

Refocusing Testing Strategy Within the Context of Product Maturity

Anna Royzman, Liquidnet

Mile High Agile, Denver 2012

How "agile" you are doesn't matter.
What matters is satisfying
customers, stakeholders,
employees.

Esther Derby

“QUALITY” – what’s in the
word?

It depends on who and when you ask

Group Exercise #1:

Identify all groups of stakeholders and brainstorm what “quality” means to each group

Group Exercise #1 (cont.). The results *might* look like that:

Stakeholders	What “Quality” means to this group
Product Owner	Delivery
Customers	Product satisfies their needs; user-friendliness (usability, intuitiveness, sense of control), stability (whatever worked before doesn’t break now)
Customer Support	Easy troubleshooting, low maintenance
Business	Business value, ROI
Operations	Scalability, system stability
Sales	Implemented features which they promised to customers are functioning well
C-level management	Revenue, reputation
IT management	Cost
Security	Data protection, anti-hacking checks
Team	Minimizing tech debt, good processes
Deployment	Error-prone and fast deployment

Suggestion: come up with your own list as a team

Risks at Different Levels of Project and Product Maturity

Project Risks

Quality Risks

Product Maturity Stages

- *Aides to initiate, fund and sell the product:*
 - Concept (*often in the form of PPT presentation*)
 - Proof Of Concept (*working prototype*)
 - Sales Demo version (*limited version of the software to demonstrate product features to a customer*)
- *Production:*
 - Pilot Product/Feature (*exposed to a limited number of 'early adopters'*)
 - New Product/Feature
 - Mature Product/Feature

Quality Risks

Bug impact is always contextual

It's important to know the business vision, the sales pitch, how the product is being supported, and the user expectations in order to have a better judgment on the quality risks

Group Exercise #2. At what stage do you need to test that the following works:

	POC	Live Demo	Pilot	New Product	Mature Product
User can login with username/password					
User cannot login with invalid username or password					
Error message on invalid input					
Field boundaries for new password					
Error message for field boundaries					
Authentication security measures (firewall rules)					
All combinations of valid/invalid /blank cases for username /password					
Error message when a user is trying to register with existing username					
Auditing and reports of who and when logged in					
Simultaneous login of 500 users					
Password expiration after 90 days					
<i>"Who Cares" (stakeholder group which gets affected if bug is found)</i>					

Group Exercise #2 (cont.) The results *might* look like that:

	POC	Live Demo	Pilot	New Product	Mature Product
User can login with username/password		x	x	x	x
User cannot login with invalid username or password			x	x	x
Error message on invalid input				x	x
Field boundaries for new password			x	x	x
Error message for field boundaries				x	x
Authentication security measures (firewall rules)				x	x
All combinations of valid/invalid/blank cases for username/password				x	x
Error message when a user is trying to register with existing username				x	x
Auditing and reports of who and when logged in			x	x	x
Simultaneous login of 500 users				x	x
Password expiration after 90 days					x
"Who Cares"					

Everyone on the Team is a Quality Advocate

Mail from Craig, our developer:

Anna this is now implemented, but there are some known issues.

Please test the new grid widget thoroughly, and help nail down very specific steps to reproduce the issues. Most of the issues center around scrolling and redrawing.

Here are some areas that I believe may have issues (please verify as I am not certain)

- -if group-by is changed and the tree grid is refreshed, the scrolling is not reset properly and it is possible to scroll past the end of the table
- if you scroll midway down the treegrid and then close/expand some branches scrolling is not reset so it is possible to scroll past the end of the table
- if you resize the browser the treegrid does not automatically resize to take up all available space. The columns stay the same size. But if you slightly adjust things for example close the filter panel or move it slightly suddenly the treegrid does adjust.

*There may be more. We need to find as many display issues as we can they are not serious but they may have a psychological effect making the product seem unpolished.
thanks!*

Everyone on the Team is a Quality Advocate

Disconnect between specialists of different disciplines in the team happen when the team members don't have clear understanding of stakeholders or users' Quality needs. Then it is common for QA to be 'gatekeepers', or various other bad things happen which make everyone's work less enjoyable.

When everyone on the team understands the quality requirements from each stakeholder group, then everyone becomes a quality advocate. Now the whole team has one goal and speaks the same language.

