Being Agile about Qualities
“Values, Practices & Patterns”

Joseph W. Yoder
Teams That Innovate
The Refactory, Inc.
Twitter: @metayoda
joe@refactory.com
http://www.refactory.com
http://www.teamsthatinnovate.com

Copyright 2016 Joseph Yoder & The Refactory, Inc.

Core Ideas / Takeaways

• Patterns and practices
• Values drive practice
• Quality-related activities
• Roles QA and architects play
• Call To Action (steps you can take)
Agile & Quality

Agile Practices

Mob Programming
A Whole Team Approach

TDD

(Re)Define Acceptance Criteria
Write some production code
(Rewrite) a test
Check if test fails
Clean up code (Refactor)
1 or more tests fail
Test fails
Check all tests succeed

Ship it!!!
Ready to release

Scrum

Good Tests!!!

2/3
Kanban (看板)
Signboard / Billboard

The basic principles of Kanban
- Limit Work in Process (WIP)
- Pull value through (WIP)
- Make progress visible
- Increase throughput
- Fixed Kanban Backlog
- Quality is part of the processed (internal)

Continuously monitor the above to improve!!!
Is this similar to a Retrospective?

Lean Development

Increase Value, \textbf{Reduce Waste (Muda)},
Improve Flow, Quality, …

\textbf{Understands} customer \textit{value} and
focus continuously to increase it

Provide \textit{perfect value} to customer and business

\textbf{Just in Time} Practice

\textbf{Learn} and \textbf{Improve}…
Agile == Lean?

Early Agilest were influenced by Lean, but:
- Many get stuck in the process
- Many Misconceptions about Agile

Agile/Lean Design Values

- Core values:
  - Design Simplicity
  - Quick Feedback
  - Communication
  - Continuous Improvement
  - Teamwork/Trust
  - Satisfying stakeholder needs
  - Building Quality Software

- Keep Learning
- Lots of Testing!!!
Continuous Improvement
“Retrospectives are Key!!!”

Small Steps we can take - next sprint!!!

architecture quality can be invisible
...especially when the spotlight is on

FEATURES

The Problem

© Can Stock Photo Inc. / alex5248
The Solution

What’s below the waterline?

all those “ilities” we can’t ignore ...

© Can Stock Photo Inc. / Freezingpicture
Complex vs Complicated Systems
(Cynefin Framework)

Complex
- Enabling constraints
- Loosely coupled
- probe-sense-respond
- Emergent Practice

Complicated
- Governing constraints
- Tightly coupled
- sense-analyse-respond
- Good Practice

*Cynefin as of 1st June 2014* by Snowded - Own work. Licensed under CC BY-SA 3.0 via Commons - https://commons.wikimedia.org/wiki/File:Cynefin_as_of_1st_June_2014.png#/media/File:Cynefin_as_of_1st_June_2014.png
Values Drive Practices
What makes a practice a pattern?

- Repeatable
- Useful (solves problems)
- Positive consequences
- Potentially negative consequences, too
  - awareness / attention can reduce or mitigate

courtesy Jordan Wirfs-Brock
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
- 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
- Automate First
- 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 ...

Agile Quality Teams
“Whole Team”

- Architects and QA work closely with the product or program teams
- Whole team works at understanding, defining, delivering, and verifying system qualities
Some decisions and actions are too important to leave until The Last Responsible Moment

so

**CHOOSE THE MOST RESPONSIBLE MOMENT**

How do you

**FIND RESPONSIBLE MOMENTS?**
Qualify the Roadmap

“All you need is the plan, the roadmap, and the courage to press on to your destination”
— Earl Nightingale

DELIVERY
- New mobile opportunity
- Oct 2017: New mobile opportunity

BUDGET
- Aug 2017: New mobile opportunity

RESOURCE
- Mobile search
- Android

ARCHITECTURE
- Performance
- Migration
- Security

DEPENDENCIES
- Performance
- Migration
- Security

RISKS
- High risk
- Normal

ISSUES
- TBD
- Low risk

ON RADAR
- TBD
- Low risk

YEAR 2017
- Jan
- Feb
- Mar
- Apr
- May
- Jun
- Jul
- Aug
- Sep
- Oct
- Nov
- Dec

YEAR 2018
- Jan
- Feb
- Mar

MOBILE WEB
- v1
- v2

PC PLATFORM
- v1
- v2

MOBILE RESEARCH
- Android
- iOS

RICH MOBILE WEB APPS

ONGOING RELEASES

RESPONSIVE DESIGN

PERSISTENCE FRAMEWORK

LOAD BALANCING

MOBILE SECURITY

PLATFORM STABILITY

CLOUD RESEARCH

MICROSERVICES

NO SQL / BIG DATA

Ongoing development activities
Qualify the Backlog

You can add backlog items for quality scenarios, system quality-related architecture work... yes, you can

