Elicitation of Quality Agile User Stories Using QFD

NDIA 20th Annual Systems Engineering Conference
“Agile in Systems Engineering“
10:15 – 10:40 AM
October 25, 2017

Sabrina J. Ussery, Shahryar Sarkani, Thomas Holzer

Dissertation Topic
Department of Engineering Management and Systems Engineering
School of Engineering and Applied Science

The George Washington University
1176 G Street NW
Washington, DC 20052
The lack of standard Requirements Engineering (RE) practices in Agile negatively impacts system quality, contributing to 24% of the causes for challenged or failed projects.

- The 2015 CHAOS Standish Group report indicates Agile projects are 3x more likely to succeed than Waterfall projects due to increased customer collaboration and customer satisfaction. [2]

- The Agile community claims that they do not really tackle requirements in a structured way, which may bring problems to the software organization responsible for software built following an Agile method. [1]

- Though more successful in some respects, the lack of stand RE practices in Agile contributes to 24% of the reasons for challenged or failed projects due to poor requirements quality (i.e., unclear or volatile). [2]
What is Agile?

incrementally

instead of all at once

Original plan

Actual plan

Cost of change curve

Traditional
Change resisted

Agile
Change welcome!

One-off activities

Continuous activities
**Requirements engineering** (RE) refers to the process of defining, documenting and maintaining requirements.\(^5\)
Agile RE: As Is

“Hall et al., reports that a large proportion (48%) of development problems stem from problems with the requirements.” [3]
“There are no documented RE activities which can be followed to obtain the user requirement in efficient manner ... The Agile manifesto and all the methodologies should have standardized and documented set of RE activities.” [3]

“The term ‘requirements engineering’ is avoided in the Agile community as it is often taken to imply heavy documentation with significant overhead.” [4]

“A lengthy requirements analysis phase is considered to hinder the speed of development.” [4]
Academic research compares Agile approaches to traditional RE activities and suggests areas of opportunity for improvement.

Table 1. Traditional and agile approach for requirements engineering (RE) activities

<table>
<thead>
<tr>
<th>RE activities</th>
<th>Traditional RE approach</th>
<th>Agile RE approach</th>
<th>Agile practices used to support the RE activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements elicitation</td>
<td>Discovering all the requirements upfront</td>
<td>Iterative: requirements evolve over time and are discovered throughout the development process.</td>
<td>Iterative RE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Face-to-face communication</td>
</tr>
<tr>
<td>Requirements analysis and negotiation</td>
<td>Focus on resolving conflicts</td>
<td>Focus on refining, changing and prioritizing requirements iteratively</td>
<td>Iterative RE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Face-to-face communication</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Constant planning</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Extreme prioritization</td>
</tr>
<tr>
<td>Requirements documentation</td>
<td>Formal documentation contains detailed requirements</td>
<td>No formal documentation</td>
<td>Face-to-face communication</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements validation</td>
<td>The consistency and completeness of requirements document</td>
<td>Focus on ascertaining whether the requirements reflect current user needs</td>
<td>Review meetings</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Face-to-face communication</td>
</tr>
</tbody>
</table>
Academic research surveys Agile approaches to traditional RE activities. Specifically, requirements documentation, stakeholder involvement, and requirements verification are called out as tractable opportunities for improvement.

<table>
<thead>
<tr>
<th>RE risk</th>
<th>Agile practice or challenge</th>
<th>Impact of practice or issue</th>
<th>Degree of impact in agile practice</th>
<th>Character of problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of requirements existence and stability</td>
<td>Face-to-face</td>
<td>Mitigates</td>
<td>Medium–High</td>
<td>Tractable</td>
</tr>
<tr>
<td></td>
<td>Iterative RE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constant planning</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issues with users’ ability and concurrence</td>
<td>Iterative RE</td>
<td>Mixed</td>
<td>High</td>
<td>Intractable</td>
</tr>
<tr>
<td></td>
<td>Customer access and participation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inadequate user-developer interaction</td>
<td>Iterative RE</td>
<td>Mixed</td>
<td>High</td>
<td>Tractable</td>
</tr>
<tr>
<td></td>
<td>Customer access and participation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overlooking a crucial requirement</td>
<td>Requirement prioritization</td>
<td>Mitigates</td>
<td>Medium–High</td>
<td>Tractable</td>
</tr>
<tr>
<td></td>
<td>Review meetings and tests</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modelling only functional requirements</td>
<td>Neglect of non-functional requirements</td>
<td>Exacerbates</td>
<td>Low</td>
<td>Intractable</td>
</tr>
</tbody>
</table>
These sentiments are shared with other researchers, who also note issues with requirements management. No written documentation results in information loss when code is implemented and refactoring costs skyrocket.

