SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP
Data de Depósito: 10.12.2002
Assinatura:

GQS-AE - uma abordagem evolucionista para garantia de qualidade de software

Gláucio Giuliano Brogini

Orientadora: *Profa. Dra. Rosely Sanches*

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências de Computação e Matemática Computacional.

USP – São Carlos
Dezembro/2002

A Comissão Julgadora:

Profa. Dra. Rosely Sanches

Rosely Sanches

Profa. Dra. Sandra Camargo Pinto Ferraz Fabbri

Sandra C. P. F. Fabbri

Profa. Dra. Silvia Regina Vergilio

Silvia Regina Vergilio

“Tudo é uma questão de manter a mente quieta, a espinha ereta e o coração tranquilo.”
(Walter Franco)

AGRADECIMENTOS

A Deus, em primeiro lugar e acima de tudo.

À memória do meu pai, pois esta é mais uma das recompensas de **nossa** luta.

À minha mãe, pelo exemplo de dedicação, amor, perseverança e, sobretudo, de vida, que me guia durante todos os meus dias.

À minha orientadora, Prof.^a Dr.^a Rosely Sanches, pela confiança, pelo incentivo, pelo carinho e por outras tantas qualidades que fizeram com que nossa relação se transformasse em uma relação de valiosa amizade.

Aos meus irmãos e aos meus familiares, por compartilharem comigo todos os dissabores e as conquistas dessa vida.

Aos meus amigos de sempre, Claudete, Jorge, Luciana, Marcos, Mônica e Renata, por mais esta vida juntos; e àqueles que, mesmo não citados, também são dignos de agradecimento.

Às meninas da Biblioteca, Giselda, Gislene, Gláucia, Ivani, Juliana, Maria, Regina, Rose Casali, Rose Zambon e Sandra, pela confiança e amizade extremamente valiosas.

Ao pessoal da LinkWay, sobretudo ao Arnaldo, ao Alexandre e ao Luis, pela oportunidade e pela indispensável colaboração.

Ao CNPQ, pelo apoio financeiro.

E a todos aqueles que, direta ou indiretamente, colaboraram para que esta minha caminhada fosse sempre produtiva e nunca solitária.

RESUMO

A exigência por produtos de software com qualidade tem-se mostrado recorrente no mercado dos dias atuais. A fim de que essa qualidade possa se manifestar, vários processos que concorrem para a "fabricação" do software necessitam ser muito bem definidos, executados e gerenciados. É, pois, imperativo, que se tenha uma abordagem para *Garantia de Qualidade de Software*, bem planejada e sistemática, para ser utilizada na avaliação dos produtos e dos processos envolvidos no desenvolvimento de software, e que permita e assegure o uso dos resultados dessa avaliação pelos gerentes e desenvolvedores ao longo de todo o ciclo de desenvolvimento.

O que o presente trabalho propõe, então, é a abordagem *GQS-Æ* (uma abordagem evolucionista para garantir a qualidade do software), que descreve a forma como manipular, organizar e gerenciar os recursos – técnicos e humanos – disponíveis em uma empresa de software, sob uma base pré-estabelecida, a fim de que sejam alcançados os objetivos de melhoria da qualidade de tal software. Ressalta-se, porém, que essa abordagem foi concebida de forma simplificada e presa às restrições que normalmente apresenta uma pequena empresa, incapacitada, na grande maioria das vezes, de adaptar modelos já existentes por serem eles mais adequados à realidade de empresas de maior porte.

ABSTRACT

The demand for high-quality software products has shown itself to be recurring nowadays. In order to accomplish such quality, various processes that concur in software “manufacturing” have to be very well defined, carried out and managed. It is therefore imperative that there be a systematic Software Quality Assurance approach, catering for the evaluation of all the products and processes involved in software development and ensuring that evaluation results are lifecycle-widely available and useful to managers and developers.

In response, this work proposes *GQS- $\mathcal{AE}$* , an evolutionary software quality assurance approach describing how to organize and manage the resources – both human and technical – available in a software enterprise, on an aprioristic basis, so as to achieve quality enhancement goals. It is worth noticing that *GQS- $\mathcal{AE}$* has been designed to be simple and friendly to small businesses, which are seldom at ease to adapt existing models, usually best fit to bigger enterprises.

ÍNDICE GERAL

CAPÍTULO 1: INTRODUÇÃO..... 1

1.1 CONSIDERAÇÕES INICIAIS..... 1

1.2 OBJETIVOS 3

1.3 METODOLOGIA..... 5

1.4 ORGANIZAÇÃO DO TRABALHO..... 6

CAPÍTULO 2: GARANTIA DE QUALIDADE DE SOFTWARE..... 8

2.1 CONSIDERAÇÕES INICIAIS..... 8

2.2 VISÃO GERAL DE QUALIDADE 9

2.3 QUALIDADE APLICADA AO SOFTWARE..... 10

2.4 ALGUMAS VISÕES PARA GARANTIA DE QUALIDADE DE SOFTWARE 14

2.4.1 Roger S. Pressman..... 14

2.4.2 A Norma ISO 9001:2000..... 16

2.4.3 A Norma ISO 12207 20

2.4.3.1 Tarefas para Implementação do Processo..... 22

2.4.3.2 Tarefas para Garantia do Produto 23

2.4.3.3 Tarefas para Garantia do Processo 23

2.4.3.4 Tarefas para Garantia dos Sistemas de Qualidade..... 24

2.4.4 O Projeto SPICE (Futura Norma ISO 15504)..... 24

2.4.5 O CMM..... 29

2.4.5.1 A Estrutura Geral..... 30

2.4.5.2 A KPA Garantia de Qualidade de Software..... 32

2.4.5.3 Capability Maturity Model Integration - CMMI..... 35

2.5 CONSIDERAÇÕES FINAIS..... 39

CAPÍTULO 3: A ABORDAGEM EVOLUCIONISTA GQS-AE 40

3.1 CONSIDERAÇÕES INICIAIS..... 40

3.2 A EVOLUÇÃO COMO PREMISSA BÁSICA..... 41

3.3 A ESTRUTURA DA ABORDAGEM GQS-AE..... 42

3.4 SOBRE AS ATIVIDADES DOS PROCESSOS..... 45

3.5 RELACIONAMENTO ENTRE OS PROCESSOS 46

3.6 PAPEIS NA ABORDAGEM..... 50

3.7 OS PROCESSOS PRIMÁRIOS 53

3.7.1 Definição 53

3.7.2 Planejamento..... 54

3.7.3 Desenvolvimento 56

3.7.4 Operação..... 59

3.7.5 Manutenção 60

3.8 OS PROCESSOS DE APOIO	61
3.8.1 Documentação.....	62
3.8.2 Acompanhamento	62
3.8.3 Gerenciamento de Configuração.....	63
3.8.4 Controle de Qualidade	65
3.8.5 Verificação	66
3.8.6 Validação	68
3.8.7 Resolução de Problemas	68
3.9 OS PROCESSOS ORGANIZACIONAIS	69
3.9.1 Gerência.....	69
3.9.2 Capacitação.....	70
3.9.3 Melhoria.....	71
3.9.4 Infra-Estrutura	72
3.9.5 Treinamento.....	73
3.10 CONSIDERAÇÕES FINAIS.....	74
CAPÍTULO 4: APLICABILIDADE DA ABORDAGEM <i>GQS-AE</i>.....	75
4.1 CONSIDERAÇÕES INICIAIS.....	75
4.2 A EMPRESA LINKWAY.....	76
4.3 O PROCESSO-BASE PARA A INSTANCIAÇÃO	76
4.4 A INSTANCIAÇÃO.....	78
4.5 RESULTADOS QUANTITATIVOS	79
CAPÍTULO 5: CONCLUSÕES E TRABALHOS FUTUROS.....	82
5.1 CONCLUSÕES	82
5.2 TRABALHOS FUTUROS.....	84
REFERÊNCIAS BIBLIOGRÁFICAS	86
BIBLIOGRAFIA COMPLEMENTAR.....	90
APÊNDICE A.....	95
APÊNDICE B	121
APÊNDICE C	123
APÊNDICE D	125

ÍNDICE DE FIGURAS

FIGURA 1: ALGUMAS DEFINIÇÕES DE QUALIDADE DE SOFTWARE.....	11
FIGURA 2: REQUISITOS DA NORMA ISO 9001:1994.....	19
FIGURA 3: PRINCIPAIS REQUISITOS DA ISO 9001:2000.....	19
FIGURA 4: A ESTRUTURA DA NORMA ISO 12207.....	21
FIGURA 5: DIAGRAMA ESTRUTURAL DO PROJETO SPICE.....	25
FIGURA 6: A ESTRUTURA GERAL DO CMM - <i>CAPABILITY MATURITY MODEL</i>	32
FIGURA 7: ESTRUTURA DA ABORDAGEM <i>GQS-AE</i>	44
FIGURA 8: <i>TEMPLATE</i> PARA DEFINIR AS ATIVIDADES QUE COMPÕEM OS PROCESSOS DA ABORDAGEM <i>GQS-AE</i>	45
FIGURA 9: RELACIONAMENTO ENTRE PRODUTOS E ATIVIDADES DOS PROCESSOS.....	47
FIGURA 10: REPRESENTAÇÃO GRÁFICA DA ABORDAGEM <i>GQS-AE</i> , EM ESCALA REDUZIDA.....	48
FIGURA 11: RELACIONAMENTO ENTRE PAPEIS E ATIVIDADES DOS PROCESSOS.....	52
FIGURA 12: PROCESSO MELHORADO DE DESENVOLVIMENTO DE SOFTWARE - SHOPPING.....	77
FIGURA 13: GRÁFICOS DO CUMPRIMENTO DAS TAREFAS DESCRITAS NA ABORDAGEM <i>GQS-AE</i>	81

CAPÍTULO 1: INTRODUÇÃO

1.1 CONSIDERAÇÕES INICIAIS

"Não se pode vencer no mercado *de hoje* usando processos *de ontem*."

Dr. H. James Harrington

Embora a frase acima pareça de significado um pouco obscuro, o real poder que ela encerra só pode ser utilizado quando se descobre (e efetivamente se usa) o que está por trás desse conjunto ordenado de letras...

Há que se destacar, nela, dois termos de vital importância: **mercado** e **processos**. Além disso, com significado também importante, porém de compreensão mais direta, merecem destaque os termos metafóricos *de hoje* e *de ontem*. O primeiro deles refere-se aos dias atuais, ao presente. Já o outro, trata dos dias passados, de um tempo que ficou para trás. E quais seriam, então, as ligações existentes entre esses quatro termos? O **mercado *de hoje*** pode ser pensado como o momento econômico e tecnológico em que vivemos nos dias atuais, ditado pelas relações sociais existentes. Já os **processos** podem ser encarados como o conjunto de práticas que fazem a ligação entre relações sociais, economia e tecnologia, implicando numa mudança constante, embora por vezes pequena.

Assim como nos vários tipos de empresas, a influência dessas relações sociais e as mudanças econômicas, tecnológicas e até mesmo sociais que elas provocam também é visível quando se trata de empresas de software.

Uma das mudanças mais significativas que têm se processado nos últimos tempos é o poder dos consumidores de criar uma referência mais concreta sobre os produtos de software a serem adquiridos. Assim, eles têm deixado de se importar apenas com o preço para concentrar suas preocupações em obter um produto confiável, que atenda às suas necessidades e, sobretudo, que apresente qualidade.

Voltando aos termos que deram origem a essa explanação, pode-se perceber que a existência de qualidade nos produtos de software comercializados no **mercado** fica dependente das práticas utilizadas desde a concepção desse software até sua entrega, nas mãos dos clientes, ou, em outras palavras, fica dependente dos **processos** utilizados ao longo desse caminho. Sendo assim, ter-se um processo de desenvolvimento de software com qualidade é uma maneira de se ter um produto de software com qualidade (LAMPRECHT, 1994).

Entretanto, um tal processo é bastante amplo, englobando vários outros *subprocessos*¹, que devem ser muito bem definidos, executados e administrados. Além disso, todos eles precisam ser passíveis de mudanças.

Ademais, cabe ressaltar que o desenvolvimento de um software exige muito mais do que as noções simplificadas de começo, meio e fim utilizadas cotidianamente. Nesse processo são necessários ciclos complexos e repetitivos de planejamento, execução, verificação e correção, que ocorrem em cada uma das várias fases do processo de desenvolvimento de software. Esses ciclos são completamente interdependentes, devendo haver trocas de informações entre eles. As informações que passam de um ciclo a outro ao longo das fases, sendo modificadas constantemente, são chamadas de **produtos intermediários**.

