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

Agile Requirements Engineering (RE)

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]

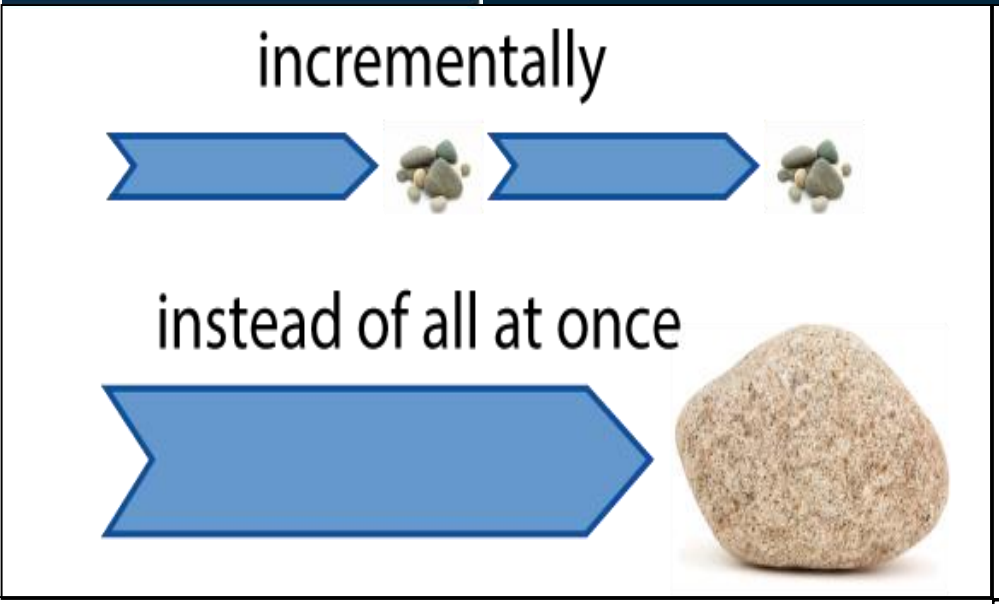
CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

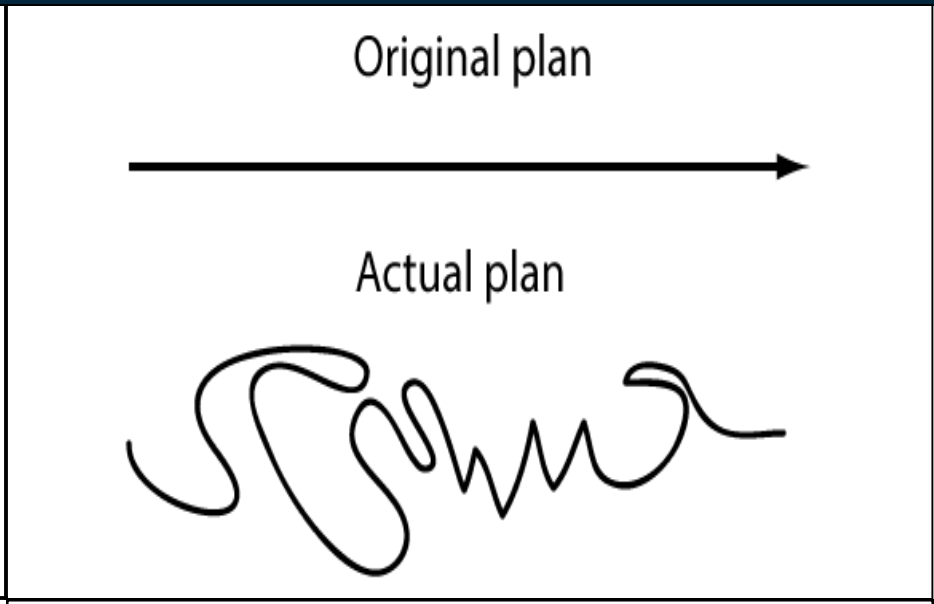
SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

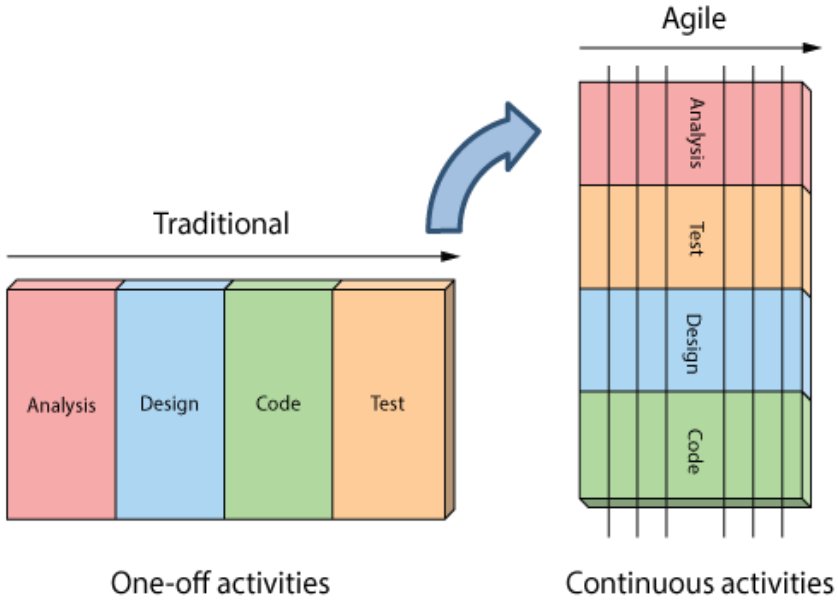
The resolution of all software projects from FY2011–2015 within the new CHAOS database, segmented by the agile process and waterfall method. The total number of software projects is over 10,000.