**Table 3.** Characterizing tractability of risks in agile requirements engineering (RE)

<table>
<thead>
<tr>
<th>RE risk</th>
<th>Agile practice or challenge</th>
<th>Impact of practice or issue</th>
<th>Degree of impact in agile practice</th>
<th>Character of problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of requirements existence and stability</td>
<td>Face-to-face</td>
<td>Mitigates</td>
<td>Medium–High</td>
<td>Tractable</td>
</tr>
<tr>
<td></td>
<td>Iterative RE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constant planning</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issues with users’ ability and concurrence</td>
<td>Iterative RE</td>
<td>Mixed</td>
<td>High</td>
<td>Intractable</td>
</tr>
<tr>
<td></td>
<td>Customer access and participation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>inadequate user–developer interaction</td>
<td>Iterative RE</td>
<td>Mixed</td>
<td>High</td>
<td>Tractable</td>
</tr>
<tr>
<td></td>
<td>Customer access and participation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overlooking a crucial requirement</td>
<td>Requirement prioritization</td>
<td>Mitigates</td>
<td>Medium–High</td>
<td>Tractable</td>
</tr>
<tr>
<td>Modelling only functional requirements</td>
<td>Review meetings and tests</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Neglect of non-functional requirements</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Exacerbates</td>
<td></td>
<td>Low</td>
<td>Intractable</td>
</tr>
</tbody>
</table>
“Stakeholder-appropriate **requirements** constitute critical determinants of **system quality**. Incorrect or missing requirements are supposed to lead to various problems in later phases such as effort and time overrun or an increased effort in acceptance testing.” [7]

<table>
<thead>
<tr>
<th>RE risk</th>
<th>Agile practice or challenge</th>
<th>Impact of practice or issue</th>
<th>Degree of impact in agile practice</th>
<th>Character of problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of requirements existence and stability</td>
<td>Face-to-face</td>
<td>Mitigates</td>
<td>Medium–High</td>
<td>Tractable</td>
</tr>
<tr>
<td></td>
<td>Iterative RE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constant planning</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issues with users’ ability and concurrence</td>
<td>Iterative RE</td>
<td>Mixed</td>
<td>High</td>
<td>Intractable</td>
</tr>
<tr>
<td></td>
<td>Customer access and participation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inadequate user–developer interaction</td>
<td>Iterative RE</td>
<td>Mixed</td>
<td>High</td>
<td>Tractable</td>
</tr>
<tr>
<td></td>
<td>Customer access and participation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overlooking a crucial requirement</td>
<td>Requirement prioritization</td>
<td>Mitigates</td>
<td>Medium–High</td>
<td>Tractable</td>
</tr>
<tr>
<td></td>
<td>Review meetings and tests</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modelling only functional requirements</td>
<td>Neglect of non-functional requirements</td>
<td>Exacerbates</td>
<td>Low</td>
<td>Intractable</td>
</tr>
</tbody>
</table>
User Story Issues

- Incompleteness (e.g., missing user story parts, business value, or acceptance criteria)
- Ambiguity
- Solution specific user stories
- Missing Non-functional requirements (NFRs)
- Inaccuracy
- Lack of bi-directional traceability leading to refactoring concerns
- Lack of integration with other RE techniques (use cases / user modeling)
- Lacking metadata for configuration management
- No automated support for user story generation [10 – 16]
Federal acquisition programs have begun to integrate aspects of Agile development into their strategy to leverage the benefits of Agile.

- Shorter time to market for innovative solutions, earlier manifestation of system benefits, minimization of rework, and better requirements management.

