Many Agile Teams “only” Focus on Functional Requirements

Functional:
- How do I …?
- Validate user stories work as advertised
  - “As a reviewer I want to add a note to a chart”
  - Compute the charge for an invoice
- Validate boundary conditions
  - Can I add more than one note at the same place?
  - Are excess charges computed correctly?
Non-functional Requirements

Accessibility  Reliability
Compatibility  Safety
Efficiency    Scalability
Effectiveness Security
Extensibility Stability
Maintainability Supportability
Performance  Usability

Other terms for non-functional requirements: "constraints", "quality attributes", and "quality of service requirements"

Qualities are usually described by "ilities" as seen in non-functional requirements…but quality can also focus on how well functional requirements are met.

BECOMING AGILE AT QUALITY

“Quality is not an act, it is a habit…”
—Aristotle
Patterns for Being Agile at Quality

Core Patterns
- Breaking Down Barriers
- Integrate Quality

Becoming Agile at Quality
- Whole Team
- Quality Focused Sprints
- Product Quality Champion
- Agile Quality Specialist
- Monitoring Qualities
- Agile QA Tester
- Spread the Quality Workload
- Shadow the Quality Expert
- Pair with a Quality Advocate

Identifying Qualities
- Finding the Qualities
- Agile Quality Scenarios
- Quality Stories
- Measureable System Qualities
- Fold-out Qualities
- Agile Landing Zone
- Recalibrate the Landing Zone
- Agree on Quality Targets

Making Qualities Visible
- System Quality Dashboard
- System Quality Radiator
- Qualify the Roadmap
- Qualify the Backlog
- Quality Checklists

Tearing Down the Walls aka “Breaking Down Barriers”

Physical Barriers, Cultural Differences
Language/Communication, Background Expertise, Lack of Time, Us and Them Mentality

- How can agile teams remove the barriers and become more agile at quality?

- Tear down the walls through various actions: include QA early on; make them part of the sprints, embed them in the teams
Agile Teams

Cross Functional
Good Communication
Focus on Stakeholders Needs
Incrementally deliver working software
Adapt to Change as needed
Collaborative and Self Organizing
Whole Team working together

Quality Assurance Teams

Understands testing well and knows how to specify and validate system qualities
Certify the functionality of the application based upon the contract and requirements

Many QA groups work independently from the software team
Usually involved late in the process and not a lot of communication with team
Usually not part of the Agile team…
Agile Quality Teams

“Whole Team” Architects and QA work closely with the team integrated during the day-to-day development. Engage in Architecture & QA activities much earlier. Works with whole-team including product owner on understanding and defining qualities. Assists teams with definition & validation of important quality requirements. Proactive working closely & coordinates between business, management and developers.

Architecture Roles and Activities

**Traditional Architects**
- Independent from development
- Keepers of the overall vision of the architecture
- Enforcers who...
  - Certify compliance with corporate architecture standards
  - Get involved on an “as needed” basis in the software lifecycle

**Agile Architects**
- More integrated with day to day development
- Stewards for ongoing sustainable development
- Mitigate architecture risks
- Work with business, product owner, QA and devs to define and improve the evolving architecture
- Establish good practices and pay attention to details
**QA Roles and Activities**

**Traditional QA**
Independent group
Gatekeepers who…
Understand testing well and know how to specify and validate system qualities
Certify app functionality based upon contracts and requirements (SLAs …)
…get involved late in the software lifecycle

**Agile QA**
Integrated with day to day development
Proactive, engage in QA activities much earlier
Work closely with business, product owner, architects and devs to understand, define, and validate quality requirements

---

**Embedding QA with Team**
aka “Pair with a Quality Advocate”

- Great experience report at Agile 2014
- [AgileAlliance.org](http://AgileAlliance.org)

Experience Report posted:
Tearing Down the Walls: Embedding QA in a TDD/Pairing and Agile Environment by Stephanie Savoia
Build quality into your project rhythms

