AI-based Mobile App Prototyping: Status Quo, Perspectives and Preliminary Insights from Experimental Case Studies

Stephan Böhm and Stefan Graser

CAEBUS Center of Advanced E-Business Studies
RheinMain University of Applied Sciences
Wiesbaden, Germany
e-mail: {stephan.boehm, stefan.graser}@hs-rm.de

***Abstract*—The market for mobile applications is characterized by a large number of applications that are often developed by smaller companies and are distributed free of charge or at low prices via a few central app store platforms. This leads to high innovation rates and high competition. Against this background, a strong customer focus, rapid development, and cost-efficient user-centered design are particularly important for successful mobile apps. For these reasons, prototyping is of great importance in app development. Various mobile app prototyping tools have emerged in recent years, ranging from simple wireframes to high-fidelity prototypes with interfaces for implementing the designed apps. Recent advances in the field of Generative Artificial Intelligence (AI) also offer a wide range of possibilities for assisting and automating app prototyping. Three approaches can be distinguished here: indirect guidance and assistance in prototyping, AI plug-ins as an extension of existing prototyping tools, and innovative prototyping solutions with integrated functionality based on Generative AI. This paper first describes application areas, status quo, and perspectives for using Generative AI in mobile app prototyping. This is followed by describing insights from experimental case studies with selected AI-based mobile app prototyping support. As a result, we demonstrate that simple mock-ups can be generated rapidly with the currently available AI support. While autogeneration of prototypes is more likely to be used for standard use cases, AI support is available for various steps in UX/UI design, which should increase the productivity of app prototyping as a whole soon.**

Keywords–Mobile app prototyping; AI-assisted prototyping; Generative AI; WebAR; ChatGPT.

I. INTRODUCTION

Mobile apps are application software for execution on mobile devices, such as smartphones, with which the functionality of the devices given by hardware and system software can be applied to solve user-specific problems. Typically, mobile apps consist of programs and data installed on the devices by the end users and thus form an important element of device personalization. The introduction of the first mobile app stores around 15 years ago significantly impacted the software market. Since then, a previously not imaginable number of software products have been established and created a new market. Users can select and easily install mobile apps from these markets for almost any purpose. The largest number of mobile apps is available for the Android mobile operating system from Google and the iOS from Apple. As of October 2023, according to [1], nearly 3.8 million such mobile apps were available for users in the Google Play Store and about 1.8 million in the Apple App Store. Many mobile apps are developed by small companies and are offered free of charge, financed by ad revenues, or offered at low prices. This leads to high competition and the need for developers to bring mobile apps into the market quickly, cost-effectively, and closely aligned with user requirements.

For the aforementioned reasons, rapid prototyping, Scrum, or user-centered design (UCD) approaches are very common in mobile app development [2]–[4]. All of these approaches typically start with a phase in which the app idea and basic features are defined by experts and documented as initial requirements. Moreover, in a UCD process, as shown in Figure 1, an attempt is made to involve users in the development process as early as possible to obtain direct feedback about their requirements and preferences [5]. Prototypes are the basis for obtaining this feedback and represent an unfinished state of development of the app concept. They are used to gather user feedback and adapt the prototype to the users' requirements in an iterative process. Since the introduction of mobile apps, more powerful tools for prototyping have been developed. These tools support mobile app designers and developers in transforming their ideas and concepts into prototypes, working on them collaboratively, and presenting them to test users.

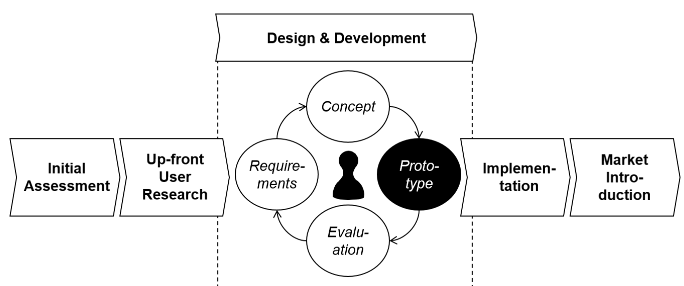

Figure 1. Simplified User Centered Design Process [3].

The fundamental problem of prototyping is to generate demonstrable archetypes from ideas, concepts, and user feedback. This task has typically been performed by screen designers, User Experience (UX) engineers, and app developers. However, prototyping follows experiences and recurring design patterns [6]. Thus, it provides a field of application for supporting and automating activities by new, content-generating forms of artificial intelligence. Such AI-based prototyping support has only recently become available in marketable solutions. This paper will describe the application fields and development status of such AI-based prototyping in an introductory way. Additionally, some first experiences in experimental case studies will be described.

Against this background, the paper is structured as follows: Section 2 gives a brief overview of existing research on the use of AI in mobile app prototyping and formulates the research objectives of this contribution. In the following Section 3, the status quo and perspectives of AI support in mobile app prototyping are discussed. For this purpose, the subject of mobile app prototyping will be specified before the application fields and emerging forms of implementation of AI-based prototyping support are outlined. Section 4 then describes initial experiences and insights from three experimental case studies of AI-based mobile app prototyping support before summarizing the conclusions and implications for practice in Section 5. The paper concludes with a brief outlook on future research topics and needs in Section 6.

II. Related Research and Research Objectives

With the advent of more complex Graphical User Interfaces (GUI) for the web and mobile apps, researchers have been trying to support the laborious prototyping process. As early as 2012, Segura et al. [7] presented the pen-based prototyping tool *UISKEI* for the design of websites, which can recognize certain user interface elements based on rough hand sketches drawn by a designer. An approach to transforming a pixel-based screenshot of a GUI design for mobile apps and web-based technologies into code was presented by Beltramelli [8]. Their *pix2code* approach used machine learning technologies based on convolutional and recurrent neural networks. For graphical user interface designs represented by a single screen, an accuracy of over 77 percent could be demonstrated. Moran et al. [9] present *ReDraw*, a more comprehensive approach for Android combining computer vision, machine learning, and software repository mining to automate prototyping by accurately detecting, classifying, and assembling GUI components. Their approach classified GUI components with a high accuracy of over 90 percent. *ReDraw* generated prototypes close to the mock-ups and a reasonable code structure. Kolthoff et al. [10] proposed *RaWi*, a data-driven GUI prototyping approach. The approach supports Natural Language (NL) searches in a large-scale GUI repository for mobile apps. *RaWi* ranks GUIs from the repository based on advanced machine learning methods (BERT-based LTR models) and provides matches as partly editable GUI screens to support interactive prototyping. Besides research, the potential of AI is also being recognized by providers of prototyping tools. Especially after the introduction of ChatGPT [11] and the increasing popularity of Generative AI, prototyping solutions are now on the market that promise easy prototyping using AI support and highlight the integration of AI or "Powered by AI" in marketing [12]–[14].