With strong leadership, a well-informed program office, and a cohesive and committed teams, Agile could enable the DoD (and similar organizations) to deliver innovative IT operational solutions faster and more effectively than traditional incremental approaches. [24]
With an Agile acquisition framework, the DoD can keep deliver capabilities faster and respond more effectively to changes in operations, technology, and budgets.

The MITRE Defense Acquisition Guide aims to adapt proven principles of Agile development specifically for DoD use and echoes the justification of the research proposed herein by reiterating the need for DoD Agile processes to support the following:

- Active user involvement in Agile Requirements Engineering activities
- Accurate, concise, testable and clear user stories
- Capturing of NFRs in users stories
- Managing user story dependencies
- Traceability of user stories to overarching mission threads
- Development of flexible requirements documentation for approval throughout the acquisition process
- Configuration Management of documentation as strategies or processes change.

“The US joint force will be smaller and leaner. But its great strength will be that it will be more agile, more flexible, ready to deploy quickly, innovative, and technologically advanced. That is the force for the future.”

- Secretary Panetta, Defense Security Review, 5 Jan 12
Call for Research

• Call for complementing Agile RE processes with traditional methods, to strike a balance between project agility and stability [18][22]

• Call for Agile RE processes and tools that [1][19]:
  - Are easy to use and not time consuming
  - Supports customer and team collaboration
  - Supports Requirements Elicitation in the user’s environment for distributed teams
  - Supports Requirements Management
  - Supports multi-dimensional prioritization
  - Supports automatic creation of user stories and related artifacts
  - Supports elicitation of NFRs
  - Support requirements storage and baselining for system reuse and refactoring
  - Automates verification of user stories to ensure quality before development
    - Are they complete?
    - Are they accurate?
    - Are they ambiguous?
    - Are they consistent?
    - Do they contain data for Configuration Management?
Provide a framework to elicit and manage quality user stories using QFD

- This study evaluates the positive benefits of utilizing Quality Function Deployment (QFD) to elicit, analyze, and manage Agile requirements.

- Prior to this research, RE practices are seen as being incompatible with Agile as they can be heavily reliant on documentation. [25]

- Requirements Engineering is one of the most challenging and important parts of Systems Engineering. The quality of system requirements highly impacts system quality and project health.

- QFD serves as a structured approach to defining and translating customer needs to produce products.
  - Combines quality control with value engineering to fully meet the customer’s expectations.

- This study will provide specific recommendations for use of QFD in Agile RE.
“A simple-but-powerful approach, coupled with a relatively inexpensive process, exists to bring the needed content, structure, organization, weighting and measurements to the decision-making process. Quality function deployment (QFD) is used in a growing number of product development organizations to provide assistance with the planning process. In the last 15 years, QFD has become a standard tool in requirements gathering, analysis and prioritization across all development organizations.” [23]

“Product [or system] planning begins with analyzing the performance of an existing product and improving or adding features. QFD can be instrumental in transforming products to meet continually changing customer needs and expectations.” [23]
For purposes of research, user story data sets (commercial and academic) to be deconstructed and recreated using QFD and quantitatively assessed for quality before and after model use. Inputs for quantitative metrics such as complexity assessments or prioritization will be uniformly randomized.
Proposed Model

Provide a framework to elicit and manage quality user stories using QFD
Proposed Model

1. Team member or customer is supported by model interface to elicit, analyze and specify requirements.

Inputs
- Team member or Customer
- Linguistics
- Problem Solving Techniques

Tool Interface
- Web-based QFD

Outputs
- User stories
- NFRs (constraints)

Traceability
- Priorities
- TRLs
- Market Analysis

Provide a framework to elicit and manage quality user stories using QFD
Provide a framework to elicit and manage quality user stories using QFD
Proposed Model

Inputs

Team member or Customer

Linguistics

Problem Solving Techniques

Tool Interface

Web-based QFD

Outputs

User stories

NFRs (constraints)

Traceability

Priorities

TRLs

Market Analysis

Tool interface will require simple inputs, using QFD as a framework, further taking into consideration linguistics for globally distributed teams as well as problem solving techniques.

