# Interview 4

**Speaker1:** [00:00:05] Hallo Interviewpartner, herzlichen Dank, dass du dich bereit erklärt hast für das Interview. Ist es für dich in Ordnung, wenn wir das Gespräch aufzeichnen?

**Speaker2:** [00:00:14] Ja, ist okay. Kein Problem.

**Speaker1:** [00:00:17] Ich habe hier einen sehr strukturierten Leitfaden vorbereitet. Das heißt, ich habe einige Fragen, die ich aber nicht in dieser Reihenfolge abhandeln muss, sondern ich kann das Gespräch dann abhängig vom Gesprächsverlauf auch auch individuell gestalten. Damit du ein bisschen einordnen kannst, um was es geht. Ich möchte die möglichen Herausforderungen bei der Low Code Softwareentwicklung durch den End User im Bezug auf die Scrum Methode untersuchen. Zu Beginn gerne einige Fragen zu deiner aktuellen Position. Kannst du mir beschreiben, was ist deine Berufsbezeichnung und wie deine Arbeit so aussieht?

**Speaker2:** [00:01:01] Ähm, ja. Offiziell bin ich Business Analyst, Business Engingeer und Test Manager. Alles ein bisschen. So eine ganz offizielle Beschreibung gibt es nicht. Aber das mache ich alles. Sagen wir es mal so, ich bin jetzt seit 14 Jahren bei der Unternehmen X, immer im Kollektiv Leben Umfeld. Die Bezeichnungen ändern. Das Thema bleibt das gleiche. Seit etwa drei Jahren beschäftige ich mich auch mit Robotics also RPA vor allem und hatte daher jetzt damit auch Kontakt und versuche es hier in der Unternehmen X ein bisschen zum, also in der Unternehmen X Schweiz ein bisschen vorsichtiger zum Fliegen zu bringen.

**Speaker1:** [00:01:46] Und dann konkret wann kammst du das erste Mal mit Low Code Software in Kontakt?

**Speaker2:** [00:01:52] Eben. Das war ca. vor drei Jahren, es gab da einen allerersten Proof of Konzept, das war noch ein Projekt, im Projekt war ich nicht dabei, aber als das Projekt dann zu Ende ging und die Diskussion dann stand, was macht man jetzt damit? Es gab dort damals auch einige Überlegungen, die man im Projekt gemacht hat. Also will man das Thema outsourcen einem Consultant, den du auch kennst. Der war damals auch schon im Boot und dann hat man Offerten eingeholt, das berechnet und

dann aber entschieden, dass das zu teuer ist für das, was man jetzt möchte. Und hat dann entschieden, dass man es zuerst selber versucht. Damals noch mit dem Produkt von Ui Path, dass man damit versucht, das mal in die Gänge zu bringen. Es gab dann einige Diskussionen, wo und wie man das ansiedelt. Nach langem Hin und Her, wurde es damals dann im sogenannten BEZ Team in der Unternehmen X Schweiz angehängt, kurz zur Erklärung. Es handelt sich dabei um ein Team, das nur kurzfristige Wetten umsetzt also für Sachen, die man ausprobieren möchte. Und bevor man jetzt zum Beispiel ein Projekt startet oder es in den Betrieb übergibt, ob das überhaupt funktioniert. Und dort wurde dann das Thema angesiedelt. Man hat dann damit ein bisschen und man hat versucht, dass es zum Laufen kommt, man hat Use Cases angeschaut, versucht das umzusetzen. Aber es hatte leider dort in diesem Team nie die Priorität, die es hätte haben sollen aufgrund anderer Prioritäten. Ganz einfach.

**Speaker1:** [00:04:07] Das heisst,Du hast selbst bereits lokale Software entwickelt.

**Speaker2:** [00:04:12] Direkt entwickelt, nicht, ich habe mich damit auseinander gesetzt, weil bei the UI Path, die Voraussetzungen doch ein bisschen höher sind, war mein Eindruck. Ich habe es mal ein bisschen angeschaut, ein bisschen herumgespielt, aber jetzt nie wirklich selbst jetzt direkt programmiert, nein.

**Speaker1:** [00:04:41] Und inwiefern unterscheidet sich aus deiner Sicht die Low Cost Softwareentwicklung durch den Enduser von der klassischen Softwareentwicklung durch einen professionellen Programmierer?

