Reusable Architecture

Based on the US Army Bradley A3 BFIST Project

Abdul Siddiqui Software Engineering Manager

Abdul Siddiqui

NPS / US Army Bradley Engineering

TACOM, Warren MI

A3 Bradley

Agenda

- Background
- A3 BFIST Project
- Programmatic Challenges
- Software Architecture Reconstruction
- Lessons Learned
- References

Background

- Based on the requirement similarities and matching, software components can be identified for reconstruction of common architecture.
- Product line approach is specially more favorable for government/military projects due to the extensive testing and validation process involved with the software systems that are developed for their use.

A3 BFIST Project

- The reconstructed software architecture making the A3 BFIST will provide the field artillery Fire Support (FIST)
- The A3 BFIST vehicle is the functional integration and technology combination of two production systems: the M2/M3A3 and the M7 BFIST programs
- A3 BFIST will be developed in parallel with the M2/M3A3 5.0 Release, Changes made to the M2/M3A3 software will be evaluated for inclusion in the A3 BFIST software

A3 BFIST Project

M2/M3 A3

Digital Maneuverability

Integrated Command Control Communication (IC3)
Improved Bradley Acquisition System
Commanders Independent Viewer
Improved Armor Protection

M7 BFIST

Digital Fire Support

Forward Observer System (FOS)
FIST Functions
Target Acquisition
Self/Target Location
AN/GYK-37 (V)4 Computer Set

A3 BFIST

Programmatic Challenges 1

- The A3 BFIST project involved system components that were developed by multiple government agencies
- **Developing and validating requirements**
  - A3 BFIST requirements sources
  - Requirements approval and validation
  - Tracking requirements

Programmatic Challenges 2

- **Parallel development of Forward Observer System (FOS)**
  - The Fire Support Software Engineering Center (FSSEC) develops the FOS to be used in the A3 BFIST
  - The FOS software system is a ongoing contract with increased capabilities
  - FOS configuration and scheduling
  - Determine an efficient means to integrate the FOS and the A3 BFIST programs

Programmatic Challenges 3

- **M3A3 design requirement change**
  - Requirements change made to the base M2/M3A3 system
  - Appliqué Integrated Command Control Communication (IC3) system for the Battlefield Situational Awareness and tie into the Tactical Internet
  - Minimize the cost and scheduling impact this requirements change would cause the A3 BFIST program

