

Evaluation of the Cross-Platform Framework Flutter Using the Example of a Cancer Counselling App

Leon Lovrić¹, Mahsa Fischer¹, Natalie Röderer² and Alexander Wunsch²

¹Faculty of Business and Transport Management (WV), Heilbronn University of Applied Sciences, Heilbronn, Germany

²University Medical Center Freiburg, Department of Psychosomatic Medicine and Psychotherapy, Faculty of Medicine, Albert Ludwigs University, Freiburg, Germany

Keywords: Cross-Platform Frameworks, Flutter, Health Applications, Cancer Counselling.

Abstract: The advance of digitalization is constantly bringing new solutions to various areas of life in our society. The COVID-19 pandemic, among other things, brought increased attention to the application and support of treatments through digital solutions in the healthcare sector due to contact restrictions. However, the development of digital solutions comes at a high cost in terms of time and expenses. Mobile app development requires the development of two separate apps for the two respective market-leading mobile operating systems, iOS and Android. Cross-platform frameworks make it possible to develop apps for both operating systems on a single code base, thus saving the development and maintenance of two separate codes. Flutter is currently the most popular cross-platform framework for the development of mobile apps. This paper has evaluated Flutter based on an existing criteria catalogue. As a usage context for the evaluation, a prototype for Cancer Counselling App of the University Medical Center Freiburg was implemented. According to the gained own prototyping experience with Flutter and a thorough literature analysis in this area, the criteria catalogue was filled out and the result was compared with other mobile App development paradigms.

1 INTRODUCTION

Digitalisation has long since arrived in the healthcare sector and is increasingly changing processes and procedures. The COVID 19 pandemic has made many people aware of the advantages that digitalisation of the health system can bring. This has resulted in a new openness to the possibilities of digitalisation. In recent years, there has also been a significant push towards digital healthcare in health legislation (Weber and Heitmann 2021).

Digital solutions are fundamentally suitable in healthcare to ensure quality for treatments. This is due to two characteristics of digital solutions, the possibility of a complete collection of data and the ability to provide data independent of time and date (Brönneke and Debatin 2022).

Together with the Heilbronn University of Applied Science and Fraunhofer Institute for Systems and Innovation Research ISI, among others, the University Medical Center Freiburg launched the Cancer Counselling App project (Krebsberatungs-App) in 2020 supported by the Federal Ministry of Education and Research in Germany. The Cancer Counselling App is intended to provide psycho-

oncological counselling for cancer patients and their relatives.

Every second person develops cancer during his or her life (Centre for Cancer Registry Data at the Robert Koch Centre 2016). The incidence of cancer has been increasing for decades, whereas mortality has been decreasing on an age standardised basis (Centre for Cancer Registry Data at the Robert Koch Centre 2016). As a result, the number of cancer patients and long-term survivors is increasing (Centre for Cancer Registry Data at the Robert Koch Centre 2016). Since cancer patients are often under considerable psychological distress (Mehnert et al. 2014), it is important that they have access to low-threshold psycho-oncological support when needed (Federal Ministry of Health 2008). Cancer patients and their relatives can receive psycho-oncological support at the Psychosocial Cancer Counselling Centre in Freiburg. Internal statistics show that demand has increased in recent years. To be able to meet the increasing demand, a Cancer Counselling App is to be developed as a supplement to the services offered.

In recent years, several app- and web-based support options have already been developed in the

field of psycho-oncology. These include, for example, psycho-oncological web-based programmes for relaxation (Urech et al., 2018), a web-based support programme for relatives (Bodschwinna et al., 2022) or an app for psycho-oncological support (Fosanis GmbH n.d.). All apps can be used largely independently of third parties, i.e., there is little or no interaction with professionals (e.g., psycho-oncologists). However, the relationship aspect has been proven to be particularly important in counselling and psychotherapy (Priebe & McCabe 2008) and should therefore be taken up as an innovative aspect in the Cancer Counselling App.

According to own research cancer patients face the following problems: Being overwhelmed, lack of knowledge about the existence of cancer counselling centres, living in a rural area, physical impairments or language barriers. Yet every second person develops cancer during his or her life; the proportion of those affected (patients and relatives) is therefore very high. The Cancer Counselling App is intended to create a solution to overcome these problems and improve the low-threshold nature of psychosocial care, initially in Freiburg and later, ideally, throughout Germany. Within this work an initial implementation of the Cancer Counselling App is made in the form of a prototype and then the cross-platform framework - Flutter - used for this is evaluated.

