

Modelo Funcional Essencial

Tem como objetivo definir *o que* o sistema deve fazer, ou seja, as funções que deve realizar para atender seus usuários.

Na análise essencial fazemos essa definição de acordo com um conjunto de princípios que nos permite escolher um modelo funcional específico entre vários possíveis, o modelo essencial.

Perspectiva histórica

A modelagem funcional de sistemas teve grande desenvolvimento a partir da criação das metodologias estruturadas de análise e projeto. Entre diferentes propostas o Diagrama de Fluxo de Dados (DFD) se tornou uma ferramenta obrigatória em todos os cursos de análise de sistemas.

Em um DFD o sistema é descrito pela composição de quatro objetos básicos:

- Agentes externos (entidades externas ou ator – no modelos orientados a objeto): interagem com o sistema.
- Processos (ou funções): caracterizam o sistema.
- Memórias (depósito de dados): contém os dados necessários para o sistema funcionar.
- Fluxos de dados: representam caminhos por onde passam os dados.

A análise estruturada se propunha tratar das questões lógicas do desenvolvimento do sistemas, em detrimento das questões físicas, que seriam tratadas na fase de projeto. Foram encontrados vários problemas na análise estruturada, entre eles: a dificuldade de manter o modelo atualizado e a possibilidade de várias pessoas fazerem modelos diferentes de um mesmo sistema.

Em 1984 conseguiram definir de forma clara um método de desenvolvimento que permite dividir, sem sombra de dúvidas, o que é essencial (lógico) do que é encarnação (físico).

Com perguntas simples permite descobrir se um requisito do sistema é verdadeiro ou falso. Esse método é uma evolução da análise estruturada e é conhecido como análise essencial.

Análise Essencial

O método que um analista empregará para o desenvolvimento de um sistema, pode ser entendido como um caminho a ser percorrido em etapas, algumas delas podendo ser desenvolvidas em paralelo, outras não. As técnicas são procedimentos parametrizados e sistemáticos, pelos quais uma tarefa é executada; é a forma de se caminhar pelo caminho escolhido.

O objetivo básico do estabelecimento de um método padronizado no desenvolvimento de sistemas é obter maior consistência no trabalho, melhor qualidade oferecida ao usuário, maior facilidade no treinamento de novos analistas, eliminação das perdas acarretadas por caminhos sem saída e, sem dúvida, melhor controle dos resultados obtidos no desenvolvimento de sistemas.

O método que revela o *estado da prática atual* é a chamada análise essencial. Na análise essencial deve-se considerar perfeito o ambiente tecnológico onde será implementado o software a ser projetado (neutralidade tecnológica). Considerar que a

memória do computador é infinita, seu tempo de resposta é instantâneo, ele não trava, não tem custo, ou seja, é infalível.

Alguns fatores de seu uso:

- a) O método mais utilizado atualmente.
- b) Princípio da abstração.
- c) Princípio da divisão.

O objetivo da análise essencial é descobrir e documentar todos os requisitos funcionais verdadeiros de um sistema e apenas esses requisitos.

O método essencial está preocupado com sistemas que funcionam sempre em resposta a algum evento fora do seu controle para o qual possamos definir uma resposta planejada.

Exemplo: vendedor de passagens de avião: podemos fazer um sistema capaz de responder as perguntas típicas como *qual o preço da passagem para São Paulo?* ou *quando sai o próximo voo para Brasília?*, porém não podemos considerar com esse método um sistema que responda a absolutamente todas as perguntas que um ser humano poderia responder, como *qual foi o resultado do último jogo do Corinthians?*.

Princípios da modelagem essencial

A idéia global do caminho a ser trilhado pelo analista de sistemas ao utilizar o método de análise são:

- Domínio do problema
- Orçamento para a complexidade
- Neutralidade tecnológica
- Tecnologia interna perfeita
- Modelo essencial mínimo
- Projeto

Domínio do problema

O primeiro momento é delimitar exatamente o que se espera do sistema a ser desenvolvido. Trata-se de estabelecer seus limites fronteiriços, exatamente o que deverá ser feito.

Exemplo: informatizar um hotel. Um hotel é um macro problema. Ele é composto de várias facetas que podem ser informatizadas, como o controle da locação de quartos, o controle financeiro (contas a pagar/receber), a folha de pagamento dos funcionários, a contabilidade do hotel, enfim, é necessário que você verifique se a expectativa de quem o contratou é realmente informatizar todas estas facetas.

Uma vez delimitado a abrangência do que deverá ser feito, o segundo passo de importância deve ser o minucioso levantamento de eventos abrangendo o conteúdo que deverá ser informatizado (levantamento de requisitos do sistema).