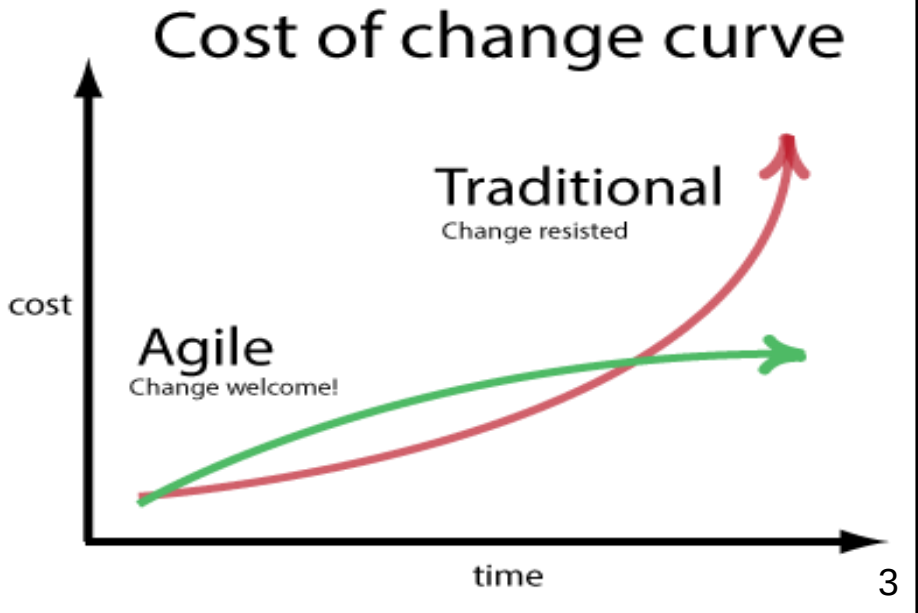
Image source: [2]

What is Agile?