Software Architecture Reconstruction 1

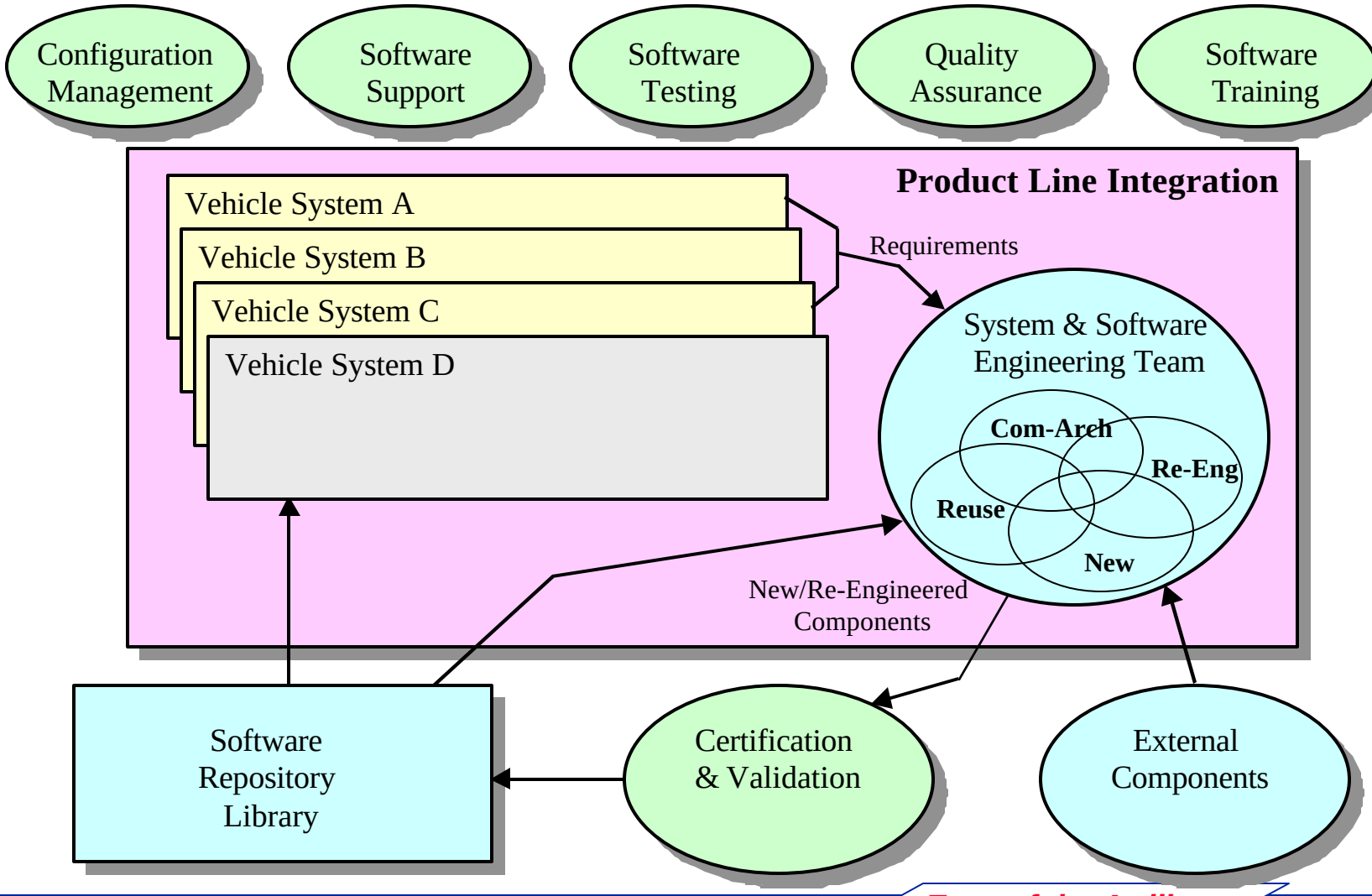
- The A3 BFIST needed to meet our requirements management and achieve our software reconstruction of common architecture
- **Prototype feasibility test**
  - A feasibility prototype testing of the new requirements was performed for A3 BFIST
  - FOS 11.016 was used as the base with the added enhancements to meet our requirements
  - Verified the functionality of the reused software components with the new software components that were to be developed

Software Architecture Reconstruction 2

- **Organization of Software Reconstruction Process**
  - Regular reviews of requirements definition being formulated
  - Software Engineering team and System Engineering team made sure there was good traceably from the systems requirement to software component requirements
  - A good understand of which software components would integrate well for reuse and which software components would need modifications were identified

