

Ten Signs of a Good Reuse Management Plan

Edwin Morris, Wm B. Anderson, Mary Catherine Ward,
and Dennis Smith

SEI/CMU
Pittsburgh, PA
www.sei.cmu.edu

Abstract. A Reuse Management Plan defines the strategy for selecting, approving and upgrading common reusable software components. The SEI, in conjunction with the U.S. Army, the Boeing Company, and the Fraunhofer USA Center for Experimental Software Engineering, is developing a Reuse Management Plan for a large Army program. Ten critical features of quality Reuse Management Plans have been identified and are presented..

1 Introduction

Commercial, military, and other government organizations continue to increase their reliance on reused software to provide major capabilities in new systems. This reused software goes by many different labels, including: commercial off the shelf (COTS), government off the shelf (GOTS), shareware, freeware, open source, and non-developmental items. While the sources of these types of software vary, they have two key characteristics in common from the perspective of an organization attempting to use them: imprecise knowledge of the internals (e.g., architecture, design, assumptions, and dependencies) and limited control over the evolution of the component.

Too frequently, organizations are disappointed in their experience using such reused software components. This is particularly the case when components are being used for large scale “systems of systems” where components can have unforeseen affect on other parts of the system. Often, the problems experienced can be directly traced to imprecise knowledge and limited control that result in faulty selection processes, conflicts between components, inappropriate integration strategies and inability to sustain the component across the system life cycle.

Acknowledgements. We wish to thank Sue Hermanson and Sam Montgomery of Boeing Company and Michele Shaw of the Fraunhofer USA Center for Experimental Software Engineering, University of Maryland for their collaboration in developing the Reuse Management Plan from which this paper was abstracted. We believe this collaboration of industry and academic experts has led to a good, and still improving, approach for managing the incorporation and sustainment of reused components.

2 The Ten Critical Features

Careful planning for appropriate processes, techniques and artifacts can help organizations avoid or overcome common problems associated with reuse. Boeing Company, supported by the Fraunhofer USA Center for Experimental Software Engineering and the Software Engineering Institute, is defining such a plan to manage commercial and other reused software on a large government program. While the details of the plan are beyond the scope of this paper, ten critical characteristics of the plan are presented here. We believe that the following characteristics are fundamental to plans addressing long term use of complex “reusable” software:

- A product line strategy
- An iterative process
- A component manager
- Risk-based management of components
- Full lifecycle coverage
- Aggressive evaluation and selection of components
- Careful configuration and change management
- A complete historical record
- A component health checkup
- Metrics that lead to improvement

These essential characteristics are discussed in sections 2.1-2.10.

2.1 A Product Line Strategy

In standard reuse approaches, a reusable asset is discovered, modified as appropriate¹, and installed in the product. While this saves some development time, this “clone-and-own” method makes each system unique, with maintenance and evolution no longer shared with other members of the family. Where several similar systems will be developed within a market segment or performing a specific mission, organizations should strive to create a software product line.

A software product line is a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way. Reusable core assets include software components, but also the associated documentation, architecture, software design, Application Programming Interface (API), user interface design, test plans, test cases, schedules and budgets, development processes, and more. Building a set of software systems as a software product line has been shown [1] to dramatically shorten development time, increase productivity, increase quality, and reduce cost, as compared to developing the systems one at a time in isolation from each other. In effect, where several similar systems will ultimately be developed, a product line strategy presents an opportunity to maximize reuse.

¹ The phrase “as appropriate” is critical. We do not advocate the modification of some sorts of reusable components, conspicuously including COTS.