Todos os aspectos envolvidos no problema devem ser levantados (entrevistas, documentos, o fluxo de trabalho deve ser entendido).

Você deverá sair desta fase sendo quase um especialista sobre o assunto que deverá informatizar (eventos e dados essenciais relativos ao assunto).

Orçamento para a complexidade

Orienta a modelar um sistema que possamos compreender. Devemos manipular a complexidade do modelo de forma a manter tanto o todo como cada uma de suas partes em um nível de complexidade compatível com a inteligência humana.

- Controle do número de componentes de cada parte do modelo.

- Controle da complexidade interna de cada parte do modelo.
- Controle da complexidade da interface entre componentes.
- Manutenção da qualidade dos nomes utilizados no modelo.
- Manutenção da qualidade da representação do modelo, por exemplo, quanto à clareza dos diagramas.

Neutralidade tecnológica

Exige que um modelo essencial não inclua em nenhuma de suas partes indícios da tecnologia de implementação. É importante manter a neutralidade não só para permitir uma análise mais objetiva do verdadeiro problema do usuário, mas também para aumentar a durabilidade dessa análise.

Tecnologia interna perfeita

O sistema deve ser modelado com a suposição que a tecnologia interna ao sistema é perfeita, ou seja, todos os recursos do sistema são ideais.

- A velocidade é infinita (não há espera para conseguir um resultado).
- A memória não possui limitações (pode guardar qualquer quantidade de informação por um período indeterminado, sem nenhum atraso no tempo de busca).
- Nunca apresenta falhas ou necessidade de manutenção.

Porém, não fazemos essa suposição sobre a tecnologia externa ao sistema. Essa suposição só é feita na fase de análise, sendo esquecida na fase de projeto, onde velocidade, tamanho de memória e gerência de riscos passam a fazer parte de nossas preocupações, junto com outras características, como controle de acesso (segurança).

Modelo essencial mínimo

O princípio do modelo essencial mínimo exige que, entre dois modelos possivelmente essenciais, a definição menos complicada é o modelo essencial.

Modelo ambiental

Define qual a relação do sistema a ser desenvolvido com o ambiente no qual ele está inserido. Descreve qual é ou quais serão os objetivos do sistema, os estímulos que o sistema receberá do meio ambiente, eventos que serão acionados e respostas que o sistema devolverá ao meio.

É a descrição da relação entre o sistema e o meio ambiente onde ele se encontra.

Modelo comportamental

Definição interna do sistema. Especifica todos os processos que irão compor o sistema. Define o modelo de dados que será utilizado para armazenar as informações por ele manipuladas.

Projeto

O objetivo é modelar o sistema determinando *como* implementar, em um ambiente de processadores a solução sistêmica idealizada na fase de análise.

Cuida das especificações referentes as limitações impostas pela tecnologia, a distribuição dos processos de acordo com os lugares onde serão executados.

As restrições de implementações da tecnologia não ideal e imperfeita serão incorporadas através de atividades de infra-estrutura administrativas.

Requisitos essenciais

Um requisito essencial é verdadeiro quando o sistema deve cumpri-lo qualquer que seja a tecnologia de implementação escolhida.

Se um sistema pode cumprir sua finalidade sem que um requisito seja implementado, então esse requisito é falso.

Requisitos falsos aparecem de várias formas dentro do sistema:

- Cópia de sistemas antigos.
- Hábito do analista.
- Pensar na tecnologia antes do tempo.

Qualquer requisito que não possa ser verificado (quantificado e mensurado) no final do projeto é um requisito falso.

O problema de introduzir requisitos falsos é que eles aumentam o risco do projeto não se completar a contento. Um requisito falso pode mascarar um requisito verdadeiro. O acúmulo de requisitos falsos aumenta a complexidade do sistema.

A busca dos requisitos verdadeiros é uma das principais formas de garantir o sucesso de um projeto.

Requisitos falsos tecnológicos

São aqueles incluídos em um sistema por antecipação de alguma característica tecnológica futura (linguagem de programação a ser utilizada), ou por alguma característica tecnológica passada.

- Incluídos pelo passado: quando incluimos algo que existe na implementação existente, mas que não é necessário ao funcionamento do sistema.
- Incluídos por antecipação: quando incluimos algo na especificação em função de alguma tecnologia escolhida para a implementação.

Requisitos falsos arbitrários

São incluídos por preciosismo ou por características da ferramenta de modelagem.

- Influência da ferramenta de modelagem: quando incluimos na especificação algo desnecessário para fazer o sistema, mas necessário por alguma característica da ferramenta de modelagem.
- Incluídos por preciosismo: quando incluimos uma função que não foi solicitada pelo usuário.

