Interview 6

Speaker1: [00:00:02] Hallo, Interviewpartner. Herzlichen Dank für die Bereitschaft des Interviews. Ist es für dich in Ordnung, wenn ich das Interview aufzeichnen?

Speaker2: [00:00:13] Hallo, Pascal. Selbstverständlich.

Speaker1: [00:00:15] Das ist sehr nett. Dann denke ich, lass uns direkt beginnen. Ich möchte kurz eine Einleitung machen und dir sagen, um was es geht. Und dann ist der Grundgedanke, dass es ein Semi-strukturiertes Interview ist. Das heißt, ich habe einige Fragen, die ich gerne stellen möchte. Dennoch können wir auch eine offene Diskussion führen. Das heißt, wenn du eigene Punkte hast, die du gerne einbringen möchtest und nicht im Rahmen von Fragen abgedeckt sind, ist das jederzeit willkommen. Ich untersuche das Themenfeld der Low-Code- Softwareentwicklung vor allem durch den End-User und möchte schauen, inwiefern dies in einem SCRUM Modell überhaupt sinnvoll abgebildet werden könnte und welche Herausforderung dabei entstehen. Jetzt zu Beginn kannst du mir deine aktuelle Position und auch deine Arbeit beschreiben.

Speaker2: [00:01:11] Ich arbeite seit viereinhalb Jahren bei Unternehmen X, habe damals begonnen als Softwareentwickler im Bereich, habe dann nach zwei Jahren gewechselt zu den Projektleitern und habe da verschiedene Projekte auch durchgeführt, geleitet als Projektleiter. Und nun seit Anfang des Jahres 2022 bin ich als Head of Application Platform verantwortlich für das ganze Thema Low Code bei Unternehmen X. Sowohl für den marktseitigen Aufbau als auch für unsere Agenturen, wo wir im Moment ein Team aus sechs Personen sind und stark am wachsen.

Speaker1: [00:01:43] Sehr schön. Jetzt. Welche Erfahrungen hast du bereits im Bereich der Low-Code- Softwareentwicklung? Bzw. Wann kamst du auch das erste Mal mit Low Code Software in Kontakt?

Speaker2: [00:01:54] Das war im Jahr 2020. Vermutlich gar nicht mehr. Ganz sicher. Aber ja, 2020 muss es gewesen sein, als wir die ersten Low Code Projekte bei Unternehmen X durchgeführt haben. Da war ich als Projektleiter mit Low-Code in Berührung, habe da auch bereits in der Angebots Phase unterstützt gehabt bei den Projekten und habe dann auch verschiedene Projekte geleitet. Wir haben da vor allem

Kunden Portale gebaut auf Basis von Low Code, namentlich von Bendix. Seitdem begleitet mich das Thema und im Moment ist es auch Vollzeit mein Job.

Speaker1: [00:02:35] Inwiefern unterscheidet sich aus deiner Sicht die Low-Code-Softwareentwicklung von einer klassischen Softwareentwicklung?

Speaker2: [00:02:50] Ich denke, da gibt es verschiedene Aspekte, die man betrachten kann. Low Code Entwicklung beginnt nicht eigentlich bei der Technologie selbst, sondern sogar noch einen Schritt weiter vorne als ein Paradigma, bei dem Low Code eigentlich die zusätzliche Abstraktion und Vereinfachung der Softwareentwicklung bieten soll. Im Vergleich zu einer Hochsprache wie C plus plus Sharp. Dies hat verschiedene Einflüsse. Beispielsweise hat das Einflüsse, wie die Applikation entwickelt wird, aber auch beispielsweise, was für organisatorische Maßnahmen notwendig sind, um diesem Paradigma gerecht zu werden. Du hast vorher gesagt, in deinem Beispiel geht es vor allem um die Entwicklung von Lösungen durch den End-Anwender. Da hat man beispielsweise Aspekte wie Governance Modelle, beispielsweise Aspekte wie Datensicherheit, die alle eigentlich führt, und das Paradigma zu ermöglichen und Low Code Entwicklung zu ermöglichen, berücksichtigt werden müssen. Also die Technologie, die Low Code Plattform selbst ist ein Teil davon. Aber das hat durchaus Aspekte, die auch in die Organisation reingehen. Allen gemeinsam ist natürlich, dass die Plattform im Gegensatz zu einer Entwicklung mit einer Hochsprache einerseits eine gewisse Standardisierung der Technologie zur Folge hat, auf der anderen Seite auch ein Baukasten bietet aus verschiedenen Aspekten, um eine Lösung zu bauen und bereits gewisse vorgefertigte Strukturen da sind und nicht wie bei einer Anwendungsentwicklung mit einer Hochsprache von Beginn an aufgebaut werden.

