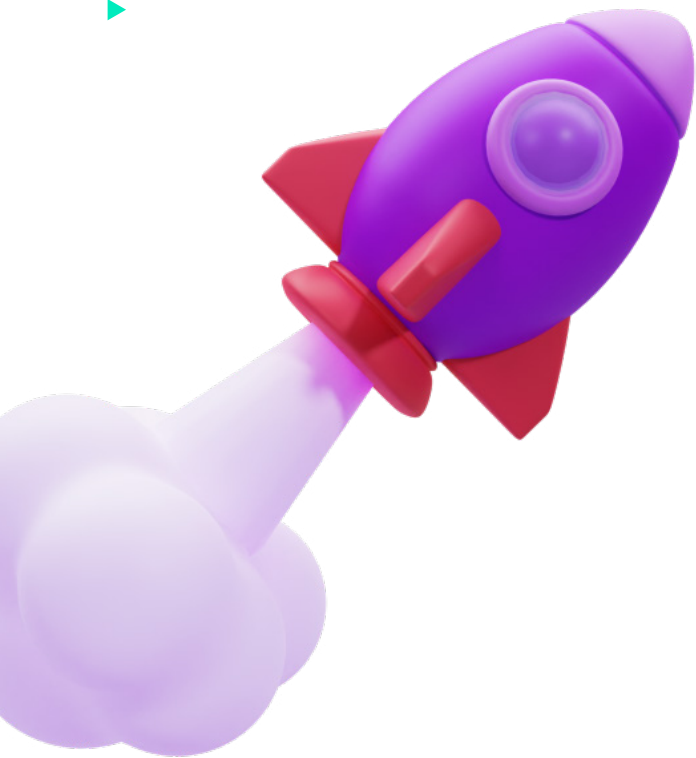


Quality meets Innovation:

► Menschzentrierte Medizintechnik



Inhalts- verzeichnis



03 Einleitung

04 Grundlagen der Produktentwicklung

05 Besonderheiten bei eHealth und Medizintechnik

- Menschen im Mittelpunkt - heterogene Anforderungen, hohe Erwartungen
- Ethik und Sicherheit

06 Bias und Annahmen: Wie solide ist die Grundlage deines Produkts?

08 Designing the right thing: Product Discovery

- Der Problemraum als solide Wissensgrundlage
- Phase 1: Map the Problem
- Phase 2: Sketch
- Phase 3: Decide

12 Ausblick: Prototyp und Test

- Zusammenfassung:
Hohe Qualität & Innovation
von Anfang an mit Product
Discovery

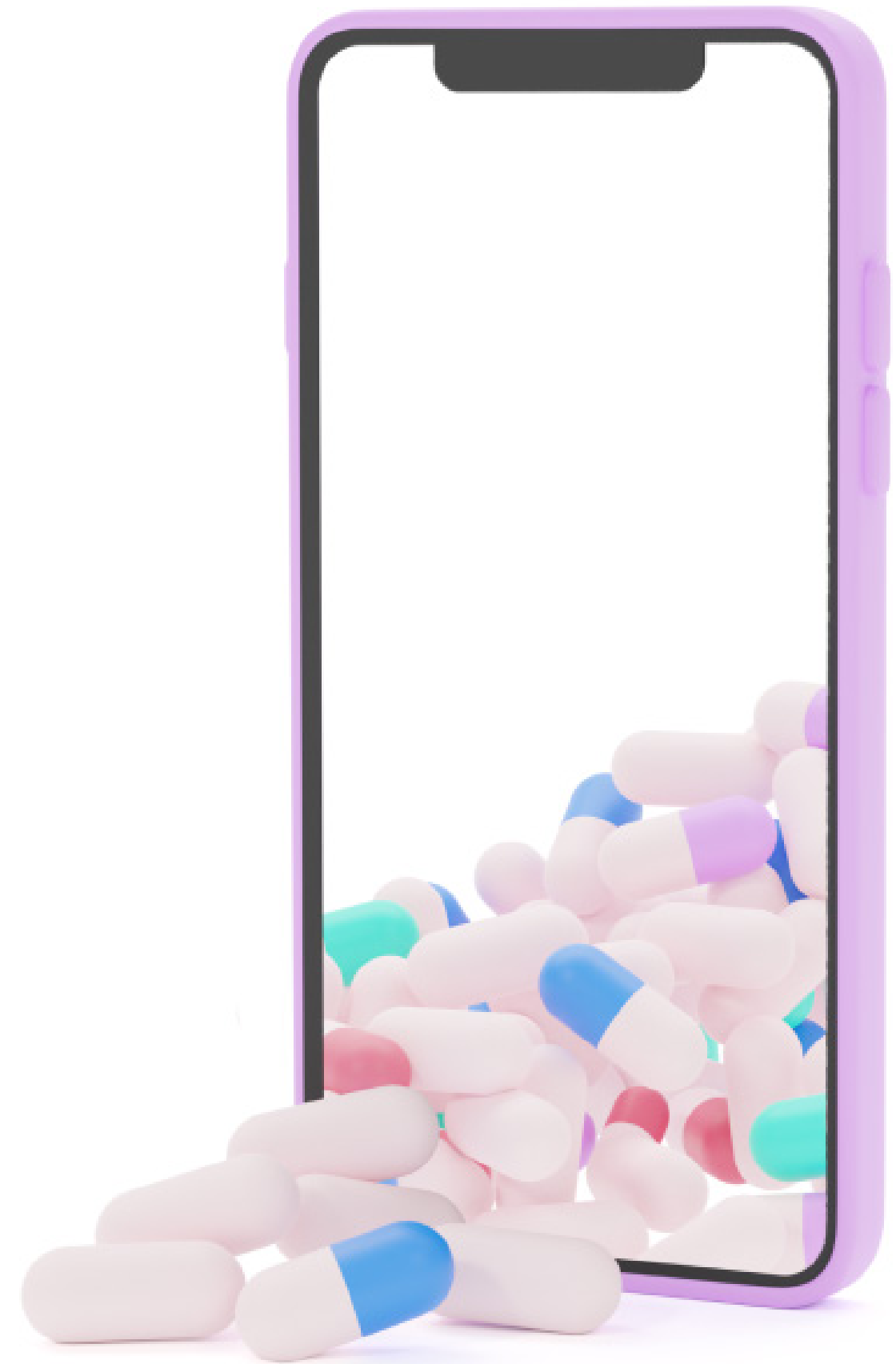
- Links + Leseempfehlungen

15 Kontakt

Einleitung

Software-Produkte in der Medizintechnik zu entwickeln, stellt Projektverantwortliche vor besonders vielschichtige, anspruchsvolle Herausforderungen. Diverse Bedürfnisse und Erwartungen aller Menschen, die damit Berührungspunkte haben, müssen berücksichtigt werden. Die klassischen Stakeholder des Herstellers bringen Anforderungen aus Unternehmenssicht mit, setzen wirtschaftliche Ziele - nicht selten müssen Entscheidungen mehrerer Menschen aus unterschiedlichen Abteilungen moderiert werden und ineinander greifen. Dann die Nutzenden: die Bandbreite reicht von hochspezialisiertem, medizinischem Personal über Servicetechniker:innen bis hin zu Privatpersonen. Eine Sonderstellung haben die Patient:innen. Auch wenn sie keine direkten Nutzenden sind, muss die Anwendung je nach Sicherheitsklasse gewährleisten, dass diese nicht zu Schaden kommen. Hierzu müssen regulatorische Auflagen erfüllt werden.

