# Reusable Architecture

## *Based on the US Army Bradley M3A3 BFIST Project*

**Abdul Siddiqui Software Engineering Manager**
**US Army TACOM, Warren, MI 48397-5000**

SiddiquA@tacom.army.mil

**ABSTRACT**
Software reuse is favorable for government/military projects due to the extensive testing and supportability process involved with the software systems that are developed for their use. One of the key efforts to meet our goal of software reuse was to manage and trace the requirements of the currently existing software architecture and the new requirements developed for product line program. In order to initiate our software reuse efforts we conducted reviews of the requirement specifications of both existing software architecture and the new system components to be developed. For the new software components requirements were developed and the process involved the establishing of the services that the customer demands from the system and the constraints under which it operates and is developed. Based on the requirement similarities and matching, software components can be identified for reuse. This effort of requirement management and analysis also gave a clear understanding of the external interface the software components have and the message/data traffic between the components in the system. In this paper we will be presenting the programmatic challenges involved, efforts of our Program Managers Office to minimize the project risks regarding the requirements management for software reuse and the lessons learned form our efforts to enhance the system architecture in product line context.

**RECONSTRUCTION FOR ENHANCED FUNCTIONALITY**

The reconstructed architecture provides the field artillery Fire support (FIST) the capability of automation, enhanced surveillance, target acquisition, target identification, target tracking, position location, and communications functionality while mounted. The system provides FIST capability to automate the command and control functions required performing fire support planning, directing and controlling cross-functional co-ordinations. The system also provides mobility, survivability, maneuverability, speed and physical signature equivalent to that of the mechanized infantry, armor and cavalry units it supports. The target vehicle (V4) is the functional integration and technology combination of three previously developed V1, V2 and V3 production systems. The new system development team are developing a block modification package for V4, a portion of which involves the inclusion of major functionality enhancements. The software portions of these functionality enhancements will enhance the following areas: 1. System control, 2. Lethality and survivability, 3. Command and control, 4. System support and sustainment, 5. Mobility 6. Fire Support

V4 will be developed in parallel with the V2-5.0 Release. Changes made to the V2 software will be evaluated for inclusion in the V4 software architecture. The existing V2 Turret Processing Unit (TPU) and Hull Processing Unit (HPU) application software will be utilized as the baseline software to develop the software for the V4 program. The existing (4.0 Release) TPU and HPU software include the above listed enhancements other than the Fire Support Computer Software

Configuration Item (CSCI). All V2 - TPU and HPU software will be retained and the reuse of this software will be maximized. The Fire Support enhancement will be added by the implementation of a V3 specific Fire Support CSCI.

## THE ARCHITECTURE RECONSTRUCTION METHODOLOGY: OVERVIEW

The V4 software system is being developed using the new software components designed to enhance the target acquisition and reused components from the V1 and the V2 production vehicle systems. The V2 is the base architecture for the hardware system on which the new system will be developed to retain the commonality between the V2 and the new V4 system architecture. The software components were identified by requirements tracing to the system level of the vehicle components.

The V4 project involved system components that were developed by multiple government agencies. These agencies worked to their own cost budgets and scheduling which initiated the challenges that we had to overcome. After the contract was signed for the V4 program and its preliminary System Requirement Review conducted there was a significant requirements change made to the base V2 system. This decision was made to have a Common Solider Machine interface to be used for all the V2 families of vehicles. The Program Managers Office had to make an effort to minimize the cost and scheduling impact this requirements change would cause the V4 program.

One of the first steps in our engineering process was to set up Information and Domain engineering. Domain engineering, that facilitates a product-line development strategy, opens up opportunities for reuse, and is integral to the requirements analysis and specification process. Information engineering is designed to maintain a stable database, while the software and procedures that automate the database are designed to accommodate change. The organization of the reconstruction process facilitated the Domain engineering effort for the V4 program. The RTM workshop database management system facilitated the Information engineering for the V4 program. These steps help support the aggressive and interdependent scheduling of the V4 program.

In order to minimize the risk of managing the requirements the team utilized the Requirements Traceability and Management (RTM) Workshop tool in V4 program. The RTM Workshop tool is used to automate the V4 requirements information gathering and engineering process, capturing and understanding the systems demands and concerns. RTM Workshop tool served as the engineering database solution for managing requirements throughout a project's lifecycle. The RTM Workshop tool proved to be a powerful information management system designed to facilitate and streamline the requirements tractability and ease the identification of software reuse components.