Speaker1: [00:04:41] Das heißt, ihr habt einerseits stelle ich mir vor, ihr habt klassische Entwickler, die jetzt Low Code Software entwickeln. Habt ihr jetzt auch schon Erfahrungen gemacht mit dem Thema, dass der End-Nutzer etwas selbst entwickelt?

Speaker2: [00:04:59] Bei Projekten, die wir bei Unternehmen X durchgeführt haben, haben wir keine sogenannten Citizen Development im Einsatz gehabt, sondern haben verschiedene Ingenieure aus unserem Team, die dann die Low Code Plattform eingesetzt haben. Und wir profitieren da natürlich von Aspekten wie beispielsweise schnellere Anwendungsentwicklung, weil beispielsweise vorgefertigte Bausteine zur

Verfügung stehen, eine homogene Landschaft, was die Bild Infrastruktur angeht, die Umgebungen, die bereits zur Verfügung stehen durch die Applikation. Ich denke für das Thema. End-Anwender-Entwicklung gilt es vor allem auch den konkreten Use Case zu betrachten. Die Lösungen, die wir gebaut haben, die haben relativ eine hohe Komplexität. Das unterscheidet sich wahrscheinlich von einer Lösung, die du mit einer Anwendung bauen willst, weil wir eigentlich wirklich Applikationen bei und die Enterprise Applikationen Business Prozesse digitalisieren und das Citizen Development setzt eigentlich auch an einem anderen Ort an, wo es eigentlich darum geht, einfachere Lösungen für und mit Endanwender zu bauen oder dass es ein kombiniertes Team ist aus Ingenieuren und Endanwender.

Speaker1: [00:06:21] Und wo siehst du die Grenzen eines Citizen Development? Also wenn man darüber nachdenkt oder man Verkaufs-Folien, wo man das Thema präsentiert, der Nicht IT Business Business Mitarbeiter, der dann eigentlich selbst Prozesse oder Applikationen entwickelt. Wo siehst du jetzt persönlich eine Grenze eines solchen Cases?

Speaker2: [00:06:47] Ich denke, da kommt es auf die Priorität drauf an, also die Einführung beispielsweise von Citizen Development beginnt nun erstmals bei Unternehmen X. Wir bieten nicht nur Implementation Projekte an, wir haben auch verschiedene sogenannte Advisory Services im Bereich Low Code, wo wir beispielsweise erst mal gucken, ob Low-Code überhaupt ein Mehrwert für dich als Organisation bringt. Da gucken wir beispielsweise auch das Thema Citizen Development an und bevor wir überhaupt zum Punkt kommen: Ja, lass mal Citizen Development machen. Geht es da darum, einerseits herauszufinden Gibt es den Use Cases, bei denen du davon profitieren kannst, dass du hast du quasi beispielsweise ein Backlog von Ideen, die du mit Citizen Development lösen kannst, lösen willst? Und was sind die gegenüber gestellten Kosten? Dazu eine der Kosten. Und da kommen wir vielleicht darauf zurück. Was du gefragt hast, ist die initiale Ausbildung der Mitarbeiter sowie die Begleitung der Mitarbeiter kontinuierlich. Wenn man Citizen Development einführen will und betreiben will, muss man eine Investition tätigen, sowohl um die Leute zu schulen, als auch kontinuierlich die Leute weiterzubilden. Und dann ist natürlich die Frage: Wie weit treibst du das? Wenn du entsprechend viel investierst, kannst du entsprechend komplexe Applikationen in Zukunft mitunter wahrscheinlich nicht ausschließen, aber mitunter auch mit Citizen Developer lösen. Aber das ist

natürlich eine Frage des Return on Investment. Wie viel willst du investieren, um die Leute zu befähigen, diese Lösungen zu bauen?

Speaker1: [00:08:22] Das heißt, du sagst, man braucht eine gewisse Mentalität und komplexere Dinge abbilden zu können.