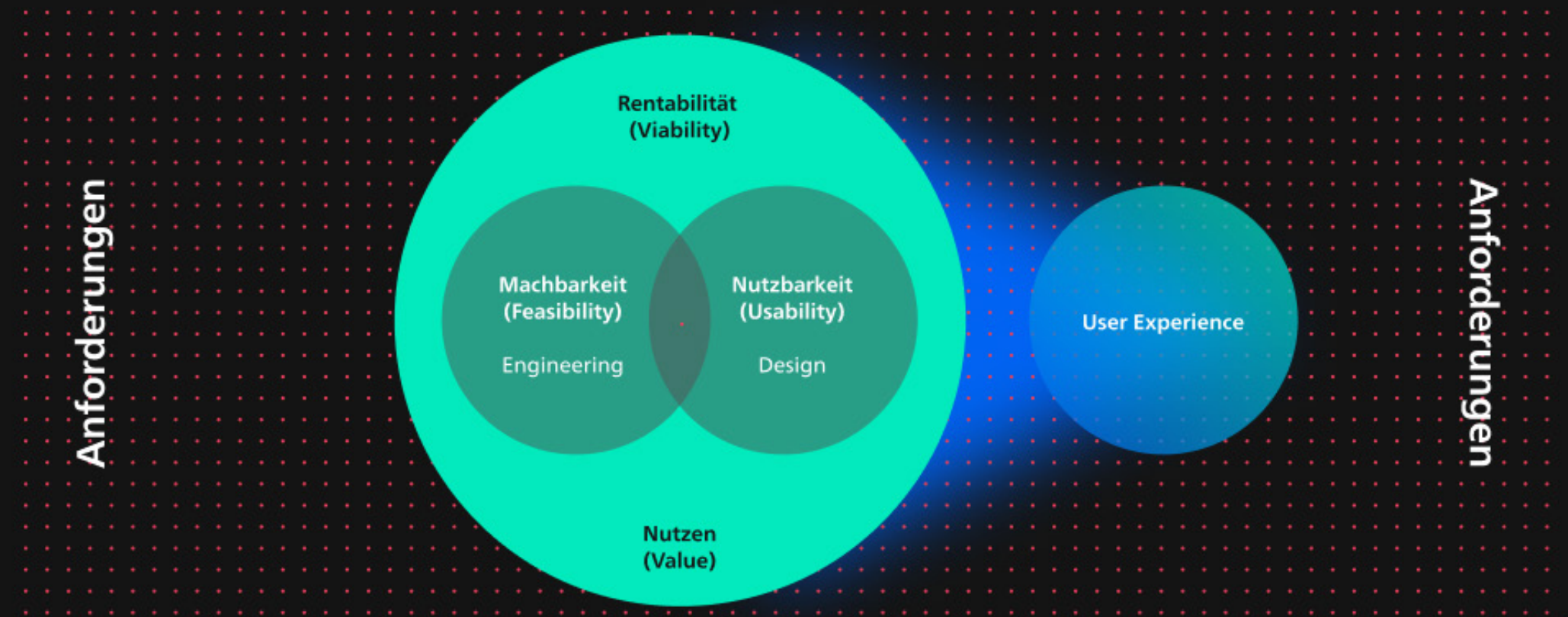
Vorgaben zu Prozessen, Dokumentationen und damit auch technische Rahmenbedingungen stecken das Feld der Möglichkeiten ganz klar ab. Wie schafft man unter diesen Bedingungen wettbewerbsfähigen Mehrwert, hebt sich gleichzeitig durch Innovation ab und erreicht eine hohe Qualität? In diesem Whitepaper soll es darum gehen, wie man schon sehr früh solide Grundsteine für eine lösungsorientierte, nachhaltige Arbeitsweise legt. Wir gehen darauf ein, wie man sich bei all der Regulatorik, hohen technologischen Standards und engen Rahmenbedingungen Raum für innovative Lösungswege freihält.



Grundlagen der Produkt- entwicklung

Moderne Software wird agil, also iterativ inkrementell erarbeitet. Agile Entwicklungsmethoden haben sich etabliert, weil sie zur Bewältigung der Komplexität von Anforderungen, Rahmenbedingungen oder sich verändernden Technologien einen idealen Rahmen bieten. Schnellere Entwicklung von kleinen Paketen und das damit verbundene frühere Lernen ist bekanntermaßen effizienter und schafft Wettbewerbsvorteile gegenüber dem Versuch, mit nur einem „Schuss“ ein fertiges Produkt zu erschaffen (um dann festzustellen, dass sich die Welt schon weiter gedreht hat). Unternehmenskulturen zielen inzwischen darauf ab, allen Menschen in der Softwareentwicklung idealen Nährboden für beste Ergebnisse und kontinuierlich hohe Qualität zu bieten. Große Beratungshäuser und Innovationsagenturen haben die Etablierung von nutzerzentrierten Problemlösungsframeworks wie „Design Thinking“ vorangetrieben, sodass Software immer mehr ganzheitlich als Produkt entwickelt wird und echten Mehrwert für Menschen schafft. So einfach sich das zusammen fassen lässt, so vielfältig sind auch die Fallstricke und Herausforderungen selbst für erfahrene Unternehmen in der IT-Branche.

Komplexität der Softwareentwicklung



Besonderheiten bei eHealth und Medizintechnik

► *Schauen wir uns genauer an, worauf es in der Medizintechnik-Softwareentwicklung ankommt. Zu den oben genannten üblichen organisatorischen und technologischen Herausforderungen der Software-Entwicklung kommen bei zulassungspflichtigen Medizinprodukten bzw. DiGA Rahmenbedingungen und Anforderungen hinzu, die für den Entwicklungsprozess von Beginn an maßgebend sind.*

Menschen im Mittelpunkt - heterogene Anforderungen, hohe Erwartungen

Wofür wird medizinische Software gemacht? Auf der einen Seite, um Menschen präventiv zu begleiten und bei der Genesung zu unterstützen. Auf der anderen Seite gilt es rund um die Patient:innen herum Prozesse zu verschlanken, Kollaboration von Fachbereichen zu ermöglichen, Erkenntnisgewinn aus der Verarbeitung großer Datenmengen zu ermöglichen. Patientenwohl, Behandlungseffektivität und Nutzungseffizienz sind die Faktoren, die den Erfolg von Software ausmachen. Die Anforderungen können so unterschiedlich sein wie das Leben: Ein Klinikalltag bedeutet häufig zeitkritische Situationen, das Tragen von Schutzkleidung oder kognitive Herausforderungen wie z.B. Multitasking. Gerade dann, wenn höchste Aufmerksamkeit auf den Patienten gerichtet sein sollte, wäre folglich eins der Qualitätsziele ein System, welches sich ganz dem Menschen anpasst und nicht umgekehrt. Das System wäre eines, das dabei unterstützt, dass menschliche Fehler nicht gemacht werden können und kognitiv nur so wenig Last wie notwendig erzeugt. Im Kontakt mit Patient:innen dagegen ist eher eine sensible, vertrauensvolle Kommunikation auf einfachem Niveau notwendig. Bei Applikationen für den privaten Gebrauch, therapiebegleitenden Anwendungen müssen komplexe medizinische Zusammenhänge für Laien verständlich dargestellt werden, damit die Anwendung zum gewünschten Ziel führt. Nutzende brauchen Anleitung, das Gefühl von Sicherheit und Vertrauen. Heterogenität, technischer Affinität oder kognitiven und körper-

lichen Fähigkeiten sind nur einige der vielfältigen Merkmale, die es in einer Anwendung zu berücksichtigen geben kann - ganz abgesehen von der Berücksichtigung der gängigsten Betriebssysteme und Endgeräte.