**Speaker2:** [00:04:57] Das kommt auf die Umsetzung darauf an! Was man halt. Es ist so, dass wir für sehr einfache Fälle geht es natürlich theoretisch einfacher, weil halt de Enduser einfach selber mit diesen Tools, egal von welchem Hersteller, etwas einfach umsetzen kann. Das ist mal der Selling Point, was ja alle verkaufen. Du kannst es dir zusammen, in Anführungszeichen "klicken" und dann hast du einen mehr oder minder komplexen Flow, den du dann automatisiert oder ein Prozess, den du automatisiert hast. Meiner Erfahrung nach ist es sehr, sehr abhängig. Was ist das für ein Use Case also oder was für ein Prozess ist es und welche Art von Systemen sind involviert? Wenn man Kern Systeme hatte involviert sind, dann unterscheidet es sich meiner Meinung nach nicht viel oder sollte sich nicht unterscheiden, weil es eben halt dann Voraussetzungen gibt, die man eigentlich erfüllen füllen müsste. Also das heißt man

müsste Testumgebung haben, man müsste das Ganze testen, durchspielen, dokumentieren, durch jemanden abnehmen lassen etc. etc.. Daher je nachdem unterscheidet es sich meiner Meinung noch nicht unbedingt. Es kommt auch sehr darauf an, wie man das ganze Thema in der Organisation aufstellt. Weil ja meistens braucht man doch irgendwo IT Unterstützung, wenn es ein bisschen komplexer wird. Aber die Tools offiziell sind natürlich einfacher als jetzt natürlich klassische Software Entwicklung, wo alles programmiert werden, alles programmiert werden muss. Also das ist halt, ich sag jetzt mal der Hauptunterschied, du hast meistens eine grafische Oberfläche und je nach Programm geht es mehr oder minder einfacher, das zu zusammenzustellen. Aber ein gewisses technisches Verständnis ist meiner Meinung nach schon hilfreich.

**Speaker1:** [00:07:19] Hast du? Hast du Selbsterfahrung im Bereich von Scrum?

**Speaker2:** [00:07:24] Ja, wir arbeiten jetzt in unserem Team schon seit vielen Jahren mit Scrum bzw. dem Dev Ops Ansatz auch von Safe, Safe Frame, Framework. Scrum ist dort ein ein Teil davon. Aber wir haben wirklich den klassischen Dev Ops, also das heißt Fach bzw. In unserem Fall Business Analysten und die Entwickler IT sind in einem Team zusammen und arbeiten Hand in Hand. Also ein klassisches Team dann auch eigentlich noch mit Planung etc..

**Speaker1:** [00:08:06] Jetzt, wenn du dir überlegst, dass du, das der End User in einem Scrum Framework die Software entwickeln würde. Inwiefern siehst du die aktuellen Rollen als geeignet oder müssen allenfalls Rollen angepasst werden? Man hat klassischerweise den, den Product Owner, den Scrum Master, den Business Analysten, den Entwickler.

**Speaker2:** [00:08:35] Also ich denke für Low Code ist also die Frage wäre, wenn man es jetzt so macht oder das es eine Voraussetzung ist.

**Speaker1:** [00:08:47] Also wenn man jetzt das ganze in einem Scrum Framework abbilden möchte, welche Anpassungen wären allenfalls sinnvoll damit dann? Welche den Bedürfnissen gerecht wird.

**Speaker2:** [00:09:02] Ja, es ist. Es ist so Scrum kommt natürlich klassisch aus der IT. Also das ist dann ein IT Ansatz. Von dem her ist in einem klassischen Scrum Team sind einfach Softwareentwickler drin. Das wäre sind bei diesem Ansatz meistens nicht der Fall. Oder sagen wir es mal so vielleicht wenn man jetzt wirklich ein Team für das machen möchte, dann werdens Juniors oder so? Die einfach irgendwo mal etwas Informatik mäßig gemacht haben, aber jetzt nicht gerade, Ich sage jetzt mal, ein volles Studium durchgemacht haben, aber grundsätzlich ich meine Scrum generell kann man natürlich auch rein in einem Fachbereich machen. Es kommt dann halt einfach auf die Zusammenstellung an, du bist einfach anders organisiert, was vielleicht der Unterschied ist, dass das Du halt dann die Arbeit auch entsprechend normalerweise dokumentiert und die ganzen Anforderungen definieren musst. Also beispielsweise mit einem Programm wie Jira Du definierst die Anforderungen oder wenn es Fehler hat etc. und das geht dann weiter an den Entwickler bzw. Jetzt vielleicht der Low Code Developer, wenn man es jetzt ein bisschen überträgt für den Citizen Developer. Das ist für mich keine Voraussetzung das kann mann, das kann man machen. Aber ist jetzt für mich keine Voraussetzung.