In this paper, we can only present selected literature on the state of research. For a detailed description of the state of research, especially on research about GUI and program code retrieval and GUI prototyping in mobile apps, we refer to [10]. However, it can be said that there are already comprehensive approaches in the literature to support the process of visual prototyping for the GUI design of interactive applications, which can be applied to mobile apps or have been developed specifically for this type of software. However, these approaches are mainly based on converting sketches into GUI designs, GUI images into program code, or identifying suitable GUI designs from a repository based on natural language queries. Generating visual prototypes using Generative AI is still an emerging field of research. Against this background, this paper addresses three research questions: (1) In which areas of mobile app prototyping can AI procedures be used, and how can they support the prototyping process? (2) In what form is AI support for mobile app prototyping currently available? (3) What results can be achieved in these areas using AI for exemplary case studies? The paper thus aims to explore the emerging field of AI-based mobile app prototyping and, above all, to generate insights for practice and identify research needs for the future.

III. Mobile App Prototyping and Potential for AI-Assistance

In the following, the process steps and tasks of mobile app prototyping will be described in more detail as a foundation for a more structured discussion of the application potential for AI in mobile app prototyping in the following sections.

A. *Mobile App Prototyping*

The term prototype comes from the Greek [protos (= the first) and typos (= archetype)] and generally refers to a sample, model, preliminary product, or, more generally, something that is still unfinished [5]. According to Sommerville [15, p. 45], the term prototype in software development refers to "an initial version of a software system that is used to demonstrate concepts, try out design options, and find out more about the problem and its possible solutions."

The basis of mobile app prototypes is usually a significantly shortened requirements elicitation phase compared to conventional software development. In UCD, the initial prototype is defined based on the team's expertise and some limited user research [5]. This phase can use a design-thinking approach and involve a concept formulation phase with semi-structured interviews to gather insights and identify significant elements for the app [16]. These insights are used to formulate the concept for the app's prototype. The prototype is intended to make the mobile app concept and the basic design features and functionalities understandable for test users. In several iterations with feedback loops, the prototype is presented to test users from the target group. The feedback is then evaluated and used to improve the prototype. This process is applied to optimize the design of the prototype in several iterations and to align it as well as possible with the users' requirements.

Figure 2 depicts the different prototyping stages based on a given use case that describes the intended interaction between the user and a system. For a first demonstration of app concepts, visual prototypes are typically used that already represent the intended screen or GUI designs of the concept but otherwise only simulate the system behavior. In contrast to functional prototypes, visual prototypes do not require any coding. However, the actual system behavior can only be tested when at least some system functions have already been implemented. In the case of a working prototype, this implementation is so advanced that an app concept can be tested in field tests in real-world application scenarios.

Typically, the (visual) prototype's abstraction level is adjusted in this iterative process. In the early phases, low-fidelity prototypes are still very different from the later end product.

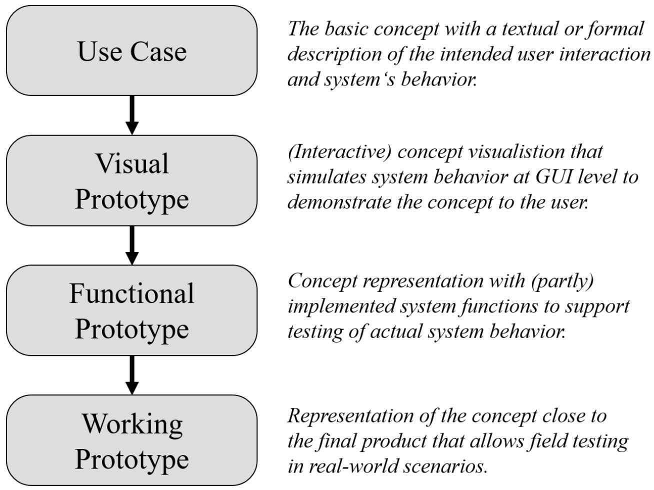

Figure 2. Use Case Definition and Prototypes.

These can be wireframes, for example, initially focusing on the mobile app’s basic functionality, screen design, and user flow. Such low-fidelity prototypes show placeholders and wildcards without sophisticated graphical visualization. In later iteration steps, graphical elements are used more intensively, and prototypes are presented to users with a higher fidelity or level of detail. The complexity is further enriched until the app design reflects the intended end product. However, the extent to which app functions are already implemented during this prototyping process can vary depending on the mobile app project and the development approach. Typically, a stable screen prototype is developed first as part of the low and high-fidelity prototyping before the actual implementation and coding of functional components is started. This has the advantage that changes can still be implemented quickly and without unnecessary programming costs.

The prototyping process for mobile applications can be simplified into the following basic steps. Overall, these steps describe prototyping in the broader sense. As mentioned, this understanding of using prototypes to involve users in the development process can be regarded as a UCD approach, as presented in Figure 1. In the narrower sense, prototyping describes creating and refining visual, functional, and working prototypes as presented in Figure 2. The respective phases are briefly described below. For a more detailed discussion, we refer to [5][17].