Dazu kommen außerdem die stark gewachsenen Erwartungen an Usability und User Experience: Durch die Allgegenwärtigkeit von Apps mit sehr guter User Experience, die jede:r im Privaten nutzt, steigen die Erwartungen an Applikationen im medizinischen Kontext. Verständlicherweise ist der Frust vorprogrammiert, wenn die Software im beruflichen Kontext den Apps auf dem privaten Mobiltelefon mehrere Jahrzehnte hinterherhinkt. Diese Aspekte zeigen auf, wie wichtig es ist, Bedürfnisse und Erlebnis dieser heterogenen Gruppen von Menschen - den Nutzenden, Behandelnden und Behandelten - bei modernen medizinisch-gesundheitlichen Anwendungen noch viel stärker in den Mittelpunkt zu stellen als in anderen Branchen.

Ethik und Sicherheit

Mediziner:innen, Behandelnde, therapeutische Fachkräfte, Pflegepersonal, Healthcare Professionals - sie alle handeln und behandeln nach ethischen Standards, die durch Regularien in der Software abgebildet werden sollen. Normen wie IEC 62366, ISO 62304 oder die FDA-Richtlinien für den US-amerikanischen Markt sind integraler Bestandteil des regulatorischen Rahmens. Sicherheit und Wirksamkeit der Technologie müssen gewährleistet und nachgewiesen werden. Alle möglichen, nicht ausschließbaren Fehlerfälle (sowohl mensch-

liche als auch technologische) müssen bis ins letzte Detail bedacht und dokumentiert sein. Dies nimmt im Kontext der perspektivisch ganzheitlichen, sektorenübergreifenden Patientjourney Dimensionen an, welche die Komplexität stark erhöhen. Wo bisher ein Mensch ein ausgedrucktes Papier in einem Umschlag überbracht hat, muss eine Software den Schutz der personenbezogenen Daten sowie Daten- und Informationssicherheit garantieren. Sie muss technologische Standards für die Kommunikation erfüllen und den Datenaustausch mit anderen bestehenden Systemen ermöglichen. Echte Interoperabilität zwischen Geräten und Informationssystemen ist aktuell noch weit von nahtlosen Experiences entfernt, die man aus Consumer-Anwendungen gewohnt ist. Ethische Standards und verpflichtende Vorgaben bezüglich aller Sicherheitsaspekte bestimmen nicht nur die Entwicklung, sondern den gesamten Lebenszyklus des Produkts - von der Konzeption bis zur Markteinführung und darüber hinaus.



Bias und Annahmen: Wie solide ist die Grundlage deines Produkts?

Ein großer, unterschätzter Stolperstein in der Software-Entwicklung liegt schlichtweg im Menschen selbst. In der Psychologie sind inzwischen mehr als 100 Verhaltensmuster bekannt, die als sogenannte „cognitive bias“ - also kognitive Verzerrungen bezeichnet werden. Niemand ist demnach frei davon, denn diese Bias sind einfach menschlich. Das Gefährliche ist aber, dass sie unbewusst unser Denken und Entscheiden beeinflussen. Warum spielen sie hier eine Rolle?

Während die Software entwickelt wird, müssen unzählige Entscheidungen von vielen Menschen getroffen werden. Es geht häufig um Abwägungen zwischen technischer Umsetzbarkeit, erwartetem Mehrwert, Nutzerbedürfnissen und wirtschaftlichen Aspekten. Die Tragweiten der Entscheidungen sind nicht immer klar absehbar. Je später festgestellt wird, dass eine Entscheidung falsch war, desto höher ist der Aufwand der Korrektur.

Diese nicht zu unterschätzende Verhaltensmuster sollte man wenigstens kennen:

Confirmation Bias: Das Confirmation Bias bezeichnet die Neigung, Informationen so wahrzunehmen und auszuwählen, dass sie die eigenen Ideen bestätigen.

Too much invested to quit syndrome: Dieses Syndrom, auch bekannt als „Sunk Cost Fallacy“, beschreibt die Tendenz, dass wir eine Idee, die sich nach einiger Zeit als nicht zielführend herausstellt, immer weiter verteidigen. Selbst wenn es rein rational betrachtet viel günstiger wäre, einfach alles wegzuwerfen und neu anzufangen.

Gender und Technology Bias: Insbesondere in der IT sind überwiegend weiße, männliche Personen beschäftigt, die von modernster Technik umgeben sind. Dies führt sehr leicht auch schon strukturell dazu, dass unbewusst „vergessen“ wird, dass alles auch ganz anders sein kann und somit Lösungen unter falschen Annahmen angegangen werden.

Insbesondere bei Anwendungen im gesundheitlich-medizinischen Bereich müssen Entscheidungen wohl informiert und überlegt getroffen werden. Die möglichen schwerwiegenden Risiken und die erhöhte Komplexität in der Healthtech-Branche erfordern offenbar eine noch größere Vorsicht bei der Identifikation von Risiken und der Erhebung von Anforderungen. Die gründliche Analyse des Nutzungskontexts, das Wissen über fachliche Inhalte, die kritischen Anforderungen an Sicherheit, Technologie, Hardware sind alle gleichermaßen mit hoher Wichtigkeit im Entwicklungsprozess zu berücksichtigen.



Wer sich also davor schützen möchte, mit seiner Produktidee blind in eine falsche Richtung zu laufen, sollte beachten:

Cognitive Bias vermeiden: Entscheidungen gemeinsam in einem Team mit diversen Perspektiven treffen.