**Speaker1:** [00:10:40] Jetzt. Denkst du, wenn du jetzt auch wiederum schaust, du hast ja unterschiedliche Elemente Werkzeuge.., was denkst du? Wie wirkt sich das auch auf die, auf das Schätzen von Entwicklungsaufwand aus? Wenn man jetzt eine Low Code Software nutzt, die eher schneller entwickelt werden kann.

**Speaker2:** [00:11:09] Also auf den Schätz-Prozess. Meinst du jetzt?

**Speaker1:** [00:11:11] Genau. Inwiefern denkst du? Man arbeitet in Scrum ja in der Regel mit einem Product Backlog, wo man Tasks erstellt, wo man Schätzungen macht, wo man zwei Wochen oder eine gewisse Zeit dann auch plant, wenn man eher iterativer arbeitet. Inwiefern sind dort Schätzungen möglich?

**Speaker2:** [00:11:40] Also rein das Schnitzen selbst wäre wahrscheinlich für mich kein Unterschied. Es würde halt einfach diese Person schätzen, die den Prozess dann halt entwickelt. Das wäre je nach Ansatz dann halt der Fachbereichs Mitarbeiter. Die Frage ist halt dann mehr Kann er das? Also wie viel Erfahrung hat er mit mit solchen Schätzungen? Weil die Schätzung beruht natürlich ein bisschen auf Erfahrung. Und je mehr Erfahrung du hast es genauer werden die Schätzungen. Das ist so ein natürlicher

Prozess, von dem her. Wenn jetzt einer neu einsteigt und das neu macht, werden die Schätzung wahrscheinlich. Wird es große Abweichungen geben. Aber je mehr. Diese Person oder das Team Zusammenarbeit, desto genauer werden natürlich die Schätzungen. Das ist für mich aber unabhängig, ob es ein Fach oder IT Mitarbeiter ist, es kommt darauf an, wie gut kennst du deine? Also wie gut kennst du jetzt die Software, wo du etwas umsetzt und wie gut kennst du das Umfeld drum herum?

**Speaker1:** [00:12:56] Das heißt, du führst es vor allem auf die Erfahrung der einzelnen Anwender zurück. Wie einfac das schätzen dann auch sein wird.

**Speaker2:** [00:13:04] Genau!, weil das Schätzen selbst ist ja grundsätzlich. Also geht schnell und einfach. Die Frage ist Wie genau ist die? Wie genau ist die Schätzung? Aufgrund der Schätze natürlich die ganze, die ganze Planung erfolgt. Darum hat man ja auch dann verschiedene Methodiken, wo man das diese Schätzungen einfach permanent prüft und challanged. Also was hast du ursprünglich geschätzt und was hast du am Ende gebraucht? Und da müsste man eigentlich, dass dann, wenn jetzt große Abweichungen existieren, müsste man das dann jeweils hinterfragen. Und sagen warum. Warum gibt es jetzt oder gab es jetzt hier einen großen Unterschied, um eben die die Genauigkeit zukünftig zu erhöhen.

**Speaker1:** [00:13:57] Und wie siehst du das auf die existierenden Gefäße, die man in Scrum hat? Man hat ja das Planning. Warum hat das Daily. Man hat Reviews. Wie siehst du das geeignet für Low Code? Bzw wenn jetzt ein Entwickler das nur Teilzeit macht, also ein dezentralisierte Entwickler, inwiefern sind diese Gefässe nach wie vor geeignet, dass deine Sicht.

**Speaker2:** [00:14:35] Also ich sage es mal so Das Daily kann man immer machen. Das ist das ist eine simpler Ansatz. Bei den anderen Geschichten kommt es halt schwer darauf an, in was für einem Umfeld befindet, also befindet sich diese Person? Also arbeitet, arbeitete auch sonst in einem Scrum Team? Ist es das gleiche Scrum Team oder ist es nur ein theoretisch virtuelles Team? Gibt es auch immer, gibt es auch immer mehr, weil faktisch natürlich. Ich sag jetzt mal das Pi Planing und raus drumrum ist ein relativ hoher administrativer Aufwand und es kommt halt darauf an lohnt sich das oder lohnt sich das nicht? Das hängt dann aber schwer auch vom Umfeld ab. Also einem Beispiel; Wenn er jetzt nur für sich selbst einen Prozess automatisiert, der keine

