Empirical Evaluation of Techniques and Methods Used for Achieving and Assessing Software High Dependability

Ioana Rus
Fraunhofer Center for Empirical Software Engineering Maryland
irus@fc-md.umd.edu

Victor Basili
Marvin Zelkowitz
University of Maryland and Fraunhofer Center for Empirical Software Engineering Maryland
Basili,mvz@cs.umd.edu

Barry Boehm
University of Southern California boehm@usc.edu

For achieving high dependability of software intensive systems, not only product dependability benchmarking is needed but also benchmarking of technologies and processes for achieving and assessing software dependability. Dependability engineering, and more specifically technology management and assessment of effectiveness and efficiency of different technology interventions, is the objective of the work we introduce here. This work is performed as part of the *High Dependability Computing Project* (HDCP)[1] that is an incremental, five-year, cooperative agreement, part of a broad strategy for dependable computing, that links NASA, corporate partners and universities and research centers such as Carnegie Mellon, University of Maryland, Fraunhofer Center Maryland, University of Southern California, Massachusetts Institute of Technology, University of Washington and University of Wisconsin. For now the focus is on NASA projects, but the results will be captured and organized in an experience base, so that they could be disseminated and applied to other organizations. For example, the first step would be to extend the results to organizations that are members of the *High Dependability Computing Consortium* (HCC)[2] and the *Sustainable Computing Consortium* (SCC)[3].

As part of our activities we are looking at a series of steps to evaluate such interventions. Developing high dependability software requires specifying the dependability requirements, using development techniques and methods (that we will call "technologies") to build-in high-dependability as the product is developed, and also technologies to verify that the required dependability has been achieved. Our research focuses on evaluating the effectiveness of these technologies with respect to achieving and assessing the desired dependability, and also the cost of using these techniques. For this purpose we are employing diverse empirical evaluation methods such as case studies, pilot projects, project monitoring, assertion, field study, literature search, lessons learned, static analysis, replicated experiment, synthetic experiment, dynamic analysis, product and process simulation.

The technologies might be evaluated with respect to dependability if applied in isolation, as well as if they are combined in various ways (since different technologies are used in different development phases and also different technologies might address different attributes of dependability). Technology comparison might also be required, therefore the need for a common set of measures that can be applied to the results of all technologies (or at least to the ones comparable to each another, i.e., addressing the same attribute). Some technologies might work for specific contexts (e.g. application domain, type of system - concurrent processes, distributed systems, real-time systems, db transactions, operational environment) but not for all situations, so these circumstances must also be studied and identified.

In order to perform technologies evaluations we need to determine the variables that we will observe, measure, and analyze. Therefore we need to have a model of dependability (sub-attributes and measures), for the delivered software. In addition we also need indicators that can be measured during development and help predicting the dependability of the operational system.

Given that dependability is a behavioral property of a system, depending on the environment and the way the system is operated, we see the following questions to be addressed for determining useful measures for our technology evaluation:

- What are the measures for the dependability of a system and how does that translate to software?

[1] http://amesnews.arc.nasa.gov/releases/2002/02_03AR.html
[2] http://www.hdcc.cs.cmu.edu/
[3] http://www.sustainablecomputing.org/

- What does *high dependability* mean (if dependability is a combination of other attributes such as reliability, security, availability, robustness, then what are the values of these attributes and how are they combined to result in high dependability)?
- What are the indicators in intermediate phases of development that allow prediction of the dependability of the deployed system?

If we consider the perspective of a maturing dependability technology we can view each high-dependability technology as passing through a series of evaluation milestones, each stressing the technology and demonstrating its context of effectiveness. Technology researchers will specify the goals for their technologies relative both to *needs*—as specified by users or identified by empirical investigation—and to the *models* for high-dependability. These goals will be established as criteria for studying the technology and identifying the characteristics of the milestone in which the technology is applied. In the assessment process, we identified four steps and corresponding milestones described below. Having a well-defined model and measures of dependability is an indispensable requirement for each of the four test-bed levels mentioned here.

Milestone 1. Internal set: Typically, the technology researcher (creator) has applied the technology to some internally developed set of examples. This set will act as a first milestone for that technology. The technology will be applied to that set of examples defining the milestone by an independent source to make sure the documentation and robustness is sufficient to allow for independent application of the new technology. Thus, before moving the initial examples to the basic common milestone, the technology must have been applied on a technologist-developed test set and that test set should be characterized and used to generate a technology specification and set of criteria for dependability specific to that technology. That initial test set of examples should be contributed to the basic common set, which will be stored in the experience base.

Milestone 2. Basic common set: We can build a basic set of common examples that we can use for applying each of the technologies. The goal is to create a larger universe of problems on which to stress and analyze technologies, both individually and in groups. As stated above, one source of such examples is the internal test sets of the individual developers. However, based on the models, the analysis of the individual test sets, and the analysis of industry problem areas, new examples can be added to this set. This set will allow various technologies to be compared and their strengths and limits assessed empirically. And, of course, it provides a larger domain of potential application for each of the technologies by enlarging the universe of examples. Experiments will be defined for this milestone based on the technology to be tested and the goals established for that technology relative to the milestone.

From industry's point of view, this level offers some insight into what combinations of technologies might be most effective under what conditions and for which problems. From the technology researcher's point of view it provides feedback on how a technique might be expanded and evolved. For the empirical researcher, this milestone will provide new insights into models and goals.

Milestone 3. HDCP domain-specific off-line set: This milestone consists of a domain-specific set of examples, from areas of greatest high-dependability. Ideally, examples in this set will have failure data from real experience associated with them. A committee consisting of NASA personnel as well as HDCP decision makers and technology researchers, again supported by empiricists, will make the choices. This milestone will provide better models of dependability more directly pointed at NASA and HDCC requirements. We will define a different class of experiments for this milestone, involving application domain experts.

Once again insights will be gained on how the various technologies can be integrated and under which circumstances each should be applied, based on decisions such as understanding of the anticipated failures for the problem, the expertise of the appliers of the technology, the effectiveness of the technology for certain classes of faults, and the cost of applying the technique. Success at this milestone should imply that the technology deserves more careful packaging for wider application—high-quality documentation, training materials, tool support, and the like.

Milestone 4. Live examples: This milestone definition is specific to part or all of a system currently under development. Although the techniques have passed through each of the prior milestones, there is clearly a need for risk mitigation. Continual observation by the empiricists is needed and alternate actions are predefined to make change possible when necessary. Experiments may consist of the technology being applied on only part of the system, so a comparison can be drawn with other parts, or it might be a case study of the entire project.

Based on the results of these studies, the technology can be fully packaged for use and placed on the NASA technology shelf as a transferred technology, or it may require a second or third live example for further study of its effectiveness. Ideally, examples in this set will have failure data sets from real experience associated with them.