- *Idea Generation:* Formulation of the app idea and the basic app concept, including the core functionality and features, the intended user value, and the target group. Usage of first design drafts and sketches for screens. At this stage, however, the screen prototypes are usually still isolated, without a screen flow, and do not yet offer interaction options.
- *Basic User Research*: Initial user research and competitive analysis to better understand user requirements, usage context, and own value proposition. For this purpose, focus groups or interviews with potential users can be conducted, or their behavior can be investigated using methods, such as user shadowing or user diaries. However, the app concept is still textual or represented by simple sketches at this stage.
- *Concept Definition:* Description of personas to represent the target audience. Formulation of use cases and the user journey to describe the core interaction between users and the app.
- *Visual Prototyping:* The concept is transferred into interactive visual prototypes. This iterative process starts with simple hand sketches and wireframes in low fidelity. The prototypes are presented to test users to obtain feedback for iterative improvement. Moreover, the level of detail is continuously improved to high-fidelity prototypes.
- *Functional Prototyping:* A functional prototype is generated by programming code for the selected mobile operating systems according to the specifications and the visual prototype. For this purpose, agile software development methods like Scrum break down the app concept into smaller sub-components that can be implemented in a given development time or sprint. The initial functional prototype contains core functions and is progressively enriched with further functionalities.
- *Working Prototype:* The app has already been developed for at least one mobile operating system to the extent that test users can use it in a closed user group outside the app stores to evaluate core functionalities in field tests. Final bugs of the software will be fixed, and the functionality and user experience of the app will be further improved.
- *Continous Improvement:* Launch in the App Store and further improvement through continuous monitoring of user feedback and app reviews in planned updates and long-term versioning. Prototyping can be used to develop and pretest improvements and variants of the app or alternative implementations of sub-functionalities.

The following paper on the support potential of AI is mainly based on prototyping in the narrower sense. It focuses on visual and functional prototypes and how this process can be improved through the use of AI.

B. AI-based Mobile App Prototyping

The discussion of the potential application fields for (Generative) AI for mobile app prototyping is structured based on the phases described above and describes the status quo at the time of writing. The application areas were researched based on a search for case studies, reports, articles, and tool descriptions on AI-based app prototyping. However, as this is an emerging and very new field, there is no relevant, generally accepted term for AI-assisted or -supported prototyping that could be used for research. For a combination of the terms ”mobile”, ”app”, ”prototyping” with ”AI” or ”artificial intelligence” in Google Scholar, there was no search result when limited by a search in titles but about 20,700 search results with these terms anywhere in the article in October 2023. Therefore, only an unsystematic explorative search of general Internet sources (e.g., [18]–[22]) could be conducted and systematically summarized. In total, the following seven basic application areas were identified:

1) *Idea Generation and Concept Definition*
2) *Concept Customisation and Refinement*
3) *Design and Mock-up Generation*
4) *Sample Content and Design Variations*
5) *Testing and User Behavior Prediction*

6) *Evaluation of User Feedback and Refinements*
7) *Implementation and Text-to-Code Automation*

These application areas for AI in Prototyping will be described in detail in the following.

1) Idea Generation and Concept Definition: In this early phase of the prototyping process, the customer problem to be solved, the corresponding value proposition, and the central features of the app to be developed are defined and narrowed down. The result can be a product definition statement [5] in which the app concept is initially described in a basic and textual manner. Traditionally, creativity techniques are often used in this phase to develop and further elaborate ideas for app concepts. Generative AI, such as ChatGPT, can be used in this phase to develop ideas for app concepts based on a problem definition and propose central functionalities of such mobile applications. In this process, the Generative AI creates and varies learned concept patterns concerning a given framing. Thus, no completely new app ideas are to be expected. However, creativity technology methods also work with a variation and reconfiguration of known concept elements. By applying Generative AI, a large number of ideas or app variants can be created in a short time.

2) Concept Customisation and Refinement: App concepts are to be adapted to the specific needs of particular target groups. In addition, the app concept must be adapted based on user feedback in an interactive process. Already in early phases and based on textual concept descriptions, potential users can be confronted with the app idea in focus group sessions, for example. Generative AI can simplify the process of editing the concepts accordingly. Instead of manual revision ChatGPT can be asked in a prompt to generate suggestions for adapting the concept. In addition, adaptations to specific target groups can be requested. For this purpose, standardized or common target group definitions can frame the prompts. However, it is also possible to frame the characteristics of a target group independently or to use a previously defined persona for this purpose. Automating customizing for specific target groups or automated refinement based on user feedback can accelerate the maturation of the concept and make this process more efficient. However, experts' expertise or potential users' involvement is still required to evaluate the created variants.

3) Automated Design and Mock-up Generation: The phases described above were still mainly related to textual concepts and thus still represent preparatory phases of the actual mobile app prototyping. As soon as the concept has been sufficiently matured and the essential functionalities and features of the app have been determined, the concept must be transferred into a visual prototype. As described, low-fidelity prototypes are typically created first, e.g., based on wireframes. Generative AI can support transforming textual concepts into visual screen designs. Two steps have to be distinguished. First, the app's overall functionality must be suitably broken down into partial functionalities to derive an efficient user flow with the best possible user experience results. In the second step, the individual screens with their User Interface (UI) elements, such as buttons, input and output fields, labels, and graphic elements, must be designed to be as user-friendly as possible. The combined support of the two steps described above usually requires specialized AI-assisted mobile app prototyping solutions, as discussed below in the form of AI plug-ins for existing prototyping tools or integrated AI-based prototyping solutions (see Section IV).

4) Generation of Sample Content and Design Variations: As discussed earlier, UCD progressively refines the level of detail from low-fidelity prototypes with a still high level of abstraction and iteratively evolves towards high-fidelity prototypes, taking into account user feedback. While low-fidelity prototypes still work predominantly with placeholders, graphical elements, such as icons, images, and texts for labels and descriptions are required later when turning to high-fidelity prototypes. In conventional app design, this is often done by using free or paid content from external sources and providers. In the case of textual content, so-called dummy texts are often used in early phases to fill the text areas on the screens. This text then serves as a pure placeholder and has no meaning related to the app. Generative AI can be used to create prototype content very efficiently in this phase, and it can already be aligned and adapted to the specific app. This not only eliminates the sometimes time-consuming research of corresponding resources but can also save licensing costs for the use of pre-produced third-party content. In addition, variants of corresponding content can be easily generated for user testing. With this form of AI support, it is essential to note that the possibility of directly generating high-fidelity prototypes does not make the use of low-fidelity prototypes obsolete. The high degree of abstraction in low-fidelity prototypes is explicitly used in UCD to initially direct the users' attention and feedback to the essential core functionalities of the app to be developed. Working with a high level of detail too early can distract from the app's core concept and increase the effort –even when using AI– to adapt the prototypes.

