Reducing Autonomy Risks through Rational Selection of Verification and Validation Strategies

Julian Richardson,[1] RIACS/USRA, NASA Ames Research Center
Dan Port, Rick Kazman, University of Hawaii
Ray Madachy, LiGuo Huang, Barry Boehm, University of Southern California

Focus issue: Mission assurance, test and validation.

December 3, 2005

[1]MS 269-2, NASA Ames Research Center, Moffett Field, CA 94035-1000. Email: julianr@riacs.edu

Reducing Autonomy Risks through Rational Selection of Verification and Validation Strategies

Julian Richardson, RIACS, NASA Ames Res. Center
Dan Port, Rick Kazman, University of Hawaii
Ray Madachy, LiGuo Huang, Barry Boehm,
University of Southern California

1. Introduction

Software development projects differ in many ways: domain, personnel, programming language and features used, complexity, use of tools etc. These differences affect the number, nature and distribution of defects in the software produced. We believe that significant improvements in verification and validation efficiency can be achieved by tailoring the choice and ordering of verification and validation tools to suit the characteristics of the software project, compared to using a fixed verification and validation strategy which does not take into account these characteristics. For example, applying a software model checking tool which specializes in detecting currency errors will not be effective when the software being validated contains little or no concurrency.

Recently, we have constructed a prototype model of *risk* for autonomy software. We have focused on this domain because autonomy enables unique capabilities for future space exploration such as automated rendezvous and docking and integrated system health management (ISHM), but uncertain risks in the development and deployment of autonomy systems create a barrier to its use [4]. Our broad goal is to remove this barrier by developing a model which can quantify autonomy risk. Most of our model is applicable to other software domains.

The model computes a measure of the risk in an autonomy software system, based on attributes of the software development and on a specification of the risks present in the system. It then computes a measure of the expected utility of different verification and validation tools if applied to the system, and presents the model user with a recommendation for the order in which verification and validation (V&V) tools should be applied in order to reduce the system risk as quickly as possible.

We have used a prototype implementation of the model to compare the effectiveness of V&V strategies recommended by the tool with fixed V&V strategies. Depending on the risks present in the autonomy software system under study, the recommended V&V strategy may perform significantly better than the fixed strategy.

This research was conducted for RIACS (an institute of the Universities Space Research Association (USRA)) under a grant from NASA's Exploration Systems Mission Directorate.

2. Autonomy System Risk

We have built a prototype model, AUTONOMO, which quantifies risk using the concept of *risk exposure* [2]. The risk exposure in a system is defined to be the sum, over all sources of loss, of the probability that the loss occurs multiplied by the magnitude of the loss.

Our initial implementation of AUTONOMO computes the risk exposure for an autonomy software system for each risk taken from a taxonomy of autonomy software risks, a fragment of which is shown in figure 2. Each node is labeled by a risk category. Children represent subcategories of their parent. A node's children do not necessarily form an exhaustive or pairwise disjoint set.

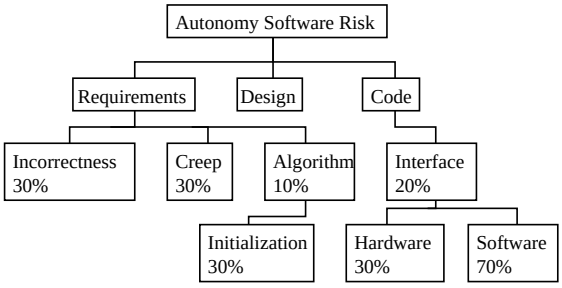

Figure 2. Fragment of risk taxonomy.

In order to apply the model to an autonomy software system, the model user must first provide values of the inputs for the *COQUALMO* model [1] which summarize aspects of the software development process including number of lines of code developed, skill levels of programmers and analysts, real time constraints, and aspects of the software verification and validation process, for example level of use of automated testing. From these attributes, COQUALMO estimates the number of defects which will remain in the software product after development and divides them into requirements, design and defect errors (3 of the top 4 nodes of our risk taxonomy — the 4^{th} is "operational").