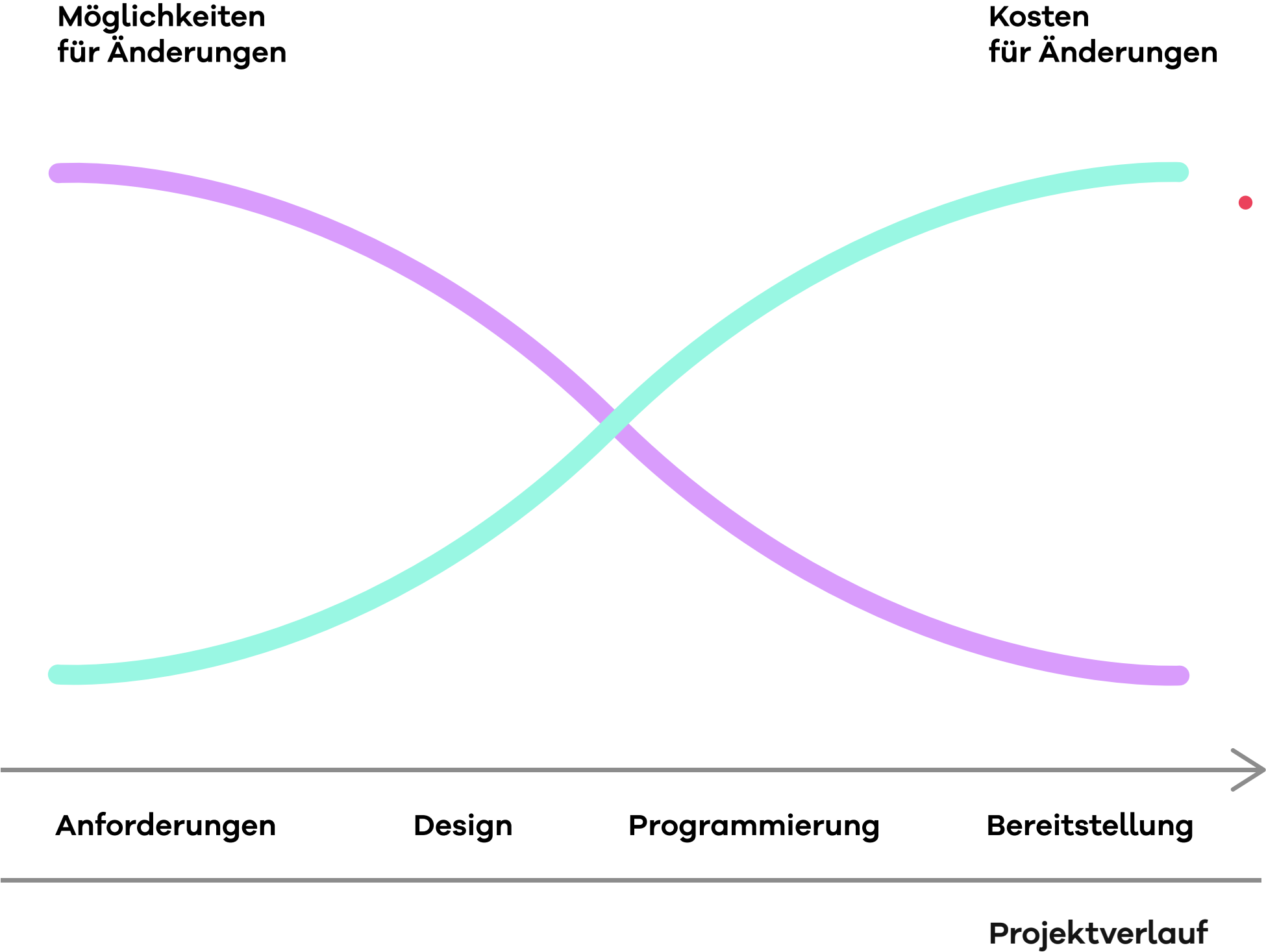
Too much invested to quit Syndrom vermeiden: Nicht zu große Zeiträume setzen, in denen Ideen nur so gut wie nötig, aber so günstig wie möglich ausgearbeitet und getestet werden. Der Google Design Sprint ist z.B. ein Format, der aus diesem Gedanken heraus entstand.

Bewusste Unterscheidung zwischen validen Informationen und Annahmen: Unvalidierte Informationen (z.B. Meinungen, Einzelfeedback, indirekte Berichte über andere) sind erstmal als Annahmen zu bewerten. Wie riskant ist es, Entscheidungen auf der Grundlage dieser ungeprüften Annahmen zu treffen? Je nach Einschätzung sollten dann Überprüfungen der Annahmen eingeplant werden.

Umgang mit Gender+Technical Bias, Annahmen über Nutzende: User Experience Design und seine Methoden von Beginn an einsetzen, um auf der Grundlage valider Informationen zu entscheiden und Innovationspotentiale nicht zu verschenken.

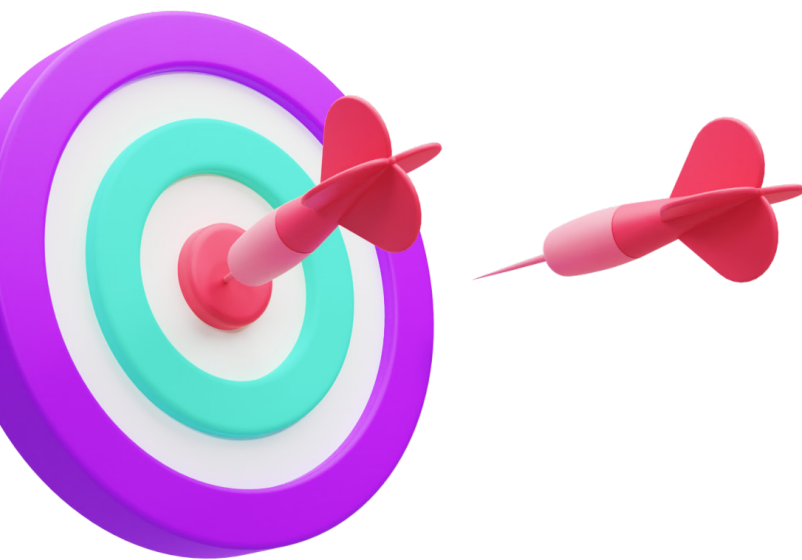
Ist das viel aufwändiger? Wer von Beginn an nach höchstem UX Qualitätsstandard nach ISO 9241-210:2019 mit allen relevanten Methoden arbeitet und vorausschauend dokumentiert, deckt bereits einen Großteil der Anforderungen nach IEC 62366, dem „Design Input“ der FDA oder auch Teile der Anforderungsanalyse für den Software-Lebenszyklus nach ISO 62304. Die Qualität, die mit dieser Vorgehensweise erreicht wird, ist jedoch höher als von der Regulatorik gefordert. Denn das nach IEC 62366 geforderte Usability-Engineering gewährleistet lediglich eine gute Usability.

The Cost of Changes



Designing the right thing:

Product Discovery



Studien belegen, dass Produktideen sehr häufig scheitern, wenn sie nicht das richtige Problem adressieren und damit kein Bedarf auf dem Markt vorhanden war.

Damit sollte bereits die initiale Entscheidung, was man für ein Produkt anbietet, nicht unterschätzt und leichtfertig getroffen werden.

Hierzu hilft es bereits, sich zu Beginn zu fragen, für wen welches Problem durch das Produkt gelöst wird. Es müssen auch nicht immer Probleme sein. Eis und Kaffee verkaufen sich auch ohne, dass sie Probleme lösen: Welche Chance kann mein Produkt so aufgreifen, dass es etwas für die Nutzer:innen signifikant verbessert?

Product Discovery zielt darauf ab, solche Unsicherheiten so weit zu reduzieren, dass das richtige Produkt für die richtige Zielgruppe entwickelt wird. Sie kann eingesetzt werden sowohl, um ein besseres Verständnis auf der „grünen Wiese“ zu erlangen oder um zu späteren Zeitpunkten Potentiale für Features oder Iterationen zu identifizieren.

Die Product Discovery setzt aber bewusst den Fokus auf den Problemraum. Das Team nimmt sich vor der Umsetzung (Delivery) die Zeit, fundamentale Fragen gründlich zu klären. Es wird Verständnis für Nutzerbedürfnisse aufgebaut und potenzielle Herausforderungen identifiziert. Die gewonnenen Erkenntnisse dienen nicht nur als Basis für eine präzise Problemdefinition, sondern schaffen eine solide Wissensgrundlage für die Entwicklung im späteren Verlauf und stärken Teamzusammenhalt und gemeinsamen Fokus.

Der Problemraum als solide Wissensgrundlage

Der Mehrwert einer Product Discovery liegt darin, dass sie klassische UX Research Methoden kombiniert mit Kreativmethoden aus dem Innovations-Umfeld - immer mit dem Ziel in realitätsnahen Tests möglichst früh zu lernen und zu evaluieren, welche Lösungswege weiter verfolgt und welche verworfen werden sollten. Unsicherheiten werden proaktiv adressiert, die Effizienz des Entwicklungsprozesses gesteigert und letztlich eine bedarfsgerechte und qualitativ hochwertige Lösung geschaffen. Durch das systematische Vorgehen wird vermieden, dass Teams Entscheidungen voreilig und unzureichend begründet „aus dem Bauch heraus“ treffen (u.a. Schutz vor Cognitive Bias) oder sich zu früh auf vermeintlich vielversprechende Lösungswege festlegen, ohne Alternativen getestet zu haben. Stattdessen ermöglicht die Product Discovery iterativen Erkenntnisgewinn, bei dem schrittweise vorgegangen wird. Diese Vorgehensweise reduziert das Risiko von Fehlentwicklungen, während sie die Entwicklung auf die Bedürfnisse der Nutzenden und Stakeholder fokussiert.

