**Interview mit Niels Anhalt, Director bei nexum AG**

[00:00:21] **Leo** Danke nochmal, dass du dich bereiterklärt hast für das Interview. Das hilft mir auf jeden Fall sehr weiter.

[00:00:28] **Niels** Wo stehst du denn jetzt gerade in deiner Arbeit? Also welchen Status hat das jetzt? Du hast jetzt durchgeführt und...

[00:00:39] **Leo** Genau, korrekt. Das haben wir Ende Juli gemacht, da haben wir einen Design Sprint gemacht und einen Component Sprint für ein ausgedachtes Produkt. Das gibt es nicht wirklich, aber wir wollten das einfach mal als Grundlage nehmen. Jetzt bin ich gerade dabei, ein bisschen zu schauen: Was hat jetzt wirklich gut geklappt, wo könnte man das vielleicht auch noch ausbessern? Und genau, die Interviews sind Teil dessen, weil ich es einfach sehr wichtig finde, da nochmal viele verschiedene Perspektiven drauf zu kriegen.

[00:01:18] **Niels** Wann bist du dann fertig oder musst fertig sein?

[00:01:22] **Leo** 2. Oktober ist Abgabe. Ich habe noch ein bisschen Zeit auf jeden Fall, aber persönliche Deadline war eigentlich der 1. September, das wird nur ein bisschen schwierig. Deshalb habe ich mir die auch sehr früh gesetzt, weil dann ist da noch ein bisschen Puffer. Erstmal ein kleiner Disclaimer am Anfang. Das Interview ist semistrukturiert. Das heißt, ich stell dir auf jeden Fall ein paar Fragen, aber es wird auch einfach viel eher wie ein Gespräch ablaufen. Wenn dir irgendwie ein Gedanke in den Kopf kommt, kannst du den einfach einwerfen. Da wollte ich dann schauen, dass das so offen wie möglich ist. Ich werde das Interview aufnehmen, aber nicht eins zu eins veröffentlichen, sondern das ist nur für die Zwecke, dass es später dann transkribiert werden kann. Das muss dann der Bachelorarbeit beigefügt werden. Und wahrscheinlich wird es so 45 Minuten dauern. Perfekt. Dann können wir direkt mit den ersten Fragen anfangen und zuerst mal ein paar Fragen zu dir als Person, einfach um das noch mal festzuhalten. Vielleicht magst du einmal kurz beschreiben, wo du arbeitest und was deine Firma macht oder was ihr so anbietet.

[00:02:44] **Niels** Ich bin Niels Anhalt, ich bin einer der Direktoren bei der nexum AG. Nexum ist eine Beratung und Agentur im digitalen Bereich. Wir schaffen große digitale Lösungen und beraten von der Strategie bis in die Umsetzung rein und betreiben dann auch solche Plattformen aus dem digitalen Marketing Bereich und im Bereich der digitalen Produktentwicklung. Meine Rolle ist da ja mittlerweile sehr stark strategische Beratung von Kunden, aber auch Teams managen und führen. Und zu meinem Hintergrund, ich habe ja mal Foto-Ingenieurswesen studiert und hab früher auch mal programmiert, mal vor langer Zeit, und hatte auch lange Zeit ein Team mit ungefähr 20 Entwicklern, die bei mir waren.

[00:03:37] **Leo** Perfekt. Du hattest auch schon mit Design Sprints gearbeitet, ist das richtig?

[00:03:41] **Niels** Ja klar, genau.

[00:03:45] **Leo** Gibt es bei Design Sprints Dinge, die dir besonders gut gefallen oder Dinge wo du sagst, dass es da vielleicht Schwächen gibt?

[00:03:57] **Niels** Besonders gut finde ich, dass es so kompakt ist und wirklich alle Phasen durchläuft. Eigentlich ist es ja fast schon Verkaufstool, um mal vernünftig mit konzentrierter Arbeit zu leisten, Prototypen auch wirklich zu machen in kürzester Zeit und dann auch einen Test darauf zu fahren. Das hat mich ja eigentlich so überzeugt von der Methodik, dass man das dann auch so in Kürze schaffen kann. Wo wir jetzt gerade dabei sind und was ich auch kritisch sehe ist, ob diese Problem-Findungsphase am Anfang so richtig ist bzw. ob man die nicht eigentlich vorlagern muss. Das kommt ja aus der Startup-Welt, wo man sagt, na gut, ich habe den Gründer direkt an Bord und das gesamte Team. Und dann machen wir das jetzt einfach. Wenn du das mit Kunden machst, die auch anders strukturiert sind, muss man eigentlich oft auch die Phase vorverlagern, weil sonst kann man den Design Sprint gar nicht richtig aufsetzen. Das ist meine ich auch bei Design Sprint 3.0, gibt's ja jetzt auch, die haben auch dieses Thema angegangen oder kritisiert oder de facto haben wir das auch, diese Problemklärungsphasen. Haben wir eigentlich vorgelagert, weil sonst kriegst du oft gar keinen Auftrag. Wenn du das nicht schon vorher genauer gefasst hast, dann ist es ja eher nur eine Rekapitulation in dem Sprint und dadurch kannst du die Phase schon etwas weiter verkürzen. Und auch die Experten Interviews, da ist auch immer fraglich, ob man die immer wirklich an dem Tag zusammenkriegt, die Leute. Das ist auch etwas, das man eventuell auch etwas öfters mal auslagern muss, ob man will oder nicht. Ist zwar schön, dass es so kompakt ist, aber wenn es nicht geht, dann geht es nicht.

[00:05:34] **Leo** Ihr macht dann sozusagen vor dem Design Sprint noch mehr Vorbereitung als eigentlich in dem Konzept angegeben. Okay, dann ein bisschen zur Software Seite. Wenn ihr Produkte entwickelt, also jetzt E-Commerce oder digital Produkte oder Webseiten, habt ihr da bestimmte Frameworks, mit denen ihr arbeitet, was die Gestaltung angeht, oder die Entwicklung?

[00:06:04] **Niels** Ja klar, kommt auch drauf an, welches Level. Also wir entwickeln, gerade wenn wir Frontends entwickeln, meistens mit React. Angular gab's auch ein paar Projekte, aber momentan ist die Nachfrage am Markt nach React sehr groß. Weil wir allerdings auch oft keine Web-Applikationen, die standardisierte UI-Elemente haben, sondern eigentlich voll individualisierte E-Commerce Shops haben, kann man relativ wenig von Angular weiterverwenden und das nutzt einfach nicht viel, und deswegen ist React... Ist ja auch kein Framework, sondern eine Library... Benutzt man da die Library und kann da viel mit machen und kombiniert mit NextJS und naja den ganzen anderen Tools, die darum herum sind, um effizienter zu arbeiten. Ich kenne sie jetzt alle gar nicht. Das ändert sich ja auch monatlich so ungefähr.

[00:07:02] **Leo** Da muss man echt immer am Ball bleiben, das stimmt.

[00:07:03] **Niels** Das wäre jetzt der Tech Stack, und dann natürlich Git und Continuous Deployment Flow. Das gesamte Setup, was man da hat, wenn man etwas moderner arbeitet. So, das wäre die Technik. Ansonsten arbeiten... Unsere Designer arbeiten mittlerweile fast nur noch mit Sketch und wir arbeiten mit InVision als Tools und das ist auch die Übergabeschnittstelle zu den Entwicklern, also in InVision die Prototypen zu haben und da dann eigentlich auch die ganzen Spezifikationen, die sind da ja abgreifbar in InVision.

