"Scrum is great!"
Activity 1: Software Processes

Why?

Every software development effort uses some process. We consider several of the most important models of software processes that have been proposed.

Learning Objectives

- Understand what a software process model is.
- Understand several important software process models.
- Understand the need for and procedure of evaluating software processes.

Success Criteria

- Be able to describe the waterfall, prototype evolution, spiral, and agile process models.
- Be able to list the advantages and disadvantages of these models.

Resources

Software Processes (pdf from Canvas)

Exercises

1. What is the difference between a software process and a software lifecycle process?

A software process: a process used **in the creation or support** of a software product.
A software lifecycle process: a process depicting the steps that occur from software product **inception through retirement** from service. In other words, it is a process that **contains all the steps in the life** of a software product.

2. What is a prototype?

A prototype is a working model of some or all of a finished product. For example, programmers can use prototype to test out program design to make changes and create good design choices.

3. What is rework?

Rework is discarding or redoing previous work products. In other words, rework is also a source of expense and delay in developing process. It is advisable to eliminate rework during any processes.

Problem

1. Process models can be prescriptive or descriptive. If some organization sets out to use a particular software process in a project, is the process model descriptive or prescriptive?
   I think the process model is prescriptive because prescriptive is like a guideline on how to organize and shape the activities of software development process. For example, how the project should be done and in what ways.

2. Which of the waterfall, prototype evolution, spiral, and agile processes are iterative? Which are incremental?

Type	Models
Iterative	Spiral, Prototype evolution, Waterfall
Incremental	Agile, Prototype evolution

3. Which of the waterfall, prototype evolution, spiral, and agile processes are lightweight and which are heavyweight?

Type	Models
Lightweight	Agile, Prototype Evolution
Heavyweight	Waterfall, Spiral

4. The waterfall model is present in some sense in all subsequent software process models. Describe the role that the waterfall model plays in the prototype evolution, spiral, and agile models.

Models	The role of waterfall model
The prototype evolution	The waterfall model exists in each iteration
Spiral	The waterfall model exists in both iteration and the whole process
Agile	The waterfall model exists in each iteration

5. List the advantages and disadvantages of each model in the table below.

Models	Advantages	Disadvantages
Waterfall	+) Everything is predictable (due to entirely plan). This helps managers to know a big project's scope, delivery date, and budget up-front to make a good decision. +) It is easy to tell whether a project is on-time and on-budget by monitoring activity and comparing it to the plan. Managers can take appropriate action to fix problems, modify schedules, adjust plans. +) Problems can be found and corrected cheaply. For instance, if a product requirement is incorrect, but it is not detected, the cost to fix will be very expensive. +) The waterfall model emphasizes production of complete and correct documentation at every stage. +) The waterfall model divides development work into distinct and relatively independent phases that are suitable for independent teams.	+) The model relies heavily on being able to produce complete and correct product specifications that do not change appreciably during the project. +) Even when the requirements are stable, it is almost impossible to make them complete and correct. +) Producing and maintaining all the documentation for the waterfall model is expensive. +) Passing a product from team A to team B during development means that every team must study all the documentation to understand what needs to be done. +) The model uses too many people in large teams who must coordinate their activities using a lot of documentation in long time. +) The model does not deliver a product until completion of the development project, which may take several years.
The prototype evolution	+) Changes to product requirements are easy to handle, and customers are more likely to get what they want. +) Customers can get useful software in a short amount of time. +) Generally, there is not a lot of documentation or management oversight required.	+) It is very hard to predict when an adequate product will be finished and how much it will cost. +) The product may be unmaintainable that is due to bad design because the product may have evolved chaotically. +) The quality control of the product may be lax, resulting in an unreliable or very buggy product.

Models	Advantages	Disadvantages
Spiral	+) The model is important because it demonstrated the central place of risk management in software development. +) The model emphasizes the importance of tailoring development practices to the project at hand. For example, what should be done and what should not be done. +) If a risk is likely happened, then work should be done to remove or reduce the risk. For example, if a user interface (GUI) is important as a success component of a product, then considerable effort should be put into its design.	+) The Spiral Model is focused and based on risk management, but not very many people are trained in and good at risk management. +) The model is very general and adaptable, and so it requires expertise in tailoring software processes that is not very common.
Agile	+) Product specifications can be changed frequently without affecting the process. +) A version of the product is delivered to customers very soon after development begins, and new versions can be delivered as well. +) Bad projects are recognized and cancelled early. +) Agile process is lightweight. Therefore, it can save time and money. +) Reducing waste and duplication of effort.	+) Customers must be involved throughout development but it is often difficult to get customers to commit their time and effort to the product development. +) The designs may not be very good, degrading product quality and increasing development effort. +) Agile processes are difficult to use on large projects because it is difficult to coordinate the activities of many teams. +) Sometimes, it is harder to predict the outcomes of agile projects.