IV&V Planning & Execution Initiative

James B. Dabney, GeoControls Systems

6 December 2018

Overview

- Research team
- Background
  - IV&V Overview
  - Capability Development Initiative Background
- Goals and objectives
- Research approach
- 4+1 Architecture Overview
- Architecture Spreadsheet Use & Examples
- Architecture-Based Risk Identification

Research Team

- Jim Dabney – GeoControl Systems (PI)
- Pavan Rajagopal – GeoControl Systems (PM)
- Mike Facemire – NASA IV&V Facility
- Paul Amoroso – Engility / TMC

NASA IV&V Overview

- Independent: Technical, Managerial, Financial
- Analytical approach to evaluate software Correctness & Completeness
- Scope
  - All NASA mission-critical software
  - Includes HEO and science missions
- Key information sources
  - IV&V Technical Framework
  - Developer artifacts

Capability Development Initiative Background

- Previous work
  - Examined IV&V Planning and Scoping (precursor to execution for every project)
  - Observed that focus limited to two architectural views
    - Capabilities (logical view)
    - Entities (implementation view)
- Other views also drive IV&V and influence risk
  - Scenarios
  - Process (threads)
  - Deployment (boxes, buses)
  - Crosscutting concepts (Technical Budgets, Stakeholders, Key Driving Requirements, Fault Management, etc)
- Other views often not explicitly documented in architecture design document

Long-Term Goals and Objectives

- Use complete architecture information to identify and evaluate risk
  - Performance
  - Safety
  - Security
  - Reliability
- Identify specific high value assurance objectives
  - Not reliably observable by SME inspection
  - Using reliable and repeatable process

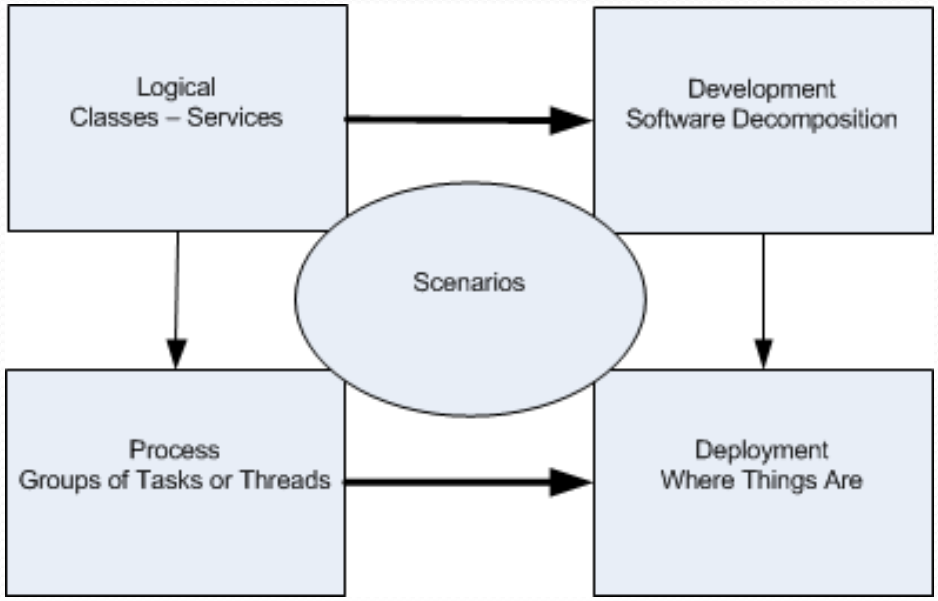
Near-Term Goals and Objectives

- Develop approach to characterize project architecture
  - Ideally lightweight process
  - Based on developer artifacts
- Develop approach to identify risks
  - Revealed by architecture analysis
  - Not apparent using SME analysis alone

Research Approach

- Partner with current IV&V projects
- Capture architectural information using 4+1 views and crosscutting concepts in hyperlinked spreadsheet
- Identify risks revealed in each view

Architecture Views & Crosscutting Concepts



- Crosscutting Concepts
  - Technical Budgets
  - Key Driving Requirements
  - Stakeholder Analysis and Needs
  - Fault Management And Redundancy
  - Information Security

Capturing a Project Architecture

- Use existing project artifacts
  - Documents
  - Presentations
  - UML and similar representations
- Document by pointing to architecture elements in project artifacts
  - Significantly less expensive than generating new document
  - Easier and less effort to maintain as architecture evolves