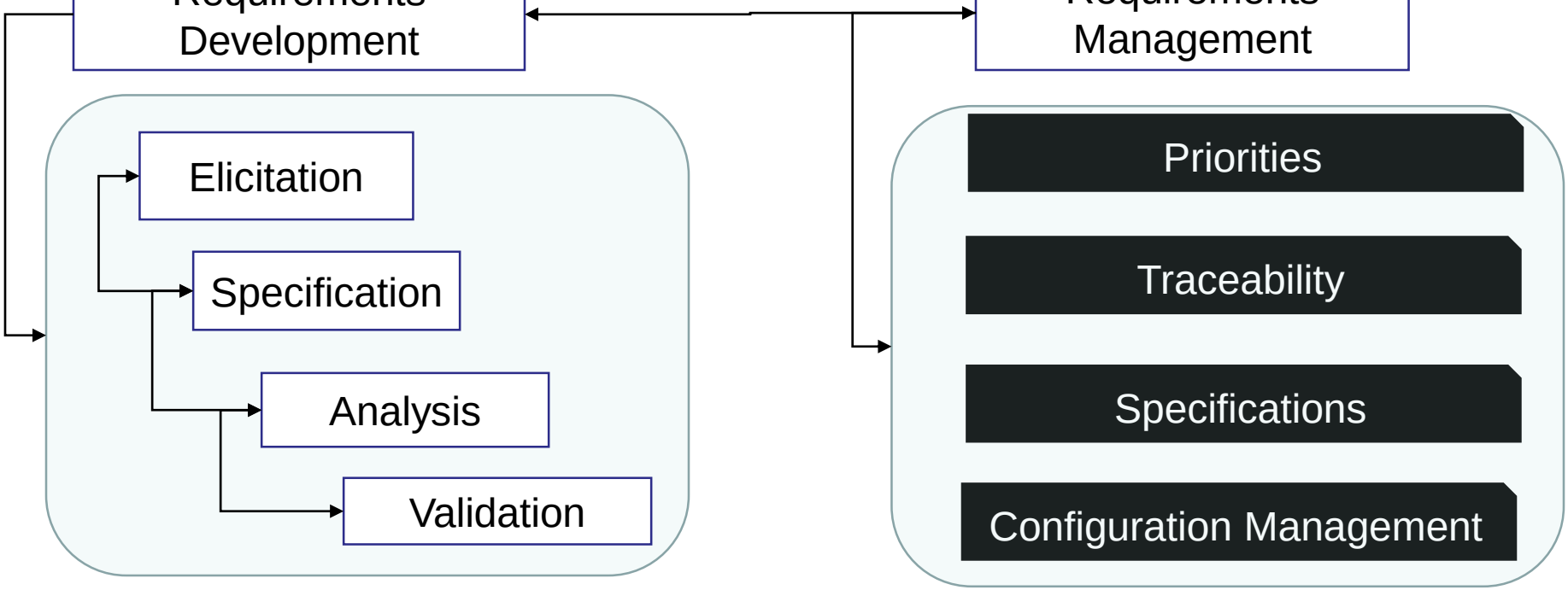






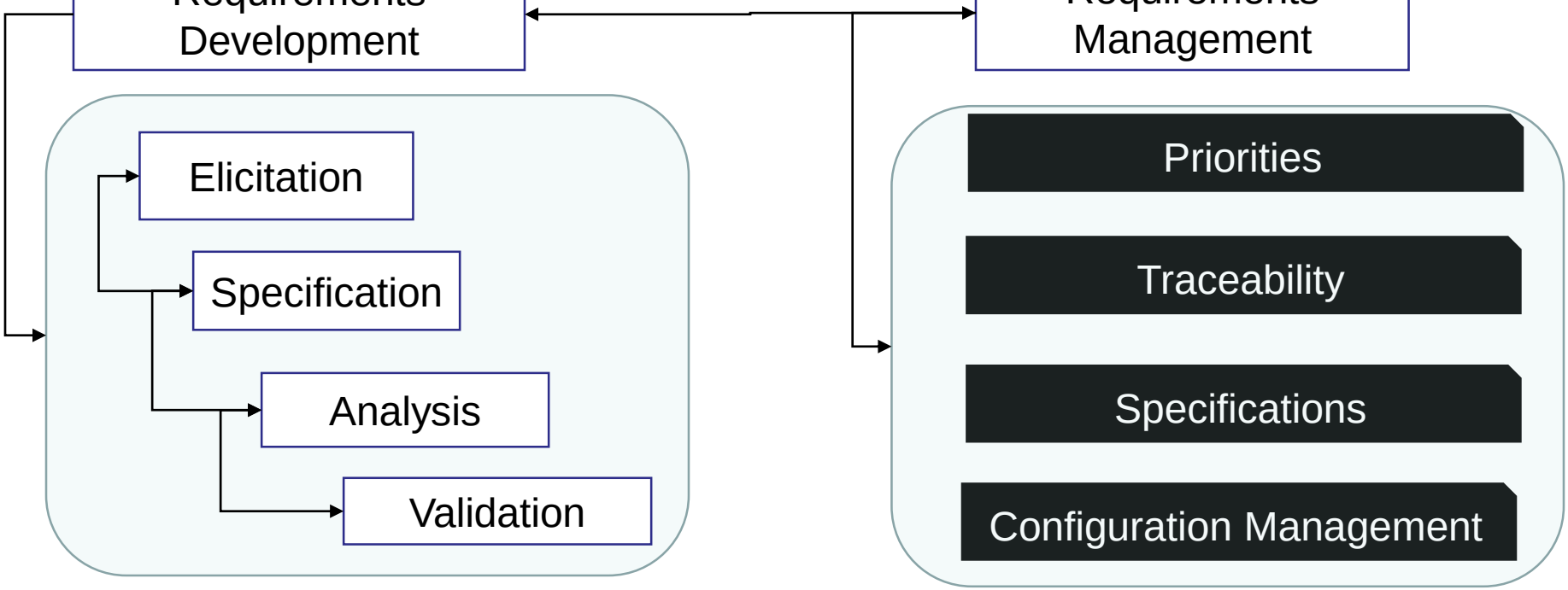
Agile RE: As Is

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]



