Prototyping as a Method in the Education of Software Developers and System Analysts

Vjeran Strahonja, Dijana Oreški
University of Zagreb, Faculty of Organization and Informatics, Varaždin, Croatia
vjeran.strahonja@foi.unizg.hr, dijana.oreski@foi.unizg.hr

Abstract The aim of this paper is to position prototyping as an approach in the education of software developers and system analysts. Based on the literature analysis, the basic approaches and prototyping methods used in the development of software components and systems were analyzed and their classification was made. The purpose of the classification is not comprehensiveness, but the definition of sets of approaches and methods whose applicability in higher education curricula will be determined in the continuation of the research. The current availability of different types (approaches and methods) of prototyping in the education of software developers and system analysts is investigated. The research was conducted on the sample of undergraduate university studies curricula at the Faculty of Organization and Informatics of the University of Zagreb. Research results were compared, discussed, consolidated, and the conclusions are drawn.

Keywords: prototype, software developers, business analysts, design thinking, educational process

University of Maribor Press
DOI https://doi.org/10.18690/um.fov.3.2023.76
ISBN 978-961-286-722-5

1 Introduction

The word **prototype** was created from the Greek "πρωτος" (protos) = first and "τύπος" (tipos) = impression, i.e. "πρωτοποτιν" (prototype) = primitive form. It began to be used in the beginnings of industrial engineering as a designation for the first copy, a sample of a product, which will later be produced in multiple copies, serially or mass-produced. From the user's point of view, it reflects all the useful characteristics of the product, its functionality, and its behaviour. It is the first copy of a component, product, or system, which is used for testing and improvement before it is finalized and manufactured.

Prototype and model are two related terms, and sometimes they are used as synonyms, although they are not.

A **model** is a simplified representation of an idea or object (component, product, system), usually at a smaller scale and in another medium (virtual, mathematical, graphic...). It serves as a means to understand, analyze or communicate some characteristics of what it represents.

A **prototype**, on the other hand, is a preliminary working version of a component, product or system. Unlike a model that is a representation of reality, a prototype is a real product that can be tested, applied, and used. It possesses the essential properties of the product.

In a non-technical context, for example in social sciences and research, the term prototype is sometimes used for an especially representative example of a given category, which is closer to a model. Design thinking and marketing know the prototyping of ideas, which is basically modeling (Reinecke, 2016).

Prototyping is a process where design teams create a prototype. Early examples of prototyping can be found in industrial design and engineering where prototypes were used to test and refine the ergonomics, functionality, aesthetic performance, and reliability of new products. Today, the use of rapid prototyping technologies, such as computer-aided design (CAD) and 3D printing based on CAD models make it easier and more efficient to create and test prototypes, which has led to increased use of prototyping in a wide range of fields.

Software prototypes and prototyping follow this basic idea, with certain specifics. Basically, every first release of a program is a prototype. Software development is an inherently prototypical process. In software development, a prototype is a rudimentary working edition of a program component or system that is made for demonstration purposes or as part of the development process. The software can be developed in phases. In doing so, analysis, design, and construction as key phases in the systems development life cycle (SDLC) are completed once, resulting in a software prototype. A prototype, as a basic version of the system is built, tested, and then reworked as necessary until an acceptable prototype is finally achieved. Such an approach was dominant in the first decades of computer development. Regardless of whether the development is applied through several stages or an agile methodological approach, today's software is developed mostly evolutionary (iterative, incremental), so that in each iteration some increment of functionality or some other property is improved or added to the application prototype that is being developed. Prototype development is evolutionary in nature, and prototyping is inherent in evolutionary development. Prototyping can enhance application development by allowing developers to quickly and easily test and iterate on different design ideas and functionality before committing to a final product. This can help to identify and resolve potential issues early on in the development process, resulting in a more acceptable and user-friendly final product. Additionally, prototyping can also help to facilitate communication and collaboration among team members (Lauff et.al., 2020), as well as with stakeholders and users, by providing a clear visual representation of the proposed application.

Considering the **purpose** of prototypes in the software development cycle, two basic types of prototypes are distinguished:

- **A throwaway prototype** is created during development as an animation and illustration of models and other project specifications. It does not evolve into a final product, but a real prototype of a component, product or system is created based on it.
- **An evolving prototype** - the first example of a program component or system that evolves through evolutionary development, in each increment the structure, functionality and behavior are corrected or supplemented.

