

1 Interview mit Niels Anhalt, Director bei nexum AG

2 [00:00:21] **Leo** Danke nochmal, dass du dich bereiterklärt 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. Erst mal ein kleiner Disclaimer am Anfang. Das Interview
18 ist semistrukturiert. Das heißt, ich stell dir auf jeden Fall ein paar Fragen, aber es
19 wird auch einfach viel eher wie ein Gespräch ablaufen. Wenn dir irgendwie ein
20 Gedanke in den Kopf kommt, kannst du den einfach einwerfen. Da wollte ich dann
21 schauen, dass das so offen wie möglich ist. Ich werde das Interview aufnehmen,
22 aber nicht eins zu eins zu veröffentlichen, sondern das ist nur für die Zwecke, dass
23 es später dann transkribiert werden kann. Das muss dann der Bachelorarbeit
24 beigefügt werden. Und wahrscheinlich wird es so 45 Minuten dauern. Perfekt. Dann
25 können wir direkt mit den ersten Fragen anfangen und zuerst mal ein paar Fragen zu
26 dir als Person, einfach um das noch mal festzuhalten. Vielleicht magst du einmal
27 kurz beschreiben, wo du arbeitest und was deine Firma macht oder was ihr so
28 anbietet.

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

34 strategische Beratung von Kunden, aber auch Teams managen und führen. Und zu
35 meinem Hintergrund, ich habe ja mal Foto-Ingenieurswesen studiert und hab früher
36 auch mal programmiert, mal vor langer Zeit, und hatte auch lange Zeit ein Team mit
37 ungefähr 20 Entwicklern, die bei mir waren.

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

40 [00:03:41] **Niels** Ja klar, genau.

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

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

64 [00:05:34] **Leo** Ihr macht dann sozusagen vor dem Design Sprint noch mehr
65 Vorbereitung als eigentlich in dem Konzept angegeben. Genau dann ein bisschen
66 zur Software Seite. Wenn ihr Produkte entwickelt, also jetzt E-Commerce oder digital

67 Produkte oder Webseiten, habt ihr da bestimmte Frameworks, mit denen ihr arbeitet,
68 was die Gestaltung angeht oder die Entwicklung?

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

80 [00:07:02] **Leo** Da muss man echt immer am Ball bleiben, das stimmt.

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

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

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

100 integriertes Team. Alle drei sind zuständig, dass am Ende HTML Code entsteht, der
101 sinnvoll ist, also nutzbar.

102 [00:08:49] **Leo** Und nutzerfreundlich wahrscheinlich auch.

103 [00:08:52] **Niels** Genau.

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

118 [00:10:05] **Niels** Ja klar, genau.

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

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

128 [00:10:48] **Leo** Von der Struktur her wird der Component Sprint ähnlich sein wie im
129 Design Sprint 2.0. Auch insgesamt vier Tage, jeden Tag von 10 Uhr bis 17 Uhr. Im
130 Vorhinein vom Component Sprint wollten wir sicherstellen, dass Storybook schon mal
131 aufgesetzt ist, dass, wenn es existierende Komponenten gibt oder ein Design
132 Systems, das schon mal irgendwie alles gesammelt wird und dass die CI-Guidelines

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

158 [00:13:44] **Niels** Ne, soweit verstanden. Das Ergebnis ist dann eine Pattern Library...

159 [00:13:48] **Leo** Genau, richtig. Das wäre dann...

160 [00:13:52] **Niels** Eine dokumentierte Pattern Library.

161 [00:13:53] **Leo** Genau.

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

164 [00:14:05] **Leo** Jetzt haben wir das... Um das ganze mal zu testen haben wir uns ein
165 eigenes Produkt ausgedacht. Das Produkt heißt Wevent, ist eine kleine Plattform wo

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

184 [00:16:05] **Niels** Wie habt ihr das jetzt gebaut, mit welchem Tool?

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

200 als Entwickler hier direkt... Ich hoffe, das lädt... Wie das alles zusammengebaut ist.
201 Du könntest sozusagen hingehen, dir den Code raus kopieren, in die App
202 reinschmeißen und dann hast du das Layout so wie es ist, mit den ganzen
203 Komponenten drin. Die Dokumentation besteht bei uns oder ist bei uns komplett auf
204 der Codeseite. Dafür geh ich mal kurz in das Repo rein. Wir haben alles so
205 dokumentiert, dass man diese Komponenten im Code Editor benutzt, dass man
206 direkt vor Ort die Dokumentation bekommt. Dieses Storybook so wie es jetzt ist
207 könnte man jetzt an das Entwickler-Team weitergeben, sodass die damit arbeiten
208 können und dann sozusagen ihre App damit bauen können. Hast du zu dem Konzept
209 so generelle Fragen oder irgendwas, irgendwelche Punkte, irgendwas was dir
210 vielleicht aufgefallen ist?