The RTM database was populated with both the existing system requirements and the new software component requirements. Based on the requirement similarities and matching, software components can be identified for reuse. This effort of requirement management and analysis also gave a clear understanding of the external interface the software components have and the message/data traffic between the components in the system. These functions of the RTM Workshop tool was also very useful when the base requirements of the V2 program were changed. The tool helped identify where the requirements were changed and which external components were effected by the change.

## V4 REQUIREMENT TRACTABILITY

The requirements developed for the new software components for the V4, the requirements from the V1 V2, and V3 vehicle systems were all placed in one database system. This allowed for the initial review of the requirements for matching and tractability between the existing software components and the new requirements that were developed. The key word search system within the tool was used to help in the initial identification of requirements similarity. The documentation tractability was made using the RTM tool during the system level and software component level reviews.

The key word search was not adequate for our efforts of identification of requirements similarity. The key word search produced excessive word matching where the majority was erroneous. Therefore, the V4 program requirements review process was developed. Both members of the Program Managers Office and contractor staff were involved in reviews. Reviews were both formal (with completed documents) and informal. Good communications between developers, program managers and users were set to resolve problems at an early stage. The software engineering team also worked close with the System Engineering team to make sure there was good traceably from the systems requirement to software component requirements. This gave all a good understand of which software components would integrate well for reuse and which software components would need modifications.

## THE ARCHITECTURE RECONSTRUCTION METHODOLOGY: PROCESS

The V4 program requirements review process was developed to keep a good communications base with the entire project teams. Regular reviews were held while the requirements definition is being formulated. Both members of the Program Managers Office and contractor staff were involved in reviews. Reviews were both formal (with completed documents) and informal. Good communications between developers, program managers and users were set to resolve problems at an early stage.

The software engineering team also worked close with the System Engineering team to make sure there was good traceably from the systems requirement to software component requirements. This gave all a good understand of which software components would integrate well for reuse and which software components would need modifications.

This working relationship of the PMO, software and system engineering was the new Software Architecture Reconstruction process that was organized for our program. The Software Architecture Reconstruction process should be able manage the changes in requirements, retain common-architecture, maintain configuration management and provide system support and maintenance. It is the newly organized Software Architecture Reconstruction process's responsibility to enable the members to identify, analyze, plan, track, and relentlessly control risk. Risk management costs time and money. It is always less expensive to be aware of and deal with risks than to respond to unexpected problems. A risk that has been analyzed and resolved ahead of time is much easier to deal with than one that surfaces unexpectedly.

The V4 program initiated a Risk Radar monitoring process to help mitigate risk. It is a basic risk management model that identifies the fundamental risk management actions that must be taken:

- Identify. Search for and locate risks before they become problems adversely affecting your program.
- Analyze. Process risk data into decision-making information.
- Plan. Translate risk information into decisions and actions (both present and future)

and implement those actions.
- Track. Monitor the risk indicators and actions taken against risks.
- Control. Correct for deviations from planned risk actions.
- Communicate. Provide visibility and feedback data internal and external to your program on current and emerging risk activities.

The V4 program organized the Software Architecture Reconstruction process to include the software reuse in the development process. Figure 1 shows the Software Architecture Reconstruction process for implementing opportunistic software reuse in our software development. Based on the requirement similarities and matching, software components can be identified for reuse. This effort of requirement management and analysis also gave a clear understanding of the external interface the software components have and the message/data traffic between the components in the system. The System and Software engineering team that made up our Domain engineering analyzed the requirements for the existing software architecture. This team is responsible for our software engineering efforts to retain common-architecture, re-engineering, reuse and development of new software components. External components developed such as FOS were also analyzed by the team. New or re-engineered software components were taken through our certification and validation process. The team identified the Domain models for the program. The domain models were identified as the Computer Software Configuration Items (CSCI) for simplicity. The resulting software components that were produced by our efforts are stored in our Software Repository Library and integrated into the V4.

