QUALITY AND RELIABILITY ENGINEERING INTERNATIONAL
Qual. Reliab. Engng. Int. (2008)
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/qre.937

Research

Soft Reliability: An Interdisciplinary Approach with a User–System Focus

A. Koca*, †, M. Funk, E. Karapanos, A. Rozinat, W. M. P. van der Aalst, H. Corporaal, J. B. O. S. Martens, P. H. A. van der Putten, A. J. M. M. Weijters and A. C. Brombacher

Eindhoven University of Technology, P.O. Box 513, NL-5600 MB Eindhoven, The Netherlands

A recent trend in technological innovation is towards the development of increasingly multifunctional and complex products to be used within rich socio-cultural contexts such as the high-end office, the digital home, and professional or personal healthcare. One important consequence of the development of strongly innovative products is a growing market uncertainty regarding 'if', 'how', and 'when' users can and will adopt such products. Often, it is not even clear to what extent these products are **understood** ***and*** **interacted with** ***in the intended manner. The mentioned problems have already become an evident concern in the field, where there is*** **a significant rise in the numbers of seemingly sound products being complained about***, signaling a lack of*** **soft reliability.** ***In this paper, we position soft reliability as a growing and critical industrial problem, whose solution requires new academic expertise from various disciplines. We illustrate*** **potential root causes** ***for soft reliability problems, such as discrepancy between the perceptions of users and designers. We discuss the necessary*** **approach to effectively capture subjective feedback** ***data from actual users, e.g. when they contact call centers. Furthermore, we present a*** **novel observation and analysis approach** ***that enables insight into actual product usage, and outline opportunities for combining such objective data with the subjective feedback provided by users.***

KEY WORDS: soft reliability; customer-perceived quality; new product development; information flow analysis; process mining; business process design

1. INTRODUCTION

Rapid technological advancements, as well as prevailing social trends (such as an aging population and a stressed job market), create and promote new opportunities for incorporating technology within our environments in an innovative manner. This is especially observable in recently developed products

*Correspondence to: A. Koca, Eindhoven University of Technology, P.O. Box 513, NL-5600 MB Eindhoven, The Netherlands.
†E-mail: a.koca@tue.nl

Contract/grant sponsor: Dutch Ministry of Economic Affairs

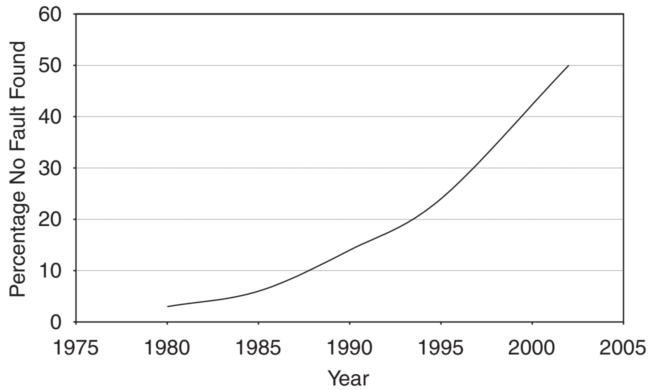


Figure 1. Rapid growth of 'no fault found' cases, where returned products seem to function correctly, in modern high-volume consumer electronics[1]

that display enhanced multi-functionality in complex socio-cultural settings, such as the high-end office, the digital home, and professional or personal healthcare. The downside is a growing market uncertainty as to 'if', 'how', and 'when' users can and will adopt such products: Many innovatively designed products, especially where ambient intelligence is involved, go so far as suggesting substantial changes in the everyday flow of people's lives; however, it is often highly uncertain to what extent intended users are able to *understand* and *interact with* such products in the intended manner, not to mention the difficulty in getting concrete ideas about how willing people are to accept (or embrace) such products. In fact, negligence of such issues has already become an evident concern in the field, where *a significant rise in the numbers of seemingly sound products being complained about can be observed* (Figure 1), signaling a crisis of lacking *soft reliability*[‡] [1–5].

The relevance of soft reliability is expected to grow with increasing product complexity, and hence with increasing product state spaces. In control engineering, the notion of a *state space* is used to characterize the behavior of a system as a set of *input*, *output*, and *state* variables (i.e. the state of the system can be represented as a vector within that state space). Owing to the exponentially increasing computer hardware capabilities (cf. Moore's law), the state spaces of digital electronic devices are growing faster and faster. This effect is further amplified by the merging of technologies, increasing networking capabilities, and by the fact that products nowadays offer a high degree of adaptability to accommodate many diverse user contexts and preferences[§]. Therefore, dynamically and sometimes unintentionally growing state spaces naturally imply an increase in uncertainty about the real field performance (e.g. compatibility in various use-contexts) of products. Accordingly, product specifications can no longer cover entire state spaces, and hence field incidents attributable to specification omissions (i.e. due to unexplored system boundaries) become more frequent than before. On the whole, *soft reliability issues observed in the field for many consumer electronics devices have already started to overtake the numbers of hard reliability problems arising from typical specifications violations*[6,7]. This trend confirms that the recently emerging issue of soft reliability is a serious problem today, and will be even more so in the future, unless the necessary know-how to tackle it is developed.

Soft reliability research is concerned with (i) understanding the sources of and then (ii) managing (i.e. forecasting, avoiding, tolerating, and remedying) user dissatisfaction with respect to the user-expected product capabilities, aesthetics, performance and usability. More specifically, it deals with analyzing user–system

[‡]Note that *soft* reliability should not be confused with *software* reliability.

[§]Adaptability of a product may lead to situations where two instances of the same product appear and behave in very different manners, e.g. hand-held computers equipped with different softwares and used for entirely different purposes (a mobile navigator, a mobile phone, etc.).

interactions that potentially reveal causes of *soft failures* (an exact definition of *soft failures* is given in the following section), thus devising a set of theories, models, methods, and tools to ultimately enable their effective and efficient handling in an industrial context. Therefore, it is natural to investigate soft reliability from two complementary perspectives: First, the *product-oriented* perspective focuses on user–system interaction and tries to identify soft reliability problems as they manifest themselves when product characteristics, which are consequences of explicit or implicit design decisions, are put to test in a realistic context of use. Second, the *process-oriented* perspective focuses on the business processes (before, during, and after product development), where the root causes of soft reliability problems, such as inadequate communication of feedback or lack of user-centeredness, need to be established. The focus of this paper is on soft failures at the user–system level (i.e. the *product-oriented* perspective). In particular, we will report several early findings from an ongoing multidisciplinary collaboration that addresses different aspects of soft reliability, characterized by a range of questions and proposed approaches:

(a) How is it possible to account for non-instrumental quality values (e.g. emotional, societal) of a targeted audience early on in the design of products?
(b) How can unexpected user or system behavior be captured and understood from instrumental and reported data obtained from real usage? How can such information be fed back to the product development processes?
(c) How can observation units within products help in gathering objective usage data that are specifically oriented toward extracting soft-reliability-related information?
(d) How can logged data from *field-feedback processes* and from *product usage processes* be processed and presented in such a way that information contributing to our understanding of soft reliability becomes evident?

The paper is organized as follows: Section 2 provides necessary background on soft reliability. Section 3 discusses an experiment that reveals potential root causes of soft reliability problems by evidencing a gap between designers' views on users' perceptions and actual users' perceptions (a). Section 4 elaborates on the necessary approach to adopt in collecting and processing *subjective* feedback data from actual users (b). Section 5 describes a novel approach that can be used to trace and analyze user behavior during actual product usage, i.e. the collection and processing of *objective* usage data (c and d). Section 6 concludes the paper and identifies directions for future work (e.g. possibilities to combine the *objective* and the *subjective* data to obtain a more complete view on the overall usage process).

