

MODULE 1 – REQUIREMENTS ENGINEERING

1.1 REQUIREMENTS ENGINEERING TECHNIQUES

1

1

INTRODUCTION

- **Applying requirements engineering techniques allows Agile teams to:**
 - polish the user stories and epics
 - add context
 - consider impacts and dependencies
 - identify any gaps, such as missing non-functional requirements.



2

2



INTRODUCTION

- Although the majority of the requirements engineering techniques discussed in this section come from traditional development approaches, they are also effective in Agile development.



3

3

AGILE VS TRADITIONAL

- Generally, in traditional projects, requirements engineering activities and techniques are formalized, performed sequentially, and are the responsibility of designated people such as:
 - Business Analysts, Functional Analysts, Technical Architects, Enterprise Architects, and Process Analysts.
- In contrast, in Agile projects, requirements engineering techniques are applied throughout the project and during each iteration via a less formal approach.



4

4

AGILE REQUIREMENTS ENGINEERING

- These requirements engineering techniques are performed more frequently, using continuous feedback loops, by all Agile team members, not just the dedicated business analyst or product owner of the team.



 RICECONSULTING

5

5

LESSON 1

1.1.1 ANALYZE USER STORIES AND EPICS USING REQUIREMENTS ENGINEERING TECHNIQUES

6

LEARNING OBJECTIVES

- **ATT-1.1.1-1 (K4) Analyze user stories and epics using requirements engineering techniques**
- **ATT-1.1.1-2 (K2) Describe the requirements engineering techniques and how they can help testers**



RICECONSULTING

7

7

TECHNIQUES

- **As a tester, to be able to assist in clarifying (and possibly improving) user stories, epics, and other Agile requirements, it is necessary to know, understand, select and use the various requirements engineering techniques that support this.**
- **Examples of such techniques are storyboards, story mapping, personas, diagrams, and use cases.**



RICECONSULTING

8

8

STORYBOARDS

- A Storyboard (not to be confused with Agile task board or Agile user story board) provides a visual representation of the system.
- Storyboards help testers to:
 - **See the thought process** behind the user stories, and the overall “story”, providing context, making it possible to quickly see the functional flow of the system and identify any gaps in the logic.
 - **Visualize groups of user stories** related to a common area of the system (Themes) which can be considered for inclusion in the same iteration, as they will likely be touching the same piece of code.
 - **Assist in Story Mapping and prioritization** of epics and related user stories in the product backlog.



9

9

STORYBOARDS

- **Assist in Story Mapping and prioritization** of epics and related user stories in the product backlog.
- **Assist with identifying acceptance criteria** for user stories and epics.
- **Assist with selecting the right test approach** based on the visual aspect on the system design.
- **Along with Story Mapping, assist in prioritizing tests** and in identifying needs for stubs, drivers and/or mocks.

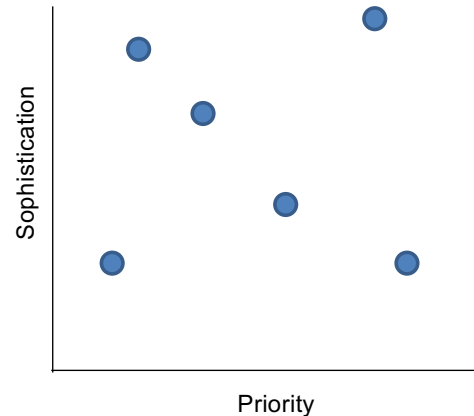


10

10

STORY MAPPING

- **Story Mapping (or User Story Mapping)** is a technique which consists of using 2 independent dimensions to order user stories.
- The horizontal axis of the map represents the order of priority of each of the user stories, while the vertical axis represents the sophistication of the implementation.



11

11

STORY MAPPING

- **Use of Story Mapping can help testers to:**
 - Determine the most basic functionalities of a system to distillate a smoke test.
 - Identify the order of functionalities to determine test priorities.
 - Visualize the scope of the system.
 - Determine the risk level of each user story.



12

12

PERSONAS

- **Personas are used to define fictional characters or archetypes that illustrate how typical users will interact with the system.**



13

13

PERSONAS CAN HELP...