Sendo assim, a questão da qualidade não é mais, simplesmente, obter um produto de software com qualidade, ao final do processo de desenvolvimento, como num passe de mágica. A qualidade deve, também, estar presente nos **produtos intermediários**. E isso,

¹ Na prática, um *processo* é decomposto em *subprocessos* que, por sua vez, podem, ainda, também ser decompostos. Para efeitos didáticos, porém, e a fim de facilitar a compreensão do que será exposto aqui, o termo *processo* será usado, intercambiavelmente, como *subprocesso*, tendo-se em mente, é claro, as devidas proporções de escopo.

como visto anteriormente, implica em se conceber um processo com qualidade, ou seja, um processo que forneça, durante sua execução, produtos [intermediários] com qualidade. É com base nessa premissa que se estabelece a garantia de qualidade de software – GQS.

A importância de se garantir a qualidade de um software é tão notória que, apesar das visões de GQS da maioria das normas e modelos de padronização e melhoria de processo – incluindo-se aqui o Capability Maturity Model – CMM (PAULK *et al*, 1995), o Software Process Improvement and Capability dEtermination – SPICE (ISO TR 15504, 1998) e as normas da International Standard Organization – ISO (ISO 9001, 1994; ISO 9001, 2000; ISO 12207, 1995) – apresentarem aspectos variados de cada um desses modelos, elas apresentam alguns pontos em comum, sobretudo ao destacarem tal atividade como um dos requisitos básicos para que a melhoria de processo de software possa ser levada a efeito.

Assim, a garantia de qualidade passa a ter uma dimensão bastante significativa na "fabricação" do software. Mas, afinal, o que se depreende de GQS?

A GQS pode ser definida, de maneira informal, como uma técnica planejada e sistemática usada para avaliação do software. Tal avaliação garante que os produtos de software estejam adequados a padrões de produtos – previamente definidos – e que os processos de desenvolvimento também estejam adequados, por sua vez, a procedimentos – também previamente definidos. Tanto os padrões quanto os procedimentos devem ser estabelecidos e seguidos durante o processo de desenvolvimento do software, o que é feito por meio de controle dos processos, de avaliações dos produtos e de auditorias periódicas (FALLAH, 1994). E isso deve ser feito de maneira que seus resultados possam ser utilizados pelos gerentes ao longo de todo o processo de desenvolvimento.

1.2 OBJETIVOS

Como a GQS engloba mais do que os aspectos técnicos do processo de desenvolvimento de software, é preciso a mobilização de todos os recursos – sejam eles técnicos ou humanos, com destacada atenção para estes últimos – de que dispõe a empresa. Entretanto, a dificuldade se apresenta na maneira como manipular, organizar e gerenciar todos os recursos, sob uma base pré-estabelecida, de forma a se alcançar os objetivos de melhoria da qualidade do

software. Essa dificuldade intensifica-se devido às próprias restrições apresentadas pelas pequenas empresas, tais como número reduzido de profissionais, condições financeiras precárias para a expansão de suas atividades, falta de novos conhecimentos e tecnologias compatíveis com suas necessidades, dentre outras².

O que se pretende, então, com este trabalho, é propor uma abordagem³ de GQS (a qual convencionou-se chamar de **Abordagem Evolucionista para Garantia de Qualidade de Software – GQS-AE**) não tão particular, a ponto de que seu uso seja restrito a uns pouquíssimos cenários, nem tão genérica, que possa ser usada como verdade universal. Mas uma abordagem a ser utilizada para a manipulação, organização e gerenciamento de todos os processos (e seus produtos) praticados por uma pequena empresa de desenvolvimento de software. Para tanto, há que serem obedecidas três diretrizes básicas, citadas a seguir, e que serão discutidas em momento oportuno, a saber:

- O objetivo da abordagem é o de garantir a qualidade do software produzido. Em adição, pretende-se que, por meio do uso iterativo dessa abordagem, os níveis de qualidade sejam constantemente incrementados.
- A abordagem deve ser passível de mudança, prevendo não só a melhoria dos processos que ela abrange, mas também de sua própria estrutura e funcionamento.
- A utilização desse ou daquele modelo de desenvolvimento de software (entenda-se ciclo de vida) não deve obstar a aplicação da abordagem.

A abordagem será representada, com o intuito de facilitar sua compreensão, na forma de diagramas de *workflow*⁴ a serem descritos por meio de uma notação própria, concebida para essa finalidade. Propostas de mudança nessa estrutura de representação caracterizam-se como melhorias a serem incorporadas à abordagem, e constituem-se num ponto a ser colocado no rol de trabalhos futuros.

² Não é tarefa deste texto discorrer sobre as peculiaridades de uma empresa, sobretudo aquelas que caracterizam seu porte, uma vez que todo esse desenvolvimento é, de certa forma, estruturado sobre uma base subjetiva.

³ Optou-se, aqui, pelo uso do termo *abordagem*, dada a natureza e os objetivos da GQS considerados neste trabalho. Entretanto, fica, o leitor, facultado a substituir o termo *abordagem* pelo termo *processo*, caso ele se sinta mais à vontade com tal termo.

⁴ *Workflow* são atividades envolvendo execução coordenada de múltiplas tarefas, realizadas por diferentes entidades de processamento (CASATI *et al.*, 1995). É a ordem em que os passos devem ser executados para produzir algo. Essa ordem pode ser serial, paralela e/ou condicional. Um passo em um *workflow* é um conjunto de ações, executadas por uma ou mais pessoas, que conduzem a uma determinada saída (resultado) (BELT, 1995).

Além do desenvolvimento da abordagem *GQS-Æ*, o presente trabalho também descreve uma instanciación⁵ da abordagem proposta, para verificar a aplicabilidade da mesma.

1.3 METODOLOGIA

Segundo VERA (1983), a escolha de um problema relevante e a identificação de suas contingências determinam, em boa parte, o tipo de pesquisa⁶ a ser desenvolvida. Escolhido o tipo, resta definir qual será o método de abordagem do problema para o qual é proposta uma solução científica.

Face às dificuldades que as pequenas empresas de software apresentam na definição, execução e gerenciamento dos processos que contribuem para que o produto de software tenha qualidade, é necessário que exista uma abordagem por meio da qual essas empresas garantam, em seu software, a qualidade por que clama o mercado consumidor (assegurando a permanência das mesmas nesse mercado), fazendo uso dos recursos – técnicos e humanos – que elas têm disponíveis. Tal é o problema para o qual o presente trabalho propõe como solução a abordagem *GQS-Æ*.

Tendo-se em vista suas particularidades, verificou-se que o problema encerrava as características de uma **pesquisa-ação**, por meio da qual devem ser analisados resultados da interação do ambiente com as soluções propostas para o problema. Além disso possuiu uma natureza qualitativa, abrigando algumas técnicas de interpretação com o intuito de descrever, decodificar, traduzir e uma série de outros termos relacionados com o entendimento e não com a frequência de ocorrência do fenômeno caracterizado pelo problema [apresentado].

Quanto ao método de abordagem, os mais comuns para se coletar dados na pesquisa qualitativa são a observação participativa, a entrevista não-estruturada ou semi-estruturada e o exame de documentos. A observação participativa permite ao pesquisador ganhar

⁵ Realizada no Departamento de Pesquisa e Desenvolvimento de Novas Tecnologias da LinkWay, empresa de prestação de serviços que atua no segmento de conexão a Internet, desenvolvimento de aplicações para Web e serviços de telecomunicações, autorizada pela ANATEL (Agência Nacional de Telecomunicações). Esse departamento fica localizado na cidade de São Carlos/SP. A instanciación tomou por base o processo de desenvolvimento de software da empresa, como definido em TAVARES (2002).

⁶ O termo *pesquisa* é genericamente assumido como o trabalho empreendido metodologicamente, quando surge um problema, para o qual se procura a solução de natureza científica adequada.

conhecimento do comportamento e da comunicação das pessoas por meio de prolongada imersão no ambiente estudado. A entrevista não-estruturada ou semi-estruturada procura descobrir a forma de pensar das pessoas. O exame de documentos, por sua vez, permite complementar as outras técnicas e verifica a validade dos dados, além de permitir acesso a outras informações.

Dessa forma, o método adotado para o desenvolvimento da abordagem *GQS-AE* foi, substancialmente, o de exame de documentos de elevada importância na área (referentes ao CMM (PAULK *et al*, 1995), ao Projeto SPICE (ISO TR 15504, 1998) e às normas ISO (ISO 9001, 1994; ISO 9001, 2000; ISO 12207, 1995)). Já no estudo de caso para verificação da aplicabilidade da abordagem, lançou-se mão da observação participativa e de algumas entrevistas com profissionais da empresa. Tais entrevistas, apesar de informais, tiveram natureza tanto não-estruturada quanto semi-estruturada.

1.4 ORGANIZAÇÃO DO TRABALHO

O restante deste documento encontra-se dividido conforme segue.

O **Capítulo 2 – Garantia de Qualidade de Software** introduz, de maneira sucinta, o papel e a importância dessa atividade dentro do processo de desenvolvimento de software, apresentando diferentes visões para ela, de destacado interesse no desenvolvimento deste trabalho (como será observado no capítulo referente aos processos constituintes da abordagem).

O capítulo seguinte, **Capítulo 3 – A Abordagem Evolucionista *GQS-AE***, apresenta uma descrição textual da abordagem proposta, indicando os produtos a serem gerados, e discutindo todos os processos que a compõem, bem como as condições para utilização de todos esses processos.

Uma instanciação dessa abordagem, proposta para a empresa LinkWay e gerada com base no processo definido em TAVARES (2002), é apresentada e comentada no capítulo seguinte, intitulado **Capítulo 4 – Aplicabilidade da Abordagem *GQS-AE***.

No **Capítulo 5 – Conclusões e Trabalhos Futuros** são apresentadas as conclusões sobre o desenvolvimento e a aplicação da abordagem e tecem-se comentários relacionados a trabalhos futuros com base neste.

Seguem-se as **Referências Bibliográficas** utilizadas na confecção deste texto e a **Bibliografia Complementar** que serviu para dissipar dúvidas surgidas ao longo do desenvolvimento do trabalho.

Por fim, apresentam-se os apêndices. No **Apêndice A** estão os *templates* (fichas) que descrevem de maneira detalhada cada uma das atividades contidas nos processos da abordagem. O **Apêndice B** mostra a tabela do relacionamento entre esses processos e os produtos produzidos/utilizados por eles. No **Apêndice C** encontra-se a representação gráfica da abordagem, contendo os diagramas de *workflow*. No **Apêndice D**, último deles, encontra-se o questionário utilizado na instânciação da abordagem.

CAPÍTULO 2: GARANTIA DE QUALIDADE DE SOFTWARE

2.1 CONSIDERAÇÕES INICIAIS

Antes do século XX, garantir a qualidade era uma responsabilidade exclusiva do artesão que desenvolvia um produto. A primeira função formal de controle e garantia de qualidade foi introduzida nos Laboratórios Bell em 1916 e espalhou-se por todo o mundo da manufatura. Hoje, todas as empresas têm mecanismos para garantir a qualidade de seus produtos. (PRESSMAN, 2002)

A história da garantia de qualidade no desenvolvimento de software tem paralelo com a história da qualidade na manufatura de hardware. Durante os primórdios da computação (décadas de 1950 e 1960), a qualidade era uma responsabilidade exclusiva do programador. Padrões de garantia de qualidade para o software foram introduzidos no ciclo de desenvolvimento sob contrato militar durante a década de 1970 e espalharam-se rapidamente para o desenvolvimento de software no mundo comercial (IEEE, 1989) *apud* (PRESSMAN, 2002).

O objetivo deste capítulo é apresentar, pois, algumas visões para GQS, com base nas quais foi elaborada a abordagem *GQS-~~AE~~*. Procurou-se, ao estudar tais visões, aproveitar e adaptar os melhores pontos de cada uma delas, condensando-os na abordagem desenvolvida.

2.2 VISÃO GERAL DE QUALIDADE

Na década de 80, o fator qualidade emergiu a nível mundial como uma necessidade básica na luta pelo mercado cada vez mais competitivo (BARROS, 1991). Hoje em dia, não basta vender barato; as novas regras de mercado são orientadas à produção de bens e serviços com qualidade, prazo de entrega determinado, atendimento correto, além de um baixo custo (WERNECK, 1994).

Qualidade, no entanto, é um termo definido ambigualmente e em diferentes situações. Além disso, de acordo com a opinião ou o enfoque de quem faz uso, diferentes significados podem ser atribuídos a ele (TOLEDO, 1987). O termo faz parte da linguagem cotidiana e a visão popular que se tem do conceito de qualidade pode ser muito diferente da forma como esse conceito é usado profissionalmente.

Na visão popular, a qualidade é pensada como:

- **Algo abstrato, sem vida própria, indefinido.** Nesse sentido, ela seria algo inatingível, que não poderia ser medido e nem avaliado através de uma base objetiva.
- **Perfeição, de maneira definitiva e imutável.** Assim, a qualidade refletiria a situação de se ter atingido um patamar máximo, não podendo mais ser alterada.
- **Classe ou categoria de um produto.** Sendo a qualidade refletida em diferentes modelos, pode-se pensar que bastaria agregar novos componentes para que sua qualidade fosse melhorada.
- **Luxo e questão de gosto.** Dessa forma, produtos caros, sofisticados e mais complexos seriam considerados de maior qualidade que produtos similares, porém mais simples.

