CSC290 Communication Skills for Computer Scientists

Lisa Zhang

Lecture 3; September 23, 2019

Critical Review Paper Choice

- TUT0101: Paper 1 (App Review)
- TUT0102: Paper 1 (App Review)
- TUT0103: Paper 2 (Education)
- TUT0104: Paper 1 (App Review)
- TUT0105: Paper 1 (App Review)
- TUT0106: Paper 1 (App Review)

Due date: October 6, 2019

Project Managment

Project Management in a Communications Skills Course?

Project management = proper communication to make sure a project is successful.

- > You will be a part of software projects.
- > You will work with other developers.
- > You will work with project managers.
- > You may even choose to become a project manager.

Your Group Project

To understand the aspects of communication in software projects, you will be working in a group to write a small piece of software.

Project Management

A Project Manager (PM) is a professional who plans, procures, and manages the execution of projects

- "Governing body" = Project Management Institute https://www.pmi.org/
- Project Management Body of Knowledge (PMBOK)
- Certification: Project Management Professional (PMP) and many others

We'll talk about project management from a software perspective.

Project management is hard!

According to a report by The Standish group:

- About 30% of software projects will be canceled before they're completed.
- Over 50% of software projects will run over budget by nearly twice as much as originally budgeted.

https://www.projectsmart.co.uk/white-papers/chaos-report.pdf

Why do software projects fail so often?

- Unrealistic or unarticulated project goals
- Inaccurate estimates of needed resources
- Badly defined system requirements
- Poor reporting of the project's status
- Unmanaged risks
- Poor communication among customers, developers, and users
- ... and more

https://spectrum.ieee.org/computing/software/why-software-fails

Today's focus

Requirement Analysis & Software Design

- Determining what to build
- Communicating what to build
- Determining how to build it
- Communicating how to build it

Requirement Analysis

Why Analyze Requirements?





How the customer explained it

How the project leader understood it



How the engineer designed it



How the programmer wrote it



How the sales executive described it

Requirement Analysis Goals

- Identify stakeholders
- Elicit requirements
- Determine deliverables
- Determine the scope of the project
- Estimate time and resource use
- Choose milestones
- Identify risks

Stakeholder

Person or party with a interest in the project/process; can affect or be affected by project/process.

Stakeholders help define their requirements of the project.

Worksheet: Identify stakeholders for the projects in both Side A and Side B.

Non-Stakeholders

- The developers and competitors are **not** stakeholders!
- They are not the users of the resulting product or service, and do not help define the requirements of the project.
- The maintainers of a project can be a stakeholder.

Requirement Elicitation

- "Elicit" defined as:
 - Draw forth or bring out (something latent or potential)
 - Call forth or draw out (as information or a response)
- Engage stakeholders actively to define requirements that are complete, clear, correct, and consistent

Software Requirements

- Functional requirements: defines the *behaviour* of the software
- Non-functional requirements: judges the operation of a system
 - Performance
 - Reliability
 - Availability
 - Security
 - Maintainability
 - Portability

Worksheet: Identify requirements (and questions about requirements) for the projects in both Side A and Side B. Write down at least one non-functional requirement

What's In Scope? What's Out of Scope?

Features that are **in scope** are part of the project.

Features that are **out of scope** are explicitly excluded from the project.

It is important to agree ahead of time what features will be in scope and out of scope.

Worksheet: Identify tasks that will be out-of-scope, or tasks that you should clarify is in- or out-of-scope.

Deliverable: A tangible or intangible product or service produced.

- The final deliverable is usually known in advance.
- Interim deliverables help keep a project on task

Worksheet: What are some potential interim deliverables for the projects on the worksheet?

A **milestone** is a significant checkpoint in project timeline.

The nature of the project might change before and after a milestone.

Worksheet: What are some potential milestones for the projects on the worksheet?

Identify Risks

- Anticipate possible problems that may occur
- Plan for possible responses to such problems

Example: In your group project, one potential risk is a team member dropping the course.

