UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE MATEMÁTICA E INFORMÁTICA

TRABALHO DE LICENCIATURA

Gestão de Qualidade no Ciclo de Desenvolvimento de Sistemas de Informação

José Rafael Armando Almoço

Maputo, Junho de 1996

IT-114

UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE MATEMÁTICA E INFORMÁTICA

TRABALHO DE LICENCIATURA

Gestão de Qualidade no Ciclo de Desenvolvimento de Sistemas de Informação

Estudante: José Rafael Armando Almoço

Supervisora: dra. Esselina Macome

Maputo, Junho de 1996

UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE MATEMÁTICA E INFORMÁTICA

TRABALHO DE LICENCIATURA

Gestão de Qualidade no Ciclo de Desenvolvimento de Sistemas de Informação

José Rafael Armando Almoço

Maputo, Junho de 1996

UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE MATEMÁTICA E INFORMÁTICA

TRABALHO DE LICENCIATURA

Gestão de Qualidade no Ciclo de Desenvolvimento de Sistemas de Informação

Estudante: José Rafael Armando Almoço

Supervisora: dra. Esselina Macome

Maputo, Junho de 1996

DECLARAÇÃO DE HONRA

"Declaro que este trabalho é resultado da minha investigação, que não foi submetido para outro grau que não seja o indicado - **Licenciatura em Informática** - da Universidade Eduardo Mondlane".

Maputo, aos 26 de Junho de 1996

O Estudante

José Rafael Armando Almoço

(José Rafael Armando Almoço)

AGRADECIMENTOS

Muitas pessoas contribuíram directa ou indirectamente para a concepção, elaboração e apresentação final deste Trabalho de Licenciatura.

Agradeço a todos os que apoiaram a minha formação no Curso de Licenciatura em Informática, e de forma particular:

- à minha supervisora, dra Esselina Macome, pelo apoio e confiança que me forneceu para a inspiração e desenvolvimento deste Trabalho;

- ao Eng. Enrique Rosa pelo esclarecimento de dúvidas que fui tendo ao longo da elaboração do trabalho;

- ao dr. Manuel Alves, Chefe do Departamento de Matemática e Informática, na criação de condições para realização do levantamento de informação em algumas organizações produtoras/fornecedoras do software de aplicação na Cidade de Maputo;

- ao Eng. Brito Marcos e ao Sr. Afonso André da Petromoc E.E. pelo apoio que me garantiram em meios informáticos (hardware, software e consumíveis);

- às organizações produtoras/fornecedoras de software CIUEM, EXI e SORT Lda, por terem aceite realizar um levantamento de informação que permitiu a verificação prática e comparação das formas reais na implementação das práticas da qualidade;

- ao Instituto Nacional de Normalização e Qualidade (INNOQ) pelo fornecimento de material bibliográfico para a consulta.

- à minha família, pela compreensão e apoio que me forneceu permitindo-me dispor de tempo para a realização deste trabalho.

José R. A. Almoço

RESUMO

A qualidade do software é um dos requisitos que as organizações produtoras de software têm de atingir para garantirem a sua participação efectiva e competitiva no mercado de vendas dos seus produtos assim como para a melhoria da prestação dos serviços dos utilizadores que forem a utilizar o software.

Através deste trabalho pretende-se descrever a gestão de qualidade do software no ciclo de desenvolvimento de sistemas de informação e sua inserção no sistema de gestão de qualidade. Pretende-se também verificar a implementação dos sistemas de gestão de qualidade em algumas organizações produtoras de software da Cidade de Maputo.

Para a realização do trabalho recorreu-se a bibliografia disponível, recolha de informação em três organizações produtoras de software de aplicação da Cidade de Maputo. A importância deste estudo é fornecer os meios e aplicabilidade das actividades de gestão, garantia e controlo de qualidade do software para os gestores de projectos de desenvolvimento de sistemas de informação, organizações produtoras de software, estudantes de informática e clientes.

Sendo a gestão de qualidade dependente de metodologias e modelos de desenvolvimento de software, neste trabalho foi estudado o método orientado a objecto assente no modelo espiral para demonstar a implementação de algumas das actividades de garantia e controlo de qualidade num ciclo de desenvolvimento de software. Para o efeito, foi também usado como referência a ISO 9000-3, uma norma internacional que define as formas de desenvolvimento, fornecimento e manutenção do software.

Do estudo feito concluiu-se que a definição de qualidade não é consensual e por isso a sua caracterização e avaliação depende dos intervenientes no desenvolvimento do software. Além disso, de acordo com a realidade moçambicana, onde não existe uma norma interna sobre software, torna-se mais difícil a introdução dum Sistema de Gestão de Qualidade. Contudo, as organizações produtoras de software preocupam-se em produzir um software com qualidade e que satisfaça as necessidades dos clientes.

ÍNDICE

ACRÓNIMOS

ANSI	American National Standards Institution
AOO	Análise Orientada a Objecto
BS	British Standard
BSI	British Standards Institution
CDP	Componente de domínio do Problema
CGA	Componente de Gestão de Actividades
CGD	Componente de Gestão de Dados
CIH	Componente de Interação Humana
CVD	Ciclo de vida de desenvolvimento
CVDS	Ciclo de vida de desenvolvimento do sistema
CVO	Ciclo de Vida do Objecto
DOO	Desenho Orientado a Objecto
GEN_ESP	Generalização/Especialização
ISO	International Standards Organization (Organização Internacional de Normalização)
LPOO	Linguagem de Programação Orientado a Objecto
MOO	Método Orientado a Objecto
SGC	Sistema de Gestão da Configuração
SGQ	Sistema de Gestão da Qualidade
SOO	Software Orientado a Objecto
SSADM	Structured Systems Analysis and Design Methodology

I. INTRODUÇÃO

A Informática em Moçambique, está num estado de evolução dependente das condições económicas, culturais, históricas e políticas. O uso dos Sistemas Informáticos está crescendo, exigindo maior necessidade de pessoal que corresponda à evolução e às exigências das diferentes organizações existentes no país.

Os Sistemas Informáticos são introduzidos nas organizações com o objectivo de melhorar a prestação dos serviços ou o fornecimento dos produtos bem como para assegurar que as organizações possam sobreviver no mercado económico competitivo que assenta nas bases de custo, disponibilidade de prestação/entrega do serviço/produto no momento desejado, e a qualidade do serviço/produto.

A introdução do Sistema Informático numa organização encarada como um projecto envolve, geralmente, pessoas com diferentes habilidades para as diferentes fases do projecto exige uma gestão de modo a atingir os objectivos nele definidos.

A gestão do projecto de desenvolvimento de sistemas informáticos compreende a gestão do tempo, dos custo, da qualidade, e dos recursos humanos e materiais. Normalmente, os projectos têm gasto acima dos custos estabelecidos e são entregues acima dos prazos previstos. Quando não é permitida a alteração dos prazos e custos, então é afectada a qualidade do produto.

Os factores que contribuem para que a qualidade, no desenvolvimento de sistemas informáticos, seja muitas vezes prejudicada, são os seguintes:

- objectivos não mensuráveis para a avaliação da qualidade não possibilitam a determinação do nível do cumprimento dos mesmos;
- mudança rápida e constante de hardware e software, implicando uma alteração constante de alguns requisitos do utilizador para adaptar-se à nova realidade do mercado económico;

- dificuldade de fornecer dados adequados para medir o desempenho do software/sistema informático;
- complexidade crescente do software actual exigindo novas ferramentas e técnicas para a sua produção e avaliação;
- não existe uma definição universal da qualidade de software o que implica que a sua avaliação seja dependente da(s) pessoa(s) que a avalia(m).

Na base do descrito anteriormente, coloca-se a questão seguinte, que será a base deste Trabalho de Licenciatura.

Como gerir a qualidade do software no ciclo de desenvolvimento de sistemas de informação e sua inserção num sistema de gestão de qualidade da empresa?

Os objectivos gerais deste trabalho são os seguintes:

- descrever a gestão da qualidade do software no ciclo de desenvolvimento de sistemas de informação e a sua inserção no sistema de gestão de qualidade;
- verificar nas organizações produtoras de software de aplicação, na Cidade de Maputo, as formas de implementação do sistema de gestão de qualidade.

Os objectivos específicos deste trabalho são os seguintes:

- descrever os princípios e práticas da gestão de qualidade aplicados no desenvolvimento dos sistemas de informação, de forma particular o caso do software;

- identificar e descrever os possíveis critérios de qualidade, as características e as métricas do software;

- determinar as actividades a realizar de forma a controlar e a garantir a qualidade do software, no ciclo de desenvolvimento de sistemas de informação;

- descrever a gestão de qualidade no ciclo de desenvolvimento do software baseado no modelo espiral e num método orientado a objecto;

- identificar, nas organizações produtoras de software de aplicação da Cidade de Maputo, a forma como elas definem e implementam um programa de garantia de qualidade.

Espera-se que este trabalho que possa ser usado pelos estudantes de informática, organizações produtoras/fornecedoras e compradoras de software para a sua formação nos princípios da gestão de projectos de desenvolvimento de software; na identificação, definição, introdução e melhoramento de um Sistema de Gestão de Qualidade nas organizações produtoras do software; e, na identificação, definição e uso das técnicas e actividades de gestão, garantia e controlo de qualidade numa organização e/ou num projecto de desenvolvimento de sistemas de informação.

1. Metodologia

Para a realização deste Trabalho de Licenciatura, o autor recorreu à bibliografia disponível, realização de entrevistas em três organizações produtoras de software de aplicação na Cidade de Maputo, nomeadamente no CIUEM, EXI e SORT Lda. Para além disso, foi feita uma observação directa do controlo da qualidade de um projecto de desenvolvimento de sistêmas de informação realizado na Petromoc E.E., no qual o autor participou.

A bibliografia permitiu realizar o estudo descritivo do Sistema de Gestão de Qualidade e seus componentes. Com as entrevistas, pretendeu-se fazer um estudo explicativo-comparativo do grau e das formas de implementação dos Sistemas de Gestão de Qualidade ou práticas que garantam a qualidade do software nas organizações produtoras de software de aplicação. A observação prática do controlo de qualidade no projecto da Petromoc E.E. permitiu identificar a importância da qualidade na perspectiva do cliente.

II. SOFTWARE E QUALIDADE

"Toda gente gosta dum produto de boa qualidade - um que seja fácil de usar, que realize o trabalho que com ele se pretende, é de valor e pode ser obtido rápida e fidedignamente."[1]

1. Sistemas Informáticos

Existem várias definições sobre sistemas. Para este trabalho, um *sistema* é um conjunto de componentes relacionados que servem um objectivo comum. Os sistemas podem ser compostos por pessoas, máquinas, ferramentas e procedimentos. Um dos sistemas existentes numa organização é o *sistema de informação*.

Davis (1994) define um *sistema de informação* como um conjunto de componentes de hardware, software, dados, procedimentos e recursos humanos que funcionam com o objectivo de gerar, colher, armazenar, processar, reaver, analisar e/ou distribuir informação.

Os sistemas de informação que usam sistema de computador para a manipulação da informação são designados por *sistemas informáticos (sistemas de informação computarizados)*.

Um *sistema de computador* consiste de hardware e de software (Tanenbaum, 1987), conforme ilustrado na Figura 1. O hardware é um conjunto de elementos físicos utilizados no processamento de informação, e o software é um programa ou conjunto de programas destinados a efectuar um processamento em computador.

Existem dois tipos principais de software: software de sistemas e software de aplicação. O *software de sistema* é composto pelo sistema operativo e um número de utilitários adicionais. O *software de aplicação* é um conjunto de programas necessários para a resolução de um determinado problema por meio do computador.

[1] DAILY, K. Quality Management For Software. England: kevin Daily, 1992. p.1.

Kendall (1992) considera que um sistema de informação passa por uma série de fases previstas, desde da sua concepção até à sua "morte", e este processo é denominado *ciclo de vida de desenvolvimento do sistema* (CVDS). Dependendo do tipo do problema a ser resolvido, escolhe-se um tipo de ciclo para o desenvolvimento do sistema.

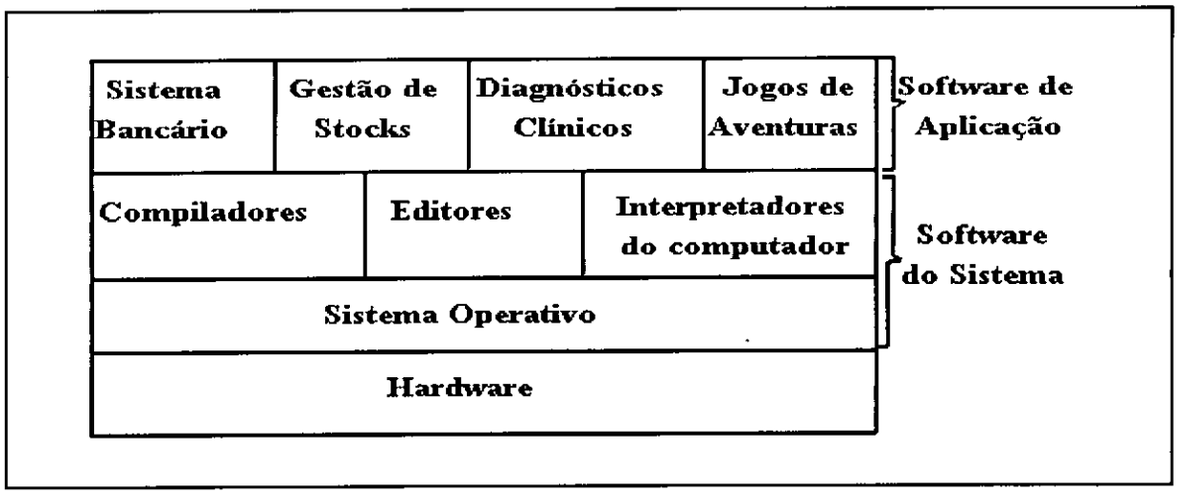

Figura 1 - Um Sistema de Computador

Os CVDSs variam dum sistema de informação para outro dependendo de vários factores. Por isso, Hawryszkiewycz (1994) considera que a escolha dum ciclo de desenvolvimento de sistemas de informação é influenciada pela natureza do problema, e em particular pelo:

- grau da estrutura do sistema;
- familiarização com a tecnologia; e
- tamanho do projecto.

Para o desenvolvimento do sistema de informação recorre-se a uma ou mais metodologias de desenvolvimento de sistemas que assentam num conjunto de passos e usam diferentes ferramentas. As metodologias devem incluir uma maneira de determinar os requisitos do sistema, usar métodos e ferramentas que garantam a

construção rápida e barata dum sistema com qualidade, isto é, um sistema que trabalhe bem, sem erros e que satisfaça todos os requisitos do utilizador.

Assim, são introduzidos no ciclo de desenvolvimento dos sistemas, mecanismos que asseguram o cumprimento dos requisitos à medida que ocorre o desenvolvimento. Estes mecanismos permitem que os requisitos não sejam perdidos ou alterados durante o ciclo e que os erros sejam mínimos.

Sendo o desenvolvimento do sistema de informação composto por várias fases e actividades, é necessário que se efectuem, ao longo do ciclo de desenvolvimento, validações dos produtos de saída de cada fase ou actividade em relação aos requisitos iniciais. É necessário verificar se as transformações ocorridas sobre as entradas, em cada fase ou actividade, fornecem as saídas correctas.

É também necessário ao longo de desenvolvimento dos sistemas de informação que existam recursos humanos, materiais e financeiros adequados, um controlo eficiente e eficaz destes recursos, supervisão e revisão constante da documentação fornecida e/ou produzida, e que isto seja realizado através de gestão de projecto.

Uma das práticas da gestão de projecto é medir o desempenho do próprio ciclo de desenvolvimento e, se necessário, melhorar o ciclo dum projecto para outro. Isto significa que através de gestão de projecto se faz o controlo do progresso do projecto e garante-se que o mesmo alcance as normas de qualidade preestabelecidas, os requisitos inicialmente mencionados em conformidade com os custos e tempo estabelecidos.

2. Qualidade

Apesar de não existir uma definição de consenso sobre o que é a qualidade, neste Trabalho é utilizada a definição do Ganhão (1991) que considera a *qualidade* como

sendo a totalidade das características de um produto ou serviço que lhe confere a capacidade de satifazer necessidades expressas ou implícitas.

Nesta definição existem dois pontos básicos a tomar em conta: a 'totalidade das características' e a 'capacidade de satisfazer necessidades'. A 'totalidade das características' significa que existe uma série de características que determinam a qualidade do produto, e a 'capacidade de satisfazer as necessidades' implica que os objectivos do produto devem ser conhecidos e a partir deles formulam-se os requisitos da qualidade do produto.

Na base desta interpretação conclui-se que, para que seja possível produzir um software com qualidade, é importante e imperioso que se definam claramente as necessidades que o cliente quer ver satisfeitas no produto a ser desenvolvido.

2.1. Qualidade do Software de Aplicação

Rocha (1990), considera a *qualidade do software* como um conjunto de propriedades a serem safisfeitas num determinado grau de modo que o software satisfaça as necessidades de seus utilizadores.

Nesta definição da qualidade do software, o aspecto 'determinado grau' é que garante a satisfação das necessidades do cliente. É responsabilidade do produtor do software e do cliente estabelecerem os níveis aceitáveis da qualidade do software de forma a que este satisfaça as necessidades do cliente.

As definições dadas por Ganhão (1991) e Rocha (1990) são similares em relação aos pontos básicos. O termo 'características' empregue por Ganhão (1991) é equivalente ao termo 'propriedades' de Rocha (1990), para além de ambos fazerem referência às necessidades do utilizador.

Para a obtenção dum software com qualidade é necessário que o cliente identifique as suas necessidades, formule os requisitos correspondentes de forma simples, clara, concisa, não ambígua e testável, negocie com o produtor/fornecedor o tipo de software a ser produzido/fornecido cumprindo com os requisitos formulados, num tempo e custo viáveis. Estes requisitos, do ponto de vista de qualidade de software, podem ser funcionais e/ou operacionais, e cumprindo as normas de qualidade definidas que podem ser específicas ou gerais.

Com os requisitos definidos, o produtor escolhe as técnicas e actividades (tratado no ponto II, 3.1) que devem ser realizadas para atingir os requisitos, para além de definirem as acções a realizar de forma a garantir que o software satisfará os requisitos apresentados pelo utilizador. Um exemplo disto é a adopção de um plano de qualidade.

