**Interview mit (anonym), Mobility Research in der Automobilbranche**

[00:01:38] **Leo** Danke, dass du dich bereiterklärt hast für das Interview, das freut mich sehr. Wenn du Lust hast, könaddddnen wir direkt einsteigen.

[00:01:49] **(anonym)** Gerne. Ich weiß aber nicht, ob ich dir so viele Fragen beantworten kann.

[00:01:56] **Leo** Wir schauen mal. Wir machen das Interview semistrukturiert, das heißt, ich stelle dir ein paar Fragen, aber ich möchte auch, dass das ein offenes Gespräch wird, einfach um ein bisschen Input zu bekommen, um ein bisschen Ideen auszutauschen und so. Als Disclaimer: Das Interview werde ich aufnehmen für spätere Transkription, werde es aber nicht so 1:1 veröffentlichen, sondern es muss einfach der Bachelorarbeit als Transkription beigelegt werden.

[00:02:32] **(anonym)** Und anonymisierst du dann die Referenten?

[00:02:37] **Leo** Also wenn dir das wichtig ist, kann ich das gerne machen.

[00:02:44] **(anonym)** Das wäre auf jeden Fall gut.

[00:02:45] **Leo** Soll ich den Namen und die Firma anonymisieren oder was ist dir da am liebsten?

[00:02:51] **(anonym)** Das wäre mir auf jeden Fall am liebsten.

[00:02:57] **Leo** Ja, dann werde ich das machen. Perfekt.

[00:03:02] **(anonym)** Gut.

[00:03:05] **Leo** Die erste Frage ist eigentlich, wo und in welcher Position du arbeitest. Das können wir dann vielleicht überspringen. Die zweite Frage wäre dann in deinem Bereich. Was bietet deine Firma an, beziehungsweise an was arbeitet dein Team?

[00:03:34] **(anonym)** An neuen Konzepten für zukünftige Fahrzeuge, an neuen Feature-Ideen und was wir meistens machen ist Prototypen erarbeiten, die wir dann mit Kunden testen.

[00:03:49] **Leo** Sind das dann Hardware-Prototypen oder ist es hauptsächlich Software?

[00:03:53] **(anonym)** Meistens eine Kombination. Viele von unseren Features haben eine Quelle im Fahrzeug. Das heißt, die sind irgendwie mit einem Fahrzeug verbunden. Das können dann... Die Sachen, die wir brauchen, um unsere Prototypen zu bauen, sind natürlich digitale Sachen. Also Software in Kombination mit irgendwelchen Modulen, die wir in Autos einbauen, zusätzlich.

[00:04:28] **Leo** Habt ihr als Team schon mal mit Designs Sprint gearbeitet oder ist dir das ein Begriff?

[00:04:34] **(anonym)** Ja, das ist mir ein Begriff und das haben wir auch schon mal gemacht.

[00:04:37] **Leo** Und hat dir das gefallen? War das eine gute Erfahrung, so ein Design Sprint?

[00:04:44] **(anonym)** Ja, ich denke das macht Sinn, wenn man eine Idee hat, das mal besser zu verstehen, wie die Kunden das eigentlich sehen könnten. Etwas zu bauen, das nicht so ein hohes Fidelity Level hat, und damit dann schon erstes Feedback einzuholen, das macht sicherlich Sinn. Es ist natürlich immer so eine Frage, wie repräsentativ es dann am Ende ist, weil man sich natürlich ein paar Leute aussucht – zumindest haben wir das so gemacht – um unsere Prototypen am Ende der Woche zu testen und das natürlich nicht großzahlig ist, und irgendwie repräsentativ. Wenn irgendwas einfach Quatsch ist, dann merkt man das ziemlich schnell.

[00:05:25] **Leo** Okay. Also das ist der einzige Punkt, wo du denkst, dass das kritisch ist, oder gibt es auch noch andere Punkte wo du denkst, das könnte kritisch sein?

[00:05:39] **(anonym)** Naja, es wäre natürlich am besten, wenn man die Design Sprints direkt mit Kunden zusammen machen würde. Vielleicht kann man die auch von Anfang an bei der Ideengenerierung mit einbeziehen. Das ist aber natürlich relativ schwierig umzusetzen. Ich glaube, das würde es nochmal verbessern.