[00:07:38] **Leo** Das heißt, ihr macht für die App auf jeden Fall das Design komplett selber und implementiert dann das Design auch komplett selber.

[00:07:49] **Niels** Ja klar, genau. Ich sehe das ja auch immer als integrierten Ansatz. Wir haben oft Leute im Team, die eher den Informationsarchitektur- oder UX-Fokus haben. Also, was will der Nutzer, Anforderungsaufnahme und architektonische Sachen, also eher Informationsarchitektur. Und dann hast du einen eher Schwerpunkt Design, also Visual Designer und Interaction Designer, der das Interface gestaltet und einen Interaktionsentwickler, also kein Frontend-Dev, sondern einen Interaktionsentwickler, der die Mensch-Maschine Schnittstelle entwickelt. Er ist ja nicht nur Coder, der irgendwelche Codezeilen abliefert, sondern er erschafft. Er ist eigentlich der Erschaffer des Interfaces, nicht der Designer und auch nicht der Konzepter, der irgendwelche Sachen ausdenkt. Das ist aber eigentlich ein integriertes Team. Alle drei sind zuständig, dass am Ende HTML Code entsteht, der sinnvoll ist, also nutzbar.

[00:08:49] **Leo** Und nutzerfreundlich wahrscheinlich auch.

[00:08:52] **Niels** Genau.

[00:08:53] **Leo** Perfekt. Dann würde ich dir einmal mal ganz kurz die Grundidee von Component Sprints vorstellen und das Projekt, was wir gemacht haben. Wir hatten ja schon mal drüber gequatscht so ein bisschen, du hattest ja wahrscheinlich auch mit Daniel wahrscheinlich schon ein bisschen drüber gesprochen. Der Component Sprint wäre sozusagen der nächste Schritt nach dem Design Sprint, in meinen Augen. Und das Ziel des Component Sprints ist es, dass du eine sogenannte Pattern Library an Komponenten schon mal entwickelst, die nachher im Produkt dann verwendet werden kann. Das sind halt so Design-Elemente, wie zum Beispiel Buttons oder Eingabefelder oder auch custom Sachen, wenn halt irgendwie bestimmte Design-Elemente wirklich für diese App gebraucht werden. Dann würden die in diesem Component Sprint schon mal entwickelt werden, sodass man direkt danach als Entwickler, der die Geschäftslogik implementiert, sich darum nicht mehr kümmern muss. Die grundsätzliche Idee orientiert sich ein bisschen an Atomic Design, hast du davon schon mal gehört?

[00:10:05] **Niels** Ja klar, genau.

[00:10:06] **Leo** Ich habe jetzt, einfach um ein bisschen simpler zu halten, gesagt, dass wir nur Atome und Moleküle und Templates sozusagen als Basis nehmen, beim Atomic Design. Du hast ja theoretisch auch noch Organismen und so, aber das habe ich jetzt erst mal rausgelassen, um das hierarchisch nicht zu komplex werden zu lassen. Die Komponenten würden dann in Storybook dokumentiert werden. Storybook ist ein Tool, wo du sehr einfach dir Dokumentation mit aufbauen kannst. Habt ihr damit schon mal gearbeitet oder sagt dir das was?

[00:10:42] **Niels** Ja, Kollegen von mir, ja, soweit ich das weiß. Persönlich aber noch nicht, ja.

[00:10:48] **Leo** Von der Struktur her wird der Component Sprint ähnlich sein wie im Design Sprint 2.0. Auch insgesamt vier Tage, jeden Tag von 10 Uhr bis 17 Uhr. Im Vorhinein vom Component Sprint wollen wir sicherstellen, dass Storybook schon mal aufgesetzt ist, dass, wenn es existierende Komponenten gibt oder ein Design Systems, das schon mal irgendwie alles gesammelt wird und dass die CI-Guidelines schon mal feststehen, also Fonts, Farben, Style, die Stimmung. Am ersten Tag ist der Plan, dass der Prototyp, den es schon gibt, also das was dann zum Beispiel in Sktech gestaltet wurde, oder in InVision, unterteilt wird, zerschnitten wird in Moleküle und Atome wo man schauen kann, welche Komponenten gibt's denn da, welche Komponenten kann man wie hierarchisch aufbauen. Hier wird dann auch geschaut, welche Varianten es für diese Komponenten geben muss, also ein Button kann ja zum Beispiel disabled sein oder ein Primary Button sein, so dass das alles schon mal festgehalten wird. Dann werden diese Komponenten priorisiert, sodass man schaut, welche müssen als Erstes implementiert werden, welche sind vielleicht von welchen abhängig. Und last but not least ist es natürlich auch wichtig, dass das gesamte Entwicklerteam da sich auf einen Codestil einigt, sich auf eine Dokumentation einigt, dass da alle dann auf demselben Wissensstand sind. Am ersten Tag selber wird noch nicht entwickelt, sondern es ist wirklich nur Wissensaustausch, um zu schauen, dass wirklich alles schon mal vorbereitet ist. Ich denke, das ist ein wichtiger Teil dafür, dass die nächsten Tage dann wirklich flüssig ablaufen. Tage zwei und drei sind dann wirklich Implementation-Tage. Da wird nur entwickelt. Wir haben das bei uns so gemacht, dass wir jede Komponente als Post-it einfach auf einer Wand hatten. Jeder, der gerade nichts zu tun hat, nimmt sich eine der Komponenten als Post-it, implementiert diese in Storybook und hängt sie dann an den fertigen Bereich, sodass jeder eine gute Übersicht darüber hat was jetzt gerade schon gemacht wurde was noch gemacht wird. Und der vierte Tag wäre dann der Tag wo vielleicht fehlende Dokumentation noch nachgearbeitet wird, wo nochmal eine Retro gemacht wird, was hat gut geklappt, was hat nicht gut geklappt. Und wo, wenn der Kunde das möchte, ein Handoff passiert, dass der Kunde darin eingeweiht wird. Das Ganze… Also wenn du Fragen hast…

[00:13:44] **Niels** Ne, soweit verstanden. Das Ergebnis ist dann eine Pattern Library...

[00:13:48] **Leo** Genau, richtig. Das wäre dann...

[00:13:52] **Niels** Eine dokumentierte Pattern Library.

[00:13:53] **Leo** Genau.

[00:13:55] **Niels** Ich habe mich eben nur gefragt, also kommt drauf an, wer ist der Kunde? Und was fängt er danach damit an?