Make Architecture Work Visible and Explicit

Visible

Positive Value
Visible Feature

Negative Value
Visible Defect

Invisible

Invisible Architectural Feature

Technical Debt

Color your backlog—Phillipe Kruchten

http://philippe.kruchten.com/2013/12/11/the-missing-value-of-software-architecture/
Fold-out Qualities

Quality-related acceptance criteria attached to user stories

Security: Use 256 bit SSL encryption….
Security: Is credit information retained? Do I have control over this?
Performance: How fast can I place an order and receive confirmation?
Performance: Order time < 2 seconds
Usability: Can I cancel my order? When?

“As a customer I want to place an order using my credit card….”

“Acceptable means done with quality”

HOW SYSTEM QUALITY WORK CAN FIT INTO YOUR RHYTHMS
Build architectural quality into your project rhythms

“QUALITY IS NOT AN ACT, IT IS A HABIT.”
—ARISTOTLE

How Quality Fits Into An Agile Process
Define Architecture Triggers

• Conditions that cause architecture investigation/ tasks
  – Quality target no longer met
  – Code quality metrics violations
  – ...
• Have broad system impact

Architecture Spikes & Explorations

• Answer deep questions / offers potential architecture solutions
• Not as tactical as an XP Design Spike
• Visible and bounded
ONGOING QUALITY ACTIVITIES

Monitor System Qualities—Build An Operational Dashboard
Incrementally Test Key Components’ Performance

- Identify key pathways and critical components
- Test components as they arrive to access performance
- Use mocks, stubs, and auto-responders to simulate missing components

Test Infrastructure To Verify Architecture Assumptions

- Benchmark early, then track
- Example:
  - Push/pull response times
  - Msg creation rates with >1 publisher
  - Consumption rates
  - Effects of adding msg dispatchers
Testing Overall System Qualities

- Some are “easy” and can be part of a frequently run automated quality test suite
- Some require “extensive” setup
- Some require near-production environments
  - Load and performance tests
  - Complex quality scenarios involving interactions with several systems/services
PAUSE POINTS HELP
EVALUATE RISK

Quality Focused Checklists

• Release Checklists*
  – Agreed upon checklist for quality and major architecture concerns

• Use at pause points
  – sprint planning, release planning, ...

*Thanks, James Thorpe for sharing your company's checklist

---

Development Release Checklist

The code and architecture should be examined prior to release into our test environment. If any checklist cannot be checked, exceptions should be noted and communicated to the Product Owner and QA lead.

**Code quality**

- All code complies with the relevant coding standard.
- All code complies without any errors or warnings (full clean build)
- Appropriate logging has been implemented throughout the code.
- All possible exceptions have been handled appropriately.
- The code has been checked for memory leaks.
- All test and debug code has been removed.
- Code is appropriately documented.
- All dead code has been removed.
- All unit tests have been run without error.
- Unit tests have been written for all new code or code changes.

**Architecture**

- No new service APIs have been created or modified without full documentation and architectural sign-off.
- No new service data structures have been created or modified without full documentation and architectural sign-off.
- No database structures have been created or modified without full documentation and architectural sign-off.

**Performance**

- All web pages render in under 500 ms with a production workload.
- All reports are generated in under 500 ms with a production workload.
- No query takes more than 500 ms to return data with production data volumes.

Notes or exceptions to the above:

____________________________________________________________________________________
____________________________________________________________________________________
Two Kinds of Checklists

1. Read-review
2. Do-confirm

Checklists at MozaicWorks*

*Thanks, Alex Balboaca for sharing
ROLES AND WHOLE TEAM DEDICATION

Who will lead? Who contributes?

- Big teams vs. small teams???

- Does system quality get the attention it needs?
How Product and Program Management and Architects Interact

Architecture Roles and Activities

Governance
- Planned Architecture
- Independent

Sustainable Development
- Incremental Architecture
- Integrated

Traditional
- Agile
Architecture Concerns

Risk-mitigation

Overall Vision
- Traditional

Good Practices
- Agile

QA Roles and Activities

Gatekeepers

Proactive

Validation & Verification

Sustainable Development

Independent

Integrated

Traditional

Agile
QA Focus

Spreading Quality Awareness

Verification

Validation

Traditional

Agile

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

Great experience report at Agile 2014

AgileAlliance.org

Experience Report posted:

Tearing Down the Walls: Embedding QA in a TDD/Pairing and Agile Environment by Stephanie Savoia
Shadow the Quality Expert
aka “Spread the Quality Expertise”

