Better Functional Specifications – A Cooperative Approach

Fabrizio Stortoni
Steve Jaworski

Overview

- Setting the Stage
- Reasons for a Functional Spec
- What Testers Have to Offer
- Gaining Acceptance
- Nuts and Bolts
- A Case Study

Setting the Stage

- Types of requirements
  - Product (customer) requirements
  - ✓ ***Functional Requirements***
  - Test requirements
  - Etc. …
- Requirements related activities
  - Gathering requirements
  - ✓ ***Writing good (useful) requirements***
  - ✓ ***Managing requirements***
  - Etc. …

Setting the Stage

- Software development teams vary in
  - The relationship between testers and developers
  - The nature of the development practices
  - The extent to which they believe in the need for a Functional Specification
- Will focus on techniques that are widely applicable and scalable

Relationship between Test And Development

- Often seen as adversarial
  - You figure out how to build it
  - I’ll figure out how to break it
- Let’s figure out how to build and test it so that we make sure it works

Nature of Development Practices

- Some shops
  - Don’t have testers
  - Don’t involve testers except at the end of development
  - Don’t tell testers or coders how the software is intended to work
  - Write requirements that can be understood by only two people in the entire world
  - Commit the cryptic requirements to email, or notepad
  - Produce a nice functional requirements document, and have it reviewed by all the affected parties, but only after the product has been coded

Why Some Avoid Functional Specs

- Believe it’s a waste of time
  - No visible increase in the end product (code)
- Other reasons
  - Coding is fun
  - Code is the end product, don’t need a blueprint to make CDs
  - Testing is fun
  - Want to avoid negotiation

Reasons…

- To have a functional specification?
  - Allows different groups to work in parallel
    - Supports software design
    - Supports test design
    - Supports technical documentation
  - Saves everybody's time because questions are answered only once
  - Defines an agreement on deliverables between management and project team

Reasons…

- To have a functional specification? (continued)
  - Defines an agreement on deliverables among all affected parties (think about Field Support)
  - Reveals hidden assumptions
- And the BP (Biggest Payoff) – it enables the MCEPDR (Most Cost-Effective Practice for Defect Removal)

Reasons…

- To keep a functional specification up to date?
  - Makes the product maintainable
  - Simplifies adding team members to the project

Life Cycles

- Waterfall
- V
- Iterative
- Agile

What Testers Have to Offer

- Their questions
- The ability to take a natural language description and play computer
- The ability to play end-user

Gaining Acceptance

- Always think about and publicize
  - What you need from other team members to do your job as a (developer, tester, writer, PM, …)
  - What you can do to help move the project forward
  - How you as a (developer, tester, writer, PM, …) can help other team members
- Respect others as you wish them to respect you
  - Cowboys?
  - Process Mongering Obstructionists?

Gaining Acceptance

- Strive to become an Expert
  - On your product
  - On your discipline
  - On what other disciplines contribute

Functional Specification – Ideal

- Based on solid requirements
- Gives designers all the information they need to design the product
- Gives testers all the information they need to test the product
- Gives writers all the information they need to document the product

Functional Specification – Reality

- Requirements change
- Some functionality can’t even be foreseen before development starts
- The project needs to move forward

How to Move Forward With a Less-than-utopian FS

- How to recognize information is missing?
- How to collect the missing information?
- How to feed missing information back to the functional specification?

Nuts and Bolts

- First pass functional specification
- Functional specification checklist
- Functional specification review log
- Collaboration agreement
- Team tutoring sessions
- Other Tips and Tricks

First Pass Functional Specification

- As complete as possible
- Reviewed by all affected parties

Functional Specification Checklist

- What information does the functional specification need to provide
- This checklist suggests:
  - To testers – how to review the specification
  - To developers – what to put in the specification

Checklist

- FS Should Include
  - Labels of things the user can see (Name, Address 1, Address 2)
  - Lists of selectable items (New, Save, Save as…)
  - Things the user can do (Add, remove from shopping cart)
  - Things the user can't do (Add more than 25 items to a shopping cart)

Checklist (continued)

- FS Should Include
  - Processing rules (The end date is calculated from the start date and number of days)
  - Where values come from (The list of items available to delete is the same as the contents of the shopping cart)
  - Business rules (Operator must acknowledge that alarm condition returned to normal)

Checklist (continued)

- FS Should Include
  - Implementation details that affect expected results (Velocity information is in meters per second)
  - Error recovery (What happens if I try to save with incomplete information)
  - Error messages
  - Protocol definitions

Checklist (continued)

- FS Should Not Include
  - Internal names (elementContextId)
  - Implementation description (…application will be packaged as a method in a dll)
  - Screen design, type of control (…checkbox indicates that this holiday recurs every year)
  - Open ended lists (..including but not limited to…)

Functional Specification Review Log

- Maintains a list of questions about the functional specification
- Developer answers initially recorded here
- Developers use the log to update the functional specification
- Don’t forget to inform the team and get buy-in
- Items that have been updated in the FS are marked closed

Functional Specification Review Log

- Must contain
  - Questions
  - Answers
  - Indication of closure
- Could be maintained in
  - Bug tracking software
  - Excel
  - Word Tables

Collaboration Agreement

- Tester(s) seeking info from developer(s)
- Developers willing to share information
- Formal and informal meetings among them
- Agreement on who maintains the review log and updates the specification

Team Tutoring Sessions

- Developers hold periodic tutoring sessions to keep the team updated on design
  - Team members get more exposure to each other
  - Design concepts shared
- Review Log can be adjusted based on actual design

Other Tips and Tricks

- Have regularly scheduled team meetings
- Use a Projector
- Get all team leads involved in functional definition
  - Developers
  - Testers
  - Writers
  - Field Support
  - Marketing
- Talk face to face whenever possible

Challenges

- Testers must be willing to move ahead
- Developers must be willing to be accessible to testers
- Team needs to have a commitment to feed information back into the functional spec

A Case Study

- Developers and testers participated
- Checklist created by testers
- Review log maintained by testers
- Review log was a Word document
- Functional specification updated by developers

Successes

- More complete FS
  - As time goes by, the FS approaches the utopian vision
- Good Test documents
  - Tests based on reality, tester more familiar with actual functionality and design
- Better product
  - Better design as a result of better docs
  - Tested better

What We Would Change Next Time

- Involve writers and designers in creating the checklist
- Involve writers and designers in entering questions in the log

Other Options

- Consider other ways of logging open items
  - Bug tracking tool
  - Excel
  - Sharepoint
  - Requirements management tool
- Consider alternate division of labor
  - Testers could help update functional spec if developers too busy

Conclusion

- Be realistic
- Be proactive
- Be open to collaboration
- Maintaining a living document is not easy without a commitment