

Die Konstruktion von Realitäten – Ansätze professioneller Entwicklung zur prototypischen Gestaltung von XR-Anwendungen²


Veronika Krauß ¹

Abstract: Diese Arbeit untersucht Entwicklungspraktiken im Kontext professioneller Softwareentwicklung für Augmented und Virtual Reality (XR) Anwendungen. Die Verbindung einer Design Science Linse mit praxeologischen Ansätzen ermöglicht einen umfassenden Einblick in existierende und aufkommende Entwicklungsprozesse in der aufstrebenden XR-Softwareindustrie. Angesichts des aktuellen Mangels an Design Guidelines, Entwicklungs- und Technologiestandards sowie unterstützenden Entwicklungswerkzeugen bietet die Arbeit einen ganzheitlichen Überblick und entwickelt mögliche Lösungsansätze sowie Designvorschläge zur (softwarebasierten) Unterstützung professioneller XR-Entwickler in interdisziplinären Teams.

1 Einleitung

In den letzten Jahren gewannen Augmented Reality und Virtual Reality (XR) zunehmend an Popularität für private und industrielle Anwendungen. Dies wurde insbesondere durch technische Innovation und erschwinglich werdenden Technologien ausgelöst. Damit einhergehend wuchs auch der Bedarf an fachkundig entwickelter Software. Es offenbarte sich jedoch ein Mangel an Entwicklungs- und Designwerkzeugen in der professionellen XR-Anwendungsentwicklung. Nicht zuletzt ist dies begründet in der beständigen Entwicklung neuer XR-Endgeräte und einen Mangel an Standards, sondern auch weil bekannte Vorgehensweisen aus der klassischen Softwareentwicklung für den 2D Bereich konzipiert waren. Beim Einsatz für die Entwicklung von dreidimensionalen Anwendungen stießen diese schnell an ihre Grenzen [As20]. Diese sogenannte *Tool Gap* [NS18] existierte trotz dekadentlanger Forschung.

Mit der voranschreitenden Adaption von XR Technologie fokussierte sich die Interaktionsdesign- und Toolforschung der Human-Computer-Interaction (HCI) Community zunehmend auf die Demokratisierung von XR Technologie – also mit dem Abbau von Hindernissen, die Menschen mit weniger technischem Wissen und Programmierkönnen von der Entwicklung für XR ausschließen [As20; NS18]. Dennoch war weder über die Hürden professioneller XR-Anwendungsentwickler, noch über deren Umgang mit daraus entstehenden Herausforderungen im Entwicklungsprozess in der Praxis viel bekannt.

1 Hochschule Bonn-Rhein-Sieg, Institut für Verbraucherinformatik, Grantham-Allee 20, 53757 Sankt Augustin, Deutschland, veronika.krauss@h-brs.de,  <https://orcid.org/0000-0002-4936-9787>

2 Englischer Titel der Dissertation: "Constructing Realities – Professionals' Approaches to Prototyping XR Experiences"

Daraus entsteht ein mangelndes Verständnis über die Implementierung der jeweiligen Designprozesse, den Einsatz unterstützender Werkzeuge und das Einbinden von Nutzenden oder anderen Stakeholdern in die Anwendungsentwicklung sowie deren Einfluss auf die final zu gestaltende Anwendung. Letztendlich kann dies zu einer wachsenden *Theorie-Praxis-Lücke* [GSW11] in der XR-Interaktionsdesign- und Toolforschung führen, da aus der Forschung hervorgehende Designrichtlinien, Methoden und Werkzeuge die Bedürfnisse und Ansätze von XR Interaktionsdesigner*innen und Fachleuten der XR-Softwareentwicklung nicht ausreichend durchdringen und berücksichtigen können, obwohl diese potentiell gesellschaftsbeeinflussende Technologien erstellen.