The development of digital solutions in the form of mobile applications is associated with high costs and time expenditures. A systematic, structured, and engineered development of apps on mobile devices entails various challenges, functions and limitations that need to be considered. High hardware and software fragmentation is one of the biggest challenges in designing, developing, and testing mobile apps. Due to the number of different mobile devices and operating system versions, there are different development paradigms for mobile applications. In addition to the native paradigm, there are other paradigms whose goal is a development that is more independent of the operating system (Vollmer 2017).

One of these paradigms is the development through cross-platform frameworks. By evaluating different paradigms, Shah et al. (2019) were able to identify the cross-compiled development paradigm as the best alternative to native development. It has several advantages and is therefore very interesting for the development of the Cancer Counselling App. Among the advantages are (El-Kassas et al. 2017):

- The App is developed once and is available to more users who use different platforms

- Ease of development as the App is written once and deployed many times on different platforms
- Reduction of the development time and efforts.

As part of a case study, a first prototype of the Cancer Counselling App is created using the cross-platform framework Flutter. Subsequently, Flutter is evaluated through a criteria catalogue. The results enable a comparison among native paradigm, Flutter and other cross-platform frameworks.

2 RELATED WORK

Flutter has steadily grown in popularity since its release in late 2017 (Zammetti 2019). However, there are hardly any works that evaluate Flutter on a broader set of criteria like with a criteria catalogue. Only partial aspects such as performance or scalability of Flutter are examined. Mobile app development is time consuming and expensive (Vollmer 2017). The resulting work should therefore support the decision of Flutter being used for the development of digital health applications as well as other types of mobile applications.

The original aim of the literature review of this work was to identify criteria suitable for evaluating and assessing Flutter in terms of the suitability of this technology for mobile app development. While conducting the literature review, a catalogue of criteria of Rieger and Majchrzak (2019) was found. The catalogue of criteria emerged from a literature review to identify criteria regarding the evaluation of cross-platform frameworks and a subsequent evaluation by experts. The literature analysis includes some works that were already found in the process of the originally planned literature analysis of this bachelor thesis. Therefore, the criteria catalogue was very well suited to be used for the evaluation of Flutter.

In the paper by Rieger and Majchrzak (2019), however, only Cordova and React Native as well as native development have been evaluated to date. This means that an evaluation of Flutter is still pending. An evaluation of Flutter to the extent of the criteria catalogue could not be identified in the literature review. The discovered works that deal with the evaluation of Flutter mostly only examine the technology regarding individual aspects, such as the performance or scalability of Flutter apps.

The second phase of the literature analysis was about evaluating criteria from the criteria catalogue that could not be evaluated through the

implementation of the prototype nor through the literature analysis. These are five criteria: Durability, scalability, security, robustness, and performance. A total of 16 sources were identified that deal with the evaluation of at least one of the unevaluatable criteria.

3 METHODOLOGY

For the methodological procedure of this work, the approach for case studies according to Lindner (2020) is applied. After formulating a research question the work is divided into three components. These are Determination of Evaluation Criteria, Case Study: Development of the Prototype and an Evaluation based on the identified Criteria.

3.1 Determination of Evaluation Criteria

For the evaluation of Flutter, the criteria catalogue for cross-platform frameworks by Rieger and Majchrzak (2019) is used.

Rieger and Majchrzak (2019) deliberately try not to provide any precise guidelines or checklists for evaluating the individual criteria to do justice to the rapid development and change in the field of mobile application development. The description of the individual criteria as well as exemplary evaluation profiles should be used to assign a score from 0 (unsatisfactory) to 5 (complete satisfactory). Nevertheless, the result of the individual evaluation should be as objective as possible at the end, which is why all criteria are weighted according to their relevance.

3.2 Case Study

Within the case study a prototype of the Cancer Counselling App was developed with Flutter. To understand the requirements and the different target groups for the app, a requirements workshop was conducted together with the University Medical Center Freiburg. An appointment scheduling, chat, information overview and intervention functionalities were identified as the main functions of the app. The users of the app can be assigned to the roles: Patients, relatives of patients, psych-oncologists and social workers.

