A Mobile Android-Based Application for In-hospital Glucose Management in Compliance with the Medical Device Directive for Software

Stephan Spat[1], Bernhard Höll[2], Peter Beck[1],
Franco Chiarugi[3], Vasilis Kontogiannis[3], Manolis Spanakis[3],
Dimitris Manousos[3], and Thomas R. Pieber[2]

[1] JOANNEUM RESEARCH Forschungsges.m.b.H.,
Institute for Biomedicine and Health Sciences, Graz, Austria
{stephan.spat,peter.beck}@joanneum.at
[2] Medical University of Graz, Department of Internal Medicine,
Division of Endocrinology and Nuclear Medicine, Graz, Austria
{bernhard.hoell,thomas.pieber}@medunigraz.at
[3] Computational Medicine Laboratory, Institute of Computer Science,
Foundation for Research and Technology - Hellas, Heraklion, Crete, Greece
{chiarugi,vasilisk,spanakis,mandim}@ics.forth.gr

Abstract. In healthcare, the distribution of smartphones and tablet PCs running operation systems like Apple iOS or Android, attracts the interest in many application fields. In this paper we present insights into the development process of a Google Android-based tablet PC system, designed for in-hospital glucose management treatment of acute ill patients with type 2 diabetes. The system provides decision support for insulin dosing and falls within the scope of the Medical Device directive for software which came into effect in March 2010. In order to support usability and fulfill the design requirements, according to IEC 62366 standard, we included physicians and nurses in the design process to implement a modular user interface based on established clinical workflows using mockups and functional prototypes. With this approach we provided a solid foundation for validating our system with the demands of the medical device directive.

Keywords: decision support, diabetes mellitus, mobile device, hospital, medical device directive.

1 Introduction

The development of applications for smartphones and tablet PCs running operation systems like Apple iOS or Google Android attracts the interest in many fields of daily living [1]. In healthcare new applications fields like home telemonitoring or ambient assisted living have been based on mobile devices acting as remote terminals for medical data collection and intuitive user interaction [2], [3], [4]. At hospital wards smartphones and tablet PCs enable clinicians and nurses to treat patients more easily

K.S. Nikita et al. (Eds.): MobiHealth 2011, LNICST 83, pp. 211–216, 2012.

directly at their hospital beds or to share and present data to clinical personnel for decision making independently on their current location [5].

In this paper we present insights into the development process of a mobile Android-based in-hospital glucose management system for the treatment of acute ill patients with diabetes type 2. The system will provide decision support for insulin dosing and therefore falls within the scope of the revised medical device directive (MDD) [6] for software design and implementation. Thus, we focus our discussion on the design and development process of software as a medical device.

2 Method

This section describes the methodology used for the user-centered design phase of the application. It gives insights into the technical aspects of developing a Google Android-based mobile application and discusses typical questions related to the medical device directive.

2.1 Medical Device Directive for Software

The revised Medical Device Directive for Software came into effect in March 2010. Medical software which falls within the scope of the MDD has to comply with the same rules as medical devices do. The question that must be asked is: If we are going to apply a medical device, operated using a dedicated software component, to a human, does this device complies with the medical device directive? If yes, how can we prove it? Which standards do we have use? Risk analysis, change requests on requirements, software life cycle management and consequently stringent documentation of all activities have to be performed. It is necessary to identify the risks and the actions taken against the risks (for each crucial step). Following standards have been considered as relevant for the development of medical software which falls within the scope of MDD [6]:

- ISO 13485 standard defines the requirements for a quality management system for medical devices.
- IEC 62304 standard has emerged as a global benchmark for management of the software development lifecycle.
- ISO 14971 has traditionally been adopted as the base standard for risk management for medical devices and will also be used for software.
- IEC 62366 and IEC 60601-1-6 provide information about the application of usability engineering for software as medical devices.

The regulations of the medical device directive and their interaction are complex. Thus, an important question for every producer of software in the medical domain is: When is a software component considered as medical device? Often the answer is complicated but experiences show that the following definition is a good starting point. Software can be considered as a medical device for applications which:

- are explicitly mentioned in the medical device directives
- control or influence medical devices
- analyze patient data generated by a medical device and that will be used for diagnosis or monitoring
- are used for diagnosis or treatment of physical or mental diseases

Thus, the purpose of the application, which has to be defined by the producer, has a great influence on consideration and classification of a software application as medical device [6].

