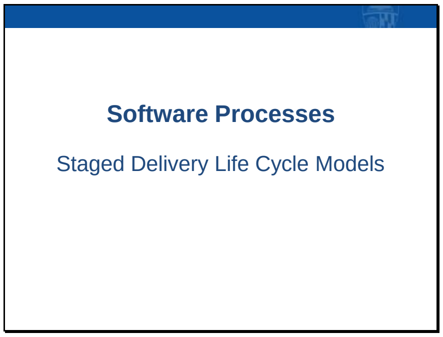
1

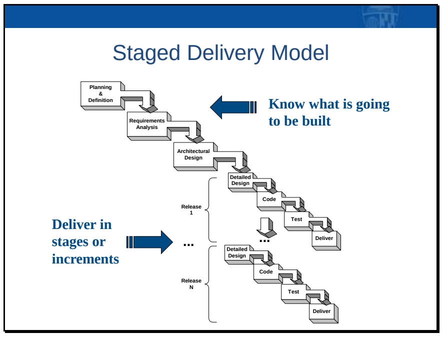


Software Processes

Staged Delivery Life Cycle Models

In this lecture we’ll discuss some examples of staged delivery life cycle models.

2



Staged Delivery Model

Know what is going to be built

Deliver in stages or increments

In the last lecture, we discussed waterfall-style life cycle models. The key distinguishing characteristic of waterfall models is that the product is not delivered until the end of the life cycle process.

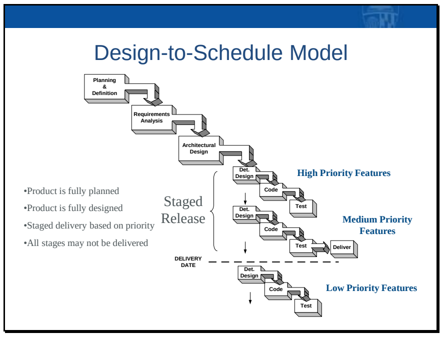
In comparison, with a staged delivery life cycle model the product is delivered in stages…sometimes called increments. It starts out like the waterfall model in that the product is defined, the requirements are specified, and an overall architectural design is created. But…from that point onward…the functionality of the product is delivered in stages. Each stage delivers a subset of overall product functionality, and within each stage detailed design, coding, testing, and delivery is performed.

The primary advantage of staged delivery is that provides some useful functionality to the customer earlier than in a waterfall approach. If properly planned, the project may also be able to deliver the most important functionality earlier…and if the project runs into schedule or budget obstacles…at least the customer has a product with the most useful features.

Staged delivery also provides more tangible measures of progress earlier in the project.

The primary challenges to the staged delivery approach are that the project must have very careful planning and ongoing management, and successive deliveries must be designed so that there are minimal dependencies.

3



Design-to-Schedule Model

- Product is fully planned
- Product is fully designed
- Staged delivery based on priority
- All stages may not be delivered

The design-to-schedule life cycle model is a variation on the staged delivery model. It is similar in that the definition, requirements, and architectural phases are completed, and the product is planned to be delivered in stages.

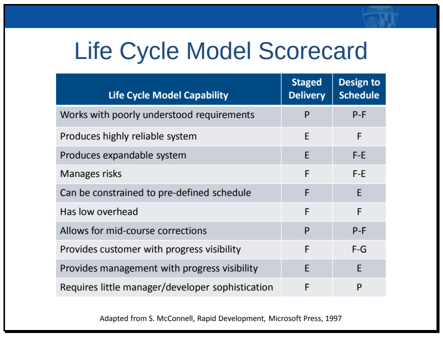
In this model, the product features are typically prioritized based on some measure of importance, with the most important features being implemented first.

The key difference between this model and the staged model, however, is that in this model we don’t know how many releases will actually be delivered…because there is a real, unmovable deadline for product delivery. We might have four releases planned, but only three may be actually delivered because of the deadline.

This model requires the same careful planning and management attention as the staged delivery model, and also requires that the product features be prioritized.

The primary disadvantage of this model is that if we don’t finish all the releases by the deadline then some of the effort spent in the definition, requirements, and architectural design phases will have been wasted…and that time might have been able to be spent on implementing a few more features.

4



Life Cycle Model Scorecard

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

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

Here’s a scorecard for the staged delivery and design-to-schedule life cycle models. As you can see, neither of these models work well with poorly understood requirements, and they don’t adapt well for mid-course changes, but customer and management visibility is better than with waterfall-style models. And..these processes do require more management attention and sophistication compared to waterfall models.