There are several popular methods for prototyping in software development. **Code-based prototyping** involves creating code to get a working prototype of the software. In doing so, some tools and development environments are used that enable fast code creation. In the past, these were 4th generation languages (4GL) and code generators, and today they are low-code/no-code programming languages and platforms. **Interactive prototyping** involves creating a clickable, high-fidelity prototype that simulates the functionality of the final product. This can be used to test the usability of the interface and gather feedback from users. Common interactive prototyping tools include Figma (Staiano, 2022), Adobe XD and InVisio.

In a broader context, a kind of prototype is also those that is not fully working but reflects the user interface and well imitates how the user and the system will communicate. This type of prototyping can be called **model-based** prototyping because it is a prototype in a medium different from the component or system under development. **Paper prototyping** involves creating a rough, hand-drawn version of the user interface on paper, to discuss it with users to gather feedback. This method requires no tools and is quick and cheap. This method was used thirty years ago as part of the broader cognitive-walktrough design method and task-centered user interface design (Lewis, Rieman, 1993). Paper prototypes are sketches that are used to create code-based prototypes. **Wireframing** is creating a simple, low-fidelity representation of the interface. The method is similar to paper-based, but wireframing tools are used, such as Balsamiq, Silk (Silva et al., 2017), Axure, and Sketch.

The concept and methods of prototyping are mentioned in the context of business modeling and service design, where the result is actually a model, which deviates from the original definition of a prototype. However, if it is about the design of something abstract, such as a business model or a service, using the term prototyping makes sense if the final artifact is detailed to a level that can be applied. Such a method that simulates the functionality of a product or service without actually building it is **Wizard of Oz (WOZ) prototyping**, where a developer ("wizard") creates a rudimentary model of the final product, which is called a prototype. The user interacts with the prototype as if it were a real product, while the "wizard" behind the scenes performs the actions that the user thinks the product is doing. There is a whole series of similar methods that simulate the user experience (UX) by prototyping the behavior of the system in interaction with users, the user interface

or other touchpoints of an application or e-service. Examples of methods are: **Experience Prototype (**Buchenau, Suri, 2000), Mock-up, User Scenarios, Role Playing, Low-Fi Prototype (Svanaes, Seland, 2004) etc.

Application prototyping can enhance business model and service development by allowing stakeholders to test and experiment with different features and functionalities of applications, but also digital platforms on which the future business model or individual services are based, before committing to a final design. This can help to identify potential issues and opportunities early on, address some issues like sustainability and feasibility (Baldassarre et.al, 2020), and make adjustments as needed. Additionally, application prototyping can also help businesses to communicate and validate their ideas with potential customers and investors, by providing a tangible representation of the proposed application and the service built on it.

This paper aims to investigate the possible place and role of prototyping, as a method and approach, in the context of education of software developers and system analysts. Therefore, some research in this domain will be taken into account. The application of the prototype will be investigated on the example of undergraduate university studies curricula at the Faculty of Organization and Informatics of the University of Zagreb.

3 Related work

The literature review revealed various examples of students` prototyping activities at different courses and different study levels. Most of the research papers were focused on the students' and/or professionals' perceptions and usage of prototypes based on surveys and semi-structured interviews. However, Lauff, Kotys-Schwartz, and Rentschler (2017) consider that the use of prototypes by students has not been properly investigated in order to be explicitly understood and documented in the curriculum, even though prototyping activities are thought to be an important component of education.

Prototypes allow students to model elements of the finished product, helping them to answer questions with confidence and reduce risk and uncertainty throughout their projects (Yang, 2005).

According to a study from an academic environment, prototyping is directly linked to knowledge and objective learning. Prototypes serve as a catalyst for students' ideas, improve communication between project participants like students, professors, and potential users, and assist students in solving difficult challenges (Berglund and Leifer, 2013). So there is a need to determine the best professional prototype practices to turn them into instructional frameworks, syllabi and/or curricula to deliver insightful information in academic settings (Lauff, Kotys-Schwartz and Rentschler, 2017). In order to encourage students' prototyping outcomes and further enhance their design skills, there is a lack of a prototyping strategy that is tailored to design education demands (Lemons et.al, 2010).

Some academic settings used organized approaches and provide the basis for good practices. E.g. Berglund and Leifer (2013) study examined two masters-level engineering design project courses: Project-Based Engineering Design, Innovation, and Development at Stanford University and Integrated Product Development at KTH, Sweden. The comparison was performed by authors based on prototyping activities that were part of this course. However, the authors found significant shortcomings that need to be fixed in order to implement the approach on the strategic level (Berglund and Leifer, 2013).