Agile RE: As Is

"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]

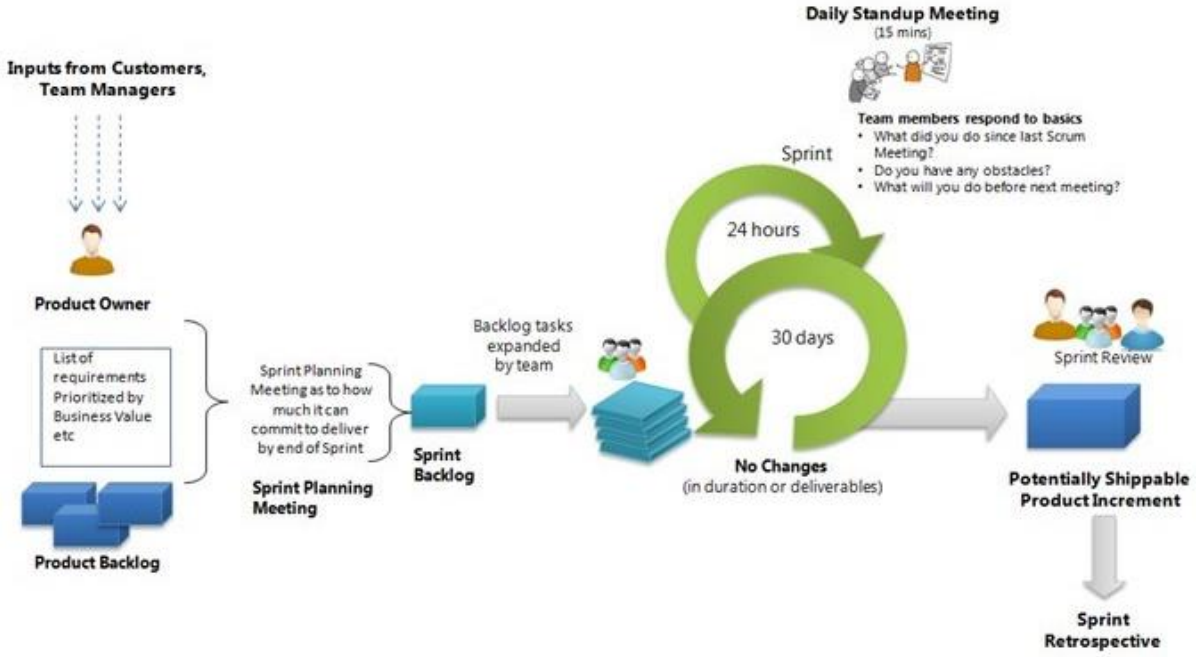


Image source: [28]