Em contrapartida à visão popular, ROTHERY (1993), numa visão profissional, define qualidade como adequação ao uso. Ou seja, um produto deve se ajustar ao fim a que se

destina, estar em conformidade com suas exigências, devendo ser adequadamente projetado e fabricado para atender à função determinada.

CROSBY (1979), também numa visão profissional, já definira qualidade de um modelo de forma semelhante a ROTHERY (1993). Para aquele, a qualidade deve ser entendida como conformidade aos requisitos, o que implica em se referir à qualidade por meio de termos específicos, ou seja, de itens que podem ser mensurados. Os requisitos devem, pois, ser claramente expostos, a fim de que não sejam mal-interpretados. Assim, no processo de desenvolvimento e de produção, medidas são feitas continuamente para verificar a adequação a esses requisitos. As não-conformidades encontradas são, então, consideradas como defeitos, como falta de qualidade.

Há uma série de outras visões de qualidade, além das duas citadas aqui, o que implica na existência, por sua vez, de muitas definições e conceitos de qualidade. Apesar disso, não há entrave para a compreensão do conceito. Apenas alguns conflitos quando da sua aplicação. De qualquer forma, tais conflitos podem ser minimizados ao se analisar a questão não só globalmente, mas também nas várias fases de produção, sem que se perca de vista o objetivo de se conseguir um produto com qualidade.

2.3 QUALIDADE APLICADA AO SOFTWARE

Com a constante demanda gerada pela vida moderna, os computadores passam a integrar a rotina diária em uma escala cada vez maior. Como consequência, a produção de software apresenta um aumento constante, fazendo com que a exigência por qualidade estenda-se à área de software. Do ponto de vista dos fornecedores, a qualidade já não é mais um fator de vantagem no mercado, mas uma condição necessária e, pode-se dizer, indispensável para que seja possível competir com sucesso.

Desde os tempos remotos, muitos problemas no desenvolvimento dos sistemas computacionais já se faziam sentir, motivo pelo qual, em 1968, o Comitê de Ciências da OTAN (Organização do Tratado do Atlântico Norte) reuniu 50 especialistas, cientistas e profissionais da indústria de software para discutir possíveis soluções para o que passou a ser conhecido como a Crise do Software (OLIVEIRA, 1995).

Foi nesse encontro que se firmou o termo Engenharia de Software, tendo sido definida formalmente a necessidade da aplicação de uma abordagem sistemática, disciplinada e quantificável para o desenvolvimento, operação e manutenção de produtos de software. Segundo KAN (1995), observando a Engenharia de Software através de uma perspectiva histórica, a década de 60 e os anos que a antecederam podem ser chamados de *Era Funcional*; os anos 70, de *Era do Modelo*; os anos 80, de *Era do Custo* e os anos 90 são conhecidos como a *Era da Qualidade*, dada a exigência crescente por essa característica nos produtos de software.

Sendo qualidade um termo que pode ter diferentes interpretações – como discutido anteriormente, com o software não seria diferente. Faz-se necessário, inicialmente, obter um consenso em relação à definição de qualidade de software que está sendo abordada.

Existem muitas definições propostas na literatura, sob diferentes pontos de vista. O quadro a seguir apresenta algumas delas.

Figura 1: Algumas definições de qualidade de software.

PRESSMAN (2002) define qualidade de software com mais detalhes. Segundo ele, a “qualidade de software é a conformidade a requisitos funcionais e de desempenho que foram explicitamente declarados, a padrões de desenvolvimento claramente documentados e a características implícitas que são esperadas de todo software desenvolvido por profissionais”.

Essa definição enfatiza três aspectos importantes. Primeiramente, os requisitos são a base a partir da qual a qualidade é medida. A falta de conformidade aos requisitos significa falta de qualidade. Além disso, padrões especificados definem um conjunto de critérios de desenvolvimento que orientam a maneira segundo a qual o software passa pelo trabalho de engenharia. Se os critérios não forem seguidos, o resultado quase que seguramente será a falta de qualidade. E, por fim, existe um conjunto de requisitos implícitos que não são freqüentemente mencionados na especificação (por exemplo, o desejo de uma boa manutenibilidade). Se o software se adequar aos seus requisitos explícitos, mas deixar de cumprir seus requisitos implícitos, a qualidade do software pode ser comprometida.

Há que se acrescentar, ainda, uma definição de qualidade de software baseada no ponto de vista gerencial. Segundo ela, o software desenvolvido dentro do prazo e do orçamento especificados pode ser um software de alta qualidade.

Com base nisso tudo, qualidade de software pode ser definida como um conjunto de atributos de software que devem ser satisfeitos de modo que o software possa atender às necessidades do usuário (que pode ser um usuário final, um desenvolvedor ou uma organização), sejam elas explícitas ou não.

A determinação dos atributos relevantes para cada software varia em função do domínio da aplicação, da tecnologia utilizada, das características específicas do projeto e das necessidades do usuário e da organização.

O usuário final e a organização avaliam o software sem conhecer seus aspectos internos; estão apenas interessados na facilidade de uso, no desempenho, na confiabilidade dos resultados obtidos, no preço do software e também no cumprimento de prazos estabelecidos no cronograma. Já os desenvolvedores avaliam aspectos internos como taxa de defeitos, confiabilidade, facilidade de manutenção e também aspectos de conformidade em relação aos requisitos dos clientes. Essas visões de qualidade de software são diferentes, mas concordam que o software não pode ter defeitos (NBR 13596, 1996).

Raramente a qualidade pode ser incorporada ao produto final, após o processo de desenvolvimento ter terminado. Dos requisitos do usuário à entrega do produto final, existe um processo de desenvolvimento complexo, que envolve uma série de estágios que podem comprometer a qualidade desse produto final. Em cada fase do processo de desenvolvimento,

um produto intermediário é produzido para um usuário intermediário – a próxima fase. Se esse produto intermediário, de alguma forma, não satisfaz aos requisitos do usuário ou algum requisito implícito, ele irá comprometer a qualidade do produto dessa fase. Portanto, cada produto intermediário tem certos atributos de qualidade que afetam a qualidade do produto intermediário da próxima fase e que, por conseguinte, afetam a qualidade do produto final (KAN, 1995).

Por essa razão, a qualidade do produto de software é um objetivo do processo de desenvolvimento. Assim, ao desenvolver-se um produto, é preciso que estejam estabelecidas, previamente e como perspectiva, as características de qualidade que se deseja alcançar. Se o processo de desenvolvimento de software considerar as características de qualidade, pode-se dizer que o produto final deverá apresentar essas características.

Um processo bem estabelecido, compreendido e controlado pode ajudar na obtenção de qualidade do produto, desde que sejam definidos claramente os requisitos de qualidade desse produto (ANDRADE *et al*, 1996).

Não se deixando de lado a integração entre produtos e processos, estudos envolvendo qualidade de software têm sido realizados de forma incessante, com o intuito de garantir, de forma cada vez mais segura, a qualidade do software desenvolvido. Muitas das conclusões a que chegam tais estudos podem ser encontradas nas normas e modelos de padronização e melhoria de processo de software disponíveis hoje em dia, que destacam a GQS como um requisito indispensável na melhoria do processo de desenvolvimento de software. Apesar das visões de GQS desses modelos apresentarem ligeiras diferenças entre si, o essencial é que haja um consenso quanto à finalidade da GQS, sem que seja esquecida a meta primeira: a produção de software com qualidade.

2.4 ALGUMAS VISÕES PARA GARANTIA DE QUALIDADE DE SOFTWARE

2.4.1 ROGER S. PRESSMAN

Segundo PRESSMAN⁷ (2002), para se garantir a qualidade do software é necessária a aplicação de uma variedade de tarefas associadas a sete grandes atividades: (1) aplicação de métodos técnicos, (2) realização de revisões técnicas formais, (3) atividades de teste de software, (4) aplicação de padrões, (5) controle de mudanças, (6) medição e (7) manutenção de registros [da qualidade] e reportagem.

A qualidade de software é projetada num produto ou sistema. Ela não é imposta após o fato. Por essa razão, a GQS inicia-se já com o conjunto de **métodos e ferramentas técnicas** que auxiliam na obtenção de uma especificação e um projeto de elevada qualidade.

Assim que uma especificação (ou protótipo) e um projeto tiverem sido criados, cada um deve ser avaliado quanto à qualidade. A atividade central que leva a efeito a avaliação da qualidade é a **revisão técnica formal**. Caracteriza-se por um encontro estilizado, realizado pelo pessoal técnico, com o propósito único de descobrir problemas de qualidade. Em muitas situações, tem-se percebido que tais revisões são tão boas quanto os testes, na descoberta de problemas relativos à qualidade.

A atividade de **teste de software** combina uma estratégia de múltiplos passos com uma série de métodos de projeto de casos de teste que ajudam a garantir uma detecção efetiva de erros. Muitos desenvolvedores de software usam a atividade de teste de software como uma "rede de segurança" de garantia de qualidade. Ou seja, os desenvolvedores pressupõem que uma atividade de teste cuidadosa revelará a maioria dos erros, abrandando a necessidade de outras atividades de GQS. Infelizmente, a atividade de testes, mesmo quando bem realizada, não é tão efetiva quanto se gostaria que fosse para todas as classes de erros (JONES, 1981) *apud* (PRESSMAN, 2002).

⁷ Roger S. Pressman é um autor e consultor reconhecido internacionalmente no campo de Engenharia de Software, tendo escrito muitos artigos técnicos, além de seis livros. Também contribui regularmente com revistas e periódicos ligados à indústria.

O grau em que **padrões e procedimentos** formais são aplicados no processo de engenharia de software varia de empresa para empresa. Em muitos casos, os padrões são determinados pelos clientes ou por imposições reguladoras. Em outros, são auto-impostos. Se existirem padrões formais, ou seja, padrões documentados, uma atividade de GQS deve ser estabelecida para garantir que eles sejam seguidos. Uma avaliação do cumprimento dos padrões deve ser levada a efeito pelos desenvolvedores de software, como parte de uma revisão técnica formal ou de uma auditoria interna.

Uma grande ameaça à qualidade de software vem de uma fonte aparentemente benigna: as mudanças. Toda mudança no software tem potencial para introduzir erros ou criar efeitos colaterais que propagam erros. O processo de **controle de mudanças** (uma atividade que faz parte do gerenciamento de configuração de software) contribui diretamente para a qualidade do software ao formalizar pedidos de mudança, avaliar a natureza da mudança e controlar seu impacto.

A **medição** é uma atividade que faz parte de qualquer disciplina de engenharia. Um objetivo importante da GQS é rastrear a qualidade e avaliar o impacto das mudanças metodológicas e procedimentais sobre essa qualidade de software. Para realizar isso, uma métrica deve ser coletada e diretrizes para sua avaliação devem ser definidas e seguidas.

A **anotação e manutenção de registros** para a GQS oferecem procedimentos para a coleta e disseminação de informações. Os resultados de revisões, auditorias, controle de mudanças, testes e outras atividades devem tornar-se parte do registro histórico de um projeto e ser levados ao conhecimento do pessoal de desenvolvimento.

Mas é realmente necessário se garantir que um software desenvolvido tenha qualidade? Todas as organizações de desenvolvimento de software têm algum mecanismo para avaliação da qualidade que faça uso de alguma das atividades citadas anteriormente? De certa forma, sim. Na parte inferior da escala, a qualidade é uma responsabilidade exclusiva do indivíduo, que pode projetar, revisar e testar em qualquer nível de satisfação. Na parte superior da escala, um grupo de GQS deve assumir a responsabilidade de estabelecer padrões e procedimentos para conseguir qualidade de software e assegurar que cada um seja seguido. A questão real para cada organização de software é "Onde nos encontramos na escala?".

Antes que procedimentos formais de GQS sejam instituídos, uma empresa de desenvolvimento de software deve adotar procedimentos, métodos e ferramentas de Engenharia de Software. Essa metodologia, quando combinada com um paradigma efetivo para o desenvolvimento de software, pode contribuir – e muito – para melhorar a qualidade de todo o software produzido por tal organização.

2.4.2 A NORMA ISO 9001:2000

A primeira versão das normas internacionais ISO 9001, 9002 e 9003 para a garantia da qualidade foi lançada em 1987. Uma primeira revisão, para melhorar o entendimento dessas normas, foi publicada em 1994 (ISO 9001, 1994; NBR 9000-3, 1997). A ISO 9000-3, publicada em 1991 e revisada em 1997, estabeleceu diretrizes para a aplicação da ISO 9001 ao desenvolvimento, ao fornecimento e à manutenção de software (ISO 9001, 1994; NBR 9000-3, 1997).