[00:14:05] **Leo** Jetzt haben wir das... Um das ganze mal zu testen haben wir uns ein eigenes Produkt ausgedacht. Das Produkt heißt Wevent, ist eine kleine Plattform wo du, ich sage jetzt mal, Veranstaltungen im kleinen Kreis planen kannst. Es geht darum, wenn du z.B. deine Geburtstagsparty feierst und diese planen möchtest und deine Freunde einladen möchtest, dass du etwas darüber machen kannst. War jetzt einfach so eine Idee, weil sich das vom Scope relativ gut irgendwie implementieren ließ und... Da haben wir in der ersten Woche einen ganz normalen Design Sprint 2.0 gemacht mit vier Personen in vier Tagen. Ich gebe jetzt mal gerade, wenn es geht, das Fenster frei. Ich hoffe, du kannst mein Bildschirm sehen. Das ist der Prototyp, der am Ende herausgekommen ist. Hier lädt wohl irgendwie das iPhone drum herum nicht, aber das kannst du dir ja vorstellen. Du kannst dich da durchklicken. Wenn du ein Event erstellst, musst du dich einmal kurz Einloggen mit Google. Dann hast du die Möglichkeit, hier Daten festzulegen, kannst einstellen, wo deine Veranstaltung stattfindet. Noch eine kleine Beschreibung anlegen. Kannst dich dann hier durchklicken und dann kannst du ein Template auswählen für ein kleines Design irgendwie. Und dann, im letzten Schritt, halt Gäste einladen. Das sind so die Screens, also das ist ein relativ einfacher Prototyp. Den haben wir auch, so wie man es im Design Sprint ja macht, dann direkt mit 5 Usern getestet, schon ein bisschen Feedback gesammelt und darüber iteriert. Also das ist jetzt die iterierte Version sozusagen.

[00:16:05] **Niels** Wie habt ihr das jetzt gebaut, mit welchem Tool?

[00:16:05] **Leo** Mit Framer. Framer ist ganz cool, weil du da direkt Komponenten die vielleicht ein bisschen interaktiver sein müssen, also wenn ich hier zum Beispiel so etwas haben möchte, das kannst du direkt in React schreiben und die einfacheren Sachen kannst du aber halt genau wie in Figma oder in InVision einfach als Vektor-Komponenten basteln. Genau. Den Component Sprint haben wir zu dritt durchgeführt, also wir waren drei Entwickler. Auch in vier Tagen, so wie ich das gerade eben beschrieben habe. Und das wäre hier das Ergebnis. Das ist jetzt sozusagen ein Storybook, wo du alle Komponenten siehst, die in diesem Prototypen verwendet wurden. Das wären dann zum Beispiel Buttons in ihren verschiedenen Zuständen. Das wäre dieser Zeit-Picker zum Beispiel, oder Eingabefelder. Da hast du halt diese ganzen Komponenten drin. Darauf aufbauend hast du Moleküle, die dann zum Beispiel mehrere Komponenten oder mehrere Atome enthalten und als höchste Kategorie hast du die Templates. Da geh ich mal kurz in die iPhone-Ansicht. Das sind dann sozusagen die fertigen Layouts wo die Komponenten alle schon drin sind, sodass man als Entwickler wirklich sieht, wie wird es benutzt. Du siehst auch als Entwickler hier direkt... Ich hoffe, das lädt... Wie das alles zusammengebaut ist. Du könntest sozusagen hingehen, dir den Code raus kopieren, in die App reinschmeißen und dann hast du das Layout so wie es ist, mit den ganzen Komponenten drin. Die Dokumentation besteht bei uns oder ist bei uns komplett auf der Codeseite. Dafür geh ich mal kurz in das Repo rein. Wir haben alles so dokumentiert, dass man diese Komponenten im Code Editor benutzt, dass man direkt vor Ort die Dokumentation bekommt. Dieses Storybook, so wie es jetzt ist, könnte man jetzt an das Entwickler-Team weitergeben, sodass die damit arbeiten können und dann sozusagen ihre App damit bauen können. Hast du zu dem Konzept so generelle Fragen oder irgendwas, irgendwelche Punkte, irgendwas, was dir vielleicht aufgefallen ist?

[00:19:17] **Niels** Das Team, das den Component Sprint durchführt, sind nur Entwickler. Oder ist dann noch jemand anderes dabei?

[00:19:25] **Leo** Wir hatten es jetzt vorerst nur mit Entwicklern gemacht, aber ich habe auch schon Potential gesehen, dass man vielleicht nicht nur Entwickler mit dabei hat, sondern vielleicht auch UX-Designer oder so.

[00:19:43] **Niels** Verstehe. Und hat das... Also gut, drei Entwickler, so gesehen zwei Tage, das sind sechs Tage reine Entwicklungszeit. Hat das soweit geklappt? Also habt ihr da eine gewisse Vollständigkeit erreicht für diese App?

[00:19:58] **Leo** Ja, total. Wir waren sogar noch viel früher fertig als wir eigentlich gedacht hatten. Wir hatten am vierten Tag eigentlich fast nichts mehr zu tun. Das war jetzt... Das muss man natürlich auch mal schauen. Die App war jetzt vom Scope her noch relativ kompakt. Wir hatten insgesamt 29 Komponenten, die implementiert wurden. Auf drei Leute verteilt ist das auf jeden Fall überschaubar. Wir haben aber auch gemerkt, dass sich dieses Entwicklerteam gut skalieren lässt, weil es implementiert ja jeder einzelne Komponenten, die Komponenten sind meistens unabhängig voneinander. Das heißt, wenn man jetzt ein größeres Projekt hätte, könnte man das Entwicklerteam auch einfach erweitern auf, sagen wir, fünf Entwickler zum Beispiel. Und ich denke, dass es sich relativ gut parallelisieren lässt, ohne dass es irgendwie zu Problemen käme.

[00:20:50] **Niels** Das habe ich mittlerweile auch gemerkt. Wenn du willst, kann ich einfach jetzt was zu sagen. Vielleicht mal vorne angefangen oder eigentlich davor. Was wir merken ist, dass du... Nach einem Design Sprint hast du zwar einen Prototyp, den hast du auch validiert, ja. Aber es kann auch sein, dass da etwas nicht so ganz funktioniert hat. Das heißt, du brauchst eigentlich aus meiner Sicht auf jeden Fall noch eine zweite, entweder, ich meine, AJ&Smart macht das ja auch im verkürzten zweiten Design Sprint.

[00:21:28] **Leo** So einen Iteration Sprint irgendwie.

[00:21:28] **Niels** Genau. So was brauchst du, damit du überhaupt die Qualität halt dann sicherstellen kannst, außer du hast perfekt gearbeitet.

[00:21:42] **Leo** Das ist selten, ja.

[00:21:42] **Niels** Also, wenn du jetzt über so einen Kontext redest in deiner Arbeit, dann müsste man ja eigentlich sagen, naja es ist nicht direkt nach dem Design Sprint, sondern dazwischen gibt es eine Phase, die kann man ja auch sehr kurz mal machen, die kann man aber auch... Aber das wäre jetzt auch nur eine Iteration auf dem Prototypen. Was wir allerdings eigentlich immer brauchen, sobald wir mit Kunden unterwegs sind, ist irgendwo eine Vision, ein fertiges Stück, wo auch der Kunde einen Haken dran macht. Also Stichwort Design. Passt das in das Corporate Design des Kunden, ist das ein neues Design? Wie weit weicht das ab? Wer muss da alles überhaupt was gesagt zu haben, damit das wahr ist. Das schöne ist ja, beim Design Sprint kannst du immer sagen, naja, ist halt ein Prototyp, da muss nicht alles so stimmen und das ist irgendwie fake. Es geht ja gar nicht darum, dass es ein finales Produkt ist. Jetzt wenn du direkt einen Entwicklungsschritt dranhängt, dann wird es schon eher so, naja, aber das meinten wir jetzt schon ernst. Weil wir bauen, auch wenn es nur acht bis zehn Personentage sind, da sind umgerechnet auch schnell 10000 Euro weg, oder 15000, je nach Overhead. Da willst du ja schon in die richtige Richtung dich bewegen. Also sprich was ich meine, ich glaube du brauchst nicht nur einen Iteration Sprint, um die Ergebnisse des Design Sprints zu verarbeiten, sondern eigentlich auch, wir hatten das jetzt mal als alles zusammen als Strategie-Sprint formuliert. Du brauchst auch einen Visionsprototypen, so haben wir ihn genannt, der auf Basis von dem Design Sprint Prototypen sein kann, aber der dann eben auch gemacht wird. Der vielleicht noch ein paar weitere Sachen hat, die man nicht im Design Sprint abtesten konnte, und insbesondere aber auf eine Fidelity gekommen ist und auf eine Ernsthaftigkeit des Designs. Da könnte man jetzt auch sagen, ja das möchte ich so. Und wenn es nur das Orange ist was ihr da benutzt habt, und die runden Ecken. Da kannst du ja mit Kunden beliebig lange noch darüber diskutieren, wenn man es auch ernst meint. Also reden können ja viele, aber wenn du wirklich sagst, das ist jetzt die Richtung, klar kann man immer später noch etwas ändern. Aber wenn ihr da 30 Komponenten baut dann willst du nicht nochmal das komplett in Frage stellen.

