

Spiral model

From Wikipedia, the free encyclopedia

The **spiral model** is a software development process combining elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom-up concepts. It is also also known as the spiral

Software development process

Activities and steps

Requirements | Architecture |
Implementation | Testing | Deployment

Models

Agile | Cleanroom | Iterative | RAD | RUP
| **Spiral** | Waterfall | XP

Supporting disciplines

Configuration management |
Documentation | Project management |
User experience design

lifecycle model, is a systems development method (SDM) used in information technology (IT). This model of development combines the features of the prototyping model and the waterfall model. The spiral model is intended for large, expensive, and complicated projects.

Contents

- 1 History
- 2 The Spiral Model
- 3 Applications
- 4 Advantages
- 5 See also
- 6 External links

History

The spiral model was defined by Barry Boehm in his 1988 article *A Spiral Model of Software Development and Enhancement*. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration matters. As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with the client (who may be internal) reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

The Spiral Model

DEFINITION - The spiral model, also known as the spiral lifecycle model, is a systems development method (SDM) used in information technology (IT). This model of development combines the features of the prototyping model and the waterfall model. The spiral model is intended for large, expensive, and complicated projects.

The steps in the spiral model can be generalized as follows:

1. The new system requirements are defined in as much detail as possible. This usually involves interviewing a number of users representing all the

- external or internal users and other aspects of the existing system.
2. A preliminary design is created for the new system.
3. A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
4. A second prototype is evolved by a fourfold procedure: **(1) evaluating the first prototype in terms of its strengths, weaknesses, and risks; (2) defining the requirements of the second prototype; (3) planning and designing the second prototype; (4) constructing and testing the second prototype.**
5. At the customer's option, the entire project can be aborted if the risk is deemed too great. Risk factors might involve development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer's judgment, result in a less-than-satisfactory final product.
6. The existing prototype is evaluated in the same manner as was the previous prototype, and, if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
7. The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
8. The final system is constructed, based on the refined prototype.
9. The final system is thoroughly evaluated and tested. Routine maintenance is carried out on a continuing basis to prevent large-scale failures and to minimize downtime.

Applications

For a typical shrink-wrap application, the spiral model might mean that you have a rough-cut of user elements (without the polished / pretty graphics) as an operable application, add features in phases, and, at some point, add the final graphics.

The spiral model is used most often in large projects. For smaller projects, the concept of agile software development is becoming a viable alternative. The US military has adopted the spiral model for its Future Combat Systems program.

Advantages

- Estimates (i.e. budget, schedule, etc.) get more realistic as work progresses, because important issues are discovered earlier.
- It is more able to cope with the (nearly inevitable) changes that software development generally entails.
- Software engineers (who can get restless with protracted design processes) can get their hands in and start working on a project earlier.

See also

- Barry Boehm
- Software engineering
- List of software engineering topics
- Waterfall model
- Chaos model
- MBASE
- V-model

External links

- A Spiral Model of Software Development and Enhancement (<http://www.sce.carleton.ca/faculty/ajila/4106-5006/Spiral%20Model%20Boehm.pdf>) - Barry Boehm's original article
- A graphical representation (http://www.elanman.org/teaching/gmu/swe620-infs622/Graphics/spiral_model.gif)

Retrieved from "http://en.wikipedia.org/wiki/Spiral_model"

Categories: Articles to be expanded since January 2007 | All articles to be expanded | Software development process

- This page was last modified 00:01, 3 August 2007.
- All text is available under the terms of the GNU Free Documentation License. (See **Copyrights** for details.)
Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a US-registered 501(c)(3) tax-deductible nonprofit charity.