Um olho no passado, dois olhos no futuro: Diagrama de Fluxo de Dados

Salve jovem padawan, mais uma vez vamos fazer uma viagem no tempo, para os primórdios da informática, onde as grandes ideias surgiram, sementes foram semeadas e gradualmente o sistema foi sendo criado.

O artigo de hoje irei para beber na fonte de dois gigantes Ed Yourdon e Larry Constantine que publicaram na década de setenta o livro Projeto Estruturado de Sistemas, uma rocha forte que serviu de alicerce para gerações de DEVS e inpirou David Martin e Gerald Estrina a criarem sua ferramenta, mas antes que pergunte-me por que falo deles numa era de orientação a objetos e tanta tecnologia de ponta.

Elementar meu caro Watson, digo padawan, existem milhões de linhas de código legado, inúmeras bases de dados esperando serem preparadas para migração através de ETL, a maior parte das pessoas que as construíram foram para a reforma e estão curtindo a boa vida de aposentados no Caribe.

Introdução

Quando falamos de migrar software legado, primeiro pensamos nas regras de negócios, abrindo as fontes para extrair a logica e migra-la para uma nova ferramenta ou tecnologia, porem uma das áreas que mais engenho necessita são os dados.

O pobre analista se ve, frente a centenas de tabelas e milhões de registros, querys que retornam tuplas e mais tuplas de dados, mas o que serão? Qual a necessidade? Por onde começar?

Seus problemas acabaram, existe uma pratica oriunda dos laboratórios da IBM e que aqueles dois gigantes ajudaram a refinar, melhorando e servindo de base para outros grandes mestres aprimorarem, estou falando do D.F.D.

O que é D.F.D.?

Eis que surge mais um anacronimo para a louca sopa de letrinhas da informática. Nunca me canso de repetir isso, essa sigla refere-se ao Diagrama de Fluxo de Dados, uma ferramenta que permite conhecer o Sistema, suas fronteiras, seus caminhos críticos, suas comunicações e repositórios de dados.

Mas que raio de ferramenta milagrosa è essa? Uma que serve de mapa, um guia das origens e destinos, possuindo vários níveis de detalhes, partindo do mais genérico até o menor detalhe, sendo uma das suas vantagens a simplicidade, possuindo poucos sinais ajudando a evitar confusões e interpretações fantasiosas.

Normalmente o DFD começa no nível 0, a visão macro descendo ao detalhe nos níveis 1 até o 5. Dependendo da instalação e das equipes de mantenimento dedicarem um tempinho para atualizarem-no.

Como funciona o DFD?

Seu funcionamento é mais simples do que imaginamos, o pulo do gato dos seus criaturas foi a simplicidade, poucos símbolos, poucas regras. Martin & Estrin sabiam que quanto mais complexidade, menos utilizade, muitas regras iriam gerar confusão e a sua aceitabilidade seria reduzida.

Existem 4 símbolos que representam todo o conjunto de trbalho

* entidade externa

um retangulo

* processo

um circulo

* armazenamento de dados

um cilindro

* fluxo de dados

uma flecha

O processo consiste em definir as fronteiras do Sistema, dentro destes limites teremos as Entidades Externas, que alimentam e recebem informações do Sistema, os Circulos que representam os processos internos de transformação da informação, os Cilindros que armazenam as informações em diversos estados e as flechas que indicam o controle do fluxo do processamento.

Qual a vantagem deste diagrama?

A vantagem é ser de fácil entendimento, onde um usaurio do Sistema, mesmo sem possuir nenhum conhecimento de programação, pode ler e entender a origem, o destino, o processo da informação em cada etapa.

O que é um diagrama de Contexto?

O DFD nível 0 também chamado de diagrama de contexto, pois é a visão mais macro e geral de todo o Sistema ou Processo. Sua principal missão é apresentar o sistema no mais alto nível, com todos os players envolvidos e limites bem definidos.

Tem que ser claro e de fácil entendimento, apresentando os principais inputs, outputs e processos internos e externos, para que o público externo possa conhecer e saber responder as perguntas básicas: O que, onde, por que e quando.

O que significa DFD 0 , DFD 1, DFD 2.etc ?

Esta numeração indica ao leitor o nível de detalhe e grau de aprofundamento do diagrama, partindo do nível 0 o mais geral possível até o nilvel N explodindo o detalhe ao menor nível.

Atualmente este diagrama esta sendo pouco usado e muitos DEVs desconhecem seu uso , meu artigo tem por objetivo resgatar e apresentar aos leitores os detalhes e auxiliar na documentação dos projetos, melhorando a qualidade do software.

Como usa-la nos dias atuais?

A maneira mais rápida é rascunhar em lápis e papel, depois convido aos aspirantes a Analistas de Sistemas a conhecerem o MS VISIO, ou qualquer outra das dezenas de ferramentas onlines de elaboração de fluxos de processos.

Não é um processo difícil é facilita enormente a projetar front-ends, aconselho aos parceiros de UX e UI, a debruçarem-se e conhecerem mais sobre o Diagrama de Fluxo de Dados, recomendação estendidas aos DBAs, afinal conhecer o processo de transformação na raiz, facilita encontrar erros e anomalias.

Um conhecimento importante, guarde sempre em mente, a força de uma corrente é determinada pelo seu elo mais fraco, então conheça o sistema, conheça a origem do dado, descubra suas transformações e seus principais processos.

Regra de ouro do DFD

Um dado somente deve ser armazenado no sistema se passar por um processo

Nao existe dados sem processamento e o armazenamento de dados obrigatoriamente deve ter sempre um fluxo, seja de entrada, seja de saida ou mesmo ambos.

O processamento é o rei, transformando sempre os dados, por isso também obrigatoriamente tem que ter uma entrada e uma saida

Num DFD todo processo ou vai para outro processo ou um armazenamento de dados.

A informação nao surge do nada, todo dado armazenado em um sistema obrigatoriamente deve passar por um processo.

Conclusão

O objetivo deste artigo foi apresentar ao jovem padawan, os conceitos fundamentais sobre o Diagrama de Fluxo de Dados, falar sobre seus símbolos e uso, deixo em aberto a utilização do UML e a evolução em novos diagramas.

A ideia foi resgatar uma ferramenta do passado e comentar sobre os benefícios do seu uso, alertando claro, que para alguns autores ela está obsoleta e sem vantagens ao DEV do século XXI, o que discordo bravamente. Afinal existem muitos sistemas legados, desenhados sob esta metodologia.