2. BACKGROUND ON SOFT RELIABILITY

Formerly, failures encountered in products were mostly technical component failures of hardware or software that displayed explicit violations of product specifications[8], and hence which could be linked back to product implementation or manufacturing processes during development. However, within the current, rapidly evolving market circumstances, new classes of failures that cannot be traced back to product implementation or manufacturing processes are increasingly being observed in the field. At present, these additional classes of failures (attributable to specifications omissions, usability/learnability problems, customer misunderstandings, or specific use context) are not identifiable and manageable within existing industrial quality and reliability information flows[1,3,5,9]. This is related to the fact that, currently deployed quality and reliability information flows exist mostly between 'product aftercare' activities and the 'product implementation-manufacturing' activities of new product development (NPD) processes; but hardly ever between 'product aftercare' activities and the front-end 'product design development' activities of NPD processes. Accordingly, the structure and content of the data transferred via these flows are not compatible with the nature of the new classes of failures. The end result is many 'product assistance' calls at call centers, 'No Fault Found' labeled products at service/repair centers, returned products at dealers that function well, and to top it all, disconfirmed expectations of unhappy customers yielding damaged brand image of companies. The

DOI: 10.1002/qre

realization of this trend (Figure 1) was the starting point for making a distinction between *hard* versus *soft reliability* concerns[1,2,4,7].

The distinction we make between hard failures and soft failures is as follows[7]. *Hard failures* are product failures where the product is incapable of performing its functions, as listed in its technical specifications, without the intervention of authorized technical support for recovery by means of repair or replacement of parts. On the other hand, *soft failures* are product failures where the product, despite being capable of performing its functions as listed in its technical specifications, still necessitates professional intervention for recovery (but not repair), through instructions or information, from an unexpected state of user–product interaction[¶].

Previously mentioned NPD, the outcome of which establishes new products in the market, is an interdisciplinary domain; and so is soft reliability, which is involved therein. In the NPD context, soft reliability specifically concerns (i) product 'quality and reliability', which is largely determined by (ii) 'customers in the market', that is

(i) 'Quality and reliability' of a product is concerned both with the hard *engineering* of it to meet technical product specifications, and the soft *design* of it (including the organizational design decisions to include/exclude certain functionalities or use of materials) to meet user requirements. Thus, the *design* of the product is concerned with the user in terms of her socio-cultural context and hence related expectations. Generally speaking, hard *engineering* is about making the product right, whereas soft *design* is about making the right product (with the required functionalities, ease of use, appeal, etc.).
(ii) Getting to know the 'customers in the market' requires getting to know 'customer segments' in an 'evolving market', both of which relate to the marketing domain. Knowledge of these aspects contributes to an improved identification of customer needs and expectations, and to a better positioning of new products within a dynamic market. In addition to these presales concerns, marketing also involves handling post-sales/aftercare services, such as minimizing customer dissatisfaction, which requires knowledge of consumer complaint behavior and the key points of what is referred to as customer relationship management (CRM).

A good understanding of all related fields is necessary in order to be able to manage soft reliability, which currently presents a growing, uncontrollable problem in NPD. The findings of soft reliability research are intended to be beneficial: (i) for the user, in that the confusion due to the discrepancy between expectations and real product capabilities is minimized; (ii) for the manufacturer, in that product investments are made in the correct and relevant areas (i.e. not for product functionalities that users do not seem to be interested in); and (iii) for the impetus of technological innovations and their acceptance by society, in that the convenience offered is perceived as greater than its burdens. Consequently, it is relevant for both academia and industry: While expanding the scope of the current quality and reliability approaches to systematically cover user-centered failures, industry is offered efficient and effective ways to cope with an increasingly threatening problem.

3. WHERE ARE SOFT FAILURES ROOTED?

The emergence of soft reliability has shifted the emphasis of product quality and reliability from engineering back to (the preceding) design of the product. Recent research in soft reliability pointed to the crucial importance of the initial concept design phase in minimizing the occurrence of soft failures[3]. As usually there is not much effective communication between the front-end concept design activities of most NPD processes

[¶]For instance, applying an in-house developed software patch as an end-user, provided proper resources (e.g. instructions and information), is also considered a *recovery* case and not a *repair* case as the actual repair is done in-house. Hence, it is a case of a soft failure, conforming to the definitions.

DOI: 10.1002/qre

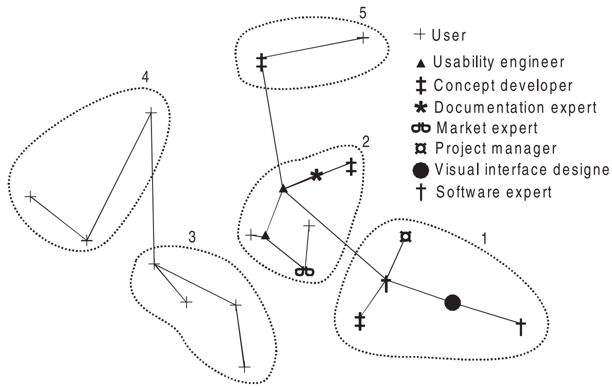


Figure 2. Two-dimensional visualization of dissimilarities between designers' and users' perceptions and hierarchical clustering (minimum variance)

and, in the end, the users of the resulting products, more research is needed to explore the compatibility of the perceptions that are in effect at both ends.

Based on the number of interviews with stakeholders in the concept design practices in a multinational printing company, we observed that it is often difficult to trace back the reasons that motivated certain design decisions. The questions raised then were on what basis are design decisions made? Can designers really foresee user preferences? To answer these questions, we conducted an experiment where designers' views on users' perceptions of a product were compared to actual users' perceptions[10]. Eleven employees of the multinational printing equipment manufacturer, all involved in the concept design phase of the TouchToPrint[‖] product, as well as eleven potential users that had no prior experience with the product, were interviewed using the Repertory Grid technique[11]. During the interview, participants (i.e. designers and users) were first asked to compare the product to five competitor products. This process resulted in a list of attributes (e.g. modern–traditional and fast–slow) that participants used to differentiate between the products. Subsequently, each participant was asked, in a product-rating procedure, to compare all products on his/her own elicited attributes as well as on overall measures of *preference* and *dissimilarity*[10]. As a result, the Repertory Grid process yielded three kinds of data: *dissimilarity*, *preference*, and *attribute* data.

To explore the discrepancies between designers' and users' perceptions, we defined the perceptual distance D_{ij} between participants i and j based on dot-product correlations R_{ij} of the k dissimilarity scores (**1**). Derived distances were then visualized in two dimensions (Figure 2) using the MDS tool XGms[12]. The two-dimensional visualization was judged as adequate (stress value $S=0.18$). Hierarchical clustering (with minimum variance) revealed two main clusters that can be further subdivided into five relatively homogeneous participant groups. Groups 3 and 4 consist entirely of users, whereas groups 1, 2, and 5 consist mostly of designers. Identification of the designers revealed that group 1 consists mostly of technically-oriented designers, whereas group 2 consists mostly of user-oriented designers

$$D_{ij}=1-R_{ij}^2, \quad R_{ij}=\frac{\sum_k D_i(k)\cdot D_j(k)}{\sqrt{\sum_k D_i^2(k)\cdot D_j^2(k)}} \tag{1}$$

where $D_i(k)$ is the dissimilarity score given by subject i on kth product comparison (k sums over all product comparisons).