- **Identify gaps in user stories by identifying different types of users that may use the system.**
- **Identify inconsistencies in user stories on how a particular type of user may use the system compared to others.**
- **Elicit user story acceptance criteria**
- **Discover additional test paths during exploratory testing**
- **Reveal test conditions, especially those related to particular user groups, thus helping to ensure sufficient user group coverage and testing of the differences between user groups.**

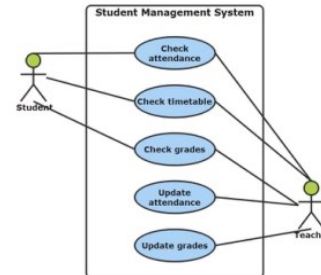


14

14

DIAGRAMS

- Diagrams such as entity relationship diagrams, class diagrams, and (other) UML diagrams can show the structure or flow of data and the functional attributes or behavior of the system, and may be used to identify gaps in system functionality.



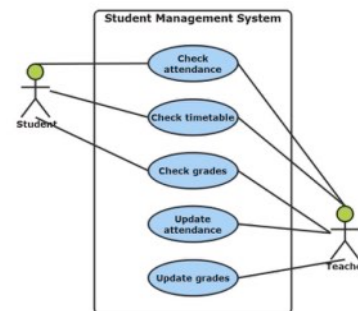
15



15

USE CASES...

- Ensure user stories are testable and properly sized.
- Determine whether the user stories need to be refined or decomposed.
- Reveal forgotten stakeholders.
- Identify interfaces and integration points, which should be considered during test design.
- See the relationships between epics and user stories, to check that the epic doesn't have any missing user stories.



16



16

EXAMPLE – REQUIREMENTS ENGINEERING TECHNIQUES

- A BA working on an agile team has been tasked with eliciting and defining requirements for a new feature.
- This feature will be used by many people who come from a variety of backgrounds and situations.
- So, the BA decided that personas would be an appropriate technique.



17



17

EXAMPLE – REQUIREMENTS ENGINEERING TECHNIQUES

- **Here is what the BA learned:**
 - Inexperienced users found the feature confusing.
 - Experienced users thought the feature lacked some things they felt were very important.
 - Younger users had no difficulty with using the feature.
 - Older users had trouble reading the font size.



18



18

EXAMPLE – REQUIREMENTS ENGINEERING TECHNIQUES

- Based on the input from the various people, the development team was able to make helpful changes that improved the user experience for all groups.



19



19

LESSON 2

1.1.2 IDENTIFYING ACCEPTANCE CRITERIA USING REQUIREMENTS ENGINEERING AND TEST TECHNIQUES

20

LEARNING OBJECTIVES

- **ATT-1.1.2-1 (K4) Create and evaluate testable acceptance criteria for a given user story using requirements engineering and test techniques**
- **ATT-1.1.2-2 (K2) Describe the elicitation techniques**



21

21

THE REQUIREMENTS ENGINEERING PROCESS

- **Elicitation**
 - The process of discovering, understanding, documenting, and reviewing the users' needs and constraints for the system.
 - Elicitation techniques should be used to derive, evaluate and enhance the acceptance criteria.



22

22

THE REQUIREMENTS ENGINEERING PROCESS

- **Documentation**

- The process of documenting the users' needs and constraints clearly and precisely.
- User stories and acceptance criteria should be documented to level corresponding to the team adherence over the principles of the Agile manifesto.
 - "Working software over comprehensive documentation"
- The type of documentation depends on the team's and stakeholders' approach.
- Acceptance criteria can be documented using natural language, using models (e.g., state transition diagrams) or using examples.



23



23

THE REQUIREMENTS ENGINEERING PROCESS

- **Negotiation and Validation**

- For every user story, multiple stakeholders might have other insights or preferences.
- As these insights and preferences might be inconsistent or even conflicting, so can each stakeholder's acceptance criteria.
- Each of these conflicts should be identified, negotiated, and resolved between all impacted stakeholders.
- Every forgotten or non- resolved conflict might endanger the success of the project.
- At the end of this step, the content of each user story is validated by the impacted stakeholders (e.g. as a Definition of Ready).