Tais normas especificam os requisitos mínimos para que as empresas possam assegurar a qualidade de seus produtos e serviços, porém não definem modelos ou sistemas de qualidade específicos a serem adotados ou implementados.

Uma nova revisão do conjunto de normas ISO 9000 foi processada, em 2000, alterando sensivelmente tanto a morfologia quanto a fisiologia desse conjunto: os requisitos da ISO 9001 foram completamente reestruturados e as normas ISO 9002 e 9003 deverão ser canceladas. Entretanto, ainda é incerto o destino da ISO 9000-3, que estabelece diretrizes para a aplicação da ISO 9001 em software.

A ISO 9001:2000 está baseada em oito princípios-guia para o gerenciamento da qualidade, derivados da experiência e do conhecimento de profissionais que fazem parte do comitê técnico ISO/TC 176, responsável pelo desenvolvimento e pela manutenção dos padrões ISO 9000. Cabe ressaltar que esses princípios-guia foram definidos para serem utilizados pela gerência das organizações como um *framework* para melhorar o desempenho dessas organizações. Existem muitas formas diferentes de se aplicar os princípios de gerenciamento da qualidade. A natureza da organização e os desafios específicos que ela enfrenta irão determinar a maneira como implementá-los. Tais princípios são discutidos a seguir.

- **Foco no cliente.** Como as organizações dependem de seus clientes, elas devem entender não só suas necessidades atuais, mas também as futuras, procurando satisfazer os requisitos desses clientes e esforçando-se para superar as expectativas deles. Para tanto, é necessário que a empresa realize medições da satisfação do cliente – utilizando sempre tais resultados – e mantenha um gerenciamento sistemático das relações empresa-cliente.
- **Liderança.** A liderança estabelece unidade de propósitos, além da direção que seguirá a empresa. É expressamente recomendado que os líderes criem e mantenham um ambiente interno no qual as pessoas possam estar completamente envolvidas, a fim de alcançarem os objetivos estabelecidos pela empresa. Dessa forma, a má comunicação entre os diferentes níveis da empresa poderá ser minimizada. Para tanto, faz-se necessário que os líderes forneçam às pessoas os recursos, treinamento e a liberdade necessários para agirem com responsabilidade dentro de um projeto e, sobretudo, dentro da empresa. Além disso, é preciso, dentre outras coisas, que esses mesmos líderes também inspirem, encorajem e reconheçam as contribuições pessoais.
- **Envolvimento das pessoas.** O pessoal, de todos os níveis, é a essência de uma organização. Assim, o completo envolvimento dessas pessoas permite que suas habilidades possam ser utilizadas em benefício da própria empresa. Para que isso seja possível, é preciso que as pessoas entendam sua importância dentro da empresa; que elas identifiquem as restrições ao seu desempenho, encarando a responsabilidade de seus próprios atos; que elas preocupem-se com oportunidades de melhorar sua competência, seu conhecimento e sua experiência. Além disso, deve haver um compartilhamento explícito de conhecimento e de experiências, sobretudo na discussão de problemas e de questões relevantes.
- **Abordagem de processos.** Um resultado que se deseja alcançar pode ser obtido mais eficazmente quando as atividades envolvidas e os recursos necessários são encarados sob a perspectiva de processo. Nesse caso, uma avaliação constante dos riscos, das conseqüências e dos impactos dessas atividades sobre os clientes, fornecedores e outras partes interessadas deve ser realizada. É necessária, ainda, uma preocupação com os recursos, métodos e materiais que irão melhorar as atividades desenvolvidas na empresa.
- **Abordagem de sistemas para a gerência.** Identificar, gerenciar e entender como um sistema os processos inter-relacionados. Isso contribui para que a empresa possa alcançar, mais efetivamente, seus objetivos. Para isso, abordagens estruturadas que

integrem os processos devem ser utilizadas. Uma melhora contínua dos sistemas estabelecidos pode ser obtida por meio do uso contínuo de medições e avaliações.

- **Melhoria contínua.** Melhorar continuamente seu desempenho global deve ser um objetivo permanente da empresa. Na medida em que isso for posto em prática, uma flexibilidade de reação mais rápida às oportunidades começará a se manifestar. No entanto, é necessário, entre outras coisas, que se mantenham os processos, os produtos e os sistemas da organização em melhoria contínua, com o reconhecimento dessa melhoria, sempre que ela acontecer.
- **Decisão com base em fatos.** Decisões efetivas são aquelas baseadas na análise de dados e de informações. Entretanto, os dados e as informações devem ser garantidamente precisos e confiáveis e estar disponíveis para aqueles que deles fazem uso. Dessa forma, tomadas de decisão e propostas de ação estarão baseadas numa análise fatural balanceada pela intuição e pela experiência das pessoas envolvidas.
- **Relações de benefício mútuo entre empresa e fornecedor.** Uma organização e seus fornecedores são interdependentes, de maneira que uma relação de benefício mútuo sempre aumenta o valor das partes na relação. Invariavelmente, existirá, então, uma otimização dos custos e dos recursos. Para que isso seja possível, é necessário uma comunicação clara e aberta entre empresa e fornecedores, além do desenvolvimento conjunto das atividades, sobretudo as de melhoria.

Com a nova ISO 9001:2000, os sistemas de gestão da qualidade ISO 9000 passam a dispensar uma maior atenção ao processo de melhoria contínua e à satisfação do cliente. De certa forma, isso coloca a GQS numa dimensão um pouco mais ampla, relacionando-a, agora, ao atendimento tanto dos requisitos explícitos quanto dos requisitos implícitos do cliente.

As figuras que se seguem apresentam os requisitos da ISO 9001, nas suas versões 1994 e 2000. É importante notar a reestruturação ocorrida e, principalmente, a mudança de objetivo.

• Estruturas	<ul style="list-style-type: none"> • Responsabilidade pela administração • Análise crítica pela administração • Controle de documentos e dados • Produto fornecido pelo cliente • Controle de produto não-conforme • Controle de registros da qualidade • Treinamento • Técnicas estatísticas
• Melhorias	<ul style="list-style-type: none"> • Análise crítica pela administração • Ação corretiva e preventiva • Auditoria interna
• Qualidade	<ul style="list-style-type: none"> • Análise crítica do contrato • Controle do projeto • Aquisição • Identificação e rastreabilidade • Controle de processo • Inspeção e ensaio • Controle de equipamentos de inspeção, medição e ensaio • Situação da inspeção e do ensaio • Preservação, armazenamento, embalagem e entrega • Serviços associados

Figura 2: Requisitos da norma ISO 9001:1994.
Referência: Traduzido de ISO 9001 (1994).

5. Responsabilidade da administração 5.1. Comprometimento da administração 5.2. Foco no cliente 5.3. Política da qualidade 5.4. Planejamento 5.4.1. Objetivos da qualidade 5.4.2. Planejamento da qualidade 5.5. Administração 5.5.1. Generalidades 5.5.2. Responsabilidades e autoridade 5.5.3. Representante da administração 5.5.4. Comunicação interna 5.5.5. Manual da qualidade 5.5.6. Controle de documentos 5.5.7. Controle de registros 5.6. Análise crítica do sistema	7. Realização do produto 7.1. Planejamento dos processos de realização 7.2. Processos relativos ao cliente 7.3. Projeto e/ou desenvolvimento 7.4. Aquisição 7.5. Operações de produção e serviços 7.5.1. Controle das operações 7.5.2. Identificação e rastreabilidade 7.5.3. Propriedade do cliente 7.5.4. Preservação do produto 7.5.5. Validação de processos 7.6. Controle de dispositivos de medição e monitoramento
6. Gerenciamento do produto 6.1. Provisão de recursos 6.2. Recursos humanos 6.2.1. Designação de pessoal 6.2.2. Treinamento, conscientização e competência 6.3. Infra-Estrutura 6.4. Ambiente de trabalho	8. Medição, análise e melhoria 8.1. Planejamento 8.2. Medição e monitoramento 8.2.1. Satisfação do cliente 8.2.2. Auditorias internas 8.2.3. Medição e monitoramento dos processos 8.2.4. Medição do produto 8.3. Controle de não-conformidades 8.4. Análise de dados 8.5. Melhoria 8.5.1. Planejamento para melhoria continua 8.5.2. Ação corretiva 8.5.3. Ação preventiva

Figura 3: Principais requisitos da ISO 9001:2000.
Referência: Extraído de ROCHA et al (2001).

De acordo com a revisão 2000 da ISO 9001, as atividades de controle, de medição e de validação devem dar suporte à garantia da qualidade. Entretanto, para que isso seja possível, fazem-se necessários o planejamento e a confecção de um manual da qualidade que contenha procedimentos para a identificação, coleta, indexação, armazenamento e manutenção de registros da qualidade. Tanto um quanto outro são considerados como sendo de **responsabilidade da administração**.

2.4.3 A NORMA ISO 12207

O padrão ISO 12207⁸ (1995) é um padrão, publicado em 1995 como norma internacional, que estabelece uma estrutura comum para os processos do ciclo de vida de software, com uma terminologia bem definida, e composto de processos, atividades e tarefas para aquisição, fornecimento, desenvolvimento, operação e manutenção do software. A norma estabelece uma arquitetura de alto nível para o ciclo de vida do software que abrange desde a concepção até a descontinuação do mesmo. Essa arquitetura é baseada em processos-chave e no inter-relacionamento entre eles, seguindo dois princípios. Pelo princípio da *modularidade*, os processos têm alta coesão e baixo acoplamento, ou seja, todas as partes de um processo são fortemente relacionadas e o número de interfaces entre os processos é mantido ao mínimo. O outro princípio, o da *responsabilidade*, prega que cada processo na norma é de responsabilidade de uma ‘parte envolvida’, a qual pode ser uma organização ou parte dela. As ‘partes envolvidas’ podem ser da mesma organização ou de organizações diferentes (ROCHA *et al*, 2001).

A norma classifica os processos do ciclo de vida de software em três categorias, a saber: *Processos Fundamentais*, *Processos de Apoio* e *Processos Organizacionais*. Existem, ainda, o que se chama de *Processos de Adaptação*. Eles tratam exatamente da cultura e dos aspectos intrínsecos da organização, auxiliando na adaptação da norma a um projeto específico. Cada processo é definido em termos de suas próprias atividades, e cada atividade é adicionalmente definida em termos de suas tarefas (ROCHA *et al*, 2001). A classificação pode ser mais bem visualizada na *Figura 4*, a seguir.

⁸ O padrão ISO 12207 será referência em relação aos processos para a futura Norma ISO 15504 (Projeto SPICE).

Figura 4: A estrutura da Norma ISO 12207.
Referência: Traduzido de ISO 12207 (1995).

Este padrão ainda fornece diretrizes que devem ser empregadas para definição, controle e melhoria dos processos de desenvolvimento do software.

De acordo com a norma ISO 12207 (1995), o processo de GQS tem o objetivo de fornecer garantia adequada de que os produtos de software e os processos do ciclo de vida envolvidos no projeto estão de acordo com os requisitos especificados e com os planos estabelecidos (que incluem tanto os padrões quanto os procedimentos). A GQS pressupõe liberdade organizacional e autoridade das pessoas diretamente responsáveis pelo desenvolvimento de determinado produto ou pela execução de certo processo, e pode ser interna ou externa, dependendo da qualidade de processo e/ou de produto demonstrada pelo fornecedor ou exigida pelo consumidor. É oportuno assinalar o relacionamento que existe entre o processo de *Garantia de Qualidade* e os processos de *Verificação*, *Validação*, *Revisões* e *Auditorias*. Segundo a norma, depreende-se que a garantia de qualidade é realizada

através da união desses cinco processos. Eis o porquê de existir uma região – que engloba os cinco processos – em destaque na *Figura 4*.

O processo de GQS compreende quatro classes principais de atividades:

- atividades relativas à implementação do processo;
- atividades relativas à garantia do produto;
- atividades relativas à garantia do processo;
- atividades relativas à garantia dos sistemas de qualidade.