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

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

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

219 [00:19:58] **Leo** Ja, total. Wir waren sogar noch viel früher fertig als wir eigentlich
220 gedacht hatten. Wir hatten am vierten Tag eigentlich fast nichts mehr zu tun. Das war
221 jetzt... Das muss man natürlich auch mal schauen. Die App war jetzt vom Scope her
222 noch relativ kompakt. Wir hatten insgesamt 30 Komponenten, die implementiert
223 wurden. Auf drei Leute verteilt ist das auf jeden Fall überschaubar. Wir haben aber
224 auch gemerkt, dass sich dieses Entwicklerteam gut skalieren lässt, weil es
225 implementiert ja jeder einzelne Komponenten, die Komponenten sind meistens
226 unabhängig voneinander. Das heißt wenn man jetzt ein größeres Projekt hätte,
227 könnte man das Entwicklerteam auch einfach erweitern auf, sagen wir, fünf
228 Entwickler zum Beispiel. Und ich denke, dass es sich relativ gut parallelisieren lässt,
229 ohne dass es irgendwie zu Problemen käme.

230 [00:20:50] **Niels** Das habe ich mittlerweile auch gemerkt. Wenn du willst kann ich
231 einfach jetzt was zu sagen. Vielleicht mal vorne angefangen oder eigentlich davor.
232 Was wir merken ist, dass du... Nach einem Design Sprint hast du zwar einen
233 Prototypen, den hast du auch validiert. Ja, aber es kann auch sein, dass da etwas

234 nicht so ganz funktioniert hat. Das heißt, du brauchst eigentlich aus meiner Sicht auf
235 jeden Fall noch eine zweite, entweder, ich meine, AJ&Smart macht das ja auch im
236 verkürzten zweiten Design Sprint.

237 [00:21:28] **Leo** So einen Iteration Sprint irgendwie.

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

240 [00:21:42] **Leo** Das ist selten, ja.

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

268 ändern. Aber wenn ihr da 30 Komponenten baut dann willst du nicht nochmal das
269 komplett in Frage stellen.

270 [00:23:57] **Leo** Ja, richtig.

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

285 [00:25:20] **Leo** Ja, richtig.

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

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

303 [00:26:44] **Niels** Ja, vielleicht kann man das auch Architektur Sprint oder sowas
304 nennen. Weil du brauchst eine Markenarchitektur, du brauchst eigentlich auch eine
305 Informationsarchitektur. Oft ist das, was in den Design Sprints rauskommt, das bildet
306 ja ein oder zwei Use-Cases ab, oder eine Journey. Das hat aber oft nicht den
307 Anspruch darauf, dass es vollständig ist. Ich mach mal ein Beispiel, Hauptnavigation,
308 Struktur der Applikation. Im Burgermenu kannst du alles Mögliche verstecken. Macht
309 aber vielleicht nicht die beste Usability dann aus. Momentan geht der Trend ja wieder
310 unten zu den Leisten da, vier Icons. Die waren mal irgendwann weg, jetzt kommen
311 sie wieder. Weiß nicht, ob ihr das verfolgt habt. Da gab es auch immer Artikel auf
312 Medium und so, und dann haben alle ihre Apps wieder umgebaut. Also wirklich alle
313 großen aus dem Silicon Valley, und alle in Deutschland guckten und sagen, oh
314 scheiße, jetzt muss ich die App wieder umbauen. Die sind ja doch wieder in, diese
315 blöden Icons da unten, oder ein großes in der Mitte oder wie auch immer. Aber da
316 musst du dich dann auf einmal entscheiden, was kommen denn da für Icons rein.
317 Also ich meine jetzt nicht grafisch, sondern was sind denn die Haupt-Use-Cases?
318 Hast du das schon im Design Sprint herausgefunden? Sind das dann nur drei oder
319 vier, reicht uns dieses Pattern aus? Oder muss man ein ganz anderes Pattern
320 benutzen, was auch sechs wichtige Sachen abbildet? Oder vielleicht haben wir auch
321 nur eins, wäre ja geil, dann könnten wir diese Leiste ganz weglassen, wie cool. Wann
322 entscheidest du das? Das kannst du oft... In dem Design Sprint kriegst du oft eine
323 Ahnung davon. Im Component Sprint ist es schon zu spät, weil ihr sagt, ja, ich bau
324 euch die Leiste. Wenn ihr sagt, die Leiste ist da, dann bau ich die. Wenn nicht, dann
325 nicht. Also wer entscheidet das? Also spricht dazwischen brauchst du eine Phase,
326 eine Architekturphase, die zumindest mal, da kannst du auch nie 100 Prozent sicher
327 sein, aber du versuchst ja, den Gesamtscope des Projekts zu verstehen. Deswegen
328 nennen wir das auch meistens Scoping-Phase, die einfach den Umfang definiert.
329 Das hat jetzt nicht nur was mit Auftragsvolumen zu tun, sondern auch, damit ein
330 Informationsarchitekt mal halbwegs versteht, wie viele Use-Cases haben wir denn
331 da, wie viele unterschiedliche Zielgruppen. Also den ganzen UX-Kram, den man
332 normalerweise vor UI abfragt. Das müsste wahrscheinlich parallel zum Design Sprint
333 und auch nach dem Design Sprint passieren, also so machen wir das dann, um dann
334 den Component Sprint erst starten zu können. Das ist dem Component Sprint im
335 Prinzip in sich geschlossen ja eigentlich egal, weil der Input ist halt irgendein

336 Prototyp oder irgendein Input, der sagt, wir haben diese Dinger. Am ersten Tag, also
337 an einem Tag, kannst du darauf aufsetzen und dann in die Tiefe gehen aber wenn du
338 das nicht gemacht hast, schaffst du es nicht glaub ich nicht, eine solide Architektur,
339 Informationsarchitektur zu schaffen, das muss davor passieren. Das ist ja jetzt für
340 dich im Prinzip als Component Sprint auch nicht schlimm. Ich glaube nur, du musst
341 die Vorbedingungen, die da sind, die du ja zumindest dann mal dokumentieren.

