Umetna inteligenca in razvoj uporabniških vmesnikov: Študija primera mobilne aplikacije X

Maksim Nikitashin
University of Primorska, Faculty of Management, Izolska vrata 2, 6000 Koper, Slovenia
max.nikitashin@gmail.com

Izvleček

Danes je umetna inteligenca postala najbolj priljubljen trend na vseh strokovnih področjih, vključno s področjem razvoja informacijskih rešitev. Njena uporaba prinaša številne koristi in priložnosti. Kljub temu pa še vedno obstaja veliko pomislekov o njeni uporabi, zlasti med strokovnim in vodstvenim kadrom. Zaradi tega je pomembno, da se na strukturiran način izpostavijo že omenjene koristi in izboljšave, najboljši način za to pa je študija primera. Ta članek predstavlja tovrstno študijo primera za prenovo uporabniškega vmesnika mobilne aplikacije. Najprej vsebuje pregled podobne študijske literature. Nato je opisan in analiziran primer tipične Android aplikacije, vključno s tehničnimi in poslovnimi podrobnostmi. V primerjavi z razvojem izvirne različice rešitve je uporaba umetne inteligence pomagala izboljšati razvojni proces v različnih pogledih. Na primer, s skoraj 50-odstotnim prihrankom časa na podlagi primerjave porabe časa enakih dejavnosti med razvojem prvotne in nove različice ter z izboljšavami splošne kakovosti končnega izdelka na podlagi primerjave števila zrušitev in težav, ugotovljenih v primerljivem času. Izkazalo se je, da so dosežene izboljšave zelo obetavne. Predstavitev primera je lahko koristna tako za tehnike, ki uvajajo orodja umetne inteligence v proces in tudi kot primer za upravičenost investicij v tovrstno integracijo za odločevalce.

Ključne besede: umetna inteligenca, mobilna aplikacija, razvoj programske opreme, testiranje programske opreme, projektni management, uporabniški vmesnik.

AI and UI development: a Case study of mobile application x

Abstract

Artificial intelligence (AI) has become the most popular trend in all professional spheres, including information solutions development. Its utilization brings a number of benefits and opportunities. However, there still many concerns about its use, especially among professionals and managers. It is therefore important to highlight the previously mentioned benefits and improvements in a structured manner, and the best way for this are case studies. The paper presents such a case study for the redesign of the user interface of a mobile application. First, it reviews the literature on similar studies. Then, a case study of a typical Android application is described and analyzed, including both technical and business-related details. Compared to the development of the original version of the solution, AI integration has helped to improve various aspects of the development process, such as design, development, and testing. For example, with nearly 50% time saved based on the comparison of the time spent on the same activities during the development of the original and the new versions as well as improvements in the general quality of the final product based on the comparison of the number of crashes and problems identified in the comparable time frame. The results and improvements attained appear to be are very promising. The presentation of the case study may be useful both for the technicians implementing AI tools into the process and as an example justifying investments into such an integration for decision makers.

Keywords: artificial intelligence, mobile application, software development, software testing, project management, user interface.

1 INTRODUCTION

Artificial Intelligence (AI) has been the most significant trend in almost all areas of human life in recent years [1]. It is reshaping nearly every professional sphere. Software development is not an exception here. At the moment, it is going through significant changes tackling almost all its sub spheres from project management to the development and deployment of the applications. These changes raise both opportunities and concerns. However, some employees working in the area, primarily non-technicians see only the potential risks and problems avoiding benefits and chances the AI opens. Especially this problem is faced in Slovenia [2]. For this reason, it is important to analyze and present successful cases of AI implementation in order to show potential benefits and justify potential investments in its implementation.

Into this paper the author would like to present the case study of the successful implementation of AI-supportive tools into a mobile application development project outstanding both the most important benefits and challenges of the process. In order to keep the business secret the application name is anonymized and some particular implementation details will not be described. However, still, the most important characteristics of the app and the AI integration will be described.