Estimating time and effort

In order to estimate the resources required for the project, we break down the project into parts.

We build a **work breakdown structure**, which organizes the team's work into manageable tasks.

Work Breakdown Structure

Hierarchical description of all of the tasks in the project. Useful for planning and reporting the status of the project.

WBS Task

Each task should...

- have a defined start and end, and associated deliverable.
- be doable without interruption (e.g. waiting for another task).
- be small enough so you can estimate its time and cost.

You should be able to identify the task status at any point.

Why?

- It is easier to estimate how long small tasks will take.
- It is easy to underestimate how long large tasks can take.

Gantt Chart

A Gantt Chart also takes into account task dependencies.



Figure 1: from http://executivepropmgmt.co/

Project Documents

- Project Proposal: to gain approval of stakeholders
- Project Charter: agreement with stakeholders
- Project Plan: to align people working on the project

They each have different goals/audiences, but are similar in that they list the project goals, stakeholders, scope, deliverables, resources, milestones, and risks.

Design

Why Design?



Much easier to fix issues during the design phase.

Software Design

- What should the software do?
- What are the components of the program?
- How will you represent those components?
- How will the different components interact with one another?

How do you communicate the software design clearly?

Somewhat standard tool for communicating software design.

Our focus:

- Use case diagrams
- Class diagrams

Use Case Diagram

Describes the different types of users called **actors** and the **actions** that the users can take in a system.



Use Case Diagram Actions

- Actions are approximately the same level of granularity.
- Each action begins with a verb (e.g. "make move" as opposed to "piece movement")
- Each action is specific (e.g. "make move" as opposed to "change board")
- Actions are written concisely.

What is wrong with this Diagram? (1 of 3)



What is wrong with this Diagram? (2 of 3)





.



What is wrong with this Diagram? (3 of 3)



Class Diagram

Describe the classes the software system.



What are the different components? How do they interact with each other?

Visual/Interaction Design: Wireframe

Visual guide that represents the skeletal framework of software. Describe how to arrange the functional elements, and how users can interact with them.

	Book a Flight		2		E Book a flight					
1. Seven	2. Flight	3. Extens	4. Pay	-	<u>1: Sea</u>	sch	2: Flights	3: Extros	4: Paymer	st
Your Flights					Your flights Review your selection					
10:00 Ams -	> 14:00 ME	ć	1500		→	10:40 Mon 9 Jon * Amsterdam (AMS)	14:05 Man 9 Jan New York (JFK)	3 x checked bog Eig	Business Closs €1,563	
					+	17:55 Fil 13 Jon * Amsterdam (AMS)	06:55 +1 Fri 13 Jon New York (JFK)	3 x checked bog Elig	Business Class	
14:00 18:00					Tickets are changeable and refundable for tree Datab					
NyE-	ANS	€ 150	~		Save and share			Total price		1
					Add tr	p to wishist	,	Price specificatio	0	
Sure & Share Price E3000					Shore trip details			Taxes, sucharges and sucharge may apply analogety. Tax will see payment method.	Takes, such anges and booking fee of (10 included (creat) card such age may apply), hicks shown may vary depending on fare analogity. Top will see the exact total price when selecting your poyment method.	
	~ ~				Follow	the price of this trip	:	Take an option We will guaran 9 November	n on this flight for (10 thee this price until Wednesday	>
Barl		Option E3000	Pays	-	Back			Total price	€3,081 Continue	

Teams often review the design of the software, before writing a single line of code.

Goal: Sanity check and improve the design of the software.

Your first presentation will be a design review presentation.

Grammar Break

The error message was confusing, users could not understand it.

The error message was confusing, users could not understand it.

Comma splice: using a comma to separate two independent clauses.

The error message was confusing, users could not understand it.

Comma splice: using a comma to separate two independent clauses.

Fixes:

- The error message was confusing. Users could not understand it.
- ► The error message was confusing; users could not understand it.
- The error message was confusing, and users could not understand it.

Since other factors play a role in hiring decisions having only a single set of skills is not enough.

