STAGED DELIVERY

Staged delivery is a lifecycle model in which software is developed in stages, usually with the most important capabilities being developed first.

Main Benefits	Increased visibility of progress and support for more frequent and predictable product releases.
Keys to Success	Ensuring the product architecture is solid and supports more than one possible product direction, defining delivery stages carefully, releasing the first stage as early as possible, releasing the stages in their order of importance, and planning to handle customer change requests.
When to Use	Staged delivery works best for well-understood systems, very large projects when your customers are eager to begin using a relatively small portion of the product's functionality, and for systems in which you can develop useful subsets of the product independently. If you can't figure out how to break the delivery of your product up into stages, staged delivery is not the right approach for you.
Main Risks	The main risk associated with staged delivery is the risk of feature creep. When customers begin to use the first release of your product they are likely to want to change what has been planned for the other releases.

Overview

In staged delivery the most important capabilities are generally developed first. Staged delivery doesn't reduce the time needed to build a software product, but it substantially reduces the risks involved in building one. It provides tangible signs of progress that are visible to the customer and useful to management in assessing project status.

The staged delivery lifecycle progresses through the waterfall model steps of defining the software concept, analyzing requirements, and creating an architectural design. It then proceeds to do detailed design, coding, debugging, and testing within each stage, creating a releasable product at the end of each stage.

CxOne Support

CxOne provides support extensive support for staged delivery through the *SPSG Process Flow*.

Interactions with other Best Practices

Although there are a few similarities, staged delivery is not a form of prototyping. Prototyping is exploratory, and staged delivery is not. Its goal is to make progress visible or to put useful software into the customers' hands more quickly. Unlike prototyping, you know the end result when you begin the process.

If staged delivery provides less flexibility than you need, you can probably use evolutionary delivery or evolutionary prototyping instead. If you know the general nature of the system you're building but still have doubts about significant aspects of it, don't use staged delivery.

© 2000-2002 Construx Software Builders, Inc. www.construx.com

Staged delivery combines well with miniature milestones. By the time you get to each stage, you should know enough about what you're building to map the milestones out in detail.

Success at developing a set of staged deliveries depends on designing a family of programs. The more you follow that design practice, the better able you'll be to avoid disaster if the requirements turn out to be less stable than you thought.

Further Reading

McConnell, Steve. *Rapid Development*. Redmond WA: Microsoft Press. 1996.

Gilb, Tom. *Principles of Software Engineering Management*. Wokingham, England: Addison-Wesley, 1988.

CxBest_MiniatureMilestones

CxBest_EvolutionaryPrototyping

CxBest_EvolutionaryDelivery

© 2000-2002 Construx Software Builders, Inc. www.construx.com