Daily (1992), considera que um trabalho de manufacturação requere três elementos principais na visão da qualidade: desenho, processos e materiais (ou matéria prima). Estes três elementos necessitam de ser convenientemente definidos para que se obtenha um produto com qualidade satisfatória.

Utilizando como base a afirmação do Daily (1992), para o desenvolvimento do software de aplicação, os componentes do software devem ser convenientemente definidos (exemplo: ficheiros de dados, interfaces e outras estruturas). É necessário que se definam claramente as técnicas, ferramentas e métodos usados assentes num modelo do ciclo de vida para identificar, criar e reunir os componentes num produto e testá-lo; e que a informação ou dados fornecidos ao analista sejam certos/adequados e os dados de entrada para a execução do software de aplicação sejam correctos.

2.2. Como medir a Qualidade do Software de Aplicação?

> "(...) é difícil imaginar uma organização que faça melhoramento da sua produtividade ou qualidade se ela não mede o que está fazendo."[2]

Avalia-se a qualidade do software de aplicação através de medidas de qualidade. Uma *medida* é uma grandeza que serve de termo de comparação, neste caso, do software. As comparações da qualidade do software são feitas através das medidas de qualidade.

A introdução de medidas no software teve como base as medidas industriais, classificadas segundo Daily (1992) em dependentes e independentes do tempo.

Uma *medida do software independente do tempo* é aquela que se ajusta aos requisitos do utilizador ao longo do tempo, desde que o software não seja alterado. A funcionalidade[3] é um exemplo de uma medida independente do tempo porque um software que tenha sido avaliado e identificado como cumprindo com os requisitos funcionais préviamente definidos, continuará a satisfazê-los ao longo do tempo a menos que o próprio software e/ou os requisitos funcionais sejam alterados.

As medidas independentes do tempo são as mais usadas nas práticas dos testes de sistema e testes de aceitação do software pois, a sua determinação não requere a variável tempo.

Uma *medida dependente do tempo* é a que varia em função do meio ambiente da execução do software, a forma como o software é usado, e/ou da realização incompleta dos testes, por exemplo, tempo médio entre falhas. Geralmente, estas

---

[2] YOURDON, Edward. Object-Oriented Systems Design: An Integrated Approach. EngleWood Cliffs: Printice-Hall Inc, 1994, p.104

[3] A funcionalidade é uma medida que determina o nível do cumprimento dum produto em relação aos seus requisitos funcionais.

medidas são determinadas durante a fase da operação do software e, quando documentadas, permitem uma tomada de decisões para a execução das acções correctivas e/ou de melhoramento.

Daily (1992) considera que um software pode estar num estado de operação, revisão ou transição e para cada um destes estados, o software deve possuir características específicas.

O software em operação deve ter utilidade, precisão, fiabilidade, eficácia, compactibilidade e integridade. Para que seja fácil a revisão, o software deve possuir as seguintes características: inteligibilidade, legibilidade, alterabilidade e testabilidade. Para isso é necessário a documentação do software que permita a identificação clara e precisa das alterações além de facilitar a realização do teste. E o software em transição deve ter: portabilidade e "reusabilidade".

Na aquisição de um software pode-se pedir a presença de algumas ou todas as características (ANEXO A), dependendo dos objectivos que o cliente definir, conhecimentos que tiver em relação às características do software e o conhecimento que possuir dos custos e benefícios da qualidade que ele exigir do software, pois cada característica implica um custo que o cliente terá de suportar.

Algumas destas características não são observadas nas organizações produtoras de software abrangidas pelo estudo, ao longo do seu desenvolvimento. Por exemplo, nas três organizações abrangidas realizam-se os testes de sistema mas não efectuam o registo dos resultados do teste. Assim, não é possível avaliar de forma eficiente o impacto duma alteração estrutural ou funcional na melhoria da qualidade do software. Esta forma de trabalho não garante a testabilidade e alterabilidade.

2.2.1. Estrutura Hierárquica de Qualidade

Adaptando uma estrutura hierárquica, as características situar-se-iam no topo da estrutura. Estas características são subdivididas em critérios[4] que permitem a verificação das mesmas. Os critérios, por sua vez, podem-se subdividir em medidas quantificáveis, que são designadas por métricas (Fig. 2). Se não for possível subdividirem-se em métricas, então, é possível elaborar uma lista de perguntas, que permita atribuir um valor ao critério correspondente.

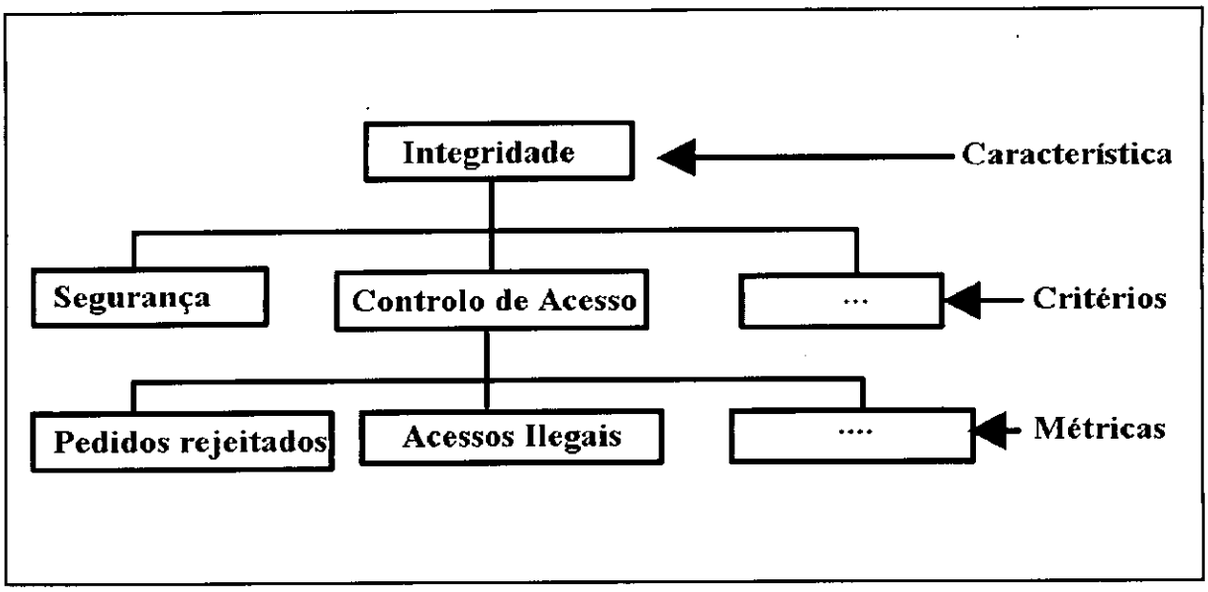

Figura 2 - Exemplo Duma Estrutura Hierárquica de Qualidade

Às métricas e/ou lista de perguntas são atribuidos valores, através do uso das técnicas de avaliação disponíveis. É através dos valores determinados que é definida a qualidade do software.

Benyon e Davis (1992), definem uma *métrica do software* como sendo um número extraído dum produto do software que fornece um "feedback" ao pessoal e/ou gestor

[4] São traços essênciais que servem para determinar ou classificar uma coisa, neste caso o software.

de qualidade para controlar a degradação estrutural do sistema durante a manutenção e para estimar projectos de software.

As métricas, segundo o objecto de avaliação, podem ser classificadas em métricas baseadas à função, métricas do desenho do sistema, métricas do fluxo de controlo, e métricas de código fonte.

Métricas baseadas à função - são essenciais para a estimativa do projecto e consistem na contagem das características de algumas especificações funcionais tais como número de entidades ou processos.

Métricas do desenho do sistema - são extraídas a partir das notações do desenho tais como gráficos do desenho. São calculadas através da contagem, por exemplo, do número de chamadas que um módulo realiza, o número de subordinados que um módulo tem. Estas medidas podem ser usadas, por exemplo, como uma medida de quantidade de acoplamento, coesão, "fan-in" e balanceamento.

Métricas do fluxo de controlo - são extraídas a partir do desenho detalhado e código do programa fonte, através da contagem das características no fluxo de controlo dum programa ou módulo, tais como o número de declarações de decisões; estas métricas são úteis na medição da legibilidade e testabilidade dos módulos.

Métricas do código fonte - são extraídas a partir da contagem das características do código fonte, na fase da implementação (programação), tais como o número de identificadores, operandos e constantes. Estas métricas, por exemplo, servem para medir a legibilidade dos programas.

O objectivo principal na identificação da métrica é encontrar medidas que indicam algo sobre o projecto que se estiver a gerir. Por isso a definição duma métrica deve conter o objecto e o motivo da medição. Estas métricas do software podem permitir a

comparação, classificação e análises matemáticas que podem ser feitas sobre o software.

3. Sistema de Gestão de Qualidade

Nesta parte do trabalho é apresentada uma descrição do Sistema de Gestão de Qualidade (SGQ) proposto para uma organização produtora de software, tendo como base a ISO[5]. A ISO define o Sistema de Gestão de Qualidade como sendo a estrutura organizacional, procedimentos, processos e recursos para o desempenho das responsabilidades de gestão de qualidade.

Um Sistema de Gestão de Qualidade é composto principalmente por: Controlo de Qualidade, Garantia de Qualidade e Gestão de Qualidade (Fig.3).

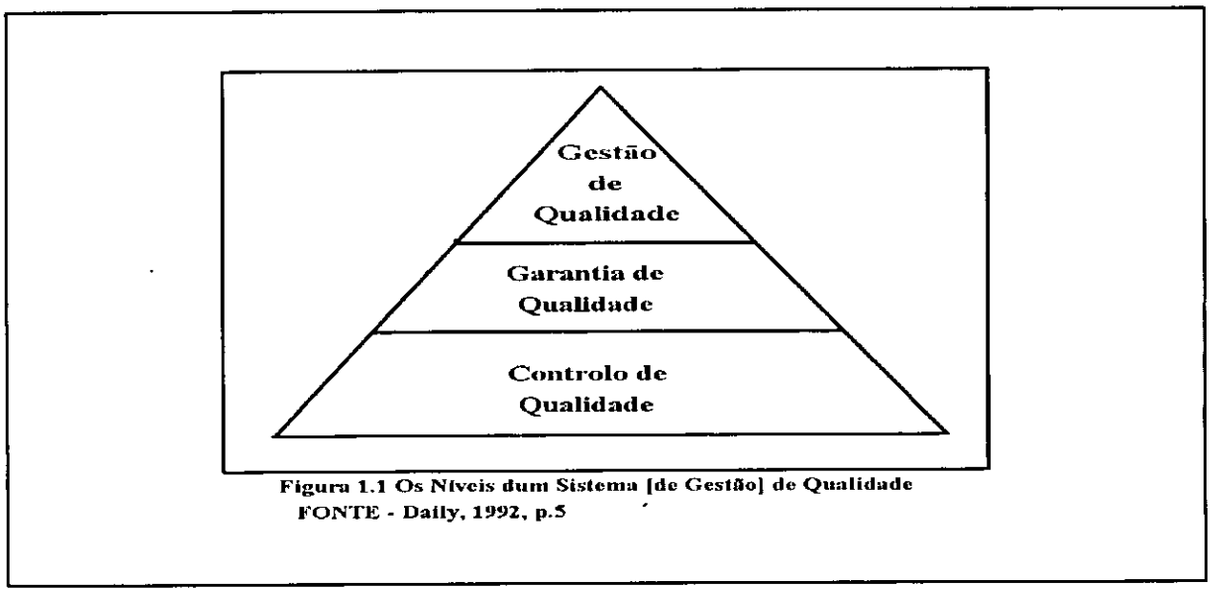

Figura 1.1 Os Níveis dum Sistema [de Gestão] de Qualidade
FONTE - Daily, 1992, p.5

Figura 3 - Níveis dum Sistema de Gestao de Qualidade

Um Sistema de Gestão de Qualidade deve ser definido e gerido em função dos objectivos gerais da organização onde o sistema pertence bem como avaliado

[5] International Standards Organization (Organização Internacional de Normalização).

sistemáticamente, corrigido e melhorado sempre que for necessário. Este sistema deve também ser capaz de adaptar-se a um novo projecto dependendo das necessidades de satisfação da organização, dos utilizadores e da expansão da organização produtora no mercado económico competitivo.

Cada projecto tem os seus objectivos específicos e suas formas de gestão para atingir esses objectivos de forma efectiva. Um Sistema de Gestão de Qualidade, ao adaptar-se a um projecto, deve ter em conta os princípios gerais de Gestão de Projecto que segundo Blum (1992) são: a compreensão do objectivo, metas e restrições existentes, a identificação e planeamento das actividades; monitorar, manter e fazer ajustamentos do plano, preservação dum ambiente de trabalho calmo, produtivo e positivo. O plano pode ser global ou específico. O plano de qualidade é um exemplo de plano específico.

É da responsabilidade do gestor de projecto a elaboração de um plano de qualidade do projecto[6] que seja compactível com o plano do projecto, que defina os métodos, práticas, normas e procedimentos a serem aplicados ao projecto; e durante a execução do projecto, a realização de revisões regulares do projecto, avaliação dos custos, tempo e o cumprimento do projecto em relação ao plano de qualidade aprovado.

Uma organização com um SGQ deve ter as práticas e os processos de trabalho usados para a produção do software escritos de forma compreensível e viável. Desta forma, é possível conseguir que as pessoas sigam o mesmo procedimento ao realizarem o mesmo tipo de trabalho, que se avalie e se aprenda com facilidade o processo de trabalho, permitindo consultas, difusão e revisões eficientes, além de serem úteis para comparações das práticas de trabalho presentes e anteriores, permitindo melhorias apropriadas.

[6] O gestor da qualidade, quando existir, participa na elaboração do plano da qualidade e avaliação do seu cumprimento.

Existem várias maneiras de definir práticas de trabalho representadas por termos, tais como procedimentos, normas (padrões), código de práticas, instrução de trabalho e guia de trabalho. Embora estes termos não sejam usados consistentemente, juntos representam dois tipos de práticas: obrigatórias e opcionais.

São tidos como documentos obrigatórios os procedimentos, normas e instruções de trabalho, que são vitais para o Sistema de Gestão de Qualidade pois cobrem todas as práticas de trabalho usadas. São considerados como documentos opcionais o código das práticas e guias de trabalho pois fornecem informação suplementar e consultiva para os documentos obrigatórios.

Estes documentos devem estar disponíveis para o uso, actualização e remoção quando forem obsoletos. O seu conteúdo deve ser claro, conciso e escrito num certo nível de detalhe em função daqueles que o usam.

3.1. Controlo de Qualidade

Uma das actividades dum Gestor de Qualidade é verificar se os componentes, e as saídas, de cada fase de desenvolvimento, satisfazem os requisitos e estão livres de erros ou defeitos. Isto constitui, *Controlo de Qualidade*, que a ISO define como sendo um conjunto de técnicas e actividades operacionais que são usadas para cumprir com os requisitos de qualidade.

Sendo o ciclo de desenvolvimento de sistemas de informação/software de aplicação composto por várias fases e que cada fase tem as suas respectivas entradas e saídas, devem ser inclusas, em cada uma destas fases, actividades e técnicas apropriadas. Em cada uma destas actividades devem ser identificadas as pessoas que participam, o processo seguido, os produtos verificados, o resultado da actividade e acções tomadas em relação ao resultado. Estas actividades devem constar num plano de qualidade.

Algumas técnicas de avaliação são as seguintes: auditoria, revisão, inspecção, revisão estruturada e teste. Nestas, existem técnicas de gestão para avaliar o progresso em relação ao plano do projecto e outras para detectar defeitos e problemas.

Em função do pessoal envolvido, a avaliação pode ser classificada em: auto-avaliação, avaliação "peer" e avaliação externa. Na *auto-avaliação* o autor ou produtor avalia o produto do seu próprio trabalho. Na *avaliação "peer"*, um software ou produto duma das fases do ciclo de desenvolvimento é avaliado por uma outra pessoa independente, mas técnicamente equivalente ao produtor. Se for alguém organizacional ou financeiramente independente a avaliar o software ou produtos duma das fases, então, este tipo de controlo designa-se por *avaliaçao externa*.

A auto-avaliação é a prática mais aplicada nas organizações abrangidas pelo estudo, devido aos custos aceitáveis que esta prática proporciona às organizações, pois não exige a contratação de pessoal, como podia acontecer para a avaliação externa. É a auto-avaliação que fornece menos evidência de qualidade ao cliente pois o autor do produto, por uma questão de orgulho e/ou auto-confiança pode não conseguir detectar ou mencionar alguns erros, devido à subjectividade.

De acordo com os métodos de avaliação da qualidade do software, Perry[7], citado por Rocha (1990), sugeriu a seguinte classificação: Avaliação por julgamento, avaliação por critérios de ausência e avaliação através de medidas.

Avaliação por julgamento - Um membro da equipa de controlo de qualidade emite uma opinião sobre a presença ou não da qualidade num produto, baseando-se na sua experiência.

Avaliação por critérios de ausência - Um membro da equipa de controlo de qualidade procura a presença ou ausência de critérios específicos de qualidade no software.

[7] PERRY, W. E.; *Effective Méthods of EDP Quality Assurance*; Prentice-Hall, 1983.

Avaliação através de medidas - O resultado da avaliação são medidas quantitativas das características de qualidade préviamente determinadas.

A aplicabilidade dum tipo de avaliação depende do domínio da mesma por parte da equipa de avaliação, do tipo do projecto, e do produto da fase em avaliação. A forma mais usada nas organizações é a avaliação por julgamento, e na EXI, além desta, aplica-se a avaliação por critérios de ausência, pois existem algumas normas pre-definidas que os analistas/programadores devem seguir, apesar de não existirem equipas formais de controlo de qualidade.

Para a realização efectiva dum controlo de qualidade é necessário a elaboração dum plano de qualidade, identificação das técnicas de avaliação e as pessoas capazes de as utilizar, atribuir responsabilidades, autoridade e as relações entre estas pessoas, verificar se as técnicas de avaliação identificadas são compactíveis com o ciclo de vida de desenvolvimento, definir os critérios de aceitação dos produtos, e fornecer meios que proporcionam a evidência da realização do controlo, que são através dos registos de qualidade.

