Agile Requirements Elicitation

Software Engineering

2004-2005

Marco Scotto (Marco.Scotto@unibz.it)

Content

- **Introduction**
- Requirement analysis
- Characteristics of Agile Requirements
- User Stories
- Gathering User Stories
- Acceptance Tests
- The three Beasts

Introduction

- *Agile practices are based on the belief that neither the customer nor the developers have full knowledge in the beginning and that the important consideration is having practices that will allow both [the customer and the developer] to learn and evolve as that knowledge is gained - without ongoing recrimination.*

Content

Requirement analysis

- We cluster under this term several, often unrelated, activities, including:
  - Requirement elicitation from the users
  - Requirement engineering, to determine the best configuration of requirements
  - Requirement management, to organize the requirements in a structure easy to deal
  - Analysis of the system to build

A web of activities and proposals

- Including:
  - Customer whish-lists
  - Formal specifications
  - Executable specifications and logic programming
  - E-R diagrams
  - Scenario-based techniques
  - Petri nets

Requirements Issues

- What?
  - Example: “Be able to configure all the variables, like user name, password, access level”
- Effort required?
  - Example: “1 week”

Requirements Issues (cont.)

- Priority?
  - Example: “Configuring the password is the most important thing, then access level, then user name”
- Risk?
  - Example: “Are the developers familiar with encryption technology?”

Key Principles

- Separate the “what” from the “how”
  - Requirements should not assume a particular design or implementation
  - What: “Information on ordered books shall be persistently stored”
  - How: “Information on ordered books shall be stored using Database A”
- Must be clear enough so that you know what to build and how to verify that you built it correctly

Controversial Thought: Requirements versus Wishes

- Requirement:
  - That which is required; an imperative or authoritative command; an essential condition; something needed or necessary; a need.
- How do you prioritize “essential conditions”?
- Sometimes not all “requirements” are essential to successfully complete a software project.
- Wishes?

Types of Requirements (1/3)

- There are three kinds of requirements:
  - Functional Requirements
  - Non-functional requirements
  - Constraints

- **Functional requirements**: Describe the interactions between the system and its environment independent from implementation
  - ✓The watch system must display the time based on its location

Types of Requirements (2/3)

- **Non-functional requirements**: User visible aspects of the system not directly related to its functional behavior
  - ✓ The response time must be less than 1 second
  - ✓ The accuracy must be within a second
  - ✓ The watch must be available 24 hours a day except from 2:00am-2:01am and 3:00am-3:01am

Types of Requirements (3/3)

- **Constraints** ("Pseudo requirements"): Imposed by the client or the environment in which the system will operate
  - ✓ The implementation language must be COBOL.
  - ✓ Must interface to the dispatcher system written in 1956.

What is usually not a requirement?

- System structure, implementation technology
- Development methodology
- Development environment
- Implementation language

It is desirable that none of these above are constrained by the client. Fight for it!

Requirements Validation

- Critical step in the development process,
  - Usually after requirements engineering or requirements analysis. Also at delivery
- Requirements validation criteria:
  - Correctness:
    - The requirements represent the client's view
  - Completeness:
    - All possible scenarios through the system are described, including exceptional behavior by the user

Requirements Validation Criteria (cont)

- Consistency:
  - ⌦There are functional or nonfunctional requirements that contradict each other
- Clarity:
  - ⌦There are no ambiguities in the requirements
- Realism:
  - ⌦Requirements can be implemented and delivered
- Traceability:
  - ⌦Each system function can be traced to a corresponding set of functional requirements

Formalizing Requirements

- Verbal
- Written
  - Notes on paper, index cards, sticky notes, formal documents
  - Web sites

- Prototyping
  - But remember, “Prototyping is a learning experience. Its value lies not in the code produced, but in the lessons learned.” (Hunt and Thomas, 2000)

Scenarios-Based Requirement Elicitation

- We will focus on scenario-based requirements elicitation, where a scenario is:
  - "A narrative description of what people do and experience as they try to make use of computer systems and applications" (M. Carrol, Scenario-based Design, Wiley, 1995)
  - A concrete, focused, informal description of a single feature of the system used by a single actor.
- Gathered by observing and interviewing users

Requirements Example

- Use Case – A piece of functionality in the system that gives a user a result of value (Jacobsen et al., 1999)
  - Think scenario or story of the system being used
- Bank machine example (Jacobsen et al., 1999)

