Interview 9

Speaker2: [00:00:02]

Speaker1: [00:00:08] Hallo, Benjamin. Ist es für dich in Ordnung, wenn mich das Gespräch aufzeichnet?

Speaker2: [00:00:16] Hallo, Pascal. Ja, das ist für mich absolut in Ordnung.

Speaker1: [00:00:19] Das ist sehr nett. Danke. Die kurze Erläuterung. Warum sitzen wir heute hier? Bzw. Warum bin ich auf dich zugekommen? Es geht darum, dass ich die möglichen Herausforderungen in Bezug auf die Low-Code- Softwareentwicklung durch den End-User analysieren möchte. Zudem möchte ich in einem weiteren Schritt schauen, inwiefern das in einem SCRUM Prozess abbildbar abbildbar wäre.

Speaker2: [00:00:45] Ja, okay.

Speaker1: [00:00:47] Zu Beginn kannst du mir deine aktuelle Position und deine aktuelle Arbeit beschreiben.

Speaker2: [00:00:54] Meine aktuelle Position in der IT Schweiz ist Product Owner. Kommissionierung. Und das sagt. Das tönt gut und ist auch gut für mich, oder? Aber ich habe kein Produkt, ich bin Produkt Owner weil ich irgendeine Rolle brauche oder genau, habe aber kein Produkt, wie das die meisten Product Owner Product Owner bei uns haben in der Deutschschweiz. Meine aktuelle Tätigkeit Haupttätigkeit ist Journey to the Cloud. Da bin ich gesamt Projektleiter für die Schweiz. Da geht es darum, unsere Anwendungen, die nichts mit Low Code oder No Code zu tun haben, sondern die klassischen IT Anwendungen aus dem Rechenzentrum I in das neue Rechenzentrum hinzubringen. Das ist die Haupttätigkeit. Ich bin immer noch dabei, gewisse Applikationen mit es geht ja um jetzt um Low Code mit Microsoft Power, Power Automate und all den Zusatztools, die da so um das Power herumschwirren zu erstellen oder zu implementieren, wenn sie schon erstellt sind oder Bugs zu beheben. Das habe ich in den letzten Jahren. Was soll ich sagen? Sommer 2019? Doch seit. Ja bald was? Seit zwei Jahren mache ich das in etwa genau.

Speaker1: [00:02:27] Und ist es auch das erste Mal, als du mit Low Code Software in Kontakt gekommen bist? Oder war das bereits früher?

Speaker2: [00:02:36] Das ist das erste Mal, dass ich mit Software in Kontakt gekommen bin, die, die den Namen oder die Bezeichnung No Code oder low Code hat. Früher hatten wir IDV Software, das individuelle Datenverarbeitung Software sah man das in den 80er Jahren getauft. Da war zum Beispiel das Microsoft Produkt Fox Pro bekannt. Mit dem habe ich auch gearbeitet. Auch Lookout ist das eigentlich sein Vorgänger von Power Automate oder MS Access ist eigentlich auch wenn man es genau anschaut, es ist fast das Gleiche. In dem Sinne schon. Man hat es einfach nicht so genannt. Ja.

Speaker1: [00:03:20] Und? Die Projekte, die innerhalb Unternehmem X umgesetzt habt. Habt irh die selbst gemacht oder wer hat die umgesetzt? Ein externer Dienstleister?

Speaker2: [00:03:30] Wir haben beide Varianten, wir haben Varianten, die hat ein externer Dienstleister umgesetzt. Wir haben Varianten, die hat ein Mitarbeiter, der mal was umgesetzt aus dem Fachbereich. Erfolgreich übrigens sehr schnell und sehr erfolgreich, oder? Die habe ich umgesetzt. Nicht ganz so schnell und nicht ganz so erfolgreich. Und sie waren aber auch komplexer.

Speaker1: [00:03:56] Und kannst du die Use Cases kurz beschreiben?

Speaker2: [00:04:00] Ich kann die beschreiben. Die Frage ist, welche der Use Cases möchtest du wissen?

Speaker1: [00:04:08] Also einfach exemplarisch.