3.1.1. Técnicas de avaliação

Nesta parte são descritas algumas das técnicas de avaliação usadas nas organizações produtoras de software.

3.1.1.1. Revisão

A *revisão* é um termo abrangente, que pode significar algumas vezes eventos técnicos de avaliação ou eventos da gestão (Daily, 1992). Numa revisão participam alguns elementos principais do projecto tais como o Gestor de Projecto, chefe da equipa, produtores do software, o utilizador ou seu representante, e uma representação do pessoal da Garantia de Qualidade.

Em função disso, a revisão pode classificar-se em: revisão técnica, de gestão e de qualidade.

- *Revisão técnica* - Avalia, num projecto, os produtos ao fim duma determinada actividade. Exemplos de revisões técnicas são: inspecção e revisão estruturada.

- *Revisão de gestão* - Avalia o progresso e o estado do projecto em relação aos planos.

- *Revisão de qualidade* - avalia o cumprimento do projecto em relação aos requisitos definidos no SGQ tais como, planos da qualidade, normas e procedimentos. Deve assegurar a realização das revisões técnicas programadas.

3.1.1.2. Inspecção

A *inspecção* é uma técnica de avaliação baseada numa leitura passo a passo do produto duma fase do ciclo de desenvolvimento. A avaliação é realizada comparando o produto com uma lista de critérios (baseados no plano de qualidade e os requisitos iniciais) e é mais adequada para as etapas iniciais de desenvolvimento do software em que os principais produtos são os documentos, como por exemplo, especificação dos requisitos, os fluxogramas e os planos de teste.

Uma equipa de inspecção pode estar composta por um moderador, o autor do produto, o examinador, o cliente/utilizador e o pessoal de garantia da qualidade (Daily, 1992), sendo no mínimo, composta pelo autor e pelo examinador.

Para a realização duma inspecção, exige-se, de forma sequêncial: preparação, reunião, e, investigação e manipulação dos resultados da reunião. Um dos requisitos para a realização da inspecção é a existência de critérios de inspecção que derivam a partir do plano da qualidade do projecto.

Na *fase de preparação* cada participante recebe o convite para participação e as cópias dos produtos por avaliar, avalia a disponibilidade em participar na reunião e o papel a desempenhar, avalia o produto em relação aos requisitos e as normas aplicáveis, e elabora uma lista de defeitos e problemas detectados.

Na *fase da reunião*, o autor do produto descreve o contexto e as actividades para a obtenção do produto que sejam importantes para a inspecção. Ao longo da explanação, os outros participantes vão apresentando as suas dúvidas, defeitos e problemas por eles detectados na fase da preparação.

Todos os resultados obtidos, tais como, pontos incorrectos a corrigir, aspectos a aperfeiçoar, pontos duvidosos a serem examinados e pontos aprovados sem consenso são reunidos num laudo de avaliação.

Na *fase de manipulação* dos resultados da reunião, o autor realiza a investigação e a correcção de defeitos tal e qual como foram definidos no laudo.

3.1.1.3. Auditoria

A *auditoria* é uma técnica de avaliação externa que consiste na colheita e análise da informação seguida da apresentação de recomendações para a tomada de decisões, e é, segundo Daily (1992), conduzida por uma pessoa ou grupo totalmente independente que seja financeiramente separado do projecto ou organização a ser avaliada.

A auditoria de qualidade avalia o grau do cumprimento, por exemplo, dos requisitos dum Sistema de Gestão de Qualidade tais como normas, procedimentos e planos de qualidade.

Para a realização duma auditoria eficiente, exige-se o treinamento do pessoal como auditores e a definição dos procedimentos a seguir durante a execução da auditoria.

O último aspecto, garante a obtenção de conclusões idênticas ao realizar-se a mesma auditoria por pessoas diferentes.

As auditorias, em função do objecto de avaliação, podem ser classificadas em: auditorias dirigidas aos projectos específicos e as auditorias dirigidas aos Sistemas de Gestão de Qualidade.

As *auditorias dirigidas aos projectos específicos* são necessárias quando o cliente/utilizador pretende uma evidência específica e sistemática do desenvolvimento do produto. Este tipo de auditoria pode ser para avaliar se um determinado software está fazendo o que devia fazer (auditoria funcional), se as diferentes representações do produto são idênticas em termos funcionais (auditoria física) e auditoria dos resultados de cada fase para acompanhar a evolução do projecto (auditoria no processso).

As *auditorias dirigidas aos Sistemas de Gestão de Qualidade* constituem um meio formal de avaliação do Sistema de Gestão de Qualidade em relação aos requisitos duma norma escolhida, que para o software, pode ser, por exemplo, ISO 9000-3, que é uma Regra ("Guidelines") da Norma ISO 9000.

As auditorias dirigidas aos Sistemas de Gestão de Qualidade podem subdividirem-se em internas e extrenas.

Uma *auditoria interna* avalia:

- o cumprimento dum Sistema de Gestão de Qualidade relativamente aos requisitos internos da organização documentados no manual de qualidade e relativamente aos requisitos duma norma escolhida pela organização (se tiver escolhido) na fase de avaliação do SGQ, durante a introdução do mesmo (mais detalhes no ponto II.3.4);

- a conveniência e efectividade/eficácia dum Sistema de Gestão de Qualidade em relação aos requisitos da organização; e

- a realização de acções recomendadas por auditorias anteriores.

Uma *auditoria externa* avalia o Sistema de Gestão de Qualidade duma organização em relação aos requisitos duma norma, examina o manual da qualidade, procedimentos documentados e os registos de qualidade; e verifica a evidência da auditoria interna efectiva. Este tipo de auditoria é realizada por um Grupo de Certificação quando uma organização pretende obter um registo formal do seu Sistema de Gestão de Qualidade.

3.1.1.4. Revisão estruturada

A *revisão estruturada* é uma técnica de avaliação utilizada para detectar erros no produto e com maior aplicabilidade nas fases iniciais do projecto de desenvolvimento do software, e é mais apropriada, segundo Rocha (1991), para avaliar documentos e códigos. Esta técnica difere da inspecção pois enquanto a revisão estruturada tem como finalidade detectar erros, a inspecção é para avaliar traços qualitativos do sistema.

Com esta técnica de avaliação pode-se detectar áreas:

- onde as normas não foram usadas, mas deviam ser usadas;
- com algorítmos ineficientes; e ou
- dum programa técnicamente correcto, apresentando, contudo, problemas futuros de manutenção (Kendall, 1992).

Uma revisão estruturada é realizada por uma equipa composta por um apresentador, um coordenador, um relator, um crítico de manutenção, um crítico de normas e um ou mais representantes do utilizador/cliente (Kendall, 1992).

A responsabilidade dos participantes duma revisão estruturada é identificar erros, que são registados num relatório da revisão estruturada pelo relator e assinado por todos.

Uma revisão estruturada pode ser realizada em qualquer uma das etapas do projecto, desde que estejam documentadas.

3.1.1.5. Teste

O *teste*, segundo Rocha (1990), é uma técnica de avaliação realizada através da execução experimental do software.

Os programas de aplicação elaborados devem ser testados antes de serem entregues aos clientes. Os testes podem ser estruturais e/ou funcionais. Um teste é *estrutural* se for realizado aos programas de acordo com a forma como foram construídos e é *funcional* se for para determinar como os programas são em relação ao que deviam fazer.

Kendall (1992) identificou dois métodos de efectuar o teste:

- *método tradicional do teste*

  Cada programador testa o produto do seu trabalho. Depois de todos os módulos serem testados com êxito, são agrupados formando um sistema, e realiza-se o teste do sistema resultante.

- *método incremental do teste*

  Neste método, depois dum módulo ser bem sucedido no teste isolado, é anexado ao outro grupo de módulos, préviamente testados, com que se relaciona, e é testado o novo grupo formado.

O método incremental exige maior planeamento que o tradicional, pois, a junção que se deve fazer entre um módulo com um grupo existente depende das relações de interdependência existentes entre os módulos.

O método incremental, por sua vez, subdivide-se em "top-down", "bottom-up" e misto. O *método incremental "top-down"* é realizado começando dos grupos mais completos/complexos de instruções (subsistema) para os grupos mais simples (módulos ou unidades).

O *método incremental "bottom-up"* é realizado começando dos módulos com funções mais elementares para os módulos com funções mais complexas.

O *método incremental misto* é efectuado por duas equipas. Uma equipa começa dos módulos com funções mais complexas, módulos do topo dum fluxograma, e a outra equipa começa dos módulos mais elementares, módulos da base, e, encontram-se numa posição média do fluxograma, definida no plano de qualidade.

Dos três métodos incrementais, o método incremental misto exige maior planeamento, e o método a escolher depende das áreas de maior risco identificadas.

Em função das etapas de desenvolvimento dum sistema de software e a estrutura básica do mesmo, os testes podem ser: teste da unidade, teste do módulo, teste do subsistema, teste de integração, teste de aceitação e teste paralelo (descritos no ANEXO B).

Destes tipos de teste, o mais frequente nas organizações abrangidas pelo estudo, é o teste de aceitação. Assim acontece, sem prejuízos para as organizações, por causa da simplicidade dos programas desenvolvidos e o número reduzido de pessoas envolvidas na programação (um programador é o mais corrente).

3.1.2. Comentários gerais sobre as técnicas de avaliação

Ao aplicar-se uma técnica de avaliação obtém-se um resultado que pode condizer ou não com os requisitos. Em função do resultado obtido, toma-se uma certa acção. Se

o resultado for positivo, prossegue-se com a fase seguinte do ciclo. Mas, se for negativo, toma-se uma das seguintes opções:

- corrigir o defeito;
- aceitar o produto sem trabalho adicional;
- rejeitar o produto; e
- aceitar parcialmente, com certos defeitos a serem submetidos à correcções.

A aplicabilidade de cada uma dessas técnicas de avaliação depende do tipo do produto a ser avaliado em cada uma das fases do ciclo de vida do desenvolvimento do software, do grau da exigência da qualidade do produto, da disponibilidade de recursos capazes de aplicarem a técnica com êxito.

3.2. Garantia de Qualidade

Na produção do software devem existir certos aspectos a considerar de forma a estabelecer um ambiente de confiança entre o produtor/fornecedor e o cliente/utilizador, com vista à *Garantia de Qualidade*, que segundo a ISO significa:

> *"todas as acções planeadas e sistemáticas necessárias para fornecer confiança adequada de que um produto ou serviço satisfará os requisitos determinados para a qualidade."*

A interpretação desta definição e a sua relação com o Controlo de Qualidade, conduz à seguinte afirmação: num Sistema de Gestão de Qualidade deve existir um conjunto de actividades que garantam que as técnicas de avaliação pertencentes ao controlo de qualidade, sejam efectivas.

Algumas das actividades principais da Garantia de Qualidade são: realizar auditoria do SGQ e do projecto, monitorar o projecto, e analisar os requisitos de qualidade.

Além disso, é necessário o envolvimento do cliente (ou seu representante) e a participação do pessoal de Garantia de Qualidade (ou pessoal da organização produtora do software técnicamente familiarizado com o tipo de trabalho mas independente ao projecto) no controlo de todo o trabalho de desenvolvimento do software.

O pessoal de Garantia de Qualidade fornece o relatório ao Gestor de Qualidade sobre a realização do trabalho e os métodos usados, e verifica se o Controlo de Qualidade tem sido feito. Eles preocupam-se pela qualidade e aceitabilidade do produto final.

Para uma execução eficiente do trabalho do pessoal de Garantia de Qualidade, é necessário, antes de mais nada, a elaboração e revisão do plano da qualidade do projecto, que poderá, também, estar disponível ao cliente. As actividades constantes num plano de qualidade variam em função das metodologias e modelos utilizados. Num modelo espiral exige-se um planeamento mais rigoroso em relação ao modelo linear.

Esta participação conjunta do pessoal de Garantia de Qualidade, o produtor e o cliente, garante a identificação dos problemas e erros o mais cedo possível assim como a tomada de acções correctivas. A observação feita na forma como o trabalho é feito, comparação do trabalho feito em relação ao esperado, e o estabelecimento dum nível de colaboração entre o fornecedor e o cliente, fornece a confiança desejada entre os intervenientes no desenvolvimento do software.

3.3. Gestão de Qualidade

A *Gestão de Qualidade* é uma das componentes do Sistema de Gestão de Qualidade, na qual se encontram definidos e documentados os objectivos, estruturas, e responsabilidades da gestão dos produtos e/ou serviços, apresentados aos vários níveis da organização através do documento designado por manual de qualidade. O manual de qualidade é um dos documentos existente no Sistema de Gestão de

Qualidade que demonstra como a política de qualidade da organização é alcançada na prática.

A ISO define a *Gestão de Qualidade* como sendo a função de gestão global que estabelece e implementa a política da qualidade.

A Gestão de Qualidade, inserida nas actividades da organização, deve estar em concordância com as responsabilidades, estrutura e objectivos de gestão geral da organização. As responsabilidades e autoridades da Gestão de Qualidade são definidas dentro da estrutura organizacional e são dirigidas pelo Gestor de Qualidade designado pela organização.

A *Política de Qualidade* é a forma de dirigir, e faz uso de um conjunto de indicações e procedimentos que orientam a organização para o cumprimento dos objectivos relacionados com a qualidade, formalmente expressos pela gerência.

Os requisitos básicos da Gestão de Qualidade, segundo Daily (1992), são:

- definição e documentação da política e objectivos da qualidade;
- identificação e documentação dos procedimentos e práticas de trabalho;
- revisão periódica da Gestão de Qualidade;
- auditorias internas de qualidade; e
- acções correctivas.

A identificação e documentação dos procedimentos das práticas de trabalho específicas permite garantir a boa comunicação entre os membros da organização produtora ou do projecto do software, e serve de base para a avaliação e aprendizagem do processo. Estes procedimentos e práticas de trabalho devem constar num plano da qualidade e, devem ser definidas as formas como eles serão demonstrados e avaliados.

Além do manual de qualidade e planos de qualidade[8] existem manuais de procedimentos (ANEXO C).

A revisão periódica da Gestão de Qualidade é realizada pelo Gestor de Qualidade na base da informação de auditoria interna do Sistema de Gestão de Qualidade num determinado período. As auditorias também são feitas na base de um programa préviamente definido pelo gestor de qualidade.

A realização duma série de auditorias internas da qualidade assegura que o Sistema de Gestão de Qualidade seja cumprido, efectivo, e permite a identificação das áreas que exigem alteração e/ou melhoramento.

Na Gestão de Qualidade não só se pretende utilizar as técnicas de avaliação disponíveis para se identificar os problemas e os erros ao longo do desenvolvimento do software como também se pretende tomar acções correctivas.

3.4. Como Introduzir um Sistema de Gestão de Qualidade numa Organização ?

Não existe uma resposta categórica sobre esta pergunta. Contudo, pode-se responder tomando em conta que o Sistema de Gestão de Qualidade é um sistema de informação que obedece a um determinado ciclo de vida de desenvolvimento. Um Sistema de Gestão de Qualidade pode ser introduzido como um projecto ou uma introdução gradual segundo a prática quotidiana. Para a introdução dum Sistema de Gestão de Qualidade é necessário que exista uma autorização por parte da direcção da organização.

Os factores a considerar para a introdução dum Sistema de Gestão de Qualidade numa organização produtora de software são: a natureza do mercado, cultura nacional e da organização, a percepção da qualidade que têm as pessoas envolvidas

[8] Este plano deve estar conforme o plano do projecto.

na transacção do software e as existentes no meio ambiente da organização. Estes factores são dependentes dos sistemas políticos, económicos, traços culturais e ambiente do mercado económico.

Quando a introdução do Sistema de Gestão de Qualidade é encarada como um projecto, além da autorização e compromisso da direcção pelo projecto, é necessário identificar um gestor do projecto que deve ser cuidadosamente escolhido. Há que se elaborar e acordar um plano do projecto, e controlar o progresso do projecto em relação ao plano do projecto elaborado. Depois da introdução do Sistema de Gestão de Qualidade e quando este estiver operacional, é necessário garantir a manutenção e o melhoramento do mesmo em função do crescimento da organização.

O projecto da introdução do Sistema de Gestão de Qualidade é melhor realizado e controlado quando dividido em etapas que permitam a sua avaliação.

O número de etapas em que se divide o projecto depende da realidade de cada organização, mas duma forma geral, segundo Daily (1992), pode-se ter as seguintes etapas:

Etapa 1: Preparação e Planeamento,
Etapa 2: Introdução,
Etapa 3: Operação (execução do Sistema de Gestão de Qualidade),
Etapa 4: Avaliação, e
Etapa 5: Melhoramento.

Etapa 1: Preparação e Planeamento

A etapa da preparação e planeamento pode ser realizada em duas fases: fase da preparação e a do planeamento. Na fase da preparação informa-se ao pessoal, sobre o compromisso e a autorização da direcção em relação à introdução do SGQ, decide-se o âmbito e os objectivos do Sistema de Gestão de Qualidade e as áreas da implementação. Feito isto, define-se uma política de qualidade, que constitue o início

da elaboração dum manual de qualidade, que é um documento do nível da organização.

Na fase do planeamento, produz-se um plano geral para a introdução, e demonstra-se a viabilidade do projecto em termos de tempo e custo; identifica-se a forma de implementação, se será global ou parcial, usará um protótipo ou não. Desta forma são introduzidas gradualmente as técnicas específicas e algumas práticas do Sistema de Gestão de Qualidade.

No fim desta primeira etapa, os objectivos e o âmbito do Sistema de Gestão de Qualidade devem estar definidos, acordados e viáveis; todo o pessoal afecto pelo Sistema de Gestão de Qualidade contactado; o custo e tempo aceitáveis, resultados e benefícios esperados identificados e aceitáveis, e por último ter recursos disponíveis. No fim desta fase deve-se ter um plano detalhado do projecto para as fases seguintes.

Etapa 2: Introdução

Esta etapa requere mais atenção e envolvimento do Gestor do Projecto. O Gestor do Projecto deverá escolher a melhor forma para a introdução, se é por unidades organizacionais/projectos ou de forma global, deve verificar se o tempo é viável para o efeito e quais as necessidades de formação do pessoal. Deverá também identificar as áreas onde existam pessoas capazes de elaborarem seus próprios procedimentos e normas.

