Interview 3

Speaker1: [00:00:08] Okay. Hallo, Interviewpartner. Herzlichen Dank, dass du dir Zeit nimmst für das Interview.

Speaker2: [00:00:14] Ja, gerne.

Speaker1: [00:00:15] Ist es für dich in Ordnung, wenn ich das Interview aufzeichne?

Speaker2: [00:00:19] Jepp.

Speaker1: [00:00:21] So, damit du ein bisschen den Kontext hast. Ich untersuche das Themenfeld der Low Code Softwareentwicklung durch den End-User und möchte unter anderem schauen, inwiefern dies auf die aktuelle Scrum Methode anwendbar wäre. In diesem Kontext habe ich einige Fragen vorbereitet. Das bedeutet, ich habe fixe Fragen, die ich stellen möchte. Dennoch ist es so, dass sich auch eine offene Diskussion ergeben darf. Das heißt, es wird nicht eine reine Frage-Antwort-Szenario. Zu Beginn kannst du mir kurz deine aktuelle Position und auch grob dein Arbeitsumfeld beschreiben.

Speaker2: [00:01:01] Ich habe zwei Rollen. Einerseits bin ich Business Analyst bei Unternehmen X, wo wir gerade den Kanton X in die Cloud bewegen. Das heisst von vielen On-Premise-Lösungen in eine Cloud basierte Lösung namens Office 365. Ein Bestandteil davon ist natürlich auch die die Power Platform, welche in das Segment von No und Low Code reinfällt. Andererseits bin ich Geschäftsführer einer kleinen Firma, wo ich mich hauptsächlich mit dem Thema Selbstorganisation und Digitalisierung auseinandersetze. Also das heißt, wie organisiere ich mich und wie kann ich Teile davon dann auch organisieren? Und dabei setze ich auch vor allem auf die Office 365 Umgebung, sehr viel Power Plattform, das heißt Model Getriebene Applikationen, Applikationen, die dann hauptsächlich auch mit Dataverse arbeiten. Power Automate als quasi eine Möglichkeit, einen Prozess zu automatisieren und natürlich auch die Prozess Steuerungs Elemente in Modell getriebenen Applikationen, die auch per se r es erlauben, Prozesse zu digitalisieren.

Speaker1: [00:02:24] Wann bist du das erste Mal mit Low Code Software in Kontakt gekommen?

Speaker2: [00:02:35] Willst du vielleicht mal definieren, was Low Code oder No Code ist?

Speaker1: [00:02:41] Das ist eine gute Frage. Low Code / No Codes sind in der Regel Applikationen, wo Modell getrieben sind. Das heißt, dass man dort mit vorgefertigten Elementen Software erstellen kann, ohne dass man explizit noch programmieren muss.

Speaker2: [00:03:00] Ja, ich würde sagen, das Jahr kann ich dir nicht mehr so genau sagen, aber ich denke, das sind vor allem oder waren in einem ersten Schritt vor allem Applikationen zum Erstellen von Webseiten. Also das war eine der ersten Anwendungsfälle, denke ich auch allgemein in der Branche, wo man weggekommen ist von davon, dass man HTML programmieren können muss, sondern einfach die Dinge quasi zusammen klickt. Eine sehr, sehr frühe Anwendung war Automater von von Mac. Also das ist das, was man heute als RPA bezeichnet. Da konntest du eigentlich etwas tun, hast es aufgezeichnet und dein Mac hat das dann einfach so wiederholt, wie du es aufgezeichnet hast. Also diese ganzen Schritte. Ja, das ist wahrscheinlich schon so zwischen sieben und zehn Jahre her, schätze ich jetzt mal ungefähr.

Speaker1: [00:03:57] Was denkst du? Inwiefern oder wo unterscheidet sich die Low-Code- Softwareentwicklung durch den End-User von einem klassischen Softwareentwickler durch einen professionellen Programmierer?

