CHAPTER 7

Conclusions

7.1 Predictable Quality?

The work on this book began with the ambitious aim of forging links between the external and internal aspects of software quality for interactive systems. To address this aim, the Working Group 2.7(13.4) adopted the strategy of associating quality factors with software phenomena. As long as the associations are valid and significant, quality can be addressed via the development process and the tools and materials that support it, rather than by well-intentioned but precarious efforts of highly skilled individuals who invariably lose contact at some point in its life cycle.

The strategy of associating quality factors with software phenomena can be summarized as follows.

- Quality factors have been expressed as external and internal properties.
- Software phenomena have been addressed as the use of methods, architectures, tools and materials within a structured and well managed development process.
- Properties have been associated with selected software phenomena (architectures, tools and materials) by demonstrating interactions between the two.

In the process, it has been shown that quality can be addressed, and in parts proven or delivered, by the judicious use of software architectures, tools and materials. The goal of this book, which is to describe relationships between the process of software construction and system quality from the users' perspective, has therefore been met. We have been able to achieve this in breadth, with some supporting detailed analyses.

The main contribution of the book is to establish a conceptual framework that supports and guides analysis of interactions between software properties and software phenomena. This framework has been exercised in three important ways: by applying it to the currently critical area of software architecture; by applying it to a wide range of tools and materials, with three in-depth analyses of commercial tools, and some broader site reports; by applying it to a demanding application area that is safety-critical, involves multiple cooperating users and has real time requirements. Taken together, this is a comprehensive initial validation of the conceptual framework. The framework has been shown to be applicable to key aspects of

C. Gram et al. (eds.), *Design Principles for Interactive Software*

software development and demanding facets of application design. Working Group 2.7 believes that more extensive in-depth analysis can be performed within the framework developed above.

The framework has strong predictive potential for analysis of systems, architectures, tools and components. This potential was demonstrated by the consistently broad range of insights that were yielded by the analysis of existing architectures and tools. The next key step is to go beyond potential to proven effectiveness. To do this, it is necessary to complete the scientific process as follows.

- All identified interactions between software properties and phenomena must be validated by showing that the interaction holds in a representative range of practical scenarios.
- The significance of all identified interactions must be assessed, i.e., what should the real rewards (or costs) of each interaction be?
- All proposed rewards (or costs) must be validated by showing that they arise in a range of practical scenarios.

Working Group 2.7's work has reached the point where it is possible to address the more practical question of significance. Granted that these interactions exist, then what are the implications? Are they imperceptible in practice, or a mild nuisance, a moderate concern, a major impedance or facilitator – or an absolute make or break? They are all of these, some of the time.

Once the significance of each interaction has been understood, it is possible to set out to test the validity of the proposed implication: does the interaction really have the identified implication? There will be some very difficult experiments to design once credible hypotheses can be made about the significance of interactions.

When the scientific process yields results, these must be finally exploited by an engineering step:

- Effective methods and tools must be developed that deliver the rewards of positive interactions and avoid the cost of negative interactions.

This is the hardest step of all. Even when there is scientific validation of the practical significance of interactions between properties and software phenomena such development is far from easy. It expresses in fact the long term goal of the work of Working Group 2.7, that is, to let quality be addressed via the development process and the tools and materials that support it, instead of via well-intentioned but precarious efforts of highly skilled individuals. The process and practice of developing interactive systems needs to change so that high quality systems could reliably be developed from the perspectives of both endusers and developers.

Figure 7.1 summarizes the process of analysis, validation and exploitation. There are six steps in this process (where the first step comprises two parallel substeps).

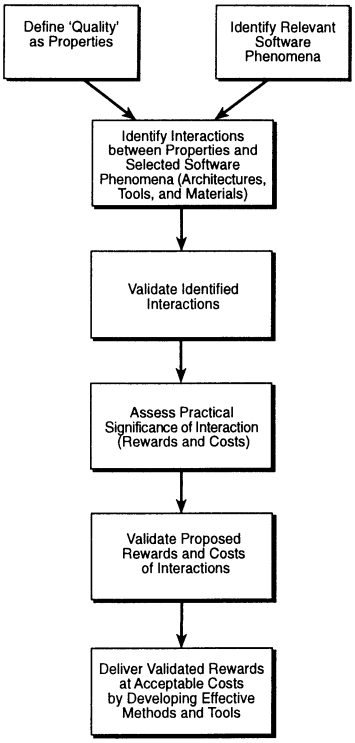

Figure 7.1 *Strategy for Effective Development of High Quality Interactive Systems.*

7.2 Contributions

We feel that we have made good progress on the first steps towards our vision of predictably effective software development for interactive systems. This claim may be justified on the basis of a thought experiment that considers how well the group has done on each of the (sub)steps from identification of properties/phenomena to the near-inevitable attainment of quality.