Within the prototype a registration and login as well as a chat function are developed. By implementing these functions, sufficient experience in development with Flutter can be gained to be able

to evaluate Flutter regarding the aspects of data management, data processing or the UI.

For the development of the login and registration function, the authentication system of Firebase was used. Firebase has libraries and services through which the creation and management of users can be implemented. A complete authentication back-end is provided for login with passwords, federated identity providers, email links and text messages. Up to 10,000 free authentications are available per month, which is fully sufficient for the volume of authentications provided by the prototype.

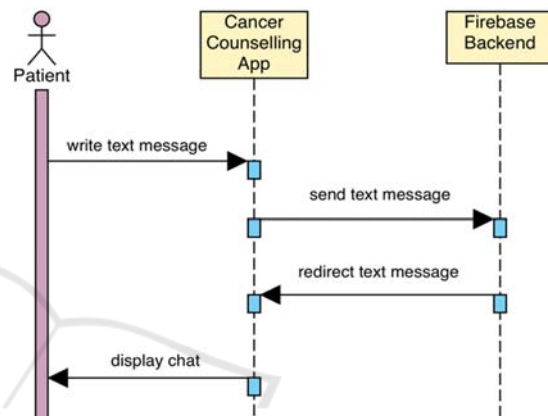


Figure 1: Sequence Diagram Chat Function.

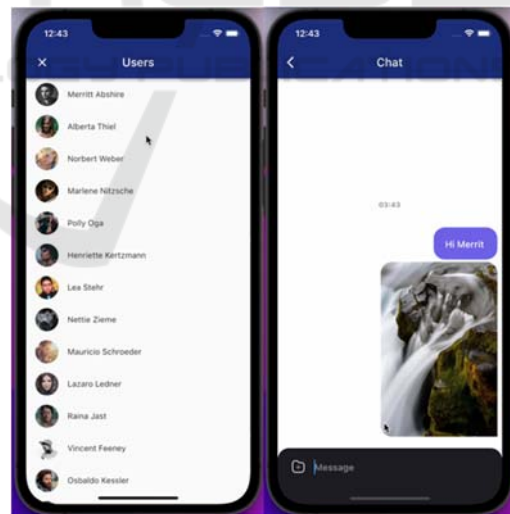


Figure 2: Chat Function Screenshots.

The chat section concept is similar to common messenger apps such as Whatsapp or Facebook Messenger. The aim of the chat functionality is to enable communication between the patients and the followings: cancer counselling centre, other counselling services (e.g. nutrition counselling, social services), self-help groups, other patients with similar

disease and emergency center. However, the design of this function is still open and yet to be fully discussed, whether patients are free to exchange information with each other in chats, there is a fear of negative conversations that would be counterproductive for the psychological well-being of the patients. Suitable ways for patients to exchange information with each other are yet to be discussed.

Nevertheless, the development of the chat function offers a good possibility to also test the integration of external libraries, among other things. The chat function was implemented using the UI library "Flyer Chat" (Flyer Chat 2022). With almost 500 likes, the "Flyer Chat" package is currently the most popular chat UI library in the official package repository for Dart and Flutter. In addition to UI widgets, the library also offers functions for integrating Firebase as a backend. This means that it can be integrated into backends that already exist for login and registration. Figure 2 illustrates how the App UI and the backend function together.

3.3 Evaluation Based on Identified Criteria

With the nature of this work being about mobile apps other weights from Rieger and Majchrzak's (2019) catalogue except the weight for smartphone app development have been removed. To have a comparison to other cross-platform frameworks, as well as to native development, the ratings of these from Rieger and Majchrzak (2019) have been retained. Criteria that cannot be assessed through the prototype development process are assessed through a literature review (Fettke 2006; Webster and Watson 2002; Brocke et al. 2009) or through Flutter's official documentation (Flutter 2022).

4 EVALUATION

The criteria of the criteria catalogue in table 1 can be categorized in infrastructure, development, app and usability criterions.

Table 1: Criteria Catalogue.

