Software Requirements: An Evolution Point of View

Julio Cesar Sampaio do Prado Leite
Department of Computer Science, University of Toronto
on leave from Departamento de Informática, PUC-Rio

talk at:
Summary

• Who we are
• Why are we here?
• Different Perspectives
• Evolution as a key aspect
• Results so far (from our group)
• Challenges
Who we are

• Member of the IFIP 2.9 WG
• Previous work on Elicitation, Modeling and Analysis
• 3 Phds and 11 masters on the topic.
• Lexicon, Scenarios, Viewpoints, Requirements Baseline, Extreme Requirements
• See more at: www.inf.puc-rio.br/~julio
• On sabbatical at UofT, involved in the goal/agent (Tropos) method (Prof. Mylopoulos)
Why are we here?

• von Neumann:
  “There is no sense in being precise when you don’t even know what you are talking about”

• Standish Group
  74% of the projects had failed!

• Tom de Marco
  56% of the errors in a software can be traced back to the requirements
Why are we here?

Phase in which fault is detected and fixed
Different Perspectives

- Waterfall Model
- V Model
- Spiral Model
- Extreme Programming
- Open Source
- Tropos
Different Perspectives

Waterfall Model

Problem → Requirements → System-Design → Code → Maintenace
Different Perspectives

V Model
Different Perspectives

Spiral Model

Boehm
Different Perspectives

XP

Extreme Programming Project
Different Perspectives

Open Source

Picture taken from:
Different Perspectives

Tropos

© 2003 John Mylopoulos
Evolution as a Key Aspect

- Requirements process
- Change of perception
- Requirements Baseline
- Is “Quality is job one” still valid?
Evolution as a Key Aspect
Requirements Process

- elicit
- model
- analyze

manage

Time .......

© jcspl 04/2003
Evolution as a Key Aspect

Requirements Process

[New System] [Existing System]

- Analyze the Problem
  - [Incorrect problem]
  - [Addressing correct problem]

- Understand Stakeholder Needs

[New Input]

- Manage Changing Requirements

- Define the System
- Manage the Scope of the System
  - [Can’t do all the work]
  - [Work in scope]

- [Requirements Definition Complete]
- Refine the System Definition

[More Iterations]
Evolution as a Key Aspect

Requirements Process

State of the Practice (Tools)

Starbase
http://www.tbi.com/caliberrm30/index.cfm

Telelogic
http://www.er.les.inf.puc-rio.br/doors.htm

Rational
http://www.rational.com/products/reqpro/index.jsp

EDS
http://www.eds.com/products/plm/teamcenter/slate/
Evolution as a Key Aspect
Change of Perception

- Completeness
- Consistency
- Frozen document
- Constant change
- Time to market
Evolution as a Key Aspect

Requirements Baseline
Evolution as a Key Aspect

Is “Quality is Job One” still valid?

• Of course, but…

• Efficacy is measured time wise

• Functional and Non-Functional requirements are tangled even more.
Results so Far
(from our group)

- Lexicon
- Scenarios
- NFR
- Scenarios Inspection (as an instance of quality driven methods)
- Extreme Requirements
Results so Far
Lexicon

Actuators / Actuator / Physical actuator
Notion:
It is a device that can be controlled by control system. An actuator has name, abbreviation, type, range, control, reaction time and a description.

Behavioral responses:
It is controlled by the control system. An actuator responds in linear time. It controls light.

Sensors / Physical sensor / Sensor
Notion:
A device that can sense state of the building, users or environment. A sensor has name, abbreviation, type, resolution, range, reaction time and conversion time.

Behavioral responses:
Analog sensors respond in exponential time. A sensor is triggered by a physical occurrence under its range. (It is not defined in the text, here is my definition)

Reaction time
Notion:
For a sensor, it is the time from a change of the sensed property to the time when the sensor has reached 90% of the change, excluding conversion time. For an actuator, it is the time to change from 0 to 100% / 100 to 0%, if different.

Behavioral responses:
It is not defined in the text. It is activated by a change in the environment (this need to be elaborated) in the case of a sensor. It is activated by the control system in the case of an actuator.

Dimmer actuators / Ile / Dimmer actuator
Notion:
It is an actuator. It controls the output of a luminaire.

Behavioral responses:
It is used to dim individually ceiling light groups
Results so Far

Scenarios

Scenario: Reoccupied room

Goal: Return to the previous light scene

Context: Motion detector is working, value T1 is known for this room. Any room in the 4th floor of building 32.

Resources: ceiling light groups, task lights, push-buttons, control panel

Actors: user, control system, motion detector, dimmer actuators, status lines, control system

Episodes:
user enters the room.
motion detector signals to control system.
system verifies how long the room has been empty
If time has been shorter than T1 the system retrieves the last chosen light scene
Exception: OCCUPIED ROOM
control system terminates the standard light scene
system implements last chosen light scene
Exception: Light Malfunction
Results so Far

NFR

Facility Manager

Safety [Room]

Safety [Room. Light scene. Current light scene >= Safe Illumination]

Safety [Light Scene. Safe Illumination=14 Lux]

Safety [Malfunction. Malfunction of OLS ]

Safety [Malfunction. Motion Detector]

Safety [Malfunction. User. Get informed]

Safety [Malfunction of OLS. All CLG set on]

Safety [Malfunction. FM Get informed]

Safety [Room. Malfunction]

Safety [Room. Control Panel]

Safety [Located Near the Door]
Results so Far

Scenarios Inspection
Results so Far
Extreme Requirements

XP Problems

• The assumption that, in the planning game, the business could be represented by just one customer.

• The lack of consideration of non-functional requirements from the standpoint of the business.

• The lack of explicit links between stories and tasks cards to the code

• The lack of a process for producing functional tests.

• The lack of a process for producing stories and tasks.
Results so Far

Extreme Requirements

XP Practices

Whole Team

Collective Ownership

Test-Driven Development

Coding Standard

Pair Programming

Refactoring

Planning Game

Customer Tests

Continuous Integration

Simple Design

Sustainable Pace

Metaphor

Small Releases
Challenges

• Produce software that is more flexible, more reliable and more quick to enter the market.

• Goal oriented approach seems to be an important step forward towards evolution

• Engineering methods for goal/agent based methods