Content

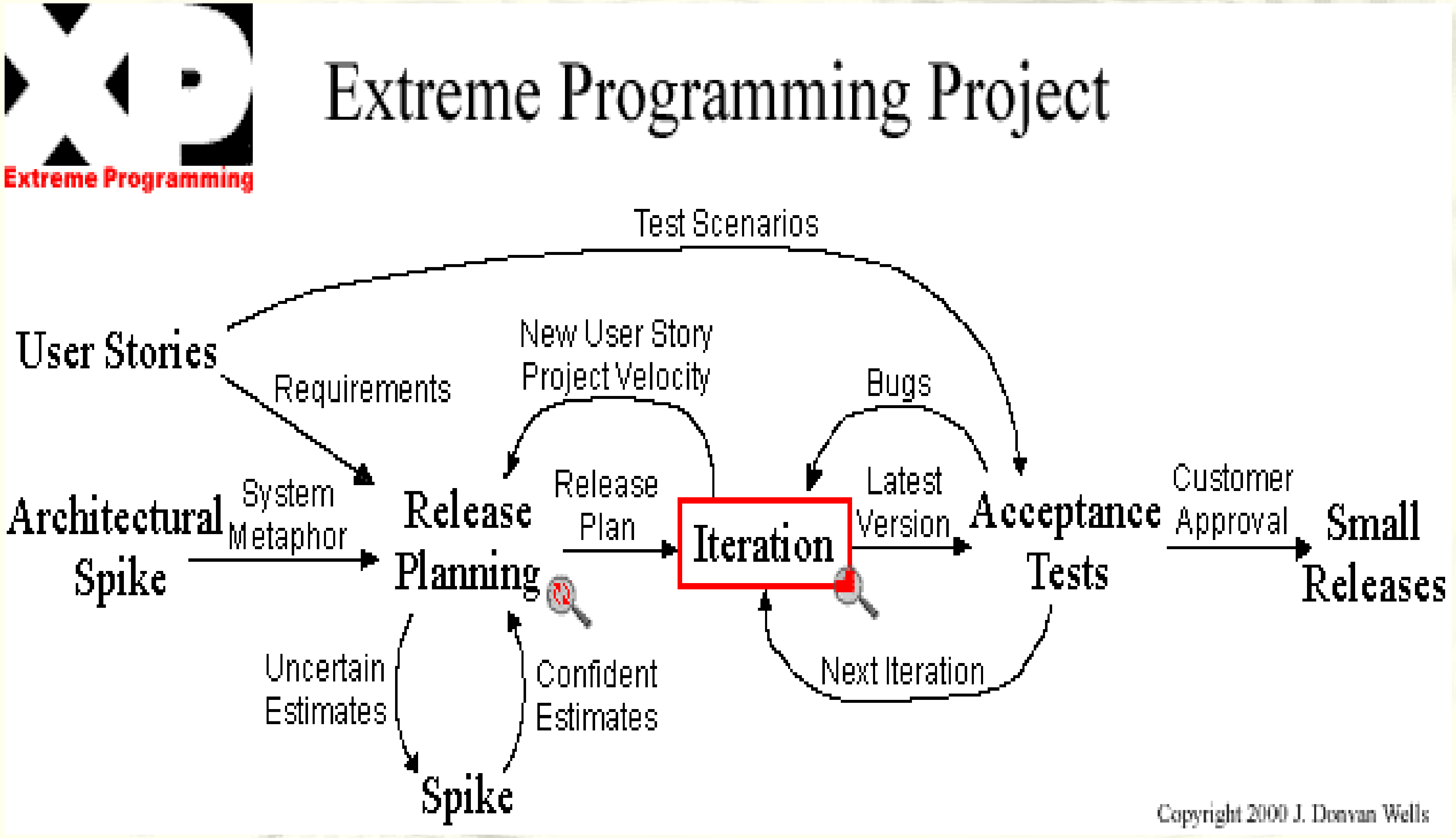
Agile Requirements Elicitation

The hardest part of the software task is arriving at a complete and consistent specification, and much of the essence of building a program is in fact the debugging of the specification

– Fred Brooks, 1987

There is nothing that focuses requirements better than seeing the nascent system come to life. Therefore, capturing the specific details about the requirement long before it is implemented is likely to result in wasted effort and premature focusing.

Agile Requirements in XP



Features of Agile Requirements

- Suitable for projects with a lot of change, not for stable projects with safety-critical implications
- Agile requirements are expressed as high-level, brief written statements of the best information fairly easily available.
- Not fully documented form of the requirements

Characteristics of Agile Requirements (1/2)

- Frequent, personal interactions with customers and/or stakeholders
  (in XP: customer on-site)
  - Necessary for the developers to understand the details of what the customer really wants
- Frequent delivery of software to customers
  (in XP: weekly iteration)
  - Only when the customer can actually use the evolving project, can he/she provide feedbacks and new, refreshed view of follow-on requirements based on the progress so far

Characteristics of Agile Requirements (2/2)

- Expressing Requirements as Features
  - A feature is a short phrase that describes what the customer wants
    - E.g. Items on auction can be added, deleted, and modified
- Expressing Requirements as User Stories
  - Of the agile methodologies, XP one specifies its requirements practices with the most detail and rigor