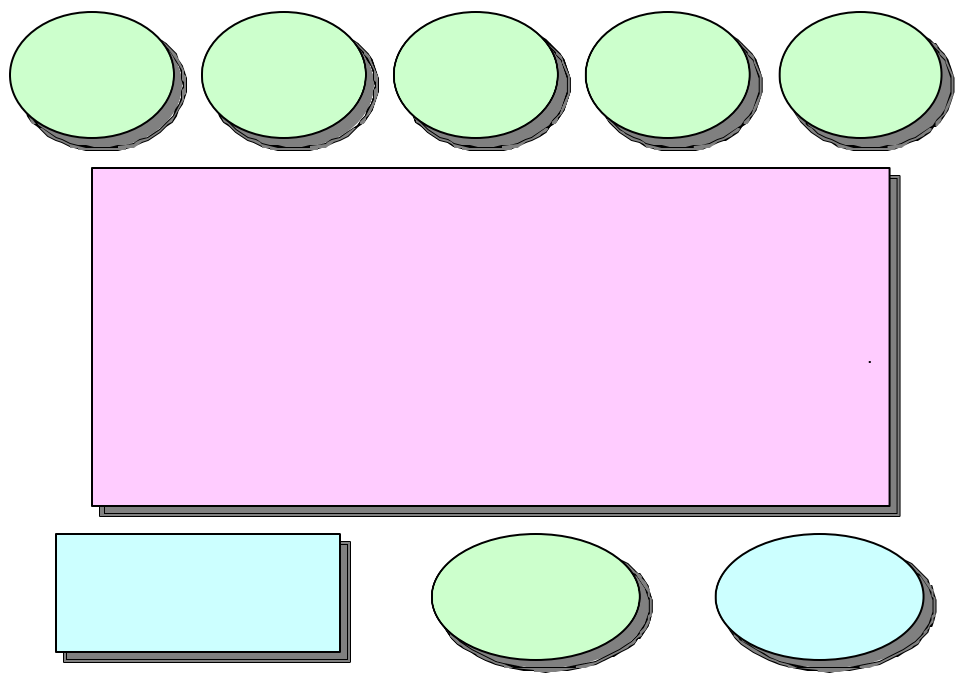

**Fig. 1. V4 Opportunistic software reuse process**

Configuration
Vehicle Reuse Com-ArchNewRe-Eng

To maintain and enhance the software components that are to be developed we also planned a Software Architecture Reconstruction process for common-software support as shown in Figure 2. This Systematic software reuse process model included a Product Line Expert Team (PLET) which is an experienced software engineering team that can understand abstract domain concepts and communicate with functional experts. The PLET would review the new requirements for the product line and determine the most efficient means to implement them. Software Configuration management would have a close involvement with the monitoring of the versions of the software components produced since the product would be fielded and there could be interoperability issues between various software versions. The common software components once produced would need to go through the integration process to best introduce them to the vehicle systems.