5) Automated Testing and User Behaviour Prediction: Once the first screens of a visual mobile app prototype have been created and linked in a screen flow so that user interaction is possible, testing can be conducted to receive user feedback. For this purpose, test scenarios must be worked out, and briefings and tasks for selected users of the potential target group must be created. User feedback is collected and evaluated within the corresponding usability and user experience testing frameworks. For corresponding test procedures, such as A/B testing, it may also be necessary to systematically vary certain design elements of the mobile application to identify the best variant accepted by the users. Such test procedures follow typical processes in which the use of Generative AI can support the preparation phase and the creation of corresponding test scenarios and briefings. Moreover, there are efficiency advantages if test contents and variants can be generated more quickly and automatically, or at least if the AI assists the test managers in generating corresponding materials. In addition, approaches already exist that enable user behavior prediction based on screen designs [23]. The corresponding AI models were trained with the attention distributions or problem areas of example designs. They can then predict corresponding user behavior without testing with real users. However, the precision of such methods and the extent to which such AI support can save user tests even for non-standard designs remains to be further investigated.

6) Automated Evaluation of User Feedback and Refinements: Large amounts of data can accumulate when testing mobile app prototypes. This is the case, for example, when a so-called Thinking Aloud procedure is used, in which users express their experiences and opinions about the prototype in parallel to the testing which is recorded. For evaluation, such recordings are first transcribed from speech into text, for which AI-based methods have been used for some time in the context of Natural Language Processing (NLP). However, with multiple users and more complex test scenarios, large amounts of text can result, that can only be processed manually with a great effort. Here, solutions based on Large Language Models (LLM) can be used to summarise corresponding texts or to extract corresponding problem areas. Procedures based on logging user behavior (e.g., click histories) also generate large amounts of often unstructured or semi-structured data that can be processed and analyzed more efficiently with the help of AI-based procedures.

7) Implementation of Prototypes and Text-to-code Automation: The fields of application so far have been limited to visual prototypes that aim at AI support of the screen design. In such solutions, the actual app functionality is typically still simulated in that the user is forwarded to certain screens depending on a predefined user interaction. Functionalities beyond screen interaction are typically not implemented in visual mobile app prototypes. To implement the functions presented in visual prototypes, programming the app in a Software Development Environment (SDK) of the respective mobile operating system is necessary. In this context, AI-based text-to-code or visual-to-code procedures can be used [13]. This form of AI support can convert (partial) functionalities represented by text prompts or visual screen designs into code and thus into functional or working prototypes. In this phase, there is a transition from prototyping in the narrower sense to software development or technical implementation of the app. However, text-to-code automation is particularly relevant for prototyping when the added value of an app is difficult to simulate or represent through linked interactive ”screen dummies”. This is the case, for example, when the value for the users depends on interacting with the app in the real world. For example, with mobile Augmented Reality (AR) apps, in which a video stream generated by the device camera is superimposed on the real environment by computer-generated image content. At least part of the AR functionality of the app must already be functionally implemented to give users a realistic impression of the app. In such cases, text-to-code automation using AI could make it possible to create initial functional prototypes without relying on advanced programming skills or corresponding external resources.

IV. Experimental Case Studies on AI-Based Mobile App Prototyping

Most of the previously described support scenarios for mobile app prototyping using AI solutions are only emerging or available in pilots or beta versions in the market. A broader dissemination or research results on the possible efficiency gains with such approaches do not yet exist. Most Internet resources from where the application fields above had been derived had a rather theoretical approach to the AI potentials without referencing a concrete app development project. Against this background, the following section will report on initial experiences with some selected case studies on the application of AI support in mobile app prototyping. Three different forms of AI support are distinguished in the following:

- *AI-based Prototyping Plug-ins*. For some time now, various software solutions have been available to support low- or high-fidelity prototyping of mobile apps. There are solutions specifically for prototyping mobile applications or more universal solutions that, for example, support the creation of prototypes for websites and apps. Additional software or so-called plug-ins from third-party providers can extend some of these solutions. Recently, such AI-based plug-ins have emerged to support app prototyping.
- *Integrated AI-based Prototyping Solutions:* With the advent of Generative AI, the first novel prototyping solutions have become available on the market, highlighting integrated, more comprehensive AI support. Such integrated solutions can, for example, directly transfer textual concepts into visual prototypes. Moreover, further AI-based additional functionalities are embedded in these applications.
- *Prototyping Support by General Generative AI:* Corresponding solutions do not have any specific functionality for mobile app prototyping. However, due to their comprehensive training database and universal applicability, they can be used in different phases of mobile app prototyping. These solutions include the previously mentioned LLMs, such as ChatGPT.

In the following, initial experiences are presented for these three forms of AI support for prototyping based on three selected solutions. For AI-based prototyping plugins, the solution Figma [24] was chosen. Figma is a popular platform for mobile app prototyping characterized by open interfaces and a market with many plug-ins [25]. Appy Pie [12], Uizard [13], and Mockitt [14] were selected as examples of emerging integrated AI-based prototyping solutions emphasizing AI support in their marketing effort. In the area of prototyping support through general Generative AI, ChatGPT [11] was selected because this LLM, launched in the market in November 2022, has the most extensive user base and awareness in the area of Generative AI solutions [26].

Due to the space limitations of this paper, only AI assistance in prototyping for a significantly reduced app concept can be described here. An example from corporate training was defined as a brief use case. The app should explain the functionality of a randomly selected electronic component – a Residual Current-operated Circuit-Breaker (RCCB). This use case was chosen because such a specific solution is uncommon in the app market. Thus, AI solutions cannot simply reproduce existing concepts and design patterns. The following prompt was used for the prototype creation with a very short and simple description that does not list any more concrete functions or content.