Speaker2: [00:08:27] Definitiv. Also ich denke, beginnen kann man mit sehr einfachen Applikationen, die auch weniger Business kritisch sind. Da kommen wir dann auch in den Bereich Governance rein, wo es darum geht, nicht nur quasi sicherzustellen, dass beispielsweise Daten manipuliert werden etc. Vielleicht ist die erste Applikation die ein Unternehmen mit Citizen Development baut mal eine Applikation, die ausschließlich Daten konsumiert, ausschließlich Daten liest, wo man dann über die Governance beispielsweise sicherstellt, dass die Entwickler gar nicht fähig sind, Systeme zurück zu schreiben, um mit zunehmendem Seniorität der Entwickler. Citizen Development kann natürlich auch diese Grenze schieben und kann dann auch komplexere Lösungen mit mehr Impact auf die Organisation entstehen. Das muss allerdings sicherlich begleitet werden, beispielsweise durch eine interne IT. Es gibt auch das Modell, zum Beispiel solche Champions zu bauen von Citizen Development, die quasi sich weiterentwickeln und dann auch zusätzliche Rechte erhalten im Vergleich zu anderen diesen Developer, weil sie selbst auch mehr ausgebildet sind.

Speaker1: [00:09:36] Was denkst du? Wo siehst du die größten Herausforderungen bei dem Citizen Development Ansatz?

Speaker2: [00:09:42] Ich glaube, du hast schon den einen der größten Punkte genannt im Moment. Das Thema ist sicherlich etwas, das im Markt sehr präsent ist und das durchaus auch viel Mehrwert bieten kann. Allerdings ist es auch ein großes Verkaufsargument, ich glaube mal erst mal herauszufinden. Ist das überhaupt ein Mehrwert für meine Organisation und kann ich das in meiner Organisation auch lösen und will ich das auch einführen? Will ich diese Investition tätigen? Ich denke, da beginnt bereits mal erst die Frage, bevor man überhaupt danach darüber diskutiert. Wo liegen die weiteren Herausforderungen? Diese zentrale Frage zu klären Will ich diese Investition tätigen, auch mit all den Folgeerscheinungen? Beispielsweise muss ich eben auch mir überlegen, was für ein Center of Excellence baue ich vielleicht auf, wo ich dann auch Leute schule, wo ich beispielsweise eben auch die Governance Strukturen

drin habe? Ist diese Investition wert überhaupt den Return on Investment ich dann davon habe? Ich denke, das ist eine der zentralsten Fragen. Und danach geht es dann eigentlich in die Umsetzung. Und da ist es dann sicherlich eine der großen Herausforderungen, wie motiviert du die Leute, diese neuen Tools auch zu nutzen? Ist Change Management dabei und gleichzeitig auch danach? Wie bilde ich die Leute eigentlich aus, die vielleicht überhaupt kein IT Background haben und stelle da auch die Qualität, sicher die Lösungen, die sie bauen. Und verhindere dabei auch, dass ich Insellösungen oder Schatten-IT-Themen habe. Ich denke, das sind so die größten Herausforderungen im Bereich Citizen Development. Den einen Dingen, die ein Unternehmen haben kann bei der Einführung und dann auch beim Betrieb.

Speaker1: [00:11:22] Und welche Maßnahmen siehst du, um diesen Herausforderungen zu begegnen?

Speaker2: [00:11:30] Klar von Anfang an sich erst einmal Gedanken machen und nicht überstürzt in das Abenteuer reinzugehen. Und danach überrascht sein, dass man eigentlich doch sehr viele Probleme sich eingehandelt hat. Ich denke, das ist allgemein ein Thema. Low-Code ermöglicht es sehr schnell Erfolge zu erzielen. Du musst auch sehr viel weniger, je nach Plattform sehr viel weniger Wissen haben, was Software Engineering im Allgemeinen bedeutet, um erste Erfolge zu erzielen. Das hat allerdings gleichzeitig zur Folge, sowohl im Citizen Development als auch wenn IT Personen Lösungen bauen, dass man entsprechend schnell erste Fehler machen kann, die dann auf längere Zeit sich fatal auswirken. Und ich denke, da ist es entsprechend wichtig, dass man und das sehen wir ja auch, dass man sich von Anfang an klaren Plan zurechtlegt. Was will ich eigentlich erreichen, wo sehe ich die Hürden? Was muss ich für vielleicht auch Maßnahmen im Voraus treffen? Und erst dann strukturiert starte. Es gibt ein gutes Beispiel eines großen Schweizer Unternehmens, wo sich das Citizen Development sehr stark etabliert haben. Aber die bestätigen beispielsweise, dass sie im Moment ganz viele Schulden Abbauen müssen, die sich am Anfang eingehandelt haben, weil sie. Wir legen mal los und jetzt sind da ganz viele tolle Applikationen, die von den Business Usern auch wirklich genutzt werden. Aber wo die IT jetzt danach Stück für Stück wieder die Applikationen mit begleiten muss, um sie beispielsweise zu migrieren, um gewisse Schulden abzubauen. Und wenn man da von Anfang an einen klaren Plan hat und das auch strukturiert und auch begleitet als als zentrale IT eines