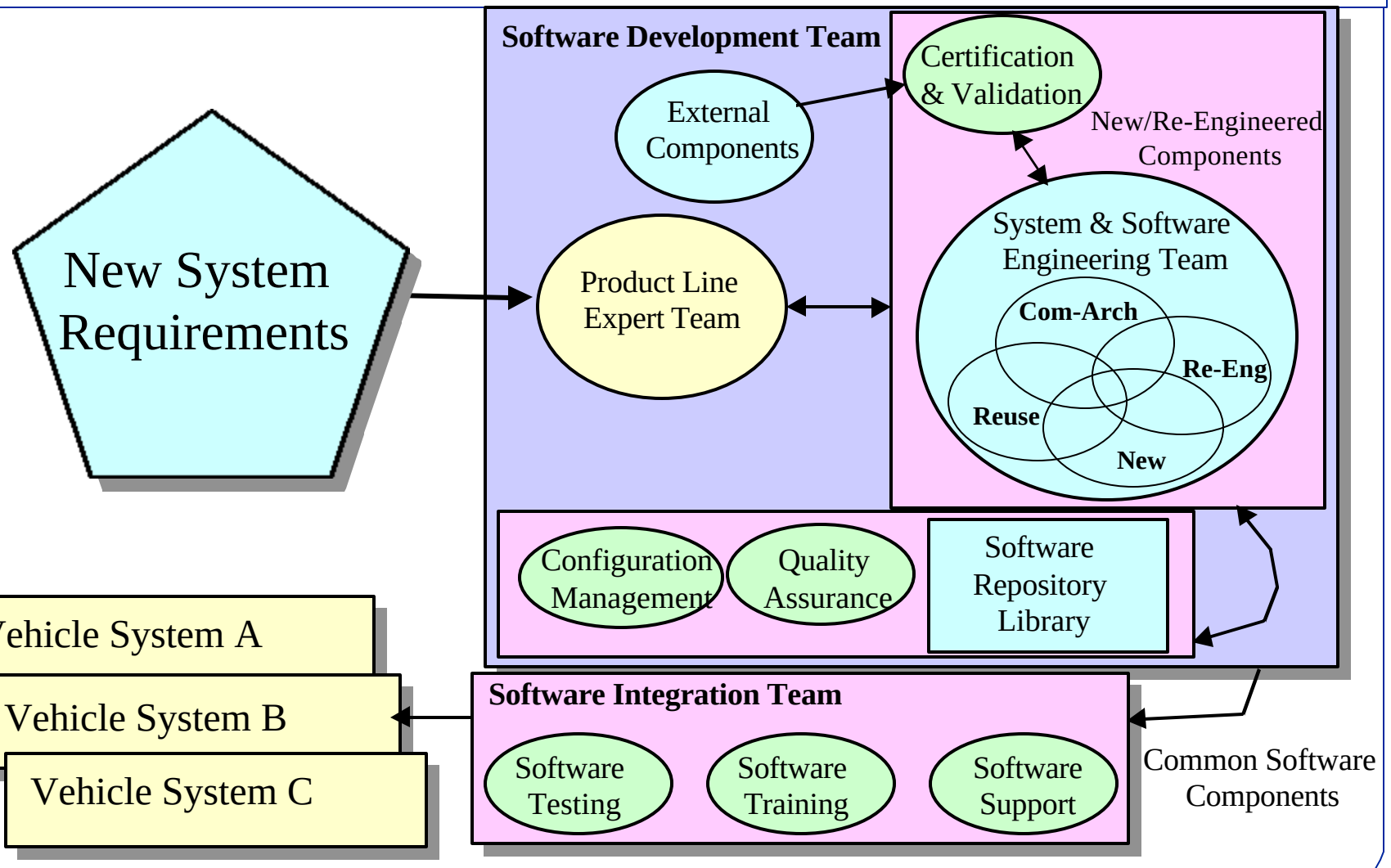
Software Engineering

Opportunistic Software Reuse Process Model



Software Engineering

Systematic Software Reuse Process Model



Software Architecture Reconstruction 3

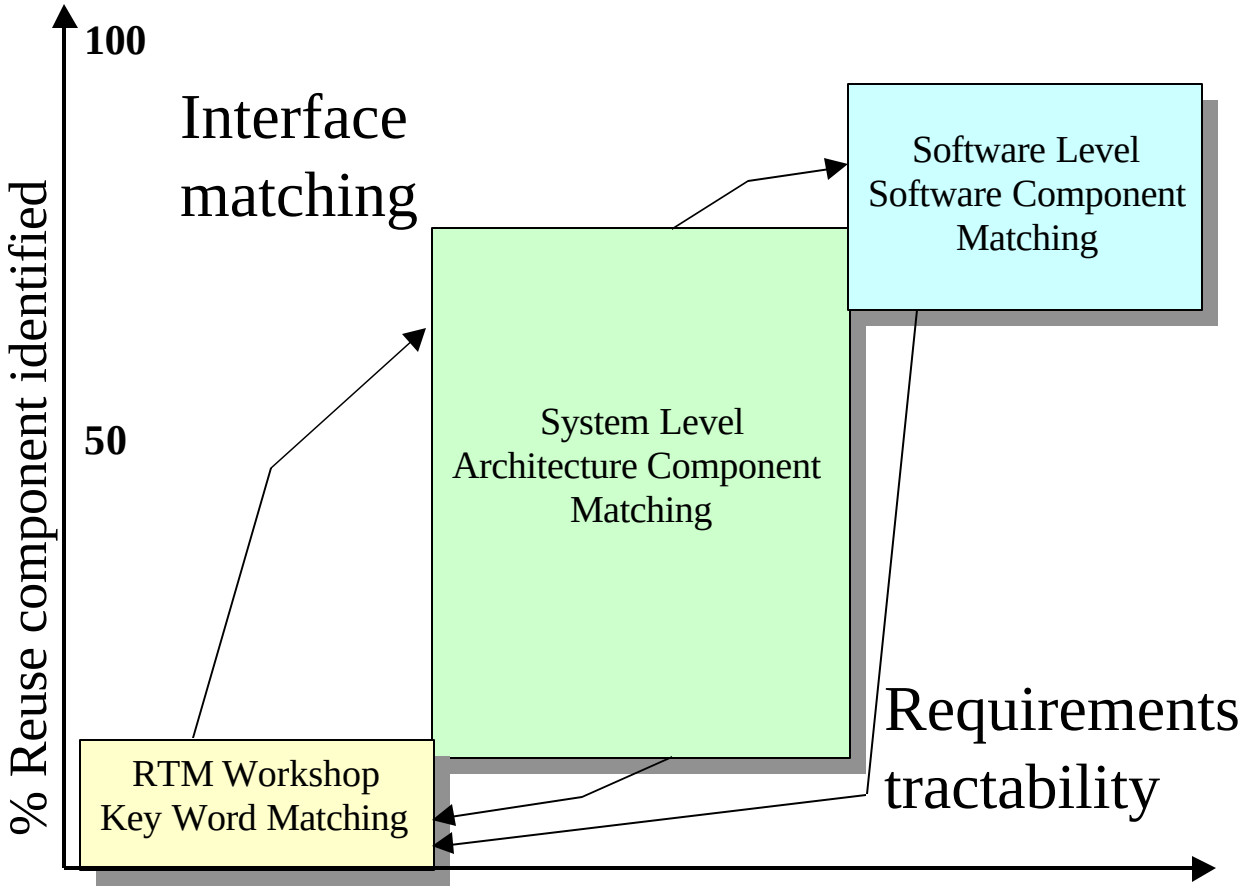
- **RTM (Requirements Traceability and Management) Workshop**
  - Automated the requirements information gathering, management and engineering process
  - Based on the requirement similarities and matching, software components can be identified for reconstruction of common architecture
  - Helped identify where the requirement were changed and which external components were effected by the change

    icExplorer --- visualization and navigation of all project data from one source

    icCONCEPT --- full-featured requirements engineering tool

    icMANAGE --- total Engineering Information Management

The Three Level Requirements Review Process



Lessons Learned 1

- Software reconstruction of common architecture was achieved by managing and tracing the requirements of the currently existing software architecture and the new requirements developed for the program
- Based on the requirement similarities and matching, software components can be identified for reuse
- Our efforts for software reuse resulted in:
  - **Reused software components: 80%**
  - **Modified software components: 5%**
  - **New software components: 15%**

Lessons Learned 2

- Importance of software documentation
- Software development process and tools for software common architecture need to be increasingly utilized
- For our A3 BFIST project we estimated about 18-22% of the cost of software code development was attributed to implementation of reusable software components
- Software engineering must continue to work with systems engineering for the tractability of requirements to components