Speaker2: [00:04:13] Ja, ich. Also es haben natürlich sehr viele Dimensionen. Aber ich glaube, ich fange jetzt mal da an, wo du möchtest. Das ist da. Was bedeutet es für den Endkunden, Anforderungen umsetzen zu lassen? Oder er kann sie selbst umsetzen? Ich glaube, so im Sinne des Marketings kann der User alles selbst. Also er kann hingehen und kann eigentlich eine nahezu komplette Applikation selbst entwickeln oder eine Webseite. Wenn man dann ein bisschen vom Marketing und dem Marketing Slices weggeht, denke ich, dann gibt es gewisse Teile, die der Benutzer mittlerweile selbst tun kann. Ich würde aber nicht behaupten, dass er den IT-ler nicht mehr braucht. Im Kontrast dazu natürlich das eher klassische Vorgehen: Ich habe eine Anforderung oder einen Anpassungswunsch, eine Software. Ich gehe mal zu IT, zu einem Business

Analysten, unterhalte mich mit ihm, nimmt das mit, lässt es entwickeln vom Entwickler. Dann schaue ich mir das an im Sinne eines Tests, vielleicht mit dem Business Analyst oder dem Entwickler zusammen. Dann wird es über die verschiedenen Stationen quasi veröffentlicht. Also zuerst Testumgebung, User Akzeptanz Umgebung und dann schlussendlich die produktive Umgebung. Ja, ist natürlich ein Zeit Thema einerseits. Aber man muss auch ganz klar dazu sagen, es ist auch ein Fähigkeiten/Funktionalität Thema. Also wo der Entwickler grundsätzlich gesagt viel mehr Spielraum hat. Eine sehr dem Benutzer angepasste Lösung zu entwickeln ist natürlich in Low Code No Code in einem relativ standardisierten Rahmen nur möglich. Aber wie gesagt, ich denke 80/20, in den meisten Fällen ist das gar kein Thema. Ist es im Gegenteil, ist es sogar gut, weil es die Leute zwingt, in einem STANDARD zu denken. Im Gegensatz dazu, wenn sie frei entwickeln können, dann bauen sie sich da Paläste, die dann nie mehr abgerissen werden und dann letztendlich plötzlich sehr kritisch sind und dann ja auch Innovation usw verhindern.

Speaker1: [00:06:34] Wie siehst du den Unterschied auch in Bezug auf die Rollen, die sich während einer Softwareentwicklung ergeben können?

Speaker2: [00:06:44] Also ich glaube, die Rollen sind die gleichen. Ich denke allerdings, dass sich die Skills verschieben. Das heißt, der Endkunde wird. Ein bisschen technischer. Also der lernt vielleicht, wie eine Software funktioniert oder seine No-Code / Low Code Plattform. Der Business Analyst geht von der Analyse. Im Sinne der Aufnahme von Anforderungen und der Abklärung von Anforderungen viel stärker in die Beratung. Das heißt, einerseits hat er mehr Beratung, andererseits muss aber auch technischer werden, weil er viel besser verstehen muss, was sich die Auswirkung dessen, was der Endkunde tun muss, weil der Entwickler selbst wenn man dann zum Entwickler geht, der wird auch eher wirklich nur noch im Backend unterwegs sein. Also das heißt wirklich sehr technischer wird das System konfigurieren an einer Ecke, wo man nicht will, dass der Benutzer darauf zugreifen kann, wo der Benutzer das Skillset nicht hat, um diese Anpassungen im System zu machen, auch weil er gar nicht versteht, was die Auswirkungen sind. Und so gesehen verschiebt sich eigentlich alles ein Stück weit nach hinten. Aber bedeutet natürlich auch, dass der Entwickler mehr in der Kommunikation ist, weniger im Kämmerchen und weniger Code schreibt. Er wird immer noch Code schreiben, weil es lässt sich nicht alles einfach customizen und ohne

Code. Aber er wird viel weniger Code schreiben. Er wird mehr zum zum Plattformbetreiber. Wenn man so so will.