Speaker2: [00:04:09] Ein exemplarisch. Also ein Arbeitskollege? Nein, ein Kollege von mir in aus dem Bereich Accounting Betrieb, Betrieb, Accounting. Der hat eine Application geschrieben, die heißt Provisionsumbuchungen . Das ist nicht wirklich eine komplexe Application. Da geht es darum, dass ein Außendienstmitarbeiter. Sich melden kann bezüglich seiner Provision und einen Antrag stellt. Wie die Provision verteilt werden soll. Also da gibt es so Vermittler und General Agenten und weiß ich was? Er

stellt diesen Antrag. Dann wird eine Bestätigungsmail geschickt und dann heißt es okay, Antrag genehmigt und dann wird es durch eine zentrale Stelle abgearbeitet. Es wird in unserem Provision System dementsprechend umgebucht und der Beleg wird abgelegt im Modus.

Speaker1: [00:05:07] Das wird automatisiert alles abgebildet.

Speaker2: [00:05:11] Die Dokumenten-Ablage ist nicht automatisiert. Das ist noch ein offener Punkt für die Ablage. Da kommen wir dann eben auch drauf zu sprechen. Das ist so ein Thema mit Low Code / No Code, da müssen wir mit dem Rest API's arbeiten. Dann fängt es dann schon an! Das ist dann schon das Ende von Low Code. Und ich mache schnell. Ja, genau. Wir haben aber so eine API gebaut. Bei uns hat ein IT Team diese gebaut. Wir müssen das sauber integrieren und wir benötigen spezielle Lizenzen, um diese API dann auch nutzen zu können. Ja, das ist auch so ein Thema. Kosten, Lizenzen und so weiter und so fort. Es ist nicht alles for free. Ein Grundgerüst ist for free und der Rest eben nicht. Okay, ein zweites Beispiel ist Wir haben durch einen externen Partner eine doch größere, ich würde sagen ja wirklich eine Business Applikation geschrieben. Ein Telefonbuch. Jetzt könnte man sagen Ja, ein Telefonbuch. Das ist ja nicht wirklich komplex. Es ist aber schon ein bisschen komplexer. Weil Daten aus dem Personal System aus dem aktiven Personal System importiert werden, braucht es auch eine spezielle Schnittstelle über Secure File Transfer. Und diese Daten werden dann ergänzt mit. Explizitem Wissen, das nicht im Personal System drin ist, damit im Außendienst und im Kundenservice als eine große Nutzerschaft, eine große Nutzerschaft noch Spezialisten gesucht werden kann. Zum Beispiel Ein Kunde ruft an und sagt Ein Kunde in. Landquart ruft an und sagt Wer wäre der beste Ansprechpartner für mich, wenn es um eine Pensionskassen Lösung für ein Kleinunternehmen gehen gehen würde? Und dann sagt der Mitarbeiter die Mitarbeiterin, die dann abnimmt, schaut in dieses Telefonbuch. Heißt Contact Book. Gibt Kanton Graubünden ein, sagt Kollektiv Leben Spezialist und findet dann die entsprechende Person aufgrund der Zusatz Daten die im System eingepflegt wurden und kann dann gleich die Kontaktdaten und Email usw an den Kunden weitergeben. Das ist eine zweite Applikation und die dritte Applikation die ich geschrieben habe ist eine Schriftgut Verwaltung. Da geht es darum, physische Schriftgut, die bei uns im Archiv, und zwar im physischen Archiv im Keller abgelegt werden, zu verwalten. Im Sinne von Ich möchte es ablegen, ich möchte es wiederholen, ich möchte es kassieren, das heißt ich möchte es löschen lassen, also

entsorgen, nicht nur löschen lassen usw mit diversen Flows und Vorgesetzten muss okay geben auswerten bezüglich Kosten. Es ist schon eine auch eine Business Applikation mit ein paar Use Cases. Man könnte wir haben ein Use Case Diagramm könnte ich dir auch geben, damit du siehst, wie wir das in etwa analysiert haben.

Speaker1: [00:08:26] Sehr spannend und, denn das heißt nur zum Verständnis, dass es dann wie ein digitaler Zwilling, den ihr habt für die Dokumente.

Speaker2: [00:08:33] Ja, es ist ein digitaler Zwilling, damit man das Dokument verwalten kann. Ja, genau. Genau.

Speaker1: [00:08:42] Dann. Würde ich gerne damit beginnen. Inwiefern unterscheidet sich aus deiner Sicht die Low-Code- Softwareentwicklung von der klassischen Softwareentwicklung?

