November 16-19, 2009 ~ Santa Clara Marriott, Santa Clara, CA

SMI CTP Explained

James Rigger

SNIA Conformance Testing Programs Manager

Agenda

- Stakeholders
- Finished Product Target
- Costs
- The Value of Use Cases
- Test Development Cycle
- Risks
- Contacts and Web Sites
- Q & A

Stakeholders

- Volunteers
  - Specification authors
  - Volunteers at large
- Code Writers
- Users
  - Members
  - Other End Users
- Staff
  - Program Management

Finished Product Target

- Test Plan
- Test Specification
- Test
- Users Guide
- Application and Agreement
- Order Form

Costs

- Volunteer Time
  - Opportunity time they sacrifice
- Equipment
  - Required for development and basic QA
- Coding
- Legal
  - Organization legal to draft and approve forms
  - Member legal to execute forms (A&A)
- Staff
  - General administrative duties

Test Development Cycle

- “Locked” Specification
- Volunteer Commitment
- Needs Analysis
- Develop Test Plan
- Develop Test Specification
- Coding and QA
- Trial Use
- Maturity

Test Development Cycle

- "Locked" Specification
  - The test cycle will rely upon a level of the specification that is somewhat locked.
  - The SMI, for example, uses its "implementation" version as the test basis.
  - This level needs to be approved, or bought into, by an appropriate level of the members of the organization.

Test Development Cycle

- Volunteer Commitment – the ability for volunteers to work on the elements of test development
  - Must have a champion who the group trusts to move things forward on their behalf
  - Must have support of the group to review and approve various documents and work as upon creation.

Test Development Cycle

- Needs Analysis
  - Bring forward list of bugs from previous releases as a start
  - Add features requested from stakeholders
  - Always improve use for speed and usability
  - Obtain buy-in from governance group and marketing group

Test Development Cycle

- Develop Test Plan – The test plan is critical as it provides the roadmap for test creation and bounds the work so scope creep does not endanger the project.
  - Created from the needs analysis
  - Created by authors and staff
  - Incorporate new features and enhancements
  - Describes development management
  - Reviewed by authors and code writers
  - Ballot to essential group to lock in place

Test Development Cycle

- Develop Test Specification
  - Determine optimal format
  - Bring forward fro previous if available
  - Roll in contents of test plan
  - Request review from author community
  - Request review from coding group
  - Rework if required
  - Ballot to appropriate groups to lock high-level concept in place

Test Development Cycle

- Coding and QA
  * NOTE – The quality of the code is directly proportional to the quality of the test specification!
- Be as modular as possible
- Run unit tests on known areas of change
- White Box Tests
- Black Box Tests

Test Development Cycle

- Trial Use – allows the prospective users to run the code under development to assist with debugging and eliminate surprises
  - SMI uses the SMI lab as a forum for this exercise – it has been very successful!
  - This phase has greatly reduced the number of “drops” required to complete the test
  - This phase allows the users to be in sync with the test so that they can be ready day 1
  - Problems are entered into bug tracking system

Test Development Cycle

- Maturity Model – validates that success has been achieved compared to plan
  - Organization must have a model that is agreed upon. SMI, for example, requires 2 separate implementations of the same profile to succeed
  - Do not accept “OEM” implementation as the second as the base code is usually the same
  - Users must submit full results to intermediate party
  - Archive results for audit purposes

Test Development Cycle

- Final Use
  - Make users aware of availability
  - Market availability to outside world
  - Show users results on web

    http://www.snia.org/ctp
  - Is it time to start over?

Test Development Cycle

- Lessons Learned
  - File drive test as much as possible
  - Have test “act like a user”
  - Pre-parse input files at start of test to save user time
  - Gather environmental information for reruns
  - Be able to log as much detail as possible
  - Provide “tamper” checks for integrity purposes

Risks

- Specification Interpretation
- Buy-In When Required
- Volunteer Availability
- Vendor Development Cycles
- Financial Constraints