Speaker1: [00:08:30] Und wie siehst du die Möglichkeit, Entwicklungsaufwände zu schätzen? Also klassischerweise nimmt man Anforderungen auf, man definiert Tasks, man schätzt sie. Jetzt in einem Scrum Gefäß ist eine Schätzung, auch ein Teil der Planung. Wie siehst du das dort?

Speaker2: [00:08:56] Muss mir kurz Gedanken dazu machen. Ja, es ist. Also. Wenn man ehrlich ist, ich meine: Wie direkt darf ich sein?

Speaker1: [00:09:21] Sehr direkt.

Speaker2: [00:09:23] Ich sage es mal anständig. Die meisten Schätzungen sind ja meistens auch in den Wind halten. Das ist eine Mischung zwischen. So fühle ich mich heute und ein bisschen Erfahrung. In wenigen Fällen sind die wirklich gut und genau. Eine Schätzung ist ja auch oft dadurch getrieben, dass die Person, die das Budget hat, das Business irgendein Termin hören will oder eine Hausnummer, was das jetzt kostet. Und das ist meiner Meinung nach hauptsächlich dadurch getrieben. Um jetzt aber deine Frage zu beantworten Ich habe das Gefühl, wir gehen weg davon zu schätzen, wie viel Aufwand das ist. Hin dazu, dass wir sagen, wir nehmen uns so viel Zeit, um zum Beispiel einen Proof of Concept zu entwickeln, um überhaupt zu verstehen, wie lässt sich das oder lässt sich das überhaupt umsetzen und wie lässt sich das umsetzen? Ich glaube, dass da muss eine Verschiebung stattfinden, weg von. Ja, dann bin ich fertig und auch so viel Zeit. Und wenn ich nicht fertig bin, dann sage ich nochmal Ich brauche so viel Zeit und da bin ich immer noch nicht fertig, weil gerade No Code Low Code hat auch viel mit Reverse Engineering zu tun, also sei dies auf Seiten des Business, also um zu verstehen, wie der Prozess aufgesetzt werden muss. Aber auch von Seiten des, des, des, des Business Analysten und des Entwicklers. Einen ersten Wurf zu machen. Zu sehen. Ist es das, was der Kunde haben möchte? Ist es das nicht? Ich glaube, es wird eher in Richtung Time Boxing gehen, das Aufwand schätzen, weil man sich diesen drei Schritten Proof of Concept, User of Acceptance Testing und Deployment auf der Produktivität oder produktiven Umgebung irgendwie annähern muss. Aber wenn du mich fragen würdest. Wäre das jetzt so intuitiv meine Antwort. Wissen tu ich's nicht.

Speaker1: [00:11:19] Da ist die Herausforderung siehst du bei der Low Code End User Programmierung.