Von dieser Situation ausgehend untersucht diese Arbeit Praktiken und damit einhergehende Herausforderungen und Behelfslösungen professioneller XR-Anwendungsentwickelnden mit industriellem Hintergrund basierend auf vier empirischen Studiens. Der Fokus liegt auf der prototypischen Anwendungskonzipierung und Interaktionsdesignentwicklung in moderner Softwareentwicklung, welche häufig auch als Prototyping bezeichnet wird. Dabei vereinen wir die Blickwinkel einer Design Science/Research into Design Perspektive mit der Social Practice Theory (Praxistheorie) nach Shove [SPW12], um ein besseres Verständnis für die Situation professioneller Entwickler*innen zu erhalten und daraus Empfehlungen für Entwicklungswerkzeuge abzuleiten.

Beiträge. Diese Arbeit fasst vier empirische Studien und deren Ergebnisse aus [Kr23] wie folgt zusammen:

- Wir analysieren Entwicklungspraktiken, Herausforderungen und Behelfslösungen von XR-Anwendungsentwickler*innen im professionellen Kontext und identifizieren drei in der Praxis auftretende Kernherausforderungen [Kr21a].
- Wir betrachten die Praktik des XR-Anwendungsprototyping und beschreiben die Verwendung von Prototypen im professionellen Kontext [Kr22]. Hierbei fokussieren wir uns auch auf häufig verwendete, universelle Elemente der Anwendungsentwicklung, die Rückschlüsse auf Unterstützungsbedarfe durch entsprechende Entwicklungswerkzeuge zulassen.
- Durch die Perspektive der Praxistheorie setzen wir unsere Erkenntnisse in einen übergreifenden Kontext, um konkrete Herausforderungen von XR als in der Entstehung befindliche Technologie zu verstehen und herauszufinden, wie Entwickler in ihrer Arbeit gewinnbringend unterstützt werden können.

2 Herausforderungen professioneller XR Interaktionsdesignpraktik

Als noch in der Entstehung befindliches Feld ist XR von Innovation getrieben und hat erst wenige Standards und allgemeingültige Konventionen hervorgebracht [As20; Kr21a]. Dies bedeutet nicht nur, dass zum Entstehungszeitpunkt dieser Arbeit ein Großteil der professionellen XR-Entwickler keine dedizierte Ausbildung in dieser Technologie durchlaufen haben,

sondern auch dass Hardware-Firmen proprietäre Lösungen anbieten, die weder im Bezug auf den Funktionsumfang und der Bedienkonzepte untereinander kompatibel sind. Folglich können wir von XR als *instabile Technologie* sprechen [MHP00] – also eine Technologie, die durch proprietäre Hardwarelösungen aus Sicht der Softwareentwickelnden fragmentiert ist und wegen der daraus entstehenden Inkompatibilität von Endgeräten, Entwicklungswerkzeugen und Softwaretools regelmäßig die Erstellung von Einzellösungen erfordert. Das beginnt beim sogenannten *Tool Chaining* [NS18] – der Auswahl geeigneter Entwicklungstools und deren Kompatibilität untereinander – und endet beim finalen Produkt, welches exakt auf einen Anwendungskontext und die zu verwendende Hardware zugeschnitten ist.

Diese Situation stellt die Auftraggebenden und Endanwendenden vor Herausforderungen: In der Vergangenheit gelernte Systeminteraktionen und zu erwartende Ausgaben können oft nicht zwischen verschiedenen Anwendungen transferiert werden, wodurch XR-Anwendungen kaum intuitiv bedienbar sind. Häufig muss die Bedienung einer unbekanntenen Anwendung von Grund auf neu erlernt werden, da universelle Interaktionsmethoden und Paradigmen, wie wir sie zum Beispiel in Form von Buttons und Links aus dem Web kennen, in XR noch nicht in dieser Form etabliert sind. Zusätzlich bieten Hardwarefirmen unterschiedliche Endgeräte und Kontrollmechanismen (z.B. Eyetracking, Belegung der Knöpfe an einem Hardwarecontroller, Gestensteuerung) an [As20; Kr21a; Kr22]. Zudem ist aus Sicht der Entwickelnden problematisch, dass für neue Aufträge wiederholt neue Prozesse und Toolchains ergründet werden müssen. Anders als in etablierten Teilbereichen der Softwaresystemgestaltung, z.B. der Webentwicklung, fehlt es XR aber noch an der fachlichen Spezialisierung und damit einhergehend einer Ausformung von spezialisierten Rollen [Kr21a], weshalb einzelne Mitglieder eines XR-Entwicklungsteams bezüglich ihrer Kompetenzen breit aufgestellt sein müssen. Trotz der dadurch entstehenden „Alleskönner“ wird professionell entwickelte XR-Software im Regelfall in interdisziplinären Teams entwickelt, wodurch eine Vielzahl von Schwierigkeiten entsteht. Zur Überwindung dieser Herausforderungen werden Prototypen eingesetzt, die mit Hilfe von Designwerkzeugen und Methoden erstellt werden. Prototyping selbst ist jedoch aus unterschiedlichen Gründen im Kontext der XR-Entwicklung ein schwieriges Unterfangen. Basierend auf unseren empirischen Studien [Kr21a; Kr21b; Kr22; KU20] fassen wir die drei Kernpunkte *Fehleinschätzung des Mediums*, *heterogene Konzepte und Begriffe* und *fehlende Werkzeugunterstützung* zusammen [Kr21a] und betrachten diese im Folgenden genauer.