Provide a framework to elicit and manage quality user stories using QFD
Proposed Model

Inputs

Team member or Customer

Tool Interface

Web-based QFD

Outputs

User stories

NFRs (constraints)

QFD components will be used to generate an initial set of user stories and NFRs, as well as their associated metadata: traceability to customer needs and solutions, priorities, Technology Readiness Levels (TRLs), and market analysis information per need.

Linguistics

Problem Solving Techniques

Traceability

Market Analysis

Priorities

Correlates

Solutions

Technology Readiness Levels (TRLs)

Provide a framework to elicit and manage quality user stories using QFD
Proposed Model

Inputs
- Team member or Customer
- Linguistics
- Problem Solving Techniques

Tool Interface

Web-based QFD

Outputs
- User stories
- NFRs (constraints)

Traceability
- Requirements will be “graded” by requirements quality tool. If requirements are not of the desired quality level, users can revise user stories within the QFD using the quality reports’ outputs as a guideline.

Market Analysis

Provide a framework to elicit and manage quality user stories using QFD
Research Definition

H1. If adapted, rule based requirements quality methods, like QFD, can provide a framework for Agile RE activities while remaining compliant with the Agile Manifesto.

H2. A number of Agile RE artifacts can be partially or fully automatically generated from the use of QFD to support process repeatability and artifact standardization.

H3. The use of a structured requirement quality method that supports distributed collaboration yields higher quality requirements than current methods.

Methods to create quality user stories

Q1. What challenges may inhibit the use of rule based requirements quality methods in Agile RE?

Q2. What Agile RE artifacts are supported by existing requirements quality methods?

Q3. Does the use of quality RE methods in Agile increase the quality of user stories over existing methods?

Automatic generation of documentation

Facilitation of distributed stakeholder involvement

Repeatable Agile RE process

Objectives
Summary

• Results of research may recommend new Agile guidance for requirements elicitation and management including the use of modified QFD as:
  o a web-collaborative, user story elicitation support tool
  o a basis for configuration and requirements management
  o a platform to identify TRLs and competitor capabilities to drive prioritization and other portfolio decisions
  o a means to assess risk and complexity of key features
  o a requirements specification generator

• Use of Natural Language Processing (NLP) quality tools as a means to verify quality of requirements generated by QFD prior to implementation. Consideration will be given to use more than one NLP tool and results will be compared in paper.

• Future research could use the same data to evaluate the feasibility of adapting other RE techniques for use in Agile.
References


Shahryar Sarkani, D.Sc., is Adjunct Professor in the Department of Engineering Management and Systems Engineering at George Washington University, Washington, D.C. He has over 20 years of experience in the field of software engineering focusing on architecture and design. Dr. Sarkani earned the D.Sc. in Systems Engineering from George Washington University, an M.S. in Mathematics from the University of New Orleans, and a B.S. in Electrical Engineering from Louisiana State University.

Thomas Holzer, D.Sc., has been Adjunct Professor of Engineering Management and Systems Engineering at George Washington University, Washington, D.C., since 1999. He is the former Director, Engineering Management Office, Enterprise Operations Directorate, National Geospatial-Intelligence Agency. He has over 35 years of experience in lifecycle systems engineering, leading large-scale information technology programs, and process improvement initiatives. Dr. Holzer was responsible for the strategic evolution of the National System for Geospatial-Intelligence technical and operational infrastructure architectures; assuring the integrity of the systems engineering performed; and development of a proficient systems engineering workforce. Dr. Holzer has D.Sc. and M.S. degrees in Engineering Management from George Washington University and a B.S. degree in Mechanical Engineering from the University of Cincinnati.

Sabrina Ussery has 10 years experience in industry working in systems engineering and program management. Sabrina has led many requirements engineering efforts for the FAA's NextGen Air Traffic Management (ATM) initiatives and spent 2 years as a Technical Product Owner in a Scaled Agile environment for a healthcare analytics organization. Sabrina is currently employed as a Senior Systems Engineer at Mosaic ATM Inc., leading systems engineering efforts for a number of FAA acquisition programs. Sabrina holds a B.S. in Applied Mathematics from Jacksonville State University and a M.S. in Aerospace Engineering from the Georgia Institute of Technology. She is currently a PhD candidate in Systems Engineering at The George Washington University.