É necessário controlar o progresso do projecto em relação as outras prioridades da organização que o possam afectar.

Etapa 3: Operação

É a etapa em que o Sistema de Gestão de Qualidade introduzido, opera sob uma supervisão ou auditoria para detectar e corrigir problemas/erros ou para confirmar que

está operando correctamente. O tempo da supervisão e auditoria dependerá da complexidade do Sistema de Gestão de Qualidade.

Nesta etapa é necessário o envolvimento dos clientes e fornecedores para que tenham o conhecimento sobre a introdução do Sistema de Gestão de Qualidade. Nesta etapa, o Sistema de Gestão de Qualidade deve estar estável até ao momento da sua avaliação; todos os projectos, as operações e unidades organizacionais abrangidas pelo SGQ devem estar operando dentro do seu âmbito. Nesta etapa, deve-se saber se o Sistema de Gestão de Qualidade é aceite perfeitamente em todas unidades organizacionais/projectos ou não, e começar a avaliar os benefícios resultantes da aplicação do Sistema de Gestão de Qualidade.

Etapa 4: Avaliação

Esta é uma etapa em que o Sistema de Gestão de Qualidade é submetido a uma avaliação externa, em relação a uma norma que a organização tenha escolhido (por exemplo BS 5750[9], EN 29000[10] AQAP1[11] e ISO 9000[12]), por meio dum Grupo de Certificação (por exemplo BSI[13], ANSI[14], e ISO), dando maior credibilidade ao Sistema montado. Após a avaliação, se o SGQ for aprovado, o Grupo de Certificação emite um certificado de qualidade.

Existem normas gerais, isto é, aplicáveis a vários tipos de produtos, e normas específicas, aplicadas a um tipo específico de produto.

[9] É uma Norma Britanica.

[10] É uma Norma Europeia.

[11] É uma Norma da NATO.

[12] É uma Norma Internacional.

[13] British Standards Institution.

[14] American National Standards Institution.

As normas usadas para uma determinada aplicação são extraídas a partir duma norma "mãe". Por exemplo na ISO 9000, tem-se a ISO 9000-1 que é constituida por conceitos básicos e guias para a selecção e uso. Para o caso do software existe a Regra (Guidelines) ISO 9000-3 (ANEXO D) onde estão definidas as formas para o desenvolvimento, fornecimento e manutenção do software.

Em Moçambique ainda não existe em uso uma norma, tanto geral como específica, que oriente as actividades de gestão de qualidade do software. As organizações abrangidas neste estudo não possuem algum SGQ, consequentemente nunca foram certificadas por alguma norma de qualidade.

Etapa 5: Melhoramento

O melhoramento é uma necessidade das organizações contemporâneas para que o seu Sistema de Gestão de Qualidade se adapte à realidade que está evoluindo, e pode ser realizado à medida que as necessidades forem surgindo através do uso das técnicas de controlo de qualidade nos projectos de desenvolvimento do software. Se se obtiver um resultado errado, ao usar-se uma determinada técnica de avaliação, procura-se saber as causas do erro e como corrigi-lo. Assim, estar-se-á a melhorar o Sistema de Gestão de Qualidade.

O melhoramento da qualidade do software deve ser uma actividade contínua e resultante da maneira como se age ao constatar-se um problema e/ou erro durante a avaliação dum produto do software em desenvolvimento. Assim, garante-se o melhoramento de algumas práticas de trabalho no Sistema de Gestão de Qualidade.

Outra forma de melhoramento pode ser realizada através da actualização das técnicas de controlo de qualidade, sempre que for necessário, ou a realização de revisões periódicas ao Sistema de Gestão de Qualidade.

Uma organização produtora de software, em expansão e com intenção de competir no mercado de vendas dos seus produtos, necessita de ter um programa contínuo de

melhorias da qualidade dos seus produtos de software. Este programa deverá induzir, ao longo do tempo, a todo o pessoal da organização ligado com a produção do software a mudar a sua forma cultural em relação a qualidade, e, a visão do cliente em relação à mesma.

Esta etapa do envolvimento de todo o pessoal para a qualidade é que é designada, por Daily (1992), de Gestão de Qualidade Total. Um Programa de Qualidade Total é um processo dinâmico de melhoramento contínuo em comparação com o Sistema de Gestão de Qualidade que é relativamente estático, e é o nível para o qual a organização deve tender atingir.

Uma das formas de introduzir melhorias da qualidade é entender que os defeitos encontrados pelas técnicas de controlo da qualidade indicam necessidades da realização de alterações ou melhorias. E que estas acções correctivas e de melhorias da qualidade implicam custos.

As equipas de melhoramento e de acções correctivas devem identificar, avaliar e ponderar estes custos das acções correctivas e de melhorias, e o tempo definido para estas operações em relação a sua rentabilidade económica e financeira.

Estando a qualidade do software associada ao factor custo, é importante que se determinem estes custos.Um *custo da qualidade do software* é um custo que resulta da aplicação das acções para obtenção dum nível satisfatório dum software com qualidade desejada. Estas acções podem ser de gestão, técnicas e de suporte.

Os custos da qualidade podem ser classificados em:

- custos de falhas,
- custos de avaliação, e
- custos de prevenção.

Os *custos de falhas* resultam do trabalho das acções correctivas que são realizadas quando é detectado um problema e/ou erro, assim como, podem resultar da rejeição do próprio produto. São custos negativos que devem ser minimizados pois reduzem os lucros tanto para o produtor assim como para o cliente. Os *custos de avaliação* resultam da implementação das técnicas de avaliação e tendem a reduzir os custos de falhas. Os *custos de prevenção* são os que resultam da realização das actividades para a prevenção de erros/problemas.

Daily (1992) considera que um dos objectivos dum programa de melhoramento é a alteração da balança dos custos da qualidade de forma que em vez de se incorrer a custos de falhas não controláveis e periódicos, os recursos sejam aplicados na avaliação e prevenção, e eventualmente, no melhoramento.

III. GARANTIA E CONTROLO DE QUALIDADE NUM CICLO DE DESENVOLVIMENTO DO SOFTWARE DE APLICAÇÃO

Tendo em conta que um dos objectivos deste trabalho é descrever como se pode garantir a qualidade num ciclo de vida de desenvolvimento do software de aplicação usando o Método Orientado a Objecto, esta parte do trabalho vai se restringir na gestão de qualidade no Método Orientado a Objecto assente no modelo espiral (ANEXO E).

Como não foi possível observar um sistema em desenvolvimento usando o método e o modelo em causa cumprindo com alguns dos requisitos dum SGQ assente na ISO 9000-3 (ANEXO D), é utilizado um exemplo imaginário, apresentado no "APÊNDICE I: 1", que serve de suporte para o estudo.

1. Método Orientado a Objecto

Considerando que o Método Orientado a Objecto é um conceito recente, convém iniciar por uma pequena apresentação.

Para o desenvolvimento do Software Orientado a Objecto (SOO) recorre-se a um dos Métodos Orientado a Objecto pois existem diferentes métodos com diferentes notações, formas de implementação, estratégias de desenvolvimento, ferramentas e técnicas utilizadas, produtos finais do método e formas de gestão do projecto.

Um Método Orientado a Objecto (MOO), assenta nos elementos designados por modelos de objectos. O modelo de objecto guia-se pelos princípios de abstracção, encapsulamento, modularidade, hereditariedade, classificação, concorrência, persistência e polimorfismo.

Abstracção - indica as características essenciais dum objecto que o distingue de outros objectos. Esta distinção permite identificar as fronteiras conceptuais em relação à perspectiva do observador.

Encapsulamento - é o processo de encobrimento de todos os detalhes dum objecto que não contribuem para as suas características essenciais. Esta característica melhora a manutenibilidade do sistema.

Modularidade - é o princípio em que um sistema se decompõe num conjunto de módulos coesos e folgadamente acoplados.

Hereditariedade - é o princípio de organização de objectos e classes em estruturas hierárquicas nas quais os objectos do nível mais baixo podem herdar propriedades dos objectos do nível mais alto. É uma sequência de abstracções.

Classificação - significa que os objectos de diferentes tipos não podem ser permutados, ou quando muito, eles podem ser permutados apenas numa forma muito limitada.

Concorrência - é o princípio que permite a distinção entre um objecto activo e um objecto não activo.

Persistência - é o princípio dum objecto através do qual sua existência transcende o tempo e/ou espaço, isto é, o objecto continua a existir depois do seu criador cessar de existir.

Polimorfismo - é a capacidade de duas ou mais classes responderem ao mesmo pedido, cada uma das classes da sua maneira.

Com base nestes princípios é que são identificados os objectos num sistema, são agrupados os objectos semelhantes para formarem classes de objectos, são estabelecidas as relações entre objectos, entre classes e objectos, e entre classes.

Objecto - é uma abstracção de algo num domínio do problema. O objecto tem estado, comportamento e identidade.

O estado do objecto é o conjunto de propriedades do objecto e os valores correntes que as propriedades assumem. O comportamento é a forma como um objecto actua e reage devido a uma mudança de estado quando passa uma messagem. Um objecto reage quando sobre ele houver uma acção (designada por operação). A implementação dessas operações é designada por métodos em algumas linguagens de programação como é o caso de Pascal.

Classe - segundo Coad e Yourdon[15], citado por Yourdon (1994), é uma colecção de um ou mais objectos com um conjunto uniforme de atributos e serviços, incluindo a forma como criar os objectos na classe.

1.1. Relações entre objectos e/ou classes

Os tipos principais de relações são os seguintes:

- hereditariedade, É_UM(A); e
- agregação, TEM_UM(A).

A hereditariedade permite definir as *relações É_UM(A)*, também, designadas por TIPO_DE ou Especialização/Generalização (GEN_ESP). Exemplo:

- O estudante É_UM(A) pessoa.
- O professor É_UM(A) pessoa.

[15] COAD, Peter e YOURDON, Edward. *Object - Oriented Analysis*, 2nd ed., (Englewood Cliffs, NJ: Yourdon Press/Prentice Hall, 1990). p.53.

A *relação de agregação, TEM_UM(A)*, também designada por PARTE_DE (TODO_PARTE), identifica que um objecto é composto por componentes ou a partir de casos elementares de outras classes. Exemplo:

- O carro TEM_UM(A) rodas.
- A roda TEM_UM(A) parafuso.

O desenvolvimento do software orientado a objecto, utiliza estruturas designadas por objectos, usa os componentes correntes préviamente desenvolvidos e testados, é realizado de forma incremental e interactiva, com refinamento constante dos componentes do sistema (sempre que necessário).

Apesar de existirem diferentes métodos de desenvolvimento do software orientado a objecto, todos eles assentam numa análise funcional, elaboração de um ou mais modelos de objectos e a implementação dos modelos numa linguagem de programação e ambiente.

Exemplo de métodos orientado a objecto, segundo Caspers (1994), são:

- Análise e Desenho Orientado a Objecto do Coad e Yourdon;
- Desenho Orientado a Objecto de Booch;
- Desenho do Software Orientado a Objecto do Wirfs - Brock et al.;
- Técnicas de Modelação do Objecto de Rumbaugh et al.;
- Martin e Odell;
- "Objectory" do Jackson et al.; e
- "Fusion" do Coleman et al..

Todos estes métodos têm como objectivo o desenvolvimento do software com um nível satisfatório de qualidade. O desenvolvimento dum software compreende um ciclo que assenta num modelo. Para este trabalho, o modelo escolhido é o espiral, pois é mais realístico, pois garante o sucesso por avaliar constantemente os objectivos em relação às restrições e às alternativas, requere a identificação dos riscos e sua resolução, e resulta da selecção das vantagens dos modelos linear,

incremental e evolucionário. Num sistema de informação complexo, o modelo espiral é o mais ideal para a garantia de qualidade.

2. Desenvolvimento do Software na Base do Método Orientado a Objecto

No desenvolvimento do software de aplicação utilizando o Método Orientado a Objecto, assente no modelo espiral, obedecendo a ISO 9000-3, deve-se ter:

- as actividades básicas do ciclo de vida exigidas pela ISO 9000-3;
- as actividades de suporte exigidas pela ISO 9000-3;
- a estrutura do modelo espiral; e
- os princípios gerais da gestão do projecto.

Um dos requisitos de gestão é a elaboração do plano de desenvolvimento (APÊNDICE I: 4). No exemplo dado, as durações das actividades foram deduzidas a partir da experiência que o autor tem. A este plano anexa-se o plano da qualidade.

2.1. Ciclo de vida do software de aplicação Orientado a Objecto

Independentemente do método de desenvolvimento utilizado, o ciclo de vida de desenvolvimento do software de aplicação orientado a objecto começa com a definição do problema até à operacionalidade, seguida pela manutenção e/ou inutilização. Dos métodos anteriormente mencionados, alguns abrangem todo o ciclo de desenvolvimento ("Fusion" e "Objectory") e outros alguma(s) fase(s) do ciclo.

As principais fases para o desenvolvimento do software de aplicação são:

- definição do problema;
- estudo de viabilidade;
- análise do sistema;
- desenho do sistema;
- implementação (codificação); e
- operação do software.

2.1.1. Definição do problema

Nesta fase, definem-se os objectivos do projecto, fronteiras do projecto, restrições do projecto (em termos de custos, tempo e qualidade), e os requisitos do utilizador, que são designados por *Termos de Referência do Projecto* (APÊNDICE.I: 2 e 3). A "definição do problema" é um requisito exigido em qualquer projecto de desenvolvimento e é definido em função da dimensão e complexidade do sistema em estudo.

O produto desta fase, Termos de Referência do Projecto, constitue o primeiro documento a ser introduzido no Sistema de Gestão da Configuração.

2.1.2. Estudo de viabilidade

Nesta fase realiza-se o estudo de viabilidade técnica, económica e operacional.
O estudo de viabilidade técnica permite a identificação das tecnologias e habilidades necessárias para a realização do projecto. O estudo de viabilidade económica permite a identificação e determinação das possibilidades de se alcançarem os objectivos do projecto com os recursos alocados. O estudo de viabilidade operacional avalia o nível de satisfação dos objectivos do utilizador com as soluções propostas e a possibilidade do equadramento destas soluções na operação do sistema corrente.

Alguns dos produtos desta fase são: relatórios dos custos/benefícios do sistema e plano mais detalhado das fases seguintes do projecto. Estes produtos são avaliados através da inspecção ou da revisão estruturada em relação aos requisitos do utilizador constantes nos termos de referência do projecto. Estes produtos, depois de serem aprovados, devem ser inclusos no Sistema de Gestão da Configuração. Por exemplo, o plano detalhado da fase de análise do projecto da UPSS (APÊNDICE I: 4.1).

Em relação à viabilidade do projecto e em função da informação contida no "APÊNDICE I. 1. Descrição do sistema" pode-se afirmar que existem tecnologias

disponíveis para a realização do projecto, só nada se pode afirmar em relação à viabilidade económica, porque não se dispõe de dados para serem utilizados como termo de comparação para fornecer uma resposta aproximadamente certa.

Quando se decide pela execução do projecto e pela necessidade de contratação duma equipa externa à organização que necessita do software/sistema, além das actividades já mencionadas, é feita a revisão do contrato com a equipa contratada para o projecto.

Alguns dos items da qualidade a constarem no contrato são: critérios de aceitação e manipulação das alterações dos requisitos do cliente durante o desenvolvimento; actividades do cliente na especificação dos requisitos, instalação e aceitação; ferramentas, facilidades e equipamento informático a ser fornecido pelo cliente; e definição dos riscos.

2.1.3. Análise do sistema

Nesta fase descreve-se como o sistema funciona; identificam-se com mais detalhe os problemas do sistema, as áreas e/ou utilizadores que exigem uma modificação ou introdução dum novo sistema.

Uma das actividades a incluir nesta fase de análise do sistema é a Gestão de Risco (ANEXO E: 2). No projecto da UPSS, são identificados os seguintes factores de risco:

- cálculo das durações das actividades em função da experiência, que é pouca;
- pouca experiência que o autor tem sobre projectos de desenvolvimento de sistemas de informação; e
- restrições impostas pelo equipamento informático a usar.

Para reduzir estes riscos, a opção pode ser a escolha dum protótipo, a 'área da gestão de stock dos produtos, material e equipamento' e, mais tarde, dependendo dos resultados da avaliação, far-se-ia um estudo ao resto do sistema.

As actividades típicas da fase de Análise Orientada a Objecto (AOO) do sistema são as seguintes:

- identificar classes e objectos básicos do modelo;
- identificar e organizar classes e objectos em estruturas;
- identificar e estabelecer relações estáticas entre classes e objectos;
- definir atributos para classes e objectos;
- definir o comportamento dos objectos; e
- identificar os serviços dos objectos.

Identificar classes e objectos básicos do modelo

A identificação de classes e objectos é fundamental para a realização das demais actividades desta fase e das fases seguintes do projecto. A notação usada para a representação das classes e objectos varia de método para método, mas, deve ser rigorosa e formal; evitar ambiguidade; e ser simples.

Para a identificação de classes e objectos são seguidos os seguintes passos:

1º - elaboração duma lista textual de classes e objectos do domínio do problema;
2º - avaliação dos objectos candidatos (APÊNDICE I: 4.2.a); e
3º - agrupamento de classes e objectos em assuntos.

Em projectos complexos, o agrupamento de classes e objectos em assuntos é um dos factores que contribui para a inteligibilidade do modelo de análise.

Identificar e organizar classes e objectos em estruturas

Esta actividade organiza as classes e objectos em estruturas hierárquicas de Generalização_Especialização (GEN_ESP) e de Todo_Parte. Nas estruturas de GEN_ESP aplica-se o princípio de hereditariedade, e as estruturas de Todo_Parte baseiam-se nos componentes do objecto. Os componentes não herdam os atributos ou comportamentos do "TODO".

Identificar e estabelecer relações entre classes e objectos

As classes e objectos existem no sistema de forma conjunta e relacionada. E as relações entre os objectos podem ser estáticas ou dinâmicas e são apresentadas num *diagrama de interação de objectos*. As relações dinâmicas descrevem o comportamento das classes e objectos.

Entre as classes e objectos podem existir os seguintes tipos de relações estáticas:

- muitas para muitas (M:M);
- recursivas;
- de casos múltiplos entre classes; e
- unárias entre classes (1:1).