2.4.3.1 TAREFAS PARA IMPLEMENTAÇÃO DO PROCESSO

Para que o processo possa ser implementado, é indispensável que sejam realizadas as seguintes tarefas:

- deve-se assegurar que os produtos de software, bem como os processos usados para prover tais produtos, estejam de acordo com os requisitos e planos estabelecidos;
- deve haver uma coordenação entre o processo de garantia de qualidade e os processos relacionados a ele: *Verificação, Validação, Revisões e Auditorias*;
- deve-se desenvolver um plano para conduzir as atividades e tarefas envolvidas no processo de GQS. O plano deve ser documentado, implementado e mantido durante a vigência do contrato (que incide sobre o projeto a ser desenvolvido) e incluir o seguinte:
 - padrões de qualidade, metodologias, procedimentos e ferramentas para a realização das atividades de garantia de qualidade (ou suas referências na documentação oficial da organização);
 - procedimentos para revisões contratuais e a forma de coordenação de tais procedimentos;
 - procedimentos para identificação, coleta, preenchimento, manutenção e disposição de registros de qualidade;
 - recursos, cronogramas e responsabilidades para se levar a efeito as atividades de qualidade;
 - atividades e tarefas selecionadas dos processos de apoio, tais como *Verificação, Validação, Revisões, Auditorias e Resolução de Problemas*;
- devem ser concluídas atividades e tarefas programadas e/ou em andamento. Quando forem detectados problemas ou não-conformidades com os requisitos do contrato,

deve-se documentá-los para que sirvam como entrada para o processo de *Resolução de Problemas*. Devem ser preparados e mantidos registros de todas as atividades e tarefas, bem como de sua execução e dos problemas e suas resoluções;

- os registros das atividades e tarefas de garantia de qualidade devem estar disponíveis aos consumidores na forma como especificado em contrato;
- deve-se assegurar que as pessoas responsáveis por garantir a adequação aos requisitos do contrato tenham liberdade organizacional, recursos necessários e autoridade para permitir avaliações objetivas e para iniciar, executar e verificar as resoluções dos problemas.

2.4.3.2 TAREFAS PARA GARANTIA DO PRODUTO

A atividade de garantia do produto consiste nas seguintes tarefas:

- garantir que todos os planos previstos pelo contrato sejam documentados e estejam de acordo com o especificado neste. Além disso, os planos devem ser mutuamente consistentes e executados como requerido;
- deve-se assegurar que os produtos de software e a documentação referente a eles estejam de acordo com o contrato e em conformidade com os planos;
- na preparação para a entrega dos produtos de software, deve-se garantir que eles satisfaçam, de forma completa, aos requisitos contratuais.

2.4.3.3 TAREFAS PARA GARANTIA DO PROCESSO

A garantia do processo exige que:

- os processos do ciclo de vida do software (*Fornecimento, Desenvolvimento, Operação, Manutenção* e processos de apoio, incluindo-se o de *Garantia de Qualidade*) empregados no projeto estejam de acordo com o contrato e sejam fiéis ao plano;
- as práticas internas de Engenharia de Software, os ambientes de desenvolvimento e de teste e as bibliotecas utilizadas também estejam de acordo com o contrato;
- no caso de subcontratos, que sejam passados ao subcontratador os requisitos do contrato primário necessários na subcontratação, de maneira que os produtos de software gerados pela subcontratação possam satisfazer aos requisitos do contrato primário;

- sejam fornecidos, ao consumidor e às outras partes, suporte e cooperação necessários, de acordo com o contrato, com as negociações e com os planos;
- as medidas, tanto de produto quanto de processo de software, estejam de acordo com padrões e procedimentos estabelecidos;
- a equipe designada para o projeto tenha habilidade e capacidade necessárias para tomar conhecimento dos requisitos do projeto e receber treinamento, quando necessário.

2.4.3.4 TAREFAS PARA GARANTIA DOS SISTEMAS DE QUALIDADE

A garantia dos sistemas de qualidade preconiza a adequação de atividades adicionais de gerenciamento de qualidade às cláusulas de qualidade ISO 9000, sem que se perca de vista o que foi especificado no contrato.

2.4.4. O PROJETO SPICE (FUTURA NORMA ISO 15504)

O Projeto *Software Process Improvement and Capability dEtermination* – SPICE – foi criado em 1993 com o objetivo de gerar normas para avaliação de processos de software, visando à melhoria contínua desses processos e à determinação de sua **capacidade**⁹. O SPICE foi projetado para satisfazer as necessidades dos clientes, dos fornecedores e dos avaliadores, e os requisitos individuais de cada um. A versão atual do SPICE foi aprovada em 1998, como Relatório Técnico, tendo previsão de publicação como Norma ISO 15504 em 2002.

A estrutura SPICE presta-se à realização de avaliações de processos de software com dois objetivos: a melhoria de processos e a determinação da capacidade de processos de uma organização (ROCHA *et al*, 2001).

Se o objetivo for a determinação da capacidade do processo, a avaliação fornece meios para identificar a capacidade de processos selecionados na unidade organizacional, tomando como base um modelo de referência. Essa análise comparativa serve tanto para identificar os

⁹ A **capacidade de um processo** é o espectro de resultados esperados que podem ser alcançados seguindo-se esse processo, inicialmente estabelecido a nível organizacional. É, ainda, um instrumento para previsão dos resultados de projetos futuros.

riscos envolvidos em empreender um novo projeto, usando os processos selecionados, quanto de motivação para melhoria desses processos. Se o objetivo for avaliar um fornecedor em potencial, obtendo o seu perfil de capacidade, a organização deve definir os objetivos e o contexto da avaliação, os modelos e métodos de avaliação e os requisitos esperados. O perfil de capacidade permite ao contratante estimar o risco associado à contratação daquele fornecedor em potencial para auxiliar na tomada de decisão de contratá-lo ou não.

Se o objetivo for a melhoria de processo, a avaliação fornece meios de caracterizar a prática atual aplicada na unidade organizacional, em termos da capacidade de processos selecionados, identificando mudanças que devem ser efetuadas para melhorá-los. A organização deve definir os objetivos e o contexto para a avaliação, deve escolher o modelo e o método para a avaliação e deve definir os objetivos de melhoria.

A versão atual do Projeto SPICE constitui-se num documento estruturado em nove partes e que proporcionam os meios para examinar e avaliar processos de software, conforme mostrado na *Figura 5*.

Figura 5: Diagrama estrutural do Projeto SPICE.

Referência: Traduzido de ISO TR 15504 (1998).

A *Parte 2 Modelo de Referência para Processo e Capacitação de Processo* define um modelo ideal de processo de software, apresentando as atividades fundamentais e essenciais a uma boa Engenharia de Software sem, no entanto, preocupar-se com a forma como tais atividades devem ser implementadas.

O *Modelo de Referência* estabelecido na norma 15504 define duas dimensões: a *dimensão de processo*, que corresponde à definição de um conjunto de processos considerados universais e fundamentais para a boa prática da Engenharia de Software¹⁰ e a *dimensão de capacidade*, que corresponde à definição de um modelo de medição com base na identificação de um conjunto de atributos que permite determinar a capacidade de um processo para atingir seus propósitos gerando os produtos de trabalho e os resultados estabelecidos (ISO TR 15504, 1998; MOREAU, 1998) *apud* (ROCHA *et al*, 2001).

Na *dimensão de processo*, os processos são idealizados e descritos em cinco categorias de acordo com o tipo de atividades que executam. Tais atividades são denominadas de *práticas básicas*. As categorias encontram-se listadas a seguir:

- **Cliente-Fornecedor.** Categoria que abrange processos cujo impacto se manifesta diretamente no cliente e na entrega do software para esse cliente. Compreende atividades de aquisição, fornecimento e operação (uso e suporte ao cliente) de software.
- **Engenharia.** Contém os processos ligados à especificação, implementação e manutenção de um sistema e do software. Compreende atividades de análise dos requisitos do sistema e do software, desenvolvimento do projeto de software, implementação, integração e teste do software, integração e teste do sistema e manutenção do software e do sistema.
- **Gerência.** Nesta categoria estão os processos necessários ao estabelecimento do projeto e que coordenam e gerenciam os recursos necessários para se obter um produto ou se fornecer um serviço que garanta a satisfação do cliente. Incluem-se, aqui, atividades de gerenciamento, sobretudo dos riscos, dos requisitos e do projeto, bem como o planejamento do ciclo de vida do projeto e o estabelecimento do plano de projeto, além de atividades de gerenciamento da qualidade.
- **Suporte.** Categoria que inclui processos a serem empregados para facilitar e/ou possibilitar a execução dos demais processos (incluindo-se os próprios processos desta categoria). Abrange as atividades de documentação, resolução de problemas, revisões conjuntas, verificação, validação e auditorias, bem como práticas referentes à realização do gerenciamento da configuração e da garantia da qualidade.
- **Organização.** Os processos desta categoria auxiliam na definição dos objetivos de negócio da empresa e, estando definidos tais objetivos, concorrem, através da

¹⁰ Nesse caso, a norma ISO 12207 está sendo incorporada à norma ISO 15504, como parte das diretrizes normativas desta.

utilização dos recursos disponíveis, para que se alcance o que foi estabelecido como meta. Estão nesta categoria práticas relativas a reuso, infra-estrutura, treinamento e alinhamento da organização. Existem, ainda, atividades de definição e melhoria dos processos utilizados pela empresa no desenvolvimento de software.

Segundo ROCHA *et al* (2001), cada processo é descrito em termos de um propósito que exprime um único objetivo funcional. Satisfazer esse propósito é o primeiro passo na efetivação do mesmo. O modelo não define como, ou em que ordem, os propósitos de determinado processo podem ser alcançados. Esses propósitos são avaliados pela constatação da execução das atividades, tarefas e práticas necessárias para gerar os produtos de trabalho pertinentes ao propósito estabelecido. As tarefas, atividades e práticas, bem como as características dos produtos de trabalho, são definidas como indicadores que demonstram se determinado processo é adequadamente praticado em um determinado nível de capacidade. Esses indicadores, que representam características do processo, são classificados de acordo com uma escala que expressa justamente o seu nível de capacidade.

A *dimensão de capacidade* estabelece a graduação dessa escala em seis níveis. São eles: *Nível 0 – Incompleto, Nível 1 – Executado, Nível 2 – Gerenciado, Nível 3 – Estabelecido, Nível 4 – Previsível e Nível 5 – Em Otimização.*

Uma das vantagens da 15504 é a grande diversidade de formatos de apresentação dos resultados, normalmente com muito mais riqueza de detalhes que o CMM (PAULK *et al*, 1993), que fornece um único número para representar o nível de maturidade dos processos da organização. Um dos possíveis formatos de apresentação mostra o perfil de capacidade típico para os processos selecionados, ilustrando, para cada processo, o grau de atendimento aos requisitos associados aos níveis de capacidade. (ROCHA *et al*, 2001)

A preocupação com a qualidade, de acordo com a estrutura SPICE, está bastante clara nas categorias **Gerência** e **Suporte**. Na primeira, existe o processo de *Gerenciamento da Qualidade*, cujo foco principal é a identificação do que necessita ser feito para se imbuir a qualidade nos produtos. Para tanto, procura estabelecer mecanismos de gerenciamento que assegurem que isso seja feito. Na segunda categoria, **Suporte**, existe um outro processo, com nome similar ao citado anteriormente, porém com propósito diferente. Trata-se do processo de *Realização da Garantia da Qualidade*, cujo foco é voltado mais para uma abordagem de revisões e auditorias, procurando garantir a conformidade dos produtos e dos processos. A

verificação e validação dos produtos também são realizadas no sentido de colaborar para que a garantia de qualidade seja levada a efeito.

O propósito do processo de *Gerenciamento da Qualidade* é gerenciar a qualidade dos produtos e serviços envolvidos no projeto, de forma a garantir que os produtos e serviços resultantes consigam satisfazer às necessidades do cliente. Gerenciar a qualidade implica em identificar as características de qualidade necessárias aos produtos, trabalhar para se alcançar essa qualidade e demonstrar que ela foi alcançada, quando isso realmente acontecer.

As *práticas básicas* deste processo são:

- **estabelecer as metas da qualidade.** Com base nos requisitos de qualidade demonstrados pelo cliente, devem ser estabelecidas metas de qualidade a serem alcançadas em vários ‘pontos de avaliação’ do projeto (isto é, ao final de cada fase).
- **definir métricas de qualidade.** Devem ser definidas métricas para se medir os resultados das atividades do projeto, a fim de se verificar se as metas de qualidade relevantes foram alcançadas.
- **identificar as atividades de qualidade.** Para cada meta de qualidade estabelecida, atividades que irão auxiliar na obtenção dessas metas devem ser identificadas e integradas ao modelo de desenvolvimento do software.
- **realizar as atividades de qualidade.** Devem ser realizadas as atividades identificadas na *prática básica* anterior.
- **avaliar a qualidade.** Nos ‘pontos de avaliação’ identificados no projeto, as métricas definidas pela *prática básica* **definir métricas de qualidade** devem ser aplicadas, a fim de se verificar se as metas de qualidade relevantes foram alcançadas.
- **adotar postura corretiva.** Quando as metas de qualidade não são atingidas, posturas corretivas devem ser adotadas. Tais posturas corretivas podem envolver a correção do produto gerado por uma atividade particular dentro do projeto ou a mudança do conjunto de atividades planejadas, ou ambas as estratégias, para se atingir, de forma melhor, as metas de qualidade estabelecidas.

