

---

# Agile Requirements Engineering

GMU Fall 2019 CS 321

Connor Baker



Compiled on September 25, 2019 at 1:18pm

## Review of Previous Class

- Domain requirements
  - A type of (typically non-functional) requirement
  - The system's operational domain imposes requirements on the system
- Metrics for evaluating requirements (for NFRs)
  - Speed
  - Size
  - Ease of use
  - Reliability
  - Robustness
  - Portability
- Discussed ethnography
- Functional vs Non-Functional Requirements

### Requirements engineering process

- Elicitation, understanding user needs
- Extract information that we can use from the requirements
- Verification vs. Validation
  - Validation is a continuous process – “this is what I took away from our meeting. Does this sound like what you're picturing?”
  - Checking that we adequately captured all the needs and expectations of the customer
- What is requirements management?
  - Planning and controlling all of the activities of the RE process
  - Typically involves use of a tool
- Requirements Evaluation process
  - Validity
    - \* Does the system provide the functions which best support the customer's needs?
  - Consistency
    - \* Are there any requirements conflicts?
  - Completeness
    - \* Are all functions required by the customer included?
  - Realism/Feasibility
    - \* Can the requirements be implemented given available budget and technology?
  - Verifiability

- Can the requirements be checked?
- Unambiguous
  - One interpretation of each stated requirement.
- Traceable
  - Origin and realization of each requirement should be clear.

## Agile Manifesto Revisited

- Individuals and interactions over processes and tools
- Customer collaboration over contract negotiation
- Working software over documentation
- Responding to change over planning

While there is value in the items on the right, we value the items on the left more.

## Challenges in RE

- Requirements change over a period of time
- Developers and requirements engineers often misinterpret or fail to capture user needs and intents and specify incorrect requirements
- Stakeholders are not very clear about their needs
- Conflicting stakeholder needs

## Agile RE Concepts and Practices

- Evolutionary Requirements
- Minimal Big Requirements Up Front
- Incremental and iterative implementation of requirements
- Accommodate change late in the development life cycle
- Minimal requirements documentation
- Just-In-Time (JIT) Requirements

## Agile RE Concepts

- Implicit V&V – more of a validation process
- Treat requirements like prioritized stack
- Adopt stakeholder terminology
- Record requirements in the user's vernacular
- Direct stakeholder involvement

## **Agile RE Process (An Overview)**

1. Identify the high level features of the system.
2. Prioritize the features.
3. For each feature to be developed, gather the details JIT from the stakeholders.
4. With the gathered details, proceed to development (Test Driven Development) and acceptance testing

## **Problem Formulation and Education**

### **Objectives**

- Establish rapport among the stakeholders
- Learn the business process of the customer
- Identify user classes

### **Output**

- Product Concept Statement

## **Features**

### **Objectives**

- Identification of release-level features
- Prioritization of the identified features

### **Activities**

- Preparation
- Elicitation
- Validation and Estimation
- Prioritization

### **Output**

- Prioritized features stack

**Example**

Assume an online payment system. The initial set of feature have to be identified before proceeding to the next phase.

**Stories**

For some feature  $f_i$ :

**Objectives**

- Create user stories
- Prioritize the stories

**Activities**

- Preparation
- Elicitation
- Validation and Estimation
- Prioritization

**Output**

- Prioritized features stack

**Example**

As a user, I want to be able to pay by credit card.

- Must be able to accept Visa and Master Card

**Tasks**

For a story  $s_i$ :

**Objectives**

- Create tasks

**Activities**

- Preparation
- Elicitation
- Validation and Estimation

**Output**

- Tasks lists

**Example**

1. Validate user id and password
2. Authorize user
3. Get the card type, card number, expiration date, CVV number and billing address
4. Get amount confirmation
5. Display confirmation message

**Development**

For a story  $s_i$  and a task  $t_j$ :

**Objectives**

- Create tests
- Write code
- Perform customer acceptance tests

**Activities**

- Test Driven Development
- Customer Acceptance Tests

**Output**

- Developed and tested tasks

## **Advantages of Agile RE**

- Accommodates change even late in the development cycle
- Reduces requirements risks and resolves conflicts
- Cost-effective

## **Disadvantages of Agile RE**

- Minimal process/ decomposition of activities in the RE phase
- Non-functional requirements are not well-defined

## **Videos**

### **Agile estimation**

<http://www.youtube.com/watch?v=sCCUEtjCpCs>

### **Agile prioritization**

<http://www.youtube.com/watch?v=zdAiirmT6OI&list=UUi2nY31hV-lnTcDQyazAHUA&feature=c4-overview>

### **NFR specifications**

<http://www.youtube.com/watch?v=NnD7UhnIsNc>