[00:23:57] **Leo** Ja, richtig.

[00:23:59] **Niels** Also ich würde schon sehen, dass du dazwischen... Vielleicht ist das auch nur eine Woche, da kann man einen Iteration Sprint plus einen... Wie würde man das dann in dieser Terminologie nennen? Das wäre dann im Prinzip eine Härtung dessen, was man da macht. Wir nennen es natürlich auch manchmal Design Directions oder Design... Vielleicht ist es auch ein Style-Guide- Ansatz. Damit auch zum nächsten Punkt. Das ist nicht ganz konkret, wie wir das Ding nennen, aber es hat ja mehrere Nutzen. Einerseits Design Sprints iterieren, dann eine Klarheit darüber, dass das wirklich auch das Ziel ist, eine Abnahme zu bekommen an der Stelle, oder Bestätigung zumindest. Und auch, weil du sagst, dass die Phase davor, da sollten ja wenigstens irgendwie CD-Vorgaben sein. Ja, da musst du wenigstens die Farben und Formen halbwegs klar haben, ich glaube aber, du musst auch noch ein paar mehr Sachen klar haben, also wirklich geklärt haben, damit das wirklich vom Design her dann sitzt. Da habt ihr euch jetzt nicht so drauf konzentriert, ihr wart ja auch euer eigener Kunde.

[00:25:20] **Leo** Ja, richtig.

[00:25:21] **Niels** Wenn auch der CEO plötzlich noch mit über die Farbe reden will, und das macht der aber dann meistens erst nach dem Component Sprint, weil ihr so schnell durch seid, dass die das gar nicht gemerkt haben. Das kommt nämlich erst in drei Wochen, dann hast du irgendwann ein Problem. Das merke ich immer, sobald du individueller wirst im Interface und auch viel näher Markenthemen berührst, dann hast du da einfach nochmal einen anderen Abstimmungsaufwand und Kreativaufwand. Unser Kreativdirektor macht sich auch gern mal einfach mal drei, vier Tage Gedanken darüber, was jetzt das gesamte Bild ist und nicht nur das, was im Design Sprint mal eben getestet wurde. Vielleicht kannst du das dann auch Brand Sprint nennen, den müsste man dazwischen schieben. Weil wenn du das nicht machst, dann baust du irgendwas, was gar nicht on-brand ist, das nichts mit der Marke zu tun hat. Wie gesagt, im Startup-Modus ist das vielleicht auch egal, oder der Gründer kann dann einfach sagen, Wevent geil, Orange finde ich auch gut, haut rein. Hauptsache das MVP ist draußen. Sobald du aber... Da brauchst du dann noch mehr.

[00:26:37] **Leo** Das ist gut, von dir da nochmal die Perspektive aus dem größeren Unternehmen zu kriegen. Das sind ja dann doch nochmal extreme Unterschiede.

[00:26:44] **Niels** Ja, vielleicht kann man das auch Architektur Sprint oder sowas nennen. Weil du brauchst eine Markenarchitektur, du brauchst eigentlich auch eine Informationsarchitektur. Oft ist das, was in den Design Sprints rauskommt, das bildet ja ein oder zwei Use-Cases ab, oder eine Journey. Das hat aber oft nicht den Anspruch darauf, dass es vollständig ist. Ich mach mal ein Beispiel, Hauptnavigation, Struktur der Applikation. Im Burgermenu kannst du alles Mögliche verstecken. Macht aber vielleicht nicht die beste Usability dann aus. Momentan geht der Trend ja wieder unten zu den Leisten da, vier Icons. Die waren mal irgendwann weg, jetzt kommen sie wieder. Weiß nicht, ob ihr das verfolgt habt. Da gab es auch immer Artikel auf Medium und so, und dann haben alle ihre Apps wieder umgebaut. Also wirklich alle großen aus dem Silicon Valley, und alle in Deutschland guckten und sagen, oh scheiße, jetzt muss ich die App wieder umbauen. Die sind ja doch wieder in, diese blöden Icons da unten, oder ein großes in der Mitte oder wie auch immer. Aber da musst du dich dann auf einmal entscheiden, was kommen denn da für Icons rein. Also ich meine jetzt nicht grafisch, sondern was sind denn die Haupt-Use-Cases? Hast du das schon im Design Sprint herausgefunden? Sind das dann nur drei oder vier, reicht uns dieses Pattern aus? Oder muss man ein ganz anderes Pattern benutzen, was auch sechs wichtige Sachen abbildet? Oder vielleicht haben wir auch nur eins, wäre ja geil, dann könnten wir diese Leiste ganz weglassen, wie cool. Wann entscheidest du das? Das kannst du oft... In dem Design Sprint kriegst du oft eine Ahnung davon. Im Component Sprint ist es schon zu spät, weil ihr sagt, ja, ich bau euch die Leiste. Wenn ihr sagt, die Leiste ist da, dann bau ich die. Wenn nicht, dann nicht. Also wer entscheidet das? Also spricht dazwischen brauchst du eine Phase, eine Architekturphase, die zumindest mal, da kannst du auch nie 100 Prozent sicher sein, aber du versuchst ja, den Gesamtscope des Projekts zu verstehen. Deswegen nennen wir das auch meistens Scoping-Phase, die einfach den Umfang definiert. Das hat jetzt nicht nur was mit Auftragsvolumen zu tun, sondern auch, damit ein Informationsarchitekt mal halbwegs versteht, wie viele Use-Cases haben wir denn da, wie viele unterschiedliche Zielgruppen. Also den ganzen UX-Kram, den man normalerweise vor UI abfragt. Das müsste wahrscheinlich parallel zum Design Sprint und auch nach dem Design Sprint passieren, also so machen wir das dann, um dann den Component Sprint erst starten zu können. Das ist dem Component Sprint im Prinzip in sich geschlossen ja eigentlich egal, weil der Input ist halt irgendein Prototyp oder irgendein Input, der sagt, wir haben diese Dinger. Am ersten Tag, also an einem Tag, kannst du darauf aufsetzen und dann in die Tiefe gehen, aber wenn du das nicht gemacht hast, schaffst du es nicht glaub ich nicht, eine solide Architektur, Informationsarchitektur zu schaffen, das muss davor passieren. Das ist ja jetzt für dich im Prinzip als Component Sprint auch nicht schlimm. Ich glaube nur, du musst die Vorbedingungen, die da sind, zumindest dann mal dokumentieren.