Criteria		Weight (%)	Flutter	Phone Gap (Cordova)	React Native	Native Apps
I1	License	5	5	5	5	5

I2	Target Platforms	6	5	5	4	1
I3	Development Platfroms	2	5	5	4	2
I4	Distribution Channels	2	5	3	3	4
I5	Monetisation	1	5	3	3	5
I6	Internationalisat ion	1	5	3	3	5
I7	Long-Term Feasibility	5	3	5	3	4
D1	Development Environment	7	4	5	4	5
D2	Preparation Time	7	4	4	4	3
D3	Scalability	2	4	3	4	3
D4	Development Process Fit	2	4	3	3	2
D5	UI Design	4	4	3	2	4
D6	Testing	3	3	4	3	5
D7	Continous Delivery	3	3	5	3	3
D8	Configuration Management	1	0	0	0	3
D9	Maintainability	2	4	4	4	2
D10	Extensibility	2	3	5	2	5
D11	Custom Code Integration	2	5	3	3	5
D12	Pace of Development	4	4	4	3	0
A1	Hardware Access	4	5	4	3	5
A2	Platform Functionality	5	5	4	3	5
A3	Connected Devices	3	5	2	2	5
A4	Input Heterogeneity	1	4	4	4	5
A5	Output Heterogeneity	1	4	4	4	5
A6	App Life Cycle	2	4	4	4	5
A7	System Integration	3	5	3	3	5
A8	Security	3	2	0	0	3
A9	Robustness	2	3	4	2	3
A10	Degree of Mobility	1	3	1	3	5
U1	Look and Feel	5	4	3	4	5
U2	Performance	4	4	2	3	5
U3	Usage Patterns	2	2	2	2	2
U4	User Authentication	3	1	0	0	1
Weighted Score			3,95	3,59	3,11	3,73

4.1 Infrastructure

Flutter is open source and licensed under BSD (GitHub 2022) (I1 Licence). Flutter supports the following target platforms: both Android and iOS apps, interactive apps that can be run via web hosting or on the desktop under Windows and, since the latest release of Flutter version 3, also under macOS or Linux (Flutter 2022) (I2 Target Platform). Flutter apps can also be developed under all these operating systems (Flutter 2022) (I3 Development Platform). In addition to the Apple App Store and the Google Play Store, the developed app packages can also be published in the Microsoft Store and Snap Store (Linux) (Flutter 2022) (I4 Distribution Channels). Flutter offers various monetisation options through the integration of advertising, in-app purchases, and the connection to Apple Pay and Google Pay (Flutter 2022) (I5 Monetisation). Regarding internationalisation, there is the possibility of identifying the language setting of the users' end device and providing corresponding language packages (a total of 78 languages are supported) (Flutter 2022) (I6 Internationalisation). Masaad Alsaïd et al. (2021) estimate the longevity of Flutter to be quite good, since Google, one of the biggest tech giants, is behind the technology. The degree of use by the community is also quite high (JetBrains 2022) and even makes Flutter the current most popular cross-platform framework, which is very promising for the longevity of the technology. However, the technology is still relatively young (I7 Long-Term Feasibility).

4.2 Development

The choice of an IDE is not limited under Flutter, but not all development environments have Flutter support. With Android Studio, IntelliJ IDEA and Visual Studio Code, Flutter is supported by popular development environments (Flutter 2022). Visual Studio was able to support the process of prototype development with helpful tools, like an auto completion for the code and emulator support (D1 Development Environment). Dart can be classified as a developer-friendly programming language. Since Dart is an object-oriented programming language, development is relatively close to the native development paradigm of Android. However, due to fewer required lines of code and many prefabricated functions, development in Flutter is even easier and faster compared to native development (D2 Preparation Time). By using state management approaches, Flutter projects can be made very scalable. Among other things, the "BLoC"