[00:05:57] **Leo** Ich gehe mal davon aus, dass ihr bei (anonym) eine feste Corporate Identity habt. Habt ihr auch feste Pattern Libraries, wie zum Beispiel ein User Interface aussehen soll? Oder wie geht ihr so etwas an?

[00:06:16] **(anonym)** Wir ignorieren solche Sachen dann erstmal noch komplett. Wir haben jetzt natürlich ein bestimmtes Design für unsere App und da gibt es Guidelines. Das ignorieren wir aber auf dem Niveau erst mal noch. Ob das Logo jetzt so oder so aussieht ist erstmal nicht so wichtig.

[00:06:35] **Leo** Das heißt, ihr baut das Design erstmal einfach für die Funktionalität und der Rest kommt dann später?

[00:06:40] **(anonym)** Richtig. Man kann das natürlich kritisieren und sagen, ja gut, man bekommt aber gar keinen richtigen Eindruck, wie nachher aussieht. Aber wir denken immer, erstmal müssen die Needs geprüft werden. Also funktionieren denn die Sachen, die wir da anbieten wollen, unabhängig von dem Design?

[00:06:58] **Leo** Okay, das ergibt auf jeden Fall Sinn. Gut. Dann würde ich mal ganz kurz das Konzept vorstellen, woran ich jetzt in der Bachelorarbeit arbeite, bzw. was ich da ausarbeite. Hattest du das Exposé gelesen?

[00:07:14] **(anonym)** Ja, das habe ich gesehen.

[00:07:16] **Leo** Perfekt. Dann weißt du ja grob zumindest, worum es geht. Es geht darum, dass ich die Entwicklung von MVPs, die ja eigentlich nach dem Design Sprint kommen, optimieren möchte. Und das dadurch erreichen möchte, dass vor der Entwicklung des MVPs ein Component Sprint angesetzt wird, in dem die ganzen UI-Komponenten, also das ganze Design vom Produkt, schon mal als Code-Komponenten gebaut wird oder implementiert wird, sodass nachher die Entwickler, die die Geschäftslogik implementieren, die App wirklich bauen, sich nicht mehr um das Design kümmern müssen, sondern nur noch um die Logik.

[00:08:03] **(anonym)** Das heißt, wir hätten eigentlich so eine Art Design-Bibliothek umgesetzt. Also so Buttons und Standardsachen wären alle schon umgesetzt und wir müssten die eigentlich nur aus der Bibliothek holen.

[00:08:16] **Leo** Richtig, genau. Das sind dann feste Komponenten, die kann man dann einfach direkt implementieren.

[00:08:25] **(anonym)** Ja, cool.

[00:08:25] **Leo** Der Component Sprint orientiert sich ein bisschen an dem Atomic Design Konzept, sagt dir das etwas?

[00:08:28] **(anonym)** Nein, das kenne ich nicht.