The (sub)steps on the route were shown in Figure 7.1. In summary, they are:

- identification of external and internal properties (substep);
- identification of relevant software phenomena (substep);
- identification of interactions between properties and phenomena;
- validation of interactions between properties and phenomena;
- determination of practical significance of interactions between properties and phenomena;

- validation of the significance of interactions between properties and phenomena;
- effective exploitation of significant interactions between properties and phenomena.

For each step, we could ask how well we have done in the work reported above. We could then ask what the probable attainment for the step would be if the best work in the world could be synthesized and condensed into a single (though very lengthy!) report. Lastly, we could ask what our likely acceleration for the step would be if Working Group 2.7 spent another year on this report (with assistance from any expert we cared to name). We have done this for each (sub)step, but prefer to leave readers to form their own judgements. However, we will conclude by summarizing the more significant answers to some questions for each (sub)step.

7.2.1 Identification of external and internal properties

The work described in this step is an answer to the question: 'Which properties encapsulate desirable properties of an interactive system's external behavior or internal structure?' Chapters 2 and 3 presented the answer.

Our set of properties is the first to span a range of both external and internal aspects of quality. In the process, we may have gone slightly beyond the best synthesis that would otherwise be available, see, e.g., Nielsen (1994) and Gram (1995). Still, we have omitted several key external and internal properties, and given this knowledge, we could address these deficiencies (e.g., we have omitted 'learnability' and 'guessability' (Jordan *et al.*, 1991) as external properties).

7.2.2 Identification of relevant software phenomena

The work done in this area is an answer to the question: 'What are the key methods, techniques, tools, structures and components for software development?' Chapter 3 started to give the answer, which was then extended in Chapters 4 and 5.

The discussions in these chapters cover several aspects of methods, techniques and tools, but there are many software phenomena that are not covered well (e.g., development methodology and process), and thus more could have done here. However, the field of software engineering is well documented, and thus with more work we could catch up with the current state-of-the-art.

7.2.3 Identification of interactions between properties and phenomena

This gives an answer to the question: 'What are the positive, neutral, and negative interactions between software quality and software phenomena,

and what must developers do to exploit positive interactions?' Chapters 4 and 5 presented answers for selected software phenomena (architectural models, tools and materials for specification and construction), but represent only a first attempt to explore and describe systematically interactions between quality and software phenomena. We feel that it is a worthwhile although limited contribution for this step. Extending the analysis of interactions seems to be straightforward, so that the coverage of interactions in Chapters 3 to 6 should be readily extendible.

7.2.4 Validation of interactions

This area of work attempts to answer the question: 'Do the identified positive, neutral and negative interactions between software quality and software phenomena arise, as predicted, for real tools, at real development sites, and for real application developments?'

Chapters 4 to 6 present our partial answer. Some selected interactions were discussed on a few focused examples. But much more supporting evidence could be gathered from the wide-ranging experience of other software developers.

7.2.5 Significance of interactions

A thorough analysis of the significance of the interactions would answer the question: 'What is the likely effect, in terms of development costs and benefits, as well as gains for users, of the identified positive interactions between software quality and software phenomena?'

We have not addressed this question systematically, although there are comments relevant to this question at several points in Chapters 4 to 6. But even the very piecemeal analysis here constitutes an improvement on the current state-of-the-art, because very little work of this nature has been reported so far.

7.2.6 Validation of significance of interactions between properties and phenomena

To validate the significance of interactions means to answer the question: 'What is the real effect, for real application developments, in terms of actual development costs and benefits, as well as measured gains for users, of the identified positive interactions between software quality and software phenomena?'

No such validation has been undertaken, because it requires substantial work and experimentation. But it is hoped that the framework presented in the book may be followed up by research groups that use it as a basis for carrying out individual experiments and studies.

7.2.7 Effective exploitation of significant interactions between properties and phenomena

Work here would answer the question: 'Where are the methodologies and supporting tools that will deliver systems with a required property profile?' To the best of our knowledge, such methods and tools do not exist and will not exist for the foreseeable future.

The lack of a good science base for designing CASE tools means that while there are many tools which are effective for many aspects of many development tasks, and which support some external properties for users, progress on this step is piecemeal and unpredictable. As a result, tools and methods can often be developed that improve some aspects of internal and external quality, but make others worse. With a better understanding of key properties and their interactions with architectures, tools and materials, such undesirable setbacks should not arise.

7.3 Epilog

The work reported here has largely been one of conceptual ploughing. The field formed by the intersection of Human–Computer Interaction and Software Engineering has turned out to be very large, and Working Group 2.7 has not been able to plough all of it as evenly or as deeply as we had originally hoped. However, a wide range of individual experiences and those reported by colleagues in their publications have combined to reinforce our view that we have been ploughing in the right direction. Our hope is that the field has been ploughed enough for many seeds to take root and flourish.