

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?

Answer: A software process is a process used to create or support a software product, whereas a software lifecycle process is a process that shows all the steps in the life of a software product.

2. What is a prototype?

Answer: A working model of some or all of a finished product.

3. What is rework?

Answer: The process of discarding or redoing previous work products.

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?

Answer: Prescriptive

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

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

Type	Models
Lightweight	Prototype evolution, agile
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	In this case, the waterfall model could use prototype evolution to help overcome some of its problems.
Spiral	Developers could decide to pursue a phase of the waterfall model within the spiral.
Agile	The waterfall model led to the creation of Agile as the waterfall model is not very flexible and rather time-consuming. It can be used to make goals clear and emphasize documentation and to help organize larger projects

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

Models	Advantages	Disadvantages
Waterfall	<ul style="list-style-type: none"> - The software product is specified and planned early, so everything is predictable. - It is easy to tell whether a project is on-time and on-budget by monitoring activity. - Problems can (in principle) be found and corrected cheaply. - The waterfall model emphasizes production of complete and correct documentation. - The waterfall model divides development work into independent phases that can be performed by independent teams. 	<ul style="list-style-type: none"> - The waterfall model relies on being able to produce product specifications that do not change appreciably during the project. - Even when requirements are stable, it is almost impossible to make them complete and correct. - Producing and maintaining all the documentation needed for the waterfall model is expensive. - Passing a product from team to team of specialists during development means that every team must study all the project documentation. - The waterfall model typically uses many people in large teams who must coordinate their activities using a lot of documentation over a long period of time. - The waterfall model does not deliver a product until completion of the development project, which may take years.
The prototype evolution	<ul style="list-style-type: none"> - Changes to product specifications are easy to handle. - Customers are more likely to get what they want. - Customers can get useful software very quickly. - There is typically not a lot of documentation or management oversight required. 	<ul style="list-style-type: none"> - It is very hard to predict when an adequate product will be finished and how much it will cost. - The design of the finished product may be very bad because it may have evolved chaotically. - The product may be unmaintainable. - There is little discipline in this process, so quality control may be lax. - It may result in an unreliable or very buggy product.

Models	Advantages	Disadvantages
Spiral	<ul style="list-style-type: none"> - Explicit incorporation of increases in fidelity and detail. - Explicit incorporation of risk management into several life cycle process models. - Emphasizes the importance of tailoring development practices to the project at hand, and how to decide what should be done and what should not be done. 	<ul style="list-style-type: none"> - It is focused on risk management, but not very many people are trained in and good at risk management. - It is general and adaptable, and so it demands expertise in tailoring software processes that are not very common.
Agile	<ul style="list-style-type: none"> - Product specifications can be changed frequently. - A version of the product is delivered to customers soon after development begins. - New versions with gradually increasing capabilities can be delivered frequently, if desired. - Bad projects can be recognized and cancelled early. - The process is lightweight, so a lot of time and effort is saved. - Waste and duplication of effort are usually greatly reduced. 	<ul style="list-style-type: none"> - Customers and users must be involved throughout development, but it is often difficult to get customers to commit so much time and effort. - Designs developed incrementally may not be very good, degrading product quality and increasing development effort. - Agile processes are difficult to use on large projects because it is hard to coordinate the activities of many teams that are evolving parts of the product in unpredictable directions. - Minimal documentation is used. - It is harder to predict the outcomes of agile projects.