Das Ziel ist jedoch nicht, alles existierende Wissen zu erlangen, sondern gerade ausreichend, um gezielt Tests durchzuführen und frühzeitig mögliche Fehler und Sackgassen zu erkennen („fail fast, fail early“). Die ersten Thesen, Ideen und Tests dienen als Ausgangspunkt für weitere Iterationen und bieten die Möglichkeit, das Produkt kontinuierlich zu verbessern und an die sich entwickelnden Anforderungen anzupassen. Durch diesen zielorientierten Ansatz wird eine effektive und ressourcenschonende Produktentwicklung ermöglicht, die auf einem fundierten Verständnis des Problemraums basiert.

Phase 1: Map the Problem

Die Product Discovery wird eröffnet, mit einem Workshop als Auftakt zu weiteren Aktivitäten, die alle ein Ziel haben: Das Team beschäftigt sich mit der Thematik und ergründet systematisch den Problemraum.

Gerade am ersten Workshoptag wird ein gemeinsames Verständnis geschaffen und erste Thesen (z.B. User Journey Map und Proto-Persona) formuliert. Dabei stellt sich relativ schnell heraus, wo das Wissen nur auf Spekulationen beruht, wo also Wissenslücken sind. Gemeinsam wird geplant, wie und wo mehr Informationen beschafft werden können. Für den weiteren Verlauf sollte ausreichend Zeit, jedoch mit kurzen Abstimmungszyklen, eingeplant werden, um auf erwartete sowie unerwartete Erkenntnisse reagieren zu können. Denn nach dem Workshop arbeiten alle an ihren „Hausaufgaben“.

Informationen, die entweder im Workshop zusammengetragen werden oder im weiteren Verlauf untersucht werden sollten:

Stakeholder-Analyse

Wer wird Entscheidungen beeinflussen und mit wem sollte man frühzeitig in den Dialog treten? Für die Dynamik im Team und die strategische Kommunikation innerhalb der eigenen Organisation kann es hilfreich sein, Stakeholder bereits früh in den Prozess einzubeziehen.

Nutzungskontext-Analyse

Dies nimmt in der Regel den größten Raum im explorativen Part der Product Discovery ein: Nach Bedarf sind alle Methoden v.a. des qualitativen User-Research einsetzbar. Vor allem Interviews

(mit Nutzenden, aber auch nutzernahen Personen innerhalb der Organisation) oder Hospitationen bzw. Job Shadowings sind sehr hilfreich und leicht umzusetzen. Hier sollte schon für eine Marktzulassung die Dokumentation berücksichtigt werden und klar sein, für welche Märkte das Produkt sein soll.

Anforderungen durch Regulatorik und Markt

In welchen Ländern muss ich welche Auflagen erfüllen? Wie sieht der Wettbewerb aus? Die frühe Kenntnis aller einschränkenden Faktoren stellt sicher, dass die Entwicklung von Anfang an genauer ausgerichtet werden kann.

Abgeschlossen wird die „Map the Problem“-Phase wieder durch einen Workshop, zu dem sich das gesamte Team zusammenfindet. Das Ziel ist die präzise Formulierung des zu lösenden Problems auf der Grundlage aller gesammelten Informationen. Durch kluges „Framing“ wird die Entscheidungsfindung strukturiert moderiert. Dieser Workshop legt den Grundstein für die nachfolgenden Phasen, in denen konkrete Lösungsansätze entwickelt und evaluiert werden.

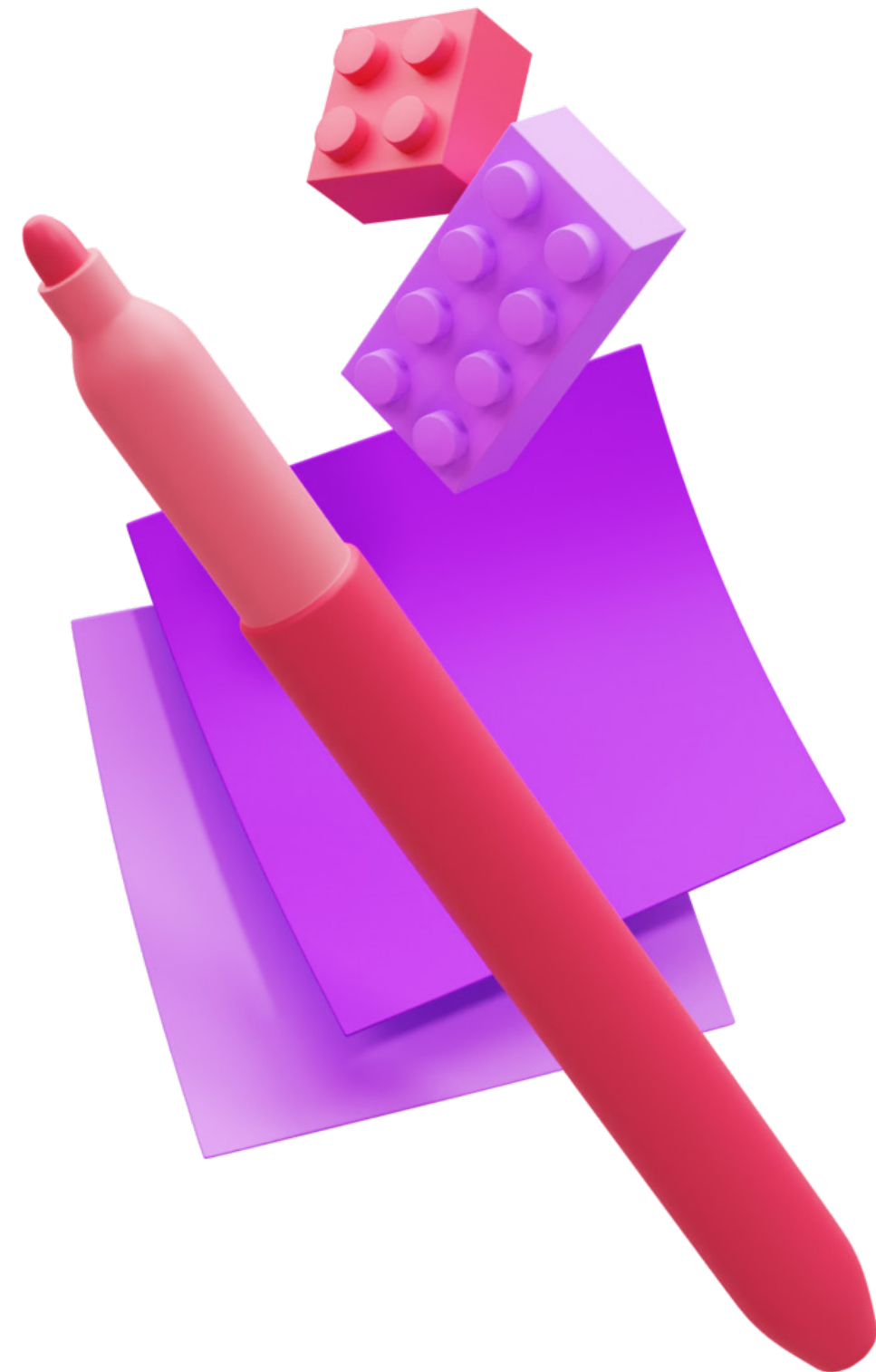
Phase 2: Sketch

Auf die Recherche folgt nun wieder eine Phase mit Workshops. Diese Phase schafft eine inspirierende Atmosphäre, in der Innovation gefördert und innovative Denkansätze hervorgebracht werden können. Der „Sketch“-Workshop zielt darauf ab, gemeinsam möglichst viele Ideen zu generieren und sie in groben Zügen visuell auszuarbeiten.