Speaker2: [00:11:25] Chaos. Also das. Es fängt da an, dass du wahrscheinlich ein Problem in der Versionierung haben wirst. Also der, der User, der wird hingehen, der wird damit etwas versuchen. Das wird der inkrementelle versuchen bisher nicht mehr weiterkommt. Manchmal wird er auch technisch etwas tun, was er vielleicht nicht tun sollte. Dann funktioniert seine Lösung nicht mehr. Dann geht er hin und kopiert die Lösung einfach und dann arbeitet er auf der nächsten Lösung weiter. Das heißt, es wird ein Life Cycling Thema geben. Welche Applikationen sind aktuell, welche sind aktuell inhaltlich, welche braucht es nicht mehr, welche braucht es noch. Das wird ein Thema sein. Ein zweites Thema ist eines, das jetzt schon beim Entwickler zum Teil kritisch ist. Dass das Thema der Dokumentation, also der Business User, wird das nicht dokumentieren. Der wird den Geschäftsprozesse dokumentieren. Aber was im Detail jetzt in Power Automate, in diesem Schritt wie, wann, wo da eingetragen hat, das wird er wahrscheinlich nirgendwo definieren. Das heißt, wenn diese Person nicht mehr da ist oder nicht verfügbar, die das gemacht hat, dann.... Wird der nächste das nehmen und wird es selbst tun. Und dann haben wir jetzt das nächste Thema im Life Cycling und der Versionierung. Das nächste wird sein, dass das Business anfangen wird. So nach dem Motto Fail fast. Nicht per se schlecht, aber sehr schnell an die Grenze kommen wird von dem, was umsetzbar und möglich ist. Und dann werden sie zu einem Business Analyst oder einem Entwickler gehen und dann werden wieder die klassischen Ressourcen absorbiert. Also das Ressourcenmanagement Thema, das geht nicht weg. Das wird einfach anders sein. Das ist dann halt nicht mehr der Betrieb, weil der Betrieb, der fällt ja dann weg. Das betreibt dann Microsoft oder irgendjemand, aber es wird hingehen zu 1000 Leute wollen etwas von dir und 900 davon tun eigentlich dasselbe, aber sie sprechen nicht miteinander. Also wirst du eine unglaubliche Redundanz haben, dass es wieder ein weiterer Punkt sein. Du bist unglaublich viele Redundanzen haben und so kannst du dich dem entlanghangeln. Weil nur weil jemand nicht mehr entwickeln muss, heißt das nicht, dass er IT nicht verstanden haben muss. Also die unterschiedlichen Schichten von da ist die Datenhaltung, das ist irgendwie Anwendungs Schicht, das ist die Repräsentationsschicht. Dieses ISO/OSI Modell, das musst du verstehen, sonst wirst du auch mit No Code Low Code nicht weiterkommen. Ich glaube. Es ist im Moment oder das Thema glänzt sehr stark auf Marketing Seite, ist aber noch nicht so

weit, dass es die Unternehmen wirklich verstanden haben und auch richtig einsetzen. Ist das eine Antwort für dich?

Speaker1: [00:14:16] Ja, sehr gut. Und welche Maßnahmen, denkst du, helfen dabei, diese Herausforderung zu überwinden?

Speaker2: [00:14:26] Klassisch: Testumgebung, wo die User selbst mal ausprobieren, testen können. Relativ stringente Prozesse, also im Sinne der Verantwortung. Und wer darf was? Auch Abgrenzungen von Seiten der IT in Sachen Support, dass der User weiß, er kann das jetzt tun, aber wenn er nicht mehr weiterkommt, dann muss er sich selbst helfen, dann kriegt er keinen Support von der IT. Das kann durchaus auch sein, dass er dann einen externen Partner nehmen wird, aber das wird er dann spüren in den Budgets. Dann wird er ein Thema kriegen mit seinen Vorgesetzten. Also ich denke mal, klare Rahmenbedingungen und eine klare Governance ist so, dass der Überbegriff das Stichwort was, was, was viel, viel besser sein muss, als das heute der Fall ist. Weil heute hast du Leute in der IT, die grundsätzlich einigermaßen verstehen, was IT wirklich ist.

Speaker1: [00:15:24] Und was sind aus deiner Sicht die die Vorteile von Scrum?

Speaker2: [00:15:33] Ähm, ja, auch da gibt es natürlich viele Dimensionen. Ich denke, die eine Dimension ist die der Zusammenarbeit. Also dieses Transdisziplinäre ist schon Teil davon, also das Business mit dem Entwickler und dem Business Analysten zusammen zu stecken und macht natürlich mit No Code Low Code einerseits mehr Sinn, aber auch mehr Spaß, weil alle beitragen können. Bis jetzt sitzt das Business da und sagt ja oder nein. Der Business Analyst ist im Sandwich und der Entwickler, der sagt, die haben nicht verstanden, was sie wollen. Also das ist ja so ein bisschen heute. Das macht es auch schwierig, mit Scrum zu arbeiten. Das heißt, es ist mal eine Dimension. Eine zweite ist, dass es so, diese, dieses iterative nicht mehr nur wenn man so davon ausgeht, dass dieses stärker involviert wird, nicht mehr nur in der Breite stattfindet, sondern auch im Business, weil das ist heute ein Riesenthema. Wenn das Business kommt, dann muss es sofort oder es ist eigentlich egal. Und nachher kriegen Sie eine gewisse Abschätzbarkeit. Wann können Sie etwas bekommen? Weil es gibt diese klassischen Sprints. Und wenn man jetzt davon ausgeht und sagt, okay, ich habe eine Time Box, ich entwickle jetzt ein Proof of Concept in einem Sprint von zwei

