II. Typical Mobile Application Failures

Quality assurance effectiveness can be enhanced by focusing quality assurance on a given input. Without knowledge of the concrete project context, typical mobile application failures and their origins can be crucial input, which can be used as the basis for a focused quality assurance method.

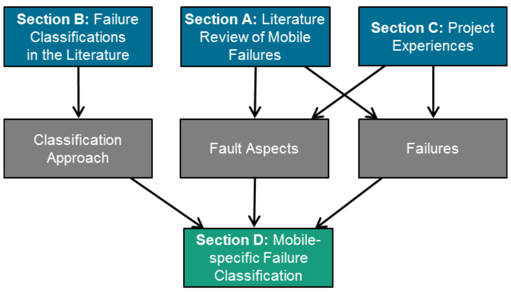

Fig. 1. Origin of the information used for deriving the classification

The origin of information used for deriving the failure classification is explained in three sections, see Fig. 1. Section II.E discusses the resulting classification by describing, among other things, the possible threats and constraints on the one hand, and the possible enhancement of quality assurance effectiveness obtained by focusing on the defined classes on the other hand.

With respect to the classification, this contribution uses the term *fault* as the origin of a failure, and the term *fault aspect* as the focus of a test case that leads to a specific failure of the mobile application.

A. Literature Review of Mobile Failures

The state of the art was captured by a literature review according to Kitchenham [3] to answer the question of which insights exist regarding failures of mobile applications.

The bibliographic database Scopus[1], which includes the contents of IEEE Xplore and ACM Digital Library, delivered 1001 results in the area of computer science published between January 2006 and January 2014. The search string to find relevant publications was based on *failure* and *mobile* related terms. No publication from the search results from 2006 and 2014 could be identified as relevant. From the database results, 26 publications could be identified to support this contribution. The information from these publications will be considered for the failure classification, including typical fault aspects. Other relevant publications also supported this contribution slightly regarding the detection of failures and fault aspects, but they described mostly similar insights.

Selected publications that support this contribution notably presented measurement-based failure characterizations of mobile phones and identified main failure types of mobile applications, explored failures reported in bug reports, or concentrated on concrete bug topics such as resource-constraints or leak triggers.

B. Failure Classifications in the Literature

Related work regarding the classification of failures (faults, errors, etc.) beyond the area of mobile applications was collected in the work of Mauser et al. [4] with a focus on human machine interfaces (HMI). Considering that mobile devices are also HMIs, the HMI-related publications are basically related to this contribution. There, the main classes for the failure classification are “behavior”, “design”, and “content”, which are divided into subclasses depending on the context. The classification of this work was encouraged by the Orthogonal Defect Classification (ODC) created in the early 1990s by IBM [5].

The IEEE Standard Classification for Software Anomalies [6] describes an approach based on lists of attributes for each failure report. The purpose is to “define a common vocabulary with which different people and organizations can communicate [...] and to establish a common set of attributes that support industry techniques for analyzing software defect and failure data” [6]. This is not a goal of this contribution, which focuses on a more lightweight classification approach. However, the failure attributes values in this standard contain the section “mode” with the possible values “wrong”, “missing”, and “extra”, which is considered as a convincing division for subclasses of mobile application failures.

C. Project Experiences

In addition to being based on the investigation of related work, this contribution is also based on the experiences made in several internal mobile application development projects for business use. In 2013, four mobile applications were developed within six months, whereupon the quality assurance experts of the development team analyzed the failure reports with the intention of creating a classification for failures and typical fault aspects. Each of the mobile applications – three Android and one iOS – are based on an average of ~10.000 lines of code and all have a backend binding. Quality assurance, especially testing, was performed continuously during the development process. Faults and failures found during the testing on the unit, integration, and system levels as well as during inspections and reviews were reported and documented in a bug tracking system. The reports were created by quality assurance experts (e.g., testers), developers (e.g., front- and backend programmers), and managers who accompanied the acceptance tests. In the end, the bug tracking database contained 101 reports with information about the faults, respectively failures, to be fixed.

D. Mobile-specific Failure Classification

Based on the reports of the project experiences, the quality assurance experts assigned categories to each report that fit the description of the report. This was used as starting point for deriving a classification for mobile application failures. The insights from the studies of related work and the quality assurance project experiences enabled the derivation of a failure classification, which includes a mapping to the typical fault aspects. As a first step, this classification was done for only one mobile application development. As part of the evaluation of the classification, it was applied to the other three mobile applications mentioned in II.C.

