What it’s all about…

(overview of software engineering processes)

Our usual emphasis..

- Programming

Common sense questions (going backwards from programming)

- What is the design on the basis of which programmers do their work?
- What is the architecture of the system used to design it?
- What should the software be doing, so that we know which architecture (& design & code) to use, and what is more important among many possible objectives?

Common sense questions (going forwards from programming)

- How do we evaluate the code?
  - Testing
  - Verification
  - Validation
  - Computational effort (time & space)
  - Deployment considerations
  - Usability
  - Security & privacy concerns
  - If you are paying attention, raise your hand
- What happens after the code is written and deployed?

Processes for software engineering: *Sophie's World* (and more)

- Sequential processes
  - "Waterfall" https://www.youtube.com/watch?v=Y_A0E1ToC_I

    [We'll be covering the early phases of this in CIS453]
- Cyclic (iterative) processes
  - "Spiral model" https://www.youtube.com/watch?v=mp22SDTnsQQ
- Parallel processes
  - "Agile" https://www.youtube.com/watch?v=Z9QbYZh1YXY

    [We'll be covering this in CIS454]

Quick question for the class

- What are the first things we must do when we build a new software product?

Re. Quick question for the class

What are the first things we must do when we build a new software product?

- Learn what problem the software is intended to address.
- Formulate what it means to be successful.
- Limit the scope of the software.

- Identify constraints (hardware, people, time, cost).
- Identify the stakeholders, and their primary concerns.
- Plan the software development process.

Project Initiation, Planning and Scheduling

1. Most projects begin with a short proposal summarizing:
   - what needs to be done,
   - why,
   - the scope of the project, as well as
   - an estimate of the resources needed.
2. Then more details are sketched out, including milestones and timeline (e.g., using Gantt charts).
3. Resource approval (people, time, hardware, software resources) must then be obtained from decision-makers (e.g., company executives).

What are the main components of the software product?

- High level perspective: What is the software intended to accomplish? If it is the component of a larger system, what does the system do?
- Requirements Specification: Details of what the software accomplishes.
- Architecture: High level description of how the software is structured.
- Design: Details of how various parts of the software and how they are connected.
- Code (Implementation)
- Test cases and results
- Future plans (enhancements, needed changes, maintenance plan)

CMU-SEI's Capabilities Maturity Model (CMM) [focus on process of software development]

Five levels along which software development organizations are evaluated:

1. *Initial* - chaotic, ad hoc.
2. *Repeatable* - the process is documented sufficiently to facilitate repeating the same steps.
3. *Defined* - the process is defined/confirmed as a standard business process
4. *Capable* - the process is quantitatively managed in accordance with agreed-upon metrics.
5. *Efficient* - process management includes deliberate process optimization/improvement.

Stakeholders: who is affected by what we do?
(E.g., for developing a mobile app software)

- Direct Stakeholders?
- Indirect Stakeholders?

Specification: what needs to be done

Look at each stakeholder's perspective:

- The ultimate users of the software (possibly different kinds of users)
- Designers
- Coders
- Managers of the software engineering team
- Sales & Marketing people in your company
- Executives concerned with costs & finances
- Human Resource & Staffing Managers
- "The general public" who may be impacted by the use of the software (e.g., patients affected by glitches in hospital accounting software)

How would we evaluate a software specification? (Discriminating Good vs. Bad Specification)

- Clear?
- Comprehensive?
- Consistent?
- Company standards?

How do we express a specification (or design)?

- English

- Formal logic

- Various diagrams that follow a well-understood format
  - “Structured Analysis & Structured Design”
  - “Unified Modeling Language” (UML)

    https://www.youtube.com/watch?v=8CBnAmYnwk0

    https://www.youtube.com/watch?v=vgYKW9O6fFE

    https://www.youtube.com/watch?v=FkRwbVUVFvE

Gantt Charts

- Describe the time duration of each task and subtask, accounting for dependencies between them.

- Example: https://www.teamgantt.com/blog/gantt-chart-example

- Many tools exist, e.g., at www.smartsheet.com

Class Exercise 1

A mobile app has to be developed to be used by customers of a hardware store chain (e.g., Home Depot, Lowe's, Truvalue).

You are the manager of a software engineering team, which has developed a software specification for this system.

What are the criteria you would use to decide whether it is acceptable, so that the Design can start?

[Work with someone sitting next to you; both of you must submit on Blackboard, clearly indicating with whom you worked.]