

Current Practices, Challenges, and Design Implications for Collaborative AR/VR Application Development

Veronika Krauß*†

Alexander Boden*‡

Leif Oppermann*

René Reiners*

* Fraunhofer Institute for Applied Information Technology FIT
Sankt Augustin, Germany
[leif.oppermann, rene.reiners]@fit.fraunhofer.de

† Verbraucherinformatik Research Group; University of Siegen
Siegen, Germany
veronika.krauss@uni-siegen.de

‡ Institut für Verbraucherinformatik, Bonn-Rhein Sieg University of Applied Science
Sankt Augustin, Germany
alexander.boden@h-brs.de

ABSTRACT

Augmented/Virtual Reality (AR/VR) is still a fragmented space to design for due to the rapidly evolving hardware, the interdisciplinarity of teams, and a lack of standards and best practices. We interviewed 26 professional AR/VR designers and developers to shed light on their tasks, approaches, tools, and challenges. Based on their work and the artifacts they generated, we found that AR/VR application creators fulfill four roles: concept developers, interaction designers, content authors, and technical developers. One person often incorporates multiple roles and faces a variety of challenges during the design process from the initial contextual analysis to the deployment. From analysis of their tool sets, methods, and artifacts, we describe critical key challenges. Finally, we discuss the importance of prototyping for the communication in AR/VR development teams and highlight design implications for future tools to create a more usable AR/VR tool chain.

CCS CONCEPTS

• **Software and its engineering** → **Collaboration in software development**; • **Human-centered computing** → *Human computer interaction (HCI)*; Interaction techniques.

KEYWORDS

Augmented Reality, AR/VR, MR, XR, authoring tools, AR development, AR design, practitioners

ACM Reference Format:

Veronika Krauß, Alexander Boden, Leif Oppermann, and René Reiners. 2021. Current Practices, Challenges, and Design Implications for Collaborative AR/VR Application Development. In *CHI Conference on Human Factors in Computing Systems (CHI '21)*, May 8–13, 2021, Yokohama, Japan. ACM, New York, NY, USA, 15 pages. <https://doi.org/10.1145/3411764.3445335>

© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *CHI '21*, May 8–13, 2021, Yokohama, Japan
<https://doi.org/10.1145/3411764.3445335>

1 INTRODUCTION

When Gartner removed AR and VR from its Hype Cycle for Emerging Technologies in 2019 [76], they indicated that the technology as such has reached a mature state. In fact, the market is evolving rapidly around consumer adoption as well as software and hardware [64]. In contrast, literature in science and practice understands AR/VR as an emerging technology with a variety of challenges [20] still requiring significant technical skill and knowledge, and is therefore difficult to be adopted for low-tech creators like artists and designers [6] but also for professional developers. Current research in HCI is focused on lowering the entry hurdles for non-technicians by demanding and providing authoring tools that require less to no coding skills [3, 51, 62, 73]. However, little is known about the situation for experienced professionals in this field, who have to face a melting pot of various disciplines, skills, motivations, and platforms which results in a fragmented environment of vocabulary, tools, methods, and approaches [8]. We contribute to the stream of research on AR/VR authoring tools by investigating how professional teams approach the challenge of AR/VR application creation, and how they make use of artifacts, tools and methods in their collaborative work.

In this paper, we report on the findings from our semi-structured interview study conducted with 26 professionals from the field of AR/VR application development. We aimed to recruit experienced creators with varying backgrounds and skill sets to gain broad insights into the current approaches and challenges faced by professional AR/VR application creators. We interviewed software developers, managers, Design Thinking practitioners, and user interface (UI) designers who create applications for diverse domains to gain a better overview of the current situation in practice. We specifically concentrated on the collaborative character of this interdisciplinary field: Based on the tasks and goals reported in our interviews, we condensed the 4 roles into (1) *Concept Developer*, (2) *Interaction Designer*, (3) *Content Author*, and (4) *Technical Developer*. Even though a single person often incorporates multiple roles, their created artefacts are distinct since they serve differing goals in this collaborative environment.

The workforce diversity of the development teams with regard to their skill sets and backgrounds is both a challenge and a benefit. In addition to pointing out a lack of standards, best practices, and tools combined with rapidly changing hardware and software platforms, the majority of participants voiced issues regarding a "missing common language". This, they argue, is required for collaboration

in the sense of coordination and workflow with interdisciplinary team members and inexperienced end-users / customers. Therefore, they aim to rapidly create interactive artifacts for communication. The tool sets and methods applied during this process originate from various fields, such as software engineering, game design, animation, user-centered design (UCD), arts, graphic design, and 2D user interface design. We learned that this patchwork of tools further creates challenges throughout the implementation process: from the contextual analysis phase, to the research and prototyping phase, to testing and evaluating until the final product is built.

Our work adds to the findings of Ashtari et al. [3], Nebeling and Speicher [62], and Gandy and MayIntyre [36] and contributes as follows:

- (1) We provide empirical insights into the collaborative work practices in professional interdisciplinary AR/VR application development.
- (2) We provide further details about key challenges and workarounds encountered in interdisciplinary teams working with emerging technology.
- (3) We provide directions as to how HCI can contribute to making AR/VR toolchains more accessible and user friendly.

2 RELATED WORK

We draw on prior work in the areas of AR/VR authoring tools, work practices of AR/VR developers, and communication of concepts and ideas in interdisciplinary teams.

2.1 AR/VR authoring tools

In research, several works exist about AR/VR authoring which target creators with different levels of skill as well as different fidelity stages of the resulting prototypes [42, 62]. Based on the current authoring tool environments, low-fidelity tools generally require less programming skills, whereas high-fidelity prototypes need to be programmed and thus require advanced programming or scripting skills [62]. In practice, commercial AR/VR game engines and software development kits such as *Unity*, *Unreal*, *ARKit*, *ARCore*, *A-Frame*, *ARToolKit*, or *WebXR* are used.

Other tools focus on supporting the low to medium fidelity prototyping stages of application development with reduced to non-required programming skills (e.g. *Pronto* [52], *ProtoAR* [61], *GestureWiz* [73], *iaTAR* [50, 51], *PowerSpace* [43], *ARVIKA* [34], *Adobe Aero*, *wiarframe*, *Microsoft Maquette*, and *Reality Composer*). However, those tools are often limited in functionality and inadequate for supporting the whole development cycle [3, 62].

The patchwork of available tools indicates that AR/VR creators have to learn multiple tools, posing questions about how they are managed and appropriated by developers [23]. However, we still lack knowledge about the practices which exist for such tool usage "in the wild". With our study, we provide insights in how professional AR/VR creators make use of the available tools in order to accomplish their goals in a collaborative and interdisciplinary environment. Additionally, we detail the challenges the creators are facing given the admixture of tools and methods. We therefore complement research by Ashtari et al. [3], who focused on the learning process and barriers of inexperienced AR/VR creators.

2.2 Practices and challenges in AR/VR application development

Several studies exist that include information about AR/VR creation from the creators' perspective. However, those studies are often conducted in scope of the evaluation of specific authoring tools [48] and therefore do not reflect realistic creation processes.

In a recent survey, Speicher et al. [72] asked 30 AR designers, developers, and users to evaluate 2 scenarios for AR development and highlight potential technical hurdles. Their work provides insights into 6 major challenges encountered in AR development: cross device / cross platform communication, mapping of the environment, obtrusiveness of devices, gesture recognition, tracking, narrow fields of view (FoV), and immersive scaling and sizing of AR objects.

Gandy et al., the creators of the AR prototyping tool *DART* [54], evaluated how creators used their application 10 years after publication [36] and detail the needs of non-technologists and requirements for future tools designed for supporting AR authoring. Similar to Nebeling and Speicher [62] and Ashtari et al. [3], they highlight the need for authoring tools that do not rely on coding skills.

Based on this prior work, Ashtari et al. [3] emphasize the importance of considering low-tech creators as a developing target group for future AR/VR authoring applications. In their work, they identified 8 barriers encountered by creators with low technical skills by interviewing 21 AR/VR creators with both, professional and amateur background: (1) knowing where to start, (2) making use of online learning resources, (3) lacking concrete design guidelines and examples, (4) designing for the physical aspects of immersive experiences, (5) planning and simulating motion in AR, (6) designing story-driven immersive experiences, (7) encountering many unknowns in development, testing, and debugging, as well as (8) testing users and evaluating challenges. Furthermore, they highlight potential entry points for building more accessible tools, such as integrating learning opportunities, supporting early-to-middle-stage AR/VR prototyping, personalizing authoring tools based on expertise, integrating access to learning resources, and integrating debugging and testing facilities.

We aim to extend this work by providing empirical insights in the authoring process of professional development teams of AR/VR applications, similar to the approach of Dow et al. [22], who investigated the challenges of designing for non-traditional systems, such as ubiquitous computing. We specifically target the challenges for AR/VR creators compared to designing 2D systems from contextual analysis to deployment, as well as challenges and workarounds imposed by available tools and methods in collaborative work environments.

2.3 Collaborative prototyping in interdisciplinary teams

Prototyping methods are widely accepted as an important part of software creation within HCI. Prototypes support the visualization of ideas and serve as boundary objects [74], enhancing learning and collaboration in the co-design projects of interdisciplinary teams with different stakeholders, such as designers and end-users [15, 21, 32, 37, 68]. Often, prototypes and their methods are differentiated regarding their fidelity and the effort necessary to create them, as

well as tools applied during the creation process [56, 62]. While this alone might not be sufficient to describe the usage and application fields of prototypes as valuable communication artifacts [29, 44, 58], one might argue that detail about the characteristics and roles the prototypes play in the design phases is lacking.

In product development, prototypes are used to save time and resources during later phases of the product evolution process by exposing flaws and misconceptions while their identification and correction is still easy and cheap. However, creators have to find the balance between effort and the required realism of created artifacts. Benmahmoud-Jouini and Midler [5] address this problem of overdesigned and overtrusted prototypes by providing a framework including 3 archetypes for categorizing prototypes according to their the purpose and intended application: stimulators for exploring the user needs and their context, demonstrators for showcasing concepts and their relevance regarding the preceded specifications, and validators for testing close-to-the-market solutions. Whereas stimulators are meant for ideation purposes and open-ended thinking, demonstrators target first concept evaluations and validators allow for detailed development.