Prompt with Use Case Description

Corporate training app for young trainees to learn the function and operation of a residual current circuit breaker via step-by-step illustrated instructions.

In addition to using the plug-ins and integrated solutions for a purely visual app prototype, we tested the application of general Generative AI for text-to-code automation. This AI support was used to create a functional mobile AR prototype, which is not feasible with standard mobile app prototyping tools focusing on GUI design. As mentioned, the following case studies can only allow for initial experiences and preliminary insights into the current state of the market solutions. Based on this, further systematic analyses and research are necessary. However, based on the results, the solutions' status can be presented, and further research needs can be derived.

A. AI-based Prototyping Plug-ins

Numerous prototyping tools are used in mobile app prototyping [27]. Popular app prototyping tools are, for example, Figma, Invision, Mockup, Marvel, and UXPin [28]. These include specialized prototyping tools as well as those that support not only apps but also other interactive applications, such as websites. As already mentioned, the widely used tool Figma, for which thousands of templates, plugins, and UI kits are available [25], will be used as an example of AI support through plugins in this paper. The plugins are available through the Figma community and are provided by independent developers. The plugins can be selected by users of the prototyping platform and integrated into the Figma prototyping environment. At the end of 2023, about 30 plug-ins were available via the community search after entering the terms ”artificial intelligence” and ”AI”. In addition to AI plug-ins for creating, improving, and evaluating prototypes, our search shows tools for task automation in Figma (e.g., naming Figma layers) or prototyping conversational AI applications. We refer to [29] for a more comprehensive description of Figma AI plug-ins for UI/UX design. The support of AI plugins relevant to mobile app prototyping ranges from AI-supported image and text generation to prompt-based creation of wireframes and prediction of user attention distribution on the screens. So far, however, only a few plug-ins support the autodesign or autogeneration of mock-ups based on a text prompt. We identified the Figma plug-in Wireframe Designer [30] that uses the ChatGPT-3.5 API to create mock-ups. Based on a given prompt, ChatGPT creates a design with suitable UI components, converts it into a machine-readable format, and transfers it to Figma for visualization [31]. The three right-hand screens in Figure 3 show the resulting design for the given prompt.

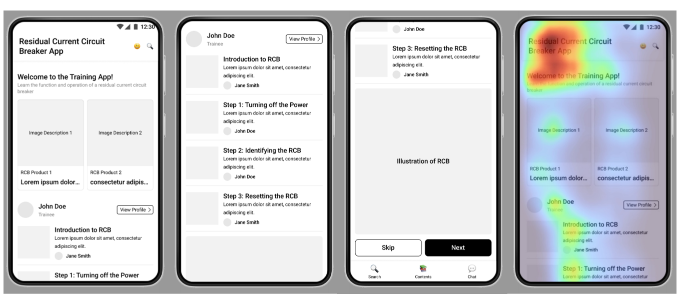

Figure 3. Example Screens and of AI-Generated Prototype with Figma and Plug-ins. [23][24][30].

The heatmap on the right side of Figure 3 was generated by the Attention Insight plugin. The red areas with ”warm” colors mark UI elements for which high user attention is predicted. This method substitutes user tests, which usually provide the basis for heatmaps to optimize the screen design. The plugin is based on training data from about 70,000 participants of real heatmap studies. It is supposed to match the results of actual eye-tracking heatmaps for general images with an accuracy of 92.5 percent [32].

Overall, it can be stated that the first working AI plug-ins for existing prototyping solutions are available. Prerequisites are open interfaces with which solutions, such as ChatGPT, can be connected and integrated. In addition, other emerging AI-based plug-ins support individual steps in prototyping up to the assessment of GUI designs.

B. Integrated AI-based Prototyping Solutions

Integrated AI-based solutions for mobile app prototyping are those tools that emphasize AI support in product positioning and offer more integrated AI support for mobile app prototyping. It is to be expected that established popular app prototyping tools will also increasingly integrate AI functions. However, in the second half of 2023, only a few providers were available on the market that positioned themselves with AI-supported prototyping. These include, as already mentioned, Uizard [13], Appy Pie [12], and Mockitt [14]. All of these tools offer an AI-based autodesign, i.e., the creation of an initial mobile app prototype based on the input of a text prompt. Appy Pie can quickly regenerate and process the proposed screen designs in the prototyping environment. Additionally, Uizard offers a wireframe and a screenshot scanner with which hand-drawn sketches and GUI screenshots can be converted into mock-ups. Moreover, design themes, images, and text content can be created and changed, and an attention focus can be predicted. Mockitt offers chat-based interactive support for prototyping. In addition to generating the prototype, flowcharts, mindmaps, and UI components, such as tables, charts, and texts, can be generated with AI support. In addition, further AI functions have been announced on their website [14].

Although the Mockitt application generated a GUI prototype for an app screen from the exemplary prompt, this contained text in Chinese and a random image. Thus, it was not further considered. The result of the Appy Pie tool is shown in Figure 4. The tool generated a straightforward navigation structure with some instructive text content and even added a suitable image. The prototype was simple but sufficient to be refined to gain initial user feedback.

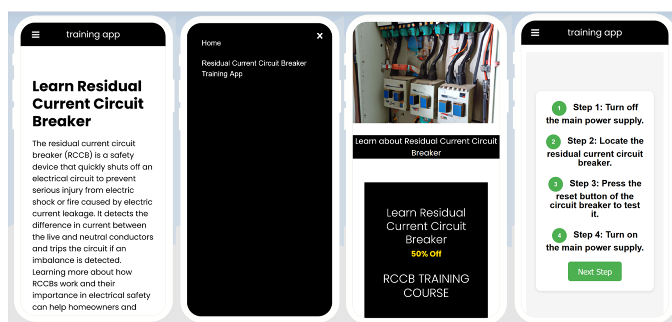

Figure 4. Example Screens of AI-Generated Prototype with Appy Pie [12].