approach enables the modularisation of the architecture of a Flutter app, making it scalable and maintainable with only a few adjustments (Bardram 2020; Payne 2020; Szczepanik and Kędziora 2020) (D3 Scalability, D4 Development Process Fit, D9 Maintainability). Flutter has many pre-built UI widgets through which a UI can be quickly built. Optionally, by using MaterialApp or Cupertino libraries, these widgets can be rendered on the respective operating system with the appropriate design language. In this case, however, two versions of the app would still have to be developed with the respective design language (D5 UI Design). Flutter offers various functions for the execution of automated tests. These can be divided into the following categories: Unit Tests, Widget Tests and Integration Tests. Testing iOS through Xcode (development environment for macOS) is not possible under Windows or Linux. There are a handful of CD tools including Codemagic, Bitrise, Appcircle and fastlane that support automated flutter testing (Flutter 2022) (D6 Testing, D7 Continuous Delivery). Due to the modularisation using a state management approach, third-party extensions can be offered well. However, with around 24,000 (Dart packages 2022a), there are currently still few third-party libraries and usually there is not much further help for integrating and adapting the library functions beyond the documentation provided by the library publishers (D10 Extensibility). Flutter also offers the possibility of platform-specific implementations through native code. Alternatively, third-party packages can also be integrated with native code (Flutter 2022) (D11 Custom Code Integration). Due to many prefabricated widgets, the development speed is relatively fast, which is also reflected in the lines of code required (D12 Pace of Development).

4.3 App

The enabled integration of native code also enables access to the hardware, as well as to various functionalities and the connection of external devices such as to smartwatches of the target platform. For this purpose, many packages are already provided in the form of third-party libraries (Flutter 2022) (A1 Hardware Access, A2 Platform Functionality, A3 Connected Devices). Flutter supports various gesture inputs with tapping, double tapping, vertical or horizontal dragging and "drag and drop" (Flutter 2022) (A4 Input Heterogeneity). Also, various functions are provided to enable screen customisation for specific devices or to allow users to customise the screen. Through the possibility of accessing the

hardware of the target platform, sounds can also be played (Flutter 2022) (A5 Output Heterogeneity). The app lifecycle can be observed through the function "AppLifecycleState" and the states can be described accordingly (Flutter - Dart API docs 2022) (A6 App Life Cycle). Flutter offers functions that enable interface communication via http, for example. In addition, functions for JSON data processing and serialisation are also provided (Flutter 2022) (A7 System Integration). Data can be processed through Flutter in a secure manner, but not as secure as in native solutions (Shah et al. 2019; Bagul et al. 2022). There are few features that facilitate secure data processing (A8 Security). Flutter apps can be classified as comparatively robust. In interaction with Firebase, various offline functionalities can be provided (HOD. Dr R Juliana et al. 2021; Bardram 2020; Payne 2020; Bhole and Kumar 2022) (A9 Robustness). With the ability to develop applications beyond mobile operating systems, there are now also new levels of mobility. However, development for WatchOS, for example, is not yet possible (A10 Degree of Mobility).

4.4 Usability

By enabling the use of design languages of the target platforms, customised representations can be developed. The prototype of the Cancer Counselling App felt like an app and not like a website (U1 Look and Feel). The performance of Flutter apps can be classified as very good. It is better than competing cross-platform frameworks, but in most cases, it does not quite match the performance of native apps (Bagul et al. 2022; Carius et al. 2022; İŞITAN and KOKLU 2020; Mahendra and Anggorojati 2020; Nawrocki et al. 2021; Rudenko et al. 2021; Biørn-Hansen et al. 2020) (U2 Performance). As already described for criterion A9 Robustness, according to HOD. Dr R Juliana et al. (2021), offline support can also be ensured in Flutter apps. Support for wallets can be achieved by integrating libraries such as the "pass-flutter" (Dart packages 2022b) (U3 Usage Patterns). Flutter does not offer any further functions described in the criterion user authentication (U4) beyond the packages provided for biometric verification or voice recognition.

5 DISCUSSION & FUTURE WORK

General deductions of a single case study are not always possible or only possible to a limited extent. Therefore, multi-case studies could lead to more expressive results.

By using multi-studies, scenarios of different development projects could be evaluated again by Flutter to crystallise special features. This could have led to different results according to changed requirements. Regarding the selection of a development paradigm for the Cancer Counselling App project, the prototype could have been created by using different development paradigms and the results could have been compared with each other.

Evaluating the results by oneself, leads to the risk of a subjective evaluation. The additional evaluation of experts could have counteracted this.

This evaluation of Flutter can be used to support future decisions regarding the selection of a development paradigm for mobile app development. Currently, few evaluations of the scope of this paper for development through Flutter can be found in academic databases. With the release of Flutter 3.0, Flutter apps can now be used on the desktop of various operating systems. Up to now, the option of web-hosting Flutter apps already existed. Therefore, it makes sense to evaluate Flutter on the other available target platforms as well.