Wochen, dann weiss das Business grundsätzlich: In zwei Wochen weiß ich, ob es machbar ist oder nicht. In vier Wochen weiß ich, ob es mir hilft oder nicht. Und in sechs Wochen habe ich's. Ich denke, das sind so Vorteile, die die Scrum mit sich bringt. Aber es gibt natürlich auch Herausforderungen. Ich meine, viele Business Abteilungen funktionieren relativ klassisch, also auch hierarchisch. Es gibt ein Team Lead und dann plötzlich mit einem Scrum Master und einem Product Owner. Und der Teamleader ist eigentlich plötzlich nur noch Coach und Mentor. Ich meine, es ist ein riesen Change. Da hat das wahrscheinlich weniger mit Scrum zu tun als mit der Unternehmenskultur und der Unternehmensführung, die das entweder wirklich offiziell einführt. Sodass die Leute das auch als verpflichtend wahrnehmen. Und sonst wird es scheitern.

Speaker1: [00:17:51] Und jetzt vielleicht einfach nochmals etwas spezifischer der Blick auf Scrum Elemente. Ich möchte dir einfach ein paar Wörter geben und du sagst mir, inwiefern das geeignet wäre für die Low Code End User Programmierung. Die klassischen Rollen innerhalb eines Scrum Teams, also der Entwickler, Product Owner Scrum Master. Der Business Analyst natürlich noch. Siehst du die als geeignet oder müsste man weitere Rollen berücksichtigen?

Speaker2: [00:18:25] Nein sie sind in jedem Fall geeignet. Man muss maximal über den Inhalt sprechen. Was ist wo? Also vielleicht ist es dann nicht mehr ein Konzept, das geschrieben wird, sondern ein Proof of Concept direkt im System, der dann irgendwie in einem gewissen STANDARD, der vereinbart wird, dann dokumentiert wird. Weil das. Aber nein, es braucht keine neuen Rollen. Das ist mehr als genug, was Scrum heute bietet.

Speaker1: [00:18:50] Und wenn du jetzt an die klassischen Scrum Events denkst, sagen wir Sprint, zum Beispiel.

Speaker2: [00:18:58] Bleibt gleich. Bleibt Daily Retro. Muss. Muss alles bleiben reviews auch. Es ist. Ja. Im Gegenteil, das wird immer wichtiger, weil gerade jetzt beispielsweise eine Review. Wenn jetzt der Business User Teil davon ist, dass Reviews, dann wird es ja mehr zu einem Transfer, zu einem Wissenstransfer. Dann geht es ja plötzlich mehr darum, wenn man jetzt unterschiedliche Produkt Teams hat, dass die voneinander sehen, wer was tut, dass es dann Synergien ergibt oder man sieht, die tun das, das wollen wir auch tun. Dann gehen wir zu den und fragen mal. Also

ich glaube es kommt wirklich, wir kommen näher und die Grund Logik und das Grundverständnis von Scrum als wir das jetzt sind mit einer klassischen Entwicklung.

Speaker1: [00:19:50] Was noch ein Thema war. Was ich auch interessant fand, war jemand in den vorhergehenden Interviews. Wir haben diskutiert, wenn jemand nur als Teilzeitentwickler arbeitet, Low Code entwickelt und weiterhin im Business ist. Wäre ein Daily dennoch angebracht oder denkst du man müsste die Häufigkeit von dalies anpassen, die Teilnahme freiwillig machen?