Schnittstellen hat. Dann braucht es diese ganzen Sachen natürlich nicht. Das wäre ein völliger Over.., Overhead. Hat er jetzt aber Abhängigkeiten zu anderen Teams, die auch noch beispielsweise nach Safe arbeiten. Dann ist er fast dazu gezwungen, das Ganze mit zu machen, damit seine allfälligen Anforderungen überhaupt umgesetzt werden. Also ich würde so sagen, nur dort, wo es halt notwendig ist. Also wenn er halt Schnittstellen hat, Abhängigkeiten und das dort auch gemacht wird, dann ja. Ansonsten womöglich. Würde ich sagen; "nicht machen", weil es halt schon ein Zeit-Fresser ist, wenn man nicht die ganze Organisation darauf abgestimmt hat, weil sonst das ist viel Organisation für den relativ wenig unter Umständen.

**Speaker1:** [00:16:33] Wo siehst du generell die Vorteile von Scrum?

**Speaker2:** [00:16:38] Grundsätzlich der iterative Ansatz ist ein Vorteil. Also man macht nicht ein Projekt und dann der Wasserfall System. Also man definiert die Anforderung und schmeißt Streit hierüber. Und nach einem halben Jahr bekommt man oder einem Jahr bekommt man etwas, sondern der iterative kurzfristige Ansatz ist sicher ein Vorteil. Wenn man es.., die Einschränkung ist, wenn man selbst unter Kontrolle hat. Je größer die Organisation und je mehr Abhängigkeiten es auf allen Ebenen gibt oder je mehr Teams du brauchst, um für etwas abzustimmen, dann ist es nicht immer so agil, also ein Beispiel; Wenn ich jetzt etwas entwickeln muss, ein Prozess oder in diesem Fall muss es umsetzen. Ich brauch aber ein anderes Team. Und dieses Team ist jetzt aber extrem ausgelastet. Da muss ich zu diesem Team gehen, das Pitchen und dann gibt es eine Reihenfolge dort. Und wenn ich Glück habe, dann wird das im nächsten PI-Planning Zyklus normalerweise drei Monate gemacht und wenn ich Pech habe, nicht. Das heißt, wenn du mehrere Male Pech hast, dann reden wir hier halt von ewigen Verzögerungen. Also iterativ ist es eigentlich dann wirklich nur, wenn alle. Permanent. Ich sage jetzt mal schnell umsetzen oder du nicht viele Abhängigkeiten hast, dann bringt das definitiv Vorteile. Ansonsten. Kann es auch zur Bürde werden. Also man ist dann nicht viel.., Ich sage jetzt mal gefühlt nicht viel schneller. Was es sicher bringt, ist Transparenz. Ich sage immer, die Transparenz ist auf jeden Fall höher als früher, weil es dann eigentlich auf den Tisch kommt und man sieht, das ist alles hier und das muss anders gemacht werden oder das geht und das geht nicht. Also Transparenz finde ich auch ein großer Vorteil von Scrum und in einem klassischen Scrum Team halt die direkte Zusammenarbeit zwischen IT und Fach oder Business Analysten. Sagen wir mal so.

**Speaker1:** [00:19:01] Welche welche Nachteile schon einige angedeutet. Welche konkreten Nachteile siehst du generell in Scrum?