Through the use of a quantitative approach based on surveys and questionnaires, another study seeks to compare the various perspectives and uses of prototypes between students and professionals (Lauff, Kotys-Schwartz, and Rentschler, 2017). This research contributes to the field of design education since it leads to various explicit prototyping instructions that emerge from opportunities that students didn't take advantage of during their prototyping activities

Various attempts have been made to analyze prototype activities in the literature by conducting investigations with engineering students. Research performed at Case Western Reserve University examined how well are engineering approaches (mostly prototyping) integrated into an improved classroom-based curriculum (Yang, 2012). Petrakis, Wodehouse and Hird (2019) investigated prototyping instances in projects of engineering design students. Their focus was on the purpose of prototyping activities. Results revealed students' need for more precise instructions and motivation. Researchers at Miami University examined the difficulties in integrating hardware prototyping platforms into the engineering curriculum from a pedagogical

perspective (Jamieson, 2015). Al-Masri, Kabu, and Dixith (2020) evaluated the level of integration of hardware prototyping platforms generally by looking at curriculum and instructional materials (such as course syllabi or outlines) utilizing freely available web educational resources. The analysis included 317 courses and a search for used platforms for hardware prototyping.

Furthermore, Jensen et al. (2002.) conclude how universities might integrate rapid prototyping technology into their design curricula. Werth et.al. (2020) reported the adaption of prototyping in two courses: one in the Bachelor program and one in the Master program. Their results revealed an increasing understanding of the given topics and indicated the concept of prototyping as one of the trendsetting features in modern education.

Previous studies have also looked at the usage of digital prototypes as a learning tool in a variety of engineering applications (Berglund, Zhou, and Martinsen, 2021). Although most of the papers focused on engineering education, there were attempts to develop undergraduate courses that integrate skills from engineering pedagogy (such as prototyping) with the skills-based instruction of entrepreneurship education (Barber et. al, 2020).

Generally, the use of prototypes by students has not yet been thoroughly investigated to develop instructional guidelines (Petrakis, Wodehouse, and Hird, 2021). Also, Al Masri (2018) considers it essential to include new technology in curriculum design to improve the learning experience for students. Mikkonen (2017) stated that the teacher should manage students' work, create a prototyping-positive attitude and incorporate prototyping into the curriculum. Future research suggestions provide motivation for our work.

4 Research methodology

The object of the research is the current availability of different types (approaches and methods) of prototyping in the education of software developers and system analysts. The sample on which the research was conducted is the curricula of undergraduate university studies at the Faculty of Organization and Informatics of the University of Zagreb. In doing so, a set of all courses on study courses and modules focused on software and database development (colloquially called

software development) and business aspects (colloquially called system analysis) were analyzed. Each of the authors of the paper independently analyzed the study programs in the syllabus of each individual course:

a)In the first review, courses were analyzed in which the application of prototyping is explicitly mentioned. It was also analyzed whether it was prototyping. At the same time, it is significant whether the students do a project whose result is a working prototype, or at least an interface prototype, the method and tool used to create the prototype, and what the learning outcomes of the course are. Some information was checked with teachers.

b) The second review included courses where prototyping is not explicitly mentioned, but the description of teaching activities and learning outcomes indicates that some method of prototyping is used. At the same time, a broader view of prototyping was applied, which includes paper-based, wireframe, and similar prototypes. These are mainly courses related to service design, user experience in interaction with the system, design thinking in the development of business models, etc. After the discussion and consolidation of the results, the results are presented in tabular form.

5 Research results

Out of a total of 65 courses (compulsory and optional) that make up the undergraduate curriculum, 9 of them mention the concept of prototypes or prototyping explicitly, while implicit prototyping is recognized in two courses (Table 1). For each course in which prototyping is used, the name of the course, a description of the application of prototyping in the course, the method and tool for prototyping, the prototyping activity (student project, lab, practical task, etc.), and the form of the prototype that is the result of prototyping (software, software and model, user interface, e-service, etc.).

