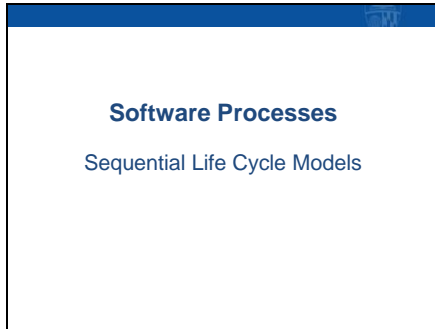
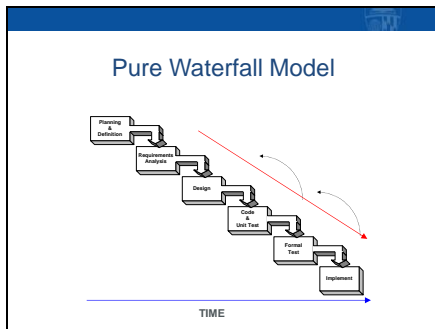


1



In this lecture we'll discuss some examples of sequential life cycle models.

2



The grandfather of sequential project life cycles is the pure waterfall model. There are numerous variations on this model, but I'll introduce this one that will serve as the basis for all the others.

In the pure waterfall model, activities are performed essentially in sequence over time...starting with problem definition and planning...and culminating with the delivery of a software product at the end of the entire process.

There are typically one or more project artifacts produced at each life cycle phase. For example, a requirements document is typically produced at the end of the requirements phase, one or more design documents are typically produced during the design phase, test documentation...such as a test plan and test procedures are produced during the testing phase, and so forth.

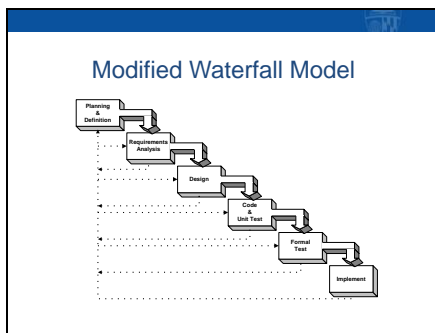
Many projects will also hold reviews at the end of each phase to determine whether it is ready to move to the next phase. If it's ready to advance it does. If it's not ready to advance, then some or all of the activities of that phase are repeated until the project is ready to advance.

At least that's the way it is supposed to work. In practice, many projects continue to the next phase even though there are issues that have not been resolved with the

current phase artifact...which ultimately increases the required rework...and can increase the project cost and schedule...because the project team must repeat the activities in multiple phases.

Now, the waterfall model gets its name from two things. First, from the sequential nature of the project activities. The requirements are all done, then the design is all done, etcetera. And, second, from the project momentum that builds as the project progresses from one phase to another. The further downstream a project is...the more difficult and expensive it is to revisit earlier phases to make changes and corrections...kind of like salmon trying to swim upstream.

3



The modified waterfall model is just like the pure waterfall model, but there is a focus on providing and enforcing project checkpoints at every phase in the project...with the understanding that issues uncovered at any checkpoint may require repeating the work activities of more than just the current project phase. For example, issues uncovered in preliminary design may require not only re-doing some of the preliminary design activities, but perhaps going back and repeating some of the activities in the requirements phase.

The idea behind this is to try and mitigate some of the momentum problems of the pure waterfall model and make it easier to perform corrections and changes earlier and more frequently in an effort to reduce the overall cost of rework.

4

Life Cycle Model Capability	Pure Waterfall	Modified Waterfall
Works with poorly understood requirements	P	F-E
Produces highly reliable system	E	E
Produces expandable system	E	E
Manages risks	P	F
Can be constrained to pre-defined schedule	F	F
Has low overhead	P	E
Allows for mid-course corrections	P	F
Provides customer with progress visibility	P	F
Provides management with progress visibility	F	F-E
Requires little manager/developer sophistication	F	P-F

Adapted from S. McConnell, Rapid Development, Microsoft Press, 1997

Here's a kind of scorecard that rates the pure waterfall and modified waterfall models based on ten life cycle model capability criteria...with relative scores ranging from poor to excellent.

The pure waterfall model can be used on projects of all sizes and complexities. It works well when customers know what they want and will make commitments. It also works well when estimates are updated and commitments are renewed one or more times in every project phase. Since this is often not the case in practice, adopting a modified waterfall in which more frequent changes can be accommodated can be more beneficial.

Other disadvantages of the waterfall style life cycles are that the requirements must be fully-specified up front, project momentum (in the pure waterfall) can make repeating phases difficult, project visibility and control may be poor without careful selection of project milestones, and...the product is not delivered until the end...which can significantly limit stakeholder vision of what the final product will look like.