NASA/CR-97- 207178

FINAL
IN-61 R
701T
065187

DEPARTMENT OF COMPUTER SCIENCE
COLLEGE OF ENGINEERING & TECHNOLOGY
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA 2359-0247

APPLICATION OF DOMAIN KNOWLEDGE TO SOFTWARE QUALITY ASSURANCE

By

Dr. J. Christian Wild, Principal Investigator

Final Report
For the period ending September 30, 1997

Prepared for
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia, 26665

Under
Research Grant NAG-1-439
Dave E. Eckhardt, Technical Monitor
ODURF #146219

February 1997

Final Report for grant NAG 1-43Ø9

Chris Wild Steven Zeil
Department of Computer Science
Old Dominion University
Norfolk, VA 22529

February 12, 1998

In the past year, several of the efforts which were undertaken as part of the research under this grant activity have reached significant milestones. Of the three PhD students supported under this grant, Daniella Rosca completed her dissertation and graduated in December 1997, Cathy Roberts is in the process of writing her dissertation and Brian Mitchell is completing his experiments and should start his dissertation soon. Their work represents the major focus of the research under this grant over the past year.

Supporting the Decision Making Process Across the Project Life Cycle

This work focused on capturing, using, and evolving a qualitative decision support structure across the life cycle of a project. The particular application of this study was towards business process reengineering and the representation of the business process in a set of Business Rules (BR). In this work, we defined a decision model which captured the qualitative decision deliberation process. It represented arguments both for and against proposed alternatives to a problem. It was felt that the subjective nature of many critical business policy decisions required a qualitative modeling approach similar to that of Lee [2] and Mylopoulos [6]. While previous work was limited almost exclusively to the decision capture phase, which occurs early in the project life cycle, we investigated the use of such a model during the later stages as well.

One of our significant developments was the use of the decision model during the operational phase of a project. By operational phase, we mean the pahse in which the system or set of policies which were earlier decided are deployed and put into practice. By making the decision model available to operational decision makers, they would have access to the arguments pro and con for a variety of actions and can thus make a more informed decision which balances the often conflicting criteria by which the value of action is measured. We also developed the concept of a "monitored decision" in which metrics of performance were identified during the decision making process and used to evaluate the quality of that decision. It is important to monitor those decision which seem at highest risk of not meeting their stated objectives. Operational decisions are

DEPARTMENT OF COMPUTER SCIENCE
COLLEGE OF ENGINEERING & TECHNOLOGY
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA 2359-0247

APPLICATION OF DOMAIN KNOWLEDGE TO SOFTWARE QUALITY ASSURANCE

By

Dr. J. Christian Wild, Principal Investigator

Final Report
For the period ending September 30, 1997

Prepared for
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia, 26665

Under
Research Grant NAG-1-439
Dave E. Eckhardt, Technical Monitor
ODURF #146219

Submitted by the
Old Dominion University Research Foundation
P.O. Box 6396
Norfolk, VA 23508-0369

February 1997

also potentially high risk decisions.

Finally, we investigated the use of performance metrics for monitored decisions and audit logs of operational decisions in order to feed an evolutionary phase of the the life cycle. During evolution, decisions are revisisted, assumptions verified or refuted, and possible reassessments resulting in new policy are made. In this regard we implemented a machine learning algorithm which automatically defined business rules based on expert assessment of the quality of operational decisions as recorded during deployment.

Project Management in Uncertain Environments

The technological advances of recent years allow unprecedented engineering, software, business, and manufacturing projects to be brought to fruition. These highly complex projects are often accompanied by highly complex problems in managing the projects to their successful completions. This complexity is further increased if the members of the project team are geographically disbursed. Projects cannot follow a pre-determined path; instead, they evolve as project decisions are made in response to scheduled and unscheduled events. Our work focused on two main aspects of managing complex engineering projects in the face of uncertainity: competitive contingency plans and support for episodic project management.

Competing contingency plans are enacted when the uncertainty about the success of two or more proposed courses of action precludes a clear decision but time considerations dictate that some action be taken in any case. These plans compete for resources, hence the name competitive contingency plans. Since one or more of the enacted plans maybe abandoned at some later stage, managing competitive contingency plans is a high risk undertaking. There is little support for this activity in present theory and practice of project management.

Support for the Episodic nature of projects. Projects proceed as a series of inter-related episodes. A project is not one isolated step-wise linear process. Events happen which require the manager to provide a set of responses involving decisions, tasks, schedule changes, resource reassignment, or re-evaluation of project goals. The triggering event, and the manager's responses to that event, play out as an episode which impacts (and is impacted by) other episodes which occur during the course of the project. For an episode to result in a satisfactory action, the decision makeer must be able to extract relevant information from the corporate memory, assess alternate course of action and set into motion a plan which will form a proper response.

