Slide 1

Welcome to SENG 321
Requirements Engineering
Let's make this an engaging course

SENG 321

Professor Hausi A. Müller PhD PEng FCAE
Department of Computer Science
Faculty of Engineering
University of Victoria

http://www.engr.uvic.ca/~seng321/
https://courses1.csc.uvic.ca/courses/201/spring/seng/321

Slide 2

Announcements

- Tue, Feb 16 (yesterday)
  - Deliverable S1 due
- Thu, Feb 18
  - Deliverable C1 due (tomorrow)
- Marks for S0 & C0
  - Posted
- S2 & C2
  - Will be posted on Saturday

- Quiz 1
  - **Wed, Feb 24 in class**
  - Class attendance must increase
- Midterm
  - Fri, Feb 26 in class
  - 3 mid questions today

2

Slide 3

SENG 321 Calendar

Deliverable S1 due	Tue, Feb 16	S1 formal req spec	10% of project
Deliverable C1	Thu, Feb 18	C1 feedback on S1	5% of project
Midterm	Fri, Feb 26	In class	14% of project
Deliverable S2a	Tue, Mar 1	S2a detailed formal req spec	10% of project
Deliverable S2b	Thu, Mar 3	S2b & demos	5% of project
Deliverable C2	Tue, Mar 8	C2 feedback on 23a&b	5% of project

3

Slide 4

BSENG Accreditation

- Sun-Tue, Feb 21-23

Thank you

- Mon, Feb 22 – 4-5 pm in ECS 227
  - Need 10 students to talk to Canadian Engineering Accreditation Board (CEAB) site visit team
  - Sign up list

4

Slide 5

Types of Projects

➔ Impact on customer-developer links
- Package (**P**)
  - (C)OTS
  - External sale
- Custom (**C**)
  - In-house development or contracted
  - Internal use
- Many shades of gray ...

5

Slide 6

Writing down the Problem

- *Goal:* Upgrade a legacy information system (e.g., UVic Banner Finance System), which provides invoicing and financial reporting between the university and its faculties and departments

- *Theme:* Improve communications between university and units

- *Development team*

- *Management's vision*

6

Writing down the Problem

- *Goal:* Upgrade a legacy information system (e.g., UVic Banner Finance System), which provides invoicing and financial reporting between the university and its faculties and departments
- *Theme:* Improve communications between university and units
- *Development team* envisioned a powerful new system that provided better financial reporting, improved invoice and statement formats, online parts ordering, and electronic mail. And oh, by the way, the team eventually *hoped* no provide the capabilities for electronic fund transfers between the university and its units.
- *Management's vision:* The primary goal of the new system is to *provide electronic fund transfer to improve the information flow of the university.* A new architecture is needed due to the increased risks of online transactions.
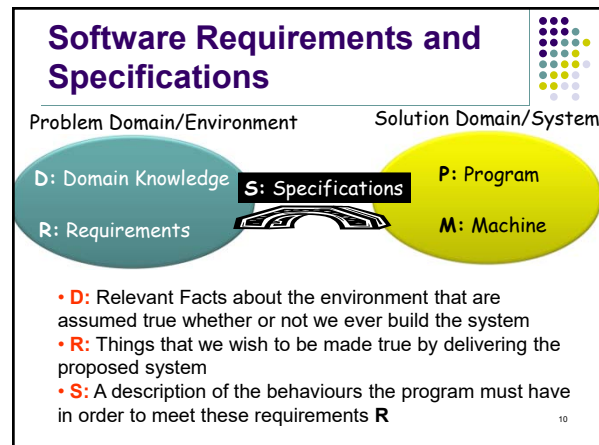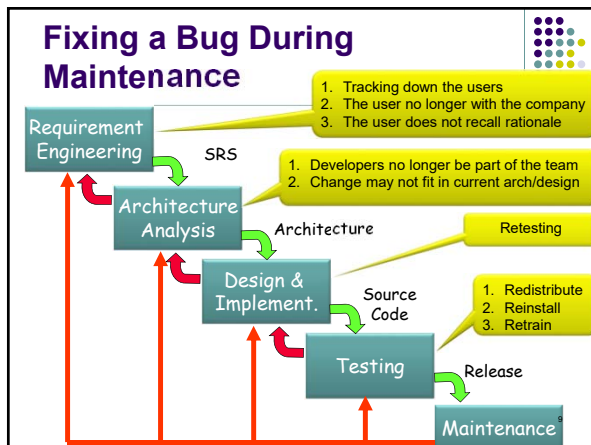
7

Back to Motivation … Why Write Requirements?

