Optimizing The Software Development Process

Scientech Symposium 2013

Ray Gagnon
Since the beginning of SW development, the development process has been debated.
- Waterfall?
- Agile?
- Hybrid?
- Others?
Waterfall?

- Waterfall
  - Based on hardware design practices
  - Structured
  - Easy to understand
  - Predictable?
  - Minimize “Rework”?  

- Pros
  - Easy to understand
  - Structure
  - Rigid
  - Cost

- Cons
  - Structure
  - Rigid
  - Cost
Actual Implementation

- Requirements
- Design
- Implementation
- Verification
- Acceptance
Agile – How “Agile” can we be?

- Proponents argue addresses unknowns faster
- More collaborative process
- Actually traces back to 1957
- Increased Customer Satisfaction

Pros
- Conceptually faster process
- More customer involved
- Iteration allows for continuous improvement

Cons
- Potential “endless loop”
- Cost control
- Concept of “rework”
Case Studies

NITSL

- Two utilities presented their experience with an Agile “experiment”.
- Not sure either fully understood what they were doing.
- Mixed but generally positive results
  - One followed process more literally
    - Appears to be producing the desired product/quality
    - PM frustrated – “Done” seems to be undefined
  - One followed a bit more of a hybrid.
    - Had “90%” requirements definition
    - Incremental releases within implementation group
    - Determining how to address SQA
Scientech Process – Some History

- In the “early days” – Process? What Process?
  - Documentation afterthought

- Moving to the current era
  - Moving up on CMM scale
  - Lessons Learned
  - Increased rigor
  - Zero defect target
  - Cost control

- Solution
  - Strict adherence to Waterfall Model
How Is It Working?

The debate goes on....

- **The Scientech Approach**
  - Work upfront to get solid requirements (SRS) set in “stone”
  - Follow with highly detailed design (SDD)
    - Content and level of detail have varied over time.
  - Heavy customer involvement in requirements and design review process

- **Concept:**
  - Catch errors early at a point when cost is lower to correct.

- **Practice:**
  - Protracted review cycles costly from an internal perspective as well as customer perspective
  - Increased schedule time required to accommodate numerous review cycles
  - Significant errors and omissions still appear late in process
The issue of the “unknown”
- Process does not accommodate the unknown very well
- Ambiguity or missing requirements discovered late
- Especially true of complex systems/interfaces
- Results can be significant redesign late in process

“Customer” Issues
- Sorry folks, but frankly you are not all that good at articulating requirements!
- Tend to think in terms of “how” not “what”
- Customer reviews tend to miss the same things internal reviews miss and focus on wrong content.

Implementation Issues
- What looks good on paper doesn’t always look so good in practice.
- Stick with it anyway because it’s approved?
Revisiting The Concepts

i.e. lessons learned, continuous improvement

- **Concept:**
  - Heavily involving customer in requirements and design decreases risk and cost. Get customer buy-in.
  - Reality? Not so much. Even with fully approved documents, customers will be customers… You want what you want and we still end up “owning” most of the risk.

- **Concept:**
  - Not starting development reduces time and cost.
  - Reality!
Revisiting The Concepts

Process Improvement?

- **Scientech Concept:**
  - Internally, we have been contemplating reducing customer involvement in design and looking for increase focus in testing and verification.
  - Scientech ultimately responsible for design. Exception might be for something with significant unknown or risk.

- **Cultural Changes:**
  - Scientech staff needs to have a more “questioning” attitude when reviewing and accepting requirements.
  - Customer staff needs to focus more on requirements, especially on missing information and error conditions.
TMI - The perfect proving ground?

TMI Process

- TMI Provided SRS’s
  - Scientech, review, revise, accept
  - Scientech develop SDD –
    - Multiple internal reviewers.
    - Release for development.
    - Developer Revise in process
    - Pre-FAT review
  - Scientech Incremental releases
  - TMI Test Case Development
    - TMI providing some test cases.
    - Heavy involvement in FAT reviews
Hybrid Waterfall - Agile

- Start with solid (90% or better) requirements
- Produce solid, not necessarily final SDD. Assure all functional requirements are covered
- Developer will produce internal revisions and releases.
  - Internal Reviews as Necessary
  - Incremental releases to Customer
  - Formal review as precursor to FAT
- SRS is FAT input – SRS should be As-Built
- Future Risks/Concerns – Tendency to compress schedule and reduce interim quality checkpoints.