[00:30:19] **Leo** Das ist ein guter Punkt.

[00:30:23] **Niels** Das trifft ähnliches wie bei Scrum. Scrum ist ein agiles Framework zur Softwareentwicklung. Das lässt nur eben die gesamte konzeptionelle Arbeit, von der ich gerade sprach... Die bildet Scrum nicht wirklich ab, sondern hat das alles in die Rolle des Product Owners ausgelagert. Der hat einen Stakeholder und kümmert sich darum, dass diese architektonischen Entscheidungen entweder schon getroffen sind oder getroffen werden. Dann kommt das Team und sagt, wir wollen jetzt hier so eine Leiste machen, dann sagt der PO, kein Problem, machen wir so. Also das ist das Verlangen von Scrum, dass er so allwissend ist, dass er das alles machen kann oder mal schnell seine Stakeholder fragt, innerhalb von wenigen Minuten, um das dann zu entscheiden zu können. Ich sage immer, ein PO, der das kann, der muss schon teilweise gottgleich sein. Man muss auch telepathische Fähigkeiten zu ungefähr 20 bis 40 Stakeholdern haben und das innerhalb Sekunden dann abklären können. Das ist ja... Weißt du, auch diese ganzen agilen Softwareentwicklungsframeworks... Also gerade Scrum lagert halt viel von diesen Themen einfach auf irgendwelche Rollen aus, also raus aus sich. Und kümmert sich aber nicht darum und die sind ja trotzdem da. Da steht da das Dev-Team im Scrum Sprint und sagt, hey laufender Motor, wir können entwickeln, wir können machen. Und die Architekten sagen, ich habe noch nicht entschieden ob die Leiste da unten vier oder acht oder gar nicht passt, weil ich weiß es ehrlich gesagt noch nicht, weil ich habe noch keine Ahnung ob wir jetzt gerade ein Mehrfamilienhaus mit zehn oder mit 50 Parteien bauen. Oder vielleicht müssen wir sogar zwei Blöcke bauen, nämlich zwei Apps. Weil es geht nicht anders bei dem Use-Case. Ich habe hier noch Terra Incognita. Ich habe mit acht Fachabteilungen noch gar nicht gesprochen ehrlich gesagt, und die haben noch nicht ihre Anforderungen. Kann sein, dass sie alle sagen, passt so. Kann auch sein, dass die sagen, oh, wie brauchen nochmal ein ganz neues Gebäude. Nur, damit du die Komplexität da... Das würde ich sehen... Das müsste man als Vorbedingung ja eigentlich haben. Das wird jetzt bei Scrum immer auch als Null bezeichnet. Da wird halt auch alles rein gepackt, ist manchmal aber auch zu klein. Manchmal... Also wir lagern eigentlich noch vor den Sprint Null eigentlich noch so einen Vor-Strategie-Sprint, in dessen Rahmen dann Scoping und Architektur und Designarchitektur-Arbeit stattfindet und Design Sprint stattfindet. Das ist ja im Prinzip dein Kontext, in dem der Component Sprint stattfinden müsste.

[00:33:04] **Leo** Genau. jetzt hatte eine der anderen Personen, mit denen ich darüber mal gequatscht hatte, vorgeschlagen: Wie wäre es denn wenn man den Component Sprint nicht unbedingt direkt nach dem Design Sprint macht, sondern ihn nach der Entwicklung der ersten Version der App wirklich legt, sodass man sozusagen die App schon hat, zwar mit Komponenten die so noch nicht wiederverwendbar sind, die jetzt einfach erst mal so implementiert wurden, wie man das normalerweise auch macht. Wo dann die Idee ist, also falls man dann mit dieser App noch wachsen möchte, dass man dann erst einmal eine Pattern Library aufbaut und die dann benutzt um das zu refactoren.

[00:33:55] **Niels** Also würde ich nur machen, wenn man ein Startup ist, was relativ schnell mit einem MVP auch an den Markt gehen muss, also mit einer App, die wirklich dann an Endkunden ausgeliefert wird. Klar, dann kannst du dreckig sein. Haben wir meistens nicht den Kontext, weil wir würden... Klar, Prototyping würde ich nicht mit Code machen, solange wie ich kann. Also gerne auch mit Teilcodes, wie ihr das jetzt bei Framer gemacht habt, weil das wäre eigentlich wahrscheinlich Ressourcenverschwendung, aus meiner Sicht. Weil eben, auf der einen Seite, ihr habt ja jetzt auch gezeigt, in einer Woche oder in vier Tagen könnt ihr auch massiv gut schon Komponenten bauen. Also ich meine warum soll ich... Ich meine, ich müsste ja genauso viel Zeit reinstecken, um diese App zu bauen. Und warum soll ich zehn, 15 Personentage in etwas reinstecken, was jetzt höchstwahrscheinlich Wegwerfcode ist, wenn ich eigentlich in der gleichen Zeit Komponenten haben kann, die ich wiederverwendbar sind. Also, kann mal Sinn machen, weiß ich jetzt nicht, aber würde ich jetzt nicht als Grundansatz eines Frameworks wählen, weil wir sagen dann auch immer, lass es uns dann in dem Moment wo wir genauer wissen was wir tun wollen... Dann lasst es uns auch richtig machen. Richtig machen heißt nicht, dass wir uns vier Monate einschließen, damit das dann auch richtig ist. Aber ihr zeigt ja eigentlich, das ist ja auch eigentlich dein Kernansatz, ja, in einer Woche kann man eigentlich eine Menge schaffen, wenn man sich konzentriert. Und dann ist das auch noch wiederverwendbar, wie geil ist das denn. Würde ja eigentlich dein Modell konterkarieren oder weiß ich nicht. Dann könnte man den Vorteil gar nicht bringen. Also ich würde sehen... Was ich nur sehe ist, dass einfach viele Sachen schon ein bisschen gehärteter sein müssen, also Klarheit schaffen. Nie 100 Prozent, weil sonst ist es zu Wasserfall-artig, aber zu 80 Prozent muss ich gewisse Sachen schon herausgefunden haben. Sei es, wie groß ist wirklich der Umfang grob, nicht mal im Detail? Wie ist die Brandarchitektur und das Brand-Design? Was ist die Vision? Muss ich alles schon mal irgendwo zu 80 Prozent stehen haben. Vision vielleicht zu 100 Prozent. Dann kann ich auch in so einen Component Sprint reingehen. Was ich überlege ist, wenn wir jetzt mal die Rahmenbedingungen haben. Hast du dich schon mal mit Scrum tiefer auseinandergesetzt?

[00:36:27] **Leo** Noch nicht super tief. Ich kenne so ein bisschen die Grundlagen davon, aber ich glaube wir hatten mal einen Entwicklungssprint nach Scrum, aber mit einem externen Unternehmen. Aber das war auch nur eine Woche lang.