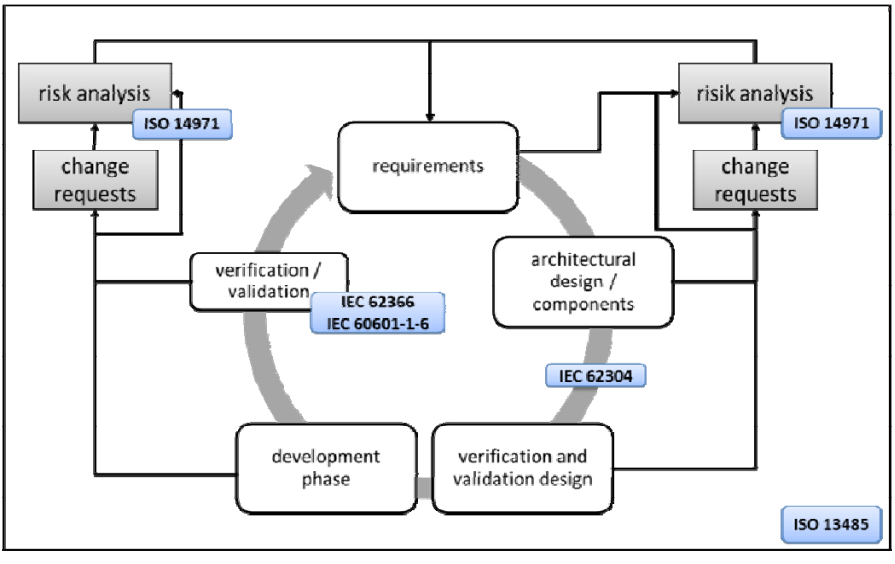

Fig. 1. A simplified graph, assigning relevant standards to the main development process components, of the mobile in-hospital glucose management system is shown in this figure

2.2 Case Study: Mobile Application for In-hospital Glucose Management

Design Phase. Referring to the requirements of the medical device directive (especially IEC 62366 and IEC 14971), which target usability of medical software and consequently the provision of clinical safety for users and patients, we have chosen a user-centered design approach.

A team consisting of technicians, software engineers, physicians and nurses from JOANNEUM RESEARCH and the Division of Endocrinology and Metabolism at the Medical University of Graz defined in weekly meetings the first version of the user interface and the functionality of the glucose management system. We used paper[1] and software mockups as trigger in order to have a basis for discussion and testing. FORTH-ICS provided external reviews for the design phase. We supported the design

[1] http://www.artfulbits.com/Android/Stencil.aspx

process by regular risk analysis sessions with all involved stakeholders. Derived risks have been collected and incorporated as change requests into the requirements. In the first iteration, we summarized the elicited requirements in an extensive specification document. A detailed discussion of the design phase can be found in [7].

Development Phase. Due to maintainability and expandability we decided to distinguish between an Android-based user interface and a platform independent backend (Java-based webserver) which contains business logic for the decision support, as well as the data storage and interfaces to the hospital information system. The exchange of data between the backend and frontend components requires mutual authentication and is completely done via encrypted web services to provide data security. The frontend application presents the data, received from the backend, in an appropriate manner to the user and collects new relevant data. The behavior of the frontend application relates in every step to the clinical workflow, which was identified together with end-users in the design phase.

During the development process of the frontend, as well as of the backend, we adapted Atlassian JIRA[2] as a requirement, issue tracker and task management tool for the documentation of each implementation step. JIRA acts as preparation for the needs of the IEC 62304 standard, and for ensuring an overview of open and already completed requirements, development tasks and identified bugs. In JIRA, each of these issues is assigned to an authorized editor who reports after finishing the issue. We connected JIRA to Atlassian Fisheye for source code management and Atlassian Bamboo for continuous integration and release management.

Verification Phase. In addition to the need of a detailed documentation, IEC 62304 and IEC 62366 demands also for verification and validation during the development process. Whilst we use TestNG[3] for the backend to verify the functionality after finishing the implementation of each system unit, we are testing the correct behavior of the frontend application on simulated user interactions. Android offers tools for instrumentation testing, which verifies, that the application prints out the desired output on the screen for every input. We used the free testing tool robotium[4] for the simulation of user interactions. Before we start frontend test cases the database is initialized with test case specific data through a separate web service. We use the Maven-Android plugin[5] as software project management tool for the generation of executable files. Maven for Android organizes application related dependencies and allows the automated execution of the frontend tests.

Thus, we created a flexible and configurable development and verification process, for the Google Android-based development environment, that provides a stable base for developing according to the regulations of the medical device directive.

[2] http://www.atlassian.com/software/jira/
[3] http://testng.org/
[4] http://code.google.com/p/robotium
[5] http://code.google.com/p/maven-android-plugin/

3 Results and Conclusion

Already in the early phase of the requirement elicitation process clinicians requested that they would prefer a software system that offers only the required base functionality and an easy to use interface, tailored to the current workflow. Usability of medical device software seemed to be one of the most important aspects, according to the avoidance of critical situations that harm patients or user. In order to avoid poor usability and consequently fulfill the requirements according to IEC 62366, we supported the physicians and nurses to design the user interface based on the established clinical workflow by their own, using paper mockups and functional prototypes. However, each ward in each hospital usually follows its own workflow. Therefore, great attention will be placed in future on the maintainability of the user interface. We are currently implementing a flexible dialog creation mechanism, which dynamically initializes Android dialogs based on a given parameter set. The type of parameter specifies what dialog is required (e.g. free text field, number picker). In a next step a configurable workflow engine component will be implemented for handling application sequences based on workflow patterns.