Fehleinschätzung des Mediums Solche Fehleinschätzungen äußern sich durch die Überschätzung der Hardware- und Softwarefunktionalitäten und der Robustheit eines spezifischen Endgeräts, welche oft durch eine mangelnde Erfahrung der Entwickelnden von XR in der Praxis oder mit speziell einem Endgerät herrührt. Gerade unerfahrene Entwickler*innen berufen sich häufig auf Marketingmaterialien entsprechender Hardwarefirmen, welche die Fähigkeiten ihres Produkts nicht immer realitätsgetreu abbilden und zum Beispiel das Trackingverhalten eines Endgeräts im tatsächlichen Einsatz weniger robust, die Gesten- und Spracherkennung schlechter oder das Field-of-View kleiner als gezeigt ist. Zudem

projizieren aus der klassischen 2D Softwareentwicklung stammende Entwickler*innen inkompatible Erfahrungen auf die Funktionsweise von XR Technologie. So lassen sie zum Beispiel außer Acht, dass eine XR-Anwendungen anders als eine klassische Webapplikation nicht auf den Bereich eines Bildschirm begrenzt ist und Interaktionselemente auch hinter einer physischen Wand verschwinden könnten.

Interdisziplinäre Teams behelfen sich hier zum Beispiel durch eine engere Zusammenarbeit zwischen technischen und weniger technischen Entwickelnden, sowie mit frühen Machbarkeitsevaluationen günstiger Designprototypen. Hierbei produzieren sie Designartefakte wie Moodboards oder einfache, funktionsfähige Softwareprototypen zur Veranschaulichung der Limitationen spezifischer Hardware oder Interaktionsdesigns.

Heterogene Konzepte und Begriffe Die Heterogenität von Konzepten sowie der Mangel einheitlicher Begriffe und Bezeichnungen in interdisziplinären Teams erweist sich gerade in Bezug auf möglichst präzise Beschreibungen von Systemverhalten und Designideen als kritisch. Dieses Problem tritt besonders in neu zusammengestellten Teams oder Teams mit unerfahrenen Mitgliedern auf und führt durch ungeeignete Werkzeuge, Designmethoden und Begrifflichkeiten, welche oftmals aus der klassischen 2D Softwaregestaltung übernommen werden, zur Erstellung unzufriedenstellender Designartefakte.

Zur Überwindung dieser Problematik veranstalten Teams zum Beispiel gemeinsame Programmiersessions, in welchen technisch versierte Teammitglieder basierend auf Livefeedback der Interaktionsgestalter*innen das Systemverhalten direkt anpassen. Auch über die Verwendung sogenannter *vergänglicher Prototypen* (Ephemeral Prototypes) [Kr21a; Kr22] wurde berichtet, also von Prototypen mit performantem Charakter und keiner Lebensdauer. Diese werden sich gegenseitig durch Gesten und Bewegung vorgespielt oder anhand vergangener und geteilter Erinnerungen oder verwendeter Referenzsysteme beschrieben. Erfahrenere Teams entwickeln zusätzlich eine eigene Designsprache, die zum Beispiel durch Farbcodierung und Perspektivenwechsel in Skizzen Systemverhalten beschreiben kann. Zusätzlich genannte Methoden umfassten auch die Verwendung von Videomaterialien, Referenzsoftware (z.B. Spiele) und Gray Boxing – eine Methode aus der Spieleentwicklung, bei welcher räumliche Aspekte durch die Platzierung grauer Boxen in 3D Software schnell konzipiert werden können.