[00:08:33] **Leo** Okay. Atomic Design ist ein Konzept von Brad Frost, und er hat sich gedacht, die traditionelle Methode wie man früher Websites entwickelt hat – das Konzept kommt aus 2012 – ist ein bisschen doof, weil es da sehr viel zu Redundanzen kommt. Ein Designer baut dann eine Seite wirklich als komplette Seite. Und dann hast du vielleicht, sagen wir, vier Buttons auf der Seite, die aber alle leicht anders aussehen. Und was Brad Frost sich da gedacht hat ist, wir unterteilen die Seite erst einmal – daher der Begriff Atomic Design – in einzelne Komponenten. Er benutzt dabei ein bisschen die Begriffe aus der Physik. Er spricht von Atomen, Molekülen, Organismen und so weiter. Atome sind dabei zum Beispiel Buttons. Das sind die kleinsten Komponenten, die es gibt. Buttons, Eingabefelder und so weiter. Moleküle bestehen dann aus mehreren Atomen, also zum Beispiel ein Suchfeld könnte ein Eingabefeld in Kombination mit einem Button sein. Oder vielleicht oben noch ein Label drüber. Templates sind dann das fertige Layout, in dem die ganzen Moleküle und Atome zusammengefasst sind. Das bietet schon mal eine sehr schöne Hierarchie irgendwie. Daran kann man sich schön orientieren. Deshalb habe ich das einfach als Grundlage genommen. Wir haben, um das Ganze zu testen, ein eigenes Produkt im Design Sprint ausgearbeitet. Das war jetzt einfach ein Testprodukt, das existiert nicht wirklich. Das ist eine kleine App, wo du im kleinen Rahmen Veranstaltungen planen kannst, mit Freunden zusammen. Den Design Sprint haben wir mit vier Leuten in vier Tagen durchgeführt. Das ist ein bisschen die verkürzte Variante, also der Design Sprint 2.0. Und das ist der Prototyp, der dabei herausgekommen ist. Hier fehlt jetzt leider ein bisschen das iPhone drumherum, aber du kannst dir das ja vorstellen. Das hier sind die Screens. Also hier hast du halt die Möglichkeit, hinzugehen und ein Event zu erstellen, eine Uhrzeit hinzuzufügen und zu sagen, wo dieses Event stattfindet. Dann hast du die Möglichkeit, ein Template auszuwählen, um das Ganze ein bisschen zu personalisieren und dann kannst du das mit Freunden teilen. Es ist eine sehr simple App, wir wollten den Scope erst mal recht klein halten. Das ist der Input für den Component Sprint, dieser Prototyp. Der ist ja dadurch, dass er im Design Sprint war, schon von Usern mehr oder weniger validiert. Wir haben schon Feedback bekommen, wir haben über dieses Feedback nochmal eine Iteration gemacht und den Prototyp nochmal ein bisschen angepasst. Und der Output vom Component Sprint ist im Grunde genommen eine Pattern Library, wo du jetzt zum Beispiel sagen kannst, das ist mein Button. Den gibt es in verschiedenen Zuständen, zum Beispiel disabled, in verschiedenen Farben, mit Icons. Oder ich nehme hier mal ein Eingabefeld. Das sind die verschiedenen Komponenten, die wir jetzt aus dem Prototyp extrahiert haben und als Code-Komponenten implementiert haben. Als Entwickler hast du hier direkt auch die Möglichkeit zu sehen, im Code, wie ist das denn eigentlich implementiert. Du könntest jetzt als Entwickler hingehen, dir dieses Stück Code hier rauskopieren, in deine App einfügen und dann hast du die Komponente so, wie sie hier dargestellt ist. Hier hast du auch die Möglichkeit direkt zu sehen, wie sähe das denn zum Beispiel auf dem iPhone aus und, wenn ich hier mal in die Templates gehe, zum Beispiel auf die Event Preview, dann siehst du hier auch, das sieht schon in etwa so aus wie der Prototyp. Da hättest du als Entwickler auch die Möglichkeit, einfach hinzugehen und zu sagen, das ist der Code für den Prototyp, den kannst du dir so rauskopieren in deine App und dann hast du das Layout zumindest fest und musst dich wirklich nur noch um die Logik kümmern. Die Struktur vom Component Sprint wird so ein bisschen wie der Design Sprint sein, allerdings nur vier Tage. Jeder Tag geht von 10:00 Uhr bis 17:00 Uhr, einfach damit diejenigen, die daran teilnehmen, vielleicht morgens noch ein bisschen was für die Arbeit erledigen können, falls da noch etwas ansteht. Grundsätzlich ist der Component Sprint in drei Schritte unterteilt. Der erste Schritt ist die Vorbereitung. Der zweite Schritt die Implementierung und der dritte Schritt ist dann Dokumentation und Handoff an Entwickler. In der Vorbereitung geht es erstmal darum, den Prototyp zu nehmen und zu schauen, welche Komponenten gibt es im Prototyp, wie könnte man das unterteilen? Welche Hierarchie kann man da vielleicht nehmen? Welche Komponenten sind von anderen Komponenten abhängig? Und es geht so ein bisschen darum, auch zu schauen, dass das gesamte Team, was den Entwicklungsstil und den Dokumentationsstil angeht, auf dem gleichen Wissensstand ist. Am zweiten Tag und am dritten Tag werden diese Komponenten implementiert. Wir haben das bei uns so gemacht, dass jede Komponente einfach ein kleines Post-it bekommen hat, wo dann draufsteht, das ist die Komponente, die hat die und die Eigenschaften sieht in etwa so aus. Und jeder Entwickler kann sich so ein Post-it nehmen, das implementieren und, wenn er fertig ist, das dann einfach wieder zurückhängen. Am letzten Tag wird das Ganze dann dokumentiert und, wenn das nötig ist, an das Entwicklerteam, das das Produkt implementiert, weitergegeben, also das Handoff dann, sozusagen. Das ist erst mal so – jetzt habe ich relativ viel am Stück geredet – in etwa das Konzept, grob wie ich mir das vorgestellt hatte. Hast du da erst mal fragen zu?