Definir atributos para classes e objectos

Os *atributos* descrevem o estado do objecto e são manipulados pelos serviços. Não pode existir objecto sem atributos, e quando assim acontecer, deve-se investigar o objecto.

Para a definição de atributos para classes e objectos, foram seguidos os seguintes passos:

1º - identificação dos atributos;
2º - colocação dos atributos numa classe hierárquica; e
3º - avaliação, através da revisão, da estrutura hierárquica (APÊNDICE I: 4.2.b).

Definir o comportamento dos objectos

Enquanto que o atributo é uma característica estática do objecto, o *comportamento* é uma característica dinâmica do objecto. O comportamento é modelado no Diagrama de Transição do Ciclo de Vida do Objecto (CVO).

Um diagrama de transição do CVO caracteriza-se por ter estados, mudanças de estados e, mostra o comportamento do objecto. A utilidade do objecto reside na sua possibilidade de sujeitar-se a diferentes mudanças significativas e válidas de estado.

Um objecto muda de estado quando estiver sujeito a determinadas condições e exige a tomada de certas acções. As acções a tomar, tais como enviar mensagens para outros objectos, produzir uma saída e visualizar uma mensagem no écran, dependem do tipo de mudança do estado.

Identificar os serviços dos objectos

Os *serviços* descrevem o comportamento activo do objecto, e é um processamento realizado por um objecto quando recebe uma mensagem. As mensagens são representadas no Diagrama de Comunicação dos Objectos por setas, que partem do emissor para o receptor.

Existem cinco tipos fundamentais de serviços (Yourdon, 1994):

- serviços implícitos, tais como criar, modificar, eliminar e processar;
- serviços associados com conexões de mensagens;
- serviços associados com conexões de casos elementares ("instances");
- serviços associados com atributos; e
- serviços propostos por diagramas do CVO.

No fim desta fase de Análise Orientada a Objecto ter-se-ia os seguintes documentos: Diagramas de Interacção dos Objectos, Diagramas de Comunicação dos Objectos, e Diagramas de Transição do Estado (Yourdon, 1994).

Estas especificações, depois de serem validadas em relação aos termos de referência (ou requisitos do utilizador) e ao sistema existente, e serem verificadas em relação aos custos/benefícios para a sua implementação e ao plano detalhado do projecto para o seu cumprimento, ficam sob o controlo da configuração do software.

As validações podem ser feitas através da inspecção ou revisão estruturada. No fim da reunião de inspecção ou revisão estruturada é elaborado um laudo que é colocado sob o controlo da configuração.

Quando o resultado da avaliação dos produtos for positivo, procede-se a uma das seguintes acções:

- elaboração do desenho Orientado a Objecto, DOO, quando a opção for para produzir o software; e
- avaliação dos produtos do software disponíveis no mercado, quando a opção for para a compra do software.

Além das actividades acima mencionadas, nesta fase da Análise Orientada a Objecto, é elaborado um plano de teste do sistema em função dos critérios de aceitação identificados a partir das especificações do utilizador.

2.1.4. Desenho do Sistema

O Desenho Orientado a Objecto (DOO) é realizado quando o produto de AOO for aprovado e, a acção a tomar for para a produção do software. O DOO gera o anteprojecto para a implementação do sistema, usa os diagramas resultantes da AOO e recorre a novas abstracções para o refinamento dos produtos da AOO.

O DOO é uma acção incremental e interativa, baseando-se nos protótipos determinados e selecionados, modelando cada um deles uma determinada área ou função importante do desenho do sistema.

O DOO é feito tendo em consideração o hardware, o sistema operativo no qual será implementado e a linguagem de programação que será utilizada. Além destes, deve-se considerar, segundo Yourdon (1994), a notação a usar, as estratégias a seguir para o DOO, e os critérios de avaliação do desenho.

A notação usada nesta fase é a mesma da fase de AOO acrescida de representações das tecnologias. A estratégia do DOO define o começo e o fim desta fase, passos a seguir e a estrutura ou arquitectura final desta fase. Os critérios de avaliação permitem a aceitação ou rejeição do desenho final desta fase.

Arquitectura do desenho

Coad e Yourdon[16], citado por Yourdon (1994), estabeleceram que uma arquitectura do desenho é composta por quatro componentes principais:

- componente da interacção humana (CIH),
- componente do domínio do problema (CDP),
- componente da gestão da actividade (CGA), e
- componente da gestão de dados (CGD).

A CIH contém classes e objectos que fornecem uma visão do interface humano. Alguns dos factores que devem ser considerados para o desenho desta componente são os seguintes: classificação dos utilizadores do sistema, por exemplo, por níveis de habilidade ou níveis organizacionais; descrição dos cenários de actividades dos utilizadores; identificação dos eventos e a sequência das actividades realizadas pelos utilizadores; desenho da hierarquia de comandos que permite a definição dos vários níveis dos menús; desenho dos diferentes mecanismos de interação entre o utilizador e o sistema; e as representações dos GUIs (Graphics User Interface) disponíveis.

Alguns tipos de interfaces humanos são: relatórios, formulários e GUIs. E, algumas das características dum GUI são: os comandos são visíveis o que não exige ao utilizador se recordar da sintaxe duma linguagem; facilidade de aprender um ícon em relação a um comando abstracto; ter um interface interactivo porque fornece um "feedback" contínuo em relação ao interface de "batch"; facilidade de usar o rato ou dedo para escolher uma opção no écran em relação a escrever um comando.

A CDP é um conjunto de classes e objectos que modela o sistema em si e é a componente principal sobre a qual realizam-se as actividades de gestão de dados e interação humana.

[16] COAD, Peter e YOURDON, Edward. *Object - Oriented Analysis*, 2nd ed., (Englewood Cliffs, NJ: Yourdon Press/Prentice Hall, 1990).

A CDP pode estar constituída apenas pelo modelo de análise do sistema assim como exigir algumas alterações do modelo. Algumas das razões que levam à alterações do modelo para torná-lo numa componente do domínio do problema da arquitectura do desenho são: existência de classes e objectos que possam ser reutilizados para o desenho e/ou programação; necessidade de estabelecer protocolos comuns com a CGD ou com outra CDP do sistema externo; necessidade de acomodar princípios de hereditariedade disponíveis numa linguagem de programação; necessidade de melhorar o desempenho.

A CGA é um conjunto de classes e objectos que controla ou coordena o comportamento de outras classes e objectos.

Nem todas as componentes devem aparecer em todos os projectos que utilizam o Método Orientado a Objecto. Por exemplo, para o problema da UPSS, não seria necessário produzir a componente da gestão de dados, visto que esta componente só é necessária quando se pretende estabelecer uma ligação entre uma base de dados relacional com uma base de dados Orientada a Objecto. Segundo a descrição do sistema da UPSS, não existe nenhuma base de dados em uso, pelo que conclue-se que não é necessária alguma ligação. E, por não existir uma base de dados informatizada então, o modelo de análise pode não estar sujeito a muitas alterações e servir como a CDP para a arquitectura do desenho.

Além dos quatro componentes antes mencionados, nesta fase do desenho podem ser fornecidos os seguintes documentos:

- proposta da configuração do equipamento específico para a base de dados,
- procedimentos detalhados do utilizador, e
- o manual do utilizador.

Estes produtos, depois de validados em relação aos requisitos do utilizador e verificados confrontando-os com os resultados da fase de AOO e o sistema existente, são inclusos num sistema de gestão da configuração.

Nesta fase do DOO devem também ser elaborados e/ou detalhados os planos do teste (usualmente são testes de integração e da unidade).

Critérios de avaliação do modelo do DOO

Para a avaliação do modelo do DOO, segundo Coad-Yourdon,[17] citado por Yourdon (1994), dispõe-se de três formas que podem ser usadas de forma isolada ou combinadas. Têm-se a prototipificação, a comparação com arquitecturas do software, e o uso de linhas de orientação.

Através da prototipificação produz-se um protótipo que é entregue ao utilizador juntamente com uma lista de perguntas, por exemplo, para avaliá-lo, e é a melhor forma de avaliar uma CIH. A CIH pode também ser avaliada através de cenários.

Utiliza-se a comparação com arquitecturas do software que constam na norma da organização produtora do software quando estas arquitecturas existirem já definidas na organização.

São utilizadas linhas de orientação, tais como acoplamento, coesão, clareza do desenho, para possuir classes e objectos simples, protocolos de mensagens e métodos simples, e minimizar o tamanho do sistema global.

Para se fazer a avaliação dos aspectos técnicos, pode-se fazer uma inspeção ou revisão estruturada e, o laudo de inspecção ou revisão estruturada do desenho é colocado sob a gestão da configuração.

2.1.5. Implementação Orientada a Objecto

A implementação Orientada a Objecto realiza-se depois de aprovados os produtos do DOO, através duma Linguagem de Programação Orientada a Objecto (LPOO).

---

[17] COAD, Peter e YOURDON, Edward. *Object - Oriented Analysis*, 2nd ed., (Englewood Cliffs, NJ: Yourdon Press/Prentice Hall, 1990).

Uma LPOO deve possuir três características básicas: polimorfismo, hereditariedade e encapsulamento.

A escolha duma LPOO baseia-se nas possibilidades que a linguagem fornece para a: criação de objectos e classes, criação de estruturas dos objectos e classes, descrição de atributos, e descrição dos serviços.

Depois da produção do programa, realiza-se a avaliação do programa através do teste experimental (que pode ser da unidade ou do sistema) e através de técnicas não experimentais tais como inspecção ou revisão estruturada.

Para a realização do teste é necessário determinar se será implementado o método incremental "top-down" ou "bottom-up"; dispôr de ferramentas e ambiente para o teste, ferramentas para depuração dos erros (bugs) e as ferramentas de gestão da configuração. Além disso, quando se emprega o método incremental "bottom-up", é necessário usar-se no ambiente do teste "simuladores de objectos" que criam desta forma um ambiente simulado.

Os produtos desta fase são: programa fonte, os resultados das técnicas de avaliação utilizadas e possívelmente o programa objecto e/ou executável. Estes produtos, já aprovados, são inclusos no Sistema de Gestão da Configuração, juntamente com os compiladores e simuladores de objectos usados.

2.1.6. Operação do software Orientado a Objecto

O software pode ser introduzido na organização (cliente) de forma incremental, paralela ou imediata. Para um sistema desenvolvido seguindo um método Orientado a Objecto assente num modelo espiral, a sua introdução pode ser incremental ou paralela.

A execução do Software Orientado a Objecto (SOO) numa organização não constitui o fim do ciclo de vida de desenvolvimento do SOO. Ao longo deste período de operação do software podem surgir necessidades de mudanças, que podem ser evolucionárias ou correctivas.

As mudanças evolucionárias, também designadas por manutenção evolutiva, que surgem devido as novas abstracções podem ser para adicionar novas classes e/ou objectos, mudar a implementação duma classe, mudar a representação duma classe, reorganizar a estrutura da classe, e alterar o interface da classe. As mudanças correctivas, também designadas por manutenção correctiva, resultam da detecção de erros que tenham passado nos testes realizados.

IV. SISTEMAS DE GESTÃO DE QUALIDADE NAS ORGANIZAÇÕES ABRANGIDAS PELO ESTUDO

Durante a realização deste trabalho foi possível contactar três organizações cujo um dos seus objectivos de negócio é o desenvolvimento de software de aplicação. O objectivo deste contacto foi para saber como é que elas têm o Sistema de Gestão de Qualidade do software dentro do seu Sistema de Gestão de Qualidade geral da organização.

Esta parte do trabalho pretende ilustrar de uma forma breve como estas organizações lidam com a problemática da qualidade de software.

Existindo diversas definições da qualidade dependendo da percepção de cada indivíduo ou organização, a qualidade é avaliada em função desta percepção através das características identificadas como sendo determinantes para a qualidade dum software. A qualidade é definida na perspectiva do produtor/fornecedor e na perspectiva do cliente/utilizador. Estas definições devem tender para um entendimento para que o software produzido seja satisfatório para os dois grupos participantes na compra/venda do mesmo.

Em função dos objectivos da organização, nível do pessoal empregue, exigências do mercado, as disponibilidades financeiras da organização e a tendência de expansão, cada organização define as suas formas de produção de modo a satisfazer o cliente abrangido e a rentabilidade financeira da organização produtora.

Os resultados obtidos são diferentes de organização para organização. Para o desenvolvimento do software de aplicação com qualidade desejada, as organizações produtoras do software abrangidas pelo estudo utilizam metodologias diferentes assentes em diferentes modelos recorrendo a métodos e ferramentas acessíveis, de acordo com o custo exigido e que seja aceite pelo cliente.

O CIUEM, por exemplo, emprega o SSADM assente num modelo incremental nos projectos complexos e o modelo linear nos projectos simples. Para o desenvolvimento, exige-se a participação constante do utilizador na fase de análise funcional do sistema.

A metodologia e o modelo são escolhidos com vista à obtenção dum software com qualidade satisfatória para o cliente e, a participação garante que o utilizador/cliente avalie os produtos de cada fase de desenvolvimento e possam ser tomadas as acções correctivas, se existirem erros, o mais cedo possível.

Para o mesmo objectivo de obter um software de aplicação com qualidade satisfatória, uma das opções da EXI é utilizar o Método Orientado a Objecto para algumas fases do CVDS, pois fornece, segundo Grosso (1996)[18], uma melhor representação do mundo real. A SORT Lda, por exemplo, não utiliza alguma metodologia de desenvolvimento do software de entre as já definidas e publicadas.

Apesar de não existirem diferenças nas definições da qualidade, existem diferenças nas disponibilidades de uso das tecnologias de informação, o conhecimento que se tem sobre as metodologias de desenvolvimento existentes, e o nível de cliente abrangido para a satisfação das suas necessidades.

Nas organizações abrangidas pelo estudo, não existe um SGQ formal. Além disso, na EXI e SORT não existe alguma unidade organizacional ou equipa que responda pela qualidade ao longo de desenvolvimento do software de aplicação, mas no CIUEM, já existe uma unidade organizacional que controla a qualidade dos serviços prestados pelo CIUEM. A unidade organizacional do CIUEM, com funções para controlar a qualidade nas suas várias áreas de actividades, está definindo, segundo Samiji (1996)[19] as formas de introducção dum SGQ.

[18] GROSSO, Orlando. EXI. Entrevista concedida em 22/04/1996.

[19] SAMIJI, Dilip. CIUEM-UEM. Entrevista concedida em 30/04/1996.

A não existência de uma equipa de qualidade nas duas organizações deve-se, segundo os entrevistados, a elevados custos que a organização iria incorrer e que não seriam suportados com as vendas do software produzido. Estar-se-ia, segundo Santos[20], a tender-se atingir altos níveis de qualidade sem rentabilidade financeira para a organização produtora. Nesta afirmação está implícita uma das abordagens para a definição da qualidade, baseada no valor do software, que segundo Broh(1988)[21], citado por Garvin (1988), a qualidade é o grau de excelência a um preço aceitável e o controlo da variabilidade a um custo aceitável.

Para a avaliação da qualidade do software, a prática corrente nas três organizações abrangidas é a participação do cliente, realização dos testes de aceitação e a auto-avaliação. Realizam-se os testes, mas os resultados do teste não são conservados, o que torna difícil medir o nível de melhoramento resultante das acções correctivas. Além da ausência de registos dos resultados dos testes do software, numa das organizações não existe a documentação de análise do sistema ou da concepção do software.

Outra razão que o autor deste trabalho distinguiu, para que a gestão de qualidade no ciclo de desenvolvimento do software de aplicação nas organizações abrangidas pelo estudo não seja muito formal, é porque normalmente o software é desenvolvido por uma a três pessoas; sendo o mais corrente, o desenvolvimento realizado por uma pessoa. Num ciclo de desenvolvimento do software em que participam uma ou duas pessoas, não constitui um problema sério a falta dum Sistema de Gestão da Configuração (ANEXO D: 1.3.1).

O mesmo não se pode afirmar dum projecto que envolva, por exemplo, mais de dez pessoas, como num projecto em que o autor participou na Petromoc E.E. na qual

[20] SANTOS, Joia. SORT Lda. Entrevista concedida em 18/03/1996.

[21] Robert A. Broh; *Managing Quality Higher Profits*, Nova Yorque: McGraw-Hill, 1982, p.3.

existiam sete equipas técnicas para sete unidades organizacionais escolhidas que constituiam o domínio do problema.

As equipas técnicas formadas, constituidas por três elementos, para a análise funcional do sistema duma determinada unidade organizacional, deviam no fim dum período preestabelecido elaborarem um documento único em que se descrevesse o sistema em estudo. Assim, devia existir uma maior coordenação entre as equipas do projecto, razão pela qual a Gestão da Configuração tornava-se importante neste projecto. Exigia-se a identificação e controlo dos documentos produzidos ao longo do desenvolvimento.

A existência dum Sistema de Gestão da Configuração, em projectos complexos e grandes que envolvem muitas pessoas é um dos factores que influencia positivamente para uma comunicação eficiente entre os membros envolvidos.

Em função do expresso nesta parte do trabalho, pode-se concluir que apesar de não existir um Sistema Gestão de Qualidade e nem serem implementadas muitas das práticas de gestão, garantia e/ou controlo de qualidade, existe a preocupação de produzir-se produtos com qualidade què corresponda aos requisitos do cliente abrangido. É devido a esta intenção de garantir e melhorar a qualidade dos produtos que o CIUEM está empenhado na introdução dum Sistema de Gestão de Qualidade.

V. CONCLUSÕES E RECOMENDAÇÕES

Para concluir pode-se afirmar que não existe uma definição consensual da qualidade, por isso, a sua avaliação varia de indivíduo/organização para indivíduo/organização e são utilizadas diferentes metodologias assentes em diferentes modelos para desenvolver um software de aplicação com qualidade que satisfaça os clientes.

Vários autores apresentam diferentes definições da qualidade dependendo da abordagem. Algumas definições são baseadas no valor do software, outras no utilizador e outras ainda na produção.

Apesar das organizações produtoras produzirem para satisfazerem o mesmo mercado, estas recorrem a diferentes formas de desenvolvimento do software, dependendo do grau da estrutura do sistema, familiarização com a tecnologia e o tamanho do projecto. As técnicas de avaliação utilizadas dependem, além dos factores antes mencionados, das necessidades e exigências dos clientes pois são eles que necessitam do software, dos custos para a sua realização e inserção num modelo, e da metodologia ou método de desenvolvimento do software.