Agile RE: As Is

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 [4]

RE activities	Traditional RE approach	Agile RE approach	Agile practices used to support the RE activities
Requirements elicitation	Discovering all the requirements upfront	Iterative: requirements evolve over time and are discovered throughout the development process.	Iterative RE Face-to face communication
Requirements analysis and negotiation	Focus on resolving conflicts	Focus on refining, changing and prioritizing requirements iteratively	Iterative RE Face-to-face communication Constant planning Extreme prioritization
Requirements documentation	Formal documentation contains detailed requirements	No formal documentation	Face-to-face communication
Requirements validation	The consistency and completeness of requirements document	Focus on ascertaining whether the requirements reflect current user needs	Review meetings Face-to-face communication

Agile RE: As Is

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.

[4]

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

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

Agile RE: As Is

These sentiments are shared with other researchers, who also note issues with requirements management. [3] [6] No written documentation results in information loss when code is implemented and refactoring costs skyrocket.

[4]

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

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

Agile RE: As Is

"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]

[4]

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

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

User Story Issues

Image source: [9]

- 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]

Agile in Federal Acquisition

- 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.

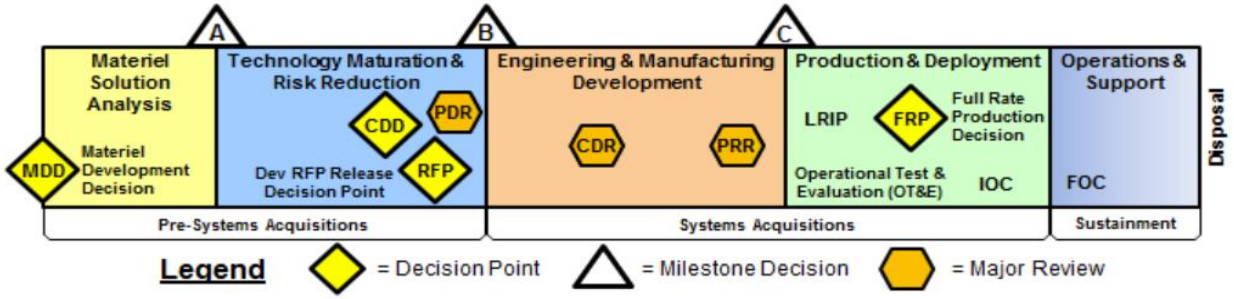


Image source: [26]

- 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]

Agile and the DoD

- 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 [24] 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?*

Abstract of Research Topic

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.

QFD

"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]

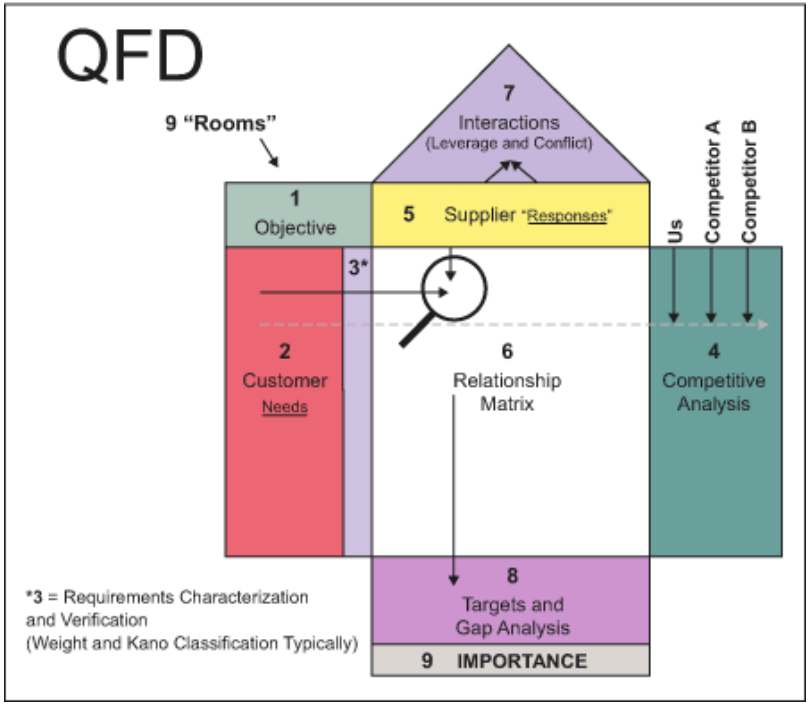
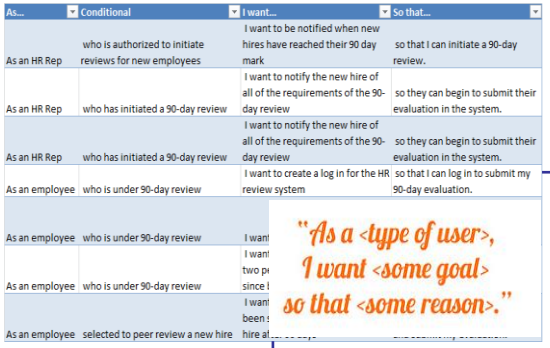


Image source: [23]

"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]

Data Collection for QFD

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.