To acquire richer insight into the ways in which designers and users differed, we focused specifically on a comparison between two of the six products: *TouchToPrint* and *Badge*. Figure 3(a) illustrates the reasons supporting preference for TouchToPrint over Badge, as it shows the number of attributes that are

‖TouchToPrint is an alternative to other user identification mechanisms such as PIN-code or Badge.

DOI: 10.1002/qre

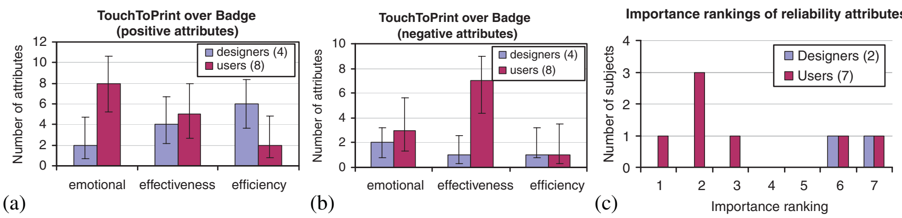


Figure 3. Attributes (a) positively ranked and (b) negatively ranked when TouchToPrint is preferred (along with 95% exact confidence intervals). (c) Importance rankings for reliability attributes for designers and users

positively ranked when TouchToPrint is preferred. Although users' most frequent reason for preference of TouchToPrint was related to emotional attributes, for designers it was efficiency attributes. All attributes in the effectiveness category were related to security. TouchToPrint was perceived as more secure than Badge by both designers and users. Users' most frequent negative concerns, shown in Figure 3(b), were related to reliability (five out of the seven effectiveness attributes had to do with reliability). This is also evident in Figure 3(c) where we can observe that only two designers expressed reliability concerns and ranked them as the sixth and seventh most important attributes, whereas five users ranked reliability within their three most important concerns. Hence, although most users preferred TouchToPrint, they had some concerns that can potentially turn into failures to motivate use.

The experiment seemed to corroborate our initial hypothesis: Designers could not predict users' responses to the product that they were designing. Although designers seemed to predict the different concerns that people would have with the product, they highly mispredicted the importance that users attach to each concern. Further, the position and background of each designer seemed to impact their sensitivity to users' concerns. While market experts and usability engineers seemed to have higher sensitivity to users' concerns due to the high exposure to users' responses, designers with technical roles (i.e. receiving less exposure to potential users) seemed to have higher misprediction rates.

The results of this experiment can be read from two different, but in our view complementary, perspectives. On the one hand, they point to the importance of user involvement in the early design phases, for the products' acceptance in the market. This view brings the requirement of *iterative* and *scalable* product evaluation practices within the NPD process. On the other hand, it brings awareness of designers' inability to foresee real users' preferences and behavior. The latter implies the need for rapid prototyping practices to enhance the ability to learn from the field. We believe that both tracks need to be pursued for managing soft failures in highly uncertain markets.

4. IN WHAT FORMS ARE SOFT FAILURES REVEALED?

Soft failures crop up due to a combination of factors. From a high-level managerial perspective, the root cause can be seen as the conflict between the perspectives of the developer, manager, and the user, which is surfaced by the rapidly changing market conditions: Products with *growing degrees of innovation* are developed and put out on the global market in *decreasing time periods* to meet the *more demanding expectations* (e.g. with respect to extended warranty periods) of a *variety of customers*. In this setting, (i) the *developer* focuses on implementing the most recent technological advancements in a product correctly; (ii) the *manager* focuses on launching a reliable and profitable product on the global market at a speed exceeding that of the competitors, with the concern of maintaining customer satisfaction before and after sales; and (iii) the *user* focuses on fulfilling her expectations about the product she chooses to invest money in. The results of the experiment presented in Section 3 is a fine example of the discrepancy between the designer's and the user's views.

DOI: 10.1002/qre

From a slightly lower-level design perspective, the roots of soft failures can naturally be attributed to specific decisions taken during the concept design of a product. However, consistently attributing *each* soft failure to a *particular* cause requires a means that enables a deeper level of analysis. First and foremost, such an analysis requires a clear understanding of the fundamental nature of soft failures, which in turn requires a thorough knowledge about the failed state during user–system interaction, because the actual revelation of soft failures occurs only during use of the product in its specific use context. Then, all identifiable properties of soft failures need to be synthesized in a structured way to relate observable failure *symptoms* to their respective *diagnosis*.

In the case of hard failures, due to their well-documented reference point of technical specifications and hence solely product-centered manifestation, it is comparatively simple to identify them and trace their fault → error → failure chains. However, this is not the case with soft failures, due to their negated stance with respect to technical specifications (i.e. they are not covered by technical specifications) and also their user-centered manifestation. This difficulty, with the human factor involved, has already been expressed in the wide-ranging related literature (e.g. systems engineering and software engineering), and the need for developing methodologies for systematic identification and improvement of user–system interaction failures has been explicitly underlined[13].

Partly as an attempt to address the above-mentioned necessities and difficulties, we developed a failure classification model[6]. This model acts as the means to analyze soft failures by taking into account both the actual (vs the ‘assumed’) product capabilities and the phases of use, specifically accounting for the use context. By phases of use we mean the phases that need to be accomplished consecutively by the user in order to ensure that the product gets successfully communicated. These phases, as defined in Reference[14], are (i) appropriate commercial product exposure and communication, (ii) creating user awareness for the (many) features and functions the product has, (iii) making these functions appealing for the user in order to motivate use, (iv) making the product intuitive to use, to ensure (v) acceptance of the product through satisfaction, and (vi) adoption of the product to the user’s daily life, for extended use. Additionally, the analysis offered by our model is designed to fit the conventional failure mechanism framework that complies with the ‘fundamental chain’ of fault → error → failure (i.e. cause → state → event)[13,15,16]. This approach is adopted to render our model compliant with the widely acknowledged fundamental concepts and taxonomy of dependable computing[13]. Consequently, we highlight the parallelism between failure mechanisms in computing sciences and in NPD, and hence encourage the development of respective fault forecasting, avoidance, tolerance, remedy techniques for soft reliability, which is part of our ongoing work.

To date, the industrial applicability of our model has been tested by classifying data from different feedback sources (i.e. call centers, service centers, Internet forums and user tests) regarding various innovative electronics products of multinational companies, and our findings reveal promising improvements as to the effective capture of classes of soft failures. An earlier pilot study, the details of which can be found in Reference[5], is summarized in what follows: In Figure 4, the classification results of 2321 calls to a call center, done in the call center itself using their own classification model (left), are compared with the classification results of the same 2321 calls, using an earlier version of our model (right). Note that the company’s own model is overpopulated with many categories, which makes it impracticable to be used correctly and consistently by the human classifiers (i.e. call center agents). In Figure 5, the classification of 1368 calls to another call center, done in the call center itself using their own classification model (left), is compared with the classification of the same 1368 calls, using the same earlier version of our model (right). Note that the company’s own model has the problem that there is only one category, namely ‘product assistance’, which is being used. The problems encountered in both cases stem from ineffective ways of classifying failure data, which has been confirmed by the respective companies who are currently in the process of exploring improved approaches based on our recommendations.