Speaker2: [00:20:18] Ja, ich glaube, man müsste mal ganz grundsätzlich verstehen, was ein Daily ist. Ich meine, ein Daily ist nicht einfach ein täglich wiederkehrender Termin, der im Kalender ist und wo man hingeht, sondern ein Daily hat einen Sinn, seinen Zweck. Und dann wird man auch je länger je mehr darauf kommen, dass wenn jemand zum Beispiel ein Teil Entwickler ist, wie du genannt hast im Sinne Business, hat aber Spaß an Low Code und entwickelt da zum Teil Konzepte mit oder Proof of Concept. Wie auch immer, dann heißt das vielleicht in Zukunft, dass er einen dezidierten Tag hat für das, also dass er nicht mehr sieben Dinge an einem Tag tut, wie er das heute tut, am Morgen, an dem, dem und dem, was an drei Daily ist. Und dann am besten noch irgendwie ein bisschen Business, ein bisschen und ein bisschen das. Sondern da heißt es vielleicht, er hat einen dezidierten Tag oder halben Tag, in dem er das tut, was überhaupt sowieso sinnvoller wäre. Und du siehst jetzt auch diese Konzepte aus der agilen Arbeit sind sehr robust. Was nicht so robust ist, ist das a das Verständnis, b die Anwendung und c daneben auch die die Ergebnisse, die du produzierst mit einer möchtegern agilen Organisation, die sich jetzt immer mehr zeigen, je agiler ein Business werden muss.

Speaker2: [00:21:33] Also eine Firma wie die, in der du arbeitest, die muss immer agiler werden. Mittlerweile nicht, weil der Kunde nachfragt, weil das ist jetzt in dem Bereich, in dem du bist, also Finance, ziemlich egal, weil Finance war immer so, wird immer so sein und es gibt keine Gründe dies zu ändern. Und sonst gibt es immer noch die Ausrede, es gibt regulatorische Gründe, wieso, wieso sein müssen. Das ist easy, da kannst du dich gegen aussen relativ entspannt abschirmen. Plus es ist Geld ohne Ende da. Es gibt keinen Druck. Aber wenn jetzt natürlich intern Druck entsteht, weil man neue Produkte, neue Leistungen, Budget, Überschreitungen usw in den Griff bekommen muss und man deswegen immer agiler werden muss. Dann hast du natürlich plötzlich

ein Problem, wenn du Agilität nicht sauber eingeführt hast oder die Leute im Grundsatz nicht verstanden haben, was Agilität ist. Das heißt, deine Frage, die sehr spezifisch war, das hat sie auch zu Beginn gesagt, die hat eine riesen Auswirkung, weil sie sehr verkettet ist zu Themen der Wertschöpfung in einem Unternehmen, der Kultur usw.

Speaker1: [00:22:36] Ich. Ich möchte noch kurz den Citizen Development Ansatz bewusst nochmals reflektieren. Vielleicht auch etwas losgelöst von Scrum. Braucht es dort generell neue Plattformen, neue Boards für die Kollaboration? Ich denke, für die Priorisierung zum Beispiel?

Speaker2: [00:23:00] Sag mal! Sag noch mal ganz kurz Citizen Development. Erklärt doch das kurz. Bitte.

Speaker1: [00:23:06] Also Citizen Development bedeutet das eigentlich nicht IT-Mitarbeitende Software entwickeln. Also keine IT Fähigkeiten haben.