Available Early-Lifecycle Data

- Every project different
- General classes
  - Concept documents
  - Functional design documents
  - Requirements documents
  - PDR presentations
  - UML
- Information frequently preliminary and evolving

Capturing Architecture Data

- Spreadsheet approach found most usable
  - Tabs for each view and crosscutting concept
  - Hyperlinks to documents in Content Management System (ECM)
    - ECM limits links to document
    - File server enables links to place in document for some file types
- Developed representative example for each partner project

Example Spreadsheet

Logical view

	Level	Name	Where Defined	Page / Paragraph / Bookmark	Notes
	0	Root			
Lvl 1	1	Control loop flow	System Dynamics Concept	Page 57	Describes control loop behavior
Lvl 1	1	Data Transfer Approach	Data Transfer Concept	Page 18	Overview of data transfer capability
Lvl 2	2	Data Transfer Requirements	Data Transfer Concept	Table 11 / Page 15	Table of capabilities
Lvl 2	2	Commanding requirements	Data Transfer Concept	Table 10 / Page 19	Table of generic capabilities
Lvl 1	1	Attitude control requirements	Attitude Management FDD	Table 3-2 / Page 17	The listed requirements are much broader than attitude estimation
Lvl 1	1	Generic spacecraft subsystems	Housekeeping FDD	Table 1 / Page 7	Numerous generic capabilities here and following tables
Lvl 1	1	Timekeeping	Onboard Time Management FDD	Section 2 / Page 5	Table lists timekeeping

Architecture Capture Results

- Significant differences among projects
  - Detail lacking in early lifecycle projects
  - More detail than needed in mature projects - challenging to maintain right hierarchical level
- With some practice, capturing architecture for each project was feasible
- Relatively easy to maintain architecture spreadsheet as project evolves

Scoping / Planning Flow

- Logical flow is architecture→risks→assurance objectives
- Risks are things that can go wrong
  - Development
  - Operation
- Assurance objectives flow from risks
  - Individual IV&V analysis questions or tasks
  - Completing an assurance objective decreases uncertainty with respect to a specific risk

Risks typically hierarchical

- Top level is failure of architectural element (of view or crosscutting concept)
- Supporting are failures that cause overall failure
- Each supporting failure varies in influence on causing top level failure
  - In some cases a single supporting failure can cause top level failure
  - In other cases we need a combination of failures

Risk Identification

- Two approaches to architecture-driven risk identification considered
- Domain specific, architecture-driven risk
  - Requires extensive risk database
  - Many projects sufficiently new that database would not contain important risks
  - Doesn't appear to be practical
- Generic architecture-driven risk categories
  - Standard sets of risks and indicators matched to each element of view or crosscutting concept
  - Found workable on variety of project types

Risk Drivers / Indicators

- Drivers are factors that influence (increase or decrease) likelihood that risk will manifest
- Indicators are things we can observe or measure that correlate to likelihood risk will manifest
- Drivers are often observable, so it's reasonable to not worry about differentiation from indicators
- Drivers feed the scoping and planning process

Risk and Driver/Indicators by View

- Developed sets of risks and drivers/indicators for each view
- Applied a subset of the risks to each element in view (each row in spreadsheet)

Scenario View Risks

Scenario fails to execute as expected

- Preconditions not met
- Incorrect triggers
- Bounds exceeded
- Fails to proceed as specified
- Fails to meet end conditions (time, state)
- Scenario not expected or specified
- Scenario conflicts with other scenario

Scenario Risk Drivers / Indicators

- Stakeholder needs captured / coordinated
- Complexity in interactions
- Coupling tightness of scenarios to other scenarios
- Scenario / use case completeness
- Maturity and completeness of operations concepts
- Clarity of specification of scenario state or data boundaries

Technical Budgets Risks

Technical budget not met

- Budget infeasible
- Budget won't satisfy user need
- Budget allocation among contributors incorrect
- One contributor exceeds allocation

Technical Budget
Risk Drivers / Indicators

- Budget management / tracking process rigor
- Number of entities (users, processes, boxes) involved in the budget
- Degree of uncertainties in environment or contributors
- Budget complexity
- Budget testability

Future Work

- Extend methodology to include assurance cases tied to risks
- Develop more accurate risk likelihood model
  - Exploit existing likelihood scoring factors
  - Capture nonlinearities with respect to the scoring factors