Fehlende Werkzeugunterstützung Fehlende Werkzeugunterstützung wurde oft genannt, wenn räumliche und interaktive Aspekte einer XR-Anwendung erstellt oder getestet werden sollten. Existierende Entwicklungs- und Designwerkzeuge, die ein zufriedenstellendes Ergebnis liefern, setzen in der Regel gute Programmierkenntnisse voraus. Fehlen diese, sind damit erstellte Prototypen für programmatisch weniger versierte Teammitglieder nicht zugänglich und veränderbar – beides sind jedoch essentielle Voraussetzungen für unabhängiges Arbeiten in iterativen und interdisziplinären Teams mit Aufgabenverteilung. Deshalb forderten Teams Möglichkeiten zur Prototypenerstellung ohne Programmieraufwand, sowie eine bessere Einbindung solcher Tools in ihre Arbeitspraxis. Existierende Werkzeuge,

welche keine Programmierkenntnisse erfordern, eignen sich häufig nicht zur Erstellung oder Extraktion konkreter Designspezifikationen aus einem erstellten Prototyp [Kr21a]. Resultierend aus dieser Problematik erstellen technisch weniger versierte Entwickelnde teilweise sehr aufwändige, physische Prototypen aus Papier oder Karton oder verwenden annotierte Flussdiagramme, Skizzen und Wireframes, welche von programmatisch versierten Teammitgliedern in softwarebasierte und interaktive Prototypen übersetzt werden.

Die oben beschriebenen Beobachtungen geben Aufschluss über die Problematiken, vor welchen interdisziplinäre XR-Entwicklerteams stehen. Diese Perspektive ist jedoch sehr auf individuelle Aspekte fokussiert. Um ein ganzheitliches Verständnis für die aktuelle Situation für XR-Entwickelnde mit industriellem Hintergrund zu erlangen, bedienen wir uns im Abschnitt 3 der Praxistheorie nach Shove [SPW12].

3 XR Interaktionsdesignpraktiken im Kontext einer entstehenden Technologie

Bisher haben wir Herausforderungen interdisziplinärer Teams sehr fokussiert anhand konkreter Probleme und entsprechender Übergangslösungen betrachtet. Diese Perspektive stellt jedoch im Rahmen von XR eine zeitlich limitiert gültige Momentaufnahme dar – als schnell veränderliches und durch beständige Innovation vorangetriebenes Medium mit wenigen Standards und Konventionen hatten wir XR im vorherigen Abschnitt bereits als instabiles Medium [MHP00] identifiziert. Damit wir ein ganzheitliches Verständnis der beschriebenen Problematiken erhalten und potenzielle Lösungsansätze in Form von Werkzeugimplikationen entwickeln können, benötigen wir folglich mehr Kontext. Deshalb bedienen wir uns der Praxistheorie, genauer gesagt der Social Practice Theory nach Shove et al. [SPW12], welche sich besonders zur Beschreibung beständiger und dynamischer Aspekte von Routinen und Gewohnheiten – sogenannte Praktiken – eignet. Shove et al. unterteilen eine Praktik in die drei Teilaspekte *Materialien*, *Kompetenzen* und *Bedeutungen* (siehe Abbildung 1).

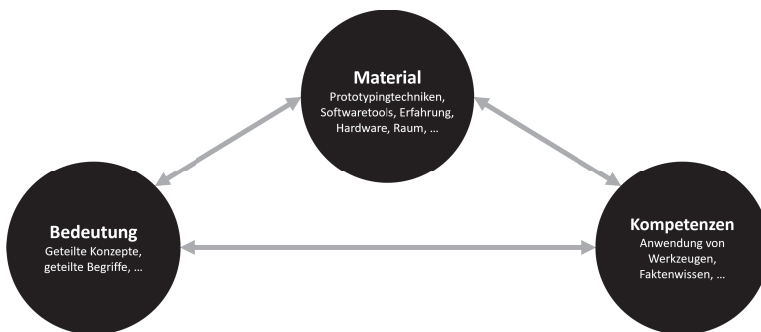


Abb. 1: Sich gegenseitig beeinflussende und formende Elemente der Praktik „XR-Anwendungsentwicklung“ nach dem Modell von Shove et al. [SPW12]