[00:36:40] **Niels** Ja. Was ich halt überlege ist, im Prinzip folgst du ja... Eigentlich folgst du mit dem Schema so einem Scrum-Ansatz, so einem Scrum Sprint. Die sagen, lass es uns lieber zwei Wochen machen, oder zwei bis vier Wochen. Und da sind noch ein paar mehr Schritte drin. Da ist ein Planungsschritt, zwei Planungsschritte drin. Da ist aber auch ein Review, eine Retro schon eingebaut, also das Handoff was du hast ist dann der Review. Da gibt's ja schon die Terminologien und ja, tatsächlich, in der Mitte, also rein von der Grundarchitektur, nur meist zwei Wochen lang, macht man das eigentlich bei Scrum auch genauso. Du hast eine Planungsphase mit zwei Schritten, wo viel Vorgaben schon erwartet werden. Dann hast du eine Dev-Phase mit Daily Standups und Pipapo. Dann hast du eine Review- und Retro-Phase also einerseits Review, Abgabe beim Auftraggeber, und Retro, um zu lernen. Und das würde eigentlich dem so entsprechen vom Schema her. Weil das überlege ich gerade. Das passt natürlich ganz gut eigentlich in ein gesamtes Scrum-Framework rein.

[00:38:03] **Leo** Viele Ähnlichkeiten auf jeden Fall.

[00:38:04] **Niels** Da sagt man jetzt... Also wir sagen dann immer, nach diesem, was ich dir gerade alles erzählt habe. Architektur- oder Scope Sprints. Und dann nochmal so ein Sprint Null, der so für technisches Setup und Backlog-Befüllung dient. Und dann gehen wir, sagen wir einfach, ja klar und jetzt gehen wir in die agilen Sprints rein. Da kann es sein, dass in einem agilen Sprint in zwei Wochen oder drei Wochen, eine Komponenten-Library entsteht. Es kann aber auch sein, dass das jetzt, das gibt jetzt Scrum nicht mehr vor oder da gibt's keine Regel was da drin passiert, sondern das entscheidet im Prinzip bei der Planung der PO und das Dev-Team, was am sinnvollsten da gemacht werden muss. Es kann sein, dass man sagt komm, lass uns möglichst schnell eine Komponenten-Library relativ vollständig und auf 70-80 Prozent bringen. Kann auch sein, dass man sagt, das ist das was wir manchmal jetzt mehr machen entlang von User Story Maps, lass uns lieber einen Durchstich machen. Also ich bau von den 30 Komponenten nur die zehn, die ich brauche, um wieder einen Ablauf zu haben. Wenigstens ich kann ein Use-Case oder User-Story abbilden. Also sprich, ich kann ein Event adden. Dann gucke ich mir nur genau die Komponenten an, die ich da brauche, und baue sie aber so, dass sie vielleicht mit dem Backend schon theoretisch funktionieren. Da gibt's aber nur diesen einen Use-Case. Da kann man auf den Screen gehen, kann da was reintippen, drückt auf add und dann erscheint das Event da. Ist aber dann auch schon in der Datenbank drin. Wäre potentially shipbar. Man könnte theoretisch das ausliefern und sagen, wenn du nur ein Event adden willst, dann kannst du das hier. Das ist voll funktionabel, durchgetestet. Hier ist die URL, ist auch echter Code, nicht mehr geprototyped, läuft alles. Das ist ja der Ansatz des Durchstichs, dass du wirklich komplett von Design über Frontend-Entwicklung und Backend-Entwicklung, Deployment, die komplette Nummer hast. Von daher könnte es sein, dass ein Component Sprint so eine besondere Art eines Scrum Sprints ist, oder einer verkürzten... Vielleicht legt man das auch sogar davor. Ich weiß es auch gerade nicht, aber es könnte sein... Also wenn du willst, dass das so in einen skalierfähigeren Gesamtentwicklungs-, im Prinzip, Ablauf reinpasst dann müsstest du überlegen, da brauchen wir einen Design Sprint, da brauchen wir einen Component Sprint. Und was passiert danach? Das meine ich mit dem Handoff, und dann ein Kunde... Also unsere Kunden sagen dann auch, vielen Dank für die Komponenten, und wer entwickelt denn jetzt da weiter? Also finde ich super, aber danach kommt ja wieder was. Vielleicht seid ihr das, können auch andere Dienstleister sein, also alles gut. Danach geht es ja weiter und dann ist ja wieder die Frage der Anschlussfähigkeit nach draußen. Warum nicht das schon so machen, dass es sich in Gesamtkontext ein... Man könnte auch sagen wir machen einen Component Sprint und das ist ein ganz guter Handoff in eine agile Weiterentwicklung. Manchmal auch ganz gut, weil oft sind auch Scrum-Entwicklungsprojekte doch sehr technisch, vielleicht auch manchmal Backend-lastig, dass man sagt, komm wir lagern das noch ein bisschen aus, das sind manchmal auch andere Dienstleister, die dann zum Zuge kommen, also eher so krasse IT-Dienstleister. Die machen da ihre agilen Sprints, denen ist es aber auch vollkommen egal, wie diese Komponenten aussehen und ob sie überhaupt aussehen. Da könnte man sagen, wir haben im Prinzip eine Vorphase, die aus Design Sprint, Scoping und Komponenten entsteht. Dadurch haben wir schon sehr viel vom UI abgedeckt und können gut ein Handoff in einen agilen Entwicklungsprozess hinein... Weil dann können die direkt damit arbeiten. Also eine Auslagerung der Komponenten. Wenn man sowieso integriert arbeitet, würde ich sagen, ja, kann man machen. Wüsste ich aber... Also ich würde das in dem Moment eigentlich das Team entscheiden lassen, ob es sinnvoll ist, 30 Komponenten am Stück zu entwickeln oder zehn und dafür noch was anderes zu machen, in dem ersten Sprint. Das sagt im Prinzip eigentlich eher der Product Owner. Der sagt, guck mal, das und das wäre mir jetzt wichtig. In zwei Wochen möchte ich das und das sehen.

[00:42:35] **Leo** Okay, also du würdest den Component Sprint nicht unbedingt direkt für das gesamte Produkt machen, sondern vielleicht erst mal für einen fokussierten Bereich?

[00:42:41] **Niels** Ja, dann aber auch eigentlich integriert in den Ablauf. Oder du sagst, wir lagern den vor, in einen agilen Ablauf. Agil, da kann man jetzt konkreter werden. Vor ein Scrum Projekt, weil agil ist ja alles Mögliche. Aber Scrum hat sich ja durchgesetzt mit den Sprints. Vor die Scrum Sprints, um schon mal ein Paketchen zu haben, das aber den agilen Prinzipien folgt. Das ist ja vielleicht wichtig. Du folgst ja insgesamt mit Design Sprints und einem Component Sprint eigentlich den agilen Prinzipien. Vielleicht ist das auch nochmal hilfreich für deine Arbeit, ich weiß nicht, ob du das soweit erwähnt hast.

[00:43:25] **Leo** Bisher noch nicht, tatsächlich.

[00:43:28] **Niels** Weil das ist ja eigentlich so der mega Hype, Trend, was auch immer, oder jetzt momentan sagen alle, agile is dead. Aber eigentlich ist da ja eine hohe Qualität drin. Das ist ja genau die Qualität, die du ja auch sagst. Sehr enge, kleine Teams, die gut miteinander etwas machen können, mit schnellen sichtbaren Ergebnissen, also Kollaboration über Dokumentation und so weiter, da gibt's ja die agilen Prinzipien. Die kannst du dir nochmal anschauen, agiles Manifesto. Da kannst du auch darauf referieren, also welche der agilen Prinzipien nutzt du denn hier, unterstützt du hier. Das ist mir auch bei dem Design Sprint aufgefallen, dass die mal so 0,0 über Agile reden. Also fand ich ganz interessant oder spannend. Die reden auch nicht über Design Thinking primär, die reden auch nicht über UX, obwohl sie eigentlich alles machen. Design Thinking Methoden, die nutzen UX-Ansätze, also endlich mal auch User Research und Testing und mit dieser Sprint Logik und diesen Zusammenarbeit-Kollaborationen bist du eigentlich mitten in der agilen nutzen eigentlich prinzipiell.