In AR/VR application creation, prototyping sessions are embedded in UCD approaches with a focus on end-user or stake-holder involvement, for example during participatory design (PD) sessions as well as co-creation and experience prototyping workshops (e.g. [11, 14, 24, 27, 77]). For instance, Brösting and Gruhn [10] developed a concept that enables developers and UI designers to collaborate on the creation of industrial AR applications. They create "Interaction Stories" as boundary objects which are modeled in-situ and transformed into code snippets. To our knowledge, besides that, the role of interactive artifacts for communicating within interdisciplinary teams in professional AR/VR creation has not yet been investigated in detail.

In our work, we provide initial results about the methods and tools used for prototyping in interdisciplinary development teams combined with the purpose and reuse of resulting artifacts. By doing this, we want to support the creation of a more user friendly AR/VR tool chain for collaborative prototyping processes.

3 STUDY DESIGN

We followed a qualitative approach to explore the field and conducted semi-structured interviews using online video conferencing tools. In total, we interviewed 26 professional user experience (UX) designers and developers who were actively working on AR/VR software creation. Along with the identification of roles and tasks as well as their interplay, we also focused on tools, methods, challenges and workarounds applied in their daily work.

3.1 Participants and recruitment

For recruitment, we followed a two-step approach. First, we asked creators we personally knew about their networks and platforms in order to establish contact with experienced professional UX designers and developers of AR/VR applications. Besides asking them to spread our study request to their personal contacts, we also posted the request on relevant Slack channels, Meet-ups, Twitter, Facebook, and LinkedIn groups. Second, we received support from recruited interviewees who also spread our request over their communication

channels. Our aim was to sample a diverse group regarding their background, application area, target devices, and local distribution. In total, we recruited 26 participants (f10/m15/o1) with different roles in the AR/VR application development process. Based on their skill set, we grouped them as follows: Creators with design skills (D), creators with design and coding skills (DC), creators with coding skills (C), and managers (M). Apart from the managers, all participants were experienced in 2D application development. Table 1 provides an overview of participants, backgrounds, experience on the job, their application area, and whether they received formal training for AR/VR. 15 participants had worked on applications based on actual customer requests (pull-applications). The remaining 9 interviewees were developing applications based on their own motivations and ideas (push-applications). All developed applications' technology readiness levels were above 5. We recruited interviewees without regard to their place of residence and ended up with professionals from Europe (Germany, Austria, Italy, Hungary, France, Great Britain) and few samples from USA and Canada. The interviews were conducted in both German and English. The literal transcripts were translated to English by a German native speaker with a C1 skill level in English. Our participants covered the following age groups: 18-24 (4), 25-34 (17), 35-44 (2), 45-54 (2), and 55-64 (1).

3.2 Study procedure

Before the interview, we discussed the details of the study and clarified remaining questions with our participants. The interviews themselves were conducted using video conferencing tools like Skype, Zoom, Google Hangouts, and Microsoft Teams, allowing participants to also show us the artifacts and tools they were using in their creation processes. In the beginning, we introduced ourselves and let the interviewees talk about their tasks, experiences with AR/VR tools and devices, their working environment including team sizes and relevant departments in addition to their employers' application domains. Furthermore, we were interested in their personal experiences with 2D application development. We then moved on to the main interview questions which we constructed based on the DAkKS usability guidelines which provide an established question catalogue for context interviews [1, 45]: We designed the questions to investigate the full process of developing an application ranging from planning, preparation, and execution to evaluation and transfer. Since we were also interested in the types of applications our participants were building, we asked them to explain their approach based on a recent or ongoing product they were developing. During their explanations, we investigated their tool usage, created and prepared artifacts, information and inspiration sources, knowledge exchange, and collaborative development approaches before letting them point out differences to other AR/VR projects they have worked on. Furthermore, we asked them to compare their practices and tool set for AR/VR development to their experiences in 2D application design and finally voice their wishes concerning future tools, methods, and approaches for creating AR/VR applications. In the end, we collected demographic information, such as age, gender, job experience in years, and occupation.

Table 1: Summary of study participants; on job (years) refers to the participant's working experience in the field of AR/VR application development

ID	Background	Occupation	On job (years)	Main topics	Had formal training
Managers (M)					
ID14	Biomechatronics	Executive Director	4	Game Design, Architecture	✗
ID22	Media Technology	Executive Director	7	Game Design, Architecture	✗
Creators without coding skills (D)					
ID2	Science	XR Experience Designer	4.5	Games, interactive Stories	✗
ID4	Graphic Design	Interaction Designer	7	Experiences for Museums	✗
ID5	Graphic Design	Design Thinking Consultant	3	Agri-Food Business, Training	✗
ID10	Service Design	UX/UI Designer	0.6	Energy, Climate Change	✗
ID13	Electronic Visualization	Experience Designer	2	Education (Space)	✓
ID16	Fine Arts	Experience Designer, Visual Artist	3	Location-based Entertainment	✓
ID25	Cognitive Science	AR UX Designer	4	Energy	✗
Creators with design and coding skills (DC)					
ID1	Human-Computer Interaction	AR Designer	4	Architecture, Manufacturing, Sales	✗
ID7	Media Informatics	Software Developer	3	Architecture, Health, Museums	✓
ID8	Industrial Design	Creative Director AR/VR	5	Virtual Environments, Art	✗
ID9	Graphic Design	Product Owner AR/VR Technologies	6	Energy	✓
ID11	Media Informatics	Software Developer	4.5	Marketing, Sales	✗
ID15	Media Informatics	Software Developer	7.5	Architecture, Marketing, Sales	✓
ID18	No higher education	Interaction Designer, Developer	1	Education	✗
ID19	Digital Media	UX Designer, Developer	2.5	Location-Based Entertainment	✗
ID20	Computer Science	Product Manager, Developer	6	Architecture, Construction	✓
ID21	Graphic Design	UI/UX Designer	5	Automotive, Remote Assistance	✗
ID23	Digital Media	UX Designer	4	Remote Assist Applications	✗
ID24	Game Design	AR Product Designer	4.5	Consumer Experiences	✗
ID26	Game Development	MR Evangelist, Dev Tech Engineer	8	Automotive, Arts, Game Design	✓
Creators with coding skills (C)					
ID3	Computer Science	Software Developer	3	Science, Architecture, Energy	✗
ID6	Media Informatics	Software Developer	12	Exhibitions (Fairs)	✓
ID12	Media Informatics	Software Developer	7	Architecture	✗
ID17	Game Design	Software Developer	8	Location-Based Entertainment	✓

3.3 Data analysis

We adopted an open coding approach much as suggested by Strauss and Corbin [75], but - because the paper is exploratory - do not seek to develop axial and selective codes to investigate roles, current practices, tools, and challenges of experienced AR/VR creators. The transcripts were organized and coded in *MAXQDA*. We developed and evaluated the coding schema dynamically on the first 8 interviews with the use of affinity diagrams created in *MIRO*. After that, we applied it to the remaining transcripts. The coding schema delivers insights into the application creation process with a focus on tools and built artifacts for communication between the different roles. Furthermore, we identified challenges and their workarounds.

The following sections present our main findings. We first introduce different roles, their tasks, tools, and the artifacts created in the process of AR/VR application development. After that, we introduce our main findings regarding challenges in an cooperative landscape encountered by AR/VR creators.

4 ROLES, TASKS, AND TOOLS IN PROFESSIONAL AR/VR CREATION

Our sample revealed the co-existence of at least 4 roles participating in the development of AR/VR experiences, even though roles cannot always sharply be distinguished from each other and transitions between tasks, tools, and artifacts happen fluidly:

I think in AR/VR, there is less definition between the roles. Whatever needs to be done next, you just pick it up and work on it. ... There is no point where my responsibility ends. (ID24-DC)

Often, a single person incorporates multiple roles, depending on the team's size and complexity of the experience in development. In our sample, teams ranged from 1 to 10 creators working on the same project with a diverse composition of skill sets. 3 of our participants worked as a freelancer, 10 were part of a mixed-skilled core team who hired freelancers if a specific expertise was needed. 1 worked in a design department of a company, 1 was part of a software development department, and 11 worked in teams with mixed professions who did not hire freelancers. Dedicated designer roles existed, but sometimes suffered from a lower standing in the department compared to colleagues with a mixed skill set or the role of a technical developer when it came to hardware and tool availability:

[A head-mounted display] is still expensive. ... From some companies' perspective, [designers] are not the ones who produce, so giving your HoloLens to developers for testing their code makes sense. Giving a HoloLens to designers for them to play with it and try random things, that's another topic. (ID25-D)

On the other hand, our participants valued the fact that there are many skills required to create outstanding applications:

You are just one person, you cannot do everything. Compared to big companies like Microsoft, where there is a dedicated designer for everything, our applications are good, they still work and people like it, but they do not have those crazy wow-effects, properly

cast shadows, highlights, ambient sound design because there was a dedicated sound designer involved. (ID23-DC)

You cannot be an expert in everything. It requires collaboration. ... The need for collaboration in AR/VR is greater because of the complexity and the need for a number of different assets. (ID13-D)

The roles we could identify based on our data set are not correlated with the ones detailed in Table 1 since several of our participants shared similar tasks regardless of their expertise and skill set and therefore incorporated multiple roles. We identified the four roles: (1) concept developer, (2) interaction designer, (3) content authors and (4) technical developer:

Concept Developers (22 of our 26 participants) typically focus on creating the first concepts and drafts of an application, ideally by ignoring technical limitations and focusing on the problem to be solved. This might happen based on concrete customer or user-needs, in which case they also act as mediators between end-users or customers and the remaining AR/VR creation team. Concept developers usually work with technology agnostic methods based on the UCD process, for example applied in contextual analysis:

If the use case is not yet clear, we usually start with investigating in the problem and decide afterwards which would be a fitting device. It could also be that the use case itself is already clear and limits the selection, for example if some hands-free interaction is required. ... We want to find a good and logical use case that is not only based on a technology hype. (ID5-D)

It could also happen that application concepts are developed from scratch based on ideas and fictional use cases, for example for location-based gaming:

We want to teach people about the development of technology in the near future and its impact on society. We think it is important that people do learn about the danger and challenges they will be facing in the future. We want to do this based on a games approach because technology is complex and it should not be boring to learn about it. ... AR/VR is also part of this technology and will have a huge impact on society. (ID16-D)

Concept designers often produce artifacts for the sake of documenting, such as video protocols of reenacted scenes in which the application concept is fictionally put to work, photo diaries, user needs and requirements, or other system specification documents, roles, and application environments.