To conclude, earlier on, we stated that the actual revelation of soft failures occurs only during use of the product in its specific use context (i.e. in contrast to simulated failures at premeditated user tests). Therefore, actual subjective user feedback about the failed interaction is crucial to identify any mismatch between

DOI: 10.1002/qre

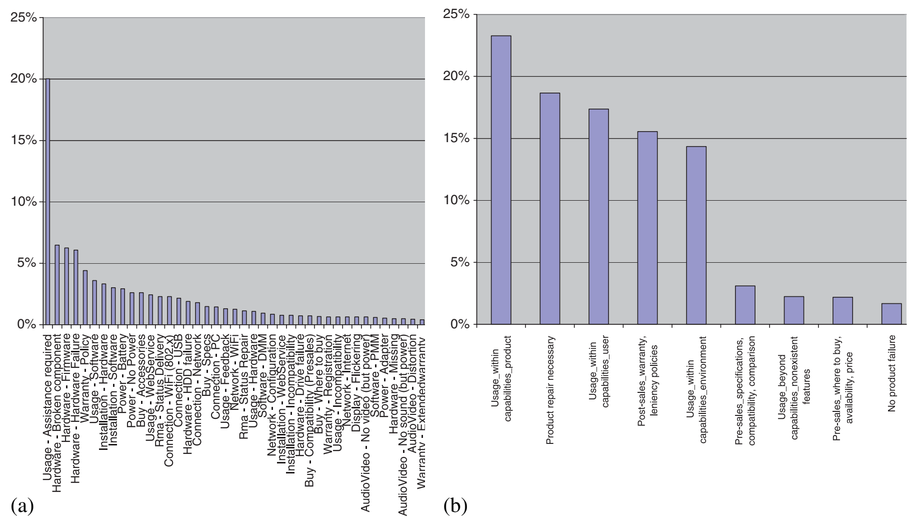


Figure 4. Percentages of call reason categories: (a) first company's own model and (b) our failure classification model. The large number of categories in (a) render the company's model impracticable to be used correctly and consistently by the call center agents, making the classification less useful

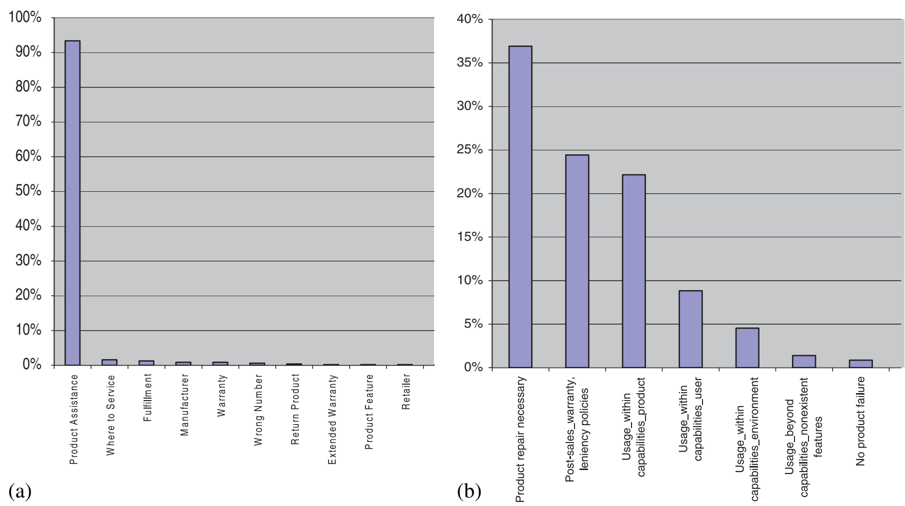


Figure 5. Percentages of call reason categories: (a) second company's own model and (b) our failure classification model. A highly used category in the company-owned model, 'product assistance', makes it impossible to gain deeper insight into the actual call reasons

(latent or explicit) user expectations and actual product specifications. However, the empirical data presented show that current industrial systems to collect user feedback from the field miss soft reliability altogether, contributing to increasing numbers of 'no fault found' cases. To this point, field feedback has remained an untapped resource of invaluable information to be utilized during NPD processes. As each contact with a customer is an opportunity to discover the 'basic', 'performance', and 'excitement' needs[17] of (potential) users, we work towards operationalizing our generic failure classification model for effective gathering of field feedback to facilitate dynamic improvement of soft reliability.

DOI: 10.1002/qre

5. HOW CAN SOFT FAILURES BE TRACED?

The successful classification of soft reliability problems reported in the field (cf. Section 4) is necessary to effectively handle customer dissatisfaction and also to enable a feedback process to the development team for improving products. However, the available data sources from service centers, call centers, the Internet forums, and dealers (e.g. about returned products) are typically very limited, i.e. they lack important context information that is needed to track down the root cause of a problem, leading to many 'no fault found' cases. Furthermore, due to a gap between designers' and users' perceptions of a product (cf. Section 3), problems often appear in user–system interaction, as users may behave in ways that designers had not anticipated. Therefore, *we suggest that objective information about real user actions on products can help to bridge this gap. Being able to reproduce experienced problems should make it easier to devise solutions that effectively remove such problems.* For this, usage data need to be collected directly by the product and processed in a structured and automatic manner. This may be applied in different scenarios that require different levels of data collection and transparency:

- during sessions of early user involvement, e.g. experiments with prototypes in an early phase of the NPD process;
- as part of the post-sales service process, e.g. as a remote diagnosis instrument while the user is in contact with an agent in the call center; or
- in the context of a more general (voluntary) feedback process, e.g. regular usage reports from motivated customers, stimulated by monetary incentives.

In what follows, we first present a flexible usage observation framework that can be used to support all these scenarios. Data collection can be specified by the actual stakeholders involved in product development and the observation framework automatically collects the data from product instances in use. Furthermore, we give an outlook on possible approaches to evaluate these data in a structured and automated form.

5.1. Observation approach

In principle, by observing users interacting with a product, two kinds of data can be captured: *subjective* data about usability and *objective* data about actual usage patterns. Customarily, such data are only collected in special testing environments. A drawback is that simulated settings in usability labs may bias the outcome of product research as, for instance, users will feel supervised and thus act differently than in real situations. In order to get more reliable and objective usage information, we propose an approach to *make products self-observing* and let them capture usage within the user's habitual environment.

The objective is to get usage information directly from within a product; however, what is mostly provided by products is data in the form of various, rather simple, events. What is missing is semantic coherence and meaning. Examples of such simple events are key presses, user-interface events, and performance samples of system components. Information at this level cannot directly be used for further analysis, as for that semantically rich information is required. To deal with this, a semantic layer has been established by combining the provided low-level product events to *complex events* (cf. Reference[18]). These events capture higher-level processes that are triggered in the product and therefore support understanding of the actual usage. In addition, it enables further (i.e. automatic) analysis as we will show in the second half of this section.

The basic idea of self-observing products is to equip commercial digital electronics products with an additional internal software component. This *observation component* can sense information, process it, and transmit captured data finally to a central instance. So-called *hooks* are places in the hardware or software where events can be observed, which are suitable as sources of observation data. Information about the event or just the fact that the event occurred forms a *simple event* and is routed to an event processing component that is responsible for the proper handling of product events. Here, more complex events are constructed by correlating low-level product events from different sources spatially and temporally. Those aggregates are described in the form of complex events annotated with semantic information provided by

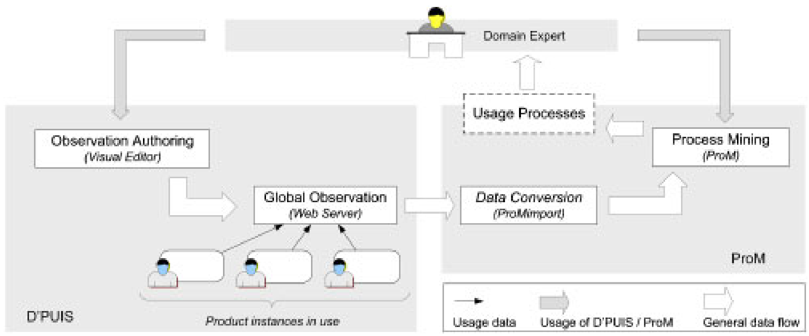


Figure 6. System overview: D'PUIS and ProM

respective stakeholders. Thus, the system generates meaningful information that can be analyzed and reused seamlessly. A small example will be provided later.