“Tell me and I forget, teach me and I remember, involve me and I learn” — Benjamin Franklin

As organizations grow, need to grow and evolve quality expertise …
Many organizations lack the resources fulfill their Quality needs …

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
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
Automate First
Quality Checklists
QA to AQ: Patterns about transitioning from Quality Assurance to Agile Quality

Joseph W. Yoder1, Rebecca Werb-Brach2, Adamar Aguilar3
1 The Resilient, Inc.,
2 Werb-Brach Associates, Inc.
3 HELP
joe@resilient.com, rebecca@wirfs-tech.com, adamar.aguilar@mrnp-ge.com

Abstract. As organizations transition from waterfall to agile processes, Quality Assurance (QA) activities and roles need to evolve. Traditionally, QA activities have occurred late in the process, after the software is fully functioning. As a consequence, QA departments have been “quality gatekeepers” rather than actively engaged in the ongoing development and delivery of quality software. Agile teams incrementally deliver working software. Incremental delivery provides an opportunity to engage in QA activities much earlier, ensuring that both functionality and important system qualities are addressed just in time, rather than too late. Agile teams embrace a “whole team” approach. Even though special skills may be required to perform certain development and Quality Assurance tasks, everyone on the team is focused on the delivery of quality software. This paper outlines 15 patterns for transitioning from a traditional QA practice to a more agile process. All of the patterns are completely presented that focus on where quality is addressed earlier in the process and QA plays a more integral role.

Continuous Inspection: A Pattern for Keeping your Code Healthy and Aligned to the Architecture, AsianPLoP 2014

Patterns to Develop and Evolve Architecture in an Agile Project, PLoP 2016

QA to AQ: Patterns about transitioning from Quality Assurance to Agile Quality, AsianPLoP 2014

QA to AQ Part Two: Shifting from Quality Assurance to Agile Quality, PLoP 2014

QA to AQ Part Three: Shifting from Quality Assurance to Agile Quality “Tearing Down the Walls”, SugarLoafPLoP 2014

QA to AQ Part Four: Shifting from Quality Assurance to Agile Quality “Prioritizing Qualities and Making them Visible”, PLoP 2015

QA to AQ Part Five: Being Agile At Quality "Growing Quality Awareness and Expertise", AsianPLoP 2016

QA to AQ Part Six: Shifting from Quality Assurance to Agile Quality "Enabling and Infusing Quality", To appear at PLoP 2016

...PATTERNS FOR TRANSITIONING FROM TRADITIONAL TO AGILE QA AND AGILE ARCHITECTURE

COPYRIGHT 2014 IS HELD BY THE AUTHOR(S). ACM 978-1-XXXX-XXXX-X.

OUR QUALITY VALUES

CALL TO ACTION

Daily Practices  Visibility  Sustainable Development

(C) by mulflinn on Flickr
Where do you start?

• Monitor qualities
• Pick some low hanging fruit
  – Make goals visible
  – Colorize your backlog
  – Create quality-related checklists
• Spread attention to system quality throughout teams
• Depends on where you are and where the pain is…

How Much Architecture Risk do you Have?

• New architecture, new product, new market, new technologies
• Transforming an existing product
• Evolving a product
• Feature extensions on a “stable” architecture

“the more risk, the more attention you need to pay to architecture”
How Big is your Project?
Small v. Large Projects

Small Projects
• 6-8 people
• Non-life critical
• Known domain

Large Projects
• Multiple teams
• Known domain but tackling a big problem
• “Naturally” emerging architecture can reflect organization structure
• Significant risks, challenges, unknowns, lots of coordination

Patterns and Practices

Quality Checklists
Quality Focused Sprints

Quality Dashboard
Quality Radiator
Automate First

Pair with Quality Advocate
Architectural Spike
Qualify the Roadmap

Architectural Explorations
Qualify the Backlog

Quality Specialists
Colorize Backlog

Quality Scenarios
Agile Landing Zone

Quality Stories
System Quality is a Journey

Commitment
Follow-through
Deliberate practice
Paying attention
Whole team engagement

Agile Mindset

Being vs Doing
Additional Resources

- The Hillside Group (patterns community): Hillside.net
- Being Agile at System Qualities workshop:
  - www.adaptiveobjectmodel.com/2015/04/qa-to-aq-shifting-towards-agile-quality
- Agile Myths: agilemyths.com
- The Refactory (www.refactory.com)
- Teams That Innovate (www.teamsthatinnovate.com)
- Pragmatic TDD:
  refactory.com/training/test-driven-development
  http://adaptiveobjectmodel.com/2012/01/what-is-pragmatic-tdd

joe@refactory.com
Twitter: @metayoda
www.joeyoder.com
www.refactory.com