Table 1: Prototyping courses

Course	Description	Method/tool	Activity	Resulting prototype
Courses that mention the concept of prototype				
Information Systems Development	Build a prototype application using the given development tools based on the information system model	Low code platform Oracle APEX	Software (Student project)	Code based
Development of Applications for Mobile and Smart Devices	Development of a mobile application supported by systems for version control and software maintenance	Visual Studio	Application sketches and wire models Software (Student project)	Model based + Code based
Software Products User Interfaces	Design and build a user interface based on given user requirements and technical specifications	Software tools for prototyping and creating user interfaces	User interface design (Practical Tasks)	Model based
Process Oriented Applications	Building a prototype of a process-oriented application	Bizagi - tool for modeling and automating business processes on a low-code development platform	Model + Application (Lab. + practical tasks)	Model based + Code based
Computer Games Development	Construct prototypes of different game types (design user interface, level, scene and characters...)	Environment for game development	UI design + Software (Student project)	Code based
Internet of Things Systems Development	Prototyping embedded devices and interoperable IOT system	Visual studio code	UI design + Software (Student project)	Code based
Embedded Systems Development	Develop a prototype of an embedded device using prototyping tools and techniques	Visual studio code	Software (Lab.)	Code based
Interactive Systems Development	Create different types of prototypes using appropriate methods and tools including multimedia elements	Figma	Model of External design and UI (Student Project)	Model-based
Computer Games Development	Construct game prototypes of different genres	Environment for game development	UI design + Software (Student Project)	Code based
Courses implicitly mention the concept of prototype				
Informatics Services Management	Creation of customer profiles, user stories, sketches and service prototype	Figma PROTO.IO, Marvel, Adobe xd..	(Student Project)	Model based
Development of software products	Plan, design, develop, test and integrate a software product	Visual Studio, Microsoft SQL Server, GitHub	(Student Project)	Code based

Most of the prototypes are code based (6 out of 11), three are model based and two are code and model based.

6 Conclusion

Prototyping is important for education of system analysts or software developers because it helps students to develop important skills such as problem-solving, critical thinking, creativity, communication, teamwork, etc.

Based on several curricular and extracurricular courses, we emphasize some best practices for prototyping in the education of software developers and system analysts:

1. Connect prototyping with theory and position it concerning other methods and approaches.
2. Define the project task, and provide a real-world case and a clear problem statement.
3. Provide a good development platform, tools, space in the cloud, etc. (in our case Oracle APEX and Oracle Cloud).
4. Encourage teamwork and collaboration.
5. Provide personal guidance and support, but also learning resources such as e-learning content, tutorials, templates, samples, videos, etc. (in our case Moodle courses and Oracle Academy resources).
6. Emphasize iterative development with feedback checks of results in a certain time frame.
7. Encourage students to present their prototypes to the class and get feedback from their peers and instructors.

Prototyping allows students to **learn by experimenting and doing**, apply the concepts they have learned in a hands-on practical way, and create tangible and functional projects, which can be used to showcase their abilities and knowledge to potential employers. Prototyping enables the creation of working software components and systems in less time, with less effort (***productive power***), and with minimal technical knowledge and skills. In university courses, the most common form of such prototyping is in system and software engineering, where students in teams design more complex applications, using the modeling and rapid prototyping.

This is especially encouraging for students who do not major in programming. Rapid prototyping allows the creation of physical or digital models of a design or product, which can be used as a tool for communication and collaboration among team members (***communication power***). The act of creating a prototype can help individuals and teams to better understand and visualize the problem (***cognitive power***) or opportunity they are trying to solve, leading to more effective problem-solving and decision-making. Additionally, the ability to quickly create and test multiple prototypes can foster design thinking and creativity by encouraging the exploration of different design concepts and ideas (***creative power***). The ability to see and interact with a prototype can also help to generate new ideas and insights and to identify potential issues or areas for improvement (***expressive power***) which is especially important in courses whose learning outcomes are related to UX. Rapid prototyping also allows for a more efficient and effective evaluation of different design concepts, as it allows users to interact with and provide feedback on a prototype more naturally and intuitively, which leads to a higher-quality of solution (***qualitative power***). In the end, such prototypes are self-documenting (***documentation power***).

Acknowledgements

The authors would like to acknowledge the support given by the European Commission through the Action Erasmus + Better Employability for Everyone with APEX (project ID 2021-1-SI01-KA220-HED-000032218), co-funded by the Erasmus+ programme of the European Union. The European Commission support for the production of this publication does not constitute endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

References

Al-Masri, E. (2018, October). Integrating hardware prototyping platforms into the classroom. In 2018 IEEE Frontiers in Education Conference (FIE) (pp. 1-4). IEEE.

Al-Masri, E., Kabu, S., & Dixith, P. (2020). Emerging Hardware Prototyping Technologies as Tools for Learning. IEEE Access, 8, 80207-80217.

Baldassarre, B., Konietzko, J., Brown, P., Calabretta, G., Bocken, N., Karpen, I. O., & Hultink, E. J. (2020). Addressing the design-implementation gap of sustainable business models by prototyping: A tool for planning and executing small-scale pilots. Journal of Cleaner Production, 255, 120295.