Requirements Quality Measurement Tools

Baseline quality score(s)

Deconstructed user stories

QFD

Reconstructed User stories

Requirements Quality Measurement Tools

Improved quality score(s)

Traceability, Priorities, etc.

Expert Judgment

Proposed Model

Team member or Customer

Tool Interface

Web-based QFD

Linguistics

Problem Solving Techniques

Inputs

Outputs

User stories

NFRs (constraints)

Traceability

Priorities

TRLs

Market Analysis

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.

Outputs

Tool Interface

Web-based QFD

User stories

NFRs (constraints)

Traceability

Priorities

TRLs

Market Analysis

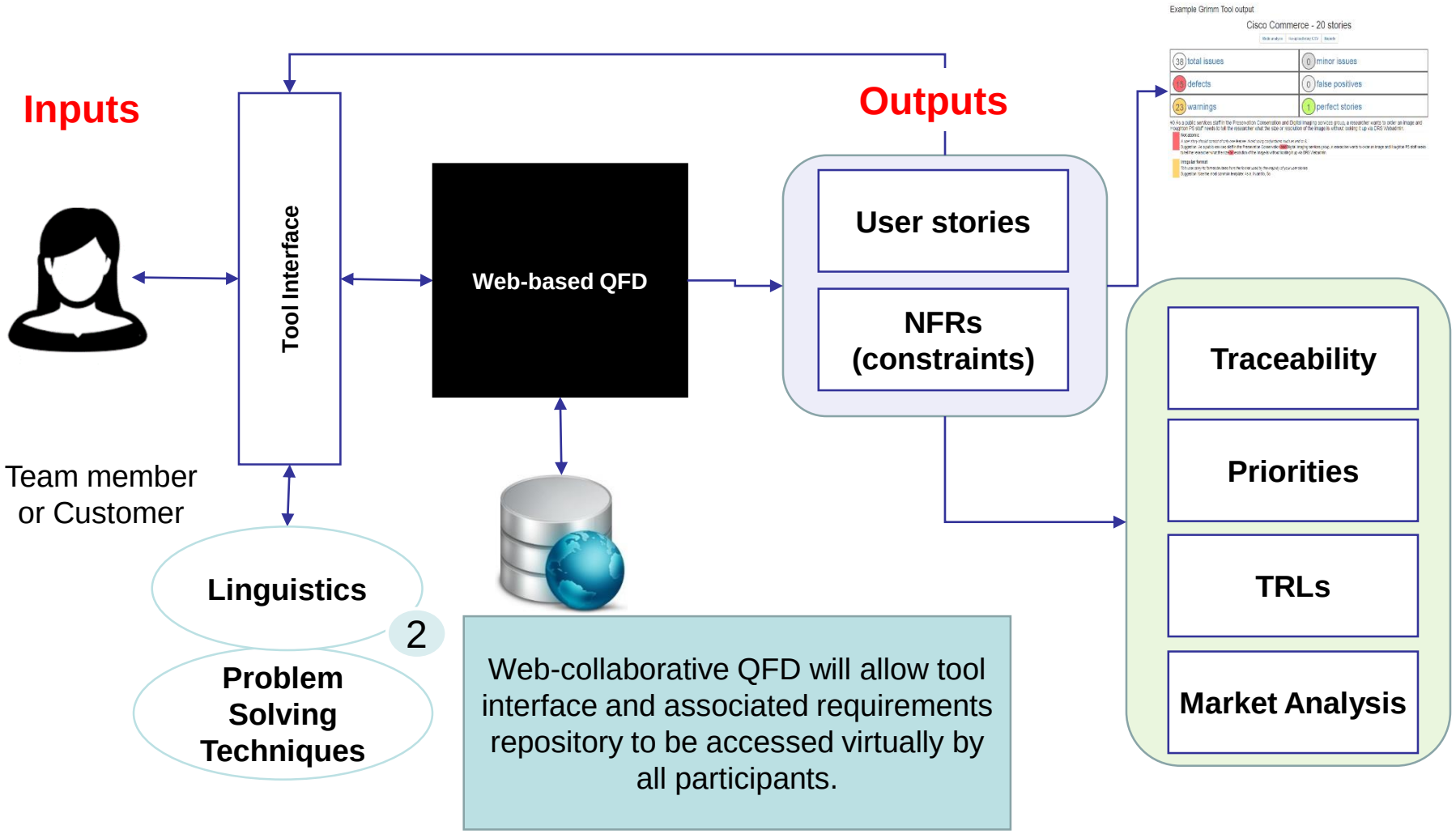
Team member or Customer

Linguistics

Problem Solving Techniques

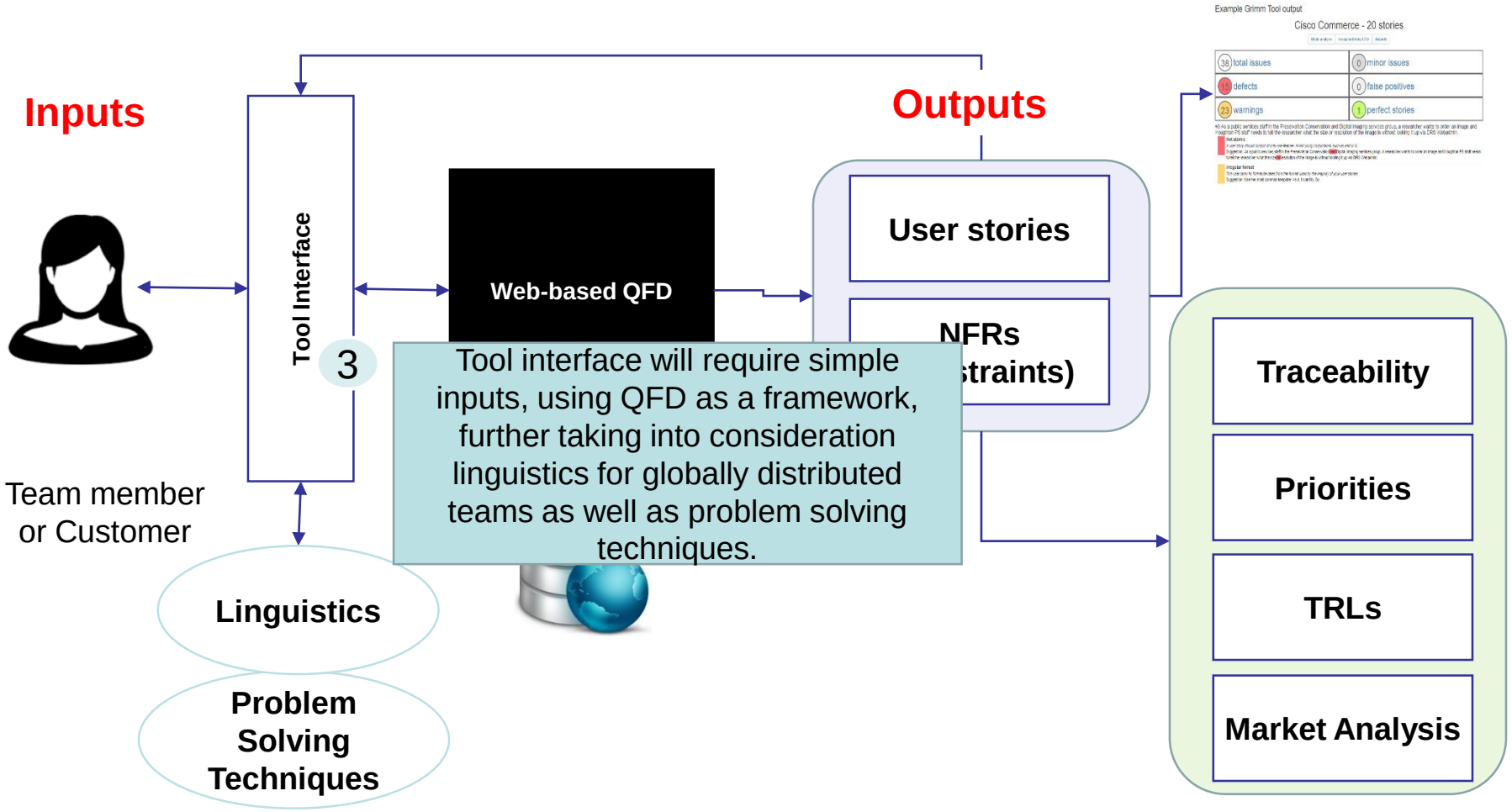
Provide a framework to elicit and manage quality user stories using QFD

Proposed Model



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

Proposed Model



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

Proposed Model

Inputs

Tool Interface

Web-based QFD

Outputs

User stories

NFRs (constraints)