Gemäß der Praxistheorie nach Shove et al. [SPW12] beeinflussen und formen sich die drei Elemente Material, Kompetenzen und Bedeutungen gegenseitig. Im Folgenden betrachten wir zunächst alle drei Elemente der XR-Entwicklungspraktik einzeln, bevor wir sie als komplexes Zusammenspiel diskutieren.

3.1 Materialien

Materialien sind alle benötigten Artefakte, die zur Durchführung einer Praktik benötigt werden. Die in der XR-Anwendungsentwicklung verwendeten Materialien sind zum einen XR-spezifisch und dadurch als *neu* zu bezeichnen. Neben spezialisierter Hardware wie XR Brillen und dazugehörige Controller zählen auch neue Softwarewerkzeuge wie Frameworks, Game Engines und Prototypingsoftware (z.B. ShapesXR) dazu. Als *alte* Materialien werten wir Dinge, die aus der klassischen Softwareentwicklung und verwandten Bereichen (z.B. Game Design, Architektur) bekannt sind, z.B. Manifestierungen von Prototypen (z.B. Papierbasierte, Softwarebasierte) und dazugehörige Prototypingmethoden (z.B. Paperprototyping, Gray Boxing). Da XR eine dreidimensionale Technologie ist, werten wir auch den physische und digitalen Raum als Material. Beide können in den *Prototyping Space* – also den Raum, in welchem eine XR-Anwendung entwickelt wird – und den *Target Space* – den Raum, in welchem eine XR-Anwendung ausgeführt werden soll oder wird – unterteilt werden.

Unter dem Aspekt der heterogenen Ausbildungen heutiger XR-Entwickler*innen stellen wir fest, dass *neue* Materialien auch solche sein können, die einem entwickelnden Individuum aufgrund einer inkompatiblen Ausbildung bisher nicht bekannt waren. Das kann für den ausgebildeten Webentwickler zum Beispiel eine in der XR-Entwicklung häufig eingesetzte 3D Modellierungssoftware sein, die für die Spieleentwicklerin jedoch längst Bestandteil ihrer täglichen Arbeit war. Zudem halten wir nochmals fest, dass sich Materialien im Kontext von XR durch Innovation und Verbesserung beständig im Wandel befinden, neu auf den Markt gebracht oder obsolet werden. Dabei mangelt es auch an Standards, die die Kompatibilität verfügbarer Materialien untereinander gewährleisten könnten.

3.2 Kompetenzen

Kompetenzen beschreiben theoretisches (explizites) und praktisches (implizites) Wissen und Können wie die Kenntnis existierender Designwerkzeuge und die Fähigkeit, diese einzusetzen, aber auch Designwissen einschließlich praktischer Erfahrungen sowie Programmierkenntnisse.

Wie wir im Abschnitt 2 bereits festgestellt haben, mangelt es gerade unerfahrenen XR-Entwickler*innen an Erfahrungen bezüglich der Fähigkeiten und Limitationen von XR als Technologie, sowie den notwendigen Fertigkeiten um vornehmlich programmatische Entwicklungswerkzeuge zu bedienen. Während einige Teams zur Bewältigung dieser Herausforderung auf gegenseitiges Lernen und Lehren von Entwicklungswerkzeugen setzen, binden

andere entsprechend erfahrene Teammitglieder früh in Konzept- und Prototypingphasen ein. Andere Teams lagern fehlende Kompetenzen, gerade bezüglich der Programmierung von Interaktionen mit entsprechenden Werkzeugen, auf fähige Teammitglieder aus. Das bedeutet im Umkehrschluss, dass technisch weniger versierte Entwickler*innen den Zugriff auf entstandene Artefakte verlieren und bei gewünschten Änderungen auf die Verfügbarkeit der entsprechenden Kolleg*innen angewiesen sind. Wir bezeichnen das als *Kompetenz-auslagerung*, welche auch mit einem Autonomieverlust betroffener Entwickelnder im Gestaltungsprozess einhergeht.