Next, the user specifies which risks are applicable to the system in question. Risks are selected from the taxonomy of autonomy risks, which is part of the model. As indicated in figure 2, each node in the taxonomy is assigned a relative frequency. The risk exposure for the risk labeling some node is the risk exposure of its

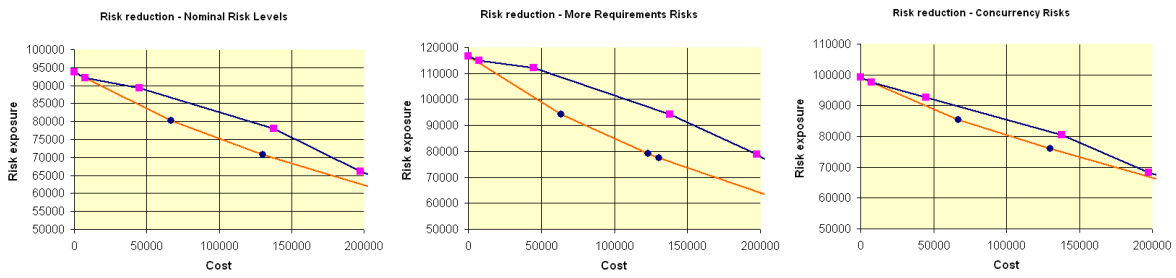

Figure 1. Effectiveness of recommended V&V strategies versus a fixed strategy for 3 risk scenarios.

parent multiplied by this frequency. The risk exposures for the top level categories are computed as some fixed multiple of the number of defects for that category estimated by COQUALMO. The taxonomy and the relative frequencies have so far been constructed on the basis of the best judgment of the authors. We plan on using defect categories and data from orthogonal defect classification [3] carried out in the aerospace and other domains to give a more scientific basis for these.

3. Rational Verification and Validation

We are less interested in the total risk exposure for the system than we are in how best to reduce that risk. The model therefore contains a taxonomy of V&V tools which can be applied to reduce risk exposure. For each ⟨tool, risk⟩ pair, the model contains a measure from 0 to 1 inclusive which specifies the effectiveness of the tool, i.e. what proportion of the risk exposure due to that risk is removed by application of the tool. Some of these measures have been derived from experiments carried out by RIACS staff at NASA Ames.

Finally, for each verification tool in the risk mitigation taxonomy, the model estimates the cost of applying that verification tool to the software system. For most verification tools, this cost is estimated to be a percentage of the total software development cost.

Given a specification of the development attributes and specific risks for an autonomy software system, the model computes the risk exposure for each risk category, and for each verification tool, the reduction in risk exposure achieved by applying that tool to the software system. The model recommends an ordering of V&V tools with those which provide the highest risk reduction per unit cost first.

4. Results

We applied our model to a prototype autonomous flight subsystem. The COQUALMO inputs and selection of applicable risks are based on discussions with developers of that system. We compared the effectiveness of AUTONOMOs recommended verification and validation strategy with a fixed (but good) strategy in various scenarios: 1. the prototype flight subsystem, 2. a version of that subsystem in which we artificially elevated the levels of software requirements errors, 3. a version of that subsystem with artificially elevated levels of concurrency errors. Figure 1 shows the risk reduction profile achieved by the recommended versus fixed V&V strategies. Each curve shows how risk exposure reduces as tools are applied in sequence, the upper curve representing the fixed V&V strategy, the lower curve the strategy recommended by AUTONOMO. The results demonstrate that for some types of software, the strategy recommended by the model may perform significantly better than a fixed strategy. For example, for nominal risk levels,the recommended strategy reduces risk exposure to 80000 with roughly half the cost of the fixed strategy.

The method of calculating risk exposure and risk reduction outlined in this document requires us to produce reasonably accurate estimates of the relative frequency of different risk types. We are also working on a complementary approach, which estimates risk exposure without separately estimating probabilities and loss values.

5. Bibliography

[1] S. Chulani, B. Boehm, and B. Steece. *Bayesian analysis of empirical software engineering cost models.* IEEE Transactions on Software Eng, 25(4), 1999.

[2] N. Levenson, *Safeware.* Addison Wesley, 1995.

[3] R. R. Lutz, I. C. Mikulski,. *Empirical Analysis of Safety-Critical Anomalies During Operations.* IEEE Transactions on Software Eng, 30(3), March 2004.

[4] T. Menzies, J. D. C. Richardson. *XOMO: Understanding Development Options for Autonomy* 2005 COCOMO Forum, University Southern California, 2005.