[00:44:39] **Leo** Das stimmt, das ist echt interessant. Ich habe es auch noch nicht gehört, also in dem Kontext auf jeden Fall noch nicht.

[00:44:43] **Niels** Das ist übrigens, aber das ist eher ein Nebensatz, vielleicht auch schon fast das Problem. Weil ich hatte auch das Gefühl, jetzt hier bei unserem Meetup. Da waren auch ein paar Leute... Der eine meinte auch, naja also mit Sprints kenne ich mich aus, also ich weiß, ich kann mir schon vorstellen, was bei einem Design Sprint passiert. Damit meinte er aber, er kennt Scrum Sprints und den Ablauf da, und dachte jetzt, naja da ist im Prinzip ein Scrum Sprint, nur für Design, spannend. Dadurch ist das aber für viele Leute, die aus dieser agilen oder aus der Scrum-Welt kommen, relativ schnell eingeframed in einen Sprint, so wie Scrum. Es ist auch gar nicht so unähnlich. Aber eigentlich ist der Design Sprint zuerst mal was komplett anderes, es benutzt nur die Prinzipien. Das nur so nebenbei. Wenn du jetzt... Wenn du da so, wie soll ich sagen, man macht ja oft in solchen Arbeiten auch mal so Andockpunkte, also die alle Welt verändert sich, agile, agile, Trend. Agile, was sind die Prinzipien. Ah, guck mal, habe ich auch in meinem... Ah, super. Eigentlich nutzt das die agilen Prinzipien, nutzt das, was Design Sprint macht. Nutzt die agilen Prinzipien oder unterstützt sie, und ist damit voll anschlussfähig in eine agile oder Scrum-Welt.

[00:45:55] **Leo** Das ist ein sehr guter Gedankenanstoß, da werde ich mich nochmal mit befassen.

[00:46:00] **Niels** Ihr habt das glaube ich auch sowieso schon intus, also eingebaut, aber noch nicht reflektiert. Muss man manchmal auch nicht, wenn man es intuitiv richtig macht, aber so...

[00:46:25] **Leo** Richtig, Bachelor ist ja nicht intuitiv.

[00:46:25] **Niels** Vielleicht noch einen weiteren Punkt: Ich sehe es eigentlich als wichtig oder interessant an, dass UX-Designer auch da im Team sind. Das wäre zumindest auch nach den agilen Prinzipien, also man würde immer versuchen, ja auch interdisziplinäre Teams zusammenzubringen. Weil sonst, ich sag jetzt mal böse, verkommt das ja zu einem reinen skalierfähigen Code-abarbeit-Sprint. Da können wir sagen, das können wir auch hochskalieren, ist eigentlich auch vollkommen egal, die arbeiten eigentlich auch nicht zusammen, weil die können auch sitzen, wo sie wollen. Wir haben die Komponenten am ersten Tag aufgeteilt. Verteilen wir die in die Welt hinein, am besten noch nach Indien, und dann kommen die ganzen Komponenten am vierten Tag zusammen und dann packen wir die in Storybook, da sind sie sowieso schon drin, dokumentieren noch ein bisschen und sind fertig. Also, auch ein Ansatz, aber das würde... Kann man überlegen, ob man so einen scalable Component Sprint oder infinite scalable Component Sprint baut. Willst du zehn oder 100 Komponenten? Das Ding ist darauf ausgerichtet, auch 100 Komponenten in der gleichen Qualität zu liefern. Aber ich glaube nicht, dass das deine Intention war, weiß ich aber nicht.

[00:47:37] **Leo** Ne, nicht unbedingt.

[00:47:39] **Niels** Wäre auch ein Ansatz, da würde jeder sagen, ich fall dir um den Hals, wenn du das ermöglichen kannst. Weil es ist auch gar nicht tatsächlich durch React-Komponenten gar nicht so unwahrscheinlich. Weil manche Leute sagen auch, naja hier gemeinsames Team hin oder her, letztlich ist es dem Komponentenentwickler egal, wenn alles, also der Rahmen, da ist. Dann kannst du jedem beliebigen... Also wir haben hier 20 Frontend-Entwickler, nimmst du noch zwei, wenn sie Zeit haben, dann bauen die dir auch noch eine Komponente nebenbei. Das wird die gleiche Qualität haben. Das entspricht nur nicht den agilen Prinzipien, aber wenn es tut, ist es ja eigentlich auch gut.

[00:48:17] **Leo** Ja, mir war das nur aufgefallen, einfach während des Component Sprints, dass sich das vielleicht ganz gut skalieren ließe. Wir haben es natürlich aber auch noch nicht getestet.

[00:48:27] **Niels** Man kann ja jetzt... So eine Arbeit ist ja auch manchmal, um Möglichkeiten aufzuzeigen. Ich denke nur, also entweder... Also das können ja auch zwei Ansätze sein. Entweder ihr sagt, na ja, ich nehme da auch UX-Designer rein, weil es gibt auch noch etwas zu designen. Braucht man natürlich... Ihr brauchtet das jetzt nicht, weil ihr hattet schon alle Komponenten da. Aber wir haben das schon noch so, dass da schon echt auch einfach einer Design-Entscheidungen trifft in dem Prozess. Also Detail-Design-Entscheidungen. Wie sieht denn jetzt der Active State aus? Wer entscheidet das jetzt? Jetzt dem Entwickler keine... Das nicht absprechen, dass der das nicht auch könnte, aber oft brauchst du doch einen Designer, der das dann... Weil er on-brand das macht und sein Designsystem auch hat und sagt, nein das machen wir jetzt nicht grün, das machen wir mal anders, oder wenn grün, dann ist es genau der Grünton. Ich gebe dir den, pass auf, kein Problem, hier sind die Hex-Werte, alles gut, weiter. Also ich will nur sagen, der kollaborative Ansatz, der ist ja dann... Kann entscheidender sein... Oder der skalierfähige. Man schreibt ja manchmal so einen Ausblick. Ausblick stärkere Interaktion oder Ausblick stärkere Skalierung.

[00:49:53] **Leo** Wie siehst du das denn, wenn wir jetzt einen Component Sprint machen, dann ist das ja in sich erstmal eine geschlossene Geschichte und dann kommt da fällt dann halt am Ende eine Pattern Library raus, die wir jetzt, sagen wir mal, an den Kunden weitergeben oder an das Unternehmen, oder wie auch immer. Jetzt hatte ich überlegt ob es sich nicht vielleicht lohnen würde, aus dem Kundenteam oder aus dem Unternehmen einen Entwickler mit rein zu nehmen in den Component Sprint, damit der einfach schon weiß, wie das Ganze aufgebaut ist, nach welchen nach welchen Regeln wir gearbeitet haben, und dann nachher dieses Wissen auch gut an sein Team noch weitergeben kann bei der Weiterentwicklung. Wie siehst du das?