Interaction Designers (23 of our 26 participants) handle the mechanics and interactivity of the application, usually on a conceptual level. They design for locomotion, navigation, and input / output, as well as the interplay of various modalities used in AR/VR systems. They research similar projects and solutions or novel approaches for interaction design on social platforms, app stores, movies, books, and in AR/VR communities:

For example, if I would design some sort of medical application, I would download all the medical applications I could find for AR/VR and observe the interactions other people have been creating. ... This is for opening up your mind because every designers is at the same spot in a way that there are not many standardized patterns, so everybody comes up with novel approaches. ... Sometimes, somebody has come up with a really neat solution that also works well in your application. (ID1-DC)

Interaction designers mainly produce artifacts that aim at communicating and evaluating ideas regarding interaction, information architecture, information flow, and structural issues, such as interaction flow diagrams, storyboards, and wireframes.

Content Authors (10 of our 26 participants) focus on the creation of animations, 3D models, visuals like shadow casts, textures, and color schemes, or sound design such as voice overs and music. Depending on the application, they also design 2D-screens or 2D elements, for example for AR applications that run on mobile phones and tablets, or text-based annotations. Unfortunately, our sample only included 2 content authors with a self-reported dedicated skill set in 3D modeling. The majority of our participants described themselves as having only basic knowledge. Therefore, they used pre-built assets or outsourced their creation:

Some other people help with the UI design, like there is one person that mainly does the art and the graphics for stuff. He is the one that models [the 3D assets] and adds all of the cool effects to them. And in his case, I'll just graybox¹ it or draw a super simple thing in my [3D modeling software] and describe what I want and he'll get it right. (ID18-DC)

Content authors produce continuously enhanced artifacts with the goal of reaching a final state. Final artifacts are then used as content in the resulting application. Intermediate states of those artifacts are used for communicating and evaluating their approaches.

Technical Developers (15 of our 26 participants) do not only produce code and develop features. They often provide consultation regarding the technical feasibility and practicability of concepts and therefore know about current hardware, development frameworks, techniques, and limitations. Besides that, they conceptualize and build the applications and implement custom interactions, image recognition, and positioning. Additionally, they provide support when it comes to hardware selection, software framework selection, tracking techniques, development environment, plug-ins, libraries and network related problems such as cross-device application development. Finally, they combine inputs from concept developers, interaction designers, and content authors and influence major design decisions regarding their technical feasibility. Artifacts produced by technical developers are mainly interactive or working AR/VR prototypes of different fidelity levels as well as custom software and script snippets, for example for gestures.

¹Gray boxing is a prototyping technique in which the designer uses gray boxes as place holders for 3D models in the virtual environment; this method is applied when the focus lies on spatial interaction, positioning and scaling without the influence of visual effects and representations and was reportedly executed in Unity. The gray boxed application is then deployed on the target device and tested.

AR/VR creators apply a mix of various tools and methods to reach their aims. This process results in a collection of artifacts that are similar for the roles and, based on their purpose, independent of applied tools and methods. Only Unity is an exception, since all of our participants' teams developed their products using this game engine. Besides that, there is no software-based tool that was favoured for prototyping. Sketching and wireframing were the methods that were mentioned most often.

5 CHALLENGES FOR AR/VR CREATION IN INTERDISCIPLINARY TEAMS

AR/VR has a unique set of challenges for creators due to the three-dimensionality and novelty of the medium. We asked our participants about the barriers they perceive as being the most important ones compared to 2D application creation and identified 3 major areas: (1) team-internal misconceptions about the medium, (2) lack of tool support and appropriate methods, and (3) the absence of a common language.

5.1 Challenges caused by misconceptions about the medium

AR/VR creation is challenging because it requires the creators to have a mix of knowledge regarding the functionality of AR/VR software and hardware as well as design practices and skills. In interdisciplinary teams, there are roles that focus more on the design aspects, whereas others are more engaged with technical limitations. In particular, creators who do not have a technical background often have unrealistic expectations about what AR/VR can or cannot provide. For them, it is difficult to differentiate between renderings for marketing purposes or actual applications which can lead to an overestimation of hardware and software performance, which might result in costly adaptations to already matured design prototypes:

... I had too many lights and it was using too many resources, so the frame rate was very low for the [device]. We had to iterate through that a lot. ... I also had some problem with transparencies which caused a jitter and I had to work on that a lot. (ID2-D)

Other participants reported trouble that was caused by being unaware of hardware limitations like only having specific input devices or modalities, or a narrow field of view:

The field of view is very important. My first experience with the Hololens1 was that I could not really understand the field of view in Unity. (ID23-DC)

... but [the device] doesn't have an amazing field of view. ... They have to reconsider all of their design thoughts. (ID26-DC)

In addition to that, the robustness of sensory input like image recognition and image detection regarding environmental factors is often overestimated. This regularly results in the need to design around hardware limitations when features do not work perfectly, such as tracking in varying lighting conditions:

You can't always predict how the lighting conditions are going to be [in the target environment], and you can't always control everything. [When I am asked if

Table 2: Summary of challenges, solutions, and created artifacts

Key challenges	Problem	Solution	Artifact examples
Misconceptions about the medium	Overestimating hardware and software performance, being unaware of hardware specific limitations, overestimating the robustness of sensory input regarding surrounding environmental factors, projection incompatible experiences from 2D design	Creating awareness via demonstration and experience sessions, creating quick and dirty evaluation artifacts, involving technical creators as feasibility evaluators throughout the design process	Mood boards, working applications
Lack of tool support	Missing a spatial environment for designing or testing, prototyping AR/VR-specific system behavior to get a working prototypes without coding, lacking a full integration into the creator’s workflow (e.g. missing design specs), creating inaccessible artifacts for creators with a non-overlapping skill set, finding tools, using prototypes as final product, causing a decline in code quality and reusability	Teaching each others tools (e.g. paired programming), creating physical prototypes in co-creation sessions, falling back to robust prototyping techniques and doing joint explanation sessions	Physical prototypes, annotated information flow diagrams, sketches, wireframes
Missing a common language and shared concepts	Sharing precise descriptions of system behavior and design ideas, creating unsatisfying artifacts due to inappropriate fall-back options (e.g. designing in 2D)	Doing joint prototyping sessions (e.g. live coding), creating interactive or animated artifacts	Video clips, animations, gray boxing

the tracking will work a 100%,], I’m like "I don’t know. It might not be". (ID11-DC)

Some creators mainly working on design tasks also reported that they tried to simply transfer their experiences from the 2D design spaces to spatial interfaces and failed. While information is typically conveyed using specific output devices and fixed locations for 2D applications, AR/VR embraces the 3D environment which is challenging when it comes for example to positioning UI elements:

... and then we realized that sometimes you have some content that is facing you, but then you will look somewhere else and then you lose where your content was. And it’s like we really realized all the difficulties to have 360 degrees interfaces instead of just one in front of you because people tend to lose their screens everywhere. (ID25-D)

To cope with the problem of misconceptions, teams came up with collaborative solutions. In their understanding, it is not necessary that all involved parties know exactly how AR/VR works on specific devices in particular environments. It is more valuable to create awareness that limitations exist and then to communicate and validate ideas to minimize the amount of wasted efforts:

Usually, you make prototypes to pitch an idea. The whole purpose is to show "Wouldn’t this be so cool?", but you also need to present a realistic idea. You need to make sure that your idea is good both in a use case and in realism, like it can actually be implemented. (ID24-DC)

Some teams also involved both developers and concept designers from the beginning on to validate ideas before they evolved into prototypes by using previously created artifacts:

[Me and the developers] went to the company and experienced the workflows on premise. ... What we always do in our workshops: Everybody has to use [AR/VR] at least once. ... We use showcases from the developers. Those are applications they developed before and ideally demonstrate both their skills and what has been done in the domain. It is impossible to understand and feel [AR/VR] if you have not used it at least once. (ID5-D)

One participant provided insights into later design phases, when user needs get broken down into concrete features and design ideas:

Developers are involved to tell the [Product Owner] or SCRUM Master if this feature is feasible or not. ... During design, there is always a developer next to me. And I tell him or her that I am designing this interaction and I ask them if this is possible. They will say: Yeah, it’s ok, you can go on. Or: No, there are some technical limitations. (ID23-D)

Another way of handling misconceptions is to create quick and dirty artifacts that are just meant to validate the technical feasibility of ideas in team-internal sessions:

... We have a product owner with no technical skills. However, she is aware about the AR subject and that there are constraints she does not know. She creates lots of mood boards with visuals, and asks: "Hey, this might look cool!", and the developer does then evaluate if the concept works or if it does not work, if it needs to be adapted, etc. (ID4-D)

The content of those mood boards is manifold but easy and fast to provide and share. They consist for example of sketches, photographs, screenshots from video games, video snippets, link collections, and animations.