Traceability

iorities

TRLs

Market Analysis

Team member or Customer

Linguistics

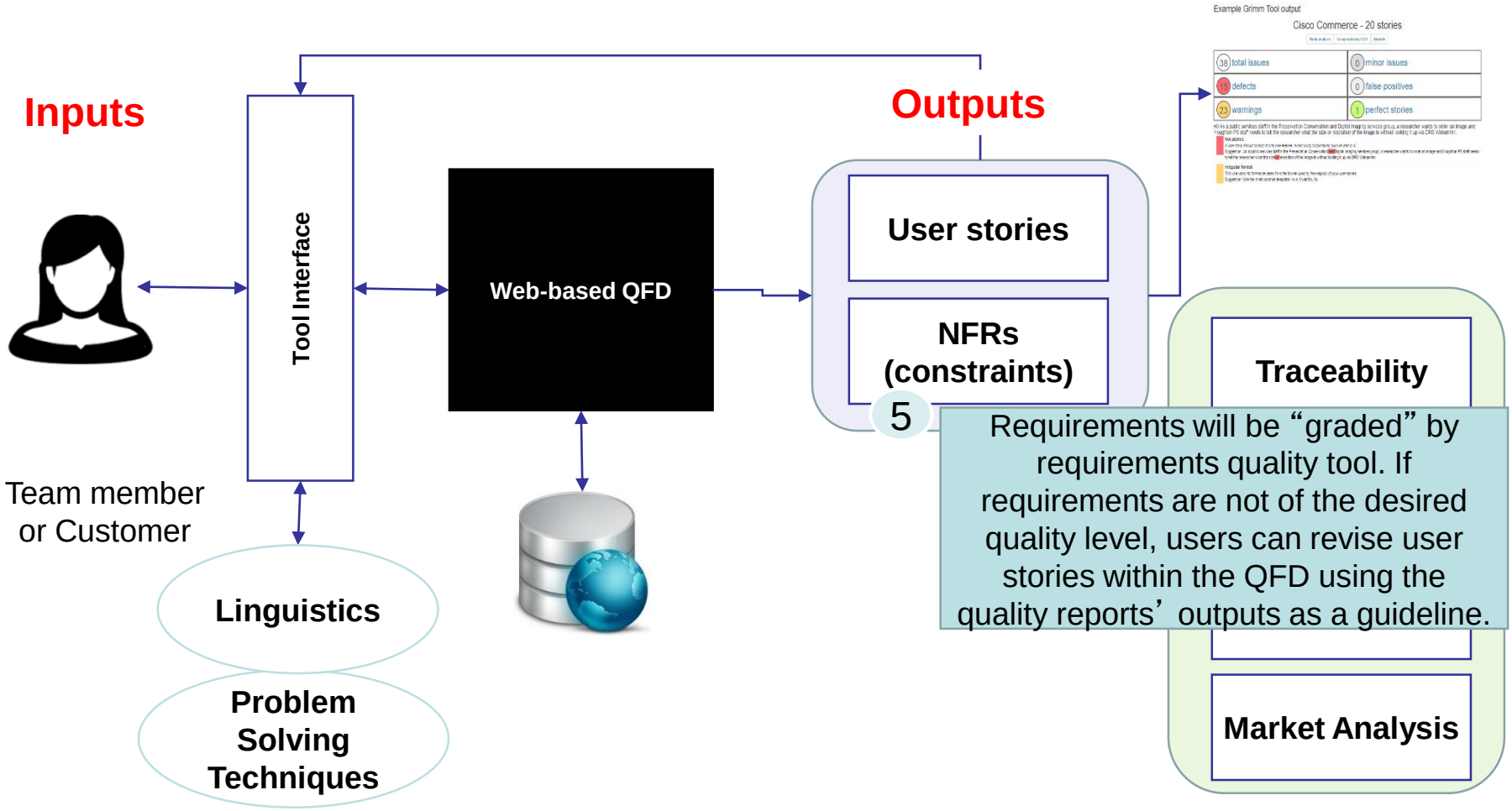
Problem Solving Techniques

4

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.

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

Proposed Model



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

Research Definition

Objectives

- **Methods to create quality user stories**
- **Automatic generation of documentation**
- **Facilitation of distributed stakeholder involvement**
- **Repeatable Agile RE process**

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?

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.

Summary

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

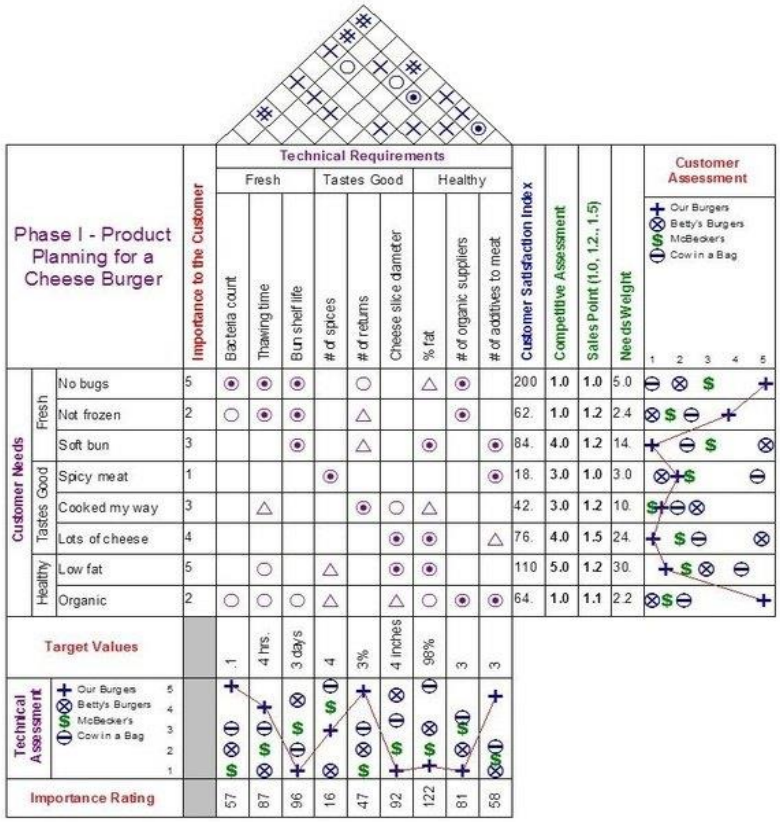


Image source: [27]

References

[1] Eberlein, A., & Leite, J. C. S. P. (2002, September). Agile requirements definition: A view from requirements engineering. In *Proceedings of the International Workshop on Time-Constrained Requirements Engineering (TCRE'02)* (pp. 4-8).

[2] Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch. (n.d.). Retrieved from https://www.infoq.com/articles/standish-chaos-2015

[3] Shah, T., & Patel, V. S. (2014). A review of requirement engineering issues and challenges in various software development methods. *International Journal of Computer Applications*, *99*(15), 36-45.