Por outro lado, o processo de *Realização da Garantia da Qualidade* tem como propósito garantir que os produtos de trabalho e as atividades estejam em conformidade com todos os padrões e procedimentos aplicáveis e respeitem os requisitos definidos. A premissa básica, aqui, é que se determine e que se relate uma visão objetiva da qualidade do processo e dos produtos de trabalho.

As *práticas básicas* deste processo são:

- **selecionar padrões de projeto.** Deve ser avaliada a completitude dos planos envolvidos no projeto, além de serem selecionados padrões que serão usados no desenvolvimento desse projeto.
- **revisar as atividades de Engenharia de Software.** Devem ser revisadas, com base nos planos e nos padrões definidos, as atividades de Engenharia de Software envolvidas no projeto.
- **realizar auditorias nos produtos de trabalho.** Os produtos de trabalho também devem ser avaliados, com base nos padrões estabelecidos para o projeto.
- **relatar os resultados.** Os resultados das práticas mencionadas anteriormente devem ser relatados, sobretudo nos casos em que houver desvios, ao grupo responsável pela atividade e principalmente à gerência competente.
- **cuidar dos desvios.** Os desvios devem ser levados ao nível apropriado da gerência, indo para um nível superior, se necessário, até que sejam dissipados.

2.4.5 O CMM

O modelo CMM – *Capability Maturity Model* – foi desenvolvido pelo SEI – *Software Engineering Institute*, ligado à Universidade Carnegie Mellon. De acordo com PAULK *et al* (1995) e HUMPHREY (1989), seu objetivo é determinar o nível de maturidade¹¹ do processo de produção do software. Esse nível está relacionado à competência e qualidade do processo (originando o nome CMM), ou seja, está relacionado à extensão na qual um processo de software específico é eficiente, explicitamente definido, gerenciado, medido e controlado. Competência também implica potencial para crescimento, riqueza do processo e consistência com a qual ele é aplicado em projetos na organização.

O CMM classifica as organizações em cinco níveis crescentes de maturidade (1 – *Caótico*, 2 – *Repetível*, 3 – *Definido*, 4 – *Gerenciado*, 5 – *Otimizado*), por meio de um instrumento de avaliação representado na forma de questionário. Tal divisão permite que sejam descritos fundamentos sucessivos para melhoria contínua do processo e que uma escala ordinal para se medir a maturidade de processo de uma organização possa ser definida.

¹¹ A **maturidade** de uma organização pode ser vista como a sua capacidade em definir, gerenciar e melhorar seus processos. Ela tem impacto direto na possibilidade de se alcançar metas de custo, qualidade e cronograma.

Uma das vantagens dos níveis de maturidade é que, por meio deles, prioridades claras são estabelecidas. Tais prioridades orientam na seleção de algumas atividades de melhoramento que serão muito úteis se implementadas imediatamente. Isso é importante porque permite à maioria das organizações focalizar somente algumas poucas atividades de melhoramento de cada vez.

2.4.5.1 A ESTRUTURA GERAL

O CMM é um modelo que representa um caminho de melhoramentos recomendados para organizações de software que querem aumentar a **capacidade de seus processos** de software. Cada nível do modelo é, portanto, um conjunto de passos dados nesse caminho.

No *Nível 1 – Inicial*, o processo de software da organização é caracterizado como *ad hoc*, sendo, por vezes, caótico. Isso se deve à imprevisibilidade desse processo, causada pelas suas constantes alterações no decorrer no projeto. As organizações que se enquadram neste nível de maturidade são classificadas como **organizações caóticas**.

O *Nível 2 – Repetível* é caracterizado pela existência de processos efetivos de planejamento e gerenciamento do projeto de software, nos quais os controles sobre procedimentos, compromissos e atividades são bem fundamentados. Esses processos devem ser documentados, treinados e controlados. Entretanto, no *Nível 2* ainda não há uma preocupação com o processo de Engenharia de Software. Organizações pertencentes a este nível de maturidade são **organizações disciplinadas**.

O *Nível 3 – Definido* é caracterizado principalmente pela existência de um processo de Engenharia de Software bem definido, documentado e que serve de padrão para a empresa. Nesse processo-padrão, as saídas de uma atividade fluem naturalmente para as entradas da próxima atividade e tanto tarefas administrativas quanto de Engenharia encontram-se integradas. Aqui se enquadram as chamadas **organizações padronizadas**.

As organizações do *Nível 4 – Gerenciado* possuem processos de software passíveis de medida. Tanto a produtividade quanto a qualidade são medidas em todas as etapas do processo de software e para todos os projetos da organização e reguladas por meio de métricas

de controle. As organizações que se enquadram neste nível de maturidade são classificadas como **organizações previsíveis**.

O *Nível 5 - Otimizado* é caracterizado pela existência de processos de software com melhoria contínua. Tais processos são avaliados para prevenir tipos de defeitos conhecidos devido à recorrência, e as lições aprendidas são disseminadas para outros projetos. Neste nível, um ambiente de excelência em Engenharia de Software foi atingido.

Os níveis de maturidade contêm, por sua vez, diferentes áreas-chave de processo (KPAs). Cada área-chave de processo foi definida para constar ou pertencer a **um único nível de maturidade** e identifica um grupo de atividades relacionadas que, quando realizadas coletivamente, atingem um conjunto de metas consideradas importantes para o aumento da capacidade do processo.

Abaixo das áreas-chave de processo encontram-se o que se chama de práticas-chave, utilizadas para descrever as primeiras. Tais práticas procuram apresentar as atividades e a infra-estrutura que mais contribuem para a efetiva implementação e institucionalização das áreas-chave de processo. Elas tratam “do que” é para ser feito e não “como” o processo deve ser implementado.

Por conveniência, as práticas-chave são organizadas em características comuns. Essas características são atributos que indicam se a implementação e institucionalização de uma área-chave de processo são eficazes, repetíveis e estáveis. As características comuns são: *Comprometimento para a realização, Condições para a realização, Atividades realizadas, Medições e análises e Verificação de implementação*.

2.4.5.2 A KPA GARANTIA DE QUALIDADE DE SOFTWARE

A *Garantia de Qualidade de Software* é uma KPA pertencente ao **Nível 2 – Repetível** do CMM. Além desta, fazem parte desse nível, como KPAs, *Gerenciamento de Requisitos*, *Planejamento de Projeto de Software*, *Acompanhamento de Projeto de Software*, *Gerenciamento de Subcontrato de Software* e *Gerenciamento de Configuração de Software*. Discorre-se sobre ela, seguindo-se a estrutura geral proposta pelo modelo CMM (*Figura 6*) que, obviamente, é a mesma para todas as KPAs.

Figura 6: A estrutura geral do CMM – *Capability Maturity Model*.

Referência: Traduzido de PAULK *et al* (1995).

Desde os primeiros estágios de um projeto de software, deve haver um grupo especializado em GQS com o objetivo de estabelecer planos, padrões e procedimentos que irão satisfazer às restrições do projeto e às políticas da organização. Ademais, fazendo isso, o grupo de GQS ajuda a garantir que tais planos, procedimentos e padrões obedeçam às necessidades do projeto e que sejam úteis na realização de revisões e auditorias a serem realizadas ao longo do processo de desenvolvimento.

A área-chave de processo *Garantia de Qualidade de Software* abrange as práticas para que o grupo de GQS realize sua função. As práticas que identificam as atividades específicas e os produtos de software que o grupo revisa ou nos quais realiza auditorias estão, normalmente, contidas na característica comum *Verificação de implementação*.

As **metas** para essa KPA podem ser descritas como:

- A GQS deve ser planejada.
- A adequação das atividades e produtos de software aos padrões, procedimentos e requisitos aplicáveis deve ser verificada objetivamente.
- As pessoas e grupos afetados devem ser informados das atividades e dos resultados da GQS.
- Não-conformidades que não puderem ser resolvidas dentro do escopo do projeto de software devem ser deixadas a cargo da gerência.

Como *Comprometimento para a realização* deve ser seguida uma política organizacional documentada para a implementação da GQS. Essa política normalmente especifica que a função de GQS deve “estar na posição correta” em todos os projetos de software. Além disso, especifica que o grupo de GQS deve ter um canal de comunicação com a gerência que seja independente¹² do gerente de projeto, do grupo de Engenharia de Software do projeto e de outros grupos relacionados¹³. Por fim, ela também especifica que tanto as atividades quanto os resultados de GQS devem ser periodicamente revisados pela gerência.

Como *Condições para a realização* citam-se quatro:

- É preciso existir um grupo responsável por coordenar e implementar a GQS no projeto (isto é, o grupo de GQS).
- Os recursos adequados e fundos necessários para as atividades de GQS também precisam existir.
- Os membros do grupo de GQS precisam ser treinados para a realização de suas atividades.
- Os membros do projeto precisam receber orientação relativa aos papéis, responsabilidades, autoridades e valores do grupo de GQS.

¹² As organizações devem determinar a estrutura organizacional que vai auxiliar nas atividades que requerem independência (tais como GQS) no contexto de suas estratégias e do ambiente de negócio. Tal independência deve fornecer aos indivíduos que participam da GQS a liberdade organizacional para que eles sejam “os olhos e os ouvidos” da gerência. Por outro lado, ela fornece a essa gerência a segurança de que informações objetivas do processo e dos produtos de software estão sendo fornecidas.

¹³ Exemplos de outros grupos relacionados incluem os grupos de gerenciamento da configuração de software e de suporte à documentação.

A lista de *Atividades realizadas* deve incluir:

- Um plano de GQS é preparado de acordo com um procedimento documentado. Tal procedimento normalmente especifica que o plano deve ser desenvolvido nos estágios iniciais de planejamento global do projeto e em paralelo com este, que esse mesmo plano deve ser revisado pelos grupos e indivíduos que por ele são afetados e, principalmente, que esse plano seja gerenciado e controlado¹⁴.
- As atividades do grupo de GQS são realizadas em conformidade com o plano, que cobre:
 - as responsabilidades e a autoridade do grupo de GQS;
 - as necessidades de recursos para o grupo (incluindo pessoal, ferramentas e instalações);
 - o cronograma e fundos para as atividades do grupo;
 - a participação do grupo no estabelecimento do plano de desenvolvimento de software, nos padrões e nos procedimentos;
 - as avaliações a serem realizadas pelo grupo, bem como as auditorias e revisões a serem conduzidas por ele;
 - os padrões e procedimentos do projeto a serem usados como base para as revisões e auditorias;
 - os procedimentos¹⁵ para documentação e verificação de não-conformidades;
 - a documentação que deve ser produzida pelo grupo;
 - as formas e a frequência com que serão fornecidos “feedbacks” ao grupo de engenharia e aos outros grupos relacionados.
- O grupo de GQS toma parte na preparação e revisão do plano de desenvolvimento do projeto de software, dos padrões e dos procedimentos.
- O grupo ainda revisa as atividades de engenharia e os produtos de trabalho com o intuito de comprovar a aquiescência dos mesmos.
- O grupo faz relatórios periódicos dos resultados de suas atividades ao grupo de Engenharia de Software.
- Desvios identificados nas atividades e nos produtos de trabalho são documentados e manipulados de acordo com um procedimento próprio (e também documentado);

¹⁴ *Gerenciado e controlado* implica em se conhecer a versão do produto de trabalho em uso num dado momento (passado ou presente), ou seja, há um controle de versão, e em se controlar a forma como são incorporadas as mudanças que ocorrem, isto é, existe um controle de mudanças.

¹⁵ Pode-se incluir esses procedimentos como uma parte do plano ou por meio de referências a outros documentos nos quais eles estejam contidos.

- O grupo efetua revisões periódicas de suas atividades e descobertas junto ao cliente, da forma que melhor se adequar.

Quanto às *Medições e análises*, elas devem ser realizadas e utilizadas para se determinar o *status* de custo e de cronograma das atividades de GQS.

Como *Verificação de implementação* destacam-se alguns aspectos, a saber:

- As atividades de GQS devem ser revisadas junto à gerência com periodicidade. O propósito primordial dessas revisões periódicas é fornecer conhecimento das atividades envolvidas no processo de software num nível apropriado de abstração e de uma forma bastante oportuna. O tempo entre revisões deve estar de acordo com as necessidades da organização, podendo ser longo, desde que mecanismos adequados para reportagem das exceções estejam disponíveis.
- A revisão das atividades com o gerente de projeto deve ser feita também periodicamente e, além disso, na medida em que os eventos acontecem.