- Requirements errors constitute the most common class of errors—80% of field errors are traceable to requirements errors
- Requirements errors are the most expensive to fix

Requirements Error is Found During … Phase	Relative Repair Cost
Requirements	1 – 2
Design	5
Coding	10
Unit Testing	20
Acceptance Testing	50
Maintenance [Davis 1993]	200

8

Fixing a Bug During Maintenance

Requirement Engineering
1. Tracking down the users
2. The user no longer with the company
3. The user does not recall rationale

SRS

Architecture Analysis
1. Developers no longer be part of the team
2. Change may not fit in current arch/design

Architecture

Design & Implement.
Retesting

Source Code
1. Redistribute
2. Reinstall
3. Retrain

Testing

Release

Maintenance

9

Software Requirements and Specifications

Problem Domain/Environment   Solution Domain/System

D: Domain Knowledge   **S:** Specifications   **P:** Program

R: Requirements   **M:** Machine

- **D:** Relevant Facts about the environment that are assumed true whether or not we ever build the system
- **R:** Things that we wish to be made true by delivering the proposed system
- **S:** A description of the behaviours the program must have in order to meet these requirements **R**

10

Analyzing the Problem

- You need a good and clear understanding of the problem that a new software system is intended to solve.
- Problem Analysis
  - The process of understanding real-world problems and user needs, and proposing solutions to meet those needs.

Acknowledgment
Use case studies and figures are from Dean Leffingwell & Don Widrig: Managing Software Requirements: A Use Case Approach, Pearson, 2003

11

Problem Analysis

- A problem
  - Understanding the difference between *things* as perceived and *things* as derived
- Analyst addressing a problem
  - Changing the user's *desire* or *perception* may be the most cost-effective approach
    - Setting and managing expectations
    - Providing workarounds or incremental improvements to existing systems
    - Providing alternative solutions that do not require new system development
    - Providing additional training

12

Problem Analysis Steps

1) Gain agreement on the problem definition
2) Understand the root causes—the problem behind the problem (i.e., understand *why*)
3) Identify stakeholders and users
4) Define the solution system boundary
5) Identify the constraints to be imposed on the solution

13

Gain Agreement on the Problem Definition

- Simply write the problem down and see if *everyone* agrees

Element	Description
The problem of ..	Describe the problem.
Affects…	Identify stakeholders affected by the problem.
And results in ..	Describe the impact of this problem in stakeholders and business activities.
Benefits of a solution ..	Indicate the proposed solution and list a few key benefits

14

Possible Problems

- Lack of profitability
- Cost of nonconformance
  - Cost of producing waste/scrap
- Too much scrap/waste
  - Production line errors?
- Inaccurate sales orders
  - Root causes

15

Understand the Root Causes using Fishbone Diagrams

16

Addressing the Root Cause

- Which problems would you fix?

17

Problem Definition

Element	Description
The problem of ..	Inaccuracies in sales orders
Affects…	sales order personnel, customers, shipping manufacturing, customer service
And results in ..	increased scrap, excessive handling costs, customer dissatisfaction and decreased profitability
Benefits of a solution ..	Increased accuracy of sales orders at point of entry, Improved reporting of sales data to management and, ultimately, higher profitability

18

Identify Stakeholders and Users

- A good understanding of the stakeholders is an important factor in developing an effective solution
- **Stakeholders:** anyone who could be materially affected by the implementation of the new system or application
- Stakeholders are the people needed to ensure the success of a project
- Stakeholders may contribute money, effort and must see the benefits of the project—otherwise they won't support the project and may obstruct it!!