A basic interactive app prototype for the given prompt could also be created with the tool Uizard. Compared to the Appy Pie prototype, more UI elements are added to the GUI design shown in Figure 5. This may be because information on the design style and the target group was also considered. Motivational elements, such as progress bars, and standard components like registration and login masks were integrated into the design. The design was made more colorful and detailed. However, the texts and the images were less specific regarding the defined use case of the RCCB training. Moreover, this prototyping tool did not create steps with instructions for the RCCB based on the prompt.

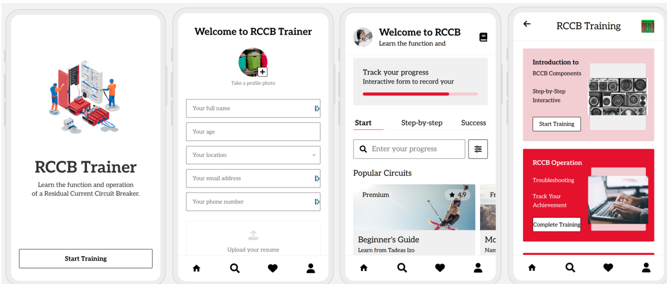

Figure 5. Example Screens of AI-Generated Prototype with Uizard [13].

These examples show that initial designs for mobile app prototypes can already be generated with the current status quo of AI-supported tools but are still in an early (beta) stage. Thus the functions for autogeneration of app designs are more likely to support idea generation for standard app concepts and smaller companies. These approaches cannot yet replace professional app designers. More promising for professional app prototyping are the AI-based productivity tools that allow content to be automatically filled, quickly changed, or even help to switch between different prototyping modes and formats (e.g., hand-drafted wireframes and high-fidelity GUI designs).

C. AI-based Prototyping Support

ChatGPT is a LLM-based chatbot developed by OpenAI and launched in November 2022. Currently, ChatGPT-4 is available for end users [11]. ChatGPT is a chatbot that processes input in natural language through so-called prompts and can be used universally. In addition to outputting text to answer questions, the LLM can generate code in different programming languages and for different software platforms. ChatGPT can generate code for web applications distributed via a web server to be executed in the browser of mobile devices. Due to the simplicity and platform independence of the resulting code and the direct execution without any necessary compilation, corresponding web technologies are well-suited for prototyping. The starting point for this use case was the existence of a 3D model (as a glb-file) of a generic RCCB and the functional specifications below. The research question here was whether a functional web-based AR (WebAR) prototype could be derived from an AI-supported text-to-code approach:

- Automated creation of step-by-step instructions.
- Implementation of the prototype as a WebAR solution (using the AR.js framework [33]).
- Simple button-based navigation between steps.
- Marker-based (Hiro marker) tracking of the RCCB 3D model.
- AR annotations of the 3D object through icon.

To have step-by-step training instructions as a basis for the WebAR app to be generated, the prompt used above was slightly adapted to provide a relevant framing on app prototyping for ChatGPT. A character limit was introduced to avoid lengthy text descriptions.

Adapted Prompt for ChatGPT-4

Prompt: An app designer wants to develop a corporate training app for young trainees to learn the function and operation of a residual current circuit breaker via step-by-step illustrated instructions. Please suggest text descriptions for five instructive steps, each 100 to 250 characters, that can be used as overlays of a WebAR prototype.

ChatGPT successfully processed the creation of step-by-step instructions for a brief training case on RCCBs. The text box below shows the first two of the steps generated. The explanations are very coherently structured and instructive. However, it gets already clear here that the context of an AR application has not been considered further. For example, the second step references a coil, which is only relevant for an AR scenario if internal components are shown. This had to be adjusted manually, as the 3D object available only shows the surface but no inner parts of this component.

ChatGPT-4 Output

1. *Introduction to RCCB:* Welcome to the world of Residual Current Circuit Breakers (RCCB). These devices detect an imbalance between live and neutral currents, ensuring your safety.
2. *Core Components:* RCCBs consist of a few key parts: the coil, the lever, and the trip mechanism. Familiarizing yourself with these will help you understand their operation.
3. ...

In a further step, an attempt was made to generate the code for a prototype WebAR app using ChatGPT. Such a functional prototype allows a realistic impression of the AR view on the smartphone. The technical requirements defined above formed the basis for a prompt. Since a marker-based solution and annotation with icons were requested, the marker to be used and the file names were predefined to avoid the need to subsequently replace corresponding placeholders in the code.

ChatGPT-4 Prompt for Text-to-Code Generation

Provide code to integrate the instructive steps into a WebAR application using the AR.js framework. The instructions for each step should be on a screen with the instructions in a text box on top and navigation buttons at the bottom. in Step 3, the users should be invited to scan a Hiro AR marker to show a 3D model (filename rccd.glb) of the RCCD. In Step 4, the 3D object should be annotated with an icon (filename icon.png).

ChatGPT generated linked HTML, JavaScript, and CSS files as output. The code was initially not executable on the server, mainly due to references to incompatible framework

 ISBN: 978-1-68558-100-8

versions, which could be explained by the outdated training data of ChatGPT. However, ChatGPT provided comprehensive support for debugging the code and isolating and eliminating the errors. In addition, many 3D parameters (position, scale, and rotation) had to be adjusted manually to correctly show the 3D object and the annotation on the smartphone screen. Adjusting the parameters and the respective export of the updated code to the server was time-consuming and not an efficient prototyping approach for AR applications. However, in the end, after several iterations of the code, a working WebAR prototype with limited functionality could be derived. Besides the support in debugging the code, the possibilities to improve the screen design by simple text input were helpful. For example, changes to the text size or the display of the text boxes and buttons were requested by prompts, and the code sections to be changed were output by ChatGPT. Figure 6 shows examples of the screens generated with AI assistance by ChatGPT.