24



24

THE REQUIREMENTS ENGINEERING PROCESS

- **Management**

- As projects progress, opinions and circumstances might change.
- Even though acceptance criteria were properly elicited, documented, negotiated, and validated, acceptance criteria are still subject to change.
- Because of the potential for changes, user stories should be managed by using good configuration and change management processes.



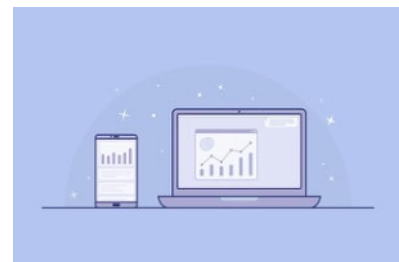
25



25

QUANTITATIVE QUESTIONNAIRES

- **Using quantitative data culled from closed-ended questions is an excellent way of making clear comparisons between various data points.**
 - This will often provide numerical data that can be included in a numerical conclusion for an acceptance criterion.
- **The quantitative questionnaire can be used as an elicitation technique for large number of stakeholders and specifically for non-functional acceptance criteria.**



26



26

QUALITATIVE QUESTIONNAIRES

- **Open-ended questions are an extremely effective way to add more quality to quantitative research.**
- **Open-ends are best used as a follow-up to key questions.**
 - This could generate additional information for which new User Stories have to be created or has to be added to existing ones.
- **The qualitative questionnaire can be used as an elicitation technique for smaller number of stakeholders – as the processing takes more time – and is suitable for functional acceptance criteria.**



27

27

QUALITATIVE INTERVIEW

- **The qualitative interview is more flexible than a quantitative query and is mainly used to acquire information about backgrounds, contexts and causes.**
 - It is unlikely to return hard data, but acceptance criteria can be derived from the responses regarding the context of a user story.
- **The qualitative interview can be the follow-up of any type of questionnaire, to deepen the derived acceptance criteria.**



28

28

OTHER TECHNIQUES

- Multiple other elicitation techniques exist, in the range of observation techniques (e.g. apprenticing), creativity techniques (e.g. 6 Thinking hats) and supporting techniques (e.g. low-fi prototyping).
- The technique toolset of the tester will influence the quality of the elicited acceptance criteria.
- INVEST and SMART, along with test techniques such as equivalence partitioning, boundary value analysis, decision tables, and state transition testing can be used to identify and evaluate acceptance criteria as well.



29

29

EXAMPLE - ELICITATION

- A business analyst is working on an agile project to elicit requirements from a small group of stakeholders.
- The company used to create detailed requirements on projects, but since adopting agile, the focus has been on user stories.
- However, they have learned that most features implemented have to be reworked due to lack of understanding of the true need.



30

30

EXAMPLE - ELICITATION

- To remedy this, the BA is interviewing each key stakeholder about their needs.
- The BA is using a combination of interviews, brainstorming, and storyboarding to get a more complete picture.
- After 3 days of this effort, the BA reviewed the draft requirements with the stakeholders and they felt everything was covered.



31

31

EXAMPLE - ELICITATION

- One of the requirement reads, “The web-based order totaling must apply the correct state and local sales taxes to the purchase.”



32

32

EXAMPLE - ELICITATION

- However, during development and testing, one of the testers noticed a condition not addressed in the requirement.
- Would an international customer have to pay taxes in their own country?



33



33

EXAMPLE - ELICITATION

- The BA asked the stakeholders, who agreed that applying international taxes is not a requirement at this time.



34



34

EXAMPLE - ELICITATION

- The BA also was concerned about another requirement that was written: “The order total must be computed and displayed to the customer in a timely manner.”



35

35

EXAMPLE - ELICITATION

- The BA explained to the stakeholders that “timely manner” was too ambiguous and needed to be more specific.
- However, some stakeholders felt that 2 seconds was good, while others felt that 3 seconds was also good.
- It took several days to reach an agreement, which turned out to be a compromise – 2.5 seconds!



36

36

EXERCISES

- **ATT-1.1.1-1(K4) Analyze user stories and epics using requirements engineering techniques**
- **ATT-1.1.2-1(K4) Create and evaluate testable acceptance criteria for a given user story using requirements engineering and test techniques**

37



37