Uma das premissas de garantia da qualidade é a participação do cliente no desenvolvimento de software, especialmente, na identificação e definição das suas necessidades duma forma clara, simples, concisa, não ambígua e testável, e a definição do tipo de software a produzir cumprindo com os requisitos formulados, e que seja uma solução a custo efectivo, oportuna e efectiva.

A identificação e definição das técnicas de avaliação a partir dos requisitos definidos pelo utilizador, e a definição das formas e actividades a realizar para controlar o cumprimento destas técnicas de avaliação num projecto específico além de depender dos clientes, estão sujeitas à complexidade dos projectos de desenvolvimento.

Duma forma geral, nas três organizações estudadas, os projectos envolvem uma a três pessoas. Geralmente é uma pessoa que faz a análise, desenho, programação assim como a avaliação dos produtos de cada fase. Este tipo de trabalho com este número de indivíduos envolvidos, não exige, com rigor, algumas práticas de gestão de qualidade. A gestão da configuração, num ciclo de desenvolvimento do software em que participa um indivíduo ou dois, não é de tanta importância como num projecto no qual participam na fase de programação, por exemplo, seis pessoas que produzam pequenos módulos para um mesmo programa.

A não existência de um sistema de gestão da configuração do software e sem efeitos negativos sérios deve-se ao número reduzido de elementos que compõem as equipas que desenvolvem o software de aplicação nas organizações abrangidas pelo estudo.

Pode-se depreender deste trabalho que não é fácil ter um Sistema de Gestão de Qualidade funcional porque envolve custos para a organização produtora, exige pessoas com habilidades para o efeito, aumentado o custo elevado do software produzido/fornecido.

De acordo a realidade de Moçambique onde existe apenas um Instituto Nacional de Normalização e Qualidade mas sem alguma Norma Nacional de Qualidade, não se pode esperar que a uma determinada organização seja atribuida um certificado de qualidade por Equipas De Certificação nacionais ou internacionais.

Apesar de não existirem normas, é importante que todos os que participam no desenvolvimento dos sistemas de informação estejam preocupados com a qualidade dos seus produtos e tentem introduzir novas atitudes de trabalho em relação à gestão de qualidade.

Seria importante que se fizesse um estudo que englobasse os principais consumidores das tecnologias de informação com o objectivo de determinar a viabilidade da criação duma norma da qualidade do software que protegêsse os interesses dos

intervenientes na transacção e de forma particular o utilizador pois este é que necessita de utilizar o software para o cumprimento dos seus objectivos organizacionais.

BIBLIOGRAFIA

1. ANGELL, I.O. e S. SMITHSON. (1991). *Information Systems Management: Opportunities and Risks*. 248 pp. London, I. O. Angel & S. Smithson.

2. BEYNON, P. e DAVIES. (1993). *Information Systems Development*, 2.ed. London, P. Beynon - Davies.

3. BLUM, B.I. (1992). *Software Engineering A Holistic View*. 588 pp. New York, Oxford University Press, Inc.

4. BOEHM, B.W. (1988). *A Spiral Model of Software Development and Enhancement*. pp 61-72. IEE Computer.

5. BOOCH, G. (1991). *Object Oriented Design with Applications*. 580 pp. California, The Benjamin/Cummings Publishing Company, Inc.

6. CASPERS, J. (1994). *Object-Oriented Programing: Analysis, Design and Implementation Methods*, 1.ed. 190 pp. South Carolina, Computer Technology Research Corp.

7. CROSBY, P.B. (1990). *Qualidade: Falando Sério*. 201 pp. Editora McGraw_Hill, Ltda e Newstec Editora Ltda.

8. CROSBY, P.B. (1992). *Quality is Free*. In: Strage, H.M. (Basil BlackWell, Ltd). *Milestones In Management: an Essential Reader*. Parte V: Calls to Arms. pp 333-340.

9. DAILY, K. (1992). *Quality Management For Software*. 185 pp. England, Kevin Daily.

10. DAVIS, W.S. (1994). *Business Systems Analysis and Design*. 534 pp. California, WadsWorth, Inc.

11. FERTUCK, L. (1995). *System Analysis & Design: With Modern Methods*. 659 pp. Dubuque, Wm. C. Browm Communications, Inc.

12. GANHÃO, F.N. (1991). *A Qualidade Total*. 101 pp. Lisboa, CEDINTEC.

13. GARVIN, D.A. (1988). *Gerenciando a Qualidade: A Visão estratégica e competitiva*. 357 pp. Rio de Janeiro, David A. Garvin.

14. HAWRYSZKIEWYCZ, I.T. (1994). *Introduction to Systems Analysis and Design*, 3.ed. 490 pp. Prentice Hall Australia.

15. International Standards Organization. (1996). *ISO 9000 Quality Management*, 6.ed. 382 pp. Genova, ISO.

16. KENDALL, P.A. (1992). *Introduction to Systems Analysis and Design: A Structured Approach*, 2.ed. 657 pp. United States of America, Wn C. Browm Publishers.

17. LAMB, D.A. (1988). *Software Engineering: Planning for Change*. 289 pp. New Jersey, Prentice Hall.

18. MARTIN, J. (1993). *Principles of Object-Oriented Analysis and Design*. 412 pp. James Martin and James J.Odell.

19. ROCHA, A.R.C. (1990). *Análise e Projecto Estruturado de Sistemas*. 141 pp. Rio de Janeiro, Campus.

20. SOMMERVILLE, I. (1989). *Software Engineering*, 3.ed. 653 pp. Great Britain, Addison - Weley Publisher Ltd.

21. TANENBAUM, A.S. (1987). *Operating Systems: Design and Implementation*. 719 pp. Englewood, Prentice-Hall Inc.

22. YOURDON, E. (1994). *Object - Oriented Systems Design: An Integrated Approach*. 400 pp. EngleWood Cliffs, Prentice - Hall Inc.

ANEXOS E APÊNDICES

APÊNDICE I. UPSS - UNIDADE DE PRESTAÇÃO DE SERVIÇOS DE SAÚDE

1. Descrição do sistema

A **UPSS**, é uma Unidade de Prestação de Serviços de Saúde, em funcionamento a partir de Janeiro de 1994, constituída por um Hospital, dez Centros e vinte Postos de Saúde distribuidos pelas províncias de Gaza, Inhambane e Maputo. A Direcção da **UPSS** e a do Hospital é a mesma e encontra-se na Cidade de Matola. Para o seu funcionamento, a **UPSS** tem uma Direcção da **UPSS**, Direcção Administrativa, e Direcção Clínica.

A Direcção da **UPSS** é responsável pela gestão estratégica da unidade de prestação de serviços de saúde. Define os objectivos, estrutura organizacional da **UPSS** e as responsabilidades dos membros nele envolvidos.

A Direcção Administrativa avalia as perspectivas futuras da **UPSS**, proporciona e controla os recursos humanos, materiais, económicos e financeiros para o funcionamento da **UPSS**, cria as condições harmoniosas entre os membros da organização assim como com o exterior, e controla sistemáticamente o grau de cumprimento dos objectivos.

A Direcção Clínica é responsável pelo controlo das actividades clínicas, nomeadamente: actividades curativas, profiláticas e educacionais para a garantia da saúde da comunidade sob seu raio de acção, através dos departamentos que a compõem, o Hospital, os Centros de Saúde e os Postos de Saúde.

Os Departamentos que compõem a Direcçao Clínica são os seguintes: Análises Clínicas, Bloco Operatório, Consultas e Triagem, Estomatologia, Farmácia, Fisioterapia, Oftalmologia e Radiologia. As Enfermarias existentes no Hospital da **UPSS** são: Cirurgia, Dermatologia, Medicina, Ortopedia, Pediatria e Oftalmologia.

O Departamento de Farmácia é responsável pelo aprovisionamento, acondicionamento e distribuição das especialidades farmacêuticas pelos Centros e Postos de Saúde e pelas enfermarias do Hospital da **UPSS**. Esta distribuição obedece aos seguintes crítérios: capacidade de atendimento da unidade sanitária, número de camas para internamento, stock do produto no armazém e os prazos de validade, acrescidos dos critérios estabelecidos pelo SNS, tais como o nível máximo do pessoal médico ou paramédico existente na unidade sanitária, e a categoria da unidade sanitária.

O Departamento de Farmácia está constituido pelo Sector de Acondicionamento; Sector das Preparações Galénicas e Magistrais; Farmácia de Venda ao público; Sector de Recepção, Conferência e Aviamento das especialidades farmacêuticas; e o Sector de Apoio Administrativo.

O Sector de Acondicionamento controla os prazos de validade, existência e movimento dos produtos e material médico-cirúrgico, elabora as requisições para a aquisição dos mesmos e elabora mapas de movimento. Os mapas de movimento podem ser de: estupefacientes e piscotrópicos, especialidades do Programa das DTS/SIDA e do Programa de Malária. Realiza a inutilização dos produtos com prazos de validade expirados, depois de confirmados no Laboratório de análises de medicamentos, no Ministério de Saúde.

O Sector das Preparações Galénicas e Magistrais emprega os reagentes para a produção de preparações galénicas ou magistrais que são entregues a Farmácia de venda ao Público ou ao Sector de Recepção, Conferência e Aviamento.

A Farmácia de Venda ao Público atende em especialidades farmacêuticas aos doentes ambulatórios com prescrições médicas da **UPSS** e é fornecida pelo Sector de Recepção, Conferência e Aviamento. Ao fim do dia, as receitas provenientes das vendas das especialidades são entregues ao Sector de Apoio Administrativo.

O Sector de Recepção, Conferência e Aviamento das especialidades farmacêuticas é responsável por: a recepção e conferência das especialidades farmacêuticas provenientes dos fornecedores e do Sector das Preparações Galénicas e Magistrais, na qual se verificam as quantidades recebidas em relação às pedidas (constantes na requisição enviada ao fornecedor) e aos prazos de validade das especialidades farmacêuticas (provenientes dos fornecedores); o aviamento das especialidades farmacêuticas para as Enfermarias do Hospital, os Centros e Postos de Saúde da **UPSS**.

O Sector de Apoio Administrativo é responsável pelo controlo das despesas internas, vendas e depósitos bancários; e pela recepção, avaliação, elaboração e distribuição de todo tipo de documentos movimentado no Departamento de Farmácia.

O Departamento de Farmácia da **UPSS**, optou por um sistema de estoque determinístico de requisição de reagentes e especialidades farmacêuticas de intervalo fixo, e elabora nas primeiras quinzenas dos mêses de Janeiro, Abril, Julho e Outubro, requisições dirigidas ao principal fornecedor com o qual elabora contratos anuais, a MEDIMOC E.E.

A **UPSS**, além da MEDIMOC E.E., pode adquirir as especialidades farmacêuticas em falta, à outras empresas importadoras e/ou produtoras de especialidades farmacêuticas, existentes no país. Dos fornecedores recebe as encomendas com as respectivas guias de remessas, em duplicado.

Para o aviamento aos Postos e Centros de Saúde e às enfermarias do Hospital, o Departamento de Farmácia recebe, primeiro, as requisições destas unidades. Baseando-se nestas requisições, efectua o aviamento das especialidades farmacêuticas, acompanhadas de uma guia de remessa.

De acordo com o regulamento de exercício da profissão farmacêutica, cada especialidade farmacêutica deve ser descrita da seguinte forma: Número do

Formulário Nacional de Medicamentos (FNM), Nome genérico, forma farmacêutica, quantidade do princípio activo, peso ou volume total e via de administração. Além desta descrição, uma especialidade farmacêutica deve caracterizar-se por número de lote, data de produção, data de expiração, Laboratório de Fabricação e País de Origem.

De acordo com o SNS, as unidades sanitárias estão classificadas em: Posto de Saúde(0); Outros Postos de Saúde(0,1); Centro de Saúde e Maternidade (0,1,2); Hospital Rural e Geral (0,1,2,3); Hospital Provincial (0,1,2,3,4); e Hospital Central(0,1,2,3,4,5). O pessoal de saúde está classificado nos seguintes níveis de prescrição médica: Agente polivalente elementar(0); Agente de medicina (0,1); Técnico de medicina(0,1,2); e Médico(0,1,2,3).

A **UPSS**, através do Departamento de Farmácia, deve fornecer nos primeiros dez dias dos mêses de Janeiro, Abril, Julho e Outubro, os mapas de controlo dos estupefacientes e psicotrópicos, mapas de controlo do movimento das especialidades farmacêuticas das DTS e os mapas de controlo de movimento das especialidades farmacêuticas do programa da Malária, aos seguintes Departamentos do SNS: Departamento Farmacêutico e Departamento da Epidemiologia e Endemias (à Secção de Malária e ao Programa Nacional de Controlo das DTS/SIDA).

A **UPSS**, ao longo dos seus dois anos de funcionamento, observou que o Departamento de Farmácia caracterizou-se com as seguintes situações:

- rupturas constantes de stocks;
- embora o calendário de aprovisionamento fosse: 1-15 de Janeiro, 1-15 de Abril, 1-15 de Julho e 1-15 de Outubro; verificou-se que o mesmo é realizado até finais dos mêses mencionados;
- atrasos acima de um mês para o envio dos mapas de controlo;
- cálculos das quantidades a distribuir levam aproximadamente um mês;
- os custos das inutilizações das especialidades farmacêuticas e similares expiradas e as quebras estão acima de 10%;

Perante isso, o Chefe do Departamento de Farmácia propos ao Director Geral da **UPSS** a introdução dum sistema informático, no Serviço Técnico, que permitisse:

- controlo do movimento das especialidades farmacêuticas;
- elaboração dos três tipos de mapas acima citados nos dias previstos;
- realização dos cálculos das necessidades trimestrais da **UPSS** e os respectivos acertos dentro da primeira quinzena de cada trimestre;
- identificação rápida e fiel dos prazos de validade de modo a reduzir os custos de inutilização para 5%;
- fornecer as inutilizações e quebras trimestrais;
- fornecimento das especialidades farmacêuticas com prazos de validade por expirar dentro de seis mêses.

O chefe do Departamento de Farmácia também gostaria que os utilizadores do sistema informático fossem atribuidos diferentes níveis para a execução das operações.

O Director Geral aprovou por saber que na UPSS existe um indivíduo que se encontra no 5° nível do Curso de Licenciatura em Informática e um programador desempenhando as funções de secretário particular do Director Geral.

Além disso, na Direcção da **UPSS** existe um computador Intel-100MHz com 16 Mb de RAM, 1.2Gbyte de espaço do disco duro; dois computadores 486DX-100 com 8Mb de RAM, 850Mb de espaço do disco duro; e uma impressora HP Lazerjet-4 Plus. Estes computadores são utilizados para a elaboração de textos e folhas de cálculo e dispõem de MS-DOS, MS-Office, Designer, Visual Basic, WP e Windows.

2. Definição do problema

Problema: Introdução, na **UPSS**, de um Sistema Informático de Gestão das Unidades Sanitárias (SIGUS), devido:

- 1 - a rupturas constantes de stock;
- 2 - aos atrasos de 15 dias no cumprimento do calendário de aprovisionamento;
- 3 - aos atrasos de um mês no envio dos mapas trimestrais ao SNS;
- 4 - ao controlo deficiente dos prazos de validade das especialidades farmacêuticas;
- 5 - aos custos de inutilização que são mais de 15%, acima do aceitável que é de 5%;
- 6 - ao período para o cálculo das quantidades a aviar que leva um mês, acima do período previsto, que é de 10 dias; e
- 7 - 10% de prejuízos devido a problemas de registo e controlo nos serviços de atendimento ao público.

Objectivos: Desenvolver um sistema informático (SI) que permita:

- 1 - a elaboração dos mapas das especialidades farmacêuticas do programa das DTS, do programa de malária e dos estupefacientes e psicotrópicos;
- 2 - o controlo do movimento das especialidades farmacêuticas;
- 3 - a determinação das necessidades trimestrais da **UPSS**;
- 4 - a realização do aprovisionamento em menos de 15 dias
- 5 - o controlo do movimento dos estupefacientes;
- 6 - a redução para 5% dos custos de inutilização, por controlar-se os prazos de validade através do sistema informático; e
- 7 - controlo eficaz de caixa.

Âmbito: As estimativas preliminares sugerem que a UPSS dispõe de recursos humanos e materiais para a execução do projecto.

Recomendações:

O projecto é técnica e operacionalmente viável pois a **UPSS** dispõe de tecnologias para o desenvolvimento e implementação, e está estimado para três mêses. Não é possível estimar a viabilidade económica, pois não possui informação suficiente para a sua estimação.

3. Determinação dos requisitos do sistema em estudo

Requisitos Funcionais:

- O sistema deve permitir o controlo do movimento das especialidades farmacêuticas .
- O SI deve permitir a elaboração dos mapas das especialidades farmacêuticas: mapa das especialidades do Programa das DTS/SIDA, mapa das especialidades do Programa da Malária, mapa dos estupefacientes e psicotrópicos, e mapa do material e de outros produtos.
- O SI deve permitir a realização do cálculo das necessidades trimestrais.
- O SI deve fornecer as especialidades farmacêuticas com prazos de validade por expirar dentro de seis mêses.

Requisitos do Interface:

- O SI deve fornecer os mapas das especialidades farmacêuticas das DTS/SIDA e da Malária prontos para o envio ao SNS.
- O SI deve fornecer mapas de estupefacientes e psicotrópicos obedecendo o formato estabelecido pela legislação do exercício farmacêutico, em vigor.
- O sistema deve fornecer as guias de remessas produzidas na Direcção prontas para o envio ao Hospital, Centros e Postos de Saúde.
- O SI deve fornecer as requisições para os fornecedores, prontas para o envio.

Requisitos de Desempenho:

- O SI deve suportar, pelo menos, seiscentos itens dum movimento de controlo de especialidades farmacêuticas;
- O SI deve suportar, pelo menos, mil itens dum movimento de controlo de material, equipamento e acessórios.

Requisitos do Desenho ou as Restrições:

- O SI é para ser implementado com o equipamento informático existente na **UPSS**.