Agentes externos

Representamos as pessoas, departamentos, empresas, máquinas ou sistemas que interagem com o sistema sendo analisado, enviando ou recebendo dados por meio de agente externos (entidades externas).

Os agentes externos controlam o funcionamento do sistema. Eles detêm o poder para iniciar as atividades essenciais, ao enviar um estímulo ao sistema. Eles também recebem todas as respostas emitidas pelo sistema.

A verdadeira essência de um sistema está relacionada com sua função no negócio em que está inserido. Os usuários do sistema não são necessariamente os agentes externos que interagem com o sistema.

Agentes externos são pessoas ou artefatos tecnológicos que detêm poder de iniciar o evento.

Exemplo: em um sistema de vendas o agente externo que inicia o evento é o comprador e não o vendedor que está usando o sistema.

O iniciador é o agente externo que inicia o sistema, o transportador é o usuário do sistema responsável por introduzir dados fornecidos pelo iniciador.

Outro tipo comum de agente externo é aquele que representa uma instituição ou departamento externo ao ambiente de uso do sistema.

Existem os agentes externos que são máquinas (sensores, sistemas, etc.).

Muitos agentes devem ser representados não só fora do sistema, mas também em sua memória. Isso acontece quando o sistema deve guardar dados sobre um agente externo, de forma a poder reconhecer ou referenciar um agente externo (exemplo: enviar uma conta para o agente externo).

No caso de um sistema acadêmico os alunos são agentes externos, pois solicitam boletins, e são entidades, pois devem guardar dados sobre os alunos. Já os funcionários da secretaria da escola são agentes externos, pois podem pedir alguns documentos específicos guardados no sistema, mas o sistema não precisa saber quem são.

Eventos essenciais

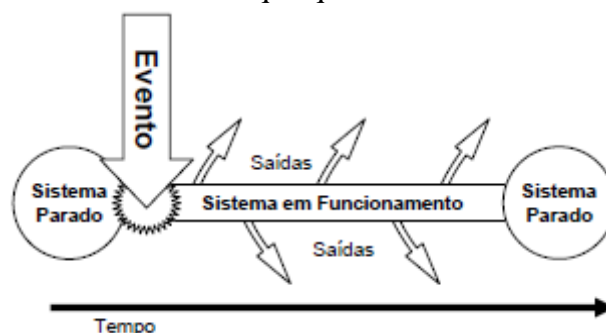
Na modelagem essencial tudo ocorre em função de eventos. Isso acontece porque imaginamos que o sistema possui dois estados: em atividade ou esperando um evento.

O acontecimento do evento faz com que o sistema entre em funcionamento e então realize todas as tarefas necessárias para atender aquele evento (a atividade essencial corresponde ao evento).

O sistema só funciona quando acontece um evento. O sistema é incapaz de gerar um evento. Não existem eventos essenciais internos ao sistema.

Evento define um estímulo, e compreende todo o conjunto de ações efetuado pelo sistema para executar a atividade (resposta planejada).

O evento funciona como um gatilho, disparando uma reação em cadeia, que para apenas pela impossibilidade de realizar qualquer outra atividade.



Se o sistema pára, só pode voltar a funcionar com um evento.

O estímulo relacionado ao evento, o fluxo de dados que parte do agente externo em direção à atividade, possui toda a informação necessária para realizar a atividade, incluindo partes opcionais.

Os eventos são descritos tendo como sujeito o agente externo que os iniciam, por exemplo, “Aluno solicita matrícula”.

Os eventos correspondem a um estímulo que cruza a fronteira do sistema de fora para dentro.

As atividades essenciais não se comunicam diretamente (através de fluxos de dados). Toda comunicação entre atividades essenciais é feita por meio do uso da memória do sistema.

Propriedades dos eventos

Um evento pode ser caracterizado pelas seguintes propriedades:

- Deve ocorrer em um momento específico no tempo.
- Deve ocorrer no ambiente e não dentro do sistema.
- A ocorrência do evento deve estar sob o controle do ambiente e não do sistema.
- O sistema pode detectar que o evento ocorre.
- O evento é relevante (o sistema deve fazer alguma coisa quando ele ocorre).

Tipos de eventos

Cada evento pode ser externo ou temporal.

Evento externo

Quando parte do ambiente para dentro do sistema. Exemplo: um comando ou pedido do usuário.

