SPIRAL DEVELOPMENT

Quando nos referimos ao modelo em espiral na escrita e documentação de casos de uso estamos a falar do processo iterativo como os casos de uso são obtidos, analisados, documentados e validados.

A seguir apresenta-se uma figura ilustrativa deste modelo.



**Análise da figura:**

Obtenção de requisitos – Definem-se os objetivos pretendidos, faz-se a recolha dos requisitos, criam-se os casos de uso.

Análise e negociação de requisitos – Define-se o que está dentro e fora da fronteira com o cliente.

Documentação de requisitos – Criam-se os diálogos e descrição dos casos de uso.

Validação de requisitos – São mostradas aplicações do domínio do software e/ou protótipos ao cliente, como instrumentos auxiliares de validação dos requisitos e casos de uso.

**Características dos casos uso:**

O desenvolvimento de casos de uso é um processo de grande complexidade e deve portanto ser tratado do geral para o particular (**BreadthBeforeDepth**) em cada iteração a precisão do caso de uso aumenta até a sua validação final (**QuittingTime**).

Os requisitos têm a tendência a mudar á medida que vão sendo analisados e o novo conhecimento obtido acaba quase sempre por revelar novas informações acerca de outros, ausência de alguns e outros que se tornaram obsoletos ou fora de contexto.

Desenvolver casos de uso num único passo é difícil, impede a incorporação de novos dados, e dificulta a descoberta de fatores de risco.

O custo de encontrar erros no desenvolvimento de casos de uso é tanto maior quanto mais avançada for a fase de progresso do software.

O padrão **SpiralDevelopment** interagecom o padrão **BreadthBeforeDepth,** é necessário estabelecer uma metodologia que inclua a revisão dos casos de uso e criar uma estratégia para lidar com a complexidade do desenvolvimento dos casos de uso.

O padrão **SpiralDevelopment** interagecom o padrão **QuittingTime,** é necessário estabelecer critérios de paragem no ciclo, esses critérios estão definidos no padrão **QuittingTime.**

**Vantagens do modelo em espiral:**

O método iterativo em espiral permite identificar e confrontar problemas antecipadamente.

Permite poupar tempo na criação do software o que em concorrência é um fator determinante.

Erros no desenvolvimento de casos de uso é muito elevado e é tanto maior quanto mais tarde esses erros forem detetados (refiro-me ao processo de desenvolvimento de software) o modelo em espiral, por ser iterativo, permite identificar estes erros com eficiência.

Criar casos de uso, muitas vezes baseados em requisitos ambíguos e inconsistentes, sem acautelar uma estratégia de revisão dos mesmos, resulta em horas de trabalho perdidas quando os erros se revelam.

A criação de casos uso é um trabalho de equipa. Quando um elemento dedica muito tempo a criar um caso de uso, vai lutar por ele, mesmo que este seja inconsistente em relação aos outros casos de uso criados por diferentes membros da equipa. O modelo em espiral por ser iterativo permite fasear e dividir o trabalho dedicado a cada caso de uso, evitando desperdício de esforço.

O método iterativo permite um conhecimento dos casos de uso de uma forma progressiva e sustentada, do que resultam casos de uso robustos e consistentes.