To deal with these issues, we have built a decision based arcitecture based on the following:

1. An information structure which represents competitive contingency plans.

2. Decision structures which capture and assist the episodic nature of project management.

3. A micro-organizational view of a project based on groups of people interested in the outcome of particular decisions.

4. A process model for navigating through a episode and communicating within the micro-organization.

5. A communications model based on episodic decision support.

6. In addition, the architecture integrates traditional project management support theory and COTS project management tools into decision-based episodic project management.

Estimating Reliability After Non-representative Testing

There is a substantial body of literature devoted to *directed* testing methods, which manipulate the choice of test inputs so as to increase the probability and/or rate of fault detection. These include most well-known testing methods, including functional and structural testing, data flow coverage, mutation analysis, and domain testing[1, 7]. Historically, a difficulty affecting the deployment of these methods has been the lack of any quantified measure of test effectiveness with external referents (i.e., that is not based upon properties defined by the criterion itself).

In contrast, a variety of reliability growth models provide quantified measures of test effectiveness in terms that are directly relevant to project management [3, 4, 5], but at the cost of restricting testing to *representative* selection, in which test data is chosen to reflect the operational distribution of the program's inputs. During testing, data is collected on the observed times between program failures (or, similarly, numbers of failures within a time interval). These observations are fitted to one of various models, which can then be used to estimate the current reliability of the program.

This project was devoted to finding a common ground between the areas of directed testing and reliability modeling. Specifically, we proposed a new Order Statistic model of reliability growth. This model can employ an arbitrary mixture of *program* failure rate data, as in conventional reliability growth models with *fault* failure rate information obtained via post-mortem analysis of the debugged faults. The primary advantages of this model are:

- Test planners regain the flexibility to employ their best testing practices, whether those involve directed testing, representative testing, or a mixture of the two. The choice of testing method is no longer solely determined by the desire to obtain numerical predictions of reliability.

- More robust experimental designs can be formulated by taking advantage of a wider variety of options for data collection.

0.1 Publications List

B. Mitchell and S. J. Zeil. A reliability model combining representative and directed testing. In *Proceedings of the 18th International Conference on Software Engineering*, pages 506–514, Los Alamitos, CA, Mar. 1996. IEEE Computer Society Press.

B. Mitchell and S. J. Zeil. Modeling the reliability growth of non-representative testing. *Annals of Software Engineering*, 4:11–29, 1997.

B. Mitchell and S. J. Zeil. An experiment in estimating reliability growth under both representative and directed testing. to appear in *1998 International Symposium on Software Testing and Analysis (ISSTA)*, Mar. 1998.

Brian Mitchell and Steven J. Zeil, *An Experiment in Estimating Reliability Growth Under Both Representative and Directed Testing* (expanded version of ISSTA paper above), Old Dominion University technical report TR–97–31, July, 1997

S. J. Zeil and B. Mitchell. Estimating reliability during non-representative testing. In 14^{th} *International Conference on Testing Computer Software*, Washington, D.C., June 1997.

Daniela Rosca, Sol Greenspan, Mark Feblowitz, and Chris Wild. A Decision Making Methodology in Support of the Business Rules Lifecyle. In *ISRE'97 Conference*, Annapolis, MD, June 1997.

C. Roberts, C. Wild, K. Maly, "Processes for Managing Project Uncertainty with Contingency Planning," IFIP WG8.3 London Conference, July, 1996 pp. 352-365.

References

[1] B. Beizer. *Software Testing Techniques.* Van Nostrand Reinhold, second edition, 1990.

[2] J. Lee and K.-Y. Lai. What's in design rationale? *Human-Computer Interaction*, 6:251–280, 1991.

[3] B. Littlewood. Theories of software reliability: How good are they and how can they be improved? *IEEE Transactions on Software Engineering*, SE–6(5):489–500, Sept. 1980.

[4] M. R. Lyu, editor. *Handbook of Software Reliability Engineering.* IEEE Computer Society Press, 1996.

[5] J. D. Musa, A. Iannino, and K. Okumoto. *Software Reliability: Measurement, Prediction, Application.* McGraw-Hill, New York, NY, 1987.

[6] J. Mylopoulos, L. Chung, and B. Nixon. Representing and using nonfunctional requirements: A process-oriented approach. *IEEE Trans. on Soft. Eng.*, 18(6):483–497, June 1992.

[7] L. J. White. Software testing and verification. In M. Yovits, editor, *Advances In Computers*, volume 26, pages 335–391. Academic Press, Inc., London, UK, 1987.