Eventos externos podem ser:

- Agendado (esperado): quando sabemos que ele vai acontecer em um instante específico, ou que tem um limite de prazo para acontecer.
Quando um evento agendado não acontece (o pagamento de uma conta, por exemplo) pode ser necessário tomar uma atitude específica. Dizemos então que eventos agendados podem necessitar que sejam definidos não-eventos (eventos que acontecem em função de outro não ter acontecido). Exemplo: “uma semana após esgotar o prazo de pagamento, enviar lembrete”.
- Não-agendado (não-esperado): quando não podemos determinar momento ou limite para seu acontecimento.

Evento temporal

Quando é provocado por uma mudança no tempo, como um alarme de relógio ou uma data no calendário.

Eventos temporais podem ser implementados como eventos internos, mas não necessariamente. A análise essencial exige que não existam eventos essenciais internos. A implementação não está limitada por essa regra.

Eventos temporais podem ser:

- Absolutos: ocorrem em função unicamente do calendário e do relógio.
- Relativos: quando são definidos em função do decorrer de um prazo depois do acontecimento de outro evento.

Um não-evento é um evento temporal relativo que deve acontecer se um evento agendado não ocorre, possivelmente considerando um prazo.

Evento temporal ocorre porque o sistema tem um contrato para entregar informação a um agente em um momento específico.

Descrevendo eventos essenciais

A sintaxe para definir eventos externos é:

<agente externo - sujeito> <verbo no presente> <objeto direto>
“Cliente solicita lista de produtos”.

Um evento agendado pode ter o seguinte padrão:

<agente externo - sujeito> <verbo no presente> <objeto direto>,
<prazo>

Onde o prazo pode ser absoluto ou relativo a outro evento ou resposta de evento.

“Fornecedor envia produtos pedidos, até 30 dias depois do pedido”.

A sintaxe para definir eventos temporais é:

<condição temporal>, é (hora/dia/etc.) de <verbo no infinitivo>
<objeto>

ou

<condição temporal>, <verbo no infinitivo> <objeto>

"Todo dia 30, é dia de enviar declaração de vendas"

"Dia 30, enviar declaração de vendas".

A sintaxe para um não-evento é:

<condição de não acontecimento de um evento>, <prazo ou condição temporal>, <verbo no infinitivo>, <objeto>

"Fornecedor não enviou produtos pedidos, depois de 30 dias do pedido, avisar comprador".

Exemplos comuns:

- Cliente envia pedido de compra.
- Fornecedor entrega mercadoria.
- Fornecedor entrega mercadoria, até 10 dias depois do pedido.
- Vendedor solicita mercadoria.
- Filial envia vendas diárias.
- Cliente aluga fita.
- Ao final do mês, imprimir folha de pagamento.
- Ao fim do dia, imprimir resumo de vendas.
- Gerente solicita relatório de produção.
- Caso o cliente não pague a conta, 20 dias depois, invocar departamento jurídico.
- Caso o aluno não apresente boletim assinado, 10 dias depois, enviar aviso aos pais.

Levantamento dos eventos essenciais

Através de entrevistas e reuniões JAD, levantar uma lista inicial de eventos e refinar essa lista enquanto analisa a resposta para cada evento.

Descrever uma lista de eventos a partir dos eventos fundamentais e depois listar todos os eventos necessários para que os eventos fundamentais funcionem. Ou partir dos eventos que geram dados para chegar aos eventos que geram relatórios no final.

Classificação dos eventos

Para testar a forma como o evento está descrito faça algumas perguntas de teste:

- Eventos externos:
 - São agendados?
 - Se agendados, possuem um não-evento correspondente?
 - São não-agendados?
 - São uma solicitação ou possuem dados?
- Evento temporais:
 - Só ocorrem dessa forma?
 - São realmente temporais ou podem ser calculados antes?
 - São relativos?
 - São não-eventos?
 - Qual é o evento original?
 - O evento original existe sempre?

Podemos construir uma tabela de classificação de eventos. Todo evento deve ser facilmente classificado. A dificuldade de classificar um evento demonstra que ele não foi compreendido e indica que ele pode não estar correto, tanto na sua interpretação ou na sua descrição, ou até que seja um requisito falso.

A tabela permite que verifiquemos se todos os eventos agendados possuem um ou mais não-eventos correspondentes.

Número	Evento	Classificação				
		Externos		Temporal		
		Agendado	Não-agendado	Relativo	Absoluto	Não-evento
1	Gerente cadastra distribuidora		×			
2	Gerente cadastra livro		×			
3	Cliente pede livros		×			
4	Sexta-feira, é hora de fazer requisições de livros para as distribuidoras				×	
5	Distribuidora entrega livros	×				
6	Gerente solicita relatório de vendas		×			
7	Gerente solicita relatório de livros em atraso		×			
8	Se a distribuidora não entregar os livros pedidos, depois de 15 dias, cancelar pedido					× ₅

Memória do sistema