**Speaker2:** [00:19:09] Ähm. Ja, eben. Es kann. Es kann sehr administrativ werden. Das betrifft jetzt aber nicht unbedingt. Das betrifft dann vor allem den PO bzw. Scrum Master, weil der Scrum Master eineigentlich für das.., für das da ist, aber ss wird halt manchmal sehr administrativ. Das hat dann aber auch mit Governance Themen zu tun, die man einfacher erfüllen muss. Man muss alles dokumentieren, man hat einen Abnahmeprozess, den man einhalten muss. Also das heißt, dass zum Beispiel ein Business Analyst etwas testet und dann der eine Fachbereich testet etwas, bevor es in Prod. geht und als solche Sachen. Natürlich hatte man das auch in der klassischen Scrum Entwicklung, aber du hast es halt ein bisschen, ich sag jetzt mal mehr, weil man halt genauer und detaillierter dokumentiert. Auf der anderen Seite, wenn es jetzt bei den Nachteilen der Vorteil ist, halt auch hier die Transparenz, also man findet alles, man muss es einfach finden. Das ist dann in der Masse wieder die andere Herausforderung. Das ist sicher ein bisschen der Nachteil. Plus, es kommt darauf an, wie es in der Gesamtorganisation eingebettet ist. Das ist auch ein bisschen eine Herausforderung. Das sieht man jetzt auch bei uns, ich nenn's jetzt nich Scrum aber Safe, man hat dann wie ein bisschen zwei Welten oder noch Schatten. Also dort war Scrum Teams und dann gibt es irgendwo doch noch Teamleiter, Abteilungsleiter, Management und alles drum herum und alle möchten dann trotzdem noch irgendwo Einfluss nehmen. Je nachdem, wie weit der Prozess schon ist von der Umstellung. Ich sage jetzt Bei uns ist es, haben wir das eigentlich nicht mehr. Von dem ist es wirklich was Team plus minus, das entscheidet. Aber ich denke vor allem Fach-Organisation tun sich da wahrscheinlich ein bisschen schwerer. Es ist halt ein Transformationsprozess, weil du gibst, man gibt dann halt Macht dem Team mit den entsprechenden Vor- und Nachteilen. Also man man komm darf mir selbst entscheiden. Man muss es dann aber auch. Das muss man halt auch zulassen. Das sind so ein bisschen.., sorry ein bisschen mixed Antwort.

**Speaker1:** [00:21:40] Das ist in Ordnung, soll Platz haben. Denkst du.., oder inwiefern denkst du, dass du die Softwareentwicklung durch den End User in einem Scrum Framework abbildbar ist?

**Speaker2:** [00:22:00] Also. Die Frage ist halt, ob es in jedem Fall notwendig ist. Also ein Beispiel; Wenn ich jetzt natürlich für mich selbst etwas entwickle, dann sollte ich ja den Prozess kennen, für den ich eine Automatisierung mache. Also brauche ich in dem Sinne ja keinen separaten Business Analysten, der mir das vorgibt und mir die Anforderung schreibt, sondern ich mache sie mir selbst. Wenn man aber sagt, man schafft die Strukturen, dass man halt Developers hat und dann wirklich klassische noch Business Analysten und die arbeiten zusammen in meinem Scrum Team, dann ist das natürlich sinnvoll. Es kommt ein bisschen auf die Organisation darauf an, das will man erreichen, aber dann ist man natürlich fast schon bei einem, klassisches Scrum Team, außer dass es halt nicht 100 % Softwareentwickler sind unter Umständen, aber das hängt ein bisschen von dem und wie breit, wen man das ganze Thema halt in einer Organisation ausrollen.

**Speaker1:** [00:23:13] Ja, eine Bevorteilung.., oder sagen wir die drei Pfeiler von Scrum beruhen auf Transparenz, auch auf Inspektion. Das heißt, dass man die Inhalte immer wieder überprüft, dass sie auf die Ziele hinauslaufen, auch darauf, dass man erkennt, dass das Anpassungen erforderlich sind. Kann man den Scrum Prozess vielleicht dennoch als Grundgedanken nutzen und einzelne Elemente anpassen für die kleinen Entwicklungen.

**Speaker2:** [00:23:55] Also ja, Coding... Ich sage mal Scrum als Grundgedanke ist.., man kann sicher immer Elemente daraus nehmen. Ich sage mal das  meiste was die.., oder das, was die meisten zuerst machen, ist ein Daily. Einfach so ein tägliches Update, das man einmal pro Tag miteinander redet und weiß, was was passiert. So im Team Neuigkeiten austauschen etc. Das ist meistens das, was man auch bei uns jetzt in den Fachbereichen eingeführt hat. Bei der Entwicklung eben, es kommt darauf an was will man auch erreichen oder welche Art der Umsetzung hat man, wenn man natürlich ein Scrum Team macht und die Leute machen nichts anderes. Der Output ist natürlich dementsprechend höher und professioneller. Einfach weil du die entsprechenden Strukturen schaffst. Aber ich denke, ein wirkliches Scrum Team macht erst Sinn in seiner vollen Ausprägung. Wenn man dann Richtung Developer geht. Wenn man sagt, man hat halbprofessionelle, "halbprofessionell", aber ja einfach nicht klassisch ausgebildete Softwareentwickler, die auch wirklich sich mit der Software auskennen, die Prozesse halbwegs kennen und dann entwickeln sollen, dann macht es Sinn. Ein Scrum Team zu machen. Weil wenn er diese.., sich auch so damit befassen kann, dann