2 MOBILE APPLICATIONS DEVELOPMENT

2.1 Mobile applications

Two of the most popular types of modern software are web and mobile applications. The second are specific kind of software intended to run on mobile devices usually with a specific platform. The most popular among those platforms are Android and IOS sharing more than ninety-nine percent of the market. At the moment, the market of mobile applications is already enormously big and still continues to grow.

Those apps have a lot of specifics, advantages, and disadvantages. Among the first are nativeness in general, the device features integrations, and platform specificness. Advantages coming from such specifics are faster and more stable operation, possibilities of device and platform features integration, and better security compared to web applications. Disadvantages are primarily related to the limitation to the one platform and consequently much more difficult development and different operation systems adoption. At the moment, there exist some solutions trying to overcome those disadvantages – cross-platform frameworks, but most of them still have limitations with the device's native features integration and operating stability.

Most modern mobile applications have similar architecture of mobile user interface (UI), application programming interface (API), and local or remote server-based database. Optionally, there could be some local storage present (Figure 1).

User interfaces of Android applications can be programmed with Java or Kotlin language and XML and of IOS ones with Objective C or SWIFT. Typically, UI consists of some interactive views and some kind of navigation between them [5].

For API communication with the back-end and database, obviously, some of the standard communication protocols are utilized in order to ensure the highest effectiveness, stability, and security. Nowadays the most popular among them is REST [6]. Databases are warehouses for data of modern mobile applications. They can be either relational or non-relational also obviously called SQL and NoSQL accordingly [5].

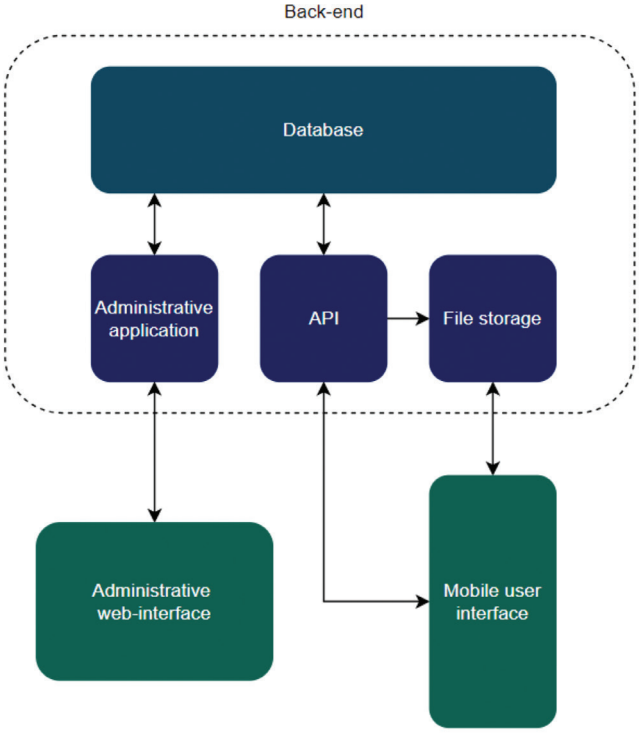

Figure 1: **Typical modern mobile application architecture**

2.2 Development process

Regardless of a model or a methodology selected for the development, its process usually includes the following steps:

1. Requirements definition,
2. Application design,
3. User interface prototyping and usability testing,
4. Code development,
5. Testing,
6. Publishing
7. Maintenance [5].

The most important in the context of this paper are 3., 4. and 5.

User interface design and user experience play very important roles in software development. It influences how users perceive an application and can significantly improve satisfaction with the app and conversion rate in the case of openly distributed ones [3]. The process usually includes first some prototype creation, based on the defined requirements. Then follows evaluation and testing. Later based on the previous step results corrections are made and the evaluation is repeated. The last step repeats until the appropriate version is not prepared [4]. Today there are a number of tools like Figma or Adobe XD intended to support the entire process and make it much more effective and easier.

Nowadays code development process commonly includes the utilization of integrated development environments (IDE) including for example code support and project management tools helping to easier and faster the process. Examples are Visual Studio Code or Android Studio [5]. Alongside, as code storage spaces repositories with integrated continuous integration, delivery, and deployment (CI/CD) pipelines are widely used. Such tools let save and share the code as far as to automatically build and test a new application version before deployment. Thus, they help to improve collaboration and ensure the maintenance of code quality and stability. Popular examples among the repositories are GitHub and GitLab and among the pipelines are Jenkins and GitHub Actions [7].