Content

User Stories basics

- User stories are a simple, few-sentence description of a functional requirement that provides value to a stakeholder
- User stories are written by a customer
- User stories are usually written on an index card which is passed to the team member who will work on it
- Developers estimate how long the stories might take to implement
  - ideal development time: how long it would take to implement the story in code without distractions, other assignments and problems
- Each story should be implemented in one iteration

Content

Gathering User Stories

- Goal-oriented approach:
  - starting point: goal of the system
  - what steps does the user need to achieve the goal?
  - describe them as user stories
- Scattergun approach:
  - no structure is imposed on the way the meeting progresses
  - user stories are generated as expectations arise in a conversation
- Story writing is iterative and interactive

How to Write an Agile Requirement User Story

- User stories appear on an index cards like this:

Title: Login		
Acceptance Test: LoginTest	Priority: 2	Story Points: 2
A user has to input his/her nickname and password to login the system. If the combination of the nickname and password is not correct, an error message will show, and the user has to input the nickname and password again. Personal page and bid functionality are available only when the user logs in successfully. Administrative functionality is available only when the user with administrative privileges logs in successfully.		

Explanation of the example

- ***Title.*** Write a two or three word title for this user story. The title should begin with a present-tense verb phrase in active voice (similar to the name of a use case). Write this title in the middle of the top line of an index card.
- ***Acceptance Test.*** List the unique identifiers of the acceptance tests for the user story. The unique identifier can be a word (such as the example above) or a alphanumeric string.
- ***Priority.*** The customer must decide how important each of the stories is so that the most important stories can be done first. We are using a 1-2-3 priority scheme where a 1 is given to the most important stories.
- ***Story Points.*** The number of days of ideal development time.
- ***Description:*** Write 1-2 sentences which are a single step toward achieving the goal.

Criteria for User Stories

- Stories must be understandable to the customer (natural language!)
- Each story must provide value to the customer
- Developers do not write stories
- Stories need to be of a size that several of them can be completed in each iteration
- Stories should be independent of each other as possible (no inter-dependency → freedom to schedule the work for the developer)
- Each story must be testable

Content

Acceptance Tests

- An acceptance test is a test case written by the customer (in partnership with the developers)
  - Verifies that the user story has been correctly implemented
- The details about the user stories are captured in the form of acceptance tests specified by the customer
- Traceability between the user story and the acceptance test(s) used to verify that user story
- At least one acceptance test case per user story

Additional requirements

- Non-functional requirements and the constraints are hard to be written as user stories
- The user story cards should be augmented with a listing of non-functional requirements and constraints
  - for example: the system must be 96% reliable

Ex. Non-functional requirements and constraints

- Non-functional requirements
  1. All transactions must complete within 5 seconds of the submit button being clicked.
  2. The mean-time-to-failure (MTTF) must be a minimum of 1000 hours. Failure downtime must not exceed 10 minutes.
- Constraints
  1. The system shall be run on a web server that uses the Tomcat application server.
  2. The system shall be run on a web server that uses a MySQL database.
  3. The system shall be developed using Java Server Page (JSP)