2.4.5.3 CAPABILITY MATURITY MODEL II INTEGRATION - CMMI¹⁶

O CMM descrito anteriormente é mais propriamente chamado de *Software Capability Maturity Model (SW-CMM)*. Isto porque, na esteira de seu sucesso, diversos outros ‘CMMs’ foram criados, procurando cobrir outras áreas de interesse. Assim surgiram os seguintes modelos:

- *Software Acquisition Capability Maturity Model (SA-CMM)*: usado para avaliar a maturidade de uma organização em seus processos de seleção, aquisição e instalação de software desenvolvido por terceiros.
- *Systems Engineering Capability Maturity Model (SE-CMM)*: avalia a maturidade da organização em seus processos de engenharia de sistemas, concebidos como algo maior que o software. Um “sistema” inclui o hardware, o software e quaisquer outros elementos que participam do produto completo.
- *Integrated Product Development Capability Maturity Model (IPD-CMM)*: ainda mais abrangente que o SE-CMM, inclui também outros processos necessários à

¹⁶ Esta subseção corresponde ao texto extraído de (BELLLOQUIM, 2002), com algumas adaptações.

produção e suporte ao produto, tais como suporte ao usuário, processos de fabricação etc.

- *People Capability Maturity Model (P-CMM)*: avalia a maturidade da organização em seus processos de administração de recursos humanos no que se refere a software, recrutamento e seleção de desenvolvedores, treinamento e desenvolvimento, remuneração etc.

O surgimento de todos estes modelos gerou alguns problemas. Em primeiro lugar, nem todos usavam a mesma terminologia, de modo que um mesmo conceito podia receber nomes diferentes em cada modelo, ou que o mesmo termo quisesse dizer coisas diferentes nos vários modelos. Além disso, a estrutura carecia de um formato padrão; os modelos tinham diferentes números de níveis ou formas diferentes de avaliar o progresso. E, por fim, os altos custos de treinamento, avaliação e harmonização para organizações que tentassem usar mais de um modelo.

Por outro lado, a experiência no uso do CMM durante uma década serviu para identificar pontos em que o modelo poderia ser melhorado. Ao mesmo tempo, o surgimento do Projeto SPICE levou à necessidade de compatibilização do CMM com a futura norma ISO 15504 (ISO TR 15504, 1998).

Por todas essas razões, o SEI iniciou um projeto chamado *Capability Maturity Model Integration (CMMI)*. Seu objetivo era gerar uma nova versão do CMM que resolvesse esses problemas. Concretamente, a primeira idéia, como o nome sugere, é integrar os diversos 'CMMs' numa estrutura única, todos com a mesma terminologia, processos de avaliação e estrutura. Além disso, o projeto também se preocupou em tornar o CMM compatível com a ISO 15504, de modo que avaliações em um modelo pudessem ser reconhecidas como equivalentes às do outro. E, naturalmente, incorporar ao CMM as sugestões de melhoria surgidas ao longo dos anos.

Assim, o CMMI já está disponível e sendo usado em algumas organizações. Existem diversas versões, dependendo do nível de abrangência necessário (incluindo ou não os aspectos SE, IPD, etc.).

A principal mudança do CMMI em relação ao CMM é a possibilidade de utilização de duas diferentes abordagens para a melhoria de processos. Essas duas abordagens são conhecidas como o “modelo contínuo” e o “modelo em estágios”.

No “modelo em estágios” do CMMI – semelhante ao CMM – existem 22 áreas-chave de processo, e cada uma situa-se em apenas um nível. Assim, para uma organização estar no *Nível 2*, é necessário que as 7 KPAs desse nível estejam institucionalizadas. Para estar no *Nível 3*, é preciso cumprir as 7 KPAs do *Nível 2* e mais as 11 do *Nível 3*. E assim por diante. Uma organização no *Nível 2* pode, por exemplo, possuir práticas de níveis mais altos, mas ser apenas *Nível 2*, por não possuir o conjunto completo das KPAs do nível mais alto. A KPA *Garantia de Qualidade de Software* ainda se encontra no *Nível 2*, que no CMMI é caracterizado como **Gerenciado**, porém passou a ter um nome mais específico: *Garantia de Qualidade de Produto e Processo*.

Alguns dos ‘outros CMMs’ citados, bem como o modelo da ISO 15504, usam uma abordagem diferente, o chamado “modelo contínuo”. Nesse caso, cada KPA possui características relativas a mais de um nível. Assim, uma KPA que, no “modelo em estágios”, pertence exclusivamente ao *Nível 2*, no “modelo contínuo” pode ter características que a coloquem em outros níveis. Em outras palavras, no “modelo contínuo”, cada KPA é classificada separadamente, de modo que a organização pode ter KPAs no *Nível 1*, outras no *Nível 2*, outras, ainda, no *Nível 3* e assim por diante. Nesse modelo, também existe a KPA de garantia da qualidade, com o mesmo nome utilizado no “modelo em estágios”, ou seja, *Garantia de Qualidade de Produto e Processo*. E essa KPA, para o “modelo contínuo” é caracterizada como uma área de processo de **Apoio**. As demais áreas de processo são **Gerenciamento de Processo, Gerenciamento de Projeto e Engenharia**.

Em ambos os casos, a KPA *Garantia de Qualidade de Produto e Processo* implica em:

- avaliar objetivamente os processos empregados, os produtos desenvolvidos e os serviços, com base em descrições de processo, padrões e procedimentos aplicáveis;
- identificar, documentar e garantir que as não-conformidades encontradas sejam tratadas;
- levar ao conhecimento dos profissionais envolvidos no projeto, incluindo-se os gerentes, os resultados das atividades de garantia da qualidade;

Para dar liberdade a organizações com realidades e necessidades distintas, o CMMI admite as duas abordagens. A opção por “modelo contínuo” ou “modelo em estágios” depende de cada organização. Cada modelo possui características que o tornam apropriado em uma situação, mas não em outra.

De modo geral, pode-se dizer que o “modelo contínuo” é mais flexível, embora mais complexo de administrar. Por exemplo, uma organização que não tem necessidade premente de todas as KPAs, ou ao menos para a qual algumas dessas KPAs sejam mais importantes ou urgentes do que outras, usando o “modelo contínuo”, pode dar prioridade à melhoria dessas KPAs, com uso mais focado de seus recursos humanos e financeiros. Outra situação em que uma organização se beneficiaria do “modelo contínuo” é quando existe uma carência de suporte pela alta administração. Se um gerente de nível médio quer implantar o CMM, mas não consegue sensibilizar a alta administração, ele pode, em escala reduzida, implantar um conjunto menor de áreas-chave de processo, com maior rapidez e custo mais baixo, de modo a ser capaz de mostrar resultados que facilitem a ‘venda’ do CMM para a alta administração.

Por outro lado, o “modelo contínuo” pode dar a falsa impressão de que as KPAs do CMM são completamente independentes, que é possível atingir o *Nível 5* em algumas e o *Nível 0* em outras (sim, na representação contínua existe *Nível 0*, que significa que a KPA nem mesmo existe; se as práticas relativas à KPA existirem mas não estiverem formalizadas e institucionalizadas, a KPA será classificada como estando no *Nível 1*). Isto, evidentemente, não é verdade. Embora essa representação dê mais flexibilidade na hora de priorizar a implantação de práticas, existem dependências entre as diversas áreas que não podem ser ignoradas.

O “modelo em estágios”, por sua vez, é indicado para organizações realmente comprometidas com a implantação do CMM em escala geral. Organizações que já usam o CMM sentir-se-ão também mais confortáveis com ele. O fato de as áreas já estarem agrupadas da forma que faz mais sentido libera a organização de ter que decidir por diferentes possibilidades de priorização.

2.5 CONSIDERAÇÕES FINAIS

Neste capítulo, algumas visões de GQS foram apresentadas, com o intuito de servirem como base para a elaboração da abordagem proposta neste trabalho. Entretanto, trata-se apenas de uma visão superficial, dada a limitação em se reproduzir toda a riqueza de detalhes e de informações contida nos documentos utilizados. Considerações bastante gerais sobre o CMMI e a evolução dos modelos CMM, também foram feitas, porém apenas para fins de atualização da revisão bibliográfica.

De qualquer forma, pode-se destacar alguns pontos de convergência entre as visões sobre as quais se discorreu. Citam-se:

- nenhuma visão específica a maneira **como** suas diretrizes devem ser aplicadas em cada organização. Essa maneira depende única e exclusivamente da própria organização, variando sensivelmente de uma para outra;
- todas as visões recomendam (note-se aqui o uso do termo *'recomendar'*, cujo significado é bastante diverso daquele expresso, por exemplo, pelo termo *'exigir'*), ainda que implicitamente, a existência de um grupo de GQS para atuar dentro do escopo do projeto a ser desenvolvido. Esse grupo deve trabalhar desde o início do projeto, no estabelecimento de planos, padrões e procedimentos que irão satisfazer às restrições apresentadas e às políticas da organização;
- deve haver um planejamento para que se possa garantir a qualidade do software produzido, além de mecanismos para avaliação e realinhamento de curso, caso tenha havido algum desvio na tentativa de se alcançar os objetivos de garantia de qualidade definidos;
- não se pode pensar em GQS como uma atividade órfã ou independente. É necessária uma estrutura procedimental¹⁷ para sua execução, capaz de explicitar o papel da GQS no desenvolvimento de software e a forma como ela está relacionada com todas as demais atividades.

Faz-se mister ressaltar, porém, que existem bastantes aspectos de divergência entre as visões apresentadas. No entanto, tais aspectos não foram comentados aqui, pois o próximo capítulo – que apresenta a abordagem *GQS-AE* desenvolvida – constitui-se em melhor oportunidade para que os comentários sobre divergências sejam desenvolvidos.

¹⁷ Como se pensa ao se escrever programas em uma linguagem de programação estruturada. Os conceitos de chamada e retorno de procedimento, valores de retorno, parâmetros de entrada e saída, recursão podem muito bem ser abstraídos, encontrando condições bastante propícias para que os conceitos sejam aplicados quando se trata de GQS.

CAPÍTULO 3: A ABORDAGEM EVOLUCIONISTA *GQS-AE*

3.1 CONSIDERAÇÕES INICIAIS

Geralmente, o maior problema por que passam as pequenas empresas de software é *garantir a qualidade dos produtos que desenvolvem*. Dadas as limitações de recursos – tanto técnicos quanto humanos, é difícil para elas fazerem uso de normas e modelos de padronização e melhoria de processo, pelas próprias características restritivas que apresentam tais estruturas. O fato é que essas normas e modelos não são concebidos para serem usados, de maneira direta, por outros tipos de empresas, que não sejam as de grande porte. Não obstante, a realidade brasileira, em que a maioria das empresas de software existentes possui características de micro e pequeno portes (MCT/SEPIN, 2002), carece de uma abordagem para garantia de qualidade de software especialmente voltada às características de tais empresas.

Desviando-se ligeiramente do problema principal – a qualidade, pode-se verificar, com grande recorrência, outros problemas que permeiam o ciclo de desenvolvimento de software. São problemas de *comunicação*, que, na maior parte das vezes, é feita de maneira desastrosa dentro da empresa, sem rotinas formais pré-estabelecidas de compartilhamento de dados e informações relevantes à execução dos processos. Problemas com a *melhoria dos processos*

utilizados, que costuma não ocorrer por ser desconhecido ou indefinido o estado em que se encontra cada processo. Problemas com *custos e cronogramas*, sempre subestimados por um planejamento desordenado e incompleto. Problemas na *obtenção ininterrupta de conhecimento de novas tecnologias, treinamento e oportunidades compatíveis com as necessidades das empresas*, gerando a possibilidade de competir com empresas maiores dentro do mercado consumidor. Além de outros, tantos, que poderiam ser compilados em algumas dezenas de páginas.

De qualquer forma, tais problemas podem ser reduzidos. Basta que se tenha uma abordagem para garantia de qualidade de software especialmente adaptada à realidade das pequenas empresas, oferecendo uma melhor compreensão dos termos, procedimentos e das sugestões utilizadas.

Tal abordagem é apresentada neste capítulo. Trata-se de uma versão inicial, a ser melhorada, dado seu caráter evolucionista. Nas seções seguintes, discorre-se sobre as características gerais dessa abordagem, através de considerações sobre sua estrutura, sobre o relacionamento entre os processos que a compõem, bem como sobre os papéis necessários para a correta execução das atividades envolvidas. Às últimas seções coube, por sua vez, apresentar, de maneira detalhada, cada uma dessas atividades.

3.2 A EVOLUÇÃO COMO PREMISA BÁSICA

O objetivo da abordagem é garantir a qualidade do produto de software final. Consegue-se tal intento por meio da garantia da qualidade dos produtos de software intermediários e melhorando-se os processos envolvidos no ciclo de desenvolvimento do software. Isso posto, procurou-se conceber a abordagem com base numa perspectiva evolucionista, ou seja, baseada na evolução. Tal evolução ocorre sob dois aspectos, o que assegura a melhoria contínua dos processos e a conseqüente qualidade dos produtos. Sob um primeiro aspecto, a abordagem propõe a evolução dos processos praticados na organização, indicando mudanças a serem realizadas para um melhor desempenho dos processos. Isso é possível pela coleta e avaliação das métricas definidas, para cada atividade, e com base em objetivos específicos e bem definidos de melhoria. Ao mesmo tempo, porém sob outro aspecto, alterações evolutivas podem ser feitas tanto na estrutura da abordagem quanto no seu funcionamento. Essas

alterações podem, por exemplo, variar do simples acréscimo de um fluxo de informação à decomposição de uma atividade em outras atividades menores. Podem, também, consistir de realocação de tarefas, ou mesmo de redefinição de papéis. Ou, ainda, de alterações quanto à ordem e à forma de execução das atividades dentro da abordagem.

