**Resumo LeCroy Corporation**

**Introdução:**

- O sistema baseado em componentes é uma nova arquitetura de software que nasceu em meados do ano 1990.

- Envolve: O desenvolvimento de componentes de software e a construção de software através de sistemas de integração de componentes de software pré-existentes

- Grandes vantagens do uso: permite recombinar componentes de software o que faz com que o produto seja lançado mais rápido no mercado, tenha melhor qualidade e reduzidos custos no seu desenvolvimento.

- Desvantagem(inconveniente): As empresas que queiram utilizar este sistema tem duas possibilidades ou fazem o reverse engenering de todo o produto, ou então começam a construção da plataforma a partir do zero o que pode dar grandes gastos quer em termos monetários quer em termos de tempo.

**LeCroy**

Razão porque escolheram a arquitetura CBD: Para ganhar vantagem competitiva no mercado visto nesta altura poucas empresas adotaram este método.

Desafios – Para além de terem de lidar com o reengeniring usando a arquitetura CBD o que na altura já era um grande desafio visto eles serem dos primeiros a usar este modelo, eles tiveram outros desafios relacionados com a distribuição do trabalho a nível global como a diferença de horas entre países diferentes a distancia e as diferenças culturais.

1 fase

Sempre que um modelo novo era desenvolvido este passaria a fazer parte do sistema monolistico já existente tornando –se este cada vez maior,mais complexo e mais difícil de adicionar novas funcionalidades pois tornava-se mais complicado de identificar as aréas e quando identificadas tornava-se ainda mais cansativo adicionar os novos plug-ins e era cada vez mais difícil lidar com as dependências entre as equipas de desenvolvimento de software.

A LecRoy no fim queria se associar a uma empresa japonesa mas não pretendia que estes tivessem acesso a todos o software da empresa, por isso começaram um projeto designado por Oracle e decidiram separar o programa monolitico em 3 partes.

2 fase

O objetivo era dividir o software em 3 partes

Módulos: Sistema Operativo,Core(Continha as funcionalidades comuns a todos os osciloscópios) e Sistema de aquisição(parte que se altera sempre que é produzido software novo).

Inicialmente para fazer o reeengenering tiveram que verificar as versões que estavam instaladas em cada oscilioscopio para isso usaram um sistema de controlo de versões.

Foi o inicio da era do software orientado a objeto passando a ser reutilizado os seus recursos

Com isto quiseram começar a construir a nova geração de telescópios. Existiam várias questões então como por exemplo: Em que tecnologia devia ser feito, como deveria ser feito, o mercado era bastante competitivo para ganhar vantagem tinham de ter uma tecnologia mais moderna em relação aos competidores.

SO: Windows(várias vantagens como:obtenção de software já feito sem ser necessário criar e programas como excell Mathlab

Como criar um escopo baseado em windows?

Eles mantiveram o escopo que tinham mantiveram o código e por cima criaram um programa que funcionava como interface com o utilizador.

Problema – A empresa Tektronix já o tinha feito

Então LeCroy para ganhar vantagem competitiva decidiu que era altura de didivir em mais componenetes.

Desafio- Existiam aspectos difíceis de remover da atual arquitectura.

Decisão – Visto a complexidade de transformar esta arquitectura em mais componenetes e visto que a Tektronix já o tinha feito da mesma forma eles decidiram que para ganhar vantagem competitiva tinham duas escolhas Ou optavam pelo caminho complicado de dividir as componentes que iria ser um caminho duro, ou então começavam tudo do 0.

Microsoft COM problema – não tinha maneira de gerir as componenetes e as suas dependências. Com o COM eles podiam criar as componenetes mas não havia forma fácil de criar essas dependências com poucos compoenentes não havia problema pois estes podiam ser modelados, mas com centenas de componenetes torna-se um problema.

Introdução

1

A arquitetura de software de um sistema é um conjunto de estruturas necessárias para refletir sobre o sistema, que incorpora elementos de software, relações entre eles, e respetivas propriedades

2

- Sistemas Embebidos

- Sistemas Monoliticos

-Sistemas Cliente-Servidor

- Sistemas Peer-to-peer

- Sistemas Distribuidos

- Sistemas baseados em Barramento.