6 CONCLUSIONS

Digitalisation has already brought about many improvements in information and communication technology that have already significantly changed our lives in many aspects. Today, it is not yet complete and is an ongoing process that also brings changes to other areas of our everyday lives.

In Germany's healthcare system, the COVID 19 pandemic provided another boost to digitalisation. Mobile applications can increase the quality of treatments. Local treatments can also be supplemented or partially replaced.

Cross-platform frameworks enable easy development because apps can be written once and deployed multiple times on different platforms through these technologies. This can reduce development time and effort. The most popular cross-platform frameworks are Cordova, Ionic, React Native and Flutter. Flutter is steadily gaining

relevance and is currently the most popular cross-platform framework.

For the evaluation, a catalogue of criteria for cross-platform frameworks was identified. Based on the criteria, Flutter was evaluated in terms of infrastructure, development, the developed app and its usability. A prototype of the Cancer Counselling App was created to be able to evaluate Flutter through the criteria catalogue based on the resulting experiences, complemented by a literature analysis and the official documentation of Flutter. The results show that Flutter with the highest overall score in the criteria catalogue is a considerable alternative to native app development.

REFERENCES

- Bagul, Apoorva; Sinkar, Pooja; Jadhav, Priyanka; Ahire, Deepali; Prof. D. D. Sharma (2022): A MENTAL HEALTH TRACKER BUILT USING FLUTTER AND FIREBASE. In: *International Journal of Science & Engineering Development Research*, S. 115–118. Online available under <https://www.ijedr.org/papers/ijedr2201009.pdf>.
- Bardram, Jakob E. (2020): The CARP Mobile Sensing Framework -- A Cross-platform, Reactive, Programming Framework and Runtime Environment for Digital Phenotyping. Online available under <https://arxiv.org/pdf/2006.11904>.
- Biørn-Hansen, Andreas; Rieger, Christoph; Grønli, Tor-Morten; Majchrzak, Tim A.; Ghinea, Gheorghita (2020): An empirical investigation of performance overhead in cross-platform mobile development frameworks. In: *Empir Software Eng* 25 (4), S. 2997–3040. DOI: 10.1007/s10664-020-09827-6.
- Bodschwinna, Daniela.; Lorenz, Inga; Bauereiß, Natalie; Gündel, Harald; Baumeister, Harald; Hönig, Klaus (2022): A psycho-oncological online intervention supporting partners of patients with cancer (PartnerCARE): Results from a randomized controlled feasibility trial. *Psycho-Oncology*. <https://onlinelibrary.wiley.com/doi/abs/10.1002/pon.5917>.
- Brocke, J.; Simons, A.; Niehaves, B.; Riemer, K.; Clevén, A. (2009): Reconstructing the giant: On the importance of rigour in documenting the literature search process. Online available under <https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1145&context=ecis2009>.