AGILE QUALITY

Focus on Quality

- QA can gather / organize quality scenarios in collaboration with the development team
- Identify architecture risks and incorporate them into the architecture to incorporate important qualities
- Additional quality scenarios can be gleaned from Service Level Agreements (SLAs)
- Include relevant tests as part of each sprint
- Test important qualities early
- Automate “easy” quality tests
Some decisions are too important to leave until The Last Responsible Moment

**SO**

**CHOOSE THE MOST RESPONSIBLE MOMENT**

---

Find Essential Qualities

System qualities are often overlooked or simplified until late in the development process causing delays and extensive refactoring and rework

- How can agile teams understand and prioritize essential qualities for an evolving system?

- Have team meetings at opportune times throughout the development process with stakeholders to brainstorm the important qualities to be considered for the system
Quality Checklists

What are the expectations for system quality? Different for new functionality but probably known for “standard” functionality that is familiar!

- It is hard to ensure quality is being met if you do not know the expectations
- Develop a checklist that includes expectations for desired system qualities that are common across the system and should be consistently met. This checklist can be reviewed by the team to ensure that qualities are met before features are released and verified by the team as part of quality assurance.

Quality Scenarios, Quality Stories, and Fold-Out Qualities

“Users initiate 1,000 order transactions per minute under normal operations; transactions are processed with an average latency of 2 seconds.”
Finding Even More Qualities: Fold-out Qualities

- Usability: Can I cancel my order? When?
- Security: Does the system retain credit information? If so, can I control how that information is retained?
- Security: Is credit information securely transmitted?
- Security: Is it protected from unauthorized access?
- Performance: How fast can I place an order and receive confirmation? When there are lots of users?
- Availability: What happens if the credit card service is unavailable?
- …

Quality-related acceptance criteria that can be attached to specific functional user stories

Quality Focused Sprints

Features don’t make a viable system; rather a viable system is accomplished by focusing on features accompanied by paying attention to system qualities

- How can you incorporate these other non-functional requirements into your system as needed?
- Therefore, take time to focus on your software’s non-functional qualities and devote a sprint to measuring and improving one or more of your system’s qualities
Make Qualities Visible

- Include quality scenarios for dev & testing in your backlog
- Maintain a separate quality scenario backlog
- Include quality and functional acceptance tests as acceptance criteria for releases
- Identify Architecture Tasks
- Part of the Roadmap
- Quality Radiators

Monitor Qualities—Build An Operational Dashboard
Patterns for Being Agile at Quality

Core Patterns
- Breaking Down Barriers
- Integrate Quality

Becoming Agile at Quality
- Whole Team
- Quality Focused Sprints
- Product Quality Champion
- Agile Quality Specialist
- Monitoring Qualities
- Agile QA Tester
- Spread the Quality Workload
- Shadow the Quality Expert
- Pair with a Quality Advocate

Identifying Qualities
- Finding the Qualities
- Agile Quality Scenarios
- Quality Stories
- Measureable System Qualities
- Fold-out Qualities
- Agile Landing Zone
- Recalibrate the Landing Zone
- Agree on Quality Targets

Making Qualities Visible
- System Quality Dashboard
- System Quality Radiator
- Qualify the Roadmap
- Qualify the Backlog
- Quality Checklists

Candidate Quality Patterns
- Exploit Your Strengths
- Value Quality
- Everyone has QA responsibilities
- Grow the Team
- Architecture Runway
- Quality Debt related to Technical Debt
- Define Quality Acceptance Criteria

- Managing Quality Debt
- Getting the Agile Mindset
- Experiment to Learn
- Responsible Moments
- Continuous Inspection
- Quality Risk Assessment
- Quality Tests
- Automate First
- Share the Quality Load
Interactive Session

Whole Team  Agile Landing Zones  Quality Scenarios  Quality Radiator

LET’S SEE HOW THESE PATTERNS RELATE …

Thanks … 谢谢 (Xièxiè)

rebecca@wirfs-brock.com  Twitter: @rebeccawb
joe@refactory.com  Twitter: @metayoda