Testing is the last stage of the development before the deployment. It is intended for quality assurance and identification of potential problems before publishing. Usually, it is split into functional and non-functional. The first one is the most important in the case of mobile applications. Standardly it includes UI, API, and service functionalities testing [8]. Nowadays testing can be performed manually by human testers or with the help of automated testing tools. The second utilizes predefined automated tests to perform some actions and thus analyze if the application behaves as required and expected. Examples of such tools are UI Automator, Robotium, and Espresso [9].

3 AI SUPPORTIVE TOOLS

AI-supportive tools have found a wide range of applications both in the software development sphere in general and in mobile development in particular. In terms of UI/UX development, the most popular and well-known kinds of them are those for designing and monitoring, code support, and testing.

3.1 Design tools

AI tools have already and still continue to change the UI/UX design process. They are capable of analyzing such amounts of data, that were impossible for humans before, enabling the creation of personalized user experiences and reducing errors and overlooked opportunities amount.

Those changes influence the nature of the work of UI/UX designers. From primarily creative roles they are now playing more business-related strategic roles of interpreters of AI insights and decision-makers. Researches show that the easiness and speed of ideation and prototype creation have now increased significantly.

The widely utilized AI tools range in UI/UX design include for example sentimental analytical, predictive, and generative ones [10].

Among previously mentioned tools the most popular are for example Framer, Uizard, ChatGPT, and in the case of some other popular platforms custom AI-supportive plugins. Most of them are pretty easy and fast to integrate even into an existing project for example using import of projects [11].

3.2 Coding tools

Coding is one of the parts of the UI/UX development process highly influenced by AI support integration. In an international environment, the most popular ones are the well-known ChatGPT and GitHub Copilot.

StackOverflow research shows that currently, sixty-two percent of developers are using AI su-

pport, and even more plan to start utilization in the near future. The most popular applications are documenting, coding by itself, and code testing. At the same time, most developers are not worrying about possible job displacements but about possible disinformation and misinformation [12].

Similar research by GitHub has shown an even higher AI utilization rate (82%) among all IT professionals. The most important benefits are improved code quality as far as easier new languages and codebases understanding and enhanced testing. At the same time, some developers face issues and limitations from the side of the companies they are employed [13].

The research conducted among Slovenian software development professionals has shown pretty much similar results. The most popular tools there are ChatGPT and GitHub Copilot and each two of the three specialists are utilizing those tools. However, those professionals are primarily technicians while business and management-related people are avoiding AI utilization and show much more concern about it [2].

While ChatGPT is an external tool GitHub Copilot can be integrated with all the most popular IDEs like Visual Studio code or any of JetBrains's ones in a few clicks. It can support development into all the most popular languages like Java and JavaScript/TypeScript [14].

3.3 Testing tools

Testing is the most influenced by AI integration part of software development. AI tools are nowadays utilized in different aspects of the process. The most important are security and integration parts. The last one is commonly more addressed in UI/UX design nevertheless both of them play an important role there.

In terms of security testing AI is utilized in order to improve its efficiency and efficacy. For instance, those tools can analyze code to identify potential vulnerabilities, find anomalies, analyze user behavior, or perform penetration tests [15]. One of the research projects conducted in this sphere shows some minor improvements in efficiency and huge reductions in time consumption [15]. Another research among IT companies has identified that most of them (70%) nowadays use AI support in DevOps and security testing in some form and most of them have reported improvements in security and operational efficiency after the start of utilization [16].

In terms of integration testing AI is utilized for already mentioned user behavior analysis as far as for class analysis and test case generation and development [15] [17]. There for instance resources of testing teams optimizations and reduced error risks are mentioned as potential benefits in one of the research projects [17].

The most popular tools utilized in UI testing nowadays are for example Katalon and BrowserStack [18] [19]. Most of them are pretty much easy to integrate and utilize into the development process. For example, both of the mentioned before has possibilities for instant pipeline integration with the most popular automating tools like Jenkins and GitHub Actions [20] [21] [22] [23].