Durch den Einsatz verschiedener Kreativmethoden werden Teammitglieder dazu angeregt, in unterschiedliche, auch unkonventionelle Richtungen zu denken. Der Wert einer professionellen Moderation zeigt sich vor allem, wenn alle Teammitglieder gleichermaßen Ideen in Form von Wireframes visualisieren - auch diejenigen, die vorher nie gestalterisch gearbeitet oder gezeichnet haben. Der Fokus liegt auf der Vielfalt der Lösungsansätze, um einen reichhaltigen Fundus für die weiteren Schritte der Produktentwicklung bereitzustellen. Die gegenseitige Diskussion und Bewertung der (Teil-)Ideen erfolgt erst so spät wie möglich, um den Raum für kreatives Denken so weit wie möglich zu öffnen.

Phase 3: Decide

Erst in diesem abschließenden Workshop mit dem gesamten Team erfolgt eine systematische Bewertung der generierten Lösungsansätze. Die Bewertung beinhaltet eine Einschätzung ihrer Machbarkeit unter Berücksichtigung aller Aspekte wie Kosten, Zeitrahmen, Chancen und regulatorischen Aspekten - jeweils aus der Perspektive aller Anwesenden im Workshop. Da dies jedoch zu diesem Zeitpunkt auch spekulativen Charakter hat, liegt der Fokus darauf zu priorisieren, welche Ansätze nicht getestet werden müssen und welche in den nächsten Phasen bis hin zum Test noch weiter verfolgt werden sollten.



Opportunity Solution Tree

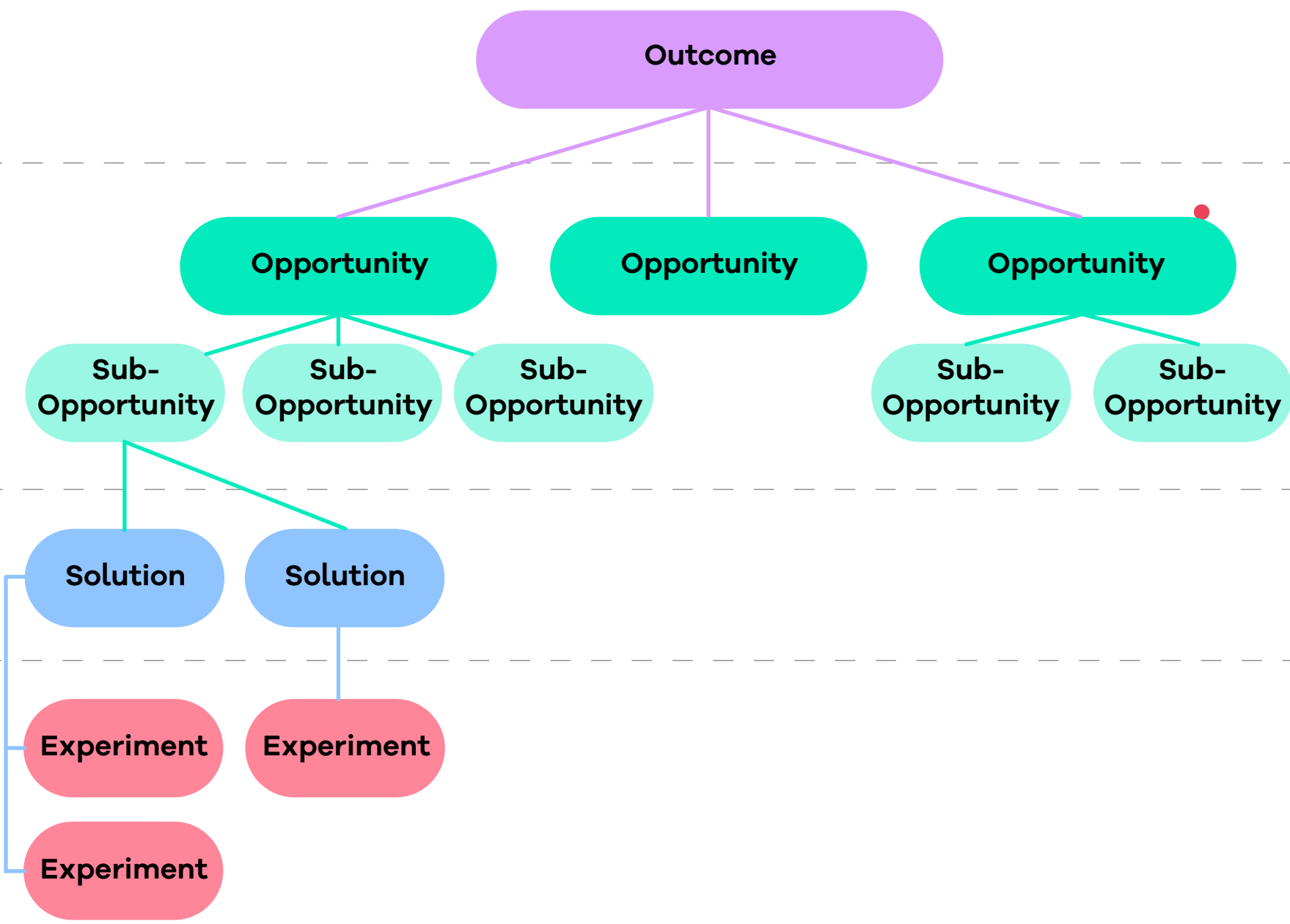
Ein Opportunity Solution Tree schafft Klarheit und Alignment während der Product Discovery, sobald die Perspektiven Technologie, Business und Nutzende diesen im Austausch miteinander ausfüllen.