The approach outlined above has been realized as the D'PUIS framework that has been used in a consumer electronics product that is connected to the Internet[19]. This framework enables one to create a large network of distributed self-observing products. Product instances in use (Figure 6) are first instructed what to observe; then, during the actual data collection, they send their information up to the global observation unit. This global unit stores the collected information in a database, which can subsequently be accessed by the ProM*import*** [20] tool in order to convert the data for automatic analysis.

Regarding the architecture and development effort to build a system that is self-observing, it is obvious that a mismatch exists between interests of various NPD stakeholders, e.g. people who are interested in usage problems and people who have to build the product. A possible stakeholder involved in NPD is, for example, a knowledge engineer, who is responsible for evaluating the feedback from the field, to predict and measure the product quality and to provide information that can lead to improvements for future product generations. Domain experts such as the knowledge engineer are precisely those people who will be interested in eventually analyzing the usage data collected in the product. However, as they should not be expected to program the observation logic themselves, they would need to be able to specify exactly to software engineers what data should be observed when.

The resulting communication overhead and inflexibility can be avoided by our D'PUIS framework in two ways. First, our D'PUIS framework deploys a visual approach for programming (Figure 6), which enables various types of domain experts without deep technical knowledge of the product architecture to specify complex events that will provide the basis for the analysis. Second, on the developer side, only the hooks need to be specified at design time. As soon as they are implemented, complex events can be specified via the visual editor, which results in a remote change of observation in the products. This has the advantage that observation logic can be modified at any point in time without changing the product itself.

The architectural and development-wise bottleneck of product observation is clearly the integration into products and the connection to product hooks that supply the required usage information. Therefore, the most efficient solution is to keep the number and assignment of product hooks stable and emphasize the flexibility of routing and assignment of complex events.

Figure 7 depicts a screenshot of the visual editor, which helps to define what shall be observed in the form of product events and how these events will be processed. It is a simplified way to program the network of distributed observation units. Technical matter is hidden and simplified as building blocks such as event sources, processing nodes, and a global event sink (triangular box in Figure 7) that directly routes incoming data to a central database. The editor represents the flexibility to dynamically change how complex events

**Software and documentation (including source code) are freely available at http://promimport.sf.net.

DOI: 10.1002/qre

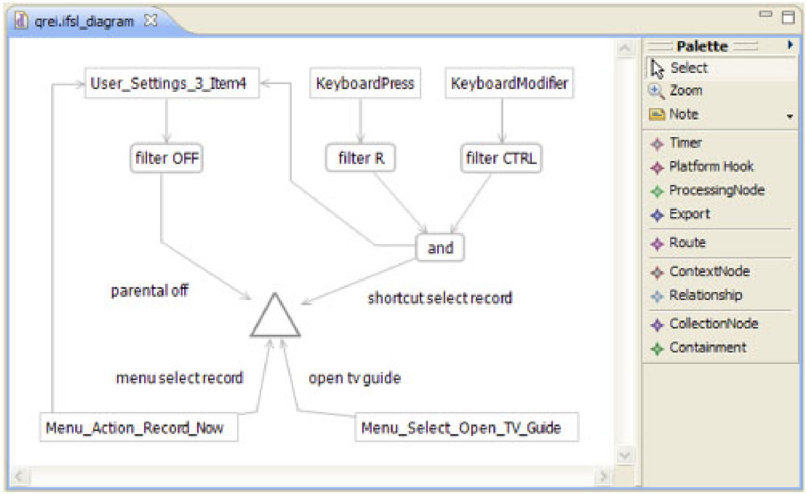


Figure 7. Visual editor showing the observation specification for the example scenario in the text

Table I. Experimental data coming from observed products using observation modules, which have been instrumented with the above-mentioned observation specification (cf. visual editor in Figure 7)

User_id	Task	Task data	Timestamp
1	Parental off		2007-08-31 17:01:01
2	Parental off		2007-08-31 17:01:23
1	Open TV guide		2007-08-31 17:02:06
3	Open TV guide		2007-08-31 17:02:07
4	Open TV guide		2007-08-31 17:02:15
2	Open TV guide		2007-08-31 17:02:55
3	Menu select record	10 p.m. news	2007-08-31 17:03:02
4	Shortcut select record	10 p.m. news	2007-08-31 17:03:04
1	Menu select record	10 p.m. news	2007-08-31 17:03:34
4	Parental off		2007-08-31 17:05:23
2	Shortcut select record	10 p.m. news	2007-08-31 17:06:37
3	Parental off		2007-08-31 17:06:41

are constructed and how they are routed within the observation system. This allows stakeholders to define their own complex events that are later identified and can be mined in an efficient manner.

As a working example, we present the case of a hard-disk recording device that allows one to record preselected TV programs. In our scenario, the users are instructed to record the 10 p.m. news, initially restricted by parental control, which simply blocks recording after 9 p.m. In order to achieve her task, the user has to disable the parental control, open the TV guide, and record the program. The observation specification used in the example (Figure 7) shows that four complex events ('parental off', 'shortcut select record', 'menu select record' and 'open TV guide') are generated by combining simple events from five hooks in the product (rectangular boxes). The combination of information can be filtering or the correlation of concurrent events with a Boolean 'and' operation. Whenever the 'shortcut select record' or 'menu select record' event occurs, the parental control settings hook ('User_Settings_3_Item4') is automatically triggered, i.e. the current state of those settings is retrieved and filtered. Only if the state then evaluates to 'OFF' the 'parental control off' complex event is generated.

Table I shows an excerpt of logged data, which was created using the observation specification. Complex events combining several data sources result in log data that are stored in a very simple format (e.g. user, task, [data], and time). Four users performed the experiment task almost simultaneously as the 'timestamp'

DOI: 10.1002/qre

data indicate. The ‘task’ column in the log data shows that the routes leading to the export node in the observation specification (the triangle in Figure 7) directly determine the contents of the resulting log data. This short excerpt of example data represents four slightly different approaches to perform the task, pointing at the complexity of usage data coming directly from products. The example incorporates only a few hooks, but potentially many more simple events are suitable for observation, for instance, system status information, various input device events, or even contextual data. This emphasizes the need for elaborate analysis tools that enable the detection of the different processes from within hundreds of thousands of such log entries in real-life cases. However, the storage and access to collected usage information inside products are standardized. Therefore, the whole workflow from data collection, aggregation, and/or filtering to the analysis can be automated.

5.2. *Data analysis*

With the storage of usage data in elementary (simple events) and aggregate (complex events) form in the database of the observation system, relevant information can be directly evaluated. For example, simple frequency measures (e.g. ‘How often was the recording started via the menu rather than the shortcut?’) can be easily extracted. However, to gain deeper insight into the user–system interaction, further analysis of the users’ behavior is needed. In the remainder of this section, we outline how *process mining* can help to fill the gap between raw log data and knowledge about the usage process.