[4] Ramesh, B., Cao, L., & Baskerville, R. (2010). Agile requirements engineering practices and challenges: an empirical study. *Information Systems Journal*, *20*(5), 449-480.

[5] Agile Scrum Methodology for Mobile Apps. (2016, February 16). Retrieved September 29, 2017, from http://www.innovapptive.com/blog/Agile-scrum-methodology-for-mobile-apps/

[6] Asanka Dilruk, Senior Technology Lead Follow. (2015, April 24). Agile Requirements Engineering Practices: An Empirical Study. Retrieved September 29, 2017, from https://www.slideshare.net/AsankaDilruk/Agile-requirements-engineering-practices-an-empirical-study

[7] De Lucia, A., & Qusef, A. (2010). Requirements engineering in Agile software development. *Journal of Emerging Technologies in Web Intelligence*, *2*(3), 212-220.

[8] Mund, J., Fernandez, D. M., Femmer, H., & Eckhardt, J. (2015, October). Does Quality of Requirements Specifications matter? Combined Results of Two Empirical Studies. In *Empirical Software Engineering and Measurement (ESEM), 2015 ACM/IEEE International Symposium on* (pp. 1-10). IEEE.

[9] Lucassen, G., Dalpiaz, F., van der Werf, J. M. E., & Brinkkemper, S. (2016). Improving Agile requirements: the quality user story framework and tool. *Requirements Engineering*, *21*(3), 383-403.

[10] Challenges with User Stories. (2011, February 23). Retrieved from http://www.seilevel.com/requirements/challenges-with-user-stories

[11] 5 Common Mistakes We Make Writing User Stories. (n.d.). Retrieved from https://www.scrumalliance.org/community/articles/2011/august/5-common-mistakes-we-make-writing-user-stories

[12] 5 Common User Story Mistakes by Roman Pichler. (2017, January 31). Retrieved from http://www.romanpichler.com/blog/5-common-user-story-mistakes/

[13] Limitations of user stories. (n.d.). Retrieved from http://www.ferolen.com/blog/limitations-of-user-stories/

[14] Dahlgren, M. (2017, March 01). 7 reasons some user stories aren't useful stories – Magnus Dahlgren – Medium. Retrieved September 29, 2017, from https://medium.com/@MagnusDahlgren/7-reasons-some-user-stories-arent-useful-stories-d3d2dddbae60

[15] Identifying and Improving Bad User Stories. (n.d.). Retrieved from https://www.Agileconnection.com/article/identifying-and-improving-bad-user-stories?page=0%2C2

References

[16] Gaikwad, V., & Joeg, P. (2016). An Empirical Study of Writing Effective User Stories. *International Journal of Software Engineering and Its Applications*, *10*(11), 387-404.

[17] Paetsch, F., Eberlein, A., & Maurer, F. (2003, June). Requirements engineering and Agile software development. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on* (pp. 308-313). IEEE.

[18] Boehm, B. (2000) Requirements that handle IKIWISI, COTS, and rapid change. *IEEE Computer*, **33**, 99–102.

[19] Bjarnason, E., Wnuk, K., & Regnell, B. (2011, July). A case study on benefits and side-effects of Agile practices in large-scale requirements engineering. In *Proceedings of the 1st Workshop on Agile Requirements Engineering* (p. 3). ACM.

[20] Pfeiffer, S., & Eberlein, A. (2003). *Requirements engineering for dynamic markets*. University of Calgary, Department of Electrical and Computer Engineering.

[21] Paetsch, F., Eberlein, A., & Maurer, F. (2003, June). Requirements engineering and Agile software development. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on* (pp. 308-313). IEEE.

[22] Boehm, B. (2003) *Balancing Agility and Discipline: A Guide for the Perplexed.* Addison-Wesley, Boston, MA, USA.

[23] QFD a Good Tool to Use for Avoiding Product Failure. (n.d.). Retrieved from https://www.isixsigma.com/tools-templates/qfd-house-of-quality/qfd-good-tool-use-avoiding-product-failure/

[24] Modigliani, P., & Chang, S. (2014, March). *Defense Agile Acquisition Guide: Tailoring DoD IT Acquisition Program Structures and Processes to Rapidly Deliver Capabilities (Rep. No. 14-0391).* Retrieved from The MITRE Corporation website: https://www.mitre.org/sites/default/files/publications/MITRE-Defense-Agile-Acquisition-Guide.pdf

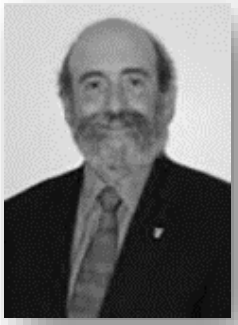
[25] US Department of Defense (DoD) is Going Agile. (n.d.). Retrieved from https://www.infoq.com/news/2014/05/DoD_Agile

[26] Acquisition Process Overview. (n.d.). Retrieved October 03, 2017, from http://acqnotes.com/acqnote/acquisitions/acquisition-process-overview

[27] What is the meaning of QFD "quality function deployment"? (n.d.). Retrieved from https://www.quora.com/What-is-the-meaning-of-QFD-quality-function-deployment

[28] Agile Scrum Methodology for Mobile Apps. (2016, February 16), Retrieved from http://www.innovapptive.com/blog/Agile-scrum-methodology-for-mobile-apps/

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.



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.

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.