4 THE APPLICATION X

Application X is a student routine supportive app. It is developed as a native Android application with an XML/Java user interface and a REST API with a relational database in its back end.

According to the customer's requirements, it was made relatively simple in order to be able to support most of the active Android versions and to ensure as stable operating as possible.

The app consists of a few relatively simple views including some standard built-in Android inputs, navigation buttons, and data holders. Also, it has some custom commonly repetitive components.

Originally UI part of the application had integrated automated end-to-end testing with the Espresso and a Jenkins build pipeline.

A few years after the application was launched the company decided to improve and modernize it. The UI part was not an exception. Visual design and functionalities needed to be significantly changed and optimized. Some views were told to be totally reworked. In order to enhance the development process the team responsible for the application has decided to integrate and start utilizing some AI-supportive tools.

5 ARTIFICIAL INTELLIGENCE INTEGRATION

First of all, the team has decided to improve the design process, in particular visual prototyping. Also, the development process itself was selected for modernization. Here code optimization and testing were identified as possible application points. Because the application is relatively simple the team did

not include any user behavior analytical tools in the list. Calculations of the time spent were prepared based on the project management and monitoring tool utilized by the team.

As mentioned before, in order to keep business secret some particular details of management of the project or the implementation can not be described in this part precisely. However, all of the data, the development company has allowed to publish are presented.

5.1 Prototyping

The development team has been utilizing the Figma prototyping tool during the development process, so first of all Autoflow plugin was added in order to improve existing prototype corrections and adoption. Then the integrated UX Pilot AI tool has been selected for prototyping enhancement with design generation and evaluation. These tools helped to speed up design corrections and the prototyping process. Development and agreement of the design of the first version of the application has taken a little bit more than two months. Designing the updated version with the same customer, the same team, and approximately the same amount of workload was completed in a little bit less than one month (Figure 2). Of course, there could have been other factors influencing the process present. However, the impact of AI tools is definitely visible too.

5.2 Implementation

After the completion of designing the new app version, the team started corrections implementation. In comparison with the first version development UI part of the development team has started to utilize GitHub Copilot in work. The efficiency increase was significant. During the original version implementation, it took almost one working week to completely implement, test, and optimize one screen view. After the start of AI support utilization, the same process in case of complete rework and reimplementation of a view required nearly three working days (Figure 2).

Additionally, the development part of the team has reported improvements in code readability, due to built-in grammar checks. Also, the utilization of AI support has helped in refactoring and optimization both of some of the existing and newly written code. This was achieved with built-in Copilot's coding support and analysis tools.

5.3 Testing

The last development stage where the improvements were the most significant was automated testing. Here the BrowserStack was integrated into the pipeline. It has taken 3 days and went through very smoothly. Here efficiency measurement could be made by the number of identified problems which has increased more than two times. At the same time, the number of crashes after the deployment of the