Speaker2: [00:08:54] Das ist jetzt wirklich schon ein bisschen, wie soll ich sagen. Nicht mit Fakten hinterlegt, obwohl ich auch ein klassischer Entwickler bin. Ich habe das mal gelernt, wie man das machen würde, wenn man es macht, oder? Und sozusagen als Power User oder Citizen Development hast du ja genannt. Oder der der Developer. Ich sehe mich jetzt auch als Citizen Developer heutzutage und nicht mehr als Klassischen Entwickler. Was ist der Unterschied? Der Unterschied ist der Citizen Developer. Ist natürlich eine Unterstellung. Macht keine so detaillierte Analyse wie ein Profi Entwickler. Also ich sage mal die Fach Analyse ist eher rudimentär derart gegenüber der klassischen. Das ist der erste Punkt. Der zweite Punkt ist Er macht sich nicht viel Gedanken über den Betrieb der Lösung. Stabilität, Fehler, Management, all diese Dinge. Das ist nicht so, im Low Code Umfeld ist das nicht so, man macht einfach. Ich kann doch schnell a Youtube, a LinkedIn Learning. Dann beginnt man dann. Der dritte Punkt ist. Er macht meistens auch keine Benutzer Dokumentation. Es ist ja intuitiv. Jeder versteht es. Ja, ist ja logo. Und der vierte Punkt ist. Er macht wahrscheinlich. Kein Test Konzept. Also der ganze Software Entwicklungsprozess ist eher amateurhaft. Das muss nicht heißen das ist schlecht, aber das ist der Unterschied. Wenn wir von einem guten guten Entwickler in einer klassischen. Einem klassischen Entwickler gegenüber einem Citizen Developer würde ich sagen der eine. Der hat seine klare Vorgehensweise. Er hat seine Richtlinien, er hat seine. Vor allem hat er Erfahrung. Er hat vor allem jahrelange Erfahrung. Und bei Low Code Entwickler sagt Das ist eine

glatte, eine tolle Sache, das mach ich doch. Und das lerne ich dann noch. By the way. Und dann schaut man so ein bisschen YouTube Videos, fragt zwei, drei Kollegen, geht irgendwo in einen Chat, sagt Ah, das kann ich nicht lösen. Wie könnte man das lösen? Er ist viel, er ist viel lockerer und unstrukturierter unterwegs. Und das Resultat selber ist eben auch unstrukturierter, als wenn ich. Wenn ich eine Lösung, eine eine klassische Applikation anschaue, dann weiß ich genau, wo ich was nachschauen muss. Also im Sinne von Wie ist das Programm aufgebaut, was ist User Interaktion? Was ist Backend? Also was ist Front End? Was ist Backend? Das ist mehr oder weniger einfach, klar und man weiß es und man weiß wo man schauen muss, in welchen Bibliotheken ich noch was schauen muss. Und bei der Low Code Entwicklung ist es, ich würde mal sagen. Unstrukturiert. Unklar. Unübersichtlich. Also, wenn ich etwas nachschauen muss, jetzt speziell in der Umgebung, die wir haben, trifft bei anderen aber auch zu. Dann kannst du nicht einfach den Code anschauen und durchgehen und den tracen. Im Sinne von top down oder so, auch wenn es Objekt orientiert ist. Es ist. Du musst implizit wissen, wo was versteckt ist, damit du das Problem finden kannst. Das habe ich bei der klassischen Entwicklung nicht. Dort ist es offensichtlich. Genau. Ich würde sagen, das sind die die größten Unterschiede meiner Meinung nach.

Speaker1: [00:12:48] Und welche Herausforderung siehst du, wenn du an einen End-User Development / Citizen Development approach siehst?

Speaker2: [00:12:59] Wenn ich das jetzt kritisch sehe. Was ich aber nicht tue. Aber wenn ich's kritisch sehen würde, würde ich sagen Das Problem ist, es gibt Wildwuchs. Jeder macht etwas. Eine Funktion wird drei Mal entwickelt. Mehr oder weniger genau das Gleiche. Ich habe keine. Ich habe kein zentrales Dictionary, wo ich etwas nachschauen kann. Und dann sage ich Ah, das nehme ich dort. Diese Funktion nehme ich da, und zwar eine. Man macht immer alles selber neu aus. Das Rad wird jedes Mal neu erfunden. Das ist eigentlich ineffizient. Die Qualität ist nicht genügend hoch. Es ist kein Betrieb sichergestellt. Es ist auch kein Ah, der Mitarbeiter wechselt die Stelle. Ja. Wer betreut die Applikation jetzt? Es gibt keinen Nachfolger, der sie betreut. Also, von dem her ist es. Trau dich. Und gefährlich. Sehr negativ gesehen. Halte schon gefährlich. Aber es kommt sicher noch die Frage, was für Vorteile es ist.