5.2 Challenges caused by a lack of tool support and fitting methods

Experienced creators select more advanced tools like *Gravity Sketch*, *Microsoft Maquette*, or *MRTK* based on their past experience. However, the short lifespan of today's tools and the fast pace of the hardware development makes it difficult to reuse previously created artifacts and forces creators to continuously learn new tools (e.g. ID1-DC, ID8-DC, ID23-DC). It is also worth mentioning that some participants reported being limited in their tool choice due to the companies' policy or practice of having a set of paid tools which needed to be learned by their employees (eg. ID1-DC, ID23-DC). Sketching and creating graphs for explaining how the envisioned application should behave are methods our participants used mainly if they were not directly jumping to Unity for creating their prototypes. Using those "classical design tools for designers" (ID8-DC) works well as long as applications are not too complex and ideas are still rough. When it comes to spatial distribution, the task of designing AR/VR applications gains complexity:

You mostly want to explain the ergonomics of the application, the layout of your UI elements, for example where is this button, where is this asset, where is the position of this map.(ID23-DC)

Participants report that available tools often lack the 3rd dimension either in the design space or the test environment:

I think the biggest [problem] is the jump between 2D applications and the fact that we actually design 3D environments and spatial interfaces. ... The missing depth is the most obvious one. Having to constrain [the design] to a 3D plane instead of actually having a spatial environment that is then shown on a screen which is also not a spatial environment feels more like doing photographs about 3D trying to tell the reader how it would look like if it would be in 3D. (ID1-DC)

If you use [tools like Unity], you are still trapped in your 2D screen. And this is the whole point: It's always good to put yourself in the shoes of the user [and their context]. (ID23-DC)

Even though there are many tools, less technical creators feel they do not match their needs since there is a trade-off between easy-to-use and effective tools. Our participants reported, that besides lacking depth as information, interactivity, animations, and story telling are also difficult to prototype. They therefore have to use a mix of tools in order to get specific about details like interactivity, spatial distribution, orientation, and scaling:

The tools are either too complex, so you need to know how to develop the application [because they are frameworks for developers]. The applications that have been designed for designers do not handle how to get a working prototype. They are too simple feature wise. (ID1-DC)

Sometimes, it is very difficult to put the animation into work. So I sometimes use Adobe After Effects just to show the animations. (ID23-DC)

Our participants reported using a mix of 2D design and mockup tools like *Adobe XD*, *Figma*, *balsamiq*, but also more advanced software such as *Doodle Lens*, *Tilt Brush*, *Gravity Sketch*, and *Microsoft Maquette*. However, the mix of tools is not well integrated into their workflows:

If we mockup [the application], there are no super good tools. Microsoft Maquette is the most advanced one, but then it is difficult to get your design specs out of it. (ID1-DC)

Therefore, less technical creators require the help of developers to move their prototypes and ideas to the 3rd space in order to evaluate "simple" things like scaling, positioning, and interaction:

[F]or me, really the biggest problem is that once we want to prototype interactions like gestures and the 3D thing, I have absolutely no tool to do that. So I really have to rely on the developers to code something so that we can test it, because otherwise there is no way. (ID25-D)

At the moment, [the biggest hindrance] is the problem of not easily being able to test out these different interactions I have on my mind, like: ... Should [the asset] follow me with ... slight delays, should it turn when the head goes outside the UI? And then I might not be able to test it out before it goes into development. So I might need the help of a developer to test out those kind of basic spatial UI things that are common. I think in [Microsoft Reality Toolkit] (MRTK), there might be scripts, but I cannot use MRTK in my current assignment because [the device is not compatible]. (ID1-DC)

However, transferring the artifact into an environment where it becomes inaccessible for alterations executed by the low-tech creator causes conflicts:

For example, for UI layout, I would just code everything. I would think it is super smart, but it's not because then the designer would have less access to changing things because he wouldn't know how to change it. (ID11-DC)

On the other hand, it also happens that technical creators with little design knowledge sometimes have to take over the role of content creators and interaction designers:

It's not like in normal mobile development [where] UI/UX designers do the big part of the design. In [AR/VR], it is mixed and developers also have to do a bit of the design, which is a very hard task because we are not designers and we have to find tools that can help us designing. (ID3-C)

One participant reported that their team addresses this tool gap by teaching each other the use of different instruments to lower the barrier and keep artifacts accessible:

On my team, right now everybody knows how to use Unity. ... We did a lot of paired programming to

show [the inexperienced creators] how it works. So we just kind of teach each other all the time. And that has evolved into everybody knowing Unity, and as a programmer, I should also know a bit of the design side, so I'm also learning Figma (the tool designers use) in order to understand each other better. I'm not a pro on that, but it is enough to do at least basic things if I have to substitute [the designers]. (ID11-DC)

Other teams use physical prototyping in collaborative sessions, where developers and designers jointly work on mimicking interactivity, positioning and scaling to better understand the overall application:

It helps to understand the physicality of the experience. Because you are surrounded with all elements that do exist in real life and it's hard to imagine that when you're just looking at the screen or just a paper. (ID25-D)

It is to make sure that your design will actually make sense. So I started with a little bit of role playing and putting stickers on top of each other. I used transparent papers for holograms and put something on top, for example for spatial annotations. I was doing these kind of things to really feel if the interaction metaphor could really work in space. (ID23-DC)

Especially the developers were positive about physical prototyping in 3D, because they usually only get those descriptions and then it is difficult for them to imagine how the application should behave. (ID3-D)

However, physical prototypes can become really complex when multi-modality has to be modeled, such as spatial sound which has then to be played by other designers or participants (ID5-D). Besides that, there are other hindrances:

[Physical prototyping for AR/VR] has a lot of constraints because you cannot do it remotely and since the environment where the application should be used is in our case difficult to be replicated in an open office space, it would not work. (ID25-D)

Physical prototypes take time to be constructed and are therefore only feasible if the project budget and resources allow it. In order to save time, participants reported that their team prototypes directly in Unity:

Eventually, the working prototype is actually the application itself and is created by developers, this is why I just stick to UX writing based on this information flow diagrams². (ID23-DC)

For 2D, it is much simpler because you can just draw on your whiteboard to show what you want to say. In AR, people have to think about it in 3D which is quite challenging to understand. Also, for 2D ... realistic prototypes can be done very easily, for example with [drawing software], but for AR, at least right now, you

have to go through the development process. (ID7-DC)

Additionally, participants reported that they had trouble in getting rid of artifacts which were no longer needed, because they had to put a lot of effort in constructing them:

If you have very complicated prototypes, such as programming your own hand gestures, you want to keep using it which might be problematic: If you spent a lot of time on it you do not want to scrap it. This is always a problem with AR prototypes. If you spent time on it, you built something like an emotional bond. (ID7-DC)

In fact, several participants reported that an artifact originally developed as a prototype was shipped as the final product due to time and budget constraints, and a lack of understanding from the customers' side for necessary code refactoring "because the prototype already looked so good" (ID6-C). This caused problems regarding code readability and code reusability in addition to an increased amount of necessary work with respect to future changes and maintenance.

5.3 Challenges caused by missing a common language and shared concepts

Our participants mentioned that a missing common language and shared mental models often cause confusion and leave uncertainties about design ideas and implementation tasks. Developing a common understanding of how the envisioned AR/VR application should behave and what they look like requires knowledge exchange via artifacts [17, 21] that can convey envisioned concepts as precisely as possible:

We think that the interactions are the most important part of an AR Game because it adds to the immersion. If you have good interaction, it feels natural. ... It is often that if you talk about something, everybody imagines it in a different way and there is always a difference between my interpretation and my second programmer's understanding. We had a lot of problems in the beginning of people implementing things in a wrong way and then having to scrap it completely. In order to cope with that, we decided to be exact about our interaction types and make them to become manifest. (ID17-C)

Furthermore, people bring the vocabulary and concepts from their field of expertise and sometimes have to fall back to a common ground of knowledge as a workaround. This results in artifacts that cannot hold up to the creators' expectations:

In the beginning, we [developer and designer] did not have the same language. This is why we went for 2D prototyping instead of 3D prototyping – we might have found something better if we would have had the same language. (ID7-DC)

As reported in Section 5.2, AR/VR is difficult to explain and talk about when using still images and sketches due to the complexity of the applications without at least some sort of interactivity:

²ID23-DC creates graph-based visualizations of the application logic and adds annotations to describe the interactivity and application flow similar to information flow diagrams [30] or user flow diagrams [53]

If you show moving pictures or simple animations, you can see how people better understand the point you want to make. (ID8-DC)

However, doing animations is expensive and time consuming due to the skills required for producing animated artifacts which provide enough interactivity to explain the intended behavior of the design. Our participants reported also falling back on more robust and fast approaches, for example sketching, and providing interactivity by explaining the ideas in face-to-face sessions:

[We communicate with the developers through] speaking and sketching. [This is] the cheapest, quickest way. When it's something a bit complex that needs to show some interaction, for instance, and it requires a really clear sketch, I do a wireframe. Otherwise, we simply discuss together and sometimes [the developers] do live coding [where we can directly assess the changes]. (ID25-D)

Other participants went for creating visual artifacts like storyboards and interaction flow diagrams with precise, annotated descriptions of how the system should behave. However, AR/VR lacks a standard set of interaction patterns [3] which makes it difficult to depict interactive and animated system behavior:

... it's more due to the understanding people have or do not have of what AR is. A 2D prototype for a 2D UI is still an approximation. No matter how polished it feels or looks, you still have to imagine transitions between screens and the state of buttons if they are not present in the prototype, but because you used a lot of those apps or websites, you can fill in the gaps with your internal libraries. This still sometimes leads to misunderstandings, but at least there is a shared library of how things work or knowledge that can be used to fill in the blanks. In the case of emerging mediums like AR, people do not have this internal library. ... It is even more complicated with spatial sound. ... You cannot expect people to understand such a medium by just talking about it. (ID4-D)

An approach reported by participants who possessed coding skills involved a combination of programmed artifacts with limited interactivity and "acted out" system behavior to demonstrate envisioned applications. This approach is also known as gray boxing:

We used a marker as anchor point to provide first impressions and check if we were talking about the same things. It was not beautiful and consisted of simple [virtual] boxes and spheres that were anchored on the markers just to see potential interactions. ... We also showed that one could now go to the left or the right and also perform other movements. We documented some of the findings afterwards as bullet points [in written form]. Others were implemented based on our memory because it would be intricate to describe exactly how the interaction behaves. (ID6-C)

6 DISCUSSION

Our findings add to the work of Ashtari et al. [3], Gandy and MacIntyre [36], and Nebeling and Speicher [62], and provide insights into the current practices, challenges, and workarounds of professional creators of AR/VR applications. Based on the reported artifacts and tasks, we identified at least 4 roles (detailed in Section 4) involved in collaborating on the creation of AR/VR applications in the professional field. Furthermore, we highlighted 3 key challenges, workarounds, and the resulting artifacts (summarized in Table 2) which surfaced during our interviews. We now want to discuss how our findings add to previous studies for enhancing future tool-support in AR/VR authoring.

