Subjects

- Introduction: VVT in the integration process
- Generic VVT integration procedure
- Key VVT integration strategies
- Sub-contractors entrapments
VVT in the Integration Process

Purpose:
- To prove that systems elements (e.g., component, subsystem) interact properly together
- To expose faults in the interfaces and in the interaction between the elements

Process:
- Continues progressively until:
  - no element has adverse impact on another and
  - the overall functionality of the system is achieved

Assumptions
- A lower level testing has been successfully performed

Rational of VVT integration

- Often, individual system elements that work fine in isolation fail to work together properly due to:
  - functional, timing or other interface problems

- In particular, VVT integration is vital when:
  - the system must meet real-time constraints
  - The system architecture involves distributed computing elements or
  - the system is composed of custom-developed hardware and software as well as Commercial-Off-The-Shelf (COTS) elements

- Generally, VVT integration will expose such system-level defects
Let's remember what is VVT

VVT = Verification, Validation and Testing

Start

VVT organization and preparations

Was the product built right?

Was the right product built?

End

System Testing

System Non-Testing

Range of system testing

Time

System Development

System Production

System Use/Maintenance & Disposal

(1) Sanity testing

(2) Exploratory testing

(3) Regressive testing

Component & Subsystem testing

Integration testing

Qualification testing

Acceptance testing

Accreditation testing

First article testing

Production testing

Installation testing

Maintenance testing

Disposal testing
Generic VVT integration procedure

**Step 1: Prepare VVT integration process**
- define integration test objectives,
- determine overall integration strategy,
- develop an integration test plan,
  - System Integration Test Plan (SysITP)
- design, create and verify the test case suites
- Define and build the test infrastructure
  - System Integration Test Description (SysITD)

**Step 2: Select component to be integrated**
- use integration test plan to select a component to be integrated with the current system
- use ingenuity and flexibility if said components are not available
- update the integration test plan as needed
Generic VVT integration procedure (cont.)

Step 3: Integrate selected component with the system

- connect the selected component to the rest of the system by:
  - replacing a temporary driver or stub with an actual component or subsystem
  - performing a rudimentary operational verification of the updated system

Generic VVT integration procedure (cont.)

Step 4: Perform integration tests

- Execute the appropriate test suite. This include:
  - functional testing
  - structural testing
  - temporal testing
  - performance testing
  - etc.
Generic VVT integration procedure (cont.)

**Step 5: Update relevant testing documentation**

- updated typically effected system integration test documents:
  - System Integration Test Report - with test results
  - System Integration Test Plan - if integration test planning should be changed
  - System Integration Test Description - if integration test procedures should be changed

VVT integration test strategies

- Different VVT integration strategies are used under different circumstances.
- The most prevalent ones are:
  1. Top-Down,
  2. Bottom-Up,
  3. Sandwich and
  4. Big-Bang
Top-Down Integration Testing (Example)

Simulated environment (A)

Subsystem (E)
Subsystem (C)
Subsystem (D)
Component (E)
Component (F)

Integration order / Time

Top-Down Integration Testing (Illustration)

Round I
Simulated environment (A)
Stub 1
Stub 2

Round II
Simulated environment (A)
Subsystem (B)
Stub 1
Stub 4

Round III
Simulated environment (A)
Subsystem (E)
Stub 3
Stub 4

Subsystem (C)
Stub 5

© Copyrights - Dr. Avner Engel
Top-Down Integration Testing (Advantages)

- It is possible to obtain an early system visibility
  - Depicting / simulating the general behavior of final system
  - Helps in early discovery of major design flaws

- Incremental approach to testing
  - Simplifies fault localization

Top-Down Integration Testing (Exercise)

INCOSE-IL Systems Testing Working Group (STWG):
Team exercise
Bottom-Up Integration Testing (Example)

Integration order / Time

Bottom-Up Integration Testing (Illustration)
Bottom-Up Integration Testing (Advantages)

- integration and testing can be undertaken early
  - in parallel with the implementation of the system

- upper-level testing tends to be more thorough
  - lower components and subsystems are fully developed objects rather than stubs

- often more “natural” to testers

- incremental approach to testing
  - simplifies fault localization

Bottom-Up Integration Testing (Exercise)

INCOSE-IL
Systems Testing Working Group (STWG):
Team exercise
Sandwich Integration Testing (Example)

Sandwich Integration Testing (Advantage)

- each half of the system can be integrated and tested in parallel
  - could significantly shorten the integration and test period

- incremental approach to testing
  - simplifies fault localization
Big-Bang Integration Testing (Example)

![Diagram of Big-Bang Integration Testing]

Big-Bang Integration Testing (Advantages)

- Effective for very small and simple systems
- Assumes all components and subsystems have been thoroughly tested
- No needs for:
  - Test stubs or
  - Test drivers
Life is not always rosy

1. Components are often not available for VVT integration as planned
2. Components delivered for integration testing are often defective:
   - Sub-contractors transfer debugging costs to main-contractors
3. VVT integration process is cyclical, not linear

Solution-I: Enforce Main-contractor strength

- Do not accept defective components for integration testing!
- Impose contractual fines on:
  - late deliveries!
  - defective deliverables!
**Solution-II: System Integration Laboratory (SIL)**

- **SIL operator**
- **Virtual system control bus**
  - Physical support systems
  - System environment simulation
  - DATABASE
  - Virtual Subsystem-I
  - Virtual Subsystem-n
  - Real System
    - Real Subsystem-A
    - Real Subsystem-B
    - Real Subsystem-C
  - Real system bus

---

**Summary – we discussed**

- Purpose, process, assumptions and rational of VVT Integration
- Five generic procedure steps in VVT Integration
- Four key VVT integration strategies
- Sub-contractors entrapments
Introduction
• System Engineering
• VVT Engineering

Systems VVT processes
• Lifecycle VVT activities
• VVT methods

Modeling VVT processes
• Process modeling
• Optimization
• Extended example

The End