[00:15:43] **(anonym)** Ja. Du sagtest, das wird dann irgendwie an das Entwicklerteam übergeben. Wieso ist keiner von dem Entwicklerteam am Anfang schon dabei? Oder ist einer dabei?

[00:15:56] **Leo** Das ist eine gute Frage. Darüber habe ich mir auch schon Gedanken gemacht. Ich war mir damals nicht sicher, aber bisher kam wirklich viel Feedback auch von den anderen Leuten, die ich gefragt habe, dass das eine wichtige Sache wäre. Deshalb denke ich auch, einen Entwickler direkt vom Entwicklerteam mit dabeizuhaben wäre auf jeden Fall gut.

[00:16:16] **(anonym)** Ich könnte mir vorstellen, dass das hilft, wenn der diese Sachen mitdefiniert, der sollte aber nicht die anderen überstimmen, also die Anderen sollten immer noch mehr Leute sein. Aber der versteht dann am Ende auch ganz gut, was die Leute, die die App gestalten, sich ausgedacht haben. Vielleicht hilft das.

[00:16:36] **Leo** Ja, das denke ich auch. Und dann kann er dieses Wissen auch direkt mitnehmen ins Entwicklerteam.

[00:16:40] **(anonym)** Dann hat er auch nicht so eine Ablehnung gegen irgendwas, weil er es selber mitgestaltet hat.

[00:16:49] **Leo** Okay, cool. Siehst du denn in den Component Sprints, so wie ich dir die jetzt beschrieben habe, Potential? Gibt es da Stellen wo du sagst, das ist besonders interessant?

[00:17:05] **(anonym)** Ja. Wir haben natürlich immer die Frage… Wenn wir so eine App oder ein Feature bauen, wo eine App enthalten ist, wollen wir das auch mit Storyboards malen. Wir versuchen dann immer, so zu malen, wie der Ablauf ist, und malen dann immer auch die Bildschirm-Ansicht dazu. Wenn das dann so ein bisschen standardisiert wäre schon, wäre das vielleicht nicht schlecht. Ich frage mich aber, ob man nicht sogar aus dem Gemalten ohne Component Sprint nachher die Komponenten ableiten kann. Also ich würde es quasi rückwärts machen. Das kann auch gehen. Wenn du jetzt das schon fertig hast und du hast schon diese Bibliothek, dann ist es viel einfacher für mich, wenn ich die Sachen aus der Bibliothek direkt nutze. Wobei, ich bin ja auch nicht der Entwickler. Das ist ein bisschen das Problem für mich. Ich muss einfach nur eine gemalte App abgeben. Das heißt, ich bin vielleicht eigentlich der Kunde für die Component Sprints.

[00:18:06] **Leo** Sondern dann eher das Team, das danach kommt?

[00:18:09] **(anonym)** Der Entwickler, genau. Ich frage mich dann, warum muss ich vier Tage Komponenten ableiten?

[00:18:17] **Leo** Das heißt, du siehst dich nicht als Teil dieses Sprints, sondern eher, dass das Entwicklerteam oder vielleicht sogar ein externes Team das übernimmt?

[00:18:29] **(anonym)** Ich müsste eigentlich Teil von dem Sprint sein, in dem wir die App definieren, oder das Produkt. Und ob der Button jetzt so oder so aussieht, ist für mich eigentlich total egal. Ich brauche eigentlich nur die Features umgesetzt. Wie es aussieht entscheide ich ja am Ende sowieso nicht. Das heißt, eigentlich müsste das Team, das nachher die Sachen programmiert, für sich den Component Sprint einmal durchführen für ihre ganzen Sachen, die sie im Auftrag haben, und sich dann aus der gemeinsamen Bibliothek bedienen. Ich glaube, dass wir das eigentlich gar nicht brauchen, außer wir sagen, wir wollen jetzt, dass wir ein Mockup von einer App bauen, das ganz schnell zusammengeklickt ist und wir haben dafür ein Tool und können dann die Komponenten nutzen, die dann da schon vorprogrammiert sind. Wir klicken uns das so zusammen. Das können wir natürlich machen. Aber da brauche ich jetzt eigentlich nicht die Komponenten aus meiner App abzuleiten, da kann ich eigentlich einfach Standardkomponenten nehmen. Ich finde die Idee cool, aber ich frage mich, was ist der Mehrwert für mich persönlich?