4. Plano de desenvolvimento

Para o desenvolvimento do SIGUS, cujo problema e objectivos estão definidos no "2. Definição do problema" e os requisitos do sistema identificados no "3. Determinação dos requisitos", foram identificados e disponibilizados pela direcção da UPSS os recursos materiais (Fig.4) e elaborou-se o plano do projecto (Fig.5) e o plano detalhado da fase de análise (Fig.6).

Foram afectos, para o desenvolvimento do SIGUS:

- um analista de sistemas e um programador, pertencentes a UPSS; e
- disponibilizados dois computadores com seguinte software: MS-Office, Perfect-Office, MS-Project 5.0, Designer, Visual Basic 3.0 e Informix 4GL.

Nota: As abreviaturas utilizadas são: C.R (Código do Recurso), Q (Quantidade), C.U (Custo Unitário), C.T. (Custo Total), Dur. (Duração), D.I. (Data do Início), Dep. (Dependência), ID (Identificador), d (dia) ed (elapsed day).

a) Recursos

ID	Nome do Recurso	C.R	Q	C.U	C.T
1	***Material Consumível***				
1.1	Disquetes, caixa	DI	1	1.68	6.72
1.2	Resma de Papel	RP	2	8.64	17.28
1.3	Separador	SP	10	5.45	54.50
1.4	Esferográficas	ES	10	0.14	1.40
1.5	Lápis	LA	6	0.18	1.08
1.6	Borracha	BO	3	0.27	0.81
1.7	Pastas de arquivo	PA	4	4.55	18.20
2	**Total**				**100.99**
2.1	Diversos		10%		10.10
3	**Total**				**111.09**

Figura 4 - Recursos Materiais para o Projecto

b) Actividades

ID	Nome da Actividade	Recursos Humanos	Dur.	D.I	Dep.
1	Início do Projecto		0d	01/04/96	
2	Análise do sistema	Analista	30d	01/04/96	1
3	Avaliação do modelo de AOO	Analista	2d	13/05/96	2
4	Entrega do modelo de AOO		0d	15/05/96	3
5	Desenho do sistema	Analista	18d	15/05/96	1
6	Avaliação do modelo de DOO	Analista	2d	10/06/96	4
7	Entrega do modelo do DOO	Analista	0d	12/06/96	5
8	Codificação	Analista, Programador	20d	12/06/96	4
9	Avaliação do programa	Analista, Programador	5d	11/08/96	7
10	Entrega do programa		0d	17/07/96	8
11	Fim do Projecto		0d	17/07/96	9

Figura 5 - Actividades do Projecto de Desenvolvimento do SIGUS

4.1. Plano Detalhado da Fase de análise

ID	Nome da Actividade	Recursos Humanos	Dur.	D.I	Dep.
1	Início da análise		0d	01/04/96	
2	Descobrir objectos e classes		2ed	01/04/96	1
2.1	- Identificar classes e objectos	Analista	0.5d	01/04/96	2.1
2.2	- Avaliar classes e objectos	Analista	1d	01/04/96	1
2.3	- Determinar os assuntos	Analista	2d	01/04/96	2.3
3	Definir estruturas de classes e objectos	Analista	5d	02/04/96	2.3
4	Definir as relações estáticas entre classes e objectos	Analista	5d	08/04/96	2.3
5	Definir atributos para classes e objectos		5ed	12/04/96	1
5.1	- Identificar os atributos	Analista	2d	12/04/96	1
5.2	- Colocar os atributos nas classes hierárquicas	Analista	3d	15/04/96	2.3
5.3	- Examinar as classes Hierárquicas	Analista	4d	15/04/96	3
6	Definir o comportamento dos objectos		4ed	18/04/96	2.3
6.1	- Elaborar diagramas do CVO	Analista	3d	18/04/96	2.3
6.2	- Avaliar diagramas do CVO	Analista	4d	18/04/96	2.3
7	Definir os serviços e identificar as mensagens		7ed	18/04/96	6.2
7.1	- Determinar os serviços exigidos	Analista	3d	18/04/96	6.2
7.2	- Documentar as conexões de mensagens	Analista	5d	18/04/96	6.2
7.3	- Descrever o comportamento detalhado de cada objecto	Analista	5d	18/04/96	6.2
8	Entrega do Diagrama de interacção dos objectos		0d	19/04/96	7.1
9	Entrega do Diagrama de comunicação dos objectos		0d	26/04/96	7.3
10	Elaboração do documento preliminar	Analista	7d	18/04/96	5.1
11	Avaliação do documento preliminar	Analista	2d	26/04/96	10
12	Elaboração do documento final	Analista	5d	03/05/96	11
13	Entrega do documento final		0d	10/05/96	12
14	Fim da análise		0d	10/05/96	13

Figura 6 - Plano Detalhado da Fase de Análise

4.2. Plano de Qualidade

O objectivo deste Plano de Qualidade (Fig.7), que engloba apenas a fase de análise, é garantir que sejam alcançados os requisitos identificados no ponto "3. Definição dos requisitos" deste "APÊNDICE" e que haja a participação do utilizador para correcções de erros além da inclusão de novas necessidades, quando for possível.

ID	Nome de Actividade	Recursos Humanos	Dur.	D.I.	Dep.
1	Definir a Notação	Analista	0.5d	01/04/96	
2	Avaliar os objectos candidatos	Analista	1d	01/04/96	1
3	Avaliar as estruturas hierárquicas	Analista	4d	15/04/96	2
4	Avaliar os atributos	Analista	4d	15/04/96	2
5	Avaliar diagramas de CVO	Analista	4d	18/04/96	4
6	Avaliação do documento final	Analista	2d	26/04/96	3,4,5

Figura 7 - Plano de Qualidade

a) Avaliar os objectos candidatos

Na avaliação dos objectos candidatos ao modelo de AOO avalia-se se:

- o objecto candidato tem alguns atributos que requeiram a conservação de certos valores;
- o objecto realiza alguma coisa que justifique a sua existência;
- a funcionalidade, ou comportamento, identificado do objecto é importante independentemente da tecnologia do hardware e software que será usada para implementar o sistema;
- os atributos podem ser colocados satisfazendo os critérios de hereditariedade;
- o objecto proposto tem mais de um atributo; e
- o que acontece em relação aos atributos, pode ser aplicado para os serviços.

b) Avaliação das estruturas hierárquicas

Recorrendo à auto-avaliação das estruturas hierárquicas das classes e objectos, pois é o que consta no Plano de Qualidade em termos de recursos humanos, avalia-se se:

- todos os atributos dos objectos da generalização são também válidos para os objectos da especialização; e
- existem outras classes que quando forem combinadas podem formam uma estrutura "TODO-PARTE" com significado.

Os atributos são avaliados para verificar se:

- as classes e objectos têm atributos comuns.

Nos diagramas do CVO avalia-se se:

- todos os estados do objecto foram definidos,
- todos os estados do objecto podem ser alcançados,
- pode-se sair do ciclo desde dum estado não final, e
- para cada estado, o objecto responde convenientemente para todas as situações.

Nota: Sendo um projecto com a participação, básica, de uma pessoa, pode não se recorrer, ao longo do projecto a uma gestão da configuração rigorosa. Mas, toda documentação produzida, deve estar sob a responsabilidade do analista de sistemas.

ANEXO A. Algumas Características do Software

A seguir é apresentada uma descrição de algumas características de software.

Utilidade - um software é útil se for usado para alcançar o objectivo inicialmente pretendido, sob as condições permitidas pelas especificações.

Precisão - um software é de precisão ou exacto se ao serem introduzidas entradas que satisfaçam as especificações de entrada sob as condições de operação permitidas, fornecer saídas que satisfaçam as especificações de saída.

Fiabilidade - Probabilidade de funcionamento sem falhas de um software, em condições de exploração definidas e num período de tempo limitado.

Eficiência - um software é eficiente se realiza um trabalho a ele exigido com o mínimo de recursos, dentro dos limites dos requisitos do tempo e custo.

Compactibilidade - um software é compactível se puder trabalhar com outros sistemas ou software.

Integridade - a integridade do software é que garante o controlo do acesso e permissão de executar apenas o que for permitido.

Inteligibilidade - um software é inteligível se sua estrutura e função forem claras. E é possível determinar-se se o software estiver documentado.

Legibilidade - um software é legível se for fácil para a leitura, e, só é possível avaliar-se esta característica se o programa fonte estiver documentado.

Alterabilidade - um software é alterável se permitir ser alterado e, só é possível se for legível e inteligível.

Testabilidade - um software é testável se permitir a realização eficiente dos testes, particularmente depois de ser alterado.

Portabilidade - é uma das características que permite ao software funcionar em diferentes ambientes.

"Reusabilidade" - é uma das características que permite ao software ser usado para diferentes ambientes, adaptar-se ao novo ambiente.

ANEXO B. Descrição de Alguns Tipos de Testes

A seguir são descritos alguns tipos de teste do software de aplicação.

Teste da unidade - é o mais simples e elementar que é executado para componentes tais como função ou objecto[22]. "Num sistema convencionalmente desenhado, cada componente deve ter uma especificação precisa e, o teste deve ser definido para verificar se os componentes se ajustam as suas especificações" (Sommerville, 1989).

Teste do módulo - é realizado depois de terem sido juntadas as unidades (testadas com êxito) interdependentes, formando um módulo.

Teste do subsistema - é realizado ao subsistema resultante da composição dos módulos relacionados, com um objectivo comum.

Teste de integração (sistema) - é realizado para identificar defeitos antes do sistema ser entregue ao utilizador/cliente.

Nos testes da unidade, módulo, subsistema e integração, usam-se normalmente dados gerados pela organização produtora do software, não reais. Os testes de aceitação e paralelo usam os dados reais do utilizador/cliente.

Teste de aceitação - é realizado, geralmente, para demonstrar se houve o cumprimento dos requisitos do sistema e os critérios de aceitação que constam na especificação do problema.

O *critério de aceitação* protege tanto ao cliente como ao analista no contrato de desenvolvimento do software. O utilizador usa os critérios como o meio para avaliar se os seus requisitos foram atingidos. E são utilizados pelo analista para provar ao

[22] No Método Orientado a Objecto é onde se utiliza a componente 'Objecto'.

utilizador que o sistema produzido cumpre com os requisitos por ele definidos. Para que um critério de aceitação seja efectivo, ele deve ser mensurável.

Teste paralelo - é realizado quando o novo sistema é executado em paralelo com o velho, trabalhando com os dados reais nos dois sistemas e comparam-se os resultados.

ANEXO C. DOCUMENTAÇÃO EXISTENTE NUM SISTEMA DE GESTÃO DE QUALIDADE

Uma organização produtora/fornecedora do software, possuindo um Sistema de Gestão de Qualidade que se apoie na "Guidelines" ISO 9000-3, deve possuir a seguinte documentação relacionada com a qualidade:

- documentos da definição e operação do Sistema de Gestão de Qualidade (SGQ); e
- documentos do ciclo de vida de desenvolvimento (CVD) do sistema/software.

A definição, autorização, publicação e controlo das alterações dos documentos existentes num SGQ deve obedecer a um conjunto de procedimentos pré-definidos para manipular os documentos. Além disso, devem existir procedimentos para retirada de documentos obsoletos, colocação dos mesmos nos locais próprios, indicação do pessoal para a sua actualização.

Os documentos da definição e operação do SGQ abrangem as actividades gerais e são independentes dos projectos. O manual de qualidade, segundo a ISO, é o documento com a definição e formas de operação do SGQ. O manual de qualidade contém a política e objectivos de qualidade, estrutura organizacional e responsabilidades pela qualidade do software.

Na estrutura organizacional são definidos os deveres, responsabilidades e autoridade do pessoal que gere, verifica, valida, testa ou realiza algum trabalho que tenha alguma influência na qualidade. A responsabilidade pela qualidade na organização produtora/fornecedora do software deve tender a envolver todas as pessoas da organização e a todos os níveis, isto é, ao pessoal da gestão, técnico e de suporte.

O manual de qualidade deve ser simples, claro e conciso, que não possua informação confidencial da organização, pois deve ser mostrado ao cliente, quando o solicitar, e faz referência a outros documentos, por exemplo, manual dos procedimentos.

O manual de qualidade deve conter, também, a descrição das actividades de revisão e auditoria do SGQ, acções de melhoramento do SGQ, actividades de "procurement" e procedimentos de treino do pessoal de qualidade.

Os documentos do CVD do sistema/software podem estar relacionados com o projecto ou com algumas fases/etapas específicas do projecto. Os documentos relacionados com o projecto abrangem as práticas de qualidade aplicáveis a algumas ou todas as etapas do projecto, tais como normas dos planos de qualidade e da documentação do projecto; procedimentos de revisão, de gestão da configuração, "backup", segurança, armazenamento, reportagem e monitoração do progresso do projecto, das acções correctivas e tratamento dos itens defeituosos.

Os documentos relativos a algumas fases/etapas específicas dum projecto podem ser, por exemplo: especificação dos requisitos do utilizador, proposta do contrato/projecto, especificação do desenho preliminar e detalhado, especificação do programa, manual do utilizador e manual das operações.

Os documentos podem ser obrigatórios ou consultivos. Os documentos obrigatórios são os seguintes: normas, manuais dos procedimentos e instruções de trabalho. Os consultivos, por exemplo, são as guias de trabalho. A norma (padrão) do software é um conjunto obrigatório de regras a cumprir durante as fases do CVD do software, abrangidas por ela. Exemplo: norma (padrão) de programação.

O procedimento explica para cada passo dum trabalho específico a identificação do passo, como é executado, quando é realizado, onde e quem realiza. Num CVD do software existem procedimentos para as etapas, tais como definição do problema, estudo de viabilidade, análise do sistema, desenho do sistema, programação e manutenção; procedimentos para processos de suporte, tais como controlo de mudanças, revisão do projecto e gestão da configuração, compra e controlo dos documentos em si.

A instrução de trabalho fornece mais detalhes sobre os procedimentos, definindo precisamente um passo particular do procedimento. Exemplo: método específico de realizar inspecção na etapa de análise, forma de realizar entrevistas ou questionários aos utilizadores.

Os documentos consultivos fornecem informação suplementar aos procedimentos e normas. Estes documentos consultivos podem incluir, por exemplo, "dicas" ou exemplos que ajudam ao analista, desenhador ou programador na utilização de uma norma ou procedimento durante o CVD.

Os documentos, em função do nível do SGQ que abrangem podem ser: documentos de gestão, da garantia e de controlo da qualidade. O documentos de gestão da qualidade podem ser: manual de qualidade, registos dos resultados das auditorias e revisões periódicas do SGQ, e registos das acções específicas de correcção e/ou melhoramento.

Os documentos da garantia de qualidade podem ser: planos do projecto, planos da qualidade do projecto, planos de gestão da configuração, registos dos resultados das auditorias do projecto, revisões e monitoração do progresso do projecto, e o registo da realização das actividades de controlo de qualidade, isto é, das técnicas de avaliação.

Os documentos do controlo de qualidade são: laudos de inspecção, registos dos resultados das técnicas de avaliação utilizadas nas fases do projecto e registo das acções correctivas e de melhoramento tomadas.

ANEXO D. ISO 9000-3

Neste anexo é apresentada a Regra (Guidelines) relacionada com a qualidade do software.

1. A Regra (Guidelines) para software: ISO 9000-3

Análise da ISO 9000-3		
Estrutura do Sistema de Gestão de Qualidade	**Actividades do Ciclo de Vida**	**Actividades de Suporte**
Responsabilidade de Gestão	Revisão do Contrato	Gestão de Configuração
Sistema de Gestão de Qualidade	Especificação dos Requisitos do Cliente	Controlo de Documentos
Auditoria Interna do Sistema de Gestão de Qualidade	Planeamento de Desenvolvimento	Registos de Qualidade
Acção Correctiva	Planeamento de Qualidade	Medidas
	Desenho e Implementação	Regras, Práticas e Convenções
	Teste e Validação	Ferramentas e Técnicas
	Aceitação	Compra
	Manutenção	Produto do Software Incluso
		Treinamento

Figura 3.7 Interpretação da ISO 9000-3 da BS 5750/ISO 9001
FONTE - Daily, 1992, p.41

Figura 8 - Interpretação da "Guidelines" ISO 9000-3

A ISO 9000-3 é uma Regra de aplicação para o desenvolvimento, fornecimento e manutenção do software. Nesta Regra, segundo Daily (1992), os requisitos do

Sistema de Gestão de Qualidade estão classificados em três grupos, ilustrados na Figura 8, que são:

- a estrutura do Sistema de Gestão de Qualidade,
- as actividades do ciclo de vida, e
- as actividades de apoio.

1.1. Estrutura do Sistema de Gestão de Qualidade

Responsabilidade da Gestão

Nesta componente é onde consta o que a Regra exige que exista num Sistema de Gestão de Qualidade, tal como:

- Definição dos objectivos, estrutura e responsabilidades da organização pela qualidade e sua difusão por todos níveis existentes na organização relacionados com a qualidade;

- Definição e documentação da política de qualidade da organização; e

- Definição dos direitos, responsabilidades e autoridades para todo o pessoal que esteja a efectuar um trabalho relacionado com a qualidade.

Sistema de Gestão de Qualidade

Nesta componente, é onde consta o que a Regra exige dum Sistema de Gestão de Qualidade, que é:

- Definição e documentação do Sistema de Gestão de Qualidade para demonstrar como é implementada a política de qualidade na organização. A documentação pode estar constituida por: manual de qualidade, planos de qualidade, manuais dos procedimentos, manuais de instruções de trabalho e guias de trabalho.

- Identificação e documentação das técnicas de avaliação e o registo dos resultados num documento denominado registos de qualidade.

Auditoria Interna do Sistema de Gestão de Qualidade

Nesta componente, a regra exige a realização duma série de auditorias internas do Sistema de Gestão de Qualidade para medir o seu nível de efectividade e identificar as possíveis áreas para alteração e/ou melhoramento além de ser uma evidência que é exigida ao Sistema de Gestão de Qualidade por uma equipa de avaliação externa para a Certificação.

Acções Correctivas

Nesta componente, a Regra exige a existência de procedimentos específicos para acções correctivas e que estas bem empregues possam conduzir a alteração e/ou melhoramento. Estes procedimentos específicos podem ser reactivos e preventivos.

Os *procedimentos reactivos* orientam na investigação das causas do desvio do produto em relação aos requisitos e a identificação das acções correctivas para prevenir a ocorrência do mesmo problema.