Speaker1: [00:14:05] Wurde die Überleitung schon gefunden. Welche Vorteile siehst du?

Speaker2: [00:14:10] Also der Vorteil ist, ich muss nicht warten bis die IT Lust hat mein Problem zu lösen. Weil wenn es nicht Prio eins ist und nicht strategisch, wird es nie gelöst in diesem Unternehmen. Ich kann es selber in die Hand nehmen. Ich kann das Tempo selber bestimmen. Ich lerne für mich selber. Ich muss. Der Aufwand ist viel geringer. Ich brauche weniger Kommunikation. Ich muss mich nicht absprechen. Ich muss keine PiS durchlaufen, damit ich dann vielleicht eventuell nach zwölf Monaten etwas erhalte oder jemand auf mich zukommt. Sagt, jetzt hätten wir gerade Zeit. Möchtest du noch sagen Nein, der Zug ist abgefahren, wir haben so implementiert, ich kann es leicht anpassen. Ich habe. Ich bin. Problem Problemlösung bin ich viel flexibler, viel flexibler. Und ich kann auch für mich etwas, für mich und die Zukunft einen added value herausholen. Ich lerne etwas, ich verstehe etwas besser. Ich bin mehr wert auf dem Markt. Ich werde vielleicht von Kollegen und Kolleginnen angesprochen. Könntest du uns. Hey! Lässig die fragt mich, ob ich kann. Ich kann. Logisch. Außer man ist ein Narzist und Egoist. Dann will man nicht. Aber ein Narzisst würde wahrscheinlich auch kein Low Code schreiben. Der lässt es schreiben. Ja, ich bin viel schneller am Markt, wenn das Thema, wenn das Thema jetzt wäre time to market, was es ja seit Jahren immer ist. Dann würde ich sofort auf Low Code gehen. Viel schneller, viel günstiger und natürlich auch viel günstiger. Viel günstiger, viel günstiger. Faktor zehn. Das wäre ein Thema. Das wäre ein Argument.

Speaker1: [00:16:18] Aber welche? Welche Maßnahmen siehst du, um den Herausforderungen zu begegnen?

Speaker2: [00:16:25] Also zuerst müssten wir mal bei uns ein sogenanntes Operation Modell definieren. Wie wollen wir überhaupt mit so einem Thema umgehen? Da müsste sich jemand berufen fühlen und sagt Hey, wenn wir das schon einsetzen, komm, lass uns darüber nachdenken, wie wir mit dem umgehen wollen. Es ist Software mit Funktionalität. Wir haben Datenschutz Themen. Und so weiter und so fort. Also braucht es ein Betriebssystem. Das wäre das erste. Das zweite wäre vielleicht. Dann komm, lass uns eine Community gründen. Wer macht da alles mit? So wie wir Communities haben bezüglich Architektur oder bezüglich Gilden bezüglich Sicherheit würde ich meinen, wäre das auch etwas für das Unternehmen. Eine Community Power oder Low Code, nennen wir es Low Code. Wir haben ja auch noch andere Tools. Deutschland hat diverse Tools im Einsatz, sind eher professioneller Art, sind zwar auch low code, aber.

Werden. Der Low Code wird durch klassische Entwickler erstellt. Die Gegensatz zu uns dann auch. Das Unternehmen Y hat auch ein Tool im Einsatz. Das haben wir auch evaluiert. Er hat einige Lösungen selbst geschrieben. Die Frage ist, wie zukunftsträchtig. Und so weiter und so fort. Das müsste man gesamthaft anschauen, ohne gleich wieder ein riesen Schiff auf das Meer zu schicken, sondern sagen Hey, wir haben jemanden, der kümmert sich um das Thema, der schaut, man macht Interviews.