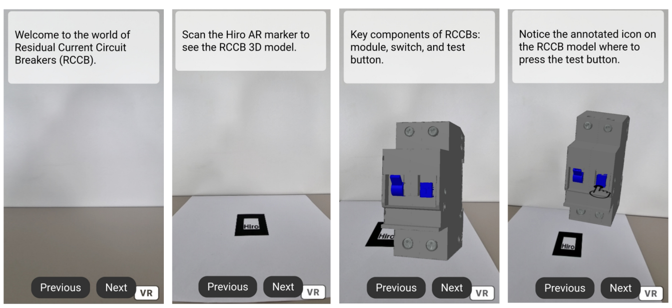

Figure 6. Example Screens of the WebAR Prototype Generated with ChatGPT AI Assistance [11].

In conclusion, it can be said that a simple ”functional” AR prototype can be developed using ChatGPT without specific programming knowledge. However, more comprehensive technical knowledge of suitable frameworks is necessary to specify the correct prompts. Also, no directly executable code can be expected, and more comprehensive debugging is necessary, although ChatGPT systematically supports approaches to debugging. The main problem, however, is the very complex positioning of 3D objects and corresponding annotations. ChatGPT was not able to establish relationships between the texts and the objects to be displayed and their properties. Adjusting these parameters iteratively with text prompts is a cumbersome process. Here, graphical AR development environments have clear advantages. On the positive side, however, it should be noted that with ChatGPT's coding assistance, one can familiarise quickly with new frameworks and programming languages and make rapid learning progress.

V. Conclusions and Implications

The applicability of currently available solutions in AI-based prototyping plug-ins, AI-based integrated prototyping solutions, and prototyping support by general Generative AI was investigated using the example of three selected experimental case studies. Almost all solutions used were able to generate corresponding visual or functional prototypes. However, autogeneration of mock-ups produces very basic designs that need further development. The use of general Generative AI requires extensive rework. It is also very complex to apply for AR prototypes due to the lack of a GUI editor or authoring environment if a text-to-code approach is chosen. However, such AI assistance was recognized positively regarding the learning effects imparted. Overall, the state of development is remarkable, considering that the breakthrough of Generative AI was less than a year ago. AI support will likely become widespread in prototyping tools in the next few years. However, the autogeneration of prototypes is probably more relevant for standard applications. More important for productive application in practice will be the AI support of repetitive prototyping sub-tasks, from which a considerable increase in the productivity of UX/UI design processes can be expected.

VI. Outlook on Future Research

This paper is a work-in-progress and presents only preliminary findings on a very new field of research. So far, hardly any papers deal with the possibilities and results of AI support in mobile app prototyping from a deployment perspective. In the past, most papers presented their technical approaches, with which partial aspects of mobile app prototyping can be supported utilizing novel approaches from the AI field. However, marketable AI applications are available, so in addition to the technology, the efficiency of process support and the quality of the results should become a stronger research focus. In the future, it should be investigated to what extent efficiency and productivity advantages can be achieved with such solutions. Of particular importance here is how AI-based prototyping support with general Generative AI such as ChatGPT will evolve, for which a new and more powerful version (ChatGPT-4V(ision) [34]) with enhanced image recognition and code conversion capabilities has just been announced at the time of publication of this article. Furthermore, there is a need for research into the user experience and acceptance of AI-generated solutions compared to conventionally generated and improved prototypes. Given the high number of existing apps in the stores, it is crucial to what extent successful apps with sufficient differentiation potential can be generated with AI or whether the support potential is somewhat limited to standard applications in which AI can increase the efficiency of development processes through a rapid reproduction of proven design patterns. In addition, the AI prototyping tools examined offer support for other formats, such as websites. Unlike apps, these interactive applications are typically characterized by more elaborate graphic designs and complex interaction paths. Thus, further research should investigate how AI prototyping tools provide suitable designs for websites and other interactive applications.

References

[1] 42matters, *Google Play vs the Apple App Store: App stats and trends*, 2023. [Online]. Available: https://42matters.com/stats [retrieved: 10/12/2023].

[2] C. Scharff and R. Verma, “Scrum to support mobile application development projects in a just-in-time learning context,” in *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering*, Y. Dittrich, C. de Souza, M. Korpela, H. Sharp, J. Singer, and H. Winshiers-Theophilus, Eds., New York, NY, USA: ACM, 2010, pp. 25–31, ISBN: 9781605589664. DOI: 10.1145/1833310.1833315.

[3] S. Böhm and B. Igler, “A tool-based approach for structuring feedback for user interface evaluations of mobile applications,” *Workshop on Prototyping to Support the Interaction Designing in Mobile Application Development (PID-MAD 2013) in Conjunction with Mobile HCI, Munich, Germany, August 27, 2013*,

2013. [Online]. Available: http://www.hciv.de/pidmad13/proc.html [retrieved: 10/12/2023].

[4] A. K. Almasri, "A proposed hybrid agile framework model for mobile applications development," *International Journal of Software Engineering & Applications*, vol. 7, no. 2, pp. 1–9, 2016, ISSN: 09762221. DOI: 10.5121/ijsea.2016.7201.

[5] S. Ginsburg, *Designing the iPhone User Experience: A User-Centered Approach to Sketching and Prototyping iPhone Apps*. Hoboken: Pearson Education, Limited, 2010, ISBN: 9780321699589.

[6] M. Jurisch, B. Igler, and S. Böhm, "PROFRAME: A prototyping framework for mobile enterprise applications," in *CENTRIC 2016*, L. Berntzen and S. Böhm, Eds., [Wilmington, DE, USA]: IARIA, 2016, pp. 7–10, ISBN: 978-1-61208-502-9.

[7] V. C. V. B. Segura, S. D. J. Barbosa, and F. P. Simões, "UISKEI," in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, G. Tortora, S. Levialdi, and M. Tucci, Eds., New York, NY, USA: ACM, 2012, pp. 18–25, ISBN: 9781450312875. DOI: 10.1145/2254556.2254564.

[8] T. Beltramelli, *Pix2code: Generating code from a graphical user interface screenshot*, May 22, 2017. [Online]. Available: http://arxiv.org/pdf/1705.07962v2.