Os *procedimentos preventivos* orientam na análise da informação disponível sobre os produtos e/ou processos de trabalho, registada nos manuais e registos de qualidade, para identificar e eliminar as causas do não cumprimento dos requisitos do utilizador.

As causas para o surgimento dum problema/erro num produto do software podem ser humanas, dos métodos usados, resultantes do meio ambiente e da informação usada (Daily, 1992).

1.2. Actividades do ciclo de vida

Estes são os requisitos técnicos e que estão relacionados com um projecto específico cujo objectivo é obter um nível satisfatório da qualidade.

A Regra ISO 9000-3 é uma proposição de requisitos em termos gerais, para qualquer tipo de projecto de software. A organização produtora deve adaptá-la à sua realidade e à do projecto a desenvolver.

1.3. Actividades de suporte

1.3.1. Gestão da configuração

A *configuração do software*, segundo Lamb (1988), é uma colecção de documentos, programas fontes, ficheiros relacionados, ferramentas usadas no desenho e implementação dum sistema de software, e as entradas para as ferramentas.

Para obter-se um controlo efectivo desta colecção de items existentes ao longo do ciclo de vida de desenvolvimento do software pode ser através da Gestão da Configuração, que a sua falta pode resultar em problemas da qualidade significativos, tais como:

- emissão duma versão errada do documento do usuário,
- alterações feitas ao programa sem expressa autorização,
- não ser possível obter a versão inicial do módulo do software,
- uso dum programa num software publicado antes do teste estar completo e aprovado, e
- não ser possível identificar claramente o efeito dum requisito alterado.

Assim, a Gestão da Configuração, segundo Lamb (1998), consiste na identificação, controlo e informação da trajectória de todos os items que constituem o produto do software e todas as versões destes items. Deste modo, as actividades básicas da Gestão da Configuração são: identificação dos items da configuração, controlo da configuração, e reportagem do estado e trajectória do item da configuração.

Identificação dos Items da Configuração

A *Identificação dos Items da Configuração* é o processo através do qual são identificados os objectos[23] que devem estar sob o controlo da configuração e as relações que devem existir entre eles. Num projecto, os objectos que podem estar sob o controlo são os documentos de desenvolvimento, ferramentas usadas para a construção do produto, o código fonte de cada módulo, laudos das avaliações e relatórios dos problemas. Os objectos a serem controlados devem constar no plano de gestão da configuração.

Controlo da Configuração

O *Controlo da Configuração* é o processo que gere as mudanças e controla as operações diárias realizadas pela equipa que desenvolve o sistema. Os detalhes a considerar para que um objecto seja controlado dependem dos critérios de decisão definidos no plano de gestão da configuração.

Deve existir o controlo da configuração para manter a estabilidade no projecto, onde geralmente estão envolvidas muitas pessoas. A estabilidade do ambiente no projecto garante o progresso, pois a relação de dependência dos produtos das fases do ciclo de vida torna-se estável.

Segundo Daily (1992), existem as seguintes formas de controlo dos items da configuração:

controlo da versão	identifica cada item e regista a sua história de desenvolvimento;
controlo da construção	controla o que foi construido a partir de items de configuração;
controlo da mudança	controla as alterações feitas nos items; e
reportagem do estado	reporta o estado e a história de cada item e o produto construido.

[23] Objecto, definido como sendo uma entidade, caracteriza-se por ter um conjunto de atributos que o identificam dos demais objectos; tem um estado, comportamento e identidade.

Reportagem do Estado e Trajectória do Item da Configuração

Esta é uma actividade que assegura que a equipa que desenvolve o sistema, gestores, e utilizadores saibam da história e o estado corrente dos objectos identificados para estarem sob o controlo da configuração.

1.3.2. Controlo da Documentação

O *Controlo da Documentação* é uma actividade que consiste no controlo cuidadoso e manipulação consistente dos documentos do ciclo de desenvolvimento do software, da definição e operação do Sistema de Gestão de Qualidade.

Assim, é necessário que sejam definidos os procedimentos para manipular estes documentos e que para Daily (1992), um dos primeiros procedimentos a estabelecer-se dentro do Sistema de Gestão de Qualidade é um *Procedimento para Manipular Procedimentos*.

O conjunto típico da documentação do ciclo de desenvolvimento do software inclue:

- documentos dos produtos das fases do desenvolvimento do software;
- manuais do utilizador e guias das operações;
- planos do projecto, registos do progresso, relatórios das revisões e outros documentos da Gestão do Projecto; e
- documentos relacionados com a qualidade tais como planos da qualidade, registos de inspecções, planos e resultados do teste.

1.3.3. Registos de qualidade

O *registo de qualidade* é um dos requisitos que fornece evidência de que estão sendo seguidos os procedimentos e práticas do Sistema de Gestão de Qualidade e é disponível para ajudar na resolução de problemas e tomada de acções correctivas.

Duma forma resumida, Daily (1992) afirma que os registos de qualidade são aqueles que:

- demonstram a realização/sucesso da qualidade exigida; e
- demonstram que o Sistema de Gestão de Qualidade está funcionando.

1.3.4. Compra

Para a realização duma compra é necessário que o 'objecto' comprado corresponda com os objectivos pretendidos e a qualidade desejada, e que o cliente possa verificá-los através dos requisitos préviamente definidos.

Assim, para que se proceda a uma compra, devem ser seguidos, segundo Daily (1992), três passos:

- especificar os requisitos do software,
- avaliar se o software satisfaz os requisitos, e
- registar a performance do software.

1.3.5. Treinamento

A Regra exige a identificação das necessidades de treinamento de todo o pessoal relacionado com a qualidade e que corresponda com as qualificações pretendidas para a realização das actividades exigidas num Sistema de Gestão de Qualidade.

1.3.6. Outros requisitos

Os outros requisitos da ISO 9000-3 são:

- registo e análise dos defeitos do produto operacional, e isto pode ser feito através das métricas analíticas préviamente identificadas;
- definição e documentação das regras, práticas e convenções, além da revisão da efectividade de cada uma delas; e

- identificação das técnicas e ferramentas que sejam sejam típicamente técnicas ou de gestão.

ANEXO E. MODELO ESPIRAL DO CICLO DE VIDA

O modelo espiral é baseado nos modelos linear, incremental e evolucionário, em função da realidade existente no desenvolvimento dum determinado software.

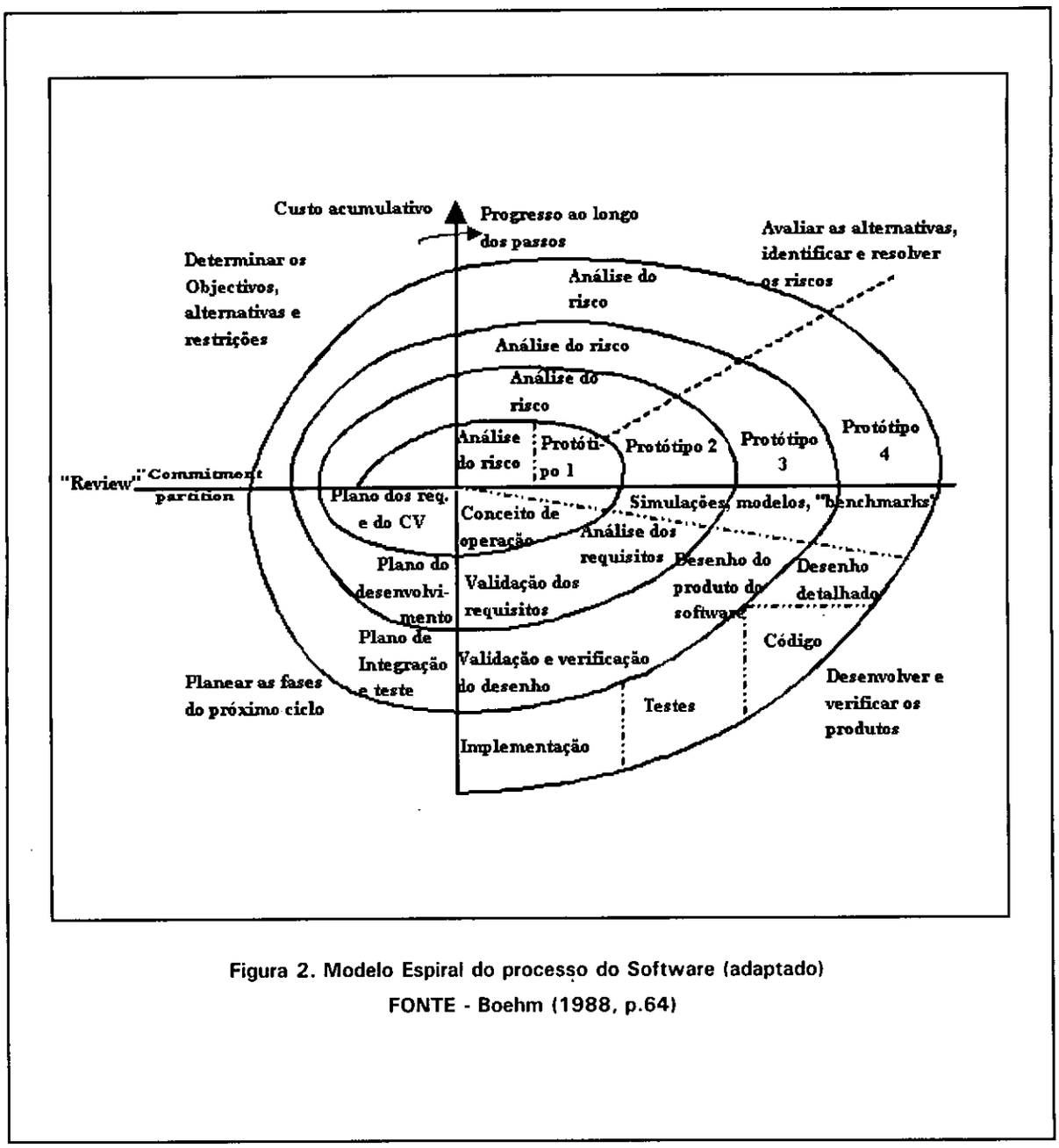

Figura 2. Modelo Espiral do processo do Software (adaptado)

FONTE - Boehm (1988, p.64)

Figura 9 - Modelo Espiral de Desenvolvimento do Software

Um modelo espiral de desenvolvimento do software é composto por cíclos, cada ciclo está dividido em quatro passos (Fig. 9), que são:

- 1° Passo: determinação dos objectivos, alternativas e restrições;
- 2° Passo: avaliação das alternativas, identificação e resolução dos riscos;
- 3° Passo: desenvolvimento e avaliação do(s) produto(s); e
- 4° Passo: planeamento das fases do ciclo seguinte.

O desenvolvimento do software de aplicação utilizando o modelo espiral começa com a concepção da operação, isto é, o surgimento da ideia da produção dum software. Assim, são definidos os requisitos e elabora-se um plano dos requisitos e do ciclo de vida do software. Faz-se a revisão do contrato e a revisão das partições escolhidas.

Depois destas revisões é que começa o primeiro passo do primeiro ciclo do modelo espiral.

1° Passo: Determinação dos objectivos, alternativas e restrições - Neste passo, determinam-se os objectivos da parte do produto a ser desenvolvido, os meios alternativos para o seu desenvolvimento e/ou implementação e as restrições impostas durante o projecto de desenvolvimento.

2° Passo: Avaliação das alternativas, identificação e resolução dos riscos - Neste passo, avaliam-se os meios alternativos identificados para a implementação operando dentro das restrições impostas de custo, tempo e qualidade, para atingir os objectivos determinados no 1° Passo.

Durante a avaliação, são identificadas as áreas de maior incerteza, que constituem fontes de risco do projecto de desenvolvimento do software. Depois da identificação das áreas, é necessário identificar as técnicas mais efectivas de resolução dos riscos que podem ser usadas de forma isolada ou conjunta. Exemplos de técnicas de resolução dos riscos são: prototipificação, simulação e verificação do termo de referência.

3º Passo: Desenvolvimento e verificação do(s) produto(s) - Se no 2º passo, for possível eliminarem-se os potenciais riscos do projecto, segue-se o modelo linear.

Mas se não for possível, e ainda, persistirem os riscos de performance e do interface, então, recorre-se ao desenvolvimento evolucionário. Isto é para resolver as questões de maior risco através do desenvolvimento dum protótipo mais detalhado.

Depois disto, submete-se os produtos resultantes do ciclo a uma avaliação. Também são avaliados os planos e identificados os recursos para o próximo ciclo. As avaliações permitem obter a opinião das partes envolvidas no projecto de desenvolvimento do software e assumirem-se os compromissos para continuar com as fases seguintes.

4º Passo: Planeamento das fases do próximo ciclo - Este processo consiste na elaboração do plano detalhado para as fases seguintes do novo ciclo.

1. Considerações sobre o modelo espiral

Ao longo do desenvolvimento do software de aplicação usando o modelo espiral é necessário identificar e eliminar as fontes de risco do projecto, através das respectivas técnicas. Deste modo surge a Gestão do Risco no projecto do software (descrito no ponto "2" deste ANEXO), que é uma das garantias da aplicação eficiente do modelo espiral.

Com a Gestão do Risco do software, surgem as técnicas para identificação e análise do risco, e estabelecimento dum plano de gestão do risco. Para o plano de gestão do risco exige-se primeiro a identificação dos items do risco. No cumprimento do plano deve-se resolver cada item do risco, actualizar a lista do risco, a medida que forem surgindo e/ou eliminando-se.

É importante realçar o estado do item do risco nas revisões periódicas do projecto, e comparar os estados dos items dum período da revisão com o período anterior. Feita esta avaliação por meio de comparação, decidem-se as acções correctivas apropriadas a tomar.

Uma característica importante do modelo espiral é que cada ciclo termina com o uso duma técnica de avaliação com o envolvimento das pessoas ou organizações interessadas com o produto. A avaliação é feita aos produtos produzidos nos ciclos anteriores, aos planos e recursos. O objectivo principal desta avaliação é obter o consenso das partes envolvidas para a continuação da fase seguinte.

2. Gestão do risco do software

Na Gestão de Projecto, um Gestor pode tomar acções preventivas e/ou correctivas para alcançar os objectivos definidos, na base das restrições impostas e alternativas disponíveis. As acções correctivas, geralmente são mais caras em relação às preventivas. A *Gestão de Risco* é uma acção preventiva pois previne a ocorrência dos problemas.

Blum (1992) define o risco como sendo a possibilidade dum prejuízo num projecto, os danos da performance ou resultado dum projecto. Cabe ao Gestor, na Gestão de Risco, avaliar e controlar os riscos com o objectivo de definir acções para eliminar ou reduzir os erros.

Boehm[24], citado por Blum (1992), identificou duas partes principais na Gestão do Risco:

- *avaliação do risco* - identificação, análise e estabelecimento de prioridades do risco; e
- *controlo do risco* - planeamento de gestão, resolução e monitoração do risco.

[24] BOEHM, B. W., *Software Risk Management*, IEEE Computer Society Press, Washington, DC, 1989

Os riscos podem ser classificados, segundo Blum (1992) em:

- *riscos gerais* - comuns a todos os projectos; e
- *riscos específicos do projecto* - dirigidos a um plano de risco dum projecto específico. Ex: requisitos de performance ambiciosos, orçamento escasso e restrições do programa.

A identificação dos riscos pode ser feita através de listas de verificação ("checklists"), análise orientada a decisões, análise das suposições e a decomposição.

As *listas de verificação* baseiam-se na experiência que o Gestor possue dos projectos anteriores. Na *análise orientada a decisões*, avaliam-se e identificam-se as decisões tomadas não tendo como base aspectos técnicos de gestão. Exemplos: a escolha dum equipamento informático com o objectivo de garantir as relações amistosas entre organizações, produzir um software no prazo mais curto possível. Estes são exemplos de tomada de decisões tendo como base questões políticas ou comerciais, e constituem fontes de riscos.

Outra forma de identificação do risco é *analisar as suposições*, pois as suposições não se baseiam na experiência e tendem a ser optimistas e não claras. Com a *decomposição*, que é na base de refinamentos e novas abstrações, novos detalhes que estavam ocultos num alto nível de abstração tornam-se visíveis, reduzindo-se assim o risco resultante destas suposições.

Análise do risco

A análise dos riscos do projecto assenta em modelos que derivam da experiência anterior em relação aos custos e programas, na base dos quais se calculam as possibilidades dos resultados desejados. Na base destes modelos efectua-se a análise das decisões, factores da qualidade e redes de trabalho.

Com o resultado desta análise podem ser estabelecidos os critérios para a ordenação e estabelecimento de prioridades dos riscos determinados.

Planear a Gestão de Risco

No planeamento, devem ser respondidas, por cada item do risco, as seguintes perguntas:

- *Por quê?* - para definir a importância do item do risco e sua relação com os objectivos do projecto;
- *O quê e quando?* - para a identificação dos marcos e resolução dos riscos;
- *Quem e onde?* - para atribuição das responsabilidades individuais e da organização;
- *Como?* - para a definição das formas de resolução dos riscos, que podem ser através de protótipos, simulações, "benchmarks", análises e fornecimento de pessoal; e
- *Quanto?* - para determinar os custos para a resolução dos riscos do projecto.

Alguns princípios de Gilb[25], citado por Blum (1992), relacionados com a gestão do risco, são:

- *O princípio do risco:* se você não ataca activamente os riscos, eles atacar-te-ão activamente.

- *O princípio da partilha do risco:* o profissional real é aquele que conhece os riscos, seus graus, suas causas e as acções necessárias para eliminá-los, e, partilha este conhecimento com seus colegas e clientes.

- *O princípio de prevenção do risco:* a prevenção do risco é mais custo-efectivo do que a detecção do risco.

- *O princípio da revelação do risco:* o grau do risco e suas causas, nunca devem ser ocultas aos tomadores de decisão.

[25] GILB, T.; Principles of Software Engineering Management; Addison Wesley, Reading, MA, 1988

- *O princípio de questionar:* se você não procurar a informação sobre riscos, você estará desejando problemas.

Para reduzir os riscos, os projectos devem obedecer a uma certa organização e que ela é baseada na experiência que o Gestor possue em relação aos projectos, pois, ele, deve ter experimentado algumas razões comuns para o fracasso dos projectos.