Talking About Quality

Pacific Northwest Software Quality Conference 2018
Kathy Iberle
Iberle Consulting Group, Inc.
Great user stories ☺

+ 

Great user stories ☺

+ 

Great user stories ☺

→ Release
A Communication Gap

• We didn’t know what the customer expected
• We didn’t talk about how to meet those expectations  

OR
Talking About Quality: Stating Expectations

Test Coverage
Defect Counts

Security, performance, reliability, availability, usability, configurability, etc.

- Nonfunctional Requirements
- Quality Attributes

A Limited Language

A Rich Expressive Language

- Capabilities
- The “Ilities”
## Our Vocabulary: Quality Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accessibility</strong></td>
<td>How easily people with a range of physical abilities can use the system.</td>
</tr>
<tr>
<td><strong>Availability</strong></td>
<td>The extent to which the system’s services are available when and where they are needed.</td>
</tr>
<tr>
<td><strong>Installability</strong></td>
<td>How easy it is to correctly install, uninstall, and re-install the application.</td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td>How quickly and predictably the system responds to user inputs or other events</td>
</tr>
<tr>
<td><strong>Reliability</strong></td>
<td>How long the system will run before experiencing a failure</td>
</tr>
<tr>
<td><strong>Robustness</strong></td>
<td>How well the system responds to unexpected operating conditions</td>
</tr>
</tbody>
</table>
Some Dialects

FURPS: Functionality, Usability, Reliability, Performance, Supportability

ISO/IEC standard 205010 [ISO2010]
My Current Favorite Quality Attributes List

• Accessibility
• Availability
• Installability
• Manufacturability
• Integrity
• Correctness
• Performance
• Reliability
• Robustness
• Safety
• Security
• Usability
• Interoperability
• Efficiency
• Scalability
• Modifiability
• Portability
• Testability
Example: Anti-Lock Braking System (ABS)

- Accessibility
- Availability
- Installability
- Manufacturability
- Integrity
- Correctness
- Performance
- Reliability
- Robustness
- Safety
- Security
- Usability
- Interoperability
- Efficiency
- Scalability
Ask the Customer What Matters to Them

• What do you want?
  • ➔ Specific goals

• How will you know when you have it?
  • ➔ Acceptance tests

• What will that do for you?
  • ➔ Understand relative value of different attributes
Precise Language: Get Smart

SMART = Specific, Measurable, Achievable, Relevant, Testable

Performance target:
For <given scenario>,
the query returns results in no more than two seconds.
SMART Examples

• Scalability: The system is able to handle up to 20,000 simultaneous connections without degrading performance of <specific actions>, and can be scaled down to under 1000 connections.
• Reliability: Mean time between failures under <specific conditions> is X days
• Recoverability: System can recover from <type of failure> in less than 5 minutes
What Else Matters to Your Stakeholders?

Some internal quality attributes:

• Modifiability
• Testability
• Portability
• Interoperability with pre-existing internal systems

Defect: Failure to meet external quality attributes:

Technical Debt: Often a failure to meet internal quality attributes
A Communication Gap Inside the Team

• We didn’t know what the customer expected
• We didn’t talk about how to meet those expectations
Quality Process – A Vocabulary for Action

- **What** you will do
- **When** you will do it
- **Who** will do it
- **Why** you will do it
- **What** happens if results are not satisfactory

- Quality Gate
- Team Operating Agreement
- Acceptance Test
- Standard Practice
- Design Rules
- Definition of Done
- Release Criteria
- Test Strategy
- Coding Standard
- Spike
Waterfall Quality Gates: The Dreaded V-Model

- Requirements
  - Spec Reviews
  - High-level Design
  - Design Reviews
  - Low-Level Design
  - Design Reviews
  - Code

- TIME

- Acceptance testing
- System Testing
- Integration testing
- Functional Testing
- Unit testing
Danger #1: Not Enough Gates

- Requirements
- High-level Design
- Low-Level Design
- Code
- Functional Testing
- System Testing
- Acceptance testing
Danger #2: Excessively Large Gates

- Requirements
  - Spec Reviews
    - High-level Design
    - Design Reviews
    - Low-Level Design
    - Design Reviews
- Code
  - Unit testing
  - Functional Testing
  - Integration testing
  - System Testing
  - Acceptance testing
User Stories for Quality Attributes

1) As a user, I want the site to be available 99.999% of the time
2) As a user, I want to run your software on Windows 7 and Windows 10.
3) As the CTO, I want the system to use our existing orders database.

Do these user stories:
• Capture the user’s point of view?
• Split work into small, independent sections?
## Alternatives for Crosscutting User Stories

<table>
<thead>
<tr>
<th>User Story #1</th>
<th>User Story #2</th>
<th>User Story #3</th>
<th>User Story #4</th>
</tr>
</thead>
</table>

- Quality Attribute User Story

- Acceptance test
- Definition of Done
- Spike
- Standard Practice
- And more...
Agile Quality Gates

Agile quality gates can be stated as:

• Story acceptance criteria  OR
• Definition of Done  OR
• Release Criteria
Standard Practices

• Coding standards
• Design standards
• Version control procedures
• Etc.

Standard practices can be defined in:
• Team Operating Agreement
• Quality Plan
# Example: ABS Team Quality Attribute Review

<table>
<thead>
<tr>
<th>Quality Attribute</th>
<th>Our Plan For This Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correctness</td>
<td>Covered by existing user stories: 522-577.</td>
</tr>
<tr>
<td>Integrity</td>
<td>Definition of Done: All data logging stories must have acceptance criteria regarding data overflow.</td>
</tr>
<tr>
<td>Manufacturability</td>
<td>User story: As a manufacturing engineer, I can automate installation of the firmware onto the chip.</td>
</tr>
<tr>
<td>Performance</td>
<td>Definition of Done: All stories must pass a test which covers expected response time and allowable range for each category of event. We need to write this test.</td>
</tr>
</tbody>
</table>
Spike

Sometimes we don’t know how to achieve quality:
What does our customer mean by “usable”? 
How can we measure “good enough“?
Will this design choice give us fast enough performance?

Don’t be afraid to spike!

Create hypothesis
Run experiment in a timebox
Assess results
Decide what to do next
Talk About Quality to Bridge Those Communication Gaps

What do stakeholders expect?
• Quality attribute list
• The three questions
• SMART nonfunctional requirements

How will we meet the expectations?
• Quality gates: Acceptance criteria, Definition of done, release criteria
• Standard operating procedures
• Spikes
Q & A

Want to know more?
Read the paper or visit my website:  http://kiberle.com/