342 [00:30:19] **Leo** Das ist ein guter Punkt.

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

370 immer auch als Null bezeichnet. Da wird halt auch alles rein gepackt, ist manchmal
371 aber auch zu klein. Manchmal... Also wir lagern eigentlich noch vor den Sprint Null
372 eigentlich noch so einen Vor-Strategie-Sprint, in dessen Rahmen dann Scoping und
373 Architektur und Designarchitektur-Arbeit stattfindet und Design Sprint stattfindet. Das
374 ist ja im Prinzip dein Kontext, in dem der Component Sprint stattfinden müsste.

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

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

404 konterkarieren oder weiß ich nicht. Dann könnte man den Vorteil gar nicht bringen.
405 Also ich würde sehen... Was ich nur sehe ist, dass einfach viele Sachen schon ein
406 bisschen gehärteter sein müssen, also Klarheit schaffen. Nie 100 Prozent, weil sonst
407 ist es zu Wasserfall-artig, aber zu 80 Prozent muss ich gewisse Sachen schon
408 herausgefunden haben. Sei es, wie groß ist wirklich der Umfang grob, nicht mal im
409 Detail? Wie ist die Brandarchitektur und das Brand-Design? Was ist die Vision? Muss
410 ich alles schon mal irgendwo zu 80 Prozent stehen haben. Vision vielleicht zu 100
411 Prozent. Dann kann ich auch in so einen Component Sprint reingehen. Was ich
412 überlege ist, wenn wir jetzt mal die Rahmenbedingungen haben. Hast du dich schon
413 mal mit Scrum tiefer auseinandergesetzt?

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

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

431 [00:38:03] **Leo** Viele Ähnlichkeiten auf jeden Fall.

432 [00:38:04] **Niels** Da sagt man jetzt... Also wir sagen dann immer, nach diesem, was
433 ich dir gerade alles erzählt habe. Architektur- oder Scope Sprints. Und dann nochmal
434 so ein Sprint Null, der so für technisches Setup und Backlog-Befüllung dient. Und
435 dann gehen wir, sagen wir einfach, ja klar und jetzt gehen wir in die agilen Sprints
436 rein. Da kann es sein, dass in einem agilen Sprint in zwei Wochen oder drei Wochen,
437 eine Komponenten-Library entsteht. Es kann aber auch sein, dass das jetzt, das gibt

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

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

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

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

496 [00:43:25] **Leo** Bisher noch nicht, tatsächlich.

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

507 eigentlich alles machen. Design Thinking Methoden, die nutzen UX-Ansätze, also
508 endlich mal auch User Research und Testing und mit dieser Sprint Logik und diesen
509 Zusammenarbeit-Kollaborationen bist du eigentlich mitten in der agilen nutzen
510 eigentlich prinzipiell.

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

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

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

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

534 [00:46:25] **Leo** Richtig, Bachelor ist ja nicht intuitiv.

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

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

551 [00:47:37] **Leo** Ne, nicht unbedingt.

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

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

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

574 das machen wir jetzt nicht grün, das machen wir mal anders, oder wenn grün, dann
575 ist es genau der Grünton. Ich gebe dir den, pass auf, kein Problem, hier sind die
576 Hex-Werte, alles gut, weiter. Also ich will nur sagen, der kollaborative Ansatz, der ist
577 ja dann... Kann entscheidender sein... Oder der skalierfähige. Man schreibt ja
578 manchmal so einen Ausblick. Ausblick stärkere Interaktion oder Ausblick stärkere
579 Skalierung.

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

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

608 [00:52:17] **Leo** Genau, es ist ja auch gut, wenn das System ein lebendes System ist,
609 was ja dann auch im Optimalfall um Komponenten erweitert wird, wenn das benötigt
610 ist oder so. Dass das nicht ein festes, fertiges Produkt ist, sondern dass das wirklich
611 auch benutzt wird.

612 [00:52:33] **Niels** Genau.

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

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

639 [00:55:00] **Leo** Also ihr macht das dann einfach, bis es fertig ist?

640 [00:55:05] **Niels** Ja, das ist jetzt in dem Fall auch noch eine Anbahnungsphase,
641 davor käme nämlich genau dieser Scoping Sprint, um das rauszufinden. Und dann

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

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

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

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

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

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

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

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

673 [00:57:26] **Niels** Super.

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

677 [00:57:37] **Niels** Jo, gerne.