[00:50:42] **Niels** Das ist ein guter Punkt. Das ist auch das, was ich immer mit Interdisziplinarität meine. Wenn der Component Sprint, mal eben böse gesagt, nur dazu da ist, schnell und in kurzer Zeit Code runterzurocken, dann kann man sagen, gute Idee, mach es bitte noch mehr skalierfähig, aber dann ist ja vieles was der Design Sprint hat... Du referenzierst ja auch den Design Sprint. Dann ist das ja nicht da und das ist ja eigentlich auch viel Co-Creation, Onboarding, ins Boot nehmen, mach es du deinem Produkt und so weiter. Und das hast du ja nicht, wenn du das getrennt machst. Von daher, ja, hundertprozentig unterstützenswert. Dann kriegt das auch nochmal eine andere Qualität, weil dann ist das ja sein Ding auch. Er arbeitet dann ja mit und damit hast du die Anschlussfähigkeit erzeugt. Also wenn du Design Sprint nur machen würdest, ohne Kunden, dann ist eigentlich der Gag vom Design Sprint schnell weg. Also das machen wir öfters hier bei Pitches, weil der Kunde... Da darf man ja nicht mit Kunden arbeiten, weil das ist ja ein Pitch. Wo du aber auch denkst, wie sinnbefreit ist das. Wir hätten jetzt so viele Fragen oder würden so schnell, so gerne, mal Sachen abklären. Können wir nicht, und da geht ja auch viel verloren, dann. Hier wäre ja dann wenigstens die Übergabe, dass der versteht, wie das Design System funktioniert, wie das Komponentensystem funktioniert und es auch wirklich weiter benutzen kann in der Organisation.

[00:52:17] **Leo** Genau, es ist ja auch gut, wenn das System ein lebendes System ist, was ja dann auch im Optimalfall um Komponenten erweitert wird, wenn das benötigt ist oder so. Dass das nicht ein festes, fertiges Produkt ist, sondern dass das wirklich auch benutzt wird.

[00:52:33] **Niels** Genau.

[00:52:34] **Leo** Eine Frage hätte ich noch, und zwar, da hatte ich mir auch noch Gedanken darum gemacht. Das wäre die Zielgruppe für Component Sprints. Da hatte ich mich so ein bisschen gefragt, ist das vielleicht eher was für Startups? Oder ist das eher was für Unternehmen, die eh schon ein Produkt haben und das optimieren wollen, weiterentwickeln wollen? Wie siehst du das, oder hast du da eine Idee, was da eine optimale Zielgruppe sein könnte?

[00:53:05] **Niels** Ja, vielleicht genau das beides. Startups synonym für kleine Teams oder Systeme mit wenig Stakeholdern, vielleicht muss man das so eher nennen. Wo du schnell Kontakt hast. Das sind meistens Startups. Ich glaube, alles was dir erzählt habe, zielt ja auch darauf ab, etwas größere Corporates, wo du halt dann doch andere Bedingungen findest. Da kann man das einbauen in so ein Gesamtkonstrukt. Bei, wo du jetzt sagst ist im Prinzip Refactoring des Interfaces, witzigerweise habe ich jetzt gleich einen Call mit einem potenziellen Kunden, wo es genau darum geht: Refactoring eines bestehenden Interfaces. Wo möglich aber auch schon viel Arbeit geleistet ist. Also alles, was ich gerade gesagt... Da stehen im Prinzip schon Komponenten, da gibt es ja irgendwas. Das muss man katalogisieren, analysieren und könnte sagen, wir gucken uns das bestehende Interface an, an Tag eins, und refactoren so viel wie geht, von dem was da ist. Dann hast du ja sehr viel schon gegeben. Dann musst du gar nicht mehr so viele Fragen stellen, außer du willst es halt vollkommen umbauen. Das muss geklärt sein. Für viele Sachen sagt man auch einfach naja. Also hier in dem Fall, sieht aus wie Windows 95, fühlt sich auch so an. Das überhaupt mal in eine moderne Interface-Welt zu bringen, und auch auf neue Technologien aufbauen, also das sind ja zwei Ziele, ist ja schon ein großer Gewinn. Dann kann man so einen Component Sprint machen. Wir haben das jetzt, also diese Phase haben wir nicht Sprint genannt, die heißt bei uns UI-Refactoring, wobei wir sie nicht auf eine Zeit festgelegt haben, wie lange es dauert.

[00:55:00] **Leo** Also ihr macht das dann einfach, bis es fertig ist?

[00:55:05] **Niels** Ja, das ist jetzt in dem Fall auch noch eine Anbahnungsphase, davor käme nämlich genau dieser Scoping Sprint, um das rauszufinden. Und dann könnten wir auch sagen, das passt vielleicht ganz gut, dann sind das in der UI-Refactoring-Phase, sind das drei Component Sprints. Tatsächlich könnte man das so nennen, weil da geht es nur darum, diese Komponenten soweit wie möglich mal auf den neuen Stand zu bringen, technisch als auch designerisch. Und das würden wir dann sowieso in Sprints machen da haben wir jetzt mal grob, also inklusive Design, schon so 20, 30 Personentage gerechnet. Das musst du sowieso dann in ein, zwei, drei Sprints aufteilen. Da passt es insofern ganz gut rein, weil vieles schon gegeben ist.

[00:55:49] **Leo** Okay, ja cool. Von meiner Seite aus sind alle Fragen beantwortet. Hast du noch irgendwie Fragen oder irgendwas, was dir gerade noch in den Kopf kommt?

[00:56:06] **Niels** Was mir noch im Kopf kommt, das wäre aber eher ein methodischer Ansatz. Du könntest nochmal schauen, was ist im Design Sprint? Was sind die Grundprinzipien im Design Sprint? Und welche davon verwendest du auch oder modifizierst sie, das ist ja natürlich eine andere Aufgabenstellung. Aber irgendwo darauf aufzubauen und zu sagen, ich nehme ein Prinzip dazu oder ich habe das weggelassen, weil das macht keinen Sinn in dem Fall, aber das Prinzip nutze ich halt. Was macht einen Design Sprint besonders und was haben wir davon? Ich weiß nicht, ob du das methodisch so aufbaust, aber das ist manchmal ganz schwer.

[00:56:39] **Leo** Ja, da orientiere ich mich super viel am Design Sprint, das ist schon fast ungesund. Das ist wie so eine Bibel, dieses Buch.

[00:56:48] **Niels** Ansonsten wird mich, also wenn du da weiter bist, wird mich das auch interessieren. Das wird ja wahrscheinlich jetzt nicht irgendwie geschlossen sein oder so.

[00:56:56] **Leo** Die Bachelorarbeit ist übrigens auch komplett open-source, also du kannst der Bachelor-Arbeit live sozusagen auf GitHub bei der Entwicklung zuschauen. Da kann ich dir auch gerne mal den Link schicken, wenn du Lust hast.

[00:57:07] **Niels** Ja, gerne. Das gebe ich auch gern mal irgendwo weiter. Aber spätestens im Oktober bist du dann ja fertig.

[00:57:17] **Leo** Ich sage dir auf jeden Fall auch nochmal Bescheid, wenn das alles fertig ist. Dann kriegst du da auch die fertige Version als PDF.

[00:57:26] **Niels** Super.

[00:57:26] **Leo** Dann danke ich Dir auf jeden Fall schon mal ganz herzlich, dass du dir die Zeit genommen hast, die ganze Stunde, um dich hier mit diesem Thema zu beschäftigen und mir da wirklich sehr viel weiterzuhelfen.

[00:57:37] **Niels** Jo, gerne.