Since other factors play a role in hiring decisions having only a single set of skills is not enough.

Need a comma to separate the independent & dependent clauses.

Since other factors play a role in hiring decisions having only a single set of skills is not enough.

Need a comma to separate the independent & dependent clauses. Fixes:

- Since other factors play a role in hiring decisions, having only a single set of skills is not enough.
- Other factors play a role in hiring decisions. Having only a single set of skills is not enough.

Semicolon (;): separates two independent clauses (two sentences) that are related.

Colon (:): gives more information about something, or start a list or a definition.

Exercise: Add punctuations

- Since other factors play a role in hiring decisions having only a single set of skills is not enough
- The "Structure and Interpretation of Computer Programs" a book by Sussman and Abelson is a great text
- The best writers are those who read a lot and write a lot
- Life is like a puzzle half the fun is in trying to work it out
- We should use first-party cookies which are cookies created by websites a user directly visits
- When I first ran the code it didn't work when I fixed the bug there were ten more I was fixing bugs for days

Software Design Example

The Company of Myself



- http://mypuzzle.org/company-of-myself
- https://www.youtube.com/watch?v=tI0RfSn8oYg

How to Design This Game?

Functional Requirements:

- What can the player(s) do?
- What are the game mechanics?
- Non-Functional Requirements:
 - Will the game lag?
- Visual/Interface Design:
 - How will the game look?
- Software Design:
 - What classes will we need?
 - What is the inheritance structure?

Complete this use case diagram

User

Ghost

What classes would you need?

There are many decisions to be made about how to structure the classes!

A class diagram provides an overall picture of the software design.

A class diagram does **not** answer all important design questions.

Data Representation

- How will you represent the board?
- How will you represent the walls and floors?
 - What about walls and floors with complex shapes?
- At what point will a character run into the wall?
- What order will you render things?

Computing Movement

At every "tick", we compute the character movements:

- Are there any key presses?
- Is the level complete?
- Do the ghosts move?
- Does the player move?

Other questions we have to answer:

- How do we store the player movement?
- Can characters move at the same time as flipping a switch?

Ordering

- Check if the level is complete (player or a ghost is at the door)
- Check for any key presses
- Check for "reset" (make ghost)
 - Create a new ghost
 - Reset tick counter and ghosts to an initial state
- Check if user wants to restart the level
- Check if a character is flipping a switch
- Move each character (what order?)
 - Is there momentum from a previous jump?
 - What if a character hits the ceiling?
 - Is there a new jump? Can a player jump?
 - Move left or right?
- Save the key press

Should the **Player** class track movement, or the **Board** class?

Do we save the **key presses** or the **pixel-wise movement** at each tick?

These are hard design decisions that can have ramifications on what kind of levels we can design.

Communication and Critical Thinking

- Making these design decisions is often an open-ended problem requiring critical thinking.
- Determining the best decision requires clearly explaining the advantages and disadvantages that you notice.
- Getting everyone on the same page can be difficult, and requires interpersonal communication skills.
- Catching issues during the design phase is much better than catching them later on!

Splitting the Work

You should be able to split the work reasonably before even writing a single line of code!

- Board
 - Rendering
 - Check if there is a barrier
 - Make ghost
 - Restart
- Character
 - Moving left and right
 - Jumping (momentum)
 - Falling (with momentum?)
- Ghost
 - Creating a Ghost from movements
- Player
 - Recording movement

Try to make the game playable as quickly as possible.

For example, implementing switches and removable barriers can come after movement.

Breakdown and Design

- The better your group communicate the design, the easier it is to work on different portions in parallel.
- Communicate a lot, especially at the start of the project! Especially if you are making decisions that will impact other people.
- If you are waiting for each other a lot, then you have not communicated the design well enough.

What to do if you make the wrong design decision?

- Try to catch and fix issues early.
- But, changes can affect multiple parts of the code.
- Keep all group members in the loop.
- Decide what process is to change the design while keeping everyone in the loop.