sachen, die am Ende dann in der Arbeit methodisch passieren sollten.

**B:** Also ich meinte mit definiert sein, dass du kannst jetzt sagen, ist das jetzt Modular oder nicht, zum Beispiel oder automatisch oder nicht, du kannst jetzt sagen, was bedeutet modular, was bedeutet automatisch in deinem Fall. Was ist halt die Zielvorstellung deiner Tests. Was mir noch einfiel, was wichtig ist, [unverständlich] ist die Kontrolle der Variablen, das ist die Softwareversion und die Hardware drin. [unverständlich] so weitere gibt, die Variablen zu kontrollieren ist. Ich weiß nicht, inwiefern das alles eine Rolle spielt, ich meine eine Power 8 kann unterschiedlich, eine Power 10 kann unterschiedlich konfiguriert sein kann. Die Hardware-Konfiguration könnte eine Rolle spielen.

**A:** Die Arbeiten sind auf Power 7 geschrieben. Die Doktorarbeiten.

**B:** Ich meine, alles will [unverständlich]. Nimm dir die Zeit und schau an, was du alles für Variablen kontrollieren möchtest.

**C:** Deswegen vielleicht, darauf aufbauend ist einfach die Variable Hardware und Software zu generisch und zu oberflächlich. [unverständlich] gesagt, dass du es vielleicht runterbrechen musst, dass die Variable Software vielleicht aus zehn kleineren Variablen besteht. Wo es dann halt sinnvoll wäre, vielleicht nur eine dieser zehn kleineren zu ändern und nicht das Gesamtpaket.

**Host:** Vielen Dank.

**B:** [unverständlich] reproduzierbarkeit auf jeden Fall auch schauen, was damit ist... ...bei allen eigentlich...

**Host:** die Definition der... die klare, wissenschaftliche Definition der Requirements.

**B:** ...und am besten für jedes Requirements auch Tests definieren. Wie kann ich herausfinden, ob mein Ding modular ist? Oder Acceptance-Kriterien sowas in der Richtung. Also am besten herleiten, die Acceptance-Kriterien und dann dein Tool evaluieren. Aber es sieht echt schon gut aus.

**Host:** Gibt es noch... An sich waren es die letzten paar Minuten schon zu dem nächsten Punkt, weil meine nächste Frage wäre gewesen, gibt es noch weiteres Feedback oder allgemeine Gedanken? Gerne auch zu der Art und Weise, wie jetzt die Focus Group abgehalten ist. Gerne auch Feedback.

**B:** Ich meine, die Focus Group ist immer gefährlich, wenn man in der konzeptionellen Phase Feedback einholt [unverständlich] so gut vorstellen könnte. [unverständlich] Da hilft meistens Visuelle Unterstützung. Dein Bildschirm, wie ich hier sehe, aber die anderen sehen nicht. Vielleicht irgendwie aufbereitet. Ich meine, klar da ist auch die Gefahr, dass man das dann nicht zeigt. [unverständlich]

**Host:** [unverständlich]

**B:** Oder gruppieren und sowas. Und was man teilweise auch sieht, die könntest ganz gut gruppieren. So nach dem Motto du hast eins, was die Tests angeht, eins was Test Setup angeht. Du hast so viele [unverständlich] Anforderungen. Du hast sowas in der Richtung.

**Host:** Also dann doch mehr Visuelle Unterstützung auch für die gesamte Focus Group selbst, nicht nur für das Abholen. Okay.

**B:** Genau. Gruppieren hauptsächlich.

**E:** Versuch dich bei sowas an irgendeine Innovationsmethode anzulehnen. Also Brainstorming wirst du wahrscheinlich kennen. Mindmapping. Versuch da was zu finden, was auch dir taugt. Was du denkst, okay, das wäre dafür ideal. Diese Innovationsmethoden sind ja eh relativ offen gestaltet. Und dann sagst du, okay, jetzt alle Punkte, die jetzt genannt werden, die werden jetzt zum Beispiel an das Flipchart geschrieben. Es gibt ja auch diese digitalen Whiteboards. Dann machst du da zum Brainstorming. Das ist nämlich, B sagt, auch als Visuelle Unterstützung für die Focus-Gruppe wichtig. Weil dann sagt A plötzlich, wenn du nochmal den Gedanken von C liest, ey, da fällt mir doch noch ein. Dann kann man das gestalten. Und dann kannst du auch von Brainstorming auf Mindmapping wechseln. Wenn jetzt plötzlich einer sagt, die Begriffe kannst du ja zusammenfassen. Sowas ist sehr, sehr hilfreich. Es gibt jetzt Dutzende verschiedene Methoden. Aber sonst war es sehr gut.

**C:** Würde ich auch so sehen, dass eine visuelle Unterstützung hilfreich gewesen wäre.

**B:** Aber die Kriterien, die man jetzt rauskriegt, die Kriterien sind ja alles sehr abstrakte Kriterien und Requirements. Das ist halt nicht sowas, nichts davon ist SAP-spezifisch. Nichts davon ist nicht mal Software-spezifisch. Zum größten Teil. Zum großen Teil.

**Host:** Das sind die allgemeinen Requirements für den Testprozess sozusagen. Und

**B:** Ich denke, da wäre es vielleicht, gibt es eine Grundlagenliteratur dafür, was du verwenden kannst, um darauf aufzubauen. Ich meine, es ist gut, dass du unsere Feedback dazu hast und die Perspektiven die wir drauf haben. Aber ich glaube, du würdest schon davon profitieren, wenn du wirklich nicht die 3 Dissertationen von den 3 hast, sondern tatsächlich so ein Methodenwerk. Also zur Experimentdurchführung oder so in der Richtung. Vielleicht zitieren die sogar so ein Werk. Und darauf kannst du aufbauen. Und dann hast du eine lückenlose und sehr viel stabilere Argumentation, warum du das gemacht hast.

**Host:** Dann, wenn es keine weiteren Punkte gibt. Vielen, vielen Dank für die kurzfristige Teilnahme, eure Zeit und das tolle Feedback. Vielen Dank.