19

Stakeholder Analysis

- Who are the stakeholders?
- What goals do they see for the system?
- Why would they contribute?
- How does the project impact their jobs?
- What costs/risks do they envision?
- What solutions, suppliers, and resources do they have in mind?
  - Replace vs. extend old system?
  - Buy vs. build?

20

Example of Stakeholders

- The sponsor who pays the product
  - Wants to see his money put to good use
  - Return on Investment (ROI)
- Daily users
- Customers
- Business partners
- Authorities
  - Safety inspectors, auditors, government
- IT and support staff
- Process gurus

21

Questions to assist in Identifying Stakeholders

- Who are the users of the system?
- Who is the customer (person who pays) for the system?
- Who else is affected by the outputs generated by the system?
- Are there internal or external users (e.g., agencies) of the system whose needs must be addressed?
- Who will maintain the new system?
- Is there anyone who should/would care?

22

Example of Stakeholders

- Users
  - Sales order entry clerks
  - Sales order supervisor
  - Production control
  - Billing clerk
- Other stakeholders
  - MIS director and development team
  - Chief Financial Officer
  - Production manager

23

Define the Extent of the Solution System

- System extent defines the border between the solution and the real world that surrounds the solution
  - Where does our system begin and end?
  - Identify actors (i.e., human or other systems) interacting with our system

24

Identifying Actors

- Who will supply, use, remove information from the system?
- Who will operate the system?
- Who will perform system maintenance?
- Where will the system be used?
- Where does the system get its information?
- What other external systems will interact with the system?

Identify actors for your project!

25

System Perspective Diagram

- **System perspective** is a block diagram that describes the boundaries of the system, its users, and other interfaces

26

Identify the Constraints to be Imposed on the Solution

- Constraints
  - Restrictions on the degree of freedom in providing a solution
  - Can severely affect our ability to deliver an envisioned solution — **ensure rationale is recorded**
- Economic constraints
  - What financial or budgetary constraints are applicable?
  - Are there costs of goods sold or any product pricing considerations?
  - Are there any licensing issues?
  - Will there be any hardware maintenance contracts (e.g., Victoria class subs)?
  - Very relevant for hardware: costs of prototypes

27

Example Constraints

Source	Constraint	Rationale
Operational	An exact copy of sales order data must remain on the legacy database for up to one year.	The risk of data loss is too great; we will need to run in parallel for up to one year.
Systems and OS	The applications footprint on the server must be less than 20 megabytes.	We have limited server memory available.
Equipment budget	The system must be developed on existing server and host; new client hardware for users may be provided.	Cost control and maintenance of existing systems.
Personnel budget	Fixed staffing resource; no outsourcing.	Fixed operating costs as per the current budget.
Technology mandate	New OO methodology to be used.	We believe that this technology will increase productivity and increase reliability of the software.

28

More Constraints

- Political constraints
  - Are there internal/external political issues that affect potential solutions?
  - Interdepartmental problems or issues?
  - Example: Encryption software (should support weak encryption)
- Environmental constraints
  - Are there environmental or regulatory constraints? Legal?
  - Security requirements?
  - What other standards might we be restricted by?

29

Other Constraints

- Technical Constraints
  - Are we restricted in our choice of technologies?
  - Are we constrained to work within existing platforms or technologies?
  - Are we prohibited from any new technologies?
  - Are we to use any purchased software packages?
  - Example: Sun Microsystems consultants using msft solution or vice versa
- System Constraints
  - Is the solution to be built on our existing systems?
  - Must we maintain compatibility with existing solutions?
  - What operating systems and environments must be supported?
- Schedule and Resources Constraints
  - Is the schedule defined? (e.g., must deliver by Christmas/shopping season)
  - Are we restricted to existing resources?
  - Can we use outside labor? (e.g., outsource to India)
  - Can we expand resources? Temporary? Permanent?

30