



44<sup>TH</sup>  
*Annual Meeting*

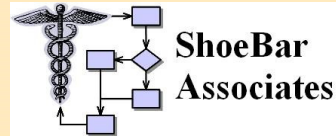


*Boston 2008*

# Validation and Quality Concepts in Open Source Clinical Software: Not an Oxymoron

**Brian Shoemaker, Ph.D.**

Principal Consultant  
ShoeBar Associates



## *Open Source Software Quality: Not an Oxymoron*

### **Could open-source approach be an alternative after all?**

Contradictions are actually superficial

Practical open-source implementation requires discipline

Still other quality / validation opportunities to explore

Model is win-win because all participants have an incentive to guarantee quality



## Could OSS be an alternative after all?

---

*My own reaction to the concept was skepticism*

### Cathedral:

- Close control of all output
- Proprietary product
- Restricted team, specific assignments

### Bazaar:

- Ideas from anywhere
- Access to source for all
- Any developer can “scratch an itch”



3

44<sup>th</sup>  
Annual Meeting



## Could OSS be an alternative after all?

---

*Long-standing issue: CSV = costly overhead*

- Often, S/W purchases only follow protracted evaluation
- Just planning / executing IQ is a project in itself
- How much do we learn from vendor audits?
- No one likes the CSV team when we find errors; no one appreciates us when we don't

Though this topic will largely be left open, it forces us to consider whether a different approach to the SDLC could accomplish more



4

44<sup>th</sup>  
Annual Meeting



## Could OSS be an alternative after all?

---

### *Open Source vs. S/W Quality / Validation – first understand what it **isn't***

- NOT an undisciplined hackers' free-for-all
- NOT a collection of tangled code with no rules
- NOT a case of constantly "shifting sands"
- NOT a nebulous, undefined end product without focus

Our focus needs to shift to what OSS **is**, to understand where validation fits in



5

44<sup>th</sup>  
Annual Meeting

## *Open Source Software Quality: Not an Oxymoron*

---

*Could open-source approach be an alternative after all?*

### **Contradictions are actually superficial**

Practical open-source implementation requires discipline

Still other quality / validation opportunities to explore

Model is win-win because all participants have an incentive to guarantee quality



6

44<sup>th</sup>  
Annual Meeting

## Contradictions are Superficial

---

Concern of CSCT Guidance: functioning system, not how system was developed

- Usage defined in the Study Protocol and controlled by SOPs
- Provide mechanisms to retain and preserve data
- Safeguards for limited access, keeping of audit trails, and date/time stamps of records
- External safeguards to limit access
- Features to minimize data entry errors
- Developed, maintained, and used by users with documented training and skills for their tasks

***Data must be Attributable, Legible, Contemporaneous, Original, and Accurate (ALCOA)***



7

44<sup>th</sup>  
Annual Meeting



## Contradictions are Superficial

---

For the “how” of development, we need a CDRH document: General Principles of Software Validation

- GPSV *does not recommend* one specific software lifecycle model
- Rather, GPSV calls for specific activities and outputs:
  - requirements
  - design
  - code
  - unit/integration test
  - system test
  - acceptance



8

44<sup>th</sup>  
Annual Meeting



## Contradictions are Superficial

- In OSS, key concepts are Responsibility augmented by Community (a new idea)
- We've always known:  
**The waterfall leaks**  
(even Winston Royce knew this!)

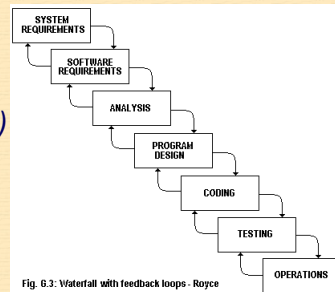


Fig. 6.3: Waterfall with feedback loops - Royce



## Open Source Software Quality: Not an Oxymoron

*Could open-source approach be an alternative after all?  
Contradictions are actually superficial*