Das definierte Ziel

Durch Research identifiziert:
Chancen, Probleme, Needs

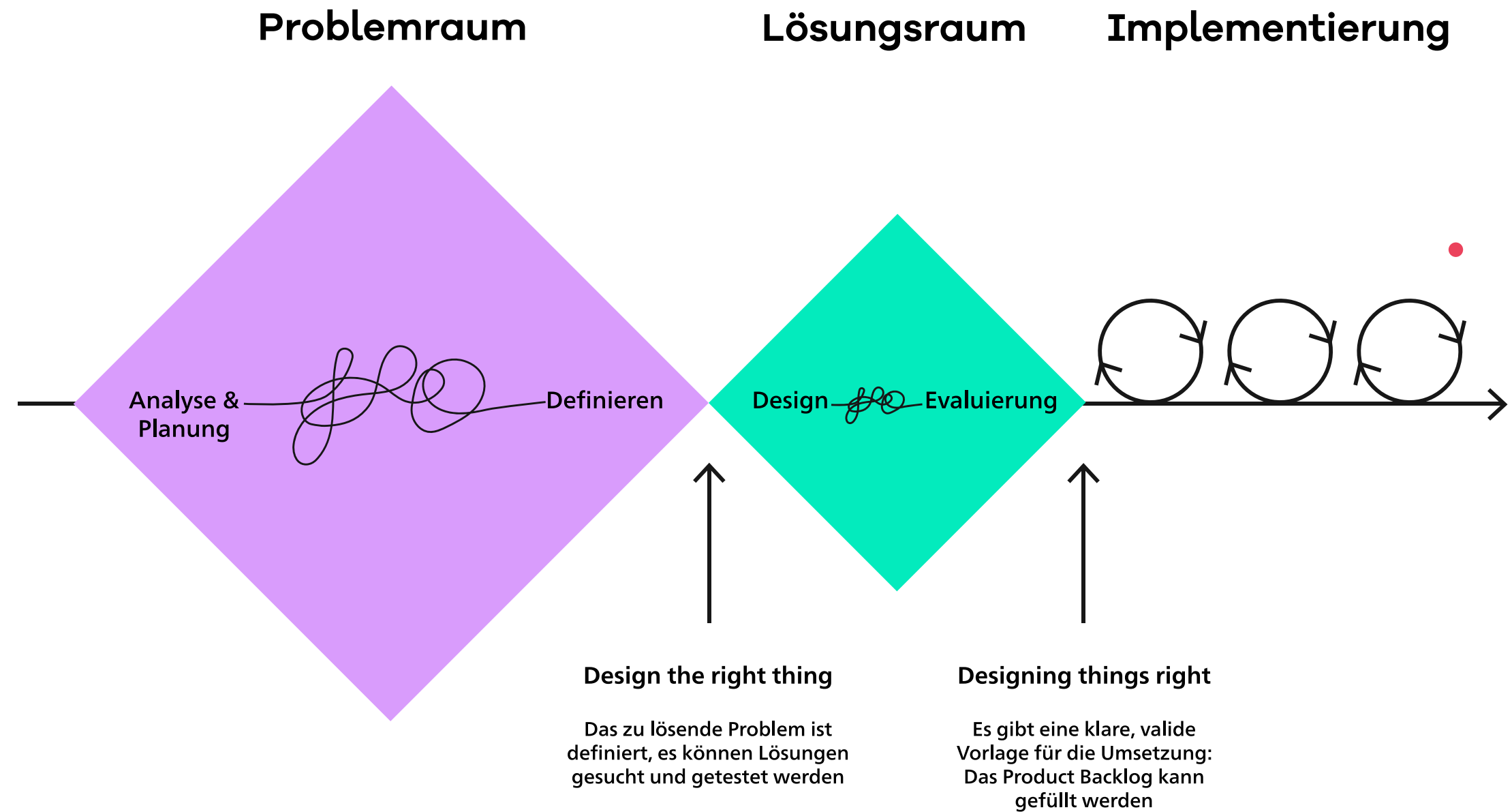
Mögliche Lösungen

Entsprechende Tests



Double Diamond

Double Diamond: Im Problemraum werden zunächst Kontext und Rahmenbedingungen umfassend analysiert, sodass ein tiefes Verständnis für Herausforderungen und Bedürfnisse entsteht. Erst mit einer klaren Formulierung des zu lösenden Problems werden die Aktivitäten im Lösungsraum gestartet. Über Kreativmethoden werden Lösungen entwickelt und iterativ verfeinert, bis eine klare Vorstellung davon existiert, wie genau umgesetzt werden soll („designing things right“).



Ausblick: Prototyp und Test

Wie macht man weiter nach der *Product Discovery*? Die generierten Ideen müssen in greifbare Lösungen überführt werden. Auf Basis der ausgewählten Ideen werden digitale, klickbare Prototypen erstellt. Es muss dafür nicht programmiert werden, mit gängiger Designsoftware können sehr echt wirkende Ergebnisse erzielt werden. Die Prototypen werden in realitätsnahen Nutzungskontexten getestet, um Feedback von den Endnutzer:innen – sowohl Patient:innen als auch medizinischem Personal – zu erhalten. Diese Tests dienen auch dazu, die Nutzerführung und Interaktionen vor der endgültigen Implementierung weiter zu verfeinern. Dieser Schritt des Testens kann, entsprechend dokumentiert, auch normkonform als formative Evaluation nach IEC 62366 durchgeführt werden. Die entstandenen Ideen bzw. Prototypen sind konkret genug, um von technologischer Seite diskutiert zu werden und die Umsetzbarkeit einzuschätzen. Ebenso die Einbindung von Experten für Medizinprodukte-Regulierung kann helfen, kostspielige Anpassungen in späteren Phasen zu vermeiden und den Weg für eine erfolgreiche Markteinführung zu ebnen, wenn aufwändige klinische Studien geplant werden müssen.



Zusammenfassung:

Hohe Qualität & Innovation von Anfang an mit Product Discovery

Erfolgreiche Software, die echten Mehrwert bietet, ist kontinuierliche crossfunktionale, nutzerzentrierte Teamarbeit.

Die interdisziplinäre, nutzerzentrierte Herangehensweise der Product Discovery bietet folgende Vorteile:

Durch die **explorative Discovery** nach dem Motto „fail fast, fail early“ wird sichergestellt, dass strukturiert für tatsächlich vorhandenen Bedarf entwickelt wird.

Risiken und Potentiale proaktiv identifizieren:

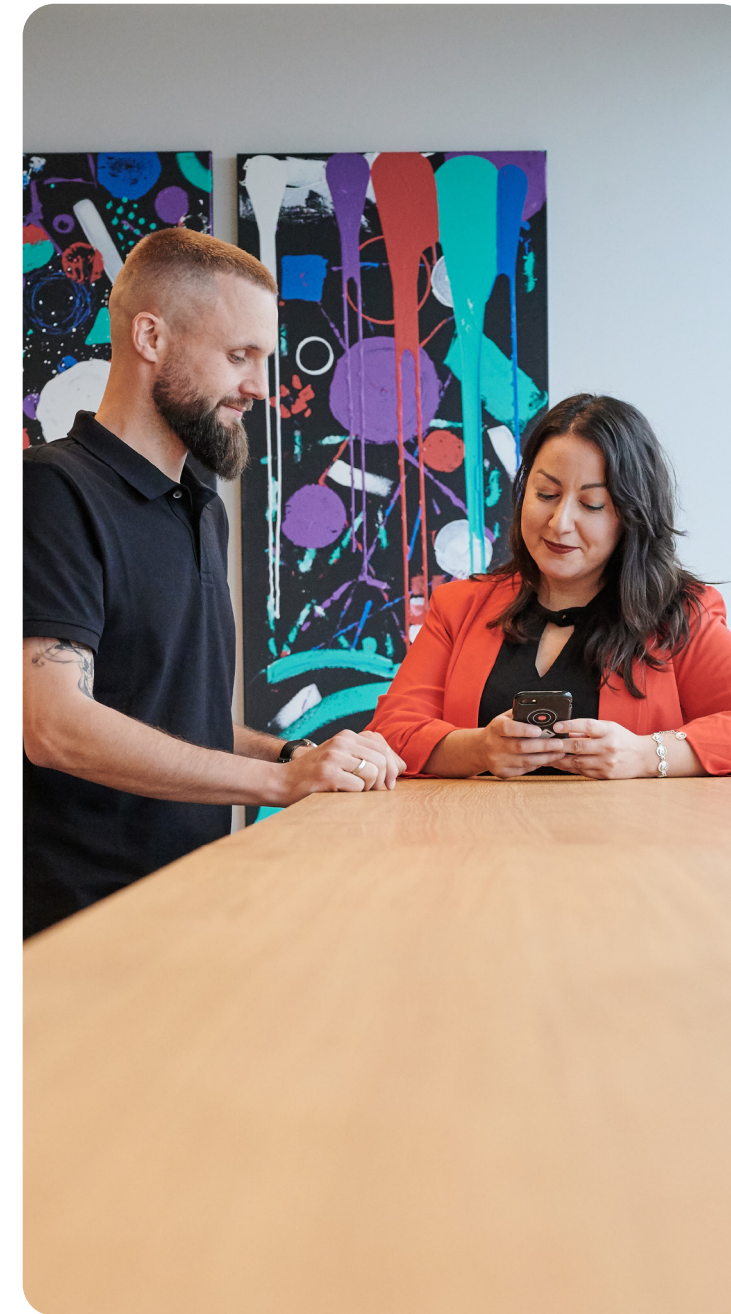
Wer qualitativ forscht, tut dies nicht nur im Sinne der Nutzenden für eine bessere Experience. Man erhält klare Hinweise auf Innovationspotentiale. Risiken werden nicht unnötig eingegangen, Potentiale werden nicht verschenkt.