6.1 Similarities and differences between end-user developers and professional AR/VR creators

To complement Ashtari et al.'s findings [3], we want to discuss our key challenges regarding similarities and differences between professional creators and end-user developers upon their tool usage and encountered problems. Following Ko et al.'s explanation [47], end-user developers differentiate from professionals regarding their priorities. Whereas professionals are being paid, end-user developers aim to support goals from their own field of expertise through the creation of software, such as hobbies or their jobs. For our comparison, we focus on 6 of the "8 Key Barriers in Authoring AR/VR Applications" [3], since we did not provide further detail about findings regarding testing and evaluation. Generally, we found that both groups have similar issues and needs when it comes to AR/VR application creation.

The first 3 key barriers (1) *Difficult to know where to start*, (2) *Difficult to make use of online learning resources*, and (3) *Lack of concrete design guidelines and examples* strongly overlap with our findings as reported in sections 5.1 and 5.3. As we reported, professional creators with few technical skills have trouble understanding the medium and its limitations regarding hardware and software and therefore risk developing designs that are difficult or even impossible to implement with current technologies. It also turns out to be difficult to discuss ideas and approaches with their more knowledgeable team members because, unlike in 2D development, there is not yet an established set of guidelines and standards. Our participants also reported on the broad tool landscape and, in case they were not restricted by their companies' policies anyways, the difficulty in finding sufficient tools (also reported by Nebeling and Speicher [62]). Here, the problem was mostly that many lacked spatial placement and interactivity features, and were either too simple or too complex for the creator's needs.

The findings detailed in Section 5.2 conform with Ashtari et al.'s key barriers (4) *Difficult to design for the physical aspect of immersive experiences* and (5) *Difficult to design story-driven immersive experiences* [3]. Similar to end-user developers, professional creators reported a lack in tool and production workflow support. We learned that creators with little or no programming skills are highly dependent on creators with a sufficient technical background to create working interactive prototypes. On the other hand, creators

with few to no design skills rely on collaborators with content creator skills in order to fill the application with life and ensure a good user experience.

The main difference that we noted between end-user developers and professional creators is their access to expertise. Interdisciplinary teams came up with collaborative solutions to work around the challenges imposed by the AR/VR development landscape and found ways of drawing from their collaborators' skills. Whereas we acknowledge that it is important to lower the entry-barriers for low-tech creators by designing new and easy-to-use tools based on their needs, we see that future tool development could beneficially draw on established approaches, workarounds, and communities present in the professional AR/VR development field. We provide further detail on this finding in Section 6.3. In addition to that, tools supporting the collaborative prototyping of creators with different skill sets and knowledge could also ease the appropriation of AR/VR as a complex medium.

6.2 Emerging roles of AR/VR creators due to specialization of skills and their impact on future authoring tools

We developed the role definitions based on existing tasks and goals in the AR/VR development cycle reported by 26 practitioners with a broad set of skills. Our participants worked on projects with varying levels of complexity and collaborated in teams of different sizes and with divergent levels of specialization. While this blurred the lines between already existing roles in our data set, we were able to identify four preliminary roles emerging from our pool of participants: Concept developers, interaction designers, content authors, and technical developers (see Section 4). Collaboration as a concept for AR/VR tool creation has been around for some time, for example Schmalstieg et al.'s Studierstube [70]. More recent AR/VR prototyping tools, such as XRDirector [60], additionally incorporate roles based on tasks or interdisciplinary collaboration [10]. However, the roles proposed in Section 4 are still at a high level and not yet unique to AR/VR teams as they can also be found in other software and game development teams. Based on our findings, we expect an increasing specialization for AR/VR designers and developers as well as the development of new roles and processes for application development as the field matures. Gandy and MacIntyre already indicated an expected "movement toward specialization among contributors" [36] for AR/VR application development. This increasing specialization also requires tools that support collaboration in interdisciplinary teams to cope with the medium's growing complexity.

A common way of handling complexity on the coding side is through convention over configuration (COC). This was introduced to web-development in 2004 through the Ruby on Rails framework and subsequently inspired many others. Initial authoring tools for AR/VR like AMIRE implemented a visual programming paradigm to support graphical designers [40] but were described as too complex for them and not domain-specific enough [42]. The TOTEM framework implemented a template-based approach to content creation for location-based games inspired by COC. It also described three distinct roles and their responsibilities in the development process: game designer, content creator, and programmers. The

assumption was that developers were needed anyway and already got first class support with rapidly developing IDE's. The benefits of decomposing systems into modules for separation of concerns have been advocated since the 1970s [63] and are part of the state of the art in object oriented programming. To support collaboration however, the programming could be separated from the content and design-related roles (although they would sometimes be held by the same person) through the tools that helped in creating structured data and exporting it in standard formats [79]. This is much like in traditional GUI programming, where widgets could be visually placed and stubs to be filled with interaction logic are automatically created. For Mixed Reality experience design, the placement and arrangement of the widgets in 3D space should be possible, maybe like in a collaborative version of the classic Tinmith mobile Augmented Reality modeling system [66].

As showcased in our study and also supported by literature [13], multi-disciplinarity and a broad skill set are needed in order to create usable experiences. However, it also comes with the pitfall of having to overcome diverging concepts, expertise, and incomplete individual knowledge, also known as "symmetry of ignorance" or "asymmetry of knowledge" [31, 69] in PD and computer-supported collaborative work (CSCW). Our participants reportedly experienced this when designers without programming background discussed their artifacts with technical developers and vice versa and realized that their individual knowledge was not sufficient to solve a given design problem due to the specificity of their skills. In addition, AR/VR is a medium that is still developing at a fast pace which renders it difficult for a single person to keep track of the most recent developments. Both experienced and novel users and creators face a landscape of hardware, software, and tools that keeps evolving with a few standards and guidelines slowly emerging [3, 62]. Therefore, one can view designing AR/VR systems themselves as being a wicked design problem [13] which requires "a greater diversity of knowledge and technical skill than any one practitioner can provide [...] for finding solutions" [46]. In her work about interdisciplinary team development for designing a platform for computer-supported collaborative play, Jennings emphasizes that multi-disciplinarity requires each team member to be "equally valued" and able to participate in the design process [46], much like Ehn and Kyng also described in the UTOPIA project [26]. Furthermore, interdisciplinary teams need well-constructed boundary objects for internal communication [15, 21, 32]. As our study and previous work [3, 36, 62] demonstrates, this is not yet possible in creating AR/VR systems, or requires additional efforts to come up with workarounds.

Another potential approach to breaking down and distributing work among interdisciplinary team members working on complex software is the creation or implementation of dedicated software engineering processes - corresponding to game and classical software development. In line with Musil et al.'s prior research about similarities and differences between game and software development processes, we see that AR/VR development teams might benefit from dedicated engineering processes to cope with challenges surfacing due to the diverse background and roles of team members [59]. While we did not specifically focus on already in-use formalized software processes in our study, we noted that the approaches

described by our participants are apparently similar to agile methodologies which are also strongly driven by incrementally evolving artifacts. Additionally, some of our participants reported that they successfully implemented SCRUM or at least follow a SCRUM-based development process. Be that as it may, McKenzie et al. investigated anticipated and practical application of agile frameworks in the game industry in New Zealand and observed that studios unintentionally diverge from the formal process without it being realized [57]. We therefore want to emphasize the need for further research to provide more detailed insights into applied development processes in AR/VR development and their analogies to game development and classical software engineering methodologies in order to improve how already existing or potentially new approaches can support interdisciplinary AR/VR development teams.

We also see that future tools for AR/VR creation could benefit from following a PD approach to better design for the needs of both, creators and end-users in AR/VR system development as well as provide support for the various roles in their development process. Ens et al. already argued that Mixed Reality (MR), and thereby AR/VR, are becoming commonplace and therefore one can "focus deeply on the nuances of supporting collaboration, rather than needing to focus on creating the enabling technology" [28]. The important point to note here is that now that MR is leaving the lab and entering workplaces and homes, it can finally be seen as an evolution of (2D) groupware and therefore lessons learned from over 30 years of CSCW studies can be applied and evolved. This includes the common ground of the development processes and CSCW [65], as well as the intersection of code and design in cooperative processes. However, one has also to consider Brooks' humbling notion of seeing our work as that of a toolsmith, who is designing a tool that is set out to make a task easier [12]. And when doing so remembering Culkin's famous quote that "we shape our tools and thereafter our tools shape us" [19].

6.3 Design implications for future prototyping and authoring tools

Given the fact that over the course of the last few decades a lot of prior research has already focused on AR/VR authoring and how such tools should be designed in order to support creators and developers, the question remains why those approaches are not yet established in tools in current use. In this regard, we argue that firstly, the field is relatively young and has not yet been fully adapted by the consumer market. While tools and hardware continue to rapidly evolve, a so-called killer use case is still missing. Additionally, standards have yet to be established. Therefore, it is a financial risk to invest in commercially developing authoring tools.

Secondly, research about the actual needs of design and developer practitioners from the field of AR/VR is scarce. With this work, we intend to close this gap by providing insights into the challenges, approaches, and needs of actual practitioners in this specific field. As pointed out by Dow et al. [22], it is crucial to focus on actual practitioners as well as the tools used in line with their strengths and weaknesses in order to push the development of future tools for non-traditional design environments - in this case AR/VR - in a promising direction.

Finally, we want to discuss design implications towards more usable development tools as suggested by prior work [3, 36, 62].

Use simple tools developed based on tasks and goals. As our study shows, practitioners favor quick and easy tools which help them to effectively, efficiently, and satisfyingly reach their goals. This is a classic usability engineering problem [45] and there is a large pool of methods to draw from, for example by applying PD in tool development. However, we experienced that AR/VR authoring tools in practice do not necessarily differentiate between roles, tasks, and goals, but are rather feature-bound and end up being either too complex or too simple in addition to not supporting crucial elements of creative work, such as the possibility of evaluating ideas and exchanging artifacts.

