Measuring the Effectiveness of Introducing New Methods in the Software Development Process

TESTART: ESSI Project 23683

M. Winokur, A. Grinman, I. Yosha, R. Gallant

25/8/98

Content

- What is TESTART
- The "Base Project"
- Measurable Goals & Objectives
- TESTART Project Major Steps
- Methods and Tools Selection
- Insertion of methods and tools
- Data Item Definition
- Historical Data
- TESTART project status
- Interim lessons learned

What is TESTART

- Experiment in software improvement process with emphasis in requirements management and software testing.
- Sponsored by the ESSI (European Systems & Software Initiative) .
- The project Initiated in April 1997.
- Expected completion date: March 1999.

Organizational Background

- TESTART is performed in an avionics "base project" at the TAMAM division of Israel Aircraft Industries (IAI).
- TAMAM has been assessed at CMM level 2 in May 96 and CMM level 3 in November 97.
- The "base project" is typical of embedded systems development at IAI's divisions.

The "Base Project"

- The project includes a new mission computer and the integration of new and existing subsystems.
- The mission computer contains:
  - Central processing card based on Power PC for computing and communication.
  - I/O cards for aircraft interfaces.
  - Video card for symbology and video capabilities.

Base Project Development Environment

- Two main development phases:
  - Phase 1 - Coding, unit testing and subsystem integration on PC workstations.
  - Phase 2 - Subsystem and system integration on the real target.
- Coding languages: C and C++ .
- PC development environment: Windows, compiler - Borland C++ .
- Target environment: pRIZM+ , O/S - pSOS , compiler - DiabData.

Measurable Goals & Objectives

- Initial estimates of improvement:
  - Increase requirements test coverage by 15% (80 to 95).
  - Increase portion of code exercising in testing by 15% (60 to 75).
  - Reduce integration testing phase by 5% (30 to 25).
  - Reduce the overall software testing cost by 10% (50 to 40).
- Initial estimates in comparing with the results being gathered.

TESTART Project Major Steps

- Definition of methodology.
- Tools selection.
- Insertion of selected methods and tools into the base project.
- Definition and Collection of historical data for measurement reference.
- Collection of performance data from the base project.
- Analysis of results and drawing conclusions.

Methods and Tools Selection

- Methods were defined, and supporting tools selected in the areas of:
  - Requirement management.
  - Software testing.
- These methods and tools are complementary to the existing development process at IAI/TAMAM.

Requirements Management Tool Selection

- Two commercial tools have been studied:
  - DOORS by Quality Software Systems.
  - RTM by Integrated Chipware.
- The selection process included:
  - Definition of the requirements management process.
  - Analysis of the impact of each tool's features on the defined process.
- As a result, RTM was selected