- Brönneke, Jan Benedikt; Debatin, Jörg Felix (2022): Digitalisierung im Gesundheitswesen und ihre Effekte auf die Qualität der Gesundheitsversorgung. In: *Bundesgesundheitsblatt, Gesundheitsforschung, Gesundheitsschutz*. DOI: 10.1007/s00103-022-03493-3.
- Carius, Lars; Eichhorn, Christian; Plecher, David A.; Klinker, Gudrun (2022): Cloud-Based Cross-Platform Collaborative AR in Flutter. In: *Proceedings, 2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops*. 12-16 March 2022, virtual event. 2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW). Christchurch, New Zealand, 3/12/2022 - 3/16/2022. Los Alamitos, California: IEEE Computer Society, Conference Publishing Services, S. 682–683.
- Centre for Cancer Registry Data at the Robert Koch Centre (2016): Bericht zum Krebsgeschehen in Deutschland 2016. (Zentrum für Krebsregisterdaten im Robert-Koch-Zentrum, Ed.). Robert-Koch-Institut.
- Dart packages (2022a): Search results for sdk:flutter. Online available under <https://pub.dev/packages?q=flutter>, last updated at 18.06.2022, last visited at 18.06.2022.
- Dart packages (2022b): pass_flutter | Flutter Package. Online available under https://pub.dev/packages/pass_flutter, last updated at 21.06.2022, last visited at 21.06.2022.
- El-Kassas, Wafaa S.; Abdullah, Bassem A.; Yousef, Ahmed H.; Wahba, Ayman M. (2017): Taxonomy of Cross-Platform Mobile Applications Development Approaches. In: *Ain Shams Engineering Journal* 8 (2), S. 163–190. DOI: 10.1016/j.asej.2015.08.004.
- Federal Ministry of Health (2008): Ziele des Nationalen Krebsplans—NKP. https://www.bundesgesundheitsministerium.de/fileadmin/Dateien/3_Downloads/N/Nationaler_Krebsplan/Nationaler_Krebsplan-Zieluebersicht.pdf.
- Fettke, Peter (2006): State-of-the-Art des State-of-the-Art. In: *Wirtsch. Inform.* 48 (4). DOI: 10.1007/s11576-006-0057-3.
- Flutter (2022): Flutter documentation. Online available under <https://docs.flutter.dev/>, last updated at 17.06.2022, last visited at 17.06.2022.
- Flutter - Dart API docs (2022). Online available under <https://api.flutter.dev/index.html>, last updated at 10.06.2022, last visited at 21.06.2022.
- Flyer Chat (2022). Online available under <https://flyer.chat/>, last updated at 21.06.2022, last visited at 21.06.2022.
- Fosanis GmbH (2022): Mika—Krebs Therapie Assistent App für iOS und Android. Online available under: <http://www.mitmika.de>.
- GitHub (2022): Flutter. Online available under <https://github.com/flutter/flutter>, last updated at 17.06.2022, last visited at 17.06.2022.
- Heitkötter, Henning; Hanschke, Sebastian; Majchrzak, Tim A. (2013): Evaluating Cross-Platform Development Approaches for Mobile Applications. In: Jose Cordeiro und Karl-Heinz Krempels (Hg.): *Web Information Systems and Technologies*. 8th International Conference, WEBIST 2012, Porto, Portugal, April 18-21, 2012, Revised Selected Papers, Bd. 140. 2013 edition. Berlin: Springer (Lecture Notes in Business Information Processing), S. 120–138.
- HOD. Dr R Juliana; Naveen Kumar VG; Richard G; Shivadarshini P (2021): Evaccurate—A Smart Event Management App Using Flutter and Firebase. Online available under https://ijsret.com/wp-content/uploads/2021/07/ijsret_v7_issue4_570.pdf.