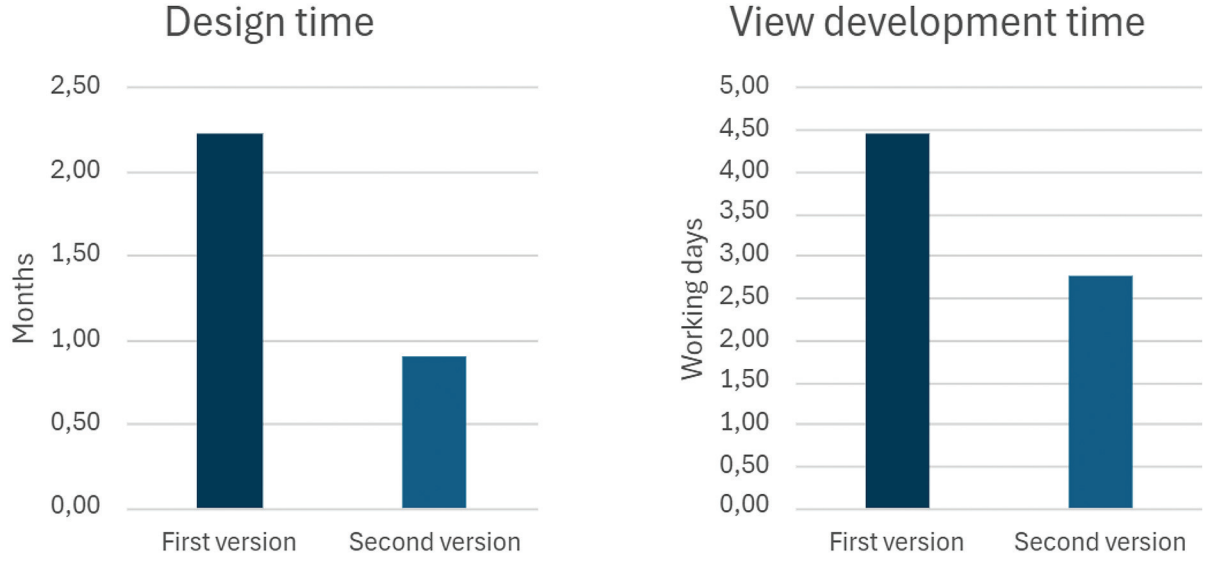

Figure 2: **Design and development time comparison**

second version decreased to zero. Of course, the period after deployment of the second version is less than the time of activity of the first version, but by the same period after publishing the original version already had a few crashes. An especially important benefit of the integration was that AI assistance multiple times helped to identify some very specific but realistic use cases and related issues that were not found by the development and testing responsible parts of the team nevertheless these problems were already present in the original version of the application and caused application crashes.

By the moment of writing, the time passed after the second version launch is still relatively short in order to compare the satisfaction and feedback of the users with the first version. For this reason, this factor cannot be analyzed and presented here.

6 CONCLUSION

By the end development and deployment of the second version has been finished and it has taken less than 2 months in comparison with a little bit more than 5 months in the case of the first version. The time spent on each of the steps described has decreased significantly. Of course, there are some limitations related to different amounts of workload in cases of development from the very beginning and of a huge rework. However, again the difference is very significant and the role of AI tools integration in such improvement can be assumed with very high probability. Also, a very important moment here is the improvement of the quality of the application.

In conclusion, it can be stated that AI integration into the software development process can significantly improve it both in terms of efficiency and quality. This case study can be used as a good example, for instance, to justify investments into such integration before business and management people or to motivate development teams to decide upon it.

LITERATURE

[1] S. Singh, »15 Global Trends for 2024,« Forbes, Dec. 12, 2023. [Online]. Available: https://www.forbes.com/sites/sarwantsingh/2023/12/11/15-global-trends-for-2024/

[2] M. Nikitashin and B. Werber, «Artificial Intelligence and Software Development in Slovenia: Adoption, Challenges, and Opportunities", submitted for publication.

[3] C. Okonkwo, »Assessment of User Experience (UX) design trends in mobile applications,« Journal of Technology and Systems, vol. 6, no. 5, pp. 29–41, Aug. 2024, doi: 10.47941/jts.2147.

[4] J. Nielsen, Usability Engineering. 1993. [Online]. Available: https://dl.acm.org/citation.cfm?id=2821575

[5] T. C. G and A. J. Devi, »A study and overview of the mobile app development industry,« International Journal of Applied Engineering and Management Letters, pp. 115–130, Jun. 2021, doi: 10.47992/ijaeml.2581.7000.0097.

[6] A. Neumann, N. Laranjeiro, and J. Bernardino, »An analysis of Public REST Web Service APIs,« IEEE Transactions on Services Computing, vol. 14, no. 4, pp. 957–970, Jun. 2018, doi: 10.1109/tsc.2018.2847344.

[7] L. Četina and L. Pavlič, Življenjski cikel cevovodov nepreki-njene namestitve informacijskih rešitev. 2024, pp. 47–56. doi: 10.18690/um.feri.4.2024.4.