Speaker2: [00:18:03] So wie du jetzt ein Interview machst, notiert sich das kann logisch Denken, erarbeiten, ein Konzept. Wie könnten wir das? Tiefe Betriebskosten, keine großen Einschränkungen, aber trotzdem ein paar Leitplanken bezüglich Namensgebung bezüglich Daten, Zugriffe bezüglich Security. Und so weiter und so fort. Und darum würde ich meinen, wäre das ein ein Top Top Ding um. Kleinere, eben nicht klassische Risiko Applikation, wo man Jahre investiert wurden. Es wird nie mit einer Low Code Lösung machen. Jetzt heutzutage würde ich das nie mit einer Low Code Lösung machen. Das ist schon vor 20 Jahren gescheitert. Hat man das schon gemacht? Mit Generatoren, dann aber kleinere Applikationen, wo man sagt, man möchte schnell auf dem Markt sein. Wir haben ein kleines Projekt, wir möchten eine Lösung haben, wir wollen einen Prototypen, wir wollen einen Prototypen. Ich baue den Prototypen mit dem Fachbereich zusammen und dann gehe ich hin und sage Das ist das, was wir brauchen. Nur möchte ich es. Professionell, stabil, performant, Security eingebaut. Da müsste man etwas investieren. Zeitmäßig. Und ein entsprechendes Betriebs modell aufbauen, definieren und aufbauen.

Speaker1: [00:19:23] Wo? Wo siehst du die Grenzen eines Citizen Development?

Speaker2: [00:19:29] Auch die sind durch ihn selbst gesetzt. Also die meisten können das gar nicht. Im Sinne von Jeder kann das Ganze: vergessen. Jeder kann gar nicht. 2/3 kann nicht einmal prinzipiell. Das ist er. Das ist der Darwinistische Auslese. Die können nicht. Das kannst du streichen in deinem Protokoll. Nein, aber siehst so, du musst. Du musst. Wirklich? Logik muss dir etwas bedeuten und ich meine nicht logischer Entscheid im Sinne von das kann noch jeder ein logischen Entscheid. Ich meine wirklich mathematische Logik muss dir etwas, das muss dir etwas sagen, mit dem musst du umgehen können. Datenstrukturen musst du verstehen. Daten Modellierung musst du auch ein bisschen verstehen. Sonst kaum. Auch wenn wir mit einer flachen Liste arbeiten. Oder vielleicht mit zwei Listen ein bisschen Du musst IT-

affin sein. Das ist mal die die Eingrenzung. Wer kann denn überhaupt? Natürlich kann jeder. Aber ich möchte dann das auch nicht anschauen. Oder es ist extrem ineffizient oder jeder kann und dann verblutete er einfach Stunden für irgendetwas, das wirklich nicht viel ist, wo man, wenn einer etwas wüsste, würde sagen, das mach schnell mit Excel und zwei Formeln, dann habe ich das auch. Nein, ich baue jetzt auch eine Power App. Ich bin auch dabei. Er. Man müsste aber den Leuten trotzdem den Freiraum geben und sagen Mach dir mal Gedanken. Wäre das etwas? Genau. Und von der Applikation her würde ich ganz klar sagen sobald es einem Richtung wirklich operativ geht und nicht nur Auskunft System, vielleicht so ein kleiner Abruf Workflow. Ist es okay wenn der 500 Franken ausbezahlt kriegt, darf der nun eine eine neue Maus bestellen kann? Oder in die Ferien gehen? Wenn es wirklich. Wenn es wirklich um Business geht. Um Versicherung bei uns jetzt. Wenn ich sage Das ist die Grenze. Das geht nicht mehr. Das würde ich nie. Das würde ich nicht mehr erlauben. Genau.

Speaker1: [00:21:43] Wenn du in einer Kombination denkst, dass man zum Beispiel sagt, man mischt noch pro Developer rein, also Leute, die programmieren können als Tandem mit, dann würden sich die Grenzen verschieben, oder?

Speaker2: [00:21:56] Ja, ganz klar. Und die Grenzen würden sich ganz klar verschieben. Dazu kommt, dass du die sowieso brauchst, sobald du nicht mehr ganz trivial bist. Also in all den Projekten, die wir gemacht haben, brauchte ich Support von entweder einem Java Entwickler, von einem Mann, der weiß, wie die heutige Welt aussieht in der Informatik. Und ich brauchte auch Support auf der administrativen Seite von BITS, wo es darum ging, etwas zu berechtigen. Hast du daran gedacht? Die Benutzer können gar nicht zugreifen, du musst sie freigeben und so weiter wie wollen wir das gestalten. Die Integration bei uns im Intranet als Beispiel. Das kann nicht jeder. Da brauchst du automatisch Support. Du brauchst Support. Das sowieso. Jetzt ist die Frage Wie weit geht der? Ich könnte mir gut vorstellen, dass man zum Beispiel sagt Komm, wir machen ein Frontend mit so einem Tool. Auch. Das gewisse Dinge kommen. Wir könnten ein Problem sein in ein einen Business Workflow einbauen und hinten hinten geht eine Profi Software und holt die Daten und bereitet die Daten sauber auf und stellt sicher, dass die Konsistenz in unseren operativen Systemen da könnt ich mir eine Zusammenarbeit gut vorstellen und auch eigentlich den gesamten Nutzen erweitern.