Process mining is a relatively young field of log-data analysis techniques that have been successfully applied to many real-life logs from, e.g. hospitals, banks, municipalities, etc. (cf. References[21,22] for example). The basic idea of process mining is to discover, monitor, and improve real processes (in contrast to assumed processes) by extracting knowledge from *event logs*. Today, many of the activities occurring in processes are either supported or monitored by information systems, such as enterprise resource planning, workflow management, or CRM systems. However, process mining is not limited to information systems and can also be used to monitor other operational processes or systems, e.g. complex X-ray machines, high-end copiers, web services, careflows in hospitals. The common denominator in the various applications of process mining is that *there is a notion of a process* and that *the occurrences of activities are recorded in so-called event logs*. Assuming that we are able to log events, a wide range of process mining techniques comes into reach: we can use process mining to (1) *discover* new models (e.g. constructing a Petri net that is able to reproduce the observed behavior), (2) check the *conformance* of a model by checking whether the modeled behavior matches the observed behavior, and (3) *extend* an existing model by projecting information extracted from the logs onto some initial model (e.g. show bottlenecks in a process model by analyzing the event log).

To enable the application of process mining techniques to the product logs recorded by the D’PUIS framework as shown in Figure 6, we first need to convert the collected log data to the mining XML (MXML[††]) format, which is used by ProM[‡‡] [23], our process mining tool. Therefore, we developed the D’PUIS plug-in of the ProM *import* framework, which facilitates log conversion tasks. Figure 8 depicts a fragment of the MXML log for the hard-disk recording example from Table I. One can see that a *process* (here the experimental process as a whole) contains several *process instances* (the different users participating in the experiment), which in turn consist of a number of *audit trail entries* (the events or process steps that are logged). Each event carries a *name* (e.g. ‘parental off’) and a *type* (here always ‘complete’), and potentially a *time stamp*, a *performer* (not used here), and additional *data* (e.g. ‘10 p.m. news’).

An important area in the field of process mining is ‘control-flow discovery’. The goal of control-flow discovery is to automatically construct a process mining model showing the causal dependencies between activities in the process. This is not only challenging but also interesting as it immediately provides an overview about the actual flow of the process. Many different approaches have been proposed in the literature

[††]The MXML schema definition and further information on process mining research can be found in our website: www.processmining.org.
[‡‡]Software and documentation (including source code) are freely available at http://prom.sf.net/.

DOI: 10.1002/qre

```
<Process id="0" description="In an experiment the goal is to record the 10pm news.
Four different users approach this task in a slightly different way.">

  <ProcessInstance id="0" description="user 1">
    <AuditTrailEntry>
      <WorkflowModelElement>parental off</WorkflowModelElement>
      <EventType>complete</EventType>
      <Timestamp>2007-08-31T17:01:01.000+02:00</Timestamp>
    </AuditTrailEntry>
    <AuditTrailEntry>
      <WorkflowModelElement>open TV guide</WorkflowModelElement>
      <EventType>complete</EventType>
      <Timestamp>2007-08-31T17:02:06.000+02:00</Timestamp>
    </AuditTrailEntry>
    <AuditTrailEntry>
      <Data>
        <Attribute name="task data">10 pm news</Attribute>
      </Data>
      <WorkflowModelElement>menu select record</WorkflowModelElement>
      <EventType>complete</EventType>
      <Timestamp>2007-08-31T17:03:34.000+02:00</Timestamp>
    </AuditTrailEntry>
  </ProcessInstance>

  <ProcessInstance id="1" description="user 2">
  ...

</Process>
```

Figure 8. Fragment of MXML log for consumer test example in Table I

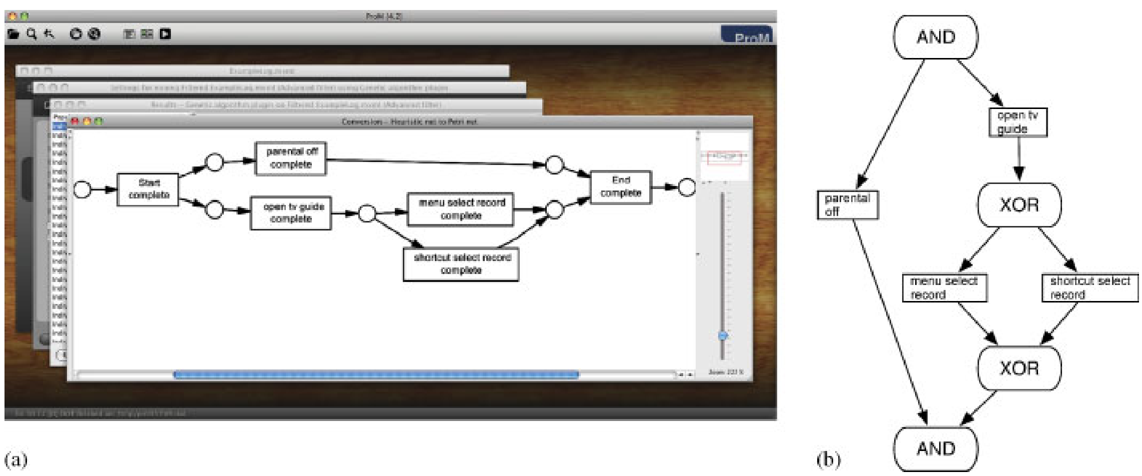
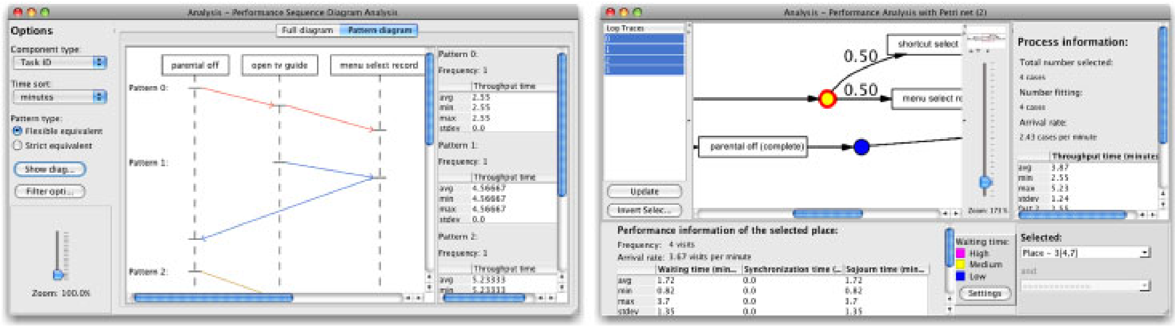


Figure 9. Discovered process model (a) in ProM and (b) conceptional visualization

(cf. Reference[21] for further references). For example, Figure 9(a) depicts the result of the Genetic Miner in ProM 4.2 based on the event log from Figure 8 in the form of a Petri net model. The same process is depicted in Figure 9(b) in a simpler notation for illustration purposes: the AND and XOR nodes are routing nodes (activating or synchronizing branches), whereas the other, rectangular nodes form the steps, or activities, in the recording experiment process. One can see that in the beginning of the process two parallel branches are started (both activated by the top-most AND node). Note that parallel activities are not causally related and can be executed in any order. The left branch only contains the task ‘parental off’. The right branch requires to first complete the task ‘open TV guide’ before either ‘menu select record’ or ‘shortcut select record’ can be performed. Both branches are required to finish before the process is completed (synchronized by the bottom AND node). Note that all the four scenarios from Table I are incorporated in the discovered process model. Their variations are captured by the choice (XOR) and the detected parallelism (AND).