Selecting Testing Methods

Within the Context

The World of Automation

Testing Tools/Frameworks (custom and “off the shelf”):

- Unit testing frameworks
- Acceptance testing tools/frameworks (incl. GUI validation)
- Performance testing tools
- Simulators, log helpers and other automated support tools

Testing Focus:

- Build’s sanity check (also automation for demos and deployments)
- Certification (3rd party validation, APIs testing between two internal systems)
- Regression testing
- Load and stress testing

Automation

- What is it
 - Usage of tools that assist with testing or checking tasks
- Benefits
 - Great for building a 'safety net' of business-critical workflows and logic that must work every time
 - Sanity checks and unit tests
 - Assisting humans with humongous or repeatable tasks (stress or load testing, data testing, 3rd party certification etc.)
- Limitations
 - Maintenance (technical debt) – like any other code
 - Bugs – like any other code
 - High liability – when too many tests are written (have to trim your automation suite often)
 - Cannot test usability
 - Upfront cost

Exploratory Testing

- What is it
 - A skilled testing activity which is described as simultaneous learning, test design and test execution. It requires thinking and asking questions.
- Benefits
 - ET seeks to find how the software actually works, not only that the 'requirements are being met'
 - It's often the only way to discover usability risks
 - It's a good method for testing business assumptions
- Limitations
 - It's manual
 - It's a skill
 - It's time-consuming activity – may not be able to cover all you need within timeframes

Group Test

- What is it
 - A group testing activity, where whole team (and sometimes stakeholders and customer support) test various scenarios at the same time on one production-like environment
- Benefits
 - Great for team building
 - Everyone is getting exposure to the whole product
 - Rapid feedback on quality of the software
 - More eyes and hands on the UI
 - Whole system integration testing
- Limitations
 - It's not skilled testing (the bugs can be missed)
 - It must be coordinated
 - You cannot do it all the time; people see it as a distraction

User Test Drive

- What is it
 - End users' direct or indirect input into product/feature usefulness and intuitiveness through hands-on experience with the software
- Benefits
 - A great way to understand how the product or feature will be used in the field
 - Prioritizing or refocusing testing coverage through discovering what's important to users
 - Establishing rapport and continuous feedback loop
- Limitations
 - It's personal
 - Usually not looking for an 'edge case' (but you will hear it later LOUD and CLEAR 😊)

Select Testing Scope and Strategy based on the context:

- Product maturity stage
- Tolerance to failures (understanding stakeholder needs)
- User feedback
- Risks (both project and quality)
- Delivery time
- Testing assets (people and tools)

My testing toolkit:

- Automation (*using variety of different tools*)
 - Production “safety net” – business workflows which **must** work all the time
 - Complex logic algorithms (usually through the unit tests, at the component level)
 - Cross-OS/browser testing
 - Load testing
 - Demo laptop: start up and check
 - Build sanity check
- Exploratory testing
 - Story and enhancements testing
 - Usability testing
 - Validating business assumptions
- Group test
 - Rapid feedback on massive changes to the software
 - Infrastructure or “look and feel” changes
- User test drive
 - New features
 - Enhancements
 - Usability

Reach out to me if you have questions, want to discuss the content or give a feedback:

ari16a @ gmail.com

On twitter: @QA_nna

Good luck!

Anna.

Appendix A: How to run a successful Group Test

Instructions for the Group Test organizer

Put an hour for the group test on team's calendar.

Deploy the build beforehand, and smoke test it. This way you will not waste time of many people later. If the build is no good, reschedule the group test.

Compile the list of scenarios to cover what you need, and split them into areas depending on the number of people participating. This way, you will ensure the “must-have” coverage, and avoid duplication of effort. Usually, after getting through assigned scenarios, people start exploring around, and discover other things.

Prepare the logins for everyone.

Prepare the instructions for recorder tool. Some bugs may not be reproducible in one click, and it could take a significant amount of time to discover the exact replication steps. Using the tool will provide a record for development. I suggest going to everyone's desk and make sure the tool is working. Checking it during group test will take time away from testing.

Right before the group test, send out the email to the group which contain:

- * The login instructions (URL or Installer location, usernames, passwords)
- * List of assignments for each participant
- * A reminder to turn on recorder tool at the login page

Once group test ends, announce it and request a list of issues from every participant with as much information as they could provide (login ID, time when incident happened, replication steps, etc.). Go around and ask everyone personally if you don't get a written response.

Give the list of issues to Product Owner and Tech Leads for prioritizing.

Repeat next day or as see fit.

Appendix B: few hints for testing at each Product Maturity Stage

Stage	Testing
Concept (usually a doc or presentation)	Review for the questionable technological claims or delivery assumptions.
Proof of Concept (working prototype)	Minimal testing (depending on the context) to assure that the demo will work; set expectations upfront.
Sales Demo	Get the list of scenarios which Sales will demo and test only these scenarios. Set expectations with Sales upfront.
Pilot Product/Feature	The “quality threshold” is lower for this stage, and the number of users is usually limited. Don’t invest too much time and effort into “complete” testing coverage. Manual testing often suffice. Include customer support in testing.
New Product/Feature	Expect that new enhancements/tweaks will be requested soon. New feature is rarely a final version. Design your automated tests suite with this notion in mind. Test for usability. Include customer support in testing.
Mature Product/Feature	Reputation of product/service quality becomes more significant. No one likes “surprises”: when the previously working business-critical feature stops working, it affects the users’ perceptions of product’s reliability; such perceptions often affect the business as well. Create and maintain a robust “safety net” automation regression suite to include all business-critical scenarios; some tests (not all) should be checking UI layer as well.