braucht er auch Input. Und da macht es Sinn, dass ich sage jetzt mal Business Analysten die entsprechende Vorarbeit machen bzw. dann auch was man gerne vergisst ist der Unterhalt der ganzen automatisierten Prozesse. Das muss ja auch immer gewährleistet sein. Und das ist natürlich in einem Scrum Team oder in einer Scrum Aufstellung tendenziell viel einfacher, weil man dann sagt was du entwickelst, für das bist du eigentlich auch verantwortlich. Also das heißt im Betrieb, man muss dann nicht nachher wieder suchen gehen, wer kümmert sich dann drum, wenn irgendetwas passiert?

**Speaker1:** [00:26:12] Und vielleicht noch noch eine andere Frage; Wie würdest du Low Code Software Development.., welche Gefäße generell jetzt nicht zwingend auf Scrum gesehen. Würdest du sinnvoll finden in einem Software Lifecycle, dass Du sagst; man startet vielleicht mit der Business Analyse oder der Definition der Anforderung. Und irgendwann muss dann ja auch das, die Software entwickelt sein oder du hast unterschiedliche Phasen und Schritte.

**Speaker2:** [00:26:48] Für mich unterscheidet es sich dann eigentlich nicht.., nicht groß. Ich muss eine. Ich mache klassisch eine Analyse. Da ist er mit dem klassischen Fachbereich, wenn ich es nicht selbst abdecken kann, bereitet das entsprechend auf. Dokumentier das irgendwo die Anforderung. Dann bespreche ich das mit dem Developer. Der Developer stellt dann seine Fragen, macht die Umsetzung. Dann muss es getestet werden. Zuerst vom Developer, eigentlich dann vom, meistens vom Business Analyst im Scrum Team und dann vom reinen Fachbereich. Und dann wird das Ganze deployed, das setzt dann eben auch voraus, die ganzen Umgebungen müssen da sein. Also für mich unterscheidet es sich eigentlich vom Prozess her nicht. Der Unterschied ist vielleicht, dass er tendenziell schneller sein sollte mit mit der Entwicklung, als wenn er jetzt klassisch.., also wenn man klassisch alles halt aus programmiert. Das soll es ja normalerweise eben nicht sein, sondern einfach, dass man halt mehr und schneller und vielleicht weniger komplexe Sachen macht. Aber der Prozess selbst wird oder sollte sich meiner Meinung nach oder man muss sich darf sich gar nicht groß unterscheiden, weil man dann halt einfach alle Kriterien. Sicherheitsstandards etc. erfüllen muss, aber erst ab Stufe, ich sag jetzt mal Developer, also wenn man so ein Team aufsetzt etc. dann ja.., Stufe darunter, individuell zu entscheiden, aber dann natürlich sehr, sehr abgespeckt, aber manchmal dann halt trotzdem auch. Ich meine, wenn es nur schon das eigene Team nutzt und es kommt

dann halt auf den Prozess an oder wenn ich etwas, wenn ich beispielsweise einen Versand automatisiere, dann muss ich natürlich einfach sicher sein, dass das richtig funktioniert und dann muss ich das eigentlich auch grundsätzlich auch testen. Wenn ich es nur für mich selbst brauche, dann bin ich selbst verantwortlich und dann fast gar nichts. Also für mich ist es Stufen.., Stufen abhängig.

**Speaker1:** [00:29:21] Kommt dann eine neue Rolle dazu oder verschmelzen allenfalls Rollen durch die neue End-User Programmierungs-Sicht. Wie ich gehört habe, ist eigentlich der Business Owner zeitgleich der Business Analyst, der Entwickler.., Fragezeichen auf der Tester?