Barber, D., Madden, L., Mayo, D., & Agarwala, R. (2020). Two-course integration with student collaboration: Rapid prototyping and entrepreneurship. Entrepreneurship Education, 3, 57-76.

Berglund A. and Leifer L. "Why we Prototype ! An International Comparison of the Linkage between Embedded Knowledge and Objective Learning," Eng. Educ. 81, 2-15, DOI

10.11120/ened.2013.00004, 2013.Berglund A. and Leifer L. "Why we Prototype ! An International Comparison of the Linkage between Embedded Knowledge and Objective Learning," Eng. Educ. 81, 2-15, DOI 10.11120/ened.2013.00004, 2013.

Berglund, A., Zhou, Y. & Martinsen, M. (2021) An Assessment Review of Learning Performance when adopting Augmented Reality in Engineering Education. In proceedings: 8th Development conference for Swedish Engineering Education, Karlstad Nov 24-25

Buchenau, M., & Suri, J. F. (2000). Experience prototyping. In Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques (pp. 424-433).

Jamieson, P. "More missing the Boat — Arduino, Raspberry PI, and small prototyping boards and engineering education needs them." IEEE Frontiers in Education Conference, 2015, pp. 1-6.

Jensen, D., Randell, C., Feland, J., & Bowe, M. (2002, June). A study of rapid prototyping for use in undergraduate design education. In 2002 Annual Conference (pp. 7-113).

Lauff, C. A., Knight, D., Kotys-Schwartz, D., & Rentschler, M. E. (2020). The role of prototypes in communication between stakeholders. Design Studies, 66, 1-34.

Lauff C., Kotys-Schwartz D. and Rentschler M.E. "Perceptions of Prototypes: Pilot Study Comparing Students and Professionals," in ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC/CIE 2017 August 6-9, 2017, Cleveland, Ohio, USA, 2017.Lauff C., Kotys-Schwartz D. and Rentschler M.E. "Perceptions of Prototypes: Pilot Study Comparing Students and Professionals," in ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC/CIE 2017 August 6-9, 2017, Cleveland, Ohio, USA, 2017.

Lauff C., Kotys-Schwartz D. and Rentschler M.E. "Perceptions of Prototypes: Pilot Study Comparing Students and Professionals," in ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC/CIE 2017 August 6-9, 2017, Cleveland, Ohio, USA, 2017

Lemons G., Carberry A., Swan C. and Rogers C. "The benefits of model building in teaching engineering design," Des. Stud., vol. 31, no. 3, pp. 288–309, 2010.

Lewis, C., & Rieman, J. (1993). Task-centered user interface design. A practical introduction.

Mikkonen, J. 2017 On the teachers role in interactive prototyping. The Design Journal
20, S1212–S1223; doi:10.1080/14606925.2017.1352651.

Petrakis, K., Wodehouse, A., & Hird, A. (2019). A QUANTITATIVE ANALYSIS OF PROTOTYPE USE FOUND IN STUDENTS'DESIGN PROJECTS. In DS 95: Proceedings of the 21st International Conference on Engineering and Product Design Education (E&PDE 2019), University of Strathclyde, Glasgow. 12th-13th September 2019.

Petrakis, K., Wodehouse, A., & Hird, A. (2021). Physical prototyping rationale in design student projects: an analysis based on the concept of purposeful prototyping. Design Science, 7, e7.

Reinecke, S. (2016). What is it that design thinking and marketing management can learn from each other?. Design thinking for innovation: Research and practice, 151-162.

Silva, T. R., Hak, J. L., Winckler, M., & Nicolas, O. (2017). A comparative study of milestones for featuring GUI prototyping tools. Journal of Software Engineering and Applications, 10(6), 564-589.

Staiano, F. (2022). Designing and Prototyping Interfaces with Figma: Learn essential UX/UI design principles by creating interactive prototypes for mobile, tablet, and desktop. Packt Publishing Ltd.

Svanaes, D., & Seland, G. (2004, April). Putting the users center stage: role playing and low-fi prototyping enable end users to design mobile systems. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 479-486).

Werth, W., Faller, L., Liechtenecker, H., & Ungermanns, C. (2020, September). Low Cost Rapid Control Prototyping–a useful method in Control Engineering Education. In 2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO) (pp. 711-715). IEEE.

Yang M.C. "A study of prototypes, design activity, and design outcome," Des. Stud. 26 649e669 doi10.1016/j.destud.2005.04.005, vol. 26, pp. 649–669, 2005.

Yang, P. "A system engineering approach: Integrating technology into the classroom-based curriculum." International Conference on Systems and Informatics, 2012, pp. 1000-1004