Speaker1: [00:23:30] Hast du in der Vergangenheit mit SCRUM gearbeitet?

Speaker2: [00:23:37] Ähm. Nein. Nicht ich nicht. Ich nicht. Aber. Ich kenne ja die Teams, die arbeiten mit SRUM. Der externe Entwickler, der hat auch mit SCRUM gearbeitet, also mit seinen 14 täglichen Sprints, mit Daily und Weekly und Reviews und Retro und weiß der Teufel was auch immer. Der hat das gemacht. Und ich habe mich nur dort eingeklinkt, wo ich gesagt habe, das bringt mir etwas. Ich selbst arbeite nicht nach Scrum.

Speaker1: [00:24:08] Okay. Die Frage, die sich für mich noch stellt Macht es Sinn, einen SCRUM Framework für Citizen Development zu nutzen?

Speaker2: [00:24:19] Das kann ich dir nicht beantworten diese Frage. Ich kann es auch nicht. Nicht einmal erahnen. Ich weiß nicht, was das bedeuten würde, weißt du? Okay. Was? Was bedeutet das? Ein Framework. Ein Framework, dass sie sich jeden Tag treffen und sagen Gestern hatte ich und heute werde ich?

Speaker1: [00:24:37] Genau. Es gibt halt einfach Gefäße oder Rollen zum Beispiel. Oder wir können bei den Rollen anfangen. Ich denke, die kennst du zum Teil auch. Da gibt es den Product Owner, der ist für das Produkt und für die Priorisierung verantwortlich. Dann gibt es den Entwickler, dann gibt es den Business Analysten. Denkst du, das wäre identisch oder müsste man die Rollen zum Beispiel neu denken? Braucht es vielleicht andere Rollen?

Speaker2: [00:25:04] Nein, das würde ich nicht machen, weil das, wie soll ich mal sagen, unser Modell oder Safe von SCRUM oder die sind bei uns definiert und werden ja über das ganze Unternehmen ausgerollt. Ja, ich würde jetzt nicht beginnen, noch weiter zu unterteilen und noch mehr zu definieren, für was denn diese Person oder diese Person, die dann diese Rolle hat. Genau. Und so weiter. Das würde ich nicht machen. Ich glaube, das ist bringt keinen Mehrwert. Dazu würde ich auch sagen, wenn du jetzt genau die klassischen Rollen nimmst oder du hast einen Business Analysten oder Business Engineer und du hast einen System Engineer. Der Citizen Development macht ja gerade beides. Er macht beides. Er macht die Business Analyse und sagt Oh doch, das könnte ich so und so und dann macht er low code. Ein Product Owner braucht er nicht für ein Backlog, aber er hat seine Pendenzen Liste, seine, sein eigenes