[1] http://www.scopus.com – SCOPUS Database

TABLE I. FAILURE CLASSIFICATION

Class	Subclass		Fault Aspects
Behavior	Transition	wrong	1d), 4c), 6b)
		missing	2a), 4c), 6a), 6b)
		extra	4c)
	Transaction	wrong	1a), 1b)
		missing	1a), 1b), 3c), 5a)
		extra	1a), 1b), 7c)
	Dynamic Content	wrong	3a)
		missing	2c), 5a), 7a)
		extra	7b)
	Event	wrong	1d)
		missing	5b), 5c)
		extra	2b)
Design	Interaction Element	wrong	1d)
		missing	4a), 4b), 6a)
		extra	6a)
	Style	wrong	4b)
		missing	-
		extra	-
Content	Static Picture	wrong	-
		missing	1c), 4a), 4b)
		extra	-
	Static Animation	wrong	-
		missing	1c), 4a), 4b)
		extra	-
	Static Text	wrong	3a), 3b)
		missing	3b) , 4a), 4b)
		extra	3b)

The dash in the column *fault aspects* of TABLE I. means that so far, no typical fault aspects of mobile applications have been identified. All the other entries of this column are links to the *ID* of TABLE III.

The main classes of the classification – see TABLE I. – were taken from the existing classification [4] outlined in Section II.B. The subclasses were derived based on the initially created categories for the reports described in Section II.C. The subclasses were each divided into three possibilities, based on [6] as mentioned in Section II.B. The subclasses of the three main classes *Behavior*, *Design*, and *Content* are defined as follows in TABLE II.

TABLE II. DEFINITION OF FAILURE CLASSES

Class Name	Subclass Name	Definition	Example
Behavior	Transition	Reaction of the user interface due to an interaction of a user.	Move to previous screen by pressing the back-button.
	Transaction	Transacted data that was sent or received by the mobile application.	Send notes to backend.
	Dynamic Content	Content that is not static but is entered or calculated during runtime.	Information about the time span since log in.
	Event	Incident that has no instant relation to a user interaction.	Amount of energy consumption.
Design	Interaction Element	Basic part of the user interface.	Labeled button.
	Style	Appearance of the user interface, respectively its interaction elements.	Shadows of text fields.
Content	Static Picture	Illustration that is not generated by the mobile application.	Profile picture of the user.
	Static Animation	Visual, respectively audiovisual sequence in motion.	Integrated video stream.
	Static Text	Textual content that is not calculated during runtime.	Help text.

These classes are orthogonal by definition. That means, there is no overlapping of the definitions regarding the addressed set of failures. Initial definitions of categories of fault aspects were created as well. These categories are linked to the failure classification based on the related work and the project experiences, including interviews with developers and testers. The categories of fault aspects are defined as follows:

TABLE III. CATEGORIES OF FAULT ASPECTS

Cat. Name	Fault Aspect	Description	ID
Connection	Network Disconnect	Temporary network disconnect.	1a)
	Network Change	Change between different networks like WiFi to mobile, mobile handover, WiFi to WiFi.	1b)
	Network Bandwidth	Use of different bandwidths due to availability of, e.g., 3G, GPRS.	1c)
	Location	Temporary GPS disconnect.	1d)
Energy	Low Battery	Low or critical battery status.	2a)
	Background Process	Energy consumption, e.g., due to memory leaks.	2b)
	Empty Battery	No available energy.	2c)
Setting	Localization	Settings regarding the location.	3a)
	Language	Settings regarding the language.	3b)
	Time	Different time or date settings of client and backend.	3c)
User Interface	Orientation	Change between vertical and horizontal orientation of the device.	4a)
	Screen Resolution	Change between different screen resolutions, respectively sizes.	4b)
	Native Button	Functionality of native buttons, such as back or home button.	4c)
Interruption	Incoming Call	Externally triggered calling event.	5a)
	Incoming Message	Externally triggered message event.	5b)
	System Message	Internally triggered message event.	5c)
Application Interface	Interface Consideration	Consideration of common application interfaces like email client and web browser.	6a)
	Interface Functionality	Functionality of common application interfaces like email client and web browser.	6b)
Data Usage	Kill Process	Application process termination.	7a)
	Permission Abuse	Possibility to access data without permission.	7b)
	Confidentiality	Functionality to send information accidentally.	7c)