[8] A. Samir, H. Amin, and N. Badr, »A SURVEY ON AUTOMATED USER INTERFACE TESTING FOR MOBILE APPLICATIONS,« International Journal of Intelligent Computing and Information Sciences/International Journal of Intelligent Computing and Information Sciences, vol. 0, no. 0, pp. 1–11, May 2022, doi: 10.21608/ijicis.2022.98138.1124.

[9] A. M. Sinaga, Y. Pratama, F. O. Siburian, and K. J. F. P. S, »Comparison of graphical user interface testing tools,« Journal of Computer Networks Architecture and High Performance Computing, vol. 3, no. 2, pp. 123–134, Jul. 2021, doi: 10.47709/cnahpc.v3i2.951.

[10] Y. Xu, Y. Liu, H. Xu, and H. Tan, »AI-Driven UX/UI Design: Empirical research and Applications in FinTech,« International Journal of Innovative Research in Computer Science & Technology, vol. 12, no. 4, pp. 99–109, Jul. 2024, doi: 10.55524/ijircst.2024.12.4.16.

[11] GeeksforGeeks, »10 Best AI Tools for UX Designers in 2024,« GeeksforGeeks, May 24, 2024. https://www.geeksforgeeks.org/best-ai-tools-for-ux-designers/

[12] »2024 Stack Overflow Developer Survey.« https://survey.stackoverflow.co/2024/

[13] K. Daigle, »Survey: The AI wave continues to grow on software development teams – The GitHub Blog,« The GitHub Blog, Sep. 03, 2024. https://github.blog/news-insights/research/survey-ai-wave-grows/

[14] »GitHub Copilot · Your AI pair programmer,« GitHub, 2024. https://github.com/features/copilot

[15] B. R. Bhimanapati, S. Jain, and P. K. G. Pandian, »Security Testing for Mobile Applications Using AI and ML Algorithms,« Journal of Quantum Science and Technology, vol. 1, no. 2, pp. 44–58, Jun. 2024, doi: 10.36676/jqst.v1.i2.15.

[16] N. B. Gajbhiye, N. A. Aggarwal, and N. S. Jain, »Automated Security Testing in DevOps Environments Using AI and ML,« International Journal for Research Publication and Seminars, vol. 15, no. 2, pp. 259–271, Jun. 2024, doi: 10.36676/jrps.v15.i2.1472.

[17] I. Alazzam, A. M. R. AlSobeh, and B. B. Melhem, »Enhancing integration testing efficiency through AI-driven combined structural and textual class coupling metric,« Online Journal of Communication and Media Technologies, vol. 14, no. 4, Oct. 2024, doi: 10.30935/ojcmt/15524.

[18] GeeksforGeeks, »Top 10 AI testing tools for test Automation in 2024,« GeeksforGeeks, Oct. 15, 2024. https://www.geeksforgeeks.org/top-ai-testing-tools-for-test-automation/

[19] »AI-Augmented Software-Testing Tools Reviews and Ratings,« Gartner. https://www.gartner.com/reviews/market/ai-augmented-software-testing-tools (accessed Nov. 05, 2024).

[20] »Jenkins integration overview | Katalon Docs.« https://docs.katalon.com/katalon-platform/integrations/cicd-integrations/jenkins-integration/jenkins-integration-overview

[21] »Katalon Studio GitHub Action | Katalon Docs.« https://docs.katalon.com/katalon-platform/integrations/cicd-integrations/katalon-studio-github-action
[22] S. A. Gurudatt and B. Shrikanth, »Jenkins for Test Automation : Tutorial | BrowserStack,« BrowserStack, Sep. 01, 2022. https://www.browserstack.com/guide/jenkins-for-test-automation
[23] »Integrate your Selenium test suite with GitHub Actions | BrowserStack Docs.« https://www.browserstack.com/docs/automate/selenium/github-actions

▪

Maksim Nikitashin is a PhD student and an assistant in the field of Business Informatics at the Faculty of Management of the University of Primorska. He holds a master's degree in Organization and Management of Information Systems from the University of Maribor. Also, he has a few years of experience developing different kinds of web and mobile information solutions. The fields of his professional interests and competencies include, but are not limited to: software engineering, business informatics, and artificial intelligence.