Eine weitere Herausforderung beim Aufbau von XR-Entwicklungscompetenzen liegt in der nicht immer zu gewährleistenden Kompatibilität zwischen unterschiedlichen Materialien. Das heißt, dass erworbene Kompetenzen spezifisch für ein Material gültig sind – zum Beispiel bedingt durch limitierte, softwareseitige Unterstützung eines bestimmten XR Brillenmodells in einer häufig zur Anwendungsentwicklung verwendeten Game Engine – und beim Wechsel auf ein anderes Material nicht transferierbar sind. Entsprechend müssen Materialien häufig neu erlernt werden, was durch eine unübersichtliche Situation am Markt [NS18] und fehlende Lernquellen [As20] zusätzlich erschwert wird.

3.3 Bedeutungen

Bedeutungen können gesellschaftlichen oder individuellen Ursprungs sein und belegen Materialien und Kompetenzen auch mit entsprechenden Werten, Emotionen und Motivationen. Im Bereich der XR-Entwicklung und des Interaktionsdesigns zählen wir geteilte Konzepte und Begrifflichkeiten genauso wie manifestierte Designintentionen und Ideen in Form von Prototypen als *Bedeutungsträger* zu dieser Gruppe.

Wir haben bereits im Abschnitt 2 einen Mangel an neuen und etablierten Bedeutungen in Form von nicht vorhandenen homogenen Bezeichnungen, Konzepten oder einer generellen Sprache festgestellt. Während einige Teams hier eine eigene Designsprache zur Kommunikation entwickelten, verwendeten andere gemeinsame Erinnerungen und Erfahrungen als geteiltes Konzept, auf welchem aufbauend Ideen diskutiert und vorgespielt werden (*Ephemeral Prototyping*). Generell spielen Prototypen hier als Bedeutungsträger eine zentrale Rolle um den Mangel an existierenden und etablierten Bedeutungen auszugleichen. Als anschauliches aber teilweise vergängliches Designartefakt vermitteln sie Ideen und Konzepte innerhalb des Entwicklungsteams, aber auch während der Abstimmung mit Kunden und Nutzenden.

3.4 Spuren von Stabilität in XR-Entwicklungspraktiken

Bei all den Dynamiken und der schnell vorangetriebenen Innovation stellt sich uns die Frage, ob es nicht doch gewisse Arten von Stabilität gibt, auf Basis welcher wir zur besseren Unterstützung von XR-Entwickelnden aufbauen könnten.

Analysieren wir die Essenz der Arbeit eines XR-Entwicklungsteams anhand der entstehenden Artefakte stellen wir fest, dass Prototypen tatsächlich Elemente aufweisen, die als *stabil* bezeichnet werden können. Diese sind neben globaleren Aspekten wie der Verwendung aus der Softwareentwicklung bekannter Prozessmodelle (Mensch-zentriertes Design, Scrum [Kr21a; KU20]) oder Prototypingtechniken [Kr22] auch die Problemstellungen, mit welcher sich entwickelte Prototypen befassen und Eigenschaften, die diese aufweisen.

Auf Basis der durchgeführten Studien in [Kr23] können wir zehn solcher Elemente identifizieren: Räumlichkeit (*Spatiality*) als die räumliche Beziehung (Distanz, Skalierung, Rotation) zwischen realen und virtuellen Objekten, Körperlichkeit (*Physicality*) zur Betrachtung physischer Eigenschaften, Weltaufbau (*World-building*) als Hintergrund für Erzählstrukturen, Erzählstruktur (*Flow – Story*) als die in einer Anwendung erzählte Geschichte, Informationsstruktur (*Flow – Hierarchy*) als logischer und hierarchischer Aufbau einer Anwendung, Bedienung (*Control*) als Interaktionstechniken oder Controllerdesign, Fortbewegung (*Locomotion*) als Navigations- und Fortbewegungsaspekt von Nutzenden in der Anwendung, Interaktivität (*Interactivity*) zur Betrachtung reaktiver und animierter Aspekte, Kinematographie (*Cinematography*) als Kameraführung und -bewegung, und Inhalte (*Content*) als digitale Aspekte wie Sound, Texturen oder (virtuelle) Objekte [Kr22].