3

Em termos de arquitetura de software a empresa LeCroy fez a transição do que era inicialmente um sistema Monolitico para um sistema baseado em componenetes.

4

Existiram vários desafios relacionados com a passagem de um sistema monolítico para um sistema baseado em componentes vamos apenas citar os mais importantes:

- Culturas,línguas e fusos horários completamente diferentes entre equipas de desenvolvimento tornando o envio de dados e as conversas complicadas.

- Na implementação do sistema Microsoft COM os programadores seniores principalmente tiveram enormes dificuldades em passar do antigo sistema para o mais recente, sendo que os gestores forneceram livros necessários á aprendizagem, e arranjou uma equipa pequena que se responsabilizava para saber tudo em relação ao software.

- Decisão entre continuar a partir o sistema a partir dos módulos que tinham ou começar do zero, foi um dos maiores desafios da empresa LeCroy.

5

Os gestores da LeCroy adoptam várias estratégias á medida que o software se vai aproximando daquilo que eles pretendem.

Inicialmente a estratégia deles era partir o sistema monolítico em três, para que dessa forma eles podessem disponibilizar apenas uma parte(a pretendida e necessária á parceria Japonesa) do seu software.

A fase correspondente aos três módulos serviu basicamente para os gestores da LeCroy ganharem tempo, eles iam desenvolvendo novos produtos á medida que iam criando um sistema baseado em componenetes do zero, como isto demoraria bastante tempo eles não podiam simplesmente parar tinham que continuar a desenvolver e foi isso que fizeram.

Outra estratégia fundamental foi a semana de convivência entre equipas, tal como os gestores disseram o mais importante aí não era apenas aprender aspetos relacionados com o Microsft COM, também era fundamental para eles naquela altura que as equipas se conhecessem e que comunicassem, naquela altura era fundamental a troca de ideias entre membros das equipas de software e isto resultou muito bem.

No final tinham um ultimo problema relacionado com as dependências entre componenetes e resolveram a questão adotando um modelo que estava a ser desenvolvido por uma outra equipa de software da LeCroy.

6

A alternativa de implementação que adotaríamos neste caso seria a utilização de sistemas distribuídos, onde basicamente as componenetes faziam parte dos servidores existindo a comunicação entre essas mesmas componentes e um grande número de clientes.

-

7

Para um Software ser considerado de qualidade podemos ver isto de duas perspectivas: Utilizador e equipa de desenvolvimento, sendo que os aspectos relacionados com a qualidade de software já se encontram de certa forma padronizados podendo estes também variar dependendo do software e tecnologias a utilizar, podemos considerar então os mais importantes como:

- funcionalidade

- confiabilidade

- facilidade de utilização

- economia

- segurança na utilização.

Existem outras que já não são tão direccionadas para o utilizador como por exemplo:

- boa documentação.

- facilidade de adicionar melhorias.

- flexibilidade

- facilidades de reparo.

- adaptabilidade

- facilidade de entendimento

8 – O software da empresa LeCroy fez precisamente essa transição do que inicialmente era considerado software sem qualidade para um software de qualidade pois a equipa de engenheiros que estava a tratar dos assuntos relacionados com o desenvolvimento da empresa teve obrigatoriamente que pensar nos assuntos relacionados com a qualidade de software, o que inicialmente era um software confuso com todas as componenetes misturadas, tornou-se um software muito mais limpo, facilitando o entendimento, a adição de melhorias(as dependências entre componenetes tornava mais fácil este processso) e as facilidades no reparo, por isso a transição era evidente.

9

Os requesitos que uma ferramenta de software distribuído deve ter são:

- Modificações concorrentes de ficheiros(a alteração por parte de alguém não dever interferir na alteração feita por outra pessoa).

- Replicação de ficheiros(Deve manter cópias actualizadas dos ficheiros em diferentes locais).

- Heterogeniadade(deve ser concebido para diferentes plataformas).

- Tolerancia a falhas

- Consistencia

- Protecção

- Eficiencia

10

A re-engenharia é um sistema utilizado pelas organizações para se manterem competitivas no mercado e alcançarem as suas metas, reformulando o seu modo de fazer negócios, as suas atividades e/ou processos.