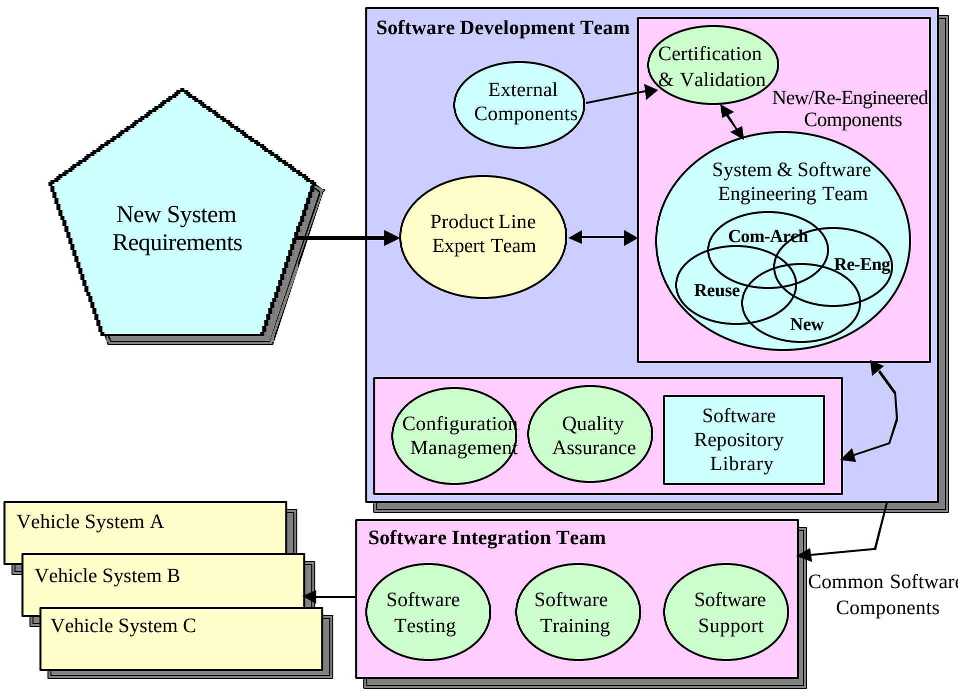


**Fig. 2. V4 Systematic software reuse process**

In order to maximize the software reuse, the architecture reconstruction methodology composed of three level requirements review process ( Fig. 3 ) :

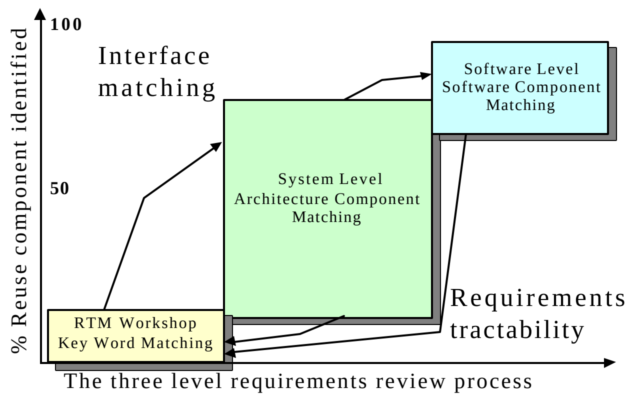


**Fig. 3. V4 Three Level Requirements Review Process**

**RTM Level -** the key word and tractability efforts to link the requirements of the existing systems to identify components for software reuse.
**System Level -** the commonality of the system architecture was used as a means to identify system level components and the interface that matched and identify components for software reuse.
**Software Level -** this review level refined the component matching that was identified by the RTM and System level analysis. This level of review finalized the identification of components for software reuse.

## CONCLUSION AND CONTRIBUTION

Software development process and tools for software reuse need to be increasingly utilized in current and future software developments and in particularly in architecture reconstruction. Software reuse is specially more favorable for government/military projects due to the extensive testing and supportability process involved with the software systems that are developed for their use.

The V4 Program is a successful project regarding software reuse in product line context and architecture reconstruction. Our efforts of managing requirements to identify components for software reuse resulted in**:** (1) Reused software components: 80% (2) Modified software components: 5% (3) New software components: 15 %.

For our V4 project we estimated about 18-22% of the cost of software code development was attributed to implementation of reusable software components. The program fell roughly around the middle of the expected cost (10-50%) per the studies completed by Prosperity Heights Software and FAA's Advanced Automated Systems (AAS). This also reflected the comparative efficiency of our reconstruction process.

Importance of software documentation was very evident in our efforts for software reuse. The documentation provided the information needed in the requirements to analysis the software components for reuse. Based on the requirement similarities and matching, software components can be identified for reuse.

The reused software components and the architecture reconstruction helped in the reduction of the cost and scheduling demands in the development of the overall software system. This effort of requirement management and analysis also gave a clear understanding of the external interface the software components have and the message/data traffic between the components in the system. Software engineering must continue to work with systems engineering for the tractability of requirements to components. Our program reduced the risk of cost and scheduling by having this review process in place.

**REFERENCE**

1. Abdul Siddiqui, PM Bradley Engineering, Software Engineering Status Reports, TACOM, Warren MI 48397, siddiqua@tacom.army.mil

2. RTM Workshop by Integrated Chipware Inc Web Link: http://www.chipware.com/

3. United Defense, A3 BFIST contractor, Technical Report, 1999.

4. Rine, D. , Nada, N., Three Empirical Studies in Validating a Reuse Reference Model, Journal of Software Practice and Experience, Vol. 30, pp. 685-722.

5.DEPARTMENT OF THE AIR FORCE Software Technology Support Center (STSC), Guidelines for Successful acquisition and Management of Software-Intensive Systems, Vol 1, Hill AFB, Utah 84056-5205. June 1996.