[9] K. Moran, C. Bernal-Cardenas, M. Curcio, R. Bonett, and D. Poshyvanyk, "Machine learning-based prototyping of graphical user interfaces for mobile apps," *IEEE Transactions on Software Engineering*, vol. 46, no. 2, pp. 196–221, 2020, ISSN: 0098-5589. DOI: 10.1109/TSE.2018.2844788.

[10] K. Kolthoff, C. Bartelt, and S. P. Ponzetto, "Data-driven prototyping via natural-language-based gui retrieval," *Automated Software Engineering*, vol. 30, no. 1, 2023, ISSN: 0928-8910. DOI: 10.1007/s10515-023-00377-x.

[11] OpenAI, *Introducing ChatGPT*, 2023. [Online]. Available: https://openai.com/blog/chatgpt [retrieved: 10/12/2023].

[12] Appy Pie, *AI app generator to generate your app. describe your app in a sentence or two and the AI will help you build it.* 2023. [Online]. Available: https://www.appypie.com/ai-app-generator [retrieved: 10/09/2023].

[13] Uizard, *Design stunning mobile apps in minutes. the world's easiest-to-use design and ideation tool - powered by AI*, 2023. [Online]. Available: https://uizard.io/ [retrieved: 10/09/2023].

[14] Wondershare, *Power your prototyping with Mockitt AI*, 2023. [Online]. Available: https://mockitt.wondershare.com/ai-prototype-generator.html [retrieved: 10/09/2023].

[15] I. Sommerville, *Software engineering*, 9th ed., International ed. Boston: Pearson, 2011, ISBN: 978-0-13-703515-1.

[16] H. M. Isa, R. M. Jusoh, M. H. A. A. Kamal, F. S. M. Amin, and P. F. M. Tamyez, "Enriching user experience among senior citizens in the digital era: A design-thinking approach to constructing a prototype of a mobile application," *Journal of Advanced Research in Business and Management Studies*, vol. 29, no. 1, pp. 20–27, 2022. DOI: 10.37934/arbms.29.1.2027.

[17] B. Bähr, *Prototyping of User Interfaces for Mobile Applications* (T-Labs Series in Telecommunication Services). Cham: Springer International Publishing, Imprint, and Springer, 2017, ISBN: 978-3-319-53209-7.

[18] D. Lane, *How Generative AI will spawn amazing new app ideas*, 2023. [Online]. Available: https://twinsunsolutions.com/blog/how-generative-ai-will-spawn-amazing-new-app-ideas/ [retrieved: 10/12/2023].

[19] Lets Nurture, *Leveraging Generative AI in for seamless mobile app development*, 2023. [Online]. Available: https://www.letsnurture.com/blog/leveraging-generative-ai-in-for-seamless-mobile-app-development.html [retrieved: 10/12/2023].

[20] M. Weaser, *The promise and perils of Generative AI in app development*, 2023. [Online]. Available: https://www.cdotrends.com/story/18381/promise-and-perils-generative-ai-app-development [retrieved: 10/12/2023].

[21] R. Dinakar, *How to use Generative AI to make app testing easy?* 2023. [Online]. Available: https://www.pcloudy.com/blogs/how-to-use-generative-ai-to-make-app-testing-easy/ [retrieved: 10/12/2023].

[22] P. Parra Pennefather, *Creative Prototyping with Generative AI: Augmenting Creative Workflows with Generative AI*. Berkeley, CA: Apress L. P, 2023, ISBN: 9781484295793. [Online]. Available: http://ebookcentral.proquest.com/lib/hsrm/detail.action?docID=30670607.

[23] Attention Insight, *Improve design performance with pre-launch analytics*, 2023. [Online]. Available: https://www.figma.com/community/plugin/968765016617421513/attention-insight [retrieved: 10/10/2023].

[24] Figma, *Work together to build the best products*, 2023. [Online]. Available: https://www.figma.com/design-overview/ [retrieved: 10/10/2023].

[25] Figma, *Welcome to Figma community: Explore thousands of free and paid templates, plugins, and UI kits to kickstart your next big idea.* 2023. [Online]. Available: https://www.figma.com/community/plugins [retrieved: 10/11/2023].

[26] K. Hu, *ChatGPT sets record for fastest-growing user base - analyst note*, 2023. [Online]. Available: https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/ [retrieved: 10/12/2023].

[27] P. Chawla, *List of 28 mobile app design prototyping tools*, 2022. [Online]. Available: https://appinventiv.com/blog/top-mobile-app-prototyping-tools/amp/ [retrieved: 10/11/2023].

[28] H. Clark, *20 best mobile app prototyping tools for product teams in 2023*, 2023. [Online]. Available: https://theproductmanager.com/tools/best-mobile-app-prototyping-tools/ [retrieved: 10/11/2023].

[29] A. Chandak, *10 AI Figma plugins every UI/UX designer must try*, UX Planet, Ed., 2022. [Online]. Available: https://uxplanet.org/10-ai-figma-plugins-every-ui-ux-designer-must-try-d71a9acff1e [retrieved: 10/10/2023].

[30] C. Wu, *Wireframe designer: An AI-powered wireframe generator*, 2023. [Online]. Available: https://www.figma.com/community/plugin/1228969298040149016/wireframe-designer [retrieved: 10/10/2023].

[31] C. Wu, *Over this weekend, I developed a Figma plugin - Wireframe Designer: Twitter post*, 2023. [Online]. Available: https://twitter.com/WCMuu/status/1648164777884266498 [retrieved: 10/10/2023].

[32] Attention Insights, *Attention insights technology*, 2023. [Online]. Available: https://attentioninsight.com/technology/ [retrieved: 10/12/2023].

[33] J. Etienne and N. Carpignoli, *Ar.js - augmented reality on the web*, 2023. [Online]. Available: https://ar-js-org.github.io/AR.js-Docs/ [retrieved: 10/11/2023].

[34] OpenAI, *ChatGPT can now see, hear, and speak*, 2023. [Online]. Available: https://openai.com/blog/chatgpt-can-now-see-hear-and-speak [retrieved: 10/24/2023].