User story summary

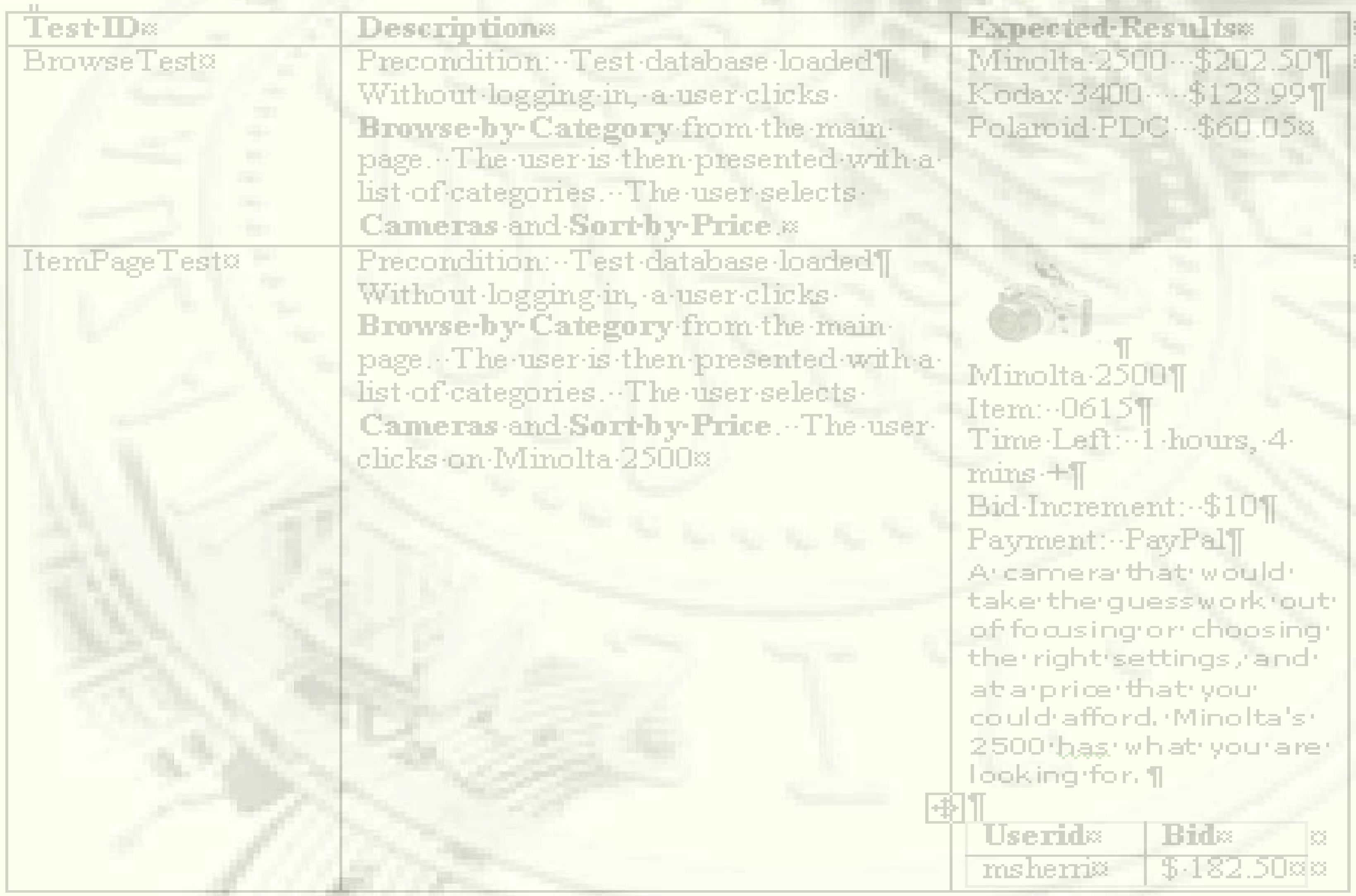
User Stories	Priority	Points	Acceptance Tests
Browse auction items	1	7	BrowseTest
View item information	1	3	ItemPageTest
Register	2	3	RegisterTest
Login	2	2	LoginTest
View/Update personal information	2	3	PersonalInfoTest
Place Bid	2	4	BidTest
Add items	2	3	ItemAddTest
Access appropriate page	3	2	MainPageTest
Update item status	3	4	ItemUpdateTest
Maintain Categories	3	3	CategoryUpdateTest
Maintain user information	3	2	UserInfoTest
Total Story Points		38	

A second user story example

Register		
Acceptance Test: RegisterTest	Priority: 2	Point: 3

The users can register online. A registration page asks the user's information, including the name of the user, a nickname (serves as the user ID), birthday, address, phone number, and email address. After registration, the user will get a temporary password that he/she may change later.
If the nickname is already registered, an error message will show and the user will have to choose another nickname.
To be registered as a seller, in addition to the user information, he/she also has to provide the SSN. If the SSN is already registered, the user will not be allowed to register again.

Sample Acceptance Test



Test ID	Description	Expected Results
BrowseTest	Precondition: Test database loaded Without logging in, a user clicks **Browse by Category** from the main page. The user is then presented with a list of categories. The user selects **Cameras** and **Sort by Price**.	Minolta 2500 $202.50 Kodax 3400 $123.99 Polaroid PDC $60.05
ItemPageTest	Precondition: Test database loaded Without logging in, a user clicks **Browse by Category** from the main page. The user is then presented with a list of categories. The user selects **Cameras** and **Sort by Price**. The user clicks on Minolta 2500	 Minolta 2500 Item: 0615 Time Left: 1 hours, 4 mins + Bid Increment: $10 Payment: PayPal A camera that would take the guesswork out of focusing or choosing the right settings, and at a price that you could afford. Minolta's 2500 has what you are looking for. **Userid** / **Bid** msherri / $ 182.50

Proposed exercise

- Write a few user stories for a traffic light system.

Content

Agile Requirements and the three beasts

- Uncertainty
  - Agile requirements embraces change, allowing the uncertain to resolve itself as project development proceeds
- Irreversibility
  - Since agile requirements consume less resource, the requirements may be economically reversible
- Complexity
  - By attacking each requirement in turn rather than trying to digest the entire project at once, project development can be kept within the intellectual control of the team