Although the control-flow perspective is a very important view on a process, there are also other interesting perspectives, such as the organizational perspective, data, and time. For example, Figure 10 depicts two screenshots of analysis plug-ins that make use of the time information in the log: Figure 10(a) visualizes

DOI: 10.1002/qre



(a) (b)

Figure 10. Screenshots of further analysis plug-ins in ProM: (a) sequence diagrams and (b) performance analysis

the observed patterns as a sequence diagram (similar to a UML sequence diagram) and provides statistics on their frequency and duration; Figure 10(b) highlights bottlenecks in the discovered process model based on an evaluation of the timestamps in the log (most time was spent between ‘open TV guide’ and either ‘menu select record’ or ‘shortcut select record’, which might hint at a usability problem). Note that there are many more mining, analysis, and visualization tools available in ProM, and further research is needed to determine which of them are most relevant to gain insight into soft failures. For example, we might want to compare the actual user behavior in such a consumer test scenario with the expected interactions. Conformance checking techniques[24] can then be used to visualize and measure potential deviations.

Note that the automatic construction of process models for task analysis is not new[25]. However, the techniques described in Reference[25] are very basic as they, for example, do not consider parallelism. Overall, process mining techniques seem very suitable for the analysis of product logs as recorded by the D’PUIS framework. Furthermore, with the ProM (and ProM*import*) framework a powerful tool set is available, which can also be easily extended to, e.g. address certain domain-specific needs.

6. CONCLUSIONS

In the past, both the quality and the reliability of a product were mainly determined by the product’s compliance with its technical specifications. Customer satisfaction came in secondary, as a nice-to-have dimension of quality. Due to the conditions of today’s competitive global market, customer satisfaction through ensuring *user-perceived* quality and reliability has become the foremost concern. Accordingly, products need to be developed to comply not only with their technical specifications but also with explicit customer requirements and implicit expectations. This shift of focus in the assessment of quality and reliability of products makes it important to distinguish between hard reliability and soft reliability. Furthermore, the lack of soft reliability in modern products has led to increasing numbers of field cases, where products that in fact *technically function well according to their specifications* are being returned or being sought compensation for.

Meeting the customer’s requirements and expectations poses a number of challenges. Firstly, it is evident that customers are becoming more and more quality demanding, backed by their increasing span of options on the global market. Irrespective of the underlying system complexity, customers seek for easy- and delightful-to-use products that do not induce learning costs. Secondly, whereas precisely defining customer requirements and expectations is crucial while developing a new product; these are rather vague at design time, and also continuously changing. Therefore, it is not trivial, if possible at all, to define a consistent set of requirements that is complete and unambiguous, especially in a highly dynamic context. Thirdly, owing to many competitors in the global arena, there is a strong pressure on time to market, which enforces pacing up of the development activities of products.

DOI: 10.1002/qre

We argue that to address these challenges, and hence improve the soft reliability of products, we need a user-centered approach to product development. There are many users with diverse profiles, cultural backgrounds, and past experiences, and the aim is to design products to suit them in their various social and physical contexts of use. Without properly taking these parameters into account, acceptance and eventual adoption of a product cannot be achieved. To alleviate the problem of vague and uncertain customer requirements, users must be involved early in the development process of a product. Moreover, user feedback at all times (i.e. both before and after sales) should be captured and integrated back in the development process to be able to account for diverse use profiles. Finally, to address the problem of the ever-shortening time to market, automatic data collection and knowledge extraction mechanisms are needed.

In this paper, we illustrated that there is the need for early user involvement due to a natural gap between the perceptions of designers and users. Then, we reported on current limitations in user-feedback collection processes and demonstrated that they can be overcome by utilizing our failure classification model. Finally, we presented our usage observation and analysis approach, which enables insights into actual product usage. *Note that the collection of both subjective* (*e.g. user perceptions and feedback*) *and objective* (*e.g. product usage*) *information are needed to cope with soft failures.* Expertise from different disciplines is needed to obtain a holistic view of the problem at hand.

Soft reliability is important. Hence, solutions need to be devised, which help companies to tackle the increasing numbers of 'no fault found' cases. We currently apply our approach in several industrial case studies. Furthermore, we explore ways to semantically enhance the collected data in order to investigate the opportunities provided by semantics-aware analysis techniques (cf. Reference[26]). Finally, the availability of both subjective and objective data about the very same usage process opens up new and interesting possibilities for combining the knowledge of the various domain experts.

Acknowledgements

This work is being carried out as part of the 'Managing Soft Reliability in Strongly Innovative Product Creation Processes' project (http://softreliability.org/), sponsored by the Dutch Ministry of Economic Affairs, under the IOP-IPCR program.

REFERENCES

1. Brombacher AC, Sander PC, Sonnemans PJM, Rouvroye JL. Managing product reliability in business processes 'under pressure'. *Reliability Engineering & System Safety* 2005; **88**(2):137–146.
2. Geudens WHJM, Sonnemans PJM, Petkova VT, Brombacher AC. Soft reliability, a new class of problems for innovative products: 'How to approach them'. *Fifty-first Annual Reliability and Maintainability Symposium*, Alexandria, VA, U.S.A., 2005.
3. Ouden den E, Lu Y, Sonnemans PJM, Brombacher AC. Quality and reliability problems from a consumer's perspective: An increasing problem overlooked by businesses? *Quality and Reliability Engineering International* 2006; **22**(7): 821–838.
4. Koca A, Lu Y, Brombacher AC, Hartmann JH. Towards establishing foundations for new classes of reliability problems concerning strongly innovative products. *Third IEEE International Conference on Management of Innovation and Technology*. IEEE Press: Singapore, 2006.
5. Koca A, Schouwenaar AJM, Brombacher AC. Field-feedback in innovative product development: A comparison of two industrial approaches. *Fourteenth International Product Development Management Conference*. EIASM: Porto, 2007.
6. Koca A, Schouwenaar AJM, Brombacher AC. Analysis of user-centered failure mechanisms in new product development for quality improvement. *Fourteenth International Annual EurOMA Conference*. EIASM: Ankara, 2007.
7. Koca A, Brombacher AC. Extracting 'broken expectations' from call center records: Why and how. *CHI'08 Extended Abstracts on Human Factors in Computing Systems*. ACM Press: Florence, 2008.
8. Pecht MG, Ramappan V. Are components still the major problem: A review of electronic system and device field failure returns. *IEEE Transactions on Components, Hybrids, and Manufacturing Technology* 1992; **16**(6):1060–1064.