Unternehmens, dann kann man gewisse Fehler verhindern, die dann hintennach sonst große Kosten verursachen.

Speaker1: [00:13:28] Ja, ist wirklich ein spannender Punkt. Ich versuche jetzt, das ganze Thema auch ein bisschen in Kombination mit SCRUM zu bringen. Was denkst du generell, wenn du darüber nachdenkst, dass ein Citizen Developer Applikationen entwickelt, vielleicht auch in einem Teilzeit Pensum? Inwiefern denkst du, es ist sinnvoll, so etwas in einem SCRUM Prozess abzubilden?

Speaker2: [00:13:55] Grundsätzlich denke ich, dass das sehr gut möglich ist. Wenn du wirklich auch eine Applikation von einer gewissen Größe baust, also die vielleicht eine gewisse Zeit hat, dann ist es sicherlich sinnvoll, auf bewährte Methodiken wie zum Beispiel Scrum zurückzugreifen. Scrum bietet dir ja vor allem die Möglichkeit, in relativ kurzen Iterationen jeweils neue Versionen eines Produkts zu bauen. Und diese Methodik minimiert ja vor allem Risiken. Quasi, dass du eine große Applikation baust, die aber nie fertig wird. Und ich denke, das ist ja unabhängig von wem. Die Applikation baut erst mal ein sehr guter Ansatz, um die Risiken da zu minimieren, dass man ein Riesenprojekt baut, das dann aber irgendwo gar nie läuft. Oder vielleicht. Gar keinen wirklichen Business Nutzen erzielt. Also das geht sicherlich mal grundsätzlich gut miteinander zusammen und passt eigentlich auch sehr gut zu der Philosophie von Low Code, dass man eigentlich noch schneller Resultate erzielen will, indem man beispielsweise eben auch diese Brücke einreißt oder diese Barriere einreißt zwischen IT und Business, was genau auch die Idee für Citizen Development bis zu einem gewissen Grad ist.

Speaker1: [00:15:13] Und wenn du jetzt an die die Rollen denkst. In einem klassischen Scrum Team hast du Product Owner hast du ja Business Analysten etc.. Also es gibt ja klassische Rollen. Denkst du die bestehenden Rollen reichen? Müsste man Rollen neu denken?

Speaker2: [00:15:32] Ja, ich denke. Im Endeffekt, wenn ich mich noch richtig an den Scrum Kurs erinnere, dann gibt es einerseits das Entwicklerteam und andererseits den Product Owner und noch den Scrum Master. Die kann man dann verschieden besetzen. Man kann sicherlich auch beispielsweise Business Analysten dazunehmen oder so, aber grundsätzlich das Scrum Team selbst besteht ja mal aus dem Product

Owner plus die Entwickler und den Scrum Master. Was sowieso üblich ist und eigentlich Scrum auch nicht verbietet ist auch, dass man zumindest den Scrum Master und die Entwickler zusammenlegt, dass ein erfahrener Entwickler auch die Scrum Master Rolle mit übernimmt. Das heißt, wenn die entsprechenden Personen vom SCRUM Team geschult sind, also beispielsweise die auch die Citizen Development geschult sind, dann spricht da mal grundsätzlich nichts dagegen. Ich denke aber, dass ist natürlich auch eine gute Chance, beispielsweise da ein kombiniertes Team zusammenzustellen von jemandem, der natürlich auch mehr technische Expertise mitbringt, vielleicht auch einen technischen Background hat, als Entwickler, als Pro Developer Quasi und Citizen Developer, der dann die Fachlichkeit reinbringt und gleichzeitig aber auch gerade die fachlichen Themen umsetzen kann. Das ergänzt sich sicher sehr gut. Wer dann am Schluss der Product Owner ist, das kann durchaus vielleicht eine spannende Frage werden, ob das dann gleichzeitig auch der Citizen Developer ist oder ob das jemand anderes ist. Ob das vielleicht auch der Auftraggeber des Projektes ist, das man umsetzen will. Ich denke, da gibt es dann pro Fall ist die Frage, wer da dann zuständig sein soll.

