

Software Requirement Engineering (SE-211)

Lecture 26: Requirement Engineering for Agile Methods

Introduction

- Agile methods are a family of software development processes
- Their aim is to deliver products faster, with high quality, and satisfy customer needs through the application of the principles of lean production to software development

Introduction

- Agile methods have become popular during the last few years
- Lean production has been conceived during the '50s in Toyota. It involves several practices that are now part of most manufacturing processes, such as just-in-time development, total quality management, and continuous process improvement

Introduction

- The principle of lean production is the constant identification and removal of waste, that is, anything that does not add value for the customer to the final product

Introduction

- Agile methods focus on
 - Delivering value for the customer
 - Ensuring that the customer understand such value and be satisfied by the project

Introduction

- Agile methods pose a lot of emphasis in producing and delivering to the customer only those features that are useful
- Producing anything that is not required is considered a mistake

Introduction

- Adding a feature that is not needed not only consumes effort without adding customer value but also creates extra code, which may contain errors and make code longer and more complex to maintain, to correct and improve

Introduction

- The waste includes general architectures that are used only partially or reusable components with functionalities that are likely to be never used

Introduction

- To achieve such elimination of waste, agile methods claim to be
 - Adaptive rather than predictive
 - People-oriented rather than process-oriented

Introduction

- A close collaboration between the development team and the customer is a must, so that
 - Requirements are fully identified and correctly understood
 - Final products reflects what the customer needs, no more no less

Agile Methods

- eXtreme Programming
- Scrum
- Dynamic Systems Development Method
- Adaptive Software Development
- The Crystal Family

Agile Manifesto

- Individuals and Interactions over Process and Tools
- Customer Collaboration over Contracts
- Working Software over Documentation
- Responding to Change over Planning

Common Practices and Behaviors

- Adaptability
- Incremental Development
- Frequent Releases
- Requirements Prioritization before every iteration
- High Customer Involvement

RE in Agile Methods

- In agile methods, the whole development team is involved in requirements elicitation and management, while in traditional approaches often only a subset of the development team is involved

RE in Agile Methods

- This approach is feasible only if the size of the problem is limited, as only a small development team can interact directly with the customer
- If the problem is bigger, the team can use other techniques for eliciting and managing requirements

RE in Agile Methods

- The understanding of requirements variability has a strong impact on the ability of agile methods to be “lean’

Focus of Agile Methods in RE

- The Customer
- Waste in Requirements
- Requirements Evolution
- Non-functional Requirements

Customer in RE of Agile Methods

- Customers/stakeholders assume a paramount role in requirements engineering in agile methods
- The interaction between the development team and stakeholders is complex due to the different perceptions of the problem that the stakeholders have

Customer in RE of Agile Methods

- In agile methods, the problem of multiple stakeholders is solved reducing their number to one, who represent all stakeholders in the project
- The customer should be a domain expert and able to make important decisions such as accepting the product, prioritize requirements, etc.

Customer-on-Site Requirements

- Availability
 - The customer has to be always available to answer questions coming from the development team. Any delay in the answer delays the development of the product
- Complete knowledge
 - The customer is the representative for all the stakeholders. Therefore, he is able to answer all questions, since he is the domain expert and knows how the application should work and input/output data required. Again, this is possible if the size of the project is limited
- Decision power
 - The customer is able to make final decisions and commitments. Changes in requirements, acceptance of the features implemented, etc., can be decided directly by the customer, allowing a fast decision making process

Waste in Requirements

- Identification and reduction of waste from requirements assume a paramount role to avoid the creation of waste later in the development process
- In lean practices, the reduction of waste is extremely important because waste always generates more waste

Waste in Requirements

- Waste in requirements includes both wrong and useless requirements
- All the waste generated is a cost for the customer both directly and indirectly
- Such costs are likely to generate further waste inside the customer organization due to the reduced amount of money available to its core business and reduced revenues

Impact of Requirements Waste on Development Process

- More source code to write and higher cost
- Increased complexity of the source code
- Delayed delivery of the final version of the application with all functionalities
- More complex and costly maintenance

Impact of Requirements Waste on Development Process

- More resources required by the application, including: memory usage, processing power, network usage, etc
- Increased complexity of the application from the point of view of the customer (e.g., more complex user interface, more effort to learn how to use the application, etc)

Impact of Requirements Waste on Development Process

- Savings produced by the application in the production process of the customer are delayed

Techniques to Focus on Interaction with Customer

- The whole development team collects requirements from the customer
 - Usage of documents to share knowledge is reduced to a minimum
- Requirements are collected using a common language
 - Requirements are collected using the language of the customer, not any formal language

Techniques to Focus on Interaction with Customer

- Direct interaction between the development team and the customer
 - Reduces both the number of documents required and the probability of misunderstanding due to unnecessary communication layers
- Requirements splitting
 - If the development team considers a requirement too complex, this technique helps the customer to split it in simpler ones

Waste Reduction Techniques

- In order to reduce the waste created by the over specification of requirements, following techniques are used
 - Requirements prioritization
 - Incremental releases

Prioritization Activities

- The development team estimates the time required to implement each functionality. If the effort required is too high, the requirement is split into simpler ones that can be implemented with less effort
- The customer specifies business priorities for each functionality

Prioritization Activities

- According to the business priorities, the development team assign a risk factor to the functionalities
- The customer and the development team identify the functionalities to implement in the iteration

Requirements Evolution

- Agile methods assume that is very hard to elicit all the requirements from the user upfront, at the beginning of a development project
- They also assume that such requirements evolve in time as the customer may change its mind or the overall technical and socio-economical environment may evolve

Requirements Evolution

- Agile companies are aware that changes are inevitable and they include the management of variability into the development process

Requirements Evolution

- Requirements are not well known at the beginning of the project
- Requirements change
- Making changes is not expensive

Requirements Evolution

- Agile methods assume that the cost of introducing changes in a product is nearly constant over the time, but this hypothesis is not true in every context

Requirements Evolution

- Usually, the cost of implementing changes grows exponentially over the time
- On the other hand, if development phases are grouped together in very short iterations and binding decisions are taken as late as possible, the growing of the cost is limited

Requirements Evolution

- In order to manage requirements evolution, agile methods use variable scope-variable price contract. This means that the features really implemented into the system and its cost evolve as well
- Requirements are negotiated throughout the project between the customer the development team

Managing Variability

- Decoupling requirements
- Requirements elicitation and prioritization

Managing Variability

- Decoupling requirements
 - Requirements have to be as independent as possible in order to clearly identify what to implement and make the order of their implementation irrelevant
- Requirements elicitation and prioritization
 - At the beginning of every iteration, there is a requirements collection and prioritization activity. During that, new requirements are identified and prioritized. This approach helps to identify the most important features inside the ongoing project.
- Features are implemented mainly according to their prioritization, not their functional dependence

Non-Functional Requirements

- Agile methods do not provide any widely accepted technique for eliciting and managing non-functional requirements
- Such requirements are collected implicitly during the requirements collection activity. The need of specifying non-functional requirements is less important than in other context due to the continuous interaction with the customer

Roles and Responsibilities

- Customer
- Developers
- Managers

Tools for Requirements Management in Agile Methods

- Paper, pencil, and pin boards
- UML modeling tools
- Requirements negotiation tools
- Instant messaging tools
- Project management tools

Summary

- Provided an introduction to agile methods and their approaches to requirements elicitation and management
- These methods are still evolving
- Customer involvement is a must