9. Ouden den E. Development of a design analysis model for consumer complaints. *Doctoral Dissertation*, Eindhoven University of Technology, Eindhoven, 2006.
10. Karapanos E, Martens J-B. On the discrepancies between designers' and users' perceptions as antecedents of failures in motivating use. *International Conference on Interfaces and Human Computer Interaction*. IADIS: Lisbon, 2007.
11. Hassenzahl M, Wessler R. Capturing design space from a user perspective: The Repertory Grid technique revisited. *International Journal of Human–Computer Interaction* 2000; **12**(3–4):441–459.
12. Martens J-B. Multidimensional modeling of image quality. *Proceedings of the IEEE Software* 2002; **90**(1):133–153.
13. Avizienis A, Laprie J-C, Randell B, Landwehr C. Basic concepts and taxonomy of dependable and secure, computing. *IEEE Transactions on Dependable and Secure Computing* 2004; **1**(1):11–33.
14. Bouwmeester den K, Bosma E. Phases of use: A means to identify factors that influence product utilization. *CHI'06 Extended Abstracts on Human Factors in Computing Systems*. ACM Press: Montreal, Canada, 2006.
15. BSI London. Reliability of systems, equipment and components. Guide to assessment of reliability of systems containing software. *British Standard 5760-8*: *1998*. BSI London, Editor, London, 1998; 5.
16. Randell B, Koutny M. *Failures*: *Their Definition*, *Modeling and Analysis* (*Lecture Notes in Computer Science*, vol. 4711). Springer: Berlin, 2007; 260.
17. Kano N, Seraku N, Takashi F, Tsuji S. Attractive quality and must-be quality. *The Journal of Japanese Society for Quality Control* 1984; **14**(2):39–48.
18. Luckham DC. *The Power of Events*: *An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, U.S.A., 2001.
19. Funk M, Putten Pvd, Corporaal H. Specification for user modeling with self-observing systems. *First International Conference on Advances in Computer–Human Interaction*, Saint Luce, Martinique, 2008.
20. Günther CW, Aalst WMPvd. A generic import framework for process event logs. *Business Process Management Workshops*, *Workshop on Business Process Intelligence* (*BPI 2006*). Springer: Vienna, Austria, 2006.
21. Aalst WMPvd, Reijers HA, Weijters AJMM, Dongen BFv, Medeiros AKAd, Song M, Verbeek HMW. Business process mining: An industrial application. *Information Systems* 2007; **32**(5):713–732.
22. Rozinat A, Jong ISMd, Günther CW, Aalst WMPvd. Process mining of test processes: A case study. *BETA Working Paper Series*, 2007.
23. Aalst WMPvd, Dongen BFv, Günther CW, Mans RS, Medeiros AKAd, Rozinat A, Rubin V, Song M, Verbeek HMW, Weijters AJMM. ProM 4.0: Comprehensive support for real process analysis. *International Conference on Applications and Theory of Petri Nets and Other Models of Concurrency* (*ICATPN 2007*), Siedcle, Poland, Springer, 2007.
24. Rozinat A, Aalst WMPvd. Conformance checking of processes based on monitoring real behavior. *Information Systems* 2007; **33**(1):64–95.
25. Rauterberg M. AMME: An automatic mental model evaluation to analyze user behavior traced in a finite, discrete state space. *Ergonomics* 1993; **36**(11):1369–1380.
26. Medeiros AKAd, Pedrinaci C, Aalst WMPvd, Domingue J, Song M, Rozinat A, Norton B, Cabral L. An outlook on semantic business process mining and monitoring. *OTM 2007 Workshops*. Springer: Berlin, 2007.

Authors' biographies

A. Koca received her BSc and MSc degrees in Computer Engineering from Bilkent University, Ankara, Turkey. Both her MSc thesis and her research at the Information Technology Research Institute (ITRI), University of Brighton, England, concerned the field of computational linguistics. Currently, her research is on Soft Reliability and has strong connections with the industry, in the context of her PhD studies ongoing in the Business Process Design group at the Faculty of Industrial Design, Eindhoven University of Technology.

M. Funk graduated from Aachen University of Technology in 2006. He is currently a PhD candidate in the Electronic Systems group at the Faculty of Electrical Engineering of Eindhoven University of Technology and visiting researcher at Philips Consumer Lifestyle, Eindhoven. His research interests include modeling of self-observing systems, model-driven engineering, and design methodologies.

E. Karapanos received a BSc in Physics from the University of Patras, Greece, and an MSc in Human-Computer Interaction from University College London, U.K. He is currently a PhD candidate in the User-Centered Engineering Group at the Faculty of Industrial Design of Eindhoven University of Technology. His

DOI: 10.1002/qre

research tries to understand diversity, and specifically that induced by time, in users' subjective judgments, in the field of user experience.

A. Rozinat received her BSc and MSc degrees in Software Engineering from the Hasso Plattner-Institute (HPI), University of Potsdam, Germany. She is currently a PhD candidate in the Information Systems group at the Faculty of Technology Management of the Eindhoven University of Technology. Her research interests include process mining, data mining, business process management, evaluation techniques, simulation, and case studies.

W. M. P. van der Aalst is a full-time professor of Mathematics and Computer Science at the Eindhoven University of Technology (TU/e). Currently he is also an adjunct professor at Queensland University of Technology (QUT) working within the BPM group there. His research interests include workflow management, process mining, Petri nets, business process management, process modeling, and process analysis. Wil van der Aalst has published more than 80 journal papers, 12 books (as author or editor), 200 refereed conference publications, and 25 book chapters. Many of his papers are highly cited and his ideas have influenced researchers, software developers, and standardization committees working on process support.

H. Corporaal has gained an MSc in Theoretical Physics from the University of Groningen, and a PhD in Electrical Engineering, in the area of Computer Architecture, from Delft University of Technology. Corporaal has been teaching at several schools for higher education, has been associate professor at the Delft University of Technology in the field of computer architecture and code generation, had a joint professor appointment at the National University of Singapore, and has been a scientific director of the joined NUS-TUE Design Technology Institute. He has also been department head and chief scientist within the DESICS (Design Technology for Integrated Information and Communication Systems) division at IMEC, Leuven (Belgium). Currently, Corporaal is a Professor in Embedded System Architectures at the Eindhoven University of Technology (TU/e) in the Netherlands. He has co-authored over 200 journal and conference papers in the (multi-)processor architecture and embedded system design area. Furthermore, he invented a new class of VLIW architectures, the Transport Triggered Architectures, which is used in several commercial products. His current research projects are on the predictable design of soft and hard real-time embedded systems.

J. B. O. S. Martens studied at the State University of Gent, Belgium, where he received his Electrical Engineering degree in 1979 and his PhD degree in 1983. In 1984 he joined the Institute for Perception Research at Eindhoven University of Technology, where he performed research on image quality, psychometrics, image processing and coding. He published a book entitled 'Image Technology Design—A Perceptual Approach' (Kluwer Academic Publishers) on these topics in 2003. He joined the department of Industrial Design in 2003, where he is a full-time professor in the field of Visual Interaction. His current research interests are in tangible interaction, augmented reality, interactive visualization, and understanding user experiences.

P. H. A. van der Putten received his Bachelor in Electrical Engineering from the IHBO in Eindhoven. Since 1974 he was a research employee and part-time Electrical Engineering student at the Eindhoven University of Technology (TU/e). He received the Professional Doctorate in Engineering (PDEng) in 1993 from the Stan Ackermans Institute (TU/e). Since 1993 he is an assistant professor at the TU/e (Electronic Systems group). In 1997 he received his PhD from the Eindhoven University of Technology. He teaches various topics for post-MSc programs and his current research interests are object-oriented analysis, requirements analysis, and model-driven design methods for hardware/software systems.

A. J. M. M. Weijters is an associate professor at the Technology Management Department of the Eindhoven University of Technology and member of the BETA research group. His current research focuses on data and process mining (i.e. to extract knowledge from event logs recorded by an information system to analyze the underlying business processes). He is the author of many scientific publications in the mentioned research field. His papers appeared in journals such as *Data Mining and Knowledge Discovering*, *Information Systems*, *IEEE Transactions on Knowledge and Data Engineering*, *Artificial Intelligence in Medicine*, *European Journal of Operational Research*, *Computers in Industry*, *Knowledge-Based Systems*, *AI-Review*.

DOI: 10.1002/qre

A. C. Brombacher obtained a BSc, MSc (cum laude) in Electrical Engineering and a PhD in engineering, science at Twente University of Technology. Aarnout Brombacher has experience in industrial product development projects with special focus on 'user-perceived quality and reliability' and in the development of supporting quality and reliability analysis methods and tools. He has authored and coauthored over 50 papers on these subjects and has written a book with the title 'Reliability by Design'. Aarnout Brombacher is currently a professor in the Faculty of Industrial Design of Eindhoven University of Technology, heading the group Business Process Design. He is also, as senior specialist, connected to Philips CFT Industrial and Innovation Support (IIS).

DOI: 10.1002/qre