Diese Elemente werden in unterschiedlicher Ausprägung anwendungsübergreifend berücksichtigt und bieten daher einen geeigneten Ansatz zur dedizierten Unterstützung von professionellen XR-Entwickelnden. Da XR-Software im Regelfall iterativ entwickelt wird, ist sowohl die Ausprägung dieser Elemente, als auch deren dargestellter Reifegrad (*Fidelity*) im Vergleich zum angestrebten Zustand des finalen Produkts variabel. Das heißt, dass prototypisch entwickelte Aspekte im Laufe eines Projekts mehrere *Transitionen* in Form von unterschiedlichen Darstellungen (Manifestationen) und Reifegraden durchlaufen um so ein möglichst ganzheitliches Bild einer idealen Lösung für ein Designproblem zu schaffen.

4 Implikationen für XR Designwerkzeuge

Aus den vorherigen Abschnitten wissen wir, dass Materialien, Kompetenzen und Bedeutungen voneinander abhängig sind und sich gegenseitig beeinflussen. Das bedeutet, dass die erfolgreiche Verwendung neuer oder unbekannter Materialien einen Kompetenzaufbau und die Etablierung dafür geeigneter Bedeutungen erfordern. Ebenso verlangt der Aufbau von Kompetenzen die Erweiterung und Adaption von Bedeutungen und reguliert die Verwendung bestimmter Materialien. Durch den beständigen Wandel von Bedeutungen kann die Verwendung einiger Materialien eingeschränkt oder gar aufgegeben werden, was wiederum den Aufbau (oder Verlust) entsprechender Kompetenzen nach sich zieht.

Durch den großen Innovationsdruck ändern sich verfügbare Materialien, die Anwendbarkeit angeeigneter Kompetenzen und die Bekanntheit von Bedeutungen beständig – deshalb bezeichneten wir XR im Abschnitt 2 als instabile Technologie [MHP00]. Eine gewisse

Form von Stabilität ist aber erforderlich, um geeignete Werkzeuge zur Designunterstützung anzubieten. Das kann nicht nur in Form von Software erfolgen, sondern schließt auch Design Guidelines, Prozesse, Methoden und Konventionen mit ein. Diese Arbeit stellt zehn Elemente von XR-Anwendungssoftware vor, die in Form wiederkehrender Aspekte und Problemstellungen stabile Eigenschaften aufweisen und als Grundlage für Unterstützungswerkzeuge herangezogen werden können.

Wenn wir mit unserer Frage nach besseren Werkzeugen bei dem Konzept **Materialien** ansetzen, identifizieren wir drei Hauptaspekte, die künftige Werkzeuge zur Unterstützung von XR-Entwickelnden aufweisen sollten: Kompatibilität und Interoperabilität (zwischen anderen Materialien, aber auch bezüglich der transitiven Eigenschaften von Prototypen), Reduktion von Design- und Produktionskomplexität (z.B. durch die Verfügbarmachung schnell einzusetzender Animationen oder Interaktionen) und Flexibilität von Werkzeugen bezüglich der Anpassbarkeit an individuelle Entwicklungspraktiken und Zielen (*Taylorability*).

Bezüglich der Unterstützung von **Kompetenzen** können wir gerade die Kompetenzauslagerung und den damit einhergehenden Autonomieverlust als Ansatz zur Verbesserung von Werkzeugen identifizieren. In diesem Kontext ist es mit der heutigen Entwicklung KI-gestützter Werkzeuge denkbar, dass künftige Tools die Rolle des programmatisch versierten Experten übernehmen und anstatt eines bloßen Werkzeugs als kompetenter Sparringspartner im Designprozess auftreten können. Ein weiterer Ansatz ist die Verfügbarmachung geeigneter Design Guidelines und Good Practices zur Kompetenzvermittlung – hier gilt es aber zu Bedenken, dass solche Entwicklungswerkzeuge in der professionellen Softwareentwicklung anderen Anforderungen folgen sollten, als es Design Guidelines aus der akademischen Forschungscommunity häufig tun [Kr21b].