Speaker1: [00:17:11] Ich finde auch das Thema das schätzen von von Entwicklungsaufwänden durch einen Citizen Developer könnte noch spannend sein. Wie siehst du das?

Speaker2: [00:17:27] Das Schätzen von Entwicklungsaufwänden ist immer eine spannende Sache. Nicht nur bei Citizen Development. Ich glaube. Das ist eine Erfahrungssache, die jetzt mal unabhängig davon ist, wer die Schätzung macht, ob er jetzt aus dem Business oder aus dem IT Hintergrund kommt. Ist Schätzen von Entwicklungsaufwand etwas, das vor allem ein eine Abschätzung ist von schon gemachtem Transferieren auf etwas Neues? Wenn also der Citizen Development schon viel Erfahrung gesammelt hat und die Plattform gut kennt und quasi den Use Case, den er jetzt neu umsetzen will, transferieren kann auf etwas, das schon immer gemacht hat, kann seine Schätzung vermutlich ähnlich oder gleich gut liegen wie die von einem Developer, der ebenfalls schon mal etwas Ähnliches gemacht hat. Und da ist wahrscheinlich die Erfahrung das Entscheidende mehr als der Hintergrund. Und das er oder sie schon mal etwas in diesem Umfeld gemacht hat, das ähnlich ist, vergleichbar ist.

Speaker1: [00:18:34] Wenn du an die Priorisierung denkst, wenn du eine Handvoll Citizen Developer hast. Normalerweise aus dem gemeinsamen Produkt Backlog. Wie könnte man das vielleicht gestalten?

Speaker2: [00:18:55] Das Priorisieren der einzelnen Backlog Items?

Speaker1: [00:18:58] Genau. Was man dann wirklich umsetzt.

Speaker2: [00:19:02] Ja, ich denke da ist SCRUM dann relativ klar eigentlich. Zumindest was die verschiedenen Sprints zu planen angeht, ist das des Products Owners seine Aufgabe. Also auch wenn du jetzt verschiedene Citizen Developer in einem Team hast, so hat man eigentlich sollte man zumindest wenn man das Scrum Team formt, sich überlegen, wer da die Product Owner Rolle hat. Nämlich der, der auch das Produkt tatsächlich besitzt. Und der sollte dann auch die entsprechende Priorisierung vornehmen können. Natürlich kann das Team da beratend zur Verfügung stehen oder die Citizen Developer und die Developer. Es gibt ja auch Abhängigkeiten, unter Umständen, die vielleicht für den Product Owner nicht ersichtlich sind. Aber wer den finalen Priorisierung Entscheid trifft, ist ja bei Scrum durch den Product Owner gegeben. Das kann natürlich vielleicht durchaus zu mehr Verwerfungen führen, wenn du plötzlich sehr viele Leute hast, die das Fach sehr gut verstehen. Aber das sollte dann zumindest in einem Refinement-Termin thematisiert werden und dann auch entsprechend entschieden werden.

Speaker1: [00:20:10] Und das Thema, Plattform für Zusammenarbeit. Man hat zum Beispiel das Planning, man hat das daily, man hat die Retro. Denkst du die Gefäße reichen, sind geeignet für einen Citizen Development Ansatz? Oder denkst du, da fehlt was?

Speaker2: [00:20:28] Im Rahmen vom SCRUM Entwicklungsprozess sind das sicherlich die gleichen Events, die stattfinden müssen. Wenn du jemanden hast, der sich entsprechend mit diesen Scrum Zeremonien auskennt, dann kann man diesen Ansatz sicherlich sehr gut verwenden. Im Prinzip kannst du ja Scrum verwenden, um nicht nur Software zu bauen. Du könntest auch etwas anderes in einem SCRUM Prozess grundsätzlich durchführen. Ob das jetzt eine Veranstaltung Plan ist oder was auch immer. Grundsätzlich, wenn du das Problem in Teilproblem zerlegen kannst, dann

