

Modelo Espiral

Modelo espiral se baseia no modelo cascata, que possui mais recursos que não são encontrados em outros modelos, onde ele consiste em trabalhar as etapas em uma espiral, isso é feito da seguinte forma, as etapas ficam determinadas em um setor do espiral, fazendo com que cada volta no espiral seja uma nova etapa do processo, nele ainda tem a conclusão que quando o trabalho ou software é terminado, além de poder começar de novo passando pelas etapas, tal modelo se pode ser usado em quase todo tipo de projeto.

Modelo Cascata

O Modelo Cascata é um processo de software – serve como uma metodologia para as atividades, ações e tarefas necessárias para desenvolver um software de alta qualidade - que foi proposto por Royce em 1970, e completamente aceito até meados de 80. Neste modelo é sugerida uma abordagem sequencial e sistemática para o desenvolvimento de software. Assim, podemos dividi-lo nos seguintes passos:

- Levantamento de requisitos ou necessidades junto ao cliente, fase de planejamento onde são definidas as estimativas;
- Cronograma e acompanhamento;
- Modelagem, onde é feita a análise de projeto;
- Construção onde é codificado e testado;
- Implementação ou emprego onde é efetuada a entrega, suporte e feedback do software concluído.

Uma vez definido o modelo de ciclo de desenvolvimento, existem três abordagens para implementá-lo:

- Cascata pura
- Incremental;
- Evolucionária

Cascata Pura

Neste caso, o ciclo de desenvolvimento é executados em sequência, uma tarefa só inicia, após o término da anterior. Esta abordagem é adequada quando:

- Existe um conjunto de Requisitos do Usuário estáveis e de alta qualidade;
- A duração do projeto é pequena, isto é, menor do que dois anos;

- O sistema completo deve estar disponível de uma única vez.

Incremental

Neste caso, o programa é sempre entregue ao cliente pronto, ficando a critério do cliente, se precisa de mais e novas modificações, sendo que a cada versão, todo o processo é feito novamente, do início.

A abordagem incremental é adequada quando:

- a liberação do software deve estar de acordo com um conjunto de prioridades definidas nos Requisitos do Usuário;
- é necessário melhorar a eficiência da integração do software com outras partes de um sistema maior;
- é requerido antecipadamente evidências de que o produto será aceito.

Evolucionária

Neste caso, o desenvolvimento só termina quando o usuário se satisfizer, ou seja, o projeto está em constante evolução. Esta abordagem é adequada quando:

- é necessária alguma experiência do usuário para refinar e completar requisitos;
- algumas partes da implementação podem depender da existência de tecnologia ainda não disponível;
- existem requisitos do usuário não bem conhecidos; e
- alguns requisitos são muito mais difíceis de serem implementados do que outros, decidindo-se não implementá-lo para não atrasar o projeto