

Oyster Creek Development Process

Ray Gagnon
SCIENTECH, LLC

August 2004

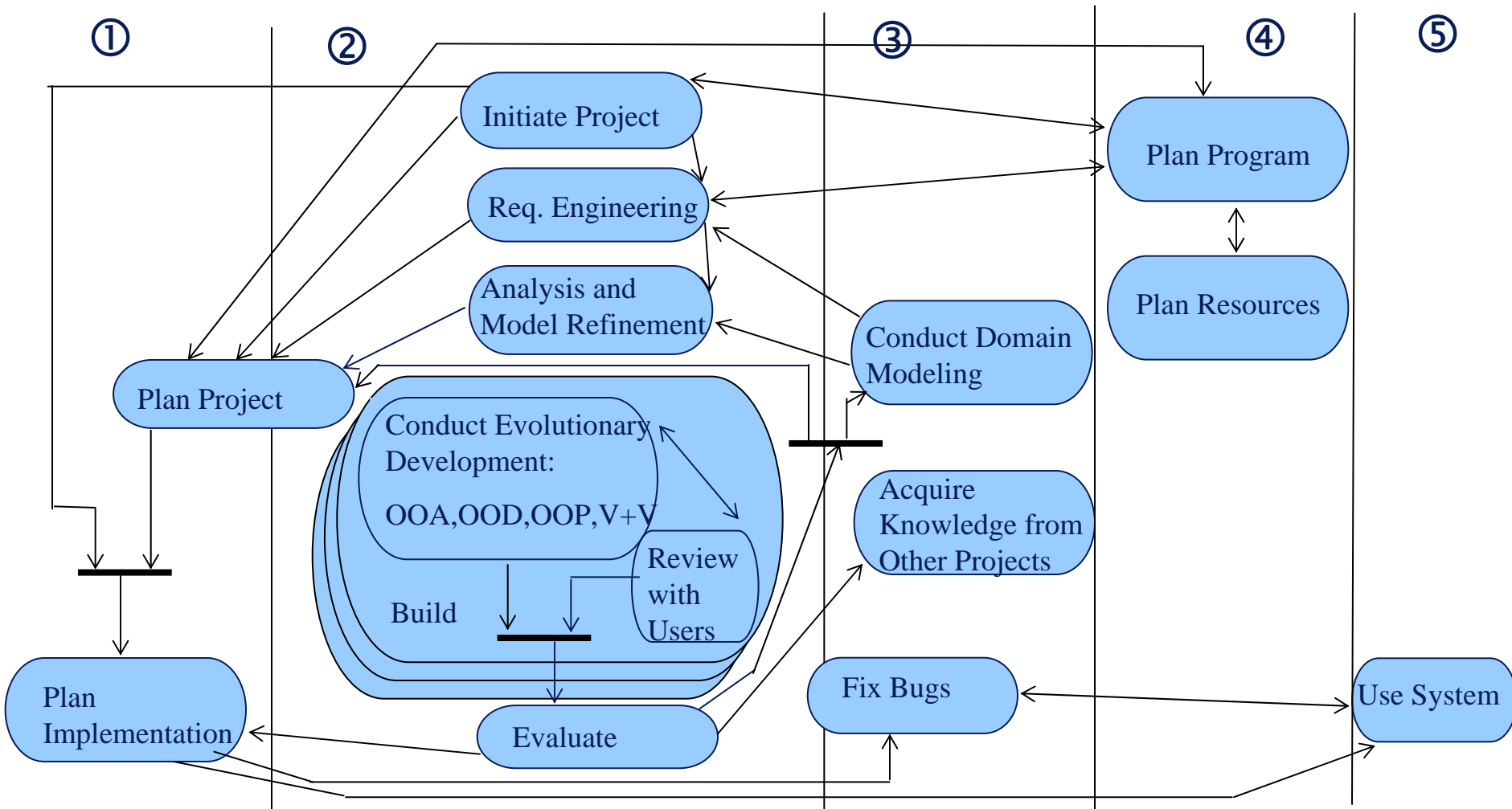


2004 Performance Software User's Group Meeting



Development Model

OPEN Framework (Graham et al., 1997)



Model Implementation

- ❖ Model Recognizes Iterative Process
 - Model Implementation limits iteration
 - Design and Requirements Definition Evolve with System Knowledge
 - Process may be iterative but can't be infinite. As implemented, process allowed for a maximum of 3 design update cycles

- ❖ Design Process Extension
 - Same process used test phases or project
 - High level of plant involvement

Design Phase

- ❖ Design Basis
 - Plant Provided SRS documents
 - RTIME Documentation and Training
- ❖ Turn SRS into Software Design Description quickly as possible to flush out gross misinterpretation early in process.
- ❖ Use Review Process to Refine Design as well as to clarify Requirements Elicitation

Development Process (Cont.)

- ❖ Early Initial Implementation Provided
 - Provided initial application and HMI to aid in design review and validation.
 - Environments provided on Engineer laptops to facilitate site review.

Test Process

- ❖ Like Design, Iterative Process
- ❖ Test Case Developed
 - Many Customer Provided
 - Cases Run and Validated by Developer
 - Procedure “Dry-Run” with Customer assistance
 - Official FAT conducted with Customer witness/assistance

Test Process (Cont.)

- ❖ Automated Testing used to the maximum extent possible
 - Benefits
 - Facilitate Retest and Regression Testing
 - Test Repeatability
 - Reduce Test Time
 - Reduce Manual Labor
 - Drawbacks
 - Test cases can be difficult to craft
 - Possible tendency toward “Over Testing”

Process Summary

❖ Benefits

- Based on Current Software Industry Thinking and Models
- Highly Customer Interactive
- Customer Focused
- Good for Fast-Track Projects
- Tends to avoid late discovery issues

Process Summary (Cont.)



Drawbacks

- Scope Creep Needs to be managed
- Requires Training
- Culture Issues

Fact or Artifact?

- ❖ Artifacts of the Process
 - Requirements Documents
 - Design Documents
 - Test Case and Procedure Documents
 - Test Result Documents
 - Etc.
- ❖ Are these Artifacts Facts?

Fact or Artifact? (Cont.)

- ❖ A general truism in software design is that:
 - Requirements are Ambiguous
 - Requirements are Incomplete
 - Requirements are Inconsistent
 - Requirements are Numerous

- ❖ All of the above prove to be consistently true. OPEN Framework strives to minimize impact.

Lessons Learned

- ❖ Heavy involvement by the client in developing SRS documents is a strong positive.
- ❖ Client Involvement in FAT Dry Run is a strong positive
- ❖ Removing Hardware from Fixed Price, Fixed Scope is a strong positive for contract change avoidance and flexibility.
- ❖ Requirements should be elicited based on what it should do not in terms of how it works now.
- ❖ In General, requirements information is aged and should be questioned for current validity.

Lessons Learned (Cont.)

- ❖ All stakeholders should be involved *Early* in the process (IT, Engineering, Ops, etc.)
- ❖ To optimize design cohesion and consistency, system should be viewed from a top level perspective.
- ❖ Open Communication Essential to analyzing why schedules/requirements change.
- ❖ Simulator work should start earlier
- ❖ Requirements will evolve