**Speaker2:** [00:29:52] Ja, das kommt jetzt ein bisschen auf die Firma drauf an, also was für eine Struktur hat die Firma, wie will man es aufziehen? Es kann neue Rollen geben und es muss vielleicht sogar neue Rollen geben. Einfach weil man beispielsweise gar nicht so viele Business Analysten hat in einer Firma wie man sitzt hier, wenn man es breit ausrollen würde vielleicht bräuchte. Also dann wäre theoretisch der reine Fach-Mitarbeiter, der mit dem Kollegen, der vielleicht ein bisschen programmieren kann, zusammenarbeitet. Wäre dann ein bisschen wie der Business Analyst. Die Frage ist dann halt immer kann diese Person das? Weil es gibt halt gewisse Fähigkeiten, die man dann mitbringen muss. Und dann stellt sich die Frage definiert man eine Rolle ein bisschen zentral? Der das dann für mehrere Teams irgendwie abdeckt oder nicht? Weil es ist klar, wenn man. Also die klassischen Rollen aus dem Scrum übernimmt. Dann heißt es erstens Head Count, und wenn man Head Count spricht, dann braucht es Volumen. Damit du diesen Head Count ja auch rechtfertigen kannst. Also das kann für mich dann schon sein, dass es wie neue Rollen wie Mischformen gibt. Einfach weil man sagt; Ich habe jetzt zwar zehn Developer, aber ich kann mir jetzt nicht leisten, noch mal zehn Business Analysten dazu anzustellen. Aber ich kann vielleicht ein oder zwei für diese zehn, wenn die es brauchen. Oder Leute weiterentwickeln, die das vielleicht unterstützen können, so irgendwie. Also es wird wahrscheinlich ein Mix und das kann der neue Rollen.., es kann dann neue Rollen geben, macht dann aber die Organisation vielleicht und je nachdem auch nicht einfacher.

**Speaker1:** [00:31:56] Ich denke, wir haben sehr viele gute Punkte angesprochen. Für mich gibt es noch zwei Fragen, die ich gerne stellen möchte. Gibt es da aus deiner

Sicht noch Punkte, welche für meine Forschung interessant sein könnten, aber bisher noch nicht angesprochen worden?

**Speaker2:** [00:32:16] Also jetzt beziehe ich bezogen auf Scrum.

**Speaker1:** [00:32:18] Genau in Bezug auf Low Development Scrum.

**Speaker2:** [00:32:22] Also ich denke, was man vielleicht schauen möchte, also schauen muss, wo setzt man Scrum in einem Framework ein? Weil es gibt jetzt eben viele, viele sogenannte.., wir haben jetzt SAVE, es gibt noch LAZE, es gibt noch weiß der Gucker was so Frameworks. Alle haben Elemente daraus, Abwandlungen. Ich glaube, die meisten machen irgendetwas mit Scrum, aber der Überbau kann dann halt noch eine Rolle spielen und dann in die Komplexität hineingehen. Also man muss halt immer vielleicht schauen, wo macht man.., oder in welchem Umfeld ist Scrum eingebettet? Also hat jetzt eine Firma rein.., sie macht nur unten Scrum, sonst gibt es kein über Gefäß. Dann wird es einfach schwierig, glaube ich, wenn man nur, also wenn es außerhalb der IT ist. Da muss man zuerst das Verständnis schaffen. Weil es.., es braucht dann halt einfach ein gewisses Verständnis, eine gewisse neue Organisation. Das ist auch nicht zu unterschätzen. Wie also das Mindset, das man damit verändern muss.., also es ist nicht nur...

**Speaker1:** [00:33:42] Also eigentlich die Maturität in Bezug auf Scrum oder eines Unternehmens, wenn ich es richtig verstehe.

**Speaker2:** [00:33:47] Genau. Also wenn man wirklich Scrum.., also jetzt mal rein Scrum bzw.die Idee dahinter umsetzt. Dann bedeutet das eigentlich, dass man sich neu strukturiert und anders strukturiert. Und gewisse klassische Führungsrollen gibt es dann eigentlich nicht mehr. Das ist aber immer ein großer Punkt. Das haben wir auch bei uns festgestellt. Das ist ein relativ langer, anstrengender Prozess, wenn man aus einer klassischen hierarchischen Organisation kommt. Und da gibt es einfach Reibungspunkte, den man sicher auch betrachten muss, denke ich, wenn wir uns, wenn man uns anschaut. Das hat aber nichts weniger mit.., mit Low Code oder so zu tun, sondern mehr mit klassisch Scrum wie Scrum aufgestellt oder eingebettet.

**Speaker1:** [00:34:51] In einen wichtigen Punkt. Unbedingt. Und jetzt meine abschließende Frage wäre Gibt es in deinem Netzwerk noch weitere Personen, die ich im Rahmen meiner Forschung befragen könnte sollte, wo du sagst, die könnten noch einen Mehrwert bringen in die in die Sicht?