Draw from existing methods, approaches, and workarounds. Participants from our study came up with a variety of workarounds to overcome the obstacles imposed by their tool sets. Since AR/VR creation unifies several disciplines and design approaches, it makes sense to make use of the complete set of already existing methods and find practical ways of adapting or redesigning them to the new medium, or supporting them in their appropriation [23]. In addition to approaches from classical UI development and filmography (e.g. [52, 60, 61]), we encountered gray boxing, a method applied in game design for creating low-cost prototypes to construct spatial layout and scaling properties. This bears striking similarities to the use and study of prototyping in systems design, as described by Floyd to include widget toolkits, very high level programming languages, and database-systems, in order "to make effective work possible" [33], generate human-feedback, and "keep trying until you get it right" [16]. Our participants also voiced the usefulness of physical prototyping because it has a realistic view point, features spatiality, physicality, and real-life scales as well as enables easy interdisciplinary collaboration due to a low to non-existing learning curve of required tools. On the other hand, it has too many drawbacks, such as construction time, transferability, potential complex workarounds for interaction and multi-modality, and the restriction to a certain physical space.

Create well-built artifacts for interdisciplinary communication. A common approach to interdisciplinary team work is the communication via artifacts or prototypes to establish a common ground of knowledge. Design patterns as introduced by Alexander [2] were developed for this purpose and got adopted in other application fields, such as software engineering, collaboration [18, 35, 67, 71], HCI [9, 41], game-design [7, 80], AR [55, 81], and interaction design [4]. Therefore, design patterns could be an appropriate approach to establish a common language for AR/VR development if the field has matured to an extent where it is possible to draw from a rich pool of applications and experiences. However, as and when the field matures to such an extent, patterns require team members to be aware of them, learn and use new vocabulary, and might therefore be difficult to apply in practical AR/VR experience creation.

We suggest to focus instead on the creation of interactive and adaptive artifacts. As we saw in our study, available tools impose their limitations on the production workflow and the resulting artifacts or AR/VR application creation. We encountered various artifacts that were built based on the need to overcome those barriers and noted that even static artifacts like user flow diagrams

required some sort of superimposed interactivity to be understood by team members due to the complexity of interaction, spatiality, and multi-modality in AR/VR. Based on that observation we conclude that future tools should facilitate the creation of artifacts with animations, interactivity, and flexibility to be collaboratively assessed, adapted and refined. Since our participants reported that their artifacts resulted from workarounds of tool limitations, we recommend analyzing the reason behind the creation of the artifact rather than simply what they show or do not show [29, 44, 58]. Having such collaborative Mixed Reality artifacts ready-at-hand during the design process of a diverse team would allow for mutual learning and languages games in a Wittgensteinian sense of language as action [25], i.e. giving the words a meaning in their use context, and thus allowing the grounding of the design in the work tradition.

Make use of all three dimensions. As reported by our participants and also detailed in literature, designing for 3D in a 2D environment or on a 2D screen is cumbersome and feels unnatural. However, based on our findings, this differentiation between design space and application space is still imposed by authoring tools used in practice. In contrast to that, current trends in AR/VR authoring tool development, for example Nebeling et al. [60] and Leiva et al. [52], follow the "What You Experience is What You Get" (WYXIWYG) Editor principle from Lee et al. [49–51], who proposed an immersive authoring tool for concurrent content creation and validation in the application space. WYXISWYG leans on the "what you see is what you get" approach of today's graphical user interface editor prototyping tools. In their work, Lee et al. describe immersive authoring as beneficial when it comes to specifying spatial arrangements and behavior [49] because the resulting design can be evaluated while being created without the need of switching between a 2D content creation environment and a 3D application execution environment. Immersive authoring can therefore reduce the entry barriers to AR/VR creation for inexperienced designers and increase the efficiency of AR/VR application developers without programming skills. This is inline with findings from the field of location-based experiences and ubiquitous computing, where the need of in-situ authoring for appropriate ideation, reflection and rearrangement of content was described [78].

Finally, it is important to note that immersive authoring is not always the best fit for approaching design challenges, especially when the design of abstract problems such as programming logic is required [49].

Allow for interdisciplinary, collaborative creation. As highlighted in our study, compared to end-user developers who tend to work alone, professional AR/VR creators benefit from having access to experience beyond their skillset. There are many ways one could adopt to establish a collaborative setting in an interdisciplinary field, starting from creating more accessible communities by embedding social networks like Slack, Facebook, and Twitter, allowing for easier asset exchange by basing new tools on existing standards and incorporating known platforms, out-sourcing the challenges to more experienced creators (e.g. [38, 39]), or by ensuring that tools are backed by an active user community that is able to provide support if needed [36]. The crucial part is to empower creators to design decent artifacts that are efficient, effective, self-explanatory, goal oriented, and could be easily shared between and accessed

as well as altered from the different roles in an interdisciplinary setting [46, 74].

6.4 Limitations

We acknowledge the dominance of lab studies in our related work as a potential limitation of this paper. The majority of our participants used Unity as integrated development environment. While for some of our participants this reflected the market situation, by its very nature of qualitative study, we have no statistical data with which to make strong claims about representativeness.

7 CONCLUSION

As our findings illustrate, tools which were previously developed in lab-scenarios, resulting from specific end-user development application areas or created by re-using already existing frameworks from other application areas might not fully satisfy the needs of practitioners in the field and therefore require designers and developers to come up with creative workarounds.

We have presented insights into current challenges, practices and design implications for professional AR/VR creators based on a study with 26 AR/VR designers and developers. Our findings add to existing work from the field of HCI tool research for spatial application authoring by presenting 3 key challenges for professional creators and how interdisciplinary teams solve them: (1) Misconceptions about the medium, (2) lack of tool support, and (3) missing a common language and shared concepts. In addition, we identified 4 roles involved in AR/VR creational processes, namely concept developers, interaction designers, content authors, and technical developers. We think that taking approaches from practice as a base for developing authoring tools is beneficial when it comes to the applicability and usefulness of the results. In addition to that, the interdisciplinarity of AR/VR application creation affords collaboration. Future authoring tools should therefore focus on supporting the construction of well-built boundary objects for communicating ideas, concepts, and approaches.

ACKNOWLEDGMENTS

We thank the anonymous participants of our study for sharing their insights and challenges with us. Work for this paper was supported by the European Commission Horizon 2020 Framework Programme under grant No. 870037 and grant No. 820621.

REFERENCES

- [1] Deutsche Akkreditierungsstelle. 2010. Leitfaden Usability.
- [2] Christopher Alexander. 1977. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, New York, NY, USA.
- [3] Narges Ashtari, Andrea Bunt, Joanna McGrenere, Michael Nebeling, and Parmit K. Chilana. 2020. Creating Augmented and Virtual Reality Applications: Current Practices, Challenges, and Opportunities. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3313831.3376722>
- [4] Elisabeth Bayle, Rachel Bellamy, George Casaday, Thomas Erickson, Sally Fincher, Beki Grinter, Ben Gross, Diane Lehder, Hans Marmolin, Brian Moore, Colin Potts, Grant Skousen, and John Thomas. 1998. Putting it all together: towards a pattern language for interaction design: A CHI 97 workshop. *ACM SIGCHI Bulletin* 30, 1 (Jan. 1998), 17–23. <https://doi.org/10.1145/280571.280580>
- [5] Sihem BenMahmoud-Jouini and Christophe Midler. 2020. Unpacking the notion of prototype archetypes in the early phase of an innovation process. *Creativity and Innovation Management* 29 (January 2020), 49–71. Issue 1. <https://doi.org/10.1111/caim.12358> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/caim.12358>