Backlog und er entwickelt ja im Normalfall alleine. Es ist ja nicht so, dass bei einer kleinen Applikation drei Leute sich die Arbeit teilen und die dann jeden Tag sagen Ich habe mal, Modul X gemacht. Hast du auf dem Modul Z gearbeitet, können wir die Schnittstelle gemeinsam testen? Das ist ja dann eher unwahrscheinlich. Ja, und der Product Owner, das ist auch ganz einfach. Das ist einfach der, der den Wunsch hat im Fachbereich. Der Product Owner müsste ja sowieso im Fachbereich sein. Original? Ja, wir lernen das ja jetzt auch im Unternehmen und sind dabei, das auch so umzusetzen. Das ist ja irgendwo in der klassischen Art wäre das so so ein Abteilungsleiter oder so ein Fachspezialist, ddr sagt, das müssen wir unbedingt lösen, sonst kann ich mit meinen Brokern nicht zusammenarbeiten. Oder ich kann auf der General Agentur, auf der General Agentur kann der Verkaufsleiter mit seinen Agenten nicht gut, ich muss noch das zusammenführen. Dann kommt der Verkaufsleiter oder der General Agent und sagt dann zum dynamischen Agenten, der gerne noch etwas in IT machen würde. Kannst du mal eine Softwareentwicklung? Und da beginnt er einfach. Also ich glaube nicht, dass da SCRUM, Scrum Definitionen irgendetwas optimieren würde. Sonst plötzlich kommen wir noch auf die ganz interessante Idee: Komm, lass uns doch den Aufwand in Story Points schätzen. Ich schätz doch kein Aufwand, wenn ich Citizen Developer bin. Ich beginne und ich habe noch keinen Citizen Development gesehen. Der Aufwand geschätzt hatte ich jetzt im Traum bin ich fertig? Ja, genau. Also, wenn Sie jetzt sagen, das ist unprofessionell und das andere ist professionell, das kann man anschauen, wie man will. Genau.

Speaker1: [00:28:05] Welche Gefäße für die Zusammenarbeit. Dennoch würdest du sehen? Also. Wie könnte man sagen, dass man dennoch das Gemeinsame hat?

Speaker2: [00:28:16] Eben ich. Ich sage es. Thema Community Ich würde wirklich eine offene, eine ich meine, wie es ja in der ganzen Welt außerhalb der Firma, wenn ich ein Problem habe, sei es mit dem Auto, sei es mit der HiFi Anlage, sei es mit mit meinem Studium oder ich möchte ich möchte Gitarre lernen. Was mache ich? Ich gehe zu diesem doktor, den es da gibt, aus Amerika oder tip etwas sein. Dann finde ich ein forum mit gleichgesinnten und und dort kann ich meine sorgen und meine ideen kann ich einbringen und das wäre sicher etwas, das es braucht. Was für Fragen habe ich? Wir wollen ja voneinander lernen oder was für Fragen habe ich? Hat jemand eine gute Antwort? Kann mir jemand einen Tipp geben, wie ich schneller oder einfacher? Also das würde ich meinen. Das würde etwas verbessern. Aber das hat mit SCRUM oder so

überhaupt nicht zu tun. Das hat es schon vor SCRUM gegeben. Also Foren. Wir helfen uns Do it yourself, oder? Genau.

Speaker1: [00:29:19] Ja.

Speaker2: [00:29:20] Das würde ich begrüßen. Frage ist der. Wer begrüßte sonst noch?

Speaker1: [00:29:26] Die Resonanz bis jetzt war gut.

Speaker2: [00:29:30] Die Resonanz.

Speaker1: [00:29:31] Ja, es war oftmals ein Thema, dass so ein Format einen Mehrwert bringen kann.

Speaker2: [00:29:36] Ja, ja, das ist meins. Das ist meistens so das Format. Und wenn ich dann schaue und ich meine, du bist ja jung, dynamisch und du arbeitest mit Teams, oder schau mal, wie viele Channels das wir haben. Ich meine, das ist ja der. Jetzt kann ich sagen ein Chanel mehr oder weniger. Spielt ja auch keine Rolle mehr. Ha! Es ist der Wahnsinn. Aber ich würde meinen, das würde wirklich etwas helfen. Ich weiß nicht, ob du das weißt, aber die meisten. Die meisten sind ja nicht wirklich die in diesem Umfeld. Sind mit low code. ist eben nicht low code, sondern sind sind ja Power BI Benutzer. Also das steigt ja. Power BI im Sinne von ich hab so ein User End Tool, wo ich schnell eine super Auswertung und grafisch und alles pipapo machen kann. Das wird ja schon extrem stark genutzt und auch dort wäre eine Community nicht schlecht, weil die Fragen am Anfang sind immer die gleichen. Wie kann ich zwei vier Excel Tabellen zusammenführen, Daten auswerten darüber und dann richtig sortieren? Und so weiter und so fort. Das wäre sicher nicht schlecht. Genau. Und Powerplay und Power Automate. Das ist so, das gehört zusammen. Das Gesamtpaket von Microsoft ist bei uns. Oder im Sinne von Low Code oder No Code sogar. Oder sag ich ein Forum wäre gut, hilfreich, unterstützend. Er würde die Qualität auch heben. Im Sinne von ein bisschen mehr. Ein bisschen mehr Professionalität.