**Speaker2:** [00:35:09] Das kommt darauf an, was du genau... Also, ich könnte dir sicher unseren Scrum Master empfehlen, wenn dich das ganze theoretische Thema jetzt interessiert, auch was SAVE betrifft etc. Wie das eingebettet ist, was es da vom Schulbuch her alles zu bedenken, zu überlegen gibt, dann kann ich dir sicher Hans Peter Graf empfehlen, weil er ist der Master of.., für diese Sachen. Er macht auch die Seife, Schulungen intern etc.. Also wenn dich das interessiert, könnte ich da den Kontakt herstellen und du kannst ihn direkt auch ansprechen. Ist kein Thema. Er ist da sicher auch gern für Auskünfte da. Das andere ist halt. Eben die verschiedenen Rollen. Da könntest du natürlich immer einen... Möchtest du wissen, wie ein PO arbeitet? Also wie tief in den jeweiligen Rollen möchte man ein.., eintauchen? Ansonsten eben interviewpartner Jäger. Da werde ich noch den Kontakt herstellen. Er ist auch gerade an einer Arbeit, am Schreiben gewesen. Mehr jetzt da wahrscheinlich auf die Technik bezogen, aber da ist sicher ein Austausch, was ihr jeweils so herausgefunden habt. Könnt ihr vielleicht gegenseitig voneinander profitieren? Und kann ich.., kann nachher sonst noch versuchen nach dem Interview, ob er noch da ist und kann ich da mal den Kontakt herstellen, dass ihr euch austauscht.

**Speaker1:** [00:36:56] Hat er selbe Anwendungsfälle im Low Code oder weißt du...

**Speaker2:** [00:37:00] Ja er macht jetzt mit Automation Hero? Das ist halt Low Code Entwicklung. Er ist aber in der IT angehängt. Er hat auch eine IT Ausbildung ist da.., er arbeitet in der BIZ. Und ja, macht jetzt eine höhere.., ich glaube auch Master in IT, weiss der Gugger was... Genau also er dort auch gewisse praktische oder fast mehr praktische Erfahrungen. So ein Mix halt mit klassisch auch Softwareentwicklung halt ein bisschen so. Ansonsten.., ich könnt dir Kontakte geben nach Deutschland oder so wie wenn du wissen möchtest, wie andere das machen. Da könnte ich Kontakte herstellen, wenn sie in die praktische...

**Speaker1:** [00:37:51] Spezifisch Low Code wäre jetzt noch noch interessant, dort zu sehen, wie. Wie Leute vor allem im Bereich vom End-User Development in der Tendenz, das sind so ein bisschen die Themenfelder, wo ich auch im Moment noch...

**Speaker2:** [00:38:07] Genau also, da könnte ich Dir von Luxemburg, von Deutschland oder auch von Belgien könnt ich dir jeweils einen Namen angeben. Belgien machts jetzt.., startet erst damit. Mehr aus.., aus Verlegenheit. Weil sie momentan keine andere.., weil sie auch keine ressourcen haben, um ihren APN, das ist eigentlich klassisch Low Code, glaube ich. Aber sie müssen jetzt auch auf RPA wechseln, weil sie keine Ressourcen bekommen, um direkte Anbindungen zu machen. Also so der klassische klassische Problematik. Luxemburg, die machen auch so kleinere Entwicklung. Wenn ich sie richtig im Kopf habe, haben die einen Developer angestellt, aber der macht nur unkritische Prozesse. Das war meine letzte Stand. Und Deutschland machts in einem kleinen zwei drei Leute wo jetzt mit dem Tool UI Path entwickeln und eine IT unterstützen.., also diesen Namen kann ich Dir alle geben, die kannst du ohne Probleme kontaktieren, kannst dich einfach auf mich beziehen und die geben dir sehr gerne Auskünfte. Falls du wissen möchtest, wie sie das organisiert haben.

**Speaker1:** [00:39:16] Kannst du mir die Namen vor allem in Bezug auf die Partner aus den anderen Ländern einhalten, vielleicht in einem Mail zusammenfassen?

**Speaker2:** [00:39:24] Kann ich dir nachher noch schicken.

**Speaker1:** [00:39:27] Herzlichen Dank für die investierte Zeit.

**Speaker2:** [00:39:31] Kein Problem, gerne.

**Speaker1:** [00:39:32] Dann würde ich sagen, wir beenden hier den offiziellen Teil. Das heißt, ich werde jetzt diese Aufzeichnung hier beenden.