**Practical open-source implementation requires discipline**

Still other quality / validation opportunities to explore

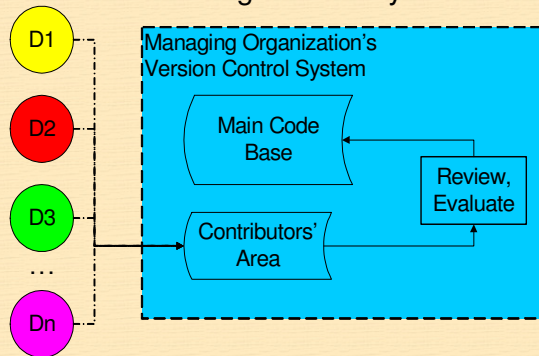
Model is win-win because all participants have an incentive to guarantee quality



# Implementation Requires Discipline

Consider models for Open Source development (1)

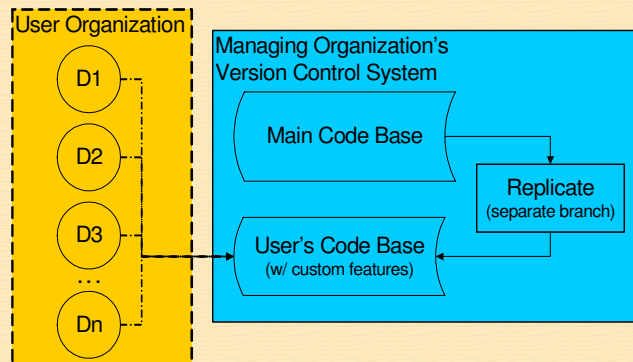
“Contributing Community” Model



# Implementation Requires Discipline

Consider models for Open Source development (2)

“User Customization” Model





## Implementation Requires Discipline

---

### Effective use of technology makes sharing possible

- Online config management: code augmented, version control maintained
- Config mgmt, bug tracking and other tools (also OSS) can be tightly integrated
- Forums / Blogs allow presenting/discussing ideas
- Public bug lists keep the community informed of issues
- Wikis allow the supporting information structure to grow
- Readily available coding standards inform contributors what's expected



## Implementation Requires Discipline

---

### *Openness invites scrutiny*



...Therefore we expect the managing organization to be careful and complete –  
*with documentation as well as code*



## Open Source Software Quality: Not an Oxymoron

---

*Could open-source approach be an alternative after all?*

*Contradictions are actually superficial*

*Practical open-source implementation requires discipline*

**Still other quality / validation opportunities to explore**

Model is win-win because all participants have an incentive to guarantee quality



## Other Open Source Opportunities?

---

- **Requirements development –**  
*Could the blog be taken one step further?*
- **Code review?**  
*Invite community developers to read documents and code for a new element, and post questions.*
- **Test writing –**  
*What if the community were invited to post test procedures for new features?*
- **Guiding concept: fresh eyes see an issue in a different way**





## Open Source Software Quality: Not an Oxymoron

*Could open-source approach be an alternative after all?*

*Contradictions are actually superficial*

*Practical open-source implementation requires discipline*

*Still other quality / validation opportunities to explore*

**Model is win-win because all participants have an incentive to guarantee quality**



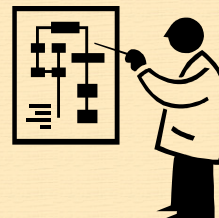
17

44<sup>th</sup>  
Annual Meeting



## Win-win model: all participants have incentive to guarantee quality

- No substitute for "making sure the software does what it's supposed to do"
- No one would seriously put out CT S/W without supporting material (per GPSV)
- Nor would any organization implement that S/W without doing validation homework



18

44<sup>th</sup>  
Annual Meeting



## References

---

FDA, *Guidance for Industry: Computerized Systems Used in Clinical Investigations* (May 2007),

<http://www.fda.gov/cber/gdlns/compclintrial.htm>

FDA, *General Principles of Software Validation* (January 11, 2002),

<http://www.fda.gov/cdrh/comp/guidance/938.html>

Raymond, Eric S., *The Cathedral and the Bazaar*

<http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/>

Royce, Winston W., "Managing the development of large software systems: Concepts and techniques," in: *Proceedings, IEEE WESCON* (August 1970). (*Waterfall method*)