Zu guter Letzt betrachten wir Möglichkeiten zur Unterstützung von **Bedeutung**. Diese Arbeit hat gezeigt, dass sowohl neue als auch alte Bedeutungen in XR-Entwicklungsteams nicht notwendiger Weise auf gemeinsamen Grundlagen aufbauen, da selbst in der Softwareentwicklung etablierte Konzepte wie *Fidelity* unterschiedlichen mentalen Modellen folgen. Ausgehend von dieser Basis ist die Entwicklung einer gemeinsamen Sprache nicht nur für XR-Entwicklungsteams, sondern potentiell für alle Entwicklungsteams entstehender (und bestehender) Technologien eine große Herausforderung. Diese erfordert, dass künftige Forschung im Bereich Interaktionsdesign und Anwendungsentwicklung existierende Konzepte und deren Lehre kritisch hinterfragt und Grundlagentheorien gemäß den Eigenschaften moderner Softwaresysteme aktualisiert.

Literaturverzeichnis

- [As20] Ashtari, N.; Bunt, A.; McGrenere, J.; Nebeling, M.; Chilana, P. K.: Creating Augmented and Virtual Reality Applications: Current Practices, Challenges, and Opportunities. In: Proceedings of the 2020 CHI Conference on Human

- Factors in Computing Systems. CHI '20, Association for Computing Machinery, New York, NY, USA, S. 1–13, 2020.
- [GSW11] Goodman, E.; Stolterman, E.; Wakkary, R.: Understanding Interaction Design Practices. In: Proceedings of the International Conference on Human Factors in Computing Systems. Vancouver, Canada, S. 1061–1070, 2011.
- [Kr21a] Krauß, V.; Boden, A.; Oppermann, L.; Reiners, R.: Current Practices, Challenges, and Design Implications for Collaborative AR/VR Application Development. In: CHI '21: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. ACM, S. 1–15, 2021.
- [Kr21b] Krauß, V.; Jasche, F.; Saßmannshausen, S. M.; Ludwig, T.; Boden, A.: Research and Practice Recommendations for Mixed Reality Design – Different Perspectives from the Community. In: 27th ACM Symposium on Virtual Reality Software and Technology (VRST '21). ACM, Osaka, Japan, 2021.
- [Kr22] Krauß, V.; Nebeling, M.; Jasche, F.; Boden, A.: Elements of XR Prototyping: Characterizing the Role and Use of Prototypes in Augmented and Virtual Reality Design. In: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems. CHI '22, ACM, New York, NY, USA, 2022.
- [Kr23] Krauß, V.: Constructing Realities – Professionals' Approaches to Prototyping XR Experiences, Diss., Universität Siegen, 2023.
- [KU20] Krauß, V.; Uzun, Y.: Supporting Medical Auxiliary Work: The Central Sterile Services Department as a Challenging Environment for Augmented Reality Applications. In: 2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). S. 665–671, 2020.
- [MHP00] Myers, B.; Hudson, S. E.; Pausch, R.: Past, Present, and Future of User Interface Software Tools. ACM Trans. Comput.-Hum. Interact. 7/1, S. 3–28, 2000.
- [NS18] Nebeling, M.; Speicher, M.: The Trouble with Augmented Reality/Virtual Reality Authoring Tools. In: 2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct). S. 333–337, 2018.
- [SPW12] Shove, E.; Pantzar, M.; Watson, M.: The dynamics of social practice: Everyday life and how it changes. Sage, 2012.



Veronika Krauß ist eine Nachwuchswissenschaftlerin am Institut für Verbraucherinformatik der Universität Siegen und der Hochschule für angewandte Wissenschaften Bonn-Rhein-Sieg. Nach ihrem Studium erforschte sie am Fraunhofer Institut für angewandte Informationstechnik und am Institut für Verbraucherinformatik menschenzentrierte Systemgestaltung, Interaktionsdesign, Prototyping und die verantwortungsbewusste Entwicklung räumlicher Systeme, sowie deren Auswirkungen auf Individuen und die Gesellschaft.