- [6] Mark Billinghurst, Adrian Clark, and Gun Lee. 2015. A Survey of Augmented Reality. *Foundations and Trends® in Human-Computer Interaction* 8, 2-3 (2015), 73–272. <https://doi.org/10.1561/1100000049>
- [7] Staffan Björk and Jussi Holopainen. 2005. *Patterns in Game Design*. Charles River Media, P.O. Box 417 403 VFW Drive Rockland, MA, United States.
- [8] Alexander Boden, Claudia Müller, and Bernhard Nett. 2011. Conducting Business Ethnography in Global Software Development Projects of Small German Enterprises. *Information and Software Technology* 53, 9 (2011), 1012–1021.
- [9] Jan Borchers. 2001. *A Pattern Approach to Interaction Design* (1st ed.). John Wiley & Sons, West Sussex, England.
- [10] Ingo Börsting and Volker Gruhn. 2020. Towards Efficient Interdisciplinary Authoring of Industrial Augmented Reality Applications. In *Conference Companion of the 4th International Conference on Art, Science, and Engineering of Programming* (Porto, Portugal) (<programming> '20). Association for Computing Machinery, New York, NY, USA, 65–68. <https://doi.org/10.1145/3397537.3398474>
- [11] Tone Bratteteig and Ina Wagner. 2016. Unpacking the Notion of Participation in Participatory Design. *Computer Supported Cooperative Work (CSCW)* 25 (December 2016), 425–475. Issue 6. <https://doi.org/10.1007/s10606-016-9259-4>
- [12] Frederick P. Brooks. 1996. The Computer Scientist as Toolsmith II. *Commun. ACM* 39, 3 (March 1996), 61–68. <https://doi.org/10.1145/227234.227243>
- [13] Richard Buchanan. 1992. Wicked Problems in Design Thinking. *Design Issues* 8, 2 (1992), 5–21. <http://www.jstor.org/stable/1511637>
- [14] Marion Buchenau and Jane Fulton Suri. 2000. Experience prototyping. In *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques* (Brooklyn, New York, USA). Association for Computing Machinery, New York, NY, USA, 424–433.
- [15] Bill Buxton. 2007. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann, San Francisco, CA, USA.
- [16] William Buxton and Richard Sniderman. 1980. Iteration in the Design of the Human-Computer Interface. In *13th Annual Meeting, Human Factors Association of Canada*. Stage, Paul, Downsview, Ontario, 72–81. <http://www.billbuxton.com/iteration.html>
- [17] Alan Cooper, Robert Reinmann, David Cronin, and Christopher Noessel. 2014. *About Face: The Essentials of Interaction Design*. Wiley, San Francisco, CA, USA.
- [18] James O. Coplien and Neil B. Harrison. 2005. *Organizational Patterns of Agile Software Development*. Pearson Prentice Hall, Hamilton, NY, USA.
- [19] John M. Culklin. 1967. A schoolman's guide to Marshall McLuhan. *The Saturday Review* (pp. 51–53, 70–72).
- [20] Arindam Dey, Mark Billinghurst, Robert W. Lindeman, and J. Edward Swan. 2018. A Systematic Review of 10 Years of Augmented Reality Usability Studies: 2005 to 2014. *Frontiers in Robotics and AI* 5 (2018), 37. <https://doi.org/10.3389/frobt.2018.00037>
- [21] Steven Dow, Julie Fortuna, Dan Schwartz, Beth Altringer, Daniel Schwartz, and Scott Klemmer. 2011. Prototyping Dynamics: Sharing Multiple Designs Improves Exploration, Group Rapport, and Results. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) (CHI '11). Association for Computing Machinery, New York, NY, USA, 2807–2816. <https://doi.org/10.1145/1978942.1979359>
- [22] Steven Dow, T. Scott Saponas, Yang Li, and James A. Landay. 2006. External Representations in Ubiquitous Computing Design and the Implications for Design Tools. In *Proceedings of the 6th Conference on Designing Interactive Systems* (University Park, PA, USA) (DIS '06). Association for Computing Machinery, New York, NY, USA, 241–250. <https://doi.org/10.1145/1142405.1142443>
- [23] Sebastian Draxler, Adrian Jung, Alexander Boden, and Gunnar Stevens. 2011. Workplace Warriors: Identifying Team Practices of Appropriation in Software Ecosystems. In *Proceedings of the 2011 ICSE Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*. Association for Computing Machinery, New York, United States, Waikiki, Honolulu HI USA, 4.
- [24] Emmanuel Dubois, Guillaume Gauffre, Cédric Bach, and Pascal Salembier. 2007. Participatory Design Meets Mixed Reality Design Models. In *Computer-Aided Design of User Interfaces V*, Gaëlle Calvary, Costin Pribeanu, Giuseppe Santucci, and Jean Vanderdonck (Eds.). Springer Netherlands, Dordrecht, 71–84.
- [25] Pelle Ehn. 1993. Scandinavian Design: On Participation and Skill. In *Participatory Design: Principles and Practices*, Douglas Schuler and Aki Namioka (Eds.). CRC Press, United States, 41–77.
- [26] Pelle Ehn and Morten Kyng. 1991. Cardboard Computers: Mocking-it-up or Hands-on the future. In *Design at Work: Cooperative Design of Computer Systems*, Joan Greenbaum and Morten Kyng (Eds.). CRC Press, United States, 170–195.
- [27] Mahzar Eisapour, Shi Cao, Laura Domenicucci, and Jennifer Boger. 2018. Participatory Design of a Virtual Reality Exercise for People with Mild Cognitive Impairment. In *CHI EA '18: Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (CHI EA '18). Association for Computing Machinery, New York, NY, USA, 1–9. <https://doi.org/10.1145/3170427.3174362>
- [28] Barrett Ens, Joel Lanir, Anthony Tang, Scott Bateman, Gun Lee, Thammathip Piumsomboon, and Mark Billinghurst. 2019. Revisiting collaboration through mixed reality: The evolution of groupware. *International Journal of Human-Computer Studies* 131 (Nov. 2019), 81–98. <https://doi.org/10.1016/j.ijhcs.2019.05.011>
- [29] Konrad Exner, Kai Lindow, Rainer Stark, Jussi Ängeslevä, Benjamin Bähr, and Emilia Nagy. 2015. A transdisciplinary perspective on prototyping. In *2015 IEEE International Conference on Engineering, Technology and Innovation/ International Technology Management Conference (ICE/ITMC)* (Belfast, UK). IEEE, Washington, DC, USA, 1–8. <https://doi.org/10.1109/ICE.2015.7438659>
- [30] Kirill Fakhroutdinov. 2020. UML Information Flow Diagrams. Retrieved September 13, 2020 from <https://www.uml-diagrams.org/information-flow-diagrams.html>.
- [31] G Fischer. 2000. Symmetry of ignorance, social creativity, and meta-design. *Knowledge-Based Systems* 13, 7 (2000), 527–537. [https://doi.org/10.1016/S0950-7051\(00\)00065-4](https://doi.org/10.1016/S0950-7051(00)00065-4)
- [32] Gerhard Fischer. 2001. External and shareable artifacts as opportunities for social creativity in communities of interest. In *Proc. 5th International Conf. on Computational and Cognitive Models of Creative Design*. Key Centre of Design Computing and Cognition University of Sydney NSW, Sydney, AUS, 67–90.
- [33] Christiane Floyd. 1984. A Systematic Look at Prototyping. In *Approaches to Prototyping*, R. Budde, K. Kuhlenkamp, Lars Mathiassen, and H. Zullighoven (Eds.). Springer, Berlin, Heidelberg, 1–18. https://doi.org/10.1007/978-3-642-69796-8_1
- [34] W. Friedrich. 2002. ARVIKA-augmented reality for development, production and service. In *Proceedings. International Symposium on Mixed and Augmented Reality*. IEEE, Darmstadt, Germany, 3–4. <https://doi.org/10.1109/ISMAR.2002.1115059>
- [35] Erich Gamma, Richard Helm, Ralph E. Johnson, and John Vlissides. 1995. *Design Patterns. Elements of Reusable Object-Oriented Software* (1 ed.). Addison-Wesley Longman, Amsterdam, NL, 416 pages.
- [36] Maribeth Gandy and Blair MacIntyre. 2014. Designer's Augmented Reality Toolkit, Ten Years Later: Implications for New Media Authoring Tools. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) (UIST '14). Association for Computing Machinery, New York, NY, USA, 627–636. <https://doi.org/10.1145/2642918.2647369>
- [37] Elizabeth Gerber and Maureen Carroll. 2012. The psychological experience of prototyping. *Design Studies* 33, 1 (2012), 64–84. <https://doi.org/10.1016/j.destud.2011.06.005>
- [38] Max Goldman, Greg Little, and Robert C. Miller. 2011. Collabode: Collaborative Coding in the Browser. In *Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering* (Waikiki, Honolulu, HI, USA) (CHASE '11). Association for Computing Machinery, New York, NY, USA, 65–68. <https://doi.org/10.1145/1984642.1984658>
- [39] Saul Greenberg and Chester Fitchett. 2001. Phidgets: Easy Development of Physical Interfaces through Physical Widgets. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology* (Orlando, Florida) (UIST '01). Association for Computing Machinery, New York, NY, USA, 209–218. <https://doi.org/10.1145/502348.502388>
- [40] P. Grimm, M. Haller, V. Paelke, S. Reinhold, C. Reimann, and R. Zauner. 2002. AMIRE - authoring mixed reality. In *The First IEEE International Workshop Augmented Reality Toolkit*. IEEE, Darmstadt, Germany, 2 pp.–. <https://doi.org/10.1109/ART.2002.1107008>
- [41] Anne Gutschmidt, Valentina Sauer, Kurt Sandkuhl, and Alexey Kashevnik. 2019. Identifying HCI Patterns for the Support of Participatory Enterprise Modeling on Multi-touch Tables. In *The Practice of Enterprise Modeling*, Jaap Gordijn, Wided Guedria, and Henderik A. Proper (Eds.). Springer International Publishing, Cham, 118–133.
- [42] Alastair Hampshire, Hartmut Seichter, Raphaël Grasset, and Mark Billinghurst. 2006. Augmented Reality Authoring: Generic Context from Programmer to Designer. In *Proceedings of the 18th Australia Conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments* (Sydney, Australia) (OZCHI '06). Association for Computing Machinery, New York, NY, USA, 409–412. <https://doi.org/10.1145/1228175.1228259>
- [43] M. Haringer and H. T. Regenbrecht. 2002. A pragmatic approach to augmented reality authoring. In *Proceedings. International Symposium on Mixed and Augmented Reality*. IEEE, Darmstadt, Germany, 237–245. <https://doi.org/10.1109/ISMAR.2002.1115093>
- [44] Stephanie Houde and Charles Hill. 1997. Chapter 16 - What do Prototypes Prototype? In *Handbook of Human-Computer Interaction (Second Edition)* (second edition ed.), Marting G. Helander, Thomas K. Landauer, and Prasad V. Prabhu (Eds.). North-Holland, Amsterdam, 367–381. <https://doi.org/10.1016/B978-044481862-1.50082-0>
- [45] ISO/TC 159/SC 4 Ergonomics of human-system interaction. 2019. *ISO 9241-210:2019 Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems* (2 ed.). Standard. International Organization for Standardization, Geneva, Switzerland.
- [46] Pamela Jennings. 2004. Reflections on interdisciplinary team development for the design of a platform for computer supported collaborative play. *Digital Creativity* 15, 4 (2004), 209–222. <https://doi.org/10.1080/1462626048520182arXiv:https://doi.org/10.1080/1462626048520182>
- [47] Andrew J. Ko, Robin Abraham, Laura Beckwith, Alan Blackwell, Margaret Burnett, Martin Erwig, Chris Scaffidi, Joseph Lawrence, Henry Lieberman, Brad Myers, Mary Beth Rosson, Gregg Rothermel, Mary Shaw, and Susan Wiedenbeck. 2011.