Speaker2: [00:23:17] Okay. Nein. Das ist eine gute Frage und die Frage ist möglicherweise auch der Genickbruch des Ganzen, weil. Du hast paar Stichworte genannt Zusammenarbeit, Abstimmung, Priorisierung. Also schlussendlich geht es um Ziele und Ergebnisse und Strategie und Taktik. Und ohne jemandem zu nahe zu treten. Aber entweder ist Strategie ein Blick in den Rückspiegel dessen, was ich die letzten fünf Jahre getan haben. Und wir projizieren das in die Zukunft und wir nennen das dann Strategie. Wir gehen hin und sagen Das ist jetzt Trend und schreiben uns diese Trends auf die Fahne und sagen dann Das ist unsere Strategie. Diese Strategien sind sehr spannend, weil die bleiben immer auf dieser initialen Geschäftsleitung Präsentations Folie. Es gibt zwar ein Team, das das umsetzen soll, aber das kann es gar nicht, weil das ist einfach nur Bullshit-Bingo! Also da gibt es gar nichts, das ist heisse Luft und das ist einfach so und ich glaube deine Frage ist aus dem Grund gut. Der Genickbruch des Ganzen. Weil jetzt das Business endlich mal wirklich eine Strategie haben muss. Und ein Ziel und eine Vision. Also nicht nur ein paar nette Worte auf einem Powerpoint, sondern Klammer auf. Ich gebe dir eine Analogie. Storytelling ist im Moment eines der angesagten Dinge. Wenn du etwas verkaufen willst, dann geh nicht hin und sag Was sind die guten Dinge, die mein Produkt kann? Sondern erzähl eine mitreißende Story. Am besten etwas mit Emotionen und Drama. Und so, dass die Leute. Es regnet. Vorne geht die Tür auf. Ich schnappe mir mein Bike, stecke meine Airports rein, die jetzt

wasserdicht sind und gehen in den Regen. Geile Story, der Punkt. Und das ist dann auch der Bogen zurück. Was die Leute noch nicht verstanden haben ist man entwickelt nicht ein Produkt und schreibt dann eine Story dazu. Man hat eine Story und entwickelt dann das Produkt. Das heißt, man hat ein Zielbild. Und auf Basis dieses Zielbild kann man eine Strategie, eine Taktik und einen Plan entwerfen. Und heute funktioniert das aber andersrum. Heute entwickelt eine Versicherung ein Versicherungsprodukte, von dem sie gesehen haben, dass es die Konkurrenz auch tut. Weil irgendein Fintech Start up das mal gemacht hat. Und Sie glauben, dass das irgendetwas bringt? Da geht man hin und versucht, das zu kopieren oder irgendwie selbst auf die Reihe zu bekommen. Und dann am Schluss geht man hin und versucht das zu verkaufen. Und darum hat das Business auch keine Ausrichtung, sondern es ist immer sehr reaktiv. Und wenn das Business jetzt diese. Ausrichtung, Priorisierung. Um den Bogen zurück zu machen zu deinen ursprünglichen Frage effektiv. Wer gewinnbringend einsetzen will und das auch richtig tun will, dann brauchen Sie eine Vision. Dann brauchen Sie von Beginn an eine Story, einen Added Value, den Sie liefern. Und daran richten Sie sich aus. Dann stellen sich all die Fragen gar nicht mehr. Ist das verständlich?

Speaker1: [00:26:42] Ja, finde ich schön, die Analogie. Für mich wären da noch zwei Fragen Einerseits Gibt es aus deiner Sicht Punkte, die für meine Forschung noch noch interessant sein können?

Speaker2: [00:27:03] Also, es stellt sich immer die Frage, wie tief willst du gehen und. Und was willst du effektiv rausholen? Aber wenn du wirklich Mehrwert liefern willst, dann solltest du zu all diesen Bereichen so ein bisschen das Wieso herausschälen. Also wieso agile Arbeit? Wieso will das Business mitentwickeln oder wieso soll das Business mitentwickeln? Wieso gibt es einen? Doch relativ tiefen Graben zwischen Business und IT. Wieso gibt es die unterschiedlichen Rollen? Diese ganzen WIESO Fragen sind glaube ich sehr sehr relevant um wirklich die Zusammenhänge auch sauber aufzuzeigen und in dem ersten Schritt überhaupt mal zu verstehen. Ich glaube es wirklich. Es ist viel Grundlagenarbeit, die zu wenig getan wird. Und die deine Arbeit bestimmt sehr viel Substanz verleihen wird und auch sehr viel Mehrwert liefern wird, weil du bei den Annahmen beginnst, weil sehr oft sind die Annahmen schon falsch. Die Annahme selbst zu entwickeln ist, dass es die KI nicht auf die Reihe bekommt, aber die IT bekommt es nicht einfach nicht auf die Reihe. Die IT muss Lösungen betreiben, die aus IT technischer Sicht völliger Schwachsinn sind, aber das Business hat es so