Speaker1: [00:31:19] Jetzt. Gibt es aus deiner Sicht noch Themen, die du grundsätzlich interessant findest, die ja für meine Forschung interessant sein könnten?

Speaker2: [00:31:33] Ja. Die Frage ist wäre zum Beispiel Wie geht denn die klassische IT mit dem Thema um? Also weshalb will sie sich abgrenzen? Weshalb hat sie Angst? Sich? Was kann sie verlieren? Eigentlich kann sie gar nichts verlieren. Sie kann nur gewinnen. Ist meine Meinung. Aber das ist eben auch so ein Change, ein Change Thema oder etwas Neues. Ist da etwas Neues bedroht? Viele sind noch in diesen Gedanken. Ich würde meinen. Das kann man optimieren, um die Gesamtleistung der Firma zu erhöhen, um schneller zu sein, um einfacher zu sein, um mehr. Es gab irgendwann mal so eine Geschichte von Ich weiß auch nicht mehr von wo, dass ich sie aber wie ein alter Witz hat. Ein hat irgendwann mal gehört. Jemand. Jemand wurde pensioniert oder es wurde jemand pensioniert. Dann hat man zwar eine Dankesrede gehalten und pipapo. Weiß ich was alles, oder? Und dann noch gesagt Ja, und was möchtest du zum Abschluss noch sagen? Und da hat er gesagt Es ist ganz einfach, ihr habt mich bezahlt. Für eine gewisse Leistung. Die habe ich gebracht. Aber ihr hättet viel mehr von mir haben können. Ich hätte auch denken können bei der Arbeit. Es wurde nicht. Gefordert. Ebenso. So. In etwa. So. Und das ist die Gefahr, finde ich, von Rollen. Also meine ganz persönliche Meinung ist, was er weglässt. Das ist die Gefahr von Rollen. Entweder werden die Rollen nicht wahrgenommen. Das sind natürlich Unterstellungen. Alle unsere Leute nehmen genau die Rolle wahr, die sie müssen. Es stimmt überhaupt nicht, ist meine Aussage. Es wäre ein tolles Streitgespräch mit so Philosophen, oder? Das zweite ist, wenn ich dann so eine Rolle habe, da ist sie ja definiert. Schaue ich dann noch über den Rand hinweg. Bin ich dann bereit, ausserhalb meiner Rolle noch etwas zu machen, breiter zu werden? Früher hat man gesagt: Du musst rund werden, du musst wie Schichten sowie wie ein Zwiebel Modell. Du lernst dazu. Es ist nicht die Leiter, die hinaufgeht, sondern du. Dein Wissen wird breiter, tiefer und breiter. Und breiter wird es, wenn ich bereit bin, Neues zu wagen, was vielleicht mit meiner Rolle eben nichts zu tun hat. Ich überschreite die Grenzen und mache. Ich mache, heißt es bei uns im Unternehmen ja auch immer: machen und nicht fragen. Aber wenn ich so klare Rollen habe, sehe ich die Gefahr, dass die Leute sich hinter der Rolle verstecken und sagen Jetzt not my job. Das HR hat gerade gesagt: Komm, Fun bei der Arbeit, ich gehe jetzt bei einem Kollegen einen Kaffe trinken. Hab ein bisschen Fun. Ich bin der Satiriker, also ich. Ich würde meinen, man könnte einiges verbessern oder verbessern. Ist das falsche Wort. Erweitern. Schneller werden. Besser werden. Mehr Mut haben, etwas auszuprobieren. Was du auch mit der klassischen IT fast nicht ganz. Das kostet immer gleich. Ich brauche ein Budget. Oder ich brauche. Ich brauche

ein Budget und brauche drei Leute, die mir helfen. Oder Das hast du in diesem Umfeld nicht. Da kannst du. mit Eigeninitiative kannst du einfach arbeiten. Das ist das Schöne und das Schnelle. Aber du musst ein Umfeld schaffen, wo das auch, ein bisschen gefördert wird. Ja. Du kannst es zu Tode regulieren oder du kannst es fördern. Das kommt drauf an, wer ist an der Macht und sagt in welche Richtung das wir gehen wollen? Meine Meinung ist klar, in welche Richtung das mehr gehen müsste.

Speaker1: [00:35:45] Dann würde ich hier die Aufzeichnung beenden.

Speaker2: [00:35:48] Ja. Gut für mich.