- The State of the Art in End-User Software Engineering. *ACM Comput. Surv.* 43, 3, Article 21 (April 2011), 44 pages. <https://doi.org/10.1145/1922649.1922658>
- [48] David Ledo, Steven Houben, Jo Vermeulen, Nicolai Marquardt, Lora Oehlberg, and Saul Greenberg. 2018. Evaluation Strategies for HCI Toolkit Research. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (CHI '18). Association for Computing Machinery, New York, NY, USA, 1–17. <https://doi.org/10.1145/3173574.3173610>
- [49] Gun A. Lee and Gerard J. Kim. 2009. Immersive authoring of Tangible Augmented Reality content: A user study. *Journal of Visual Languages & Computing* 20, 2 (2009), 61–79. <https://doi.org/10.1016/j.jvlc.2008.07.001>
- [50] Gun A. Lee, Gerard J. Kim, and Mark Billinghurst. 2005. Immersive Authoring: What You eXperience Is What You Get (WYXIWYG). *Commun. ACM* 48, 7 (July 2005), 76–81. <https://doi.org/10.1145/1070838.1070840>
- [51] Gun A. Lee, Claudia Nelles, Mark Billinghurst, and Gerard Jounghyun Kim. 2004. Immersive Authoring of Tangible Augmented Reality Applications. In *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR '04)*. IEEE, Washington, DC, USA, 172–181. <https://doi.org/10.1109/ISMAR.2004.34>
- [52] Germán Leiva, Cuong Nguyen, Rubaiat Habib Kazi, and Paul Asente. 2020. Pronto: Rapid Augmented Reality Video Prototyping Using Sketches and Enaction. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3313831.3376160>
- [53] Lucidchart Content Team. 2020. How to Make a User Flow Diagram. Retrieved September 13, 2020 from <https://www.lucidchart.com/blog/how-to-make-a-user-flow-diagram>.
- [54] Blair MacIntyre, Maribeth Gandy, Steven Dow, and Jay David Bolter. 2004. DART: A Toolkit for Rapid Design Exploration of Augmented Reality Experiences. In *Proceedings of the 17th annual ACM Symposium on User Interface Software and Technology* (Santa Fe, NM, USA) (UIST '04). Association for Computing Machinery, New York, NY, USA, 197–206. <https://doi.org/10.1145/1029632.1029669>
- [55] Asa MacWilliams, Thomas Reicher, Gudrun Klinker, and Bernd Bruegge. 2004. Design Patterns for Augmented Reality Systems. *Mixer* 4 (2004), 1–8.
- [56] Michael McCurdy, Christopher Connors, Guy Pyrzak, Bob Kanefsky, and Alonso Vera. 2006. Breaking the Fidelity Barrier: An Examination of Our Current Characterization of Prototypes and an Example of a Mixed-Fidelity Success. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Québec, Canada) (CHI '06). Association for Computing Machinery, New York, NY, USA, 1233–1242. <https://doi.org/10.1145/1124772.1124959>
- [57] Timothy McKenzie, Miguel Morales Trujillo, and Simon Hoermann. 2019. Software Engineering Practices and Methods in the Game Development Industry. In *CHI PLAY '19 Extended Abstracts: Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts* (Barcelona, Spain) (CHI PLAY '19 Extended Abstracts). Association for Computing Machinery, New York, NY, USA, 181–193. <https://doi.org/10.1145/3341215.3354647>
- [58] Jessica Menold, Kathryn Jablow, and Timothy Simpson. 2017. Prototype for X (PFx): A holistic framework for structuring prototyping methods to support engineering design. *Design Studies* 50 (2017), 70–112. <https://doi.org/10.1016/j.destud.2017.03.001>
- [59] Juergen Musil, Angelika Schweda, Dietmar Winkler, and Stefan Biffl. 2010. Improving Video Game Development: Facilitating Heterogeneous Team Collaboration through Flexible Software Processes. In *Systems, Software and Services Process Improvement*, Andreas Riel, Rory O'Connor, Serge Tichkiewitch, and Richard Messnarz (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 83–94.
- [60] Michael Nebeling, Katy Lewis, Yu-Cheng Chang, Lihan Zhu, Michelle Chung, Piaoyang Wang, and Janet Nebeling. 2020. XRDirector: A Role-Based Collaborative Immersive Authoring System. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3313831.3376637>
- [61] Michael Nebeling, Janet Nebeling, Ao Yu, and Rob Rumble. 2018. ProtoAR: Rapid Physical-Digital Prototyping of Mobile Augmented Reality Applications. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3173574.3173927>
- [62] Michael Nebeling and Maximilian Speicher. 2018. The Trouble with Augmented Reality/Virtual Reality Authoring Tools. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE, Washington, DC, USA, 333–337. <https://doi.org/10.1109/ISMAR-Adjunct.2018.00098>
- [63] D. L. Parnas. 1972. On the criteria to be used in decomposing systems into modules. *Commun. ACM* 15, 12 (Dec. 1972), 1053–1058. <https://doi.org/10.1145/361598.361623>
- [64] Perkins Coie LLP. 2020. 2020 Augmented And Virtual Reality Survey Report. Retrieved September 4, 2020 from <https://www.perkinscoie.com/images/content/2/3/v4/231654/2020-AR-VR-Survey-v3.pdf>.
- [65] Jeanne M. Pickering and Rebecca E. Grinter. 1995. Software engineering and CSCW: A common research ground. In *Software Engineering and Human-Computer Interaction (Lecture Notes in Computer Science)*, Richard N. Taylor and Joëlle Coutaz (Eds.). Springer, Berlin, Heidelberg, 241–250. <https://doi.org/10.1007/BFb0035820>
- [66] Wayne Piekarski and Bruce H. Thomas. 2002. The tinmith system—demonstrating new techniques for mobile augmented reality modelling. *Journal of Research and Practice in Information Technology* 34, 2 (2002), 82–97. Publisher: Citeseer.
- [67] René Reiners. 2014. *An Evolving Pattern Library for Collaborative Project Documentation*. Shaker, Aachen, Aachen, GER.
- [68] Holger Rhinow, Eva Köppen, and Christoph Meinel. 2012. Design Prototypes as Boundary Objects in Innovation Processes. In *Proceedings of the 2012 International Conference on Design Research Society (DRS 2012)* (Bangkok, Thailand), Vol. 4. Design Research Society, London, UK, 1581–1590.
- [69] Horst W. J. Rittel. 1984. Second-generation design methods. In *Developments in design methodology*. John Wiley & Sons, New York, Chichester, New York, Brisbane, Toronto, Singapore, Chapter 5.2, 317–327.
- [70] Dieter Schmalstieg, Anton Fuhrmann, Gerd Hesina, Zsolt Szalavári, L. Miguel Encarnação, Michael Gervautz, and Werner Purgathofer. 2002. The Studierstube Augmented Reality Project. *Presence: Teleoperators and Virtual Environments* 11, 1 (2002), 33–54. <https://doi.org/10.1162/105474602317343640>
- [71] Till Schümmer and Stephan Lukosch. 2007. *Patterns for Computer-Mediated Interaction*. John Wiley & Sons, West Sussex, England, 600 pages.
- [72] Maximilian Speicher, Brian D. Hall, Ao Yu, Bowen Zhang, Haihua Zhang, Janet Nebeling, and Michael Nebeling. 2018. XD-AR: Challenges and Opportunities in Cross-Device Augmented Reality Application Development. *Proc. ACM Hum.-Comput. Interact.* 2, EICS, Article 7 (June 2018), 24 pages. <https://doi.org/10.1145/3229089>
- [73] Maximilian Speicher and Michael Nebeling. 2018. GestureWiz: A Human-Powered Gesture Design Environment for User Interface Prototypes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (CHI '18). ACM, New York, NY, USA, Article 107, 11 pages. <https://doi.org/10.1145/3173574.3173681>
- [74] Susan Leigh Star and James R. Griesemer. 1989. Institutional Ecology, “Translations” and Boundary Objects: Amateurs and Professionals in Berkeley’s Museum of Vertebrate Zoology, 1907–39. *Social Studies of Science* 19, 3 (1989), 387–420. <https://doi.org/10.1177/030631289019003001>
- [75] Anselm Strauss and Juliet Corbin. 1990. *Basics of qualitative research*. Sage publications, United States.
- [76] The Ghost Howls. 2019. Augmented reality is reaching a mature state according to Gartner. Retrieved September 4, 2020 from <https://skarredghost.com/2019/09/04/augmented-reality-mature-gartner/>.
- [77] Ben Thompson, Laura Levy, Amelia Lambeth, David Byrd, Joelle Alcaindinho, Iulian Radu, and Maribeth Gandy. 2016. Participatory Design of STEM Education AR Experiences for Heterogeneous Student Groups: Exploring Dimensions of Tangibility, Simulation, and Interaction. In *2016 IEEE International Symposium on Mixed and Augmented Reality Adjunct Proceedings* (Merida, Mexico). IEEE, Washington, DC, USA, 53–58. <https://doi.org/10.1109/ISMAR-Adjunct.2016.0038>
- [78] Mark J. Weal, Eva Hornecker, Don G. Cruickshank, Danis T. Michaelides, David E. Millard, John Halloran, David C. De Roure, and Geraldine Fitzpatrick. 2006. Requirements for In-Situ Authoring of Location Based Experiences. In *MobileHCI*. Association for Computing Machinery, Helsinki, Finland, 121–128. <http://eprints.ecs.soton.ac.uk/12589/1/fp011-weal.pdf>
- [79] Richard Wetzel, Lisa Blum, Audrius Jurgelionis, and Leif Oppermann. 2012. Shapes, Marbles and Pebbles: Template-Based Content Creation for Location-Based Games. In *Proceedings of the IADIS International Conferences Interfaces and Human Computer Interaction 2012 and Game and Entertainment Technologies*. International association for development of the information society, Lisbon, 11–20.
- [80] Richard Wetzel, Tom Rodden, and Steve Benford. 2017. Developing ideation cards for mixed reality game design. *Transactions of the Digital Games Research Association* 3, 2 (2017), 175–211.
- [81] Y. Xu, E. Barba, I. Radu, M. Gandy, R. Shemaka, B. Schrank, B. MacIntyre, and T. Tseng. 2011. Pre-patterns for designing embodied interactions in handheld augmented reality games. In *2011 IEEE International Symposium on Mixed and Augmented Reality - Arts, Media, and Humanities* (Basel, Switzerland). IEEE, Washington, DC, USA, 19–28.