gewollt. Also vieles ist eine Selbstbeübung in der Organisation und wenn man das lernt zu verstehen, dann lernt man auch zu verstehen, wer welche Verantwortung trägt und das die inhaltliche Verantwortung dafür, dass die richtige Lösung entsteht. Die liegt immer beim Business. Und wenn dann eine schlechte Software dabei herauskommt, dann Mag es wohl am Entwickler oder sonst wem liegen, aber es ist trotzdem die Verantwortung des Business. Das heißt, das Business muss dann mehr in den Driver Seat und nicht einfach sagen Ja, wir haben das jetzt über den Haag geschmissen usw Ich glaube, da wirst du sehr viel Mehrwert kriegen, wenn du so ein bisschen. Einfach einen Schritt tiefer gehst, kurz die Annahme challangest um für dich auch sauber zu sein und dann aus einer sauberen Basis heraus auch deine Argumente suchst oder vielleicht auch widerlegt oder bestätigst oder widerlegt?

Speaker1: [00:29:15] Und was denkst du? Was sind die Grenzen des Citizen Development Ansatzes? Also wo siehst du die die Grenze. Auch vielleicht aus einer Risikosicht?

Speaker2: [00:29:37] Das ist sehr, sehr abhängig davon, von was für einem Unternehmen wir sprechen. Ich glaube, bei einem großen Unternehmen wie du in einem arbeitest, wird es diese Kern Systeme oder Legacy Systeme geben. Die die die die kannst darfst willst du nicht ablösen, weil ein Vertrag, der im Extremfall über 100 Jahre läuft über solche Systeme hinwegzuziehen, das ist eine Utopie. Und es wäre keine gute Idee. Aber gerade jetzt, im Bereich der Automatisierung von Business Prozessen, wenn wir wirklich vom Business sprechen, davon, Dinge nicht mehr in einem Excel zu führen, sondern neu in einer SharePoint Liste oder in einem DataVerse sehr, sehr sinnvoll. Also ich glaube, es wird so ein bisschen. Ich habe jetzt nicht gerade das richtige Wort dafür, aber es wird so ein bisschen etwas mit der Tiefe zu tun haben in der Wertschöpfungskette und der Kritikalität. Wo eine Entscheidung stattfinden wird, ob das jetzt Low Code geeignet ist oder nicht. Ich persönlich würde mir vorstellen, dass es eine Art eine Tabelle gibt, wo ich mich eigentlich daran orientieren kann, wo ich dann einen Quadranten sehe, wo ich sehe okay, das ist Low Code/No-Code. Das kann ich als Business allein entwickeln, das ist No Code, Low Code. Da brauche ich vielleicht Unterstützung. Das ist ganz bestimmt kein No Code Low Code mehr, sondern es ist echter Code. Aber das Business sollte sehr, sehr stark involviert sein. Und dann gibt es Backend Prozesse, Life Cycle Themen. Davon muss will das Business nichts wissen. Das ist IT Arbeit und da sind sie dann auch nicht mehr involviert. Ich denke es muss

irgendwie eine Möglichkeit geben so ein bisschen das zu clustern respektive eben auch so eine Entscheidungs-Matrix zu haben. Wo setze ich das ein und wo setze ich das nicht ein?

Speaker1: [00:31:45] Besten Dank, dann wären wir schon am Ende. Herzlichen Dank für die Zeit. Dann würde ich hier die die Aufnahme beenden.

Speaker2: [00:31:53] Jupp, gerne.