COMP 3111
SOFTWARE ENGINEERING

LECTURE 10
SYSTEM REQUIREMENTS CAPTURE

SYSTEM REQUIREMENTS CAPTURE OUTLINE

- ✓ System Requirements Capture Overview
  - Life-cycle Role
  - Importance of Requirements Capture
  - Why Requirements Capture is Difficult

- ➔ **System Requirements Capture Activities**
  - ✓ Capture Data Requirements: Domain Modeling
  - ✓ Capture Functional Requirements: Use-Case Modeling
  - ☞ **Capture Nonfunctional Requirements**
  - **Validate System Requirements**

NONFUNCTIONAL REQUIREMENTS

A *nonfunctional requirement* places a constraint on a use case or on the system.

It is identified by asking questions about the system's:

Design Quality – reliability, supportability, maintainability, etc.

Documentation – what is required and for who?

Hardware – implementation platform, memory size, storage capacity.

Implementation – standards, languages, error handling.

Interface – user interface (learnability, usability);
external system interface (formats, timing).

Management – system back up, installation, maintenance.

Performance – speed, throughput, response time, accuracy, etc.

Physical environment – abnormal conditions; distributed operation

Security – system access; data access; physical access.

NONFUNCTIONAL REQUIREMENTS EXAMPLE: SATWATCH

- SatWatch is a wrist watch that:
  - uses GPS satellites to determine its location and displays the time based on its current location.
  - uses internal data structures to convert this location into a time zone.
  - never requires the owner to reset the time due to the information it stores and its accuracy.
  - has no buttons or controls available to the user since it adjusts the time and the date displayed as the watch owner crosses time zones.
  - assumes that it does not cross a time zone boundary during a GPS blackout period, but adjusts its time zone as soon as possible after the blackout period.
  - has a two-line display showing, on the top line, the time (hour, minute, second, time zone) and on the bottom line, the date (weekday, day, month, year).
  - has a readable display even under poor light conditions.
  - can have its software upgraded using the WebifyWatch device (provided with the watch) and a personal computer connected to the Internet.

NONFUNCTIONAL REQUIREMENTS EXAMPLE: SATWATCH (cont'd)

- Any user who knows how to read a digital watch and understand international time zone abbreviations should be able to use SatWatch.

Type of nonfunctional requirement?

- As SatWatch has no buttons, no software faults requiring the resetting of the watch should occur.

Type of nonfunctional requirement?

- SatWatch should accept upgrades to its onboard processor via the USB interface.

Type of nonfunctional requirement?

NONFUNCTIONAL REQUIREMENTS EXAMPLE: SATWATCH

(cont'd)

- SatWatch should display the correct time zone within 5 minutes of the end of a GPS blackout period.

Type of nonfunctional requirement?

- SatWatch should measure time within 1/100th second over 5 years.

Type of nonfunctional requirement?

- SatWatch should display time correctly in all 24 time zones.

Type of nonfunctional requirement?

NONFUNCTIONAL REQUIREMENTS EXAMPLE: SATWATCH (cont'd)

- All related software associated with SatWatch will be written using Java.

Type of nonfunctional requirement?

- SatWatch complies with the physical, electrical, and software interfaces defined by WebifyWatch API 2.0.

Type of nonfunctional requirement?

SPECIFYING NONFUNCTIONAL REQUIREMENTS

Nonfunctional requirements are specified as supplementary requirements on use cases or on the system as a whole.

- Some nonfunctional requirements will be implemented as administration use cases.
  - Login
  - System start up
  - System shut down
  - System backup

Administration use cases **are use cases that deal with non-functional requirements such as security or system operation and maintenance.**

SYSTEM REQUIREMENTS CAPTURE OUTLINE

- ✓ System Requirements Capture Overview
  - Life-cycle Role
  - Importance of Requirements Capture
  - Why Requirements Capture is Difficult

- ➔ **System Requirements Capture Activities**
  - ✓ Capture Data Requirements: Domain Modeling
  - ✓ Capture Functional Requirements: Use-Case Modeling
  - ✓ Capture Nonfunctional Requirements
  - ☞ **Validate System Requirements**

VALIDATE SYSTEM REQUIREMENTS

The system requirements specification (SRS) should be validated continuously with the client/user to verify that they are:

complete – the requirements describe all possible features of interest, including the handling of exceptional behaviour.

- All aspects of the system are represented in the SRS.

consistent – the requirements do not contradict themselves.

clear – the requirements define exactly one system.

- It is not possible to interpret the SRS in two or more different ways.

correct – the requirements represent the features of interest to the client.

- Everything in the SRS accurately represents an aspect of the system.

realistic – the system can be implemented within the given constraints.

Acceptance tests are the primary means to validate that the system implementation satisfies the requirements.

REQUIREMENTS VALIDATION EXAMPLE: SATWATCH

Incompleteness

Problem: The SatWatch specification does not specify the boundary behavior when the user is standing within GPS accuracy limitations of a time zone's boundary.

Solution:

REQUIREMENTS VALIDATION EXAMPLE: SATWATCH (cont'd)

Inconsistent

Problem: SatWatch software should not have bugs nor need to be upgraded.

SatWatch software should be easily upgraded using the USB interface.

Solution:

REQUIREMENTS VALIDATION EXAMPLE: SATWATCH (cont'd)

Unclear

Problem: The SatWatch specification refers to time zones.

Does the SatWatch deal with daylight saving time or not?

Solution:

REQUIREMENTS VALIDATION EXAMPLE: SATWATCH (cont'd)

Incorrect

Problem: SatWatch supports supports only 24 time zones (24 hours).

There are more than 24 time zones. Several countries and territories are half an hour ahead of a neighboring time zone.

Solution:

SYSTEM REQUIREMENTS CAPTURE: RETROSPECTIVE

Domain Modeling

Captures the data requirements of an application.

class diagram – shows classes and the relationships among them.

Use-case Modeling

Captures the functional requirements of an application.

use-case model – shows use cases that provide system functionality and actors that use the functionality.

flow of events – describes the sequence of actions that comprise a use case's functionality.

Requirements Validation

Verifies that the system meets all stated requirements.

SYSTEM REQUIREMENTS CAPTURE: SUMMARY

- Requirements are captured over several iterations by specifying:
  - a domain model
  - a use-case model
  - plus any nonfunctional requirements.

These are all documented in the
System Requirements Specification (SRS).

- In subsequent iterations/phases we refine and/or transform the requirements by specifying:
  - more detail for each of the above artifacts.
  - a set of test cases in the test model.
  - matching use-case realizations in the analysis model.
  - matching use-case realizations in the design model.

The use cases drive the subsequent iterations/phases.

COMP 3111 SYLLABUS

1. ✓ **Introduction**
2. ✓ **Software Development**
3. ✓ **Modeling Software Systems using UML**
4. ✓ **System Requirements Capture**
5. Implementation
6. Testing
7. System Analysis and System Design
8. Software Quality Assurance
9. Managing Software Development

SYSTEM REQUIREMENTS CAPTURE

REQUIREMENTS EXERCISE