Die **nutzerzentrierte Arbeitsweise** spart Korrekturaufwände, die entstehen würden, wenn essentielle Anforderungen zu spät erst im laufenden Projekt erst identifiziert würden.

Modernes Teamwork mit all seinen Vorzügen:

Bias wird bewusst entgegengesteuert durch Wertschätzung von verschiedenen Perspektiven und Herangehensweisen.

Interdisziplinärer Wissensaufbau: Aufbrechen von Silos und Sektorengrenzen durch die lösungsorientierte Kommunikation mit relevanten Fachbereichen, Endnutzer:innen, Abteilungen in der eigenen Organisation.



Berücksichtigung der Regulatorik von Beginn an: Da Gebrauchstauglichkeit ein Teil der User-Experience ist, wird beim Arbeiten nach den branchenüblichen Standards (u.a. ISO 92410) speziell für zulassungspflichtige Medizinprodukte bereits das geforderte Usability-Engineering inkl. der formativen Evaluationen nach IEC 62366 abgedeckt. Mehr noch, es wird darüber hinaus eine qualitativ hochwertige User-Experience erreicht. Wer vorausschauend dokumentiert, stellt gleichzeitig auch Rückverfolgbarkeit sicher.

Was unterscheidet Design Thinking, Human Centered Design, Design Sprint?

Der Prozess des „Human Centered Design“ (HCD) hat seine Ursprünge in der Software-Ergonomie. Dieser ist verankert in der internationalen Norm ISO 9241, welche Richtlinien zur Gebrauchstauglichkeit und der Ergonomie interaktiver Systeme beschreibt. In dieser Normenreihe werden u.a. Nutzende und Nutzungskontext in den Mittelpunkt gestellt und definiert klar, welche Kriterien ein interaktives System erfüllen muss, um nutzbar zu sein. Der iterative Prozess nach ISO 9241-210:2019 umfasst die Schritte:

- Verstehen und Festlegen des Nutzungskontextes
- Definition der Nutzungsanforderungen
- Erarbeiten von Designlösungen
- Evaluieren der Designlösungen

Design Thinking ist eine Strategie zur Lösung von Problemen mit Designmethoden. Der Ansatz geht zurück auf die Gründer der amerikanischen Innovationsagentur IDEO. Endgültig populär wurde Design Thinking durch die Unterstützung des SAP Gründers Hasso Plattner und des gleichnamigen Instituts. Im Design Thinking werden die Nutzenden und ihr Problem in den Mittelpunkt gestellt, um vor allem innovative, u.a. disruptive Lösungen zu finden. Es werden keine festen Methoden vorgegeben; das Ziel soll über insgesamt diese 5-6 Phasen erreicht werden:

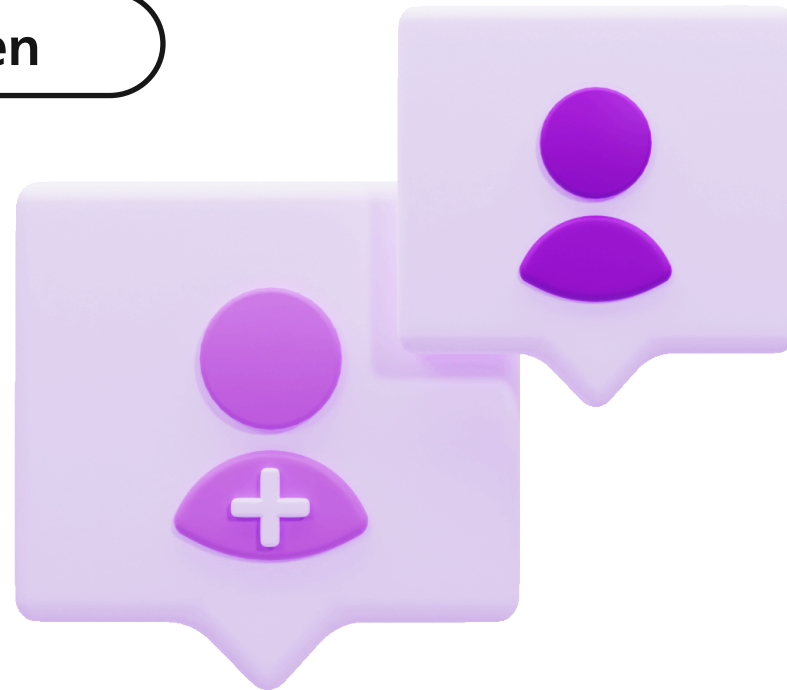
- Verstehen, Beobachten („Empathize“)
- Sichtweise Definieren
- Ideen finden
- Prototyp erstellen
- Testen

„Design Sprint“ ist ein von Google Ventures entwickeltes Methodenset, welches darauf abzielt, eine Lösung innerhalb von 5 Tagen zu erarbeiten und mit echten Nutzenden zu testen. Für größtmögliche Effizienz und schnellstmöglichen Lerneffekt (fail fast) werden alle Phasen des Design Thinking im kleinstmöglichen Zeitraum durchlaufen. Wichtig für den Erfolg sind gute Vorausplanung und Moderation.



Gemeinsam in die digitale Zukunft

Projekt starten



Florian Berner
Business Manager
HealthTech

📍 Hamburg
✉ fberner@slashwhy.de
🌐 [florianberner](#)
☎ 0152 549 190 42



Curie Kure
Senior UX Designer
HealthTech

📍 Hamburg
✉ ckure@slashwhy.de
🌐 [curiekure](#)



This work is licensed under the Creative Commons
CC BY-ND 4.0 DEED Attribution-NoDerivs 4.0
International License. To view a copy of this license,
visit <https://creativecommons.org/licenses/by-nd/4.0/>
or send a letter to Creative Commons,
PO Box 1866, Mountain View, CA 94042, USA.

slashwhy GmbH & Co. KG
Netter Platz 2 | 49090 Osnabrück

+49 541 96104-0
info@slashwhy.de
www.slashwhy.de

Links + Leseempfehlungen

Teresa Torres, "Continuous Discovery Habits"
McKinsey Studie: <https://www.mckinsey.com/capabilities/mckinsey-design/our-insights/the-business-value-of-design>
Warum Produkt-Startups scheitern <https://www.johner-institut.de/blog/gesundheitswesen/medical-startups-healthcare-startups/>
Harvard Business Review: <https://hbr.org/2011/04/why-most-product-launches-fail>
Technology Bias: <https://sloanreview.mit.edu/article/evaluating-new-technology-youre-more-biased-than-you-may-realize/>
UX Design mit IEC 62366 – Ein sicheres HealthTech-Produkt durch Usability Engineering
<https://slashwhy.de/de/blog/usability-engineering-healthtech/>