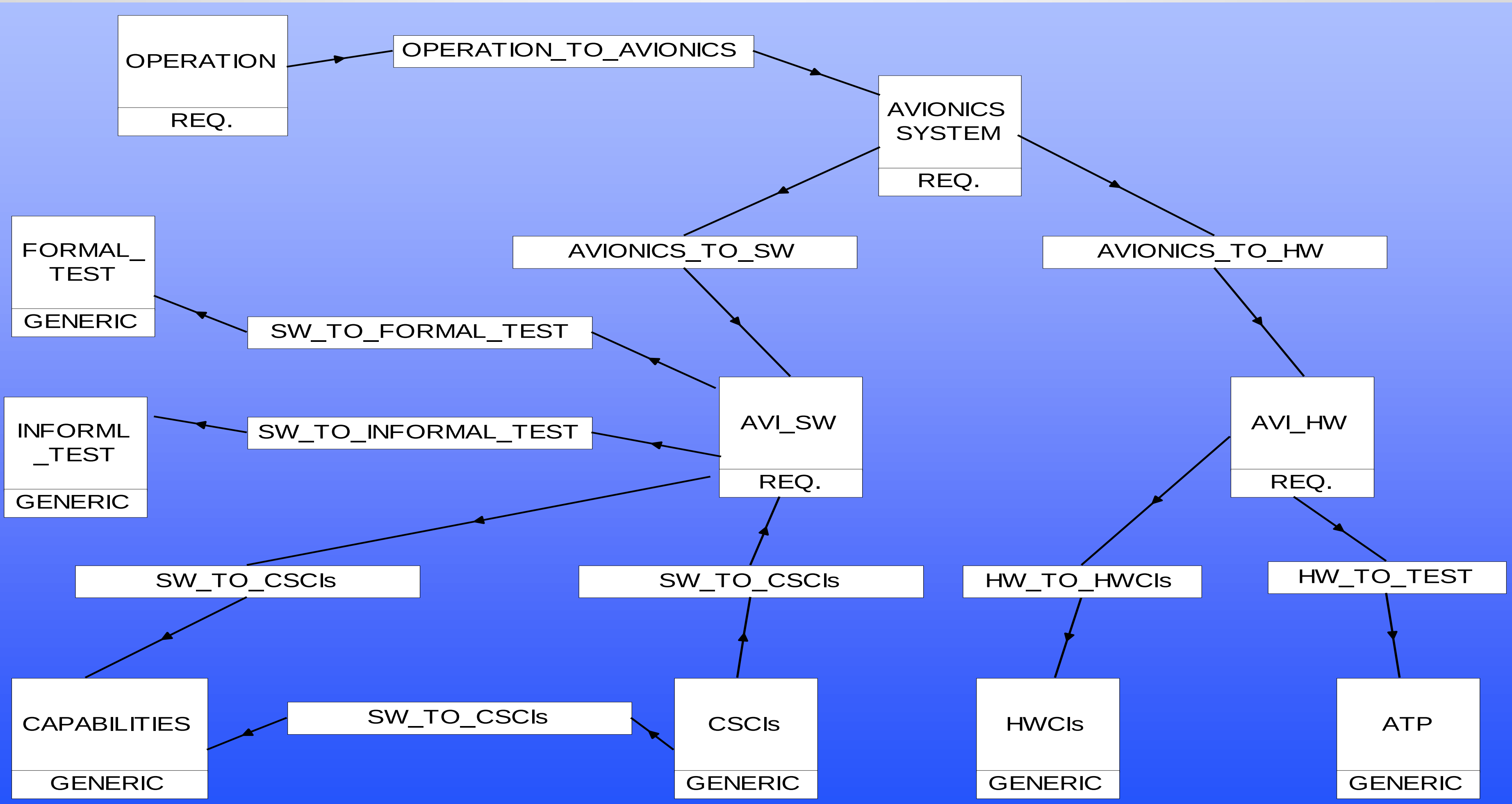
Features Supporting RTM's Selection

- Class definition diagram.
- Graphical Audit trail.
- Graphical Interface for query and reports.
- Based on commercial database ORACLE for tracking large projects.

RTM Class Definition Diagram

- User defined project specific definition diagram:
  - Classes and relationships (associations).
  - Class attributes.
  - Access rights.
- Provides utilization of project - tailored requirements management process.

Base Project class diagram



Software Testing Tool Selection

- Selection criteria for testing tool:
  - Support static and dynamic testing.
  - Support code coverage testing.
  - Automatic code generation for drivers and stubs.
  - Configuration management of test cases and results.
  - User friendliness.
  - Tool support by the vendor.
- The tool selected: Cantata from Information Processing Limited (IPL).

Insertion of Methods and Tools

- Requirements management and testing methodologies training.
- Tools training:
  - provided by senior vendor representatives at user facilities.
  - includes hands on exercising.
- Training was provided to the project technical staff and to core people within the organization.
- For each tool we tailored user manuals for project needs.

Insertion of Methods and Tools (cont)

- Tool interoperability:
  - Interface between RTM and TAMAM’s existing metric and requirements change management tool (CDSD).
  - Interface between tools and existing PC development environment (Windows, Word, ...).

Data Item Definition

- To evaluate the quantitative impact of the experiment the following data items were defined:
  - Software integration duration [hours / line of code].
  - Relative cost of software testing [test cost / total cost].
  - Coverage of requirements in software testing.
  - Software code coverage.
  - Cost of requirements change [hours / change].

Historical Data

	METRIC	Proj. A	Proj. A1	Proj. B	Proj. C
1.	Testing as a part of the overall project effort	= 15.3%	= 11%	= 16.6%	=8.57%
2.	Integration Time as a part of overall project development time	= 16.6% Sys Integ = 16.6%	= 7.3%	= 11.1% Sys Integ = 25.0%	= 30.6%
3.	Cost of Requirement Change [hours / change]	23.3	16.7	13	8.75
4.	Functional Coverage	Coverage =80%		Not available	Not available
5.	Code Coverage	Coverage =55% According to literature			

TESTART Project Status

- Definition of methodology.
- Tools selection.
- Insertion of selected methods and tools into the base project.
- Definition and Collection of historical data for measurement reference.
- Collection of performance data from the base project.
- *Analysis of results and drawing conclusions*.

Interim Results

- RTM is an integral part of the “base project” development environment:
  - Formulation of the requirements baseline including 600 main requirements.
  - Requirements change management using CDSD and RTM: 10 major requirements changes approved.
- Use of CANTATA has started for the unit test of new modules.

Cantata application: an example

- Two new units were tested with Cantata (effective C code lines: Unit A 104, Unit B 139).
- Decision, statement and assertion coverage: 100%
- Testing time: Unit A 26 hours, Unit B 35 hours.
- No errors found in Unit A and two errors found in Unit B.

Intended Use of CANTATA

- From the experienced acquired in TESTART until now, we have defined the following:
  - 55% of the code will be tested in development environment at unit test level.
  - 20% of the code (at least) will be tested on target at CSCI integration level.

Interim lessons learned

- Tools and methods training is crucial for successful insertion.
- Interoperabilty of new tools with the existing environment has practical and cultural impact.
- Early tangible benefits of tool usage are critical to acceptance by development staff.
- Gradual acquaintance with tool features increases the willingness to use them.
- Our experience until now with automatic unit testing shows that a suitable effort must be invested.