Testers Have Requirements Too!

Moira Tuffs
PNSQC 2021
Traditional Requirements Gathering

• Input from Customers for new technology
• Input from Sales team for competitive advantage
• Input from Applications team for techniques and features
• High profile bug fixes
Leads to

• Untestable Requirements
• No success criteria
• Lack of baseline metrics
• Very broad requests
Causes Problems for the QA team

• Guestimate for baseline metrics
• Inconsistent understanding for validating a requirement
• Evidence of testing is incomplete
The Devil is in the details

Requirement to show completed job details on the UI

• What details?
  • Job name
  • Job start time
    • Include printer preparation time?
  • Job end time
    • Include cool down or drain time?
• Material used
• Where to show the details?
  • Queue, Build report, Status screen?
Source of data: Log Files

- Searching log files for material usage/job data
- Evidence is hidden but sometimes there
- Tools need to be written (by the QA team)
  - Log files change on a release-to-release basis
Log Files – Job data

- Job name
- Material used
- Total layers printed
- Success
Log Files – Timing data

- Job starting
  - Thermally ready
  - Material checks
  - Surface smooth
- Job started
  - Support layers
  - Part layers
- Job complete
  - Move to drain
  - Cool down
- Time the job was removed
Log Files – Testing Requirements

- At the start of the job report
  - Job name
  - Initial start time (when the user presses play)
- Throughout the job report
  - Layer printed
- At the end of the job report
  - Job name
  - Complete/Abort
  - Duration of the job
  - Amount of material used
  - Total layers printed
Requirement 20% quicker

• 20% quicker than what?
• 20% on average (how many to average)
• 20% on every job
• 20% on all materials
• 20% on all dimensions of jobs (short, full plate, tall, intricate)
Baseline Metrics Requirements

• Specify
  • Set of jobs
  • Material set
  • Baseline metric

Details will allow the QA team to measure and report speed improvements.
Convenience of testing

- Efficient testing
  - Shorten a count down timer
  - Enable manipulation of printer states
  - Set material levels
  - Cover the laser
Appropriate Hardware and Timelines

- Simulators, Emulators
- Toasters, Mules, Testbeds
- Complete system
- Schedule dependencies
When

- MRD review – not sure what that baseline is
- Too late – we’re deep into testing, don’t have time to make that a requirement now.
- Sprint planning – how can we test that story
- Typical process
  - MRD review
  - Engineering response
  - Detailed planning
MRD Review

• Look for the red flags
  • 30% faster
  • Simpler to use
  • More information on error codes
• Ask for early versions
• Review appropriate sections with QA team.
Engineering response

- Define the baseline to measure 30% from.
  - Standard set of customer-like builds
  - Manufacturing build
- Ask for customer usability metrics.
- Have a workshop on the error codes – which ones are bad/poor.
- Simulation/Emulation capabilities
Detailed Planning

- What’s the success criteria on a story
- What evidence of testing will be needed
  - Log files
  - Convenience/testing tools
  - Hardware
- Schedule dependencies
  - Hardware needs updating which needs s/w changes which need validation
Where to document QA requirements

- JIRA stories as acceptance criteria
- Design details
- Improved or updated MRD
- Addendum to the MRD
Who can identify QA requirements

- Everyone on the QA team
- QA manager
- Software and Technical Program managers
- Development team
Summary

Testers Have Requirements Too!
Summary

• Testers HAVE requirements.
• Dig into requirements at all levels
• Document the testing requirements early on.
• Scour the user stories
• Work with developers
Q&A