- İŞİTAN, Mehmet; KOKLU, Murat (2020): Comparison and Evaluation of Cross Platform Mobile Application Development Tools. In: *International Journal of Applied Mathematics Electronics and Computers*, S. 273–281. DOI: 10.18100/ijamec.832673.
- Jan Christoph; Daniel Rösch; Thomas Schuster; Lukas Waidelich (2019): *Current Progress in*
- JetBrains (2022): *Cross-platform mobile frameworks used by global developers 2021* | Statista. Online available under <https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/>, last updated at 24.03.2022, last visited at 24.03.2022.
- Lindner, Dominic (2020): *Forschungsdesigns der Wirtschaftsinformatik. Empfehlungen für die Bachelor- und Masterarbeit*. 1st ed. 2020. Wiesbaden: Springer Fachmedien Wiesbaden; Imprint Springer Gabler (Springer eBook Collection).
- Luft, Nikolaus; Dirisamer, Martin; Priglinger, Siegfried G. (2021): *Brillenfreiheit - welche*
- Mahendra, Mohammad; Anggorojati, Bayu (2020): Evaluating the performance of Android based Cross-Platform App Development Frameworks. In: *2020 the 6th International Conference on Communication and Information Processing. ICCIP 2020: 2020 the 6th International Conference on Communication and Information Processing*. Tokyo Japan, 27 11 2020 29 11 2020. New York, NY, United States: Association for Computing Machinery (ACM Digital Library), S. 32–37.
- Masaad Alsaïd, Mohamed Abdal Mohsin; Ahmed, Tarig Mohamed; Jan, Sadeeq; Khan, Fazal Qudus; Mohammad; Khattak, Amjad Ullah (2021): A Comparative Analysis of Mobile Application Development Approaches. In: *PPASA 58 (1)*, S. 35–45. DOI: 10.53560/PPASA(58-1)717.
- Mehnert, Anja; Brähler, Elmar; Faller, Hermann; Härter; Martin; Keller, Monika; Schulz, Holger; Wegscheider, Karl; Weis, Joachim; Boehncke, Anna; Hund, Bianca; Reuter, Katrin; Richard, Matthias; Sehner, Susanne; Sommerfeldt, Sabine; Szalai, Carina; Wittchen, Hans-Ulrich; Koch, Uwe (2014): Four-week prevalence of mental disorders in patients with cancer across major tumor entities. *Journal of Clinical Oncology: Official Journal of the American Society of Clinical Oncology*, 32(31), 3540–3546. <https://doi.org/10.1200/JCO.2014.56.0086>.
- Nawrocki, Piotr; Wrona, Krzysztof; Marczak, Mateusz; Sniezynski, Bartlomiej (2021): A Comparison of Native and Cross-Platform Frameworks for Mobile Applications. In: *Computer 54 (3)*, S. 18–27. DOI: 10.1109/MC.2020.2983893.
- Payne, Rap (2020): *Beginning App Development with Flutter. Create cross-platform mobile apps*. [Place of publication not identified]: APRESS. Online available under <https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=5989318>.
- Priebe, Stefan; McCabe, Rosemarie (2008): Therapeutic relationships in psychiatry: The basis of therapy or therapy in itself? *International Review of Psychiatry*, 20(6), 521–526. <https://doi.org/10.1080/09540260802565257>.
- Rieger, Christoph; Majchrzak, Tim A. (2019): Towards the definitive evaluation framework for cross-platform app development approaches. In: *Journal of Systems and Software 153*, S. 175–199. DOI: 10.1016/j.jss.2019.04.001.
- Rudenko, Sergey; Shakhov, Anatoliy; Pitera, Varvara; Chernova, Lyudmila; Sherstiuk, Olha (2021): Application of balanced scorecard for managing university development projects. In: *2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT)*. 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT). LVIV, Ukraine, 9/22/2021 - 9/25/2021. [S.l.]: IEEE, S. 311–314.
- Shah, Kewal; Sinha, Harsh; Mishra, Payal (2019): Analysis of Cross-Platform Mobile App Development Tools. In: *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*. 2019 IEEE 5th International Conference for Convergence in Technology (I2CT). Bombay, India, 3/29/2019 - 3/31/2019; IEEE, S. 1–7.
- Szczepanik, Michał; Kędziora, Michał (2020): State Management and Software Architecture Approaches in Cross-platform Flutter Applications. In: *Proceedings of the 15th International Conference on Evaluation of Novel Approaches to Software Engineering: SCITEPRESS - Science and Technology Publications*.
- Urech, Corrine; Grossert, Astrid; Alder, Judith; Scherer, Sandra; Handschin, Barbara; Kasenda, Benjamin; Borislavova, Borislava; Degen, Sven; Erb, Jennifer; Faessler, Alexandra; Gattlen, Laura; Schibli, Sarah; Werndli, Celine; Gaab, Jens; Berger, Thomas; Zumbund, Thomas; Hess, Viviane (2018): Web-Based Stress Management for Newly Diagnosed Patients With Cancer (STREAM): A Randomized, Wait-List Controlled Intervention Study. *Journal of Clinical Oncology*, 36(8), 780–788. <https://doi.org/10.1200/JCO.2017.74.8491>.
- Vollmer, Guy (2017): *Mobile App Engineering. Eine systematische Einführung - von den Requirements zum Go Live*. 1. Auflage. Heidelberg: dpunkt.verlag.
- Weber, Stefanie; Heitmann, Kai U. (2021): *Interoperabilität im Gesundheitswesen: auch für digitale Gesundheitsanwendungen (DiGA) verordnet*. In: *Bundesgesundheitsblatt, Gesundheitsforschung, Gesundheitsschutz 64 (10)*, S. 1262–1268. DOI: 10.1007/s00103-021-03414-w.
- Webster, Jane; Watson, Richard (2002): Analyzing the past to prepare for the future: Writing a literature review (*MIS Quarterly*, Vol. 26, No. 2). Online available under <https://www.jstor.org/stable/4132319>.
- Zammetti, Frank (2019): *Practical Flutter*. Berkeley, CA: APRESS. Online available under <https://link.springer.com/content/pdf/10.1007/978-1-4842-4972-7.pdf>.