Outra característica relacionada à evolução, e que também se procurou contemplar na elaboração da abordagem, foi a independência quanto ao modelo de desenvolvimento de software (ciclo de vida) adotado pela empresa. Isso é possível porque, independentemente do paradigma escolhido ou mesmo da área de aplicação, do tamanho do projeto ou de sua complexidade, o “produzir um software” inclui três fases genéricas: a *definição* (cujo foco é o ‘o quê’), o *desenvolvimento* (que se preocupa em definir o ‘como’) e a *manutenção* (que diz respeito às mudanças ocorridas, sejam elas de natureza corretiva, adaptativa ou perfectiva). A abordagem *GQS-AE* foi, portanto, construída com base nas fases genéricas do desenvolvimento do software.

3.3 A ESTRUTURA DA ABORDAGEM *GQS-AE*

A **Abordagem Evolucionista para Garantia de Qualidade de Software – *GQS-AE*** define uma estrutura a ser seguida pela empresa para pôr em prática as atividades envolvidas na realização de um projeto. O objetivo da abordagem é o de garantir a qualidade do software produzido. Essa qualidade, através do uso iterativo da abordagem, será sempre melhorada. Para que tal objetivo seja alcançado, a abordagem foi concebida de maneira a fornecer (não só à gerência da empresa, mas a todos os profissionais que tomam parte nas atividades envolvidas na produção do software) a visibilidade apropriada, tanto das atividades, tecnologias, práticas e métodos que estão sendo usados no desenvolvimento quanto do produto que está sendo desenvolvido (PAULK *et al*, 1993). A estrutura dos processos que compõem a abordagem segue aquela utilizada pela norma ISO 12207 (1995), que define as atividades do ciclo de vida do software através de cinco processos primários, oito processos de apoio e quatro processos organizacionais, agrupados da seguinte maneira:

- Os **processos primários** definem as atividades relativas à construção do software. Correspondem à aquisição, fornecimento, desenvolvimento, operação e manutenção de software.

- Os **processos de apoio** são executados ao longo da realização dos processos primários e são responsáveis pelo sucesso dos projetos. Referem-se à documentação, Gerenciamento de Configuração, garantia da qualidade, verificação, validação, revisões e auditorias e resolução de problemas.
- Os **processos organizacionais** estabelecem a estrutura requerida para a realização das tarefas dos outros processos. Suas atividades ocorrem não necessariamente durante a realização de um projeto de software e correspondem à gerência, infraestrutura, melhoria e treinamento.

Para o caso específico deste trabalho, alguns processos foram redefinidos e, às três categorias, foram introduzidos outros processos, visto que a abordagem proposta é preferencialmente destinada a empresas de pequeno porte. As alterações realizadas na estrutura da ISO 12207, para que se adequasse às características exigidas por este trabalho, são listadas a seguir:

- O processo de *Aquisição* foi substituído por um processo de *Definição* do problema. Tal alteração baseou-se no fato de que, no ciclo de desenvolvimento de software das pequenas empresas, dificilmente ocorre a aquisição de um sistema ou produto de software.
- O processo de *Fornecimento* deu lugar ao processo de *Planejamento*. O raciocínio é o mesmo. Faz-se necessário, antes de um processo para fornecimento de um produto de software, que a empresa tenha – bem definido – um processo para o planejamento das atividades de desenvolvimento do seu software.
- O processo de *Garantia da Qualidade* teve seu nome alterado para *Controle de Qualidade*, uma vez que, no escopo deste trabalho, a garantia da qualidade do software é função da abordagem como um todo. Um processo, sozinho, falharia na tentativa de garantir qualidade. Além disso, o processo definido está relacionado à qualidade dos processos primários e de apoio, somente. Já a abordagem cuida da qualidade de todos os processos, incluindo-se, aqui, os organizacionais.
- Os processos de *Revisão Conjunta* e *Auditoria* deram lugar ao processo de *Acompanhamento*, o qual faz uso de revisões e auditorias para verificar o andamento das atividades do ciclo de vida do software.

- » Aos quatro processos organizacionais foi acrescentado o processo de *Capacitação*, que se baseia na **capacidade**¹⁸ (a ser melhorada constantemente) dos recursos humanos disponíveis na empresa.

A estrutura da abordagem é apresentada na *Figura 7*, a seguir.

Figura 7: Estrutura da abordagem *GQS-AE*.

Referência: Adaptado de ISO 12207 (1995).

¹⁸ Em linhas gerais, o conceito assemelha-se àquele utilizado no Projeto SPICE (ISO TR 15504, 1998).

3.4 SOBRE AS ATIVIDADES DOS PROCESSOS

Cada processo é constituído por um conjunto de atividades a serem realizadas que, por sua vez, são compostas por tarefas. Uma tarefa é a menor unidade que colabora para o perfeito andamento dos processos dentro de uma empresa de software. Assim sendo, uma atividade pode ser entendida como uma entidade abstrata, caracterizada, apenas, por uma sequência de tarefas.

As atividades que compõem os processos foram concebidas com base nos documentos analisados (referentes ao CMM (PAULK *et al*, 1995), ao Projeto SPICE (ISO TR 15504, 1998) e às normas ISO (ISO 9001, 1994; ISO 9001, 2000; ISO 12207, 1995)). Muitos aspectos das normas e modelos de padronização e melhoria de processo foram considerados na elaboração da abordagem, não se perdendo de vista as restrições que apresenta uma empresa de pequeno porte. Para a concepção de tais atividades foi definida uma estrutura básica contendo as informações necessárias e relativas a cada atividade. Essa estrutura foi utilizada como *template* para descrever as atividades e está representada na *Figura 8*, que mostra o *template* preenchido para a primeira atividade do processo de **Definição**, a de *identificar as necessidades iniciais (Def.I)*.

Atividade:	identificar as necessidades iniciais (Def.I)
Objetivo:	definir o problema e a necessidade do cliente em adquirir, desenvolver ou aprimorar um sistema, produto ou serviço de software
Responsáveis:	gerente de projeto e analistas
Critério de inicialização:	o cliente apresentou um problema para o qual deseja uma solução computacional
Elementos de entrada:	descrição informal, tanto do problema quanto dos requisitos do sistema a ser desenvolvido (informação externa) (28)
Tarefas:	1. definir, mais formalmente, o problema 2. fazer um levantamento inicial dos requisitos do sistema, incluindo-se aspectos de segurança, confiabilidade e desempenho 3. determinar o escopo do sistema e as características do ambiente de operação 4. aprovar esse levantamento inicial – tarefa do gerente de projeto
Elementos de saída:	levantamento inicial dos requisitos (17)
Critério de finalização:	o levantamento foi estruturado em conformidade com a vontade do cliente, obtendo a aprovação do gerente de projeto
Métricas registradas:	<ul style="list-style-type: none">tempo gasto na identificação das necessidades iniciais

Figura 8: Template para definir as atividades que compõem os processos da abordagem GQS-AE

De acordo com a *Figura 8*, o campo *Atividade* contém o nome da atividade a ser desenvolvida, incluindo-se seu código de referência. O código indica o processo do qual ela

faz parte e sua ordem de execução dentro desse processo. O campo *Objetivo* apresenta a finalidade a que se destina a atividade. O campo *Responsável* indica, dentro dos papéis definidos na empresa, quais os responsáveis pela execução da atividade. O campo *Critério de Inicialização* representa o evento que dá início à realização da atividade. O próximo campo, *Elementos de Entrada*, indica quais são os produtos a serem utilizados na execução da atividade. Esses produtos podem ser externos ou provenientes de uma outra atividade (que, por sua vez, pode ser parte de um outro processo). O campo *Tarefas* apresenta uma lista das tarefas que, quando executadas, levam à realização da atividade proposta. O campo *Elementos de Saída* indica os produtos gerados quando executada a atividade. Tais produtos, por certo, serão utilizados por outras atividades, inclusive de outros processos. O campo *Critério de Finalização* apresenta o evento que caracteriza o fim da atividade, ou seja, o evento que permite constatar que a atividade foi concluída. O último campo, *Métricas*, sugere uma lista de métricas a serem utilizadas na avaliação quantitativa da atividade e, num nível mais geral, do andamento dos processos dentro da empresa¹⁹. Nos casos em que a informação requerida pelo campo não se aplica, ele foi preenchido com os dizeres *não identificado*.

Ao todo, são 50 atividades, apresentadas em 50 *templates*, a serem consultados no **Apêndice A** deste documento.

3.5 RELACIONAMENTO ENTRE OS PROCESSOS

Os processos relacionam-se funcionalmente. Os desenvolvedores de software desenvolvem um sistema, cuja documentação e configuração são definidas e coordenadas pelos processos de apoio. Ao iniciar um projeto, os processos organizacionais determinam a infra-estrutura necessária à realização das atividades. Isso implica em que a definição do processo é estática, porquanto sua dinâmica somente é determinada quando o processo é aplicado a um projeto de software. (MAIDANTCHIK, 1999)

¹⁹ Para se alcançar níveis cada vez mais altos de qualidade, é necessário melhorar cada passo do ciclo de desenvolvimento (OMAN, 1997). Logo, dados quantitativos, capazes de descrever a realidade do processo, precisam ser obtidos e devidamente analisados. É por isso que a realização de medições foi reconhecida como um pré-requisito indispensável para se introduzir a disciplina da engenharia ao desenvolvimento, manutenção e uso dos produtos de software (BASILI *et al*, 1994a).

Como foi observado na descrição do *template*, produtos intermediários são utilizados e produzidos ao longo da execução dos processos (os chamados elementos de entrada e saída). Eles são produzidos por uma dada atividade e utilizados por outra, por vezes, de processos diversos. São os produtos que “asseguram” a dinâmica dos processos. Para ilustrar a relação existente entre os produtos e os processos foi utilizada uma tabela que, na primeira coluna lista todos os *produtos* gerados ao longo do ciclo de desenvolvimento de software, e, nas demais, os *processos* envolvidos, decompostos em atividades. O cruzamento de um produto e uma atividade – representado por uma cela colorida – indica que existe uma relação entre ambos. Ou o produto origina-se da atividade (nos casos em que as celas são vermelhas), ou é utilizado por ela (quando se têm celas azuis). Há, ainda, o caso em que ocorre tanto uma quanto outra situações (o que é representado nas celas verdes). As atividades estão referenciadas pelo código que as define, como citado anteriormente, na explicação do *template*. Já os números dos produtos, na tabela, seguem aqueles estabelecidos nos campos *Elementos de Entrada* e *Elementos de Saída*, ainda nos *templates*, e colocados entre parênteses.

Uma parte dessa tabela pode ser vista na *Figura 9*²⁰, que destaca as atividades do processos de *Definição* e os produtos envolvidos nessas atividades.

Produtos	Def.I	Def.II
...		
(14) informações sobre os recursos humanos, técnicos e econômicos disponíveis na empresa		
(15) diretrizes de produção dos documentos necessários ao longo do projeto		
(16) documentos a serem armazenados/rearmazenados (provenientes de alguma atividade que os gerou)		
(17) levantamento inicial dos requisitos		
(18) relatório da viabilidade o projeto		
...		
...		
(28) descrição informal, tanto do problema quanto dos requisitos do sistema a ser desenvolvido (informação externa).		
...		

Figura 9: Relacionamento entre produtos e atividades dos processos.

Além disso, interligando-se os documentos já comentados – *templates* e a tabela, foi desenvolvida a representação gráfica da abordagem, ou seja, um diagrama de fluxo de

²⁰ A tabela completa encontra-se no **Apêndice B** deste documento.

trabalho, apresentado na *Figura 10*, em escala reduzida, de forma a apenas situar o leitor no contexto da representação²¹.

Figura 10: Representação gráfica da abordagem *GQS-AE*, em escala reduzida.

Ao se desenvolver tal representação, procurou-se mostrar o relacionamento entre as atividades dos processos envolvidos no desenvolvimento de software através de uma linha temporal de eventos e sob uma perspectiva global. O diagrama deve ser lido da esquerda para a direita. No eixo horizontal estão os instantes, crescentes, e no eixo vertical, os processos. Assim, fixado um momento no eixo horizontal, é possível saber as atividades que estão sendo executadas, bem como as já executadas e as que ainda não o foram, para cada processo de

²¹ A figura na escala original, que possibilita uma análise mais rica em detalhes, encontra-se no Apêndice C deste documento.

software. Os processos organizacionais, por não estarem ligados a um projeto, são executