

1 Interview mit Niels Anhalt, Director bei nexum AG

2 [00:00:21] **Leo** Danke nochmal, dass du dich bereiterklärst hast für das Interview. Das
3 hilft mir auf jeden Fall sehr weiter.

4 [00:00:28] **Niels** Wo stehst du denn jetzt gerade in deiner Arbeit? Also welchen
5 Status hat das jetzt? Du hast jetzt durchgeführt und...

6 [00:00:39] **Leo** Genau, korrekt. Das haben wir Ende Juli gemacht, da haben wir
7 einen Design Sprint gemacht und einen Component Sprint für ein ausgedachtes
8 Produkt. Das gibt es nicht wirklich, aber wir wollten das einfach mal als Grundlage
9 nehmen. Jetzt bin ich gerade dabei, ein bisschen zu schauen: Was hat jetzt wirklich
10 gut geklappt, wo könnte man das vielleicht auch noch ausbessern? Und genau, die
11 Interviews sind Teil dessen, weil ich es einfach sehr wichtig finde, da nochmal viele
12 verschiedene Perspektiven drauf zu kriegen.

13 [00:01:18] **Niels** Wann bist du dann fertig oder musst fertig sein?

14 [00:01:22] **Leo** 2. Oktober ist Abgabe. Ich habe noch ein bisschen Zeit auf jeden Fall,
15 aber persönliche Deadline war eigentlich der 1. September, das wird nur ein
16 bisschen schwierig. Deshalb habe ich mir die auch sehr früh gesetzt, weil dann ist da
17 noch ein bisschen Puffer. Erstmal ein kleiner Disclaimer am Anfang. Das Interview ist
18 semistrukturiert. Das heißt, ich stell dir auf jeden Fall ein paar Fragen, aber es wird
19 auch einfach viel eher wie ein Gespräch ablaufen. Wenn dir irgendwie ein Gedanke
20 in den Kopf kommt, kannst du den einfach einwerfen. Da wollte ich dann schauen,
21 dass das so offen wie möglich ist. Ich werde das Interview aufnehmen, aber nicht
22 eins zu eins veröffentlichen, sondern das ist nur für die Zwecke, dass es später dann
23 transkribiert werden kann. Das muss dann der Bachelorarbeit beigelegt werden. Und
24 wahrscheinlich wird es so 45 Minuten dauern. Perfekt. Dann können wir direkt mit
25 den ersten Fragen anfangen und zuerst mal ein paar Fragen zu dir als Person,
26 einfach um das noch mal festzuhalten. Vielleicht magst du einmal kurz beschreiben,
27 wo du arbeitest und was deine Firma macht oder was ihr so anbietet.

28 [00:02:44] **Niels** Ich bin Niels Anhalt, ich bin einer der Direktoren bei der nexum AG.
29 Nexum ist eine Beratung und Agentur im digitalen Bereich. Wir schaffen große
30 digitale Lösungen und beraten von der Strategie bis in die Umsetzung rein und
31 betreiben dann auch solche Plattformen aus dem digitalen Marketing Bereich und im
32 Bereich der digitalen Produktentwicklung. Meine Rolle ist da ja mittlerweile sehr stark
33 strategische Beratung von Kunden, aber auch Teams managen und führen. Und zu

34 meinem Hintergrund, ich habe ja mal Foto-Ingenieurswesen studiert und hab früher
35 auch mal programmiert, mal vor langer Zeit, und hatte auch lange Zeit ein Team mit
36 ungefähr 20 Entwicklern, die bei mir waren.

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

39 [00:03:41] **Niels** Ja klar, genau.

40 [00:03:45] **Leo** Gibt es bei Design Sprints Dinge, die dir besonders gut gefallen oder
41 Dinge wo du sagst, dass es da vielleicht Schwächen gibt?

42 [00:03:57] **Niels** Besonders gut finde ich, dass es so kompakt ist und wirklich alle
43 Phasen durchläuft. Eigentlich ist es ja fast schon Verkaufstool, um mal vernünftig mit
44 konzentrierter Arbeit zu leisten, Prototypen auch wirklich zu machen in kürzester Zeit
45 und dann auch einen Test darauf zu fahren. Das hat mich ja eigentlich so überzeugt
46 von der Methodik, dass man das dann auch so in Kürze schaffen kann. Wo wir jetzt
47 gerade dabei sind und was ich auch kritisch sehe ist, ob diese Problem-
48 Findungsphase am Anfang so richtig ist bzw. ob man die nicht eigentlich vorlagern
49 muss. Das kommt ja aus der Startup-Welt, wo man sagt, na gut, ich habe den
50 Gründer direkt an Bord und das gesamte Team. Und dann machen wir das jetzt
51 einfach. Wenn du das mit Kunden machst, die auch anders strukturiert sind, muss
52 man eigentlich oft auch die Phase vorverlagern, weil sonst kann man den Design
53 Sprint gar nicht richtig aufsetzen. Das ist meine ich auch bei Design Sprint 3.0, gibt's
54 ja jetzt auch, die haben auch dieses Thema angegangen oder kritisiert oder de facto
55 haben wir das auch, diese Problemklärungsphasen. Haben wir eigentlich
56 vorgelagert, weil sonst kriegst du oft gar keinen Auftrag. Wenn du das nicht schon
57 vorher genauer gefasst hast, dann ist es ja eher nur eine Rekapitulation in dem
58 Sprint und dadurch kannst du die Phase schon etwas weiter verkürzen. Und auch die
59 Experten Interviews, da ist auch immer fraglich, ob man die immer wirklich an dem
60 Tag zusammenkriegt, die Leute. Das ist auch etwas, das man eventuell auch etwas
61 öfters mal auslagern muss, ob man will oder nicht. Ist zwar schön, dass es so
62 kompakt ist, aber wenn es nicht geht, dann geht es nicht.

63 [00:05:34] **Leo** Ihr macht dann sozusagen vor dem Design Sprint noch mehr
64 Vorbereitung als eigentlich in dem Konzept angegeben. Okay, dann ein bisschen zur
65 Software Seite. Wenn ihr Produkte entwickelt, also jetzt E-Commerce oder digital
66 Produkte oder Webseiten, habt ihr da bestimmte Frameworks, mit denen ihr arbeitet,
67 was die Gestaltung angeht, oder die Entwicklung?

68 [00:06:04] **Niels** Ja klar, kommt auch drauf an, welches Level. Also wir entwickeln,
69 gerade wenn wir Frontends entwickeln, meistens mit React. Angular gab's auch ein
70 paar Projekte, aber momentan ist die Nachfrage am Markt nach React sehr groß.
71 Weil wir allerdings auch oft keine Web-Applikationen, die standardisierte UI-
72 Elemente haben, sondern eigentlich voll individualisierte E-Commerce Shops haben,
73 kann man relativ wenig von Angular weiterverwenden und das nutzt einfach nicht
74 viel, und deswegen ist React... Ist ja auch kein Framework, sondern eine Library...
75 Benutzt man da die Library und kann da viel mit machen und kombiniert mit NextJS
76 und naja den ganzen anderen Tools, die darum herum sind, um effizienter zu
77 arbeiten. Ich kenne sie jetzt alle gar nicht. Das ändert sich ja auch monatlich so
78 ungefähr.

79 [00:07:02] **Leo** Da muss man echt immer am Ball bleiben, das stimmt.

80 [00:07:03] **Niels** Das wäre jetzt der Tech Stack, und dann natürlich Git und
81 Continuous Deployment Flow. Das gesamte Setup, was man da hat, wenn man
82 etwas moderner arbeitet. So, das wäre die Technik. Ansonsten arbeiten... Unsere
83 Designer arbeiten mittlerweile fast nur noch mit Sketch und wir arbeiten mit InVision
84 als Tools und das ist auch die Übergabeschnittstelle zu den Entwicklern, also in
85 InVision die Prototypen zu haben und da dann eigentlich auch die ganzen
86 Spezifikationen, die sind da ja abgreifbar in InVision.

87 [00:07:38] **Leo** Das heißt, ihr macht für die App auf jeden Fall das Design komplett
88 selber und implementiert dann das Design auch komplett selber.

89 [00:07:49] **Niels** Ja klar, genau. Ich sehe das ja auch immer als integrierten Ansatz.
90 Wir haben oft Leute im Team, die eher den Informationsarchitektur- oder UX-Fokus
91 haben. Also, was will der Nutzer, Anforderungsaufnahme und architektonische
92 Sachen, also eher Informationsarchitektur. Und dann hast du einen eher
93 Schwerpunkt Design, also Visual Designer und Interaction Designer, der das
94 Interface gestaltet und einen Interaktionsentwickler, also kein Frontend-Dev, sondern
95 einen Interaktionsentwickler, der die Mensch-Maschine Schnittstelle entwickelt. Er ist
96 ja nicht nur Coder, der irgendwelche Codezeilen abliefert, sondern er erschafft. Er ist
97 eigentlich der Erschaffer des Interfaces, nicht der Designer und auch nicht der
98 Konzepter, der irgendwelche Sachen ausdenkt. Das ist aber eigentlich ein
99 integriertes Team. Alle drei sind zuständig, dass am Ende HTML Code entsteht, der
100 sinnvoll ist, also nutzbar.

101 [00:08:49] **Leo** Und nutzerfreundlich wahrscheinlich auch.

102 [00:08:52] **Niels** Genau.

103 [00:08:53] **Leo** Perfekt. Dann würde ich dir einmal mal ganz kurz die Grundidee von
104 Component Sprints vorstellen und das Projekt, was wir gemacht haben. Wir hatten ja
105 schon mal drüber gequatscht so ein bisschen, du hattest ja wahrscheinlich auch mit
106 Daniel wahrscheinlich schon ein bisschen drüber gesprochen. Der Component Sprint
107 wäre sozusagen der nächste Schritt nach dem Design Sprint, in meinen Augen. Und
108 das Ziel des Component Sprints ist es, dass du eine sogenannte Pattern Library an
109 Komponenten schon mal entwickelst, die nachher im Produkt dann verwendet
110 werden kann. Das sind halt so Design-Elemente, wie zum Beispiel Buttons oder
111 Eingabefelder oder auch custom Sachen, wenn halt irgendwie bestimmte Design-
112 Elemente wirklich für diese App gebraucht werden. Dann würden die in diesem
113 Component Sprint schon mal entwickelt werden, sodass man direkt danach als
114 Entwickler, der die Geschäftslogik implementiert, sich darum nicht mehr kümmern
115 muss. Die grundsätzliche Idee orientiert sich ein bisschen an Atomic Design, hast du
116 davon schon mal gehört?

117 [00:10:05] **Niels** Ja klar, genau.

118 [00:10:06] **Leo** Ich habe jetzt, einfach um ein bisschen simpler zu halten, gesagt,
119 dass wir nur Atome und Moleküle und Templates sozusagen als Basis nehmen, beim
120 Atomic Design. Du hast ja theoretisch auch noch Organismen und so, aber das habe
121 ich jetzt erst mal rausgelassen, um das hierarchisch nicht zu komplex werden zu
122 lassen. Die Komponenten würden dann in Storybook dokumentiert werden.
123 Storybook ist ein Tool, wo du sehr einfach dir Dokumentation mit aufbauen kannst.
124 Habt ihr damit schon mal gearbeitet oder sagt dir das was?

125 [00:10:42] **Niels** Ja, Kollegen von mir, ja, soweit ich das weiß. Persönlich aber noch
126 nicht, ja.

127 [00:10:48] **Leo** Von der Struktur her wird der Component Sprint ähnlich sein wie im
128 Design Sprint 2.0. Auch insgesamt vier Tage, jeden Tag von 10 Uhr bis 17 Uhr. Im
129 Vorhinein vom Component Sprint wollen wir sicherstellen, dass Storybook schon mal
130 aufgesetzt ist, dass, wenn es existierende Komponenten gibt oder ein Design
131 Systems, das schon mal irgendwie alles gesammelt wird und dass die CI-Guidelines
132 schon mal feststehen, also Fonts, Farben, Style, die Stimmung. Am ersten Tag ist
133 der Plan, dass der Prototyp, den es schon gibt, also das was dann zum Beispiel in
134 Sktech gestaltet wurde, oder in InVision, unterteilt wird, zerschnitten wird in Moleküle
135 und Atome wo man schauen kann, welche Komponenten gibt's denn da, welche

136 Komponenten kann man wie hierarchisch aufbauen. Hier wird dann auch geschaut,
137 welche Varianten es für diese Komponenten geben muss, also ein Button kann ja
138 zum Beispiel disabled sein oder ein Primary Button sein, so dass das alles schon mal
139 festgehalten wird. Dann werden diese Komponenten priorisiert, sodass man schaut,
140 welche müssen als Erstes implementiert werden, welche sind vielleicht von welchen
141 abhängig. Und last but not least ist es natürlich auch wichtig, dass das gesamte
142 Entwicklerteam da sich auf einen Codestil einigt, sich auf eine Dokumentation einigt,
143 dass da alle dann auf demselben Wissensstand sind. Am ersten Tag selber wird
144 noch nicht entwickelt, sondern es ist wirklich nur Wissensaustausch, um zu schauen,
145 dass wirklich alles schon mal vorbereitet ist. Ich denke, das ist ein wichtiger Teil
146 dafür, dass die nächsten Tage dann wirklich flüssig ablaufen. Tage zwei und drei
147 sind dann wirklich Implementation-Tage. Da wird nur entwickelt. Wir haben das bei
148 uns so gemacht, dass wir jede Komponente als Post-it einfach auf einer Wand
149 hatten. Jeder, der gerade nichts zu tun hat, nimmt sich eine der Komponenten als
150 Post-it, implementiert diese in Storybook und hängt sie dann an den fertigen Bereich,
151 sodass jeder eine gute Übersicht darüber hat was jetzt gerade schon gemacht wurde
152 was noch gemacht wird. Und der vierte Tag wäre dann der Tag wo vielleicht fehlende
153 Dokumentation noch nachgearbeitet wird, wo nochmal eine Retro gemacht wird, was
154 hat gut geklappt, was hat nicht gut geklappt. Und wo, wenn der Kunde das möchte,
155 ein Handoff passiert, dass der Kunde darin eingeweiht wird. Das Ganze... Also wenn
156 du Fragen hast...

157 [00:13:44] **Niels** Ne, soweit verstanden. Das Ergebnis ist dann eine Pattern Library...

158 [00:13:48] **Leo** Genau, richtig. Das wäre dann...

159 [00:13:52] **Niels** Eine dokumentierte Pattern Library.

160 [00:13:53] **Leo** Genau.

161 [00:13:55] **Niels** Ich habe mich eben nur gefragt, also kommt drauf an, wer ist der
162 Kunde? Und was fängt er danach damit an?

163 [00:14:05] **Leo** Jetzt haben wir das... Um das ganze mal zu testen haben wir uns ein
164 eigenes Produkt ausgedacht. Das Produkt heißt Wevent, ist eine kleine Plattform wo
165 du, ich sage jetzt mal, Veranstaltungen im kleinen Kreis planen kannst. Es geht
166 darum, wenn du z.B. deine Geburtstagsparty feierst und diese planen möchtest und
167 deine Freunde einladen möchtest, dass du etwas darüber machen kannst. War jetzt
168 einfach so eine Idee, weil sich das vom Scope relativ gut irgendwie implementieren

169 ließ und... Da haben wir in der ersten Woche einen ganz normalen Design Sprint 2.0
170 gemacht mit vier Personen in vier Tagen. Ich gebe jetzt mal gerade, wenn es geht,
171 das Fenster frei. Ich hoffe, du kannst mein Bildschirm sehen. Das ist der Prototyp,
172 der am Ende herausgekommen ist. Hier lädt wohl irgendwie das iPhone drum herum
173 nicht, aber das kannst du dir ja vorstellen. Du kannst dich da durchklicken. Wenn du
174 ein Event erstellst, musst du dich einmal kurz Einloggen mit Google. Dann hast du
175 die Möglichkeit, hier Daten festzulegen, kannst einstellen, wo deine Veranstaltung
176 stattfindet. Noch eine kleine Beschreibung anlegen. Kannst dich dann hier
177 durchklicken und dann kannst du ein Template auswählen für ein kleines Design
178 irgendwie. Und dann, im letzten Schritt, halt Gäste einladen. Das sind so die
179 Screens, also das ist ein relativ einfacher Prototyp. Den haben wir auch, so wie man
180 es im Design Sprint ja macht, dann direkt mit 5 Usern getestet, schon ein bisschen
181 Feedback gesammelt und darüber iteriert. Also das ist jetzt die iterierte Version
182 sozusagen.

183 [00:16:05] **Niels** Wie habt ihr das jetzt gebaut, mit welchem Tool?

184 [00:16:05] **Leo** Mit Framer. Framer ist ganz cool, weil du da direkt Komponenten die
185 vielleicht ein bisschen interaktiver sein müssen, also wenn ich hier zum Beispiel so
186 etwas haben möchte, das kannst du direkt in React schreiben und die einfacheren
187 Sachen kannst du aber halt genau wie in Figma oder in InVision einfach als Vektor-
188 Komponenten basteln. Genau. Den Component Sprint haben wir zu dritt
189 durchgeführt, also wir waren drei Entwickler. Auch in vier Tagen, so wie ich das
190 gerade eben beschrieben habe. Und das wäre hier das Ergebnis. Das ist jetzt
191 sozusagen ein Storybook, wo du alle Komponenten siehst, die in diesem Prototypen
192 verwendet wurden. Das wären dann zum Beispiel Buttons in ihren verschiedenen
193 Zuständen. Das wäre dieser Zeit-Picker zum Beispiel, oder Eingabefelder. Da hast
194 du halt diese ganzen Komponenten drin. Darauf aufbauend hast du Moleküle, die
195 dann zum Beispiel mehrere Komponenten oder mehrere Atome enthalten und als
196 höchste Kategorie hast du die Templates. Da geh ich mal kurz in die iPhone-Ansicht.
197 Das sind dann sozusagen die fertigen Layouts wo die Komponenten alle schon drin
198 sind, sodass man als Entwickler wirklich sieht, wie wird es benutzt. Du siehst auch
199 als Entwickler hier direkt... Ich hoffe, das lädt... Wie das alles zusammengebaut ist.
200 Du könntest sozusagen hingehen, dir den Code raus kopieren, in die App
201 reinschmeißen und dann hast du das Layout so wie es ist, mit den ganzen
202 Komponenten drin. Die Dokumentation besteht bei uns oder ist bei uns komplett auf

203 der Codeseite. Dafür geh ich mal kurz in das Repo rein. Wir haben alles so
204 dokumentiert, dass man diese Komponenten im Code Editor benutzt, dass man
205 direkt vor Ort die Dokumentation bekommt. Dieses Storybook, so wie es jetzt ist,
206 könnte man jetzt an das Entwickler-Team weitergeben, sodass die damit arbeiten
207 können und dann sozusagen ihre App damit bauen können. Hast du zu dem Konzept
208 so generelle Fragen oder irgendwas, irgendwelche Punkte, irgendwas, was dir
209 vielleicht aufgefallen ist?

210 [00:19:17] **Niels** Das Team, das den Component Sprint durchführt, sind nur
211 Entwickler. Oder ist dann noch jemand anderes dabei?

212 [00:19:25] **Leo** Wir hatten es jetzt vorerst nur mit Entwicklern gemacht, aber ich habe
213 auch schon Potential gesehen, dass man vielleicht nicht nur Entwickler mit dabei hat,
214 sondern vielleicht auch UX-Designer oder so.

215 [00:19:43] **Niels** Verstehe. Und hat das... Also gut, drei Entwickler, so gesehen zwei
216 Tage, das sind sechs Tage reine Entwicklungszeit. Hat das soweit geklappt? Also
217 habt ihr da eine gewisse Vollständigkeit erreicht für diese App?

218 [00:19:58] **Leo** Ja, total. Wir waren sogar noch viel früher fertig als wir eigentlich
219 gedacht hatten. Wir hatten am vierten Tag eigentlich fast nichts mehr zu tun. Das war
220 jetzt... Das muss man natürlich auch mal schauen. Die App war jetzt vom Scope her
221 noch relativ kompakt. Wir hatten insgesamt 29 Komponenten, die implementiert
222 wurden. Auf drei Leute verteilt ist das auf jeden Fall überschaubar. Wir haben aber
223 auch gemerkt, dass sich dieses Entwicklerteam gut skalieren lässt, weil es
224 implementiert ja jeder einzelne Komponenten, die Komponenten sind meistens
225 unabhängig voneinander. Das heißt, wenn man jetzt ein größeres Projekt hätte,
226 könnte man das Entwicklerteam auch einfach erweitern auf, sagen wir, fünf
227 Entwickler zum Beispiel. Und ich denke, dass es sich relativ gut parallelisieren lässt,
228 ohne dass es irgendwie zu Problemen käme.

229 [00:20:50] **Niels** Das habe ich mittlerweile auch gemerkt. Wenn du willst, kann ich
230 einfach jetzt was zu sagen. Vielleicht mal vorne angefangen oder eigentlich davor.
231 Was wir merken ist, dass du... Nach einem Design Sprint hast du zwar einen
232 Prototyp, den hast du auch validiert, ja. Aber es kann auch sein, dass da etwas nicht
233 so ganz funktioniert hat. Das heißt, du brauchst eigentlich aus meiner Sicht auf jeden
234 Fall noch eine zweite, entweder, ich meine, AJ&Smart macht das ja auch im
235 verkürzten zweiten Design Sprint.

236 [00:21:28] **Leo** So einen Iteration Sprint irgendwie.

237 [00:21:28] **Niels** Genau. So was brauchst du, damit du überhaupt die Qualität halt
238 dann sicherstellen kannst, außer du hast perfekt gearbeitet.

239 [00:21:42] **Leo** Das ist selten, ja.

240 [00:21:42] **Niels** Also, wenn du jetzt über so einen Kontext redest in deiner Arbeit,
241 dann müsste man ja eigentlich sagen, naja es ist nicht direkt nach dem Design
242 Sprint, sondern dazwischen gibt es eine Phase, die kann man ja auch sehr kurz mal
243 machen, die kann man aber auch... Aber das wäre jetzt auch nur eine Iteration auf
244 dem Prototypen. Was wir allerdings eigentlich immer brauchen, sobald wir mit
245 Kunden unterwegs sind, ist irgendwo eine Vision, ein fertiges Stück, wo auch der
246 Kunde einen Haken dran macht. Also Stichwort Design. Passt das in das Corporate
247 Design des Kunden, ist das ein neues Design? Wie weit weicht das ab? Wer muss
248 da alles überhaupt was gesagt zu haben, damit das wahr ist. Das schöne ist ja, beim
249 Design Sprint kannst du immer sagen, naja, ist halt ein Prototyp, da muss nicht alles
250 so stimmen und das ist irgendwie fake. Es geht ja gar nicht darum, dass es ein
251 finales Produkt ist. Jetzt wenn du direkt einen Entwicklungsschritt dranhängst, dann
252 wird es schon eher so, naja, aber das meinten wir jetzt schon ernst. Weil wir bauen,
253 auch wenn es nur acht bis zehn Personentage sind, da sind umgerechnet auch
254 schnell 10000 Euro weg, oder 15000, je nach Overhead. Da willst du ja schon in die
255 richtige Richtung dich bewegen. Also sprich was ich meine, ich glaube du brauchst
256 nicht nur einen Iteration Sprint, um die Ergebnisse des Design Sprints zu verarbeiten,
257 sondern eigentlich auch, wir hatten das jetzt mal als alles zusammen als Strategie-
258 Sprint formuliert. Du brauchst auch einen Visionsprototypen, so haben wir ihn
259 genannt, der auf Basis von dem Design Sprint Prototypen sein kann, aber der dann
260 eben auch gemacht wird. Der vielleicht noch ein paar weitere Sachen hat, die man
261 nicht im Design Sprint abtesten konnte, und insbesondere aber auf eine Fidelity
262 gekommen ist und auf eine Ernsthaftigkeit des Designs. Da könnte man jetzt auch
263 sagen, ja das möchte ich so. Und wenn es nur das Orange ist was ihr da benutzt
264 habt, und die runden Ecken. Da kannst du ja mit Kunden beliebig lange noch darüber
265 diskutieren, wenn man es auch ernst meint. Also reden können ja viele, aber wenn
266 du wirklich sagst, das ist jetzt die Richtung, klar kann man immer später noch etwas
267 ändern. Aber wenn ihr da 30 Komponenten baut dann willst du nicht nochmal das
268 komplett in Frage stellen.

269 [00:23:57] **Leo** Ja, richtig.

270 [00:23:59] **Niels** Also ich würde schon sehen, dass du dazwischen... Vielleicht ist das
271 auch nur eine Woche, da kann man einen Iteration Sprint plus einen... Wie würde
272 man das dann in dieser Terminologie nennen? Das wäre dann im Prinzip eine
273 Härtung dessen, was man da macht. Wir nennen es natürlich auch manchmal
274 Design Directions oder Design... Vielleicht ist es auch ein Style-Guide- Ansatz. Damit
275 auch zum nächsten Punkt. Das ist nicht ganz konkret, wie wir das Ding nennen, aber
276 es hat ja mehrere Nutzen. Einerseits Design Sprints iterieren, dann eine Klarheit
277 darüber, dass das wirklich auch das Ziel ist, eine Abnahme zu bekommen an der
278 Stelle, oder Bestätigung zumindest. Und auch, weil du sagst, dass die Phase davor,
279 da sollten ja wenigstens irgendwie CD-Vorgaben sein. Ja, da musst du wenigstens
280 die Farben und Formen halbwegs klar haben, ich glaube aber, du musst auch noch
281 ein paar mehr Sachen klar haben, also wirklich geklärt haben, damit das wirklich vom
282 Design her dann sitzt. Da habt ihr euch jetzt nicht so drauf konzentriert, ihr wart ja
283 auch euer eigener Kunde.

284 [00:25:20] **Leo** Ja, richtig.

285 [00:25:21] **Niels** Wenn auch der CEO plötzlich noch mit über die Farbe reden will,
286 und das macht der aber dann meistens erst nach dem Component Sprint, weil ihr so
287 schnell durch seid, dass die das gar nicht gemerkt haben. Das kommt nämlich erst in
288 drei Wochen, dann hast du irgendwann ein Problem. Das merke ich immer, sobald
289 du individueller wirst im Interface und auch viel näher Markenthemen berührst, dann
290 hast du da einfach nochmal einen anderen Abstimmungsaufwand und
291 Kreativaufwand. Unser Kreativdirektor macht sich auch gern mal einfach mal drei,
292 vier Tage Gedanken darüber, was jetzt das gesamte Bild ist und nicht nur das, was
293 im Design Sprint mal eben getestet wurde. Vielleicht kannst du das dann auch Brand
294 Sprint nennen, den müsste man dazwischen schieben. Weil wenn du das nicht
295 machst, dann baust du irgendwas, was gar nicht on-brand ist, das nichts mit der
296 Marke zu tun hat. Wie gesagt, im Startup-Modus ist das vielleicht auch egal, oder der
297 Gründer kann dann einfach sagen, Wevent geil, Orange finde ich auch gut, haut rein.
298 Hauptsache das MVP ist draußen. Sobald du aber... Da brauchst du dann noch
299 mehr.

300 [00:26:37] **Leo** Das ist gut, von dir da nochmal die Perspektive aus dem größeren
301 Unternehmen zu kriegen. Das sind ja dann doch nochmal extreme Unterschiede.

302 [00:26:44] **Niels** Ja, vielleicht kann man das auch Architektur Sprint oder sowas
303 nennen. Weil du brauchst eine Markenarchitektur, du brauchst eigentlich auch eine

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

339 jetzt für dich im Prinzip als Component Sprint auch nicht schlimm. Ich glaube nur, du
340 musst die Vorbedingungen, die da sind, zumindest dann mal dokumentieren.

341 [00:30:19] **Leo** Das ist ein guter Punkt.

342 [00:30:23] **Niels** Das trifft ähnliches wie bei Scrum. Scrum ist ein agiles Framework
343 zur Softwareentwicklung. Das lässt nur eben die gesamte konzeptionelle Arbeit, von
344 der ich gerade sprach... Die bildet Scrum nicht wirklich ab, sondern hat das alles in
345 die Rolle des Product Owners ausgelagert. Der hat einen Stakeholder und kümmert
346 sich darum, dass diese architektonischen Entscheidungen entweder schon getroffen
347 sind oder getroffen werden. Dann kommt das Team und sagt, wir wollen jetzt hier so
348 eine Leiste machen, dann sagt der PO, kein Problem, machen wir so. Also das ist
349 das Verlangen von Scrum, dass er so allwissend ist, dass er das alles machen kann
350 oder mal schnell seine Stakeholder fragt, innerhalb von wenigen Minuten, um das
351 dann zu entscheiden zu können. Ich sage immer, ein PO, der das kann, der muss
352 schon teilweise gottgleich sein. Man muss auch telepathische Fähigkeiten zu
353 ungefähr 20 bis 40 Stakeholdern haben und das innerhalb Sekunden dann abklären
354 können. Das ist ja... Weißt du, auch diese ganzen agilen
355 Softwareentwicklungsframeworks... Also gerade Scrum lagert halt viel von diesen
356 Themen einfach auf irgendwelche Rollen aus, also raus aus sich. Und kümmert sich
357 aber nicht darum und die sind ja trotzdem da. Da steht da das Dev-Team im Scrum
358 Sprint und sagt, hey laufender Motor, wir können entwickeln, wir können machen.
359 Und die Architekten sagen, ich habe noch nicht entschieden ob die Leiste da unten
360 vier oder acht oder gar nicht passt, weil ich weiß es ehrlich gesagt noch nicht, weil
361 ich habe noch keine Ahnung ob wir jetzt gerade ein Mehrfamilienhaus mit zehn oder
362 mit 50 Parteien bauen. Oder vielleicht müssen wir sogar zwei Blöcke bauen, nämlich
363 zwei Apps. Weil es geht nicht anders bei dem Use-Case. Ich habe hier noch Terra
364 Incognita. Ich habe mit acht Fachabteilungen noch gar nicht gesprochen ehrlich
365 gesagt, und die haben noch nicht ihre Anforderungen. Kann sein, dass sie alle
366 sagen, passt so. Kann auch sein, dass die sagen, oh, wir brauchen nochmal ein
367 ganz neues Gebäude. Nur, damit du die Komplexität da... Das würde ich sehen...
368 Das müsste man als Vorbedingung ja eigentlich haben. Das wird jetzt bei Scrum
369 immer auch als Null bezeichnet. Da wird halt auch alles rein gepackt, ist manchmal
370 aber auch zu klein. Manchmal... Also wir lagern eigentlich noch vor den Sprint Null
371 eigentlich noch so einen Vor-Strategie-Sprint, in dessen Rahmen dann Scoping und

372 Architektur und Designarchitektur-Arbeit stattfindet und Design Sprint stattfindet. Das
373 ist ja im Prinzip dein Kontext, in dem der Component Sprint stattfinden müsste.

374 [00:33:04] **Leo** Genau. jetzt hatte eine der anderen Personen, mit denen ich darüber
375 mal gequatscht hatte, vorgeschlagen: Wie wäre es denn wenn man den Component
376 Sprint nicht unbedingt direkt nach dem Design Sprint macht, sondern ihn nach der
377 Entwicklung der ersten Version der App wirklich legt, sodass man sozusagen die App
378 schon hat, zwar mit Komponenten die so noch nicht wiederverwendbar sind, die jetzt
379 einfach erst mal so implementiert wurden, wie man das normalerweise auch macht.
380 Wo dann die Idee ist, also falls man dann mit dieser App noch wachsen möchte,
381 dass man dann erst einmal eine Pattern Library aufbaut und die dann benutzt um
382 das zu refactoren.

383 [00:33:55] **Niels** Also würde ich nur machen, wenn man ein Startup ist, was relativ
384 schnell mit einem MVP auch an den Markt gehen muss, also mit einer App, die
385 wirklich dann an Endkunden ausgeliefert wird. Klar, dann kannst du dreckig sein.
386 Haben wir meistens nicht den Kontext, weil wir würden... Klar, Prototyping würde ich
387 nicht mit Code machen, solange wie ich kann. Also gerne auch mit Teilcodes, wie ihr
388 das jetzt bei Framer gemacht habt, weil das wäre eigentlich wahrscheinlich
389 Ressourcenverschwendung, aus meiner Sicht. Weil eben, auf der einen Seite, ihr
390 habt ja jetzt auch gezeigt, in einer Woche oder in vier Tagen könnt ihr auch massiv
391 gut schon Komponenten bauen. Also ich meine warum soll ich... Ich meine, ich
392 müsste ja genauso viel Zeit reinstecken, um diese App zu bauen. Und warum soll ich
393 zehn, 15 Personentage in etwas reinstecken, was jetzt höchstwahrscheinlich
394 Wegwerfcode ist, wenn ich eigentlich in der gleichen Zeit Komponenten haben kann,
395 die ich wiederverwendbar sind. Also, kann mal Sinn machen, weiß ich jetzt nicht,
396 aber würde ich jetzt nicht als Grundansatz eines Frameworks wählen, weil wir sagen
397 dann auch immer, lass es uns dann in dem Moment wo wir genauer wissen was wir
398 tun wollen... Dann lasst es uns auch richtig machen. Richtig machen heißt nicht,
399 dass wir uns vier Monate einschließen, damit das dann auch richtig ist. Aber ihr zeigt
400 ja eigentlich, das ist ja auch eigentlich dein Kernansatz, ja, in einer Woche kann man
401 eigentlich eine Menge schaffen, wenn man sich konzentriert. Und dann ist das auch
402 noch wiederverwendbar, wie geil ist das denn. Würde ja eigentlich dein Modell
403 konterkarieren oder weiß ich nicht. Dann könnte man den Vorteil gar nicht bringen.
404 Also ich würde sehen... Was ich nur sehe ist, dass einfach viele Sachen schon ein
405 bisschen gehärteter sein müssen, also Klarheit schaffen. Nie 100 Prozent, weil sonst

406 ist es zu Wasserfall-artig, aber zu 80 Prozent muss ich gewisse Sachen schon
407 herausgefunden haben. Sei es, wie groß ist wirklich der Umfang grob, nicht mal im
408 Detail? Wie ist die Brandarchitektur und das Brand-Design? Was ist die Vision? Muss
409 ich alles schon mal irgendwo zu 80 Prozent stehen haben. Vision vielleicht zu 100
410 Prozent. Dann kann ich auch in so einen Component Sprint reingehen. Was ich
411 überlege ist, wenn wir jetzt mal die Rahmenbedingungen haben. Hast du dich schon
412 mal mit Scrum tiefer auseinandergesetzt?

413 [00:36:27] **Leo** Noch nicht super tief. Ich kenne so ein bisschen die Grundlagen
414 davon, aber ich glaube wir hatten mal einen Entwicklungssprint nach Scrum, aber mit
415 einem externen Unternehmen. Aber das war auch nur eine Woche lang.

416 [00:36:40] **Niels** Ja. Was ich halt überlege ist, im Prinzip folgst du ja... Eigentlich
417 folgst du mit dem Schema so einem Scrum-Ansatz, so einem Scrum Sprint. Die
418 sagen, lass es uns lieber zwei Wochen machen, oder zwei bis vier Wochen. Und da
419 sind noch ein paar mehr Schritte drin. Da ist ein Planungsschritt, zwei
420 Planungsschritte drin. Da ist aber auch ein Review, eine Retro schon eingebaut, also
421 das Handoff was du hast ist dann der Review. Da gibt's ja schon die Terminologien
422 und ja, tatsächlich, in der Mitte, also rein von der Grundarchitektur, nur meist zwei
423 Wochen lang, macht man das eigentlich bei Scrum auch genauso. Du hast eine
424 Planungsphase mit zwei Schritten, wo viel Vorgaben schon erwartet werden. Dann
425 hast du eine Dev-Phase mit Daily Standups und Pipapo. Dann hast du eine Review-
426 und Retro-Phase also einerseits Review, Abgabe beim Auftraggeber, und Retro, um
427 zu lernen. Und das würde eigentlich dem so entsprechen vom Schema her. Weil das
428 überlege ich gerade. Das passt natürlich ganz gut eigentlich in ein gesamtes Scrum-
429 Framework rein.

430 [00:38:03] **Leo** Viele Ähnlichkeiten auf jeden Fall.

431 [00:38:04] **Niels** Da sagt man jetzt... Also wir sagen dann immer, nach diesem, was
432 ich dir gerade alles erzählt habe. Architektur- oder Scope Sprints. Und dann nochmal
433 so ein Sprint Null, der so für technisches Setup und Backlog-Befüllung dient. Und
434 dann gehen wir, sagen wir einfach, ja klar und jetzt gehen wir in die agilen Sprints
435 rein. Da kann es sein, dass in einem agilen Sprint in zwei Wochen oder drei Wochen,
436 eine Komponenten-Library entsteht. Es kann aber auch sein, dass das jetzt, das gibt
437 jetzt Scrum nicht mehr vor oder da gibt's keine Regel was da drin passiert, sondern
438 das entscheidet im Prinzip bei der Planung der PO und das Dev-Team, was am
439 sinnvollsten da gemacht werden muss. Es kann sein, dass man sagt komm, lass uns

440 möglichst schnell eine Komponenten-Library relativ vollständig und auf 70-80
441 Prozent bringen. Kann auch sein, dass man sagt, das ist das was wir manchmal jetzt
442 mehr machen entlang von User Story Maps, lass uns lieber einen Durchstich
443 machen. Also ich bau von den 30 Komponenten nur die zehn, die ich brauche, um
444 wieder einen Ablauf zu haben. Wenigstens ich kann ein Use-Case oder User-Story
445 abbilden. Also sprich, ich kann ein Event adden. Dann gucke ich mir nur genau die
446 Komponenten an, die ich da brauche, und baue sie aber so, dass sie vielleicht mit
447 dem Backend schon theoretisch funktionieren. Da gibt's aber nur diesen einen Use-
448 Case. Da kann man auf den Screen gehen, kann da was reintippen, drückt auf add
449 und dann erscheint das Event da. Ist aber dann auch schon in der Datenbank drin.
450 Wäre potentially shipbar. Man könnte theoretisch das ausliefern und sagen, wenn du
451 nur ein Event adden willst, dann kannst du das hier. Das ist voll funktionabel,
452 durchgetestet. Hier ist die URL, ist auch echter Code, nicht mehr geprototyped, läuft
453 alles. Das ist ja der Ansatz des Durchstichs, dass du wirklich komplett von Design
454 über Frontend-Entwicklung und Backend-Entwicklung, Deployment, die komplette
455 Nummer hast. Von daher könnte es sein, dass ein Component Sprint so eine
456 besondere Art eines Scrum Sprints ist, oder einer verkürzten... Vielleicht legt man
457 das auch sogar davor. Ich weiß es auch gerade nicht, aber es könnte sein... Also
458 wenn du willst, dass das so in einen skalierfähigeren Gesamtentwicklungs-, im
459 Prinzip, Ablauf reinpasst dann müsstest du überlegen, da brauchen wir einen Design
460 Sprint, da brauchen wir einen Component Sprint. Und was passiert danach? Das
461 meine ich mit dem Handoff, und dann ein Kunde... Also unsere Kunden sagen dann
462 auch, vielen Dank für die Komponenten, und wer entwickelt denn jetzt da weiter?
463 Also finde ich super, aber danach kommt ja wieder was. Vielleicht seid ihr das,
464 können auch andere Dienstleister sein, also alles gut. Danach geht es ja weiter und
465 dann ist ja wieder die Frage der Anschlussfähigkeit nach draußen. Warum nicht das
466 schon so machen, dass es sich in Gesamtkontext ein... Man könnte auch sagen wir
467 machen einen Component Sprint und das ist ein ganz guter Handoff in eine agile
468 Weiterentwicklung. Manchmal auch ganz gut, weil oft sind auch Scrum-
469 Entwicklungsprojekte doch sehr technisch, vielleicht auch manchmal Backend-lastig,
470 dass man sagt, komm wir lagern das noch ein bisschen aus, das sind manchmal
471 auch andere Dienstleister, die dann zum Zuge kommen, also eher so krasse IT-
472 Dienstleister. Die machen da ihre agilen Sprints, denen ist es aber auch vollkommen
473 egal, wie diese Komponenten aussehen und ob sie überhaupt aussehen. Da könnte
474 man sagen, wir haben im Prinzip eine Vorphase, die aus Design Sprint, Scoping und

475 Komponenten entsteht. Dadurch haben wir schon sehr viel vom UI abgedeckt und
476 können gut ein Handoff in einen agilen Entwicklungsprozess hinein... Weil dann
477 können die direkt damit arbeiten. Also eine Auslagerung der Komponenten. Wenn
478 man sowieso integriert arbeitet, würde ich sagen, ja, kann man machen. Wüsste ich
479 aber... Also ich würde das in dem Moment eigentlich das Team entscheiden lassen,
480 ob es sinnvoll ist, 30 Komponenten am Stück zu entwickeln oder zehn und dafür
481 noch was anderes zu machen, in dem ersten Sprint. Das sagt im Prinzip eigentlich
482 eher der Product Owner. Der sagt, guck mal, das und das wäre mir jetzt wichtig. In
483 zwei Wochen möchte ich das und das sehen.

484 [00:42:35] **Leo** Okay, also du würdest den Component Sprint nicht unbedingt direkt
485 für das gesamte Produkt machen, sondern vielleicht erst mal für einen fokussierten
486 Bereich?

487 [00:42:41] **Niels** Ja, dann aber auch eigentlich integriert in den Ablauf. Oder du
488 sagst, wir lagern den vor, in einen agilen Ablauf. Agil, da kann man jetzt konkreter
489 werden. Vor ein Scrum Projekt, weil agil ist ja alles Mögliche. Aber Scrum hat sich ja
490 durchgesetzt mit den Sprints. Vor die Scrum Sprints, um schon mal ein Paketchen zu
491 haben, das aber den agilen Prinzipien folgt. Das ist ja vielleicht wichtig. Du folgst ja
492 insgesamt mit Design Sprints und einem Component Sprint eigentlich den agilen
493 Prinzipien. Vielleicht ist das auch nochmal hilfreich für deine Arbeit, ich weiß nicht, ob
494 du das soweit erwähnt hast.

495 [00:43:25] **Leo** Bisher noch nicht, tatsächlich.

496 [00:43:28] **Niels** Weil das ist ja eigentlich so der mega Hype, Trend, was auch immer,
497 oder jetzt momentan sagen alle, agile is dead. Aber eigentlich ist da ja eine hohe
498 Qualität drin. Das ist ja genau die Qualität, die du ja auch sagst. Sehr enge, kleine
499 Teams, die gut miteinander etwas machen können, mit schnellen sichtbaren
500 Ergebnissen, also Kollaboration über Dokumentation und so weiter, da gibt's ja die
501 agilen Prinzipien. Die kannst du dir nochmal anschauen, agiles Manifesto. Da kannst
502 du auch darauf referieren, also welche der agilen Prinzipien nutzt du denn hier,
503 unterstützt du hier. Das ist mir auch bei dem Design Sprint aufgefallen, dass die mal
504 so 0,0 über Agile reden. Also fand ich ganz interessant oder spannend. Die reden
505 auch nicht über Design Thinking primär, die reden auch nicht über UX, obwohl sie
506 eigentlich alles machen. Design Thinking Methoden, die nutzen UX-Ansätze, also
507 endlich mal auch User Research und Testing und mit dieser Sprint Logik und diesen

508 Zusammenarbeit-Kollaborationen bist du eigentlich mitten in der agilen nutzen
509 eigentlich prinzipiell.

510 [00:44:39] **Leo** Das stimmt, das ist echt interessant. Ich habe es auch noch nicht
511 gehört, also in dem Kontext auf jeden Fall noch nicht.

512 [00:44:43] **Niels** Das ist übrigens, aber das ist eher ein Nebensatz, vielleicht auch
513 schon fast das Problem. Weil ich hatte auch das Gefühl, jetzt hier bei unserem
514 Meetup. Da waren auch ein paar Leute... Der eine meinte auch, naja also mit Sprints
515 kenne ich mich aus, also ich weiß, ich kann mir schon vorstellen, was bei einem
516 Design Sprint passiert. Damit meinte er aber, er kennt Scrum Sprints und den Ablauf
517 da, und dachte jetzt, naja da ist im Prinzip ein Scrum Sprint, nur für Design,
518 spannend. Dadurch ist das aber für viele Leute, die aus dieser agilen oder aus der
519 Scrum-Welt kommen, relativ schnell eingeframed in einen Sprint, so wie Scrum. Es
520 ist auch gar nicht so unähnlich. Aber eigentlich ist der Design Sprint zuerst mal was
521 komplett anderes, es benutzt nur die Prinzipien. Das nur so nebenbei. Wenn du
522 jetzt... Wenn du da so, wie soll ich sagen, man macht ja oft in solchen Arbeiten auch
523 mal so Andockpunkte, also die alle Welt verändert sich, agile, agile, Trend. Agile,
524 was sind die Prinzipien. Ah, guck mal, habe ich auch in meinem... Ah, super.
525 Eigentlich nutzt das die agilen Prinzipien, nutzt das, was Design Sprint macht. Nutzt
526 die agilen Prinzipien oder unterstützt sie, und ist damit voll anschlussfähig in eine
527 agile oder Scrum-Welt.

528 [00:45:55] **Leo** Das ist ein sehr guter Gedankenanstoß, da werde ich mich nochmal
529 mit befassen.

530 [00:46:00] **Niels** Ihr habt das glaube ich auch sowieso schon intus, also eingebaut,
531 aber noch nicht reflektiert. Muss man manchmal auch nicht, wenn man es intuitiv
532 richtig macht, aber so...

533 [00:46:25] **Leo** Richtig, Bachelor ist ja nicht intuitiv.

534 [00:46:25] **Niels** Vielleicht noch einen weiteren Punkt: Ich sehe es eigentlich als
535 wichtig oder interessant an, dass UX-Designer auch da im Team sind. Das wäre
536 zumindest auch nach den agilen Prinzipien, also man würde immer versuchen, ja
537 auch interdisziplinäre Teams zusammenzubringen. Weil sonst, ich sag jetzt mal
538 böse, verkommt das ja zu einem reinen skalierfähigen Code-abarbeit-Sprint. Da
539 können wir sagen, das können wir auch hochskalieren, ist eigentlich auch
540 vollkommen egal, die arbeiten eigentlich auch nicht zusammen, weil die können auch

541 sitzen, wo sie wollen. Wir haben die Komponenten am ersten Tag aufgeteilt.
542 Verteilen wir die in die Welt hinein, am besten noch nach Indien, und dann kommen
543 die ganzen Komponenten am vierten Tag zusammen und dann packen wir die in
544 Storybook, da sind sie sowieso schon drin, dokumentieren noch ein bisschen und
545 sind fertig. Also, auch ein Ansatz, aber das würde... Kann man überlegen, ob man so
546 einen scalable Component Sprint oder infinite scalable Component Sprint baut. Willst
547 du zehn oder 100 Komponenten? Das Ding ist darauf ausgerichtet, auch 100
548 Komponenten in der gleichen Qualität zu liefern. Aber ich glaube nicht, dass das
549 deine Intention war, weiß ich aber nicht.

550 [00:47:37] **Leo** Ne, nicht unbedingt.

551 [00:47:39] **Niels** Wäre auch ein Ansatz, da würde jeder sagen, ich fall dir um den
552 Hals, wenn du das ermöglichen kannst. Weil es ist auch gar nicht tatsächlich durch
553 React-Komponenten gar nicht so unwahrscheinlich. Weil manche Leute sagen auch,
554 naja hier gemeinsames Team hin oder her, letztlich ist es dem
555 Komponentenentwickler egal, wenn alles, also der Rahmen, da ist. Dann kannst du
556 jedem beliebigen... Also wir haben hier 20 Frontend-Entwickler, nimmst du noch
557 zwei, wenn sie Zeit haben, dann bauen die dir auch noch eine Komponente
558 nebenbei. Das wird die gleiche Qualität haben. Das entspricht nur nicht den agilen
559 Prinzipien, aber wenn es tut, ist es ja eigentlich auch gut.

560 [00:48:17] **Leo** Ja, mir war das nur aufgefallen, einfach während des Component
561 Sprints, dass sich das vielleicht ganz gut skalieren ließe. Wir haben es natürlich aber
562 auch noch nicht getestet.

563 [00:48:27] **Niels** Man kann ja jetzt... So eine Arbeit ist ja auch manchmal, um
564 Möglichkeiten aufzuzeigen. Ich denke nur, also entweder... Also das können ja auch
565 zwei Ansätze sein. Entweder ihr sagt, na ja, ich nehme da auch UX-Designer rein,
566 weil es gibt auch noch etwas zu designen. Braucht man natürlich... Ihr brauchtet das
567 jetzt nicht, weil ihr hattet schon alle Komponenten da. Aber wir haben das schon
568 noch so, dass da schon echt auch einfach einer Design-Entscheidungen trifft in dem
569 Prozess. Also Detail-Design-Entscheidungen. Wie sieht denn jetzt der Active State
570 aus? Wer entscheidet das jetzt? Jetzt dem Entwickler keine... Das nicht absprechen,
571 dass der das nicht auch könnte, aber oft brauchst du doch einen Designer, der das
572 dann... Weil er on-brand das macht und sein Designsystem auch hat und sagt, nein
573 das machen wir jetzt nicht grün, das machen wir mal anders, oder wenn grün, dann
574 ist es genau der Grünton. Ich gebe dir den, pass auf, kein Problem, hier sind die

575 Hex-Werte, alles gut, weiter. Also ich will nur sagen, der kollaborative Ansatz, der ist
576 ja dann... Kann entscheidender sein... Oder der skalierfähige. Man schreibt ja
577 manchmal so einen Ausblick. Ausblick stärkere Interaktion oder Ausblick stärkere
578 Skalierung.

579 [00:49:53] **Leo** Wie siehst du das denn, wenn wir jetzt einen Component Sprint
580 machen, dann ist das ja in sich erstmal eine geschlossene Geschichte und dann
581 kommt da fällt dann halt am Ende eine Pattern Library raus, die wir jetzt, sagen wir
582 mal, an den Kunden weitergeben oder an das Unternehmen, oder wie auch immer.
583 Jetzt hatte ich überlegt ob es sich nicht vielleicht lohnen würde, aus dem
584 Kundenteam oder aus dem Unternehmen einen Entwickler mit rein zu nehmen in den
585 Component Sprint, damit der einfach schon weiß, wie das Ganze aufgebaut ist, nach
586 welchen nach welchen Regeln wir gearbeitet haben, und dann nachher dieses
587 Wissen auch gut an sein Team noch weitergeben kann bei der Weiterentwicklung.
588 Wie siehst du das?

589 [00:50:42] **Niels** Das ist ein guter Punkt. Das ist auch das, was ich immer mit
590 Interdisziplinarität meine. Wenn der Component Sprint, mal eben böse gesagt, nur
591 dazu da ist, schnell und in kurzer Zeit Code runterzurocken, dann kann man sagen,
592 gute Idee, mach es bitte noch mehr skalierfähig, aber dann ist ja vieles was der
593 Design Sprint hat... Du referenzierst ja auch den Design Sprint. Dann ist das ja nicht
594 da und das ist ja eigentlich auch viel Co-Creation, Onboarding, ins Boot nehmen,
595 mach es du deinem Produkt und so weiter. Und das hast du ja nicht, wenn du das
596 getrennt machst. Von daher, ja, hundertprozentig unterstützenswert. Dann kriegt das
597 auch nochmal eine andere Qualität, weil dann ist das ja sein Ding auch. Er arbeitet
598 dann ja mit und damit hast du die Anschlussfähigkeit erzeugt. Also wenn du Design
599 Sprint nur machen würdest, ohne Kunden, dann ist eigentlich der Gag vom Design
600 Sprint schnell weg. Also das machen wir öfters hier bei Pitches, weil der Kunde... Da
601 darf man ja nicht mit Kunden arbeiten, weil das ist ja ein Pitch. Wo du aber auch
602 denkst, wie sinnbefreit ist das. Wir hätten jetzt so viele Fragen oder würden so
603 schnell, so gerne, mal Sachen abklären. Können wir nicht, und da geht ja auch viel
604 verloren, dann. Hier wäre ja dann wenigstens die Übergabe, dass der versteht, wie
605 das Design System funktioniert, wie das Komponentensystem funktioniert und es
606 auch wirklich weiter benutzen kann in der Organisation.

607 [00:52:17] **Leo** Genau, es ist ja auch gut, wenn das System ein lebendes System ist,
608 was ja dann auch im Optimalfall um Komponenten erweitert wird, wenn das benötigt

609 ist oder so. Dass das nicht ein festes, fertiges Produkt ist, sondern dass das wirklich
610 auch benutzt wird.

611 [00:52:33] **Niels** Genau.

612 [00:52:34] **Leo** Eine Frage hätte ich noch, und zwar, da hatte ich mir auch noch
613 Gedanken darum gemacht. Das wäre die Zielgruppe für Component Sprints. Da
614 hatte ich mich so ein bisschen gefragt, ist das vielleicht eher was für Startups? Oder
615 ist das eher was für Unternehmen, die eh schon ein Produkt haben und das
616 optimieren wollen, weiterentwickeln wollen? Wie siehst du das, oder hast du da eine
617 Idee, was da eine optimale Zielgruppe sein könnte?

618 [00:53:05] **Niels** Ja, vielleicht genau das beides. Startups synonym für kleine Teams
619 oder Systeme mit wenig Stakeholdern, vielleicht muss man das so eher nennen. Wo
620 du schnell Kontakt hast. Das sind meistens Startups. Ich glaube, alles was dir erzählt
621 habe, zielt ja auch darauf ab, etwas größere Corporates, wo du halt dann doch
622 andere Bedingungen findest. Da kann man das einbauen in so ein Gesamtkonstrukt.
623 Bei, wo du jetzt sagst ist im Prinzip Refactoring des Interfaces, witzigerweise habe
624 ich jetzt gleich einen Call mit einem potenziellen Kunden, wo es genau darum geht:
625 Refactoring eines bestehenden Interfaces. Wo möglich aber auch schon viel Arbeit
626 geleistet ist. Also alles, was ich gerade gesagt... Da stehen im Prinzip schon
627 Komponenten, da gibt es ja irgendwas. Das muss man katalogisieren, analysieren
628 und könnte sagen, wir gucken uns das bestehende Interface an, an Tag eins, und
629 refactoren so viel wie geht, von dem was da ist. Dann hast du ja sehr viel schon
630 gegeben. Dann musst du gar nicht mehr so viele Fragen stellen, außer du willst es
631 halt vollkommen umbauen. Das muss geklärt sein. Für viele Sachen sagt man auch
632 einfach naja. Also hier in dem Fall, sieht aus wie Windows 95, fühlt sich auch so an.
633 Das überhaupt mal in eine moderne Interface-Welt zu bringen, und auch auf neue
634 Technologien aufbauen, also das sind ja zwei Ziele, ist ja schon ein großer Gewinn.
635 Dann kann man so einen Component Sprint machen. Wir haben das jetzt, also diese
636 Phase haben wir nicht Sprint genannt, die heißt bei uns UI-Refactoring, wobei wir sie
637 nicht auf eine Zeit festgelegt haben, wie lange es dauert.

638 [00:55:00] **Leo** Also ihr macht das dann einfach, bis es fertig ist?

639 [00:55:05] **Niels** Ja, das ist jetzt in dem Fall auch noch eine Anbahnungsphase,
640 davor käme nämlich genau dieser Scoping Sprint, um das rauszufinden. Und dann
641 könnten wir auch sagen, das passt vielleicht ganz gut, dann sind das in der UI-
642 Refactoring-Phase, sind das drei Component Sprints. Tatsächlich könnte man das so

643 nennen, weil da geht es nur darum, diese Komponenten soweit wie möglich mal auf
644 den neuen Stand zu bringen, technisch als auch designerisch. Und das würden wir
645 dann sowieso in Sprints machen da haben wir jetzt mal grob, also inklusive Design,
646 schon so 20, 30 Personentage gerechnet. Das musst du sowieso dann in ein, zwei,
647 drei Sprints aufteilen. Da passt es insofern ganz gut rein, weil vieles schon gegeben
648 ist.

649 [00:55:49] **Leo** Okay, ja cool. Von meiner Seite aus sind alle Fragen beantwortet.
650 Hast du noch irgendwie Fragen oder irgendwas, was dir gerade noch in den Kopf
651 kommt?

652 [00:56:06] **Niels** Was mir noch im Kopf kommt, das wäre aber eher ein methodischer
653 Ansatz. Du könntest nochmal schauen, was ist im Design Sprint? Was sind die
654 Grundprinzipien im Design Sprint? Und welche davon verwendest du auch oder
655 modifizierst sie, das ist ja natürlich eine andere Aufgabenstellung. Aber irgendwo
656 darauf aufzubauen und zu sagen, ich nehme ein Prinzip dazu oder ich habe das
657 weggelassen, weil das macht keinen Sinn in dem Fall, aber das Prinzip nutze ich
658 halt. Was macht einen Design Sprint besonders und was haben wir davon? Ich weiß
659 nicht, ob du das methodisch so aufbaust, aber das ist manchmal ganz schwer.

660 [00:56:39] **Leo** Ja, da orientiere ich mich super viel am Design Sprint, das ist schon
661 fast ungesund. Das ist wie so eine Bibel, dieses Buch.

662 [00:56:48] **Niels** Ansonsten wird mich, also wenn du da weiter bist, wird mich das
663 auch interessieren. Das wird ja wahrscheinlich jetzt nicht irgendwie geschlossen sein
664 oder so.

665 [00:56:56] **Leo** Die Bachelorarbeit ist übrigens auch komplett open-source, also du
666 kannst der Bachelor-Arbeit live sozusagen auf GitHub bei der Entwicklung
667 zuschauen. Da kann ich dir auch gerne mal den Link schicken, wenn du Lust hast.

668 [00:57:07] **Niels** Ja, gerne. Das gebe ich auch gern mal irgendwo weiter. Aber
669 spätestens im Oktober bist du dann ja fertig.

670 [00:57:17] **Leo** Ich sage dir auf jeden Fall auch nochmal Bescheid, wenn das alles
671 fertig ist. Dann kriegst du da auch die fertige Version als PDF.

672 [00:57:26] **Niels** Super.

673 [00:57:26] **Leo** Dann danke ich Dir auf jeden Fall schon mal ganz herzlich, dass du
674 dir die Zeit genommen hast, die ganze Stunde, um dich hier mit diesem Thema zu
675 beschäftigen und mir da wirklich sehr viel weiterzuhelfen.

676 [00:57:37] **Niels** Jo, gerne.