sollte das eigentlich irgendwie auch funktionieren. Was wahrscheinlich aber fehlt, wenn du nur die SCRUM Prozess Zeremonien hast, fehlt genau der Education Part der nötig ist bei Citizen Development und der Erfahrungsaustausch. Also ich glaube du musst wie du hast vielleicht horizontal und vertikal wäre dann quasi das, was fehlt. Nämlich da wäre dann ein Gefäß, dass eben beispielsweise durch ein Centre of Excellence auch betrieben wird, wo beispielsweise die Vernetzung zwischen den verschiedenen Citizen Developer im Unternehmen, beispielsweise auch der Erfahrungsaustausch, beispielsweise auch vielleicht wichtige Informationen. Wenn du dann an Platform Updates denkst, dann hast du vielleicht im Center of Excellence, jemand der auch die Platform Update sich ansieht, die verschiedenen Citizen Development auch brieft und ihnen sagt Hör mal, wenn diese Connector geht jetzt irgendwie auf eine neue Version, da muss man Anpassungen machen an euren Lösungen, wenn ihr den einsetzt. Und das hat ja zwar nichts mit dem Entwicklungsprozess zu tun, aber das sicherlich umso entscheidender, wenn man eine dezentrale Entwicklung hat in den verschiedenen Fachabteilungen durch Citizen Development.

Speaker1: [00:22:11] Ich denke, wir haben jetzt wirklich viel besprochen. Wir haben die Rollen angeschaut. Wir haben die Scrum Events angeschaut. Gibt es aus deiner Sicht noch Punkte, welche für meine Forschung interessant sein könnten, die ich noch nicht angesprochen habe?

Speaker2: [00:22:50] Es ist eine spannende Frage, weil das Thema selbst ist natürlich sehr breit. Ich denke, was vielleicht entscheidend ist, ist auch. Hast du dich mit den verschiedenen Plattformen oder Lösungen dieser Markt beschäftigt?

Speaker1: [00:23:06] Nicht im Detail. Als ich aktuell arbeite ich selbst vorallem mit Power Automate. Das ist ein bisschen mein mein Referenzwert. Es gibt natürlich, wenn man klassischerweise noch weiter zurückgehen würde, könnte man auch noch Makros und Excel etc. und Web basierte Anwendungen, also generell halt auch die ganze Webseiten Entwicklung. Gibt es dort noch was? Was was du sagst wäre interessant.

Speaker2: [00:23:33] Ich denke, was vielleicht da noch spannend sein kann. Es gibt eine Breite an Plattformen im Low Code Markt und ich glaube, die Komplexität der verschiedenen Plattformen und was sie eigentlich adressieren wollen, unterscheidet sich bis zu einem gewissen Grad. Die Power Platform von Microsoft versucht ein

anderes Problem zu lösen als beispielsweise auch eine Rapid Development Platform, die im Prinzip eigentlich auch eine lokale Plattform ist wie Mendix für Out Systems. Und da hat es sicherlich auch je nachdem, mit welcher Plattform man arbeitet, unterscheidet sich wahrscheinlich auch der Einsatz von Citizen Development. Nochmals ich denke Power Platform ist ein sehr gutes Beispiel für auch Citizen Development, das ich auch schon erfolgreich in der Praxis gesehen habe, wo nicht IT Personal schnell Lösungen bauen kann. Wenn man dann mehr in die Thematik reingeht, auch da ist es ein Versprechen der Plattformen. Diese Lösungen ermöglichen dir aber einiges komplexere Applikationen zu bauen, als mit Power Automate oder Power Platform selbst normalerweise gebaut wird. Ich sage jetzt nicht, dass es nicht möglich ist, aber normal ist wahrscheinlich gebaut wird. Und da ist natürlich dann auch irgendwo der limitierende Faktor. Wie viel Wissen willst du überhaupt aufbauen in den Fachabteilungen? Weil irgendwann schickst du alle in eine Informatik Ausbildung. Da können sie dann diese Plattformen zum vollen extend benutzen und ich denke, das kann ein spannender Aspekt einfach noch grundsätzlich sein. Wo ist das ein Verkaufsgesprächen der Plattformen und wo ist es dann auch tatsächlich eigentlich, wo das Fachpersonal sich auch wohlfühlt?

Speaker1: [00:25:26] Danke. Dann denke ich, wären wir an dieser Stelle bereits am Ende. Ich würde jetzt hier die die Aufzeichnung abschließen.