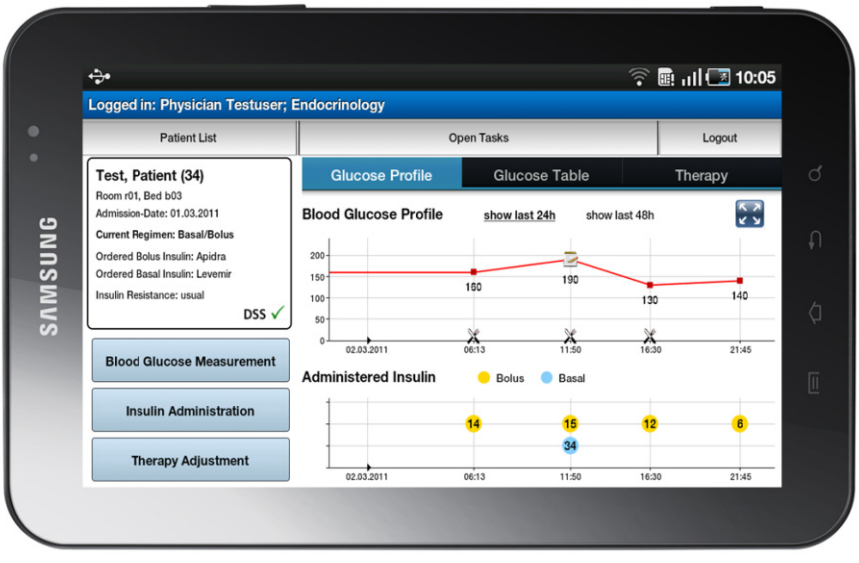

Fig. 2. This screenshot shows the main screen of the mobile glucose management application. The application runs on a Samsung Galaxy Tab with a seven inch touch screen and Android 2.2[6] as operating system. The main screen visualizes the blood glucose profile and the administered medication and provides the main functionalities of the application.

In addition to usability, other challenges appeared during the implementation process. The Android framework provides a wide range of available libraries but it is not explicitly designed to be used for medical applications. We are missing sufficient

[6] http://developer.android.com/sdk/android-2.2.html

security components that allow for secure transfer of patient data via web services. A secure SOAP client able to support encrypted communication and mutual authentication of both communication endpoints is currently under development.

The Android development platform provides a basic framework for developing software components according to the MDD directives while overcoming the various limitations on available libraries and, at the same time, offers to designers a powerful testing tool allowing to produce high level validation and verification tests for applications.

The decision support service for insulin dosing is based on a basal/bolus insulin regimen protocol, which has been developed with clinicians in parallel to the development process. Currently, this protocol is validated on paper in a clinical trial at the department of Endocrinology and Cardiology Medical University Hospital in Graz. In a next step, the validated protocol will be implemented into the electronic glucose management system for decision support and another clinical trial will performed. As precondition the system must meet the requirements of the medical device directive to get the approval of the Ethics Commission.

Acknowledgments. This work was partly funded by the E. C. under the 7th Framework Program in the area of Personal Health Systems under Grant Agreement no. 248590.

References

1. Keijzers, J., den Ouden, E., Lu, Y.: Usability benchmark study of commercially available smart phones (Internet). In: Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services - MobileHCI 2008, p. 265 (2008)
2. Kollmann, A., Riedl, M., Kastner, P., Schreier, G., Ludvik, B.: Feasibility of a mobile phone-based data service for functional insulin treatment of type 1 diabetes mellitus patients. J. Med. Internet Res. 9, e36 (2007)
3. Boulos, M.N.K., Wheeler, S., Tavares, C., Jones, R.: How smartphones are changing the face of mobile and participatory healthcare: an overview, with example from eCAALYX. Biomed. Eng. Online 10, 10–24 (2011)
4. Weaver, A., Young, A.M., Rowntree, J., Townsend, N., Pearson, S., Smith, J., Gibson, O., Cobern, W., Larsen, M., Tarassenko, L.: Application of mobile phone technology for managing chemotherapy-associated side-effects. Annals of Oncology 18, 1887–1892 (2007)
5. Biemer, M., Hampe, J.F.: A Mobile Medical Monitoring System: Concept, Design and Deployment. In: International Conference on Mobile Business (ICMB 2005), pp. 464–471. IEEE Computer Society, Sydney (2005)
6. Hall, K.: Devloping Medical Device Software to IEC 62304. European Device Technology 1(6) (2010), http://www.emdt.co.uk/article/developing-medical-device-software-iso-62304
7. Hoell, B., Spat, S., Plank, J., Schaupp, L., Neubauer, K., Beck, P., Chiarugi, F., Kontogiannis, V., Pieber, T.R., Holzinger, A.: Design of a mobile, safety-critical in-hospital Glucose Management System. In: Proceedings of MIE 2011 (2011)