[00:19:42] **Leo** Okay. Das ist schon mal sehr gutes Feedback. Das beantwortet auch schon die zweite Frage, die ich mir noch aufgeschrieben habe, und zwar: Wo könnte es zu Problemen kommen? Das wäre ja dann eigentlich schon die Antwort dazu. Wenn du dich in diese Rolle eures Entwicklerteams versetzt, könntest du dir denn da so einen Component Sprint generell vorstellen?

[00:20:33] **(anonym)** Wo ich es benutzen könnte wäre, wenn ich ohne Einbindung von irgendwelchen Entwicklern etwas bauen will. Dann kann ich das benutzen. Nur dann ist halt die Frage, ob ich da direkt mit so einem Design... Guck mal, ich kann doch irgendein Tool benutzen, wo irgendwelche vorgefertigten Buttons sind und so. Die kann ich doch einfach nutzen. Ich würde es mir einfach zusammenklicken und ich mache – weißt du, ich bin jetzt nicht der Designer, der das (anonym)-Identity Marketing Design macht. Das heißt, ich würde das eh nicht erfinden. Ich bin dafür nicht zuständig. Ich kann das, und darf das gar nicht. Das heißt, irgendwer müsste das standardisiert schon mal machen und mir das zur Verfügung stellen. Oder ich nehme einfach irgendwas, was ich finde. Dann muss ich eigentlich nur einmal durchgehen und gucken, wo sind die Design-Komponenten, die ich habe, und die dann konsistent einsetzen.

[00:21:48] **Leo** Das heißt, im Fall von eurer Firma – die ist ja auch schon sehr groß – würde es vielleicht eher Sinn machen, das einmal global zu machen. Also so ein Komponentensystem aufzubauen und das nicht auf Projektbasis zu sehen.

[00:22:01] **(anonym)** Ich glaube, die würden das einmal aufsetzen: Hier sind unsere Standardbuttons und wenn ihr ein MVP bauen wollt, dann macht ihr das einfach damit. Dann könnte jeder das nutzen, das wäre perfekt.

[00:22:16] **Leo** Siehst du denn jetzt außerhalb von eurem Unternehmen noch Zielgruppen, wo der Component Sprint vielleicht interessant sein könnte? Zum Beispiel Startups oder andere Bereiche?

[00:22:36] **(anonym)** Klar. Ich denke, was du eben gesagt hast als Argument, dass nicht jeder den Button neu erfinden muss. Da macht es Sinn, wenn alle, also Grafiker, Entwickler, usw., die an so etwas arbeiten, sich aus so einem abgestimmten Katalog bedienen. Das macht voll Sinn. Deswegen sollte jedes Entwicklerteam das machen.

[00:22:59] **Leo** Perfekt. Von meiner Seite aus war es das auch schon an Fragen. Hast Du noch irgendwelche Fragen oder Gedanken, die dir gerade noch im Kopf herumschwirren zu dem Thema?

[00:23:12] **(anonym)** Habt ihr denn vor, irgendwie so eine Umgebung zu entwickeln, in der man dann solche Component Sprints easy abwickeln kann, oder machst du das nur für deine Arbeit?

[00:23:25] **Leo** Das Ziel ist es schon, nachher Component Sprints auch wirklich kommerziell anbieten zu können, so wie Design Sprints ja auch von Design Sprint Agenturen angeboten werden. Wir greifen aber hauptsächlich, was die Software angeht, auf bestehende Systeme zurück. Da gibt es schon sehr gute Systeme, um so Pattern Libraries zu dokumentieren und zu bauen. Und da müssen wir dann nicht das Rad neu erfinden. Das einzige, was wirklich neu erfunden ist, ist die Struktur von diesem Sprint.

[00:24:00] **(anonym)** Interessant. Finde ich spitze. Es macht sicherlich Sinn, da mal drüber nachzudenken und das mal aufzubereiten.

[00:24:09] **Leo** Cool, perfekt. Dann danke ich dir für das Interview. Das hat mir sehr guten Input gegeben und wird mir auf jeden Fall für die Bachelorarbeit sehr viel weiterhelfen. Ich wünsche dir noch einen ganz schönen Arbeitstag und melde mich auf jeden Fall nochmal, wenn die Bachelorarbeit fertig ist.

[00:24:36] **(anonym)** Ja, schick die mal rüber.