

Integrating Configuration Management Processes and Tools into Legacy Systems

Tonja Cooper¹; James Lin²; Mark Chatillon³; Robert W Sible, Jr.¹

¹Jet Propulsion Laboratory (JPL)/ California Institute of Technology (Caltech), Pasadena, California, USA;

²Chase Computing International, Corp., Alhambra, California USA;

³Raytheon Australia, Canberra Deep Space Communication Complex

Implementing updated Configuration Management processes and tools into a project that goes back more than 30 years with a legacy of paper forms is a formidable challenge. The manual processes and the old databases made it extremely difficult to produce metrics or extrapolate data within a reasonable time that provided accurate and useful information on product performance, product changes, process performance, or configuration management accounting. The Quality Engineering & Implementation Office (QE&I) of the Deep Space Mission Systems (DSMS) of the Jet Propulsion Laboratory, (JPL) has incrementally developed and integrated a set of tools based on a common architecture to automate and manage the processes for tracking problems, data outages, equipment status, change requests, and code changes for the DSMS Products. As we continue towards completing the automation and integration of the processes and tools that manage different aspects of configuration control, the rewards are already being realized by management in available on-line information and metrics, by the users in the convenience of documenting changes and reporting progress against plans, by developers in the ease of gathering data for delivery reviews and release description documents, and by the CM Team in producing metrics and reports and by reducing maintenance time by utilizing common COTS, scripts, and scheduled routines across the tools.

Each tool was incrementally introduced, training sessions were conducted and the data from other databases was converted into the new system. This incremental introduction reduced the culture shock and resistance to change and allowed us to budget the development cost incrementally. By building the tools on a common architecture, utilizing the same COTS, and sharing routines, the user interface of each was similar and provided easier maintenance and training by the development staff.

Each tool has a workflow, based on phases and actions, that manages the process defined for the product. The processes for the different aspects of change control (problem reporting, discrepancy reporting, change requests, code control, document changes) interact at one or more points within the lifecycle. This "process orchestration" is the key that relates the business rules across the different processes and drives the interfaces that create the cohesive integration of the different tools.

The ability to document changes at a Programmatic level and have those drive change requests to our service commitment requirements which drive changes to the DSMS products help us to more accurately understand the scope and cost of high level changes and to know when the last piece has been implemented. Providing the ability to relate problems in operational service to a flaw in hardware or software helps us improve those products more efficiently. Reporting changes against a predefined set of products organized in a hierarchy allows reporting of metrics at assembly, subsystem, and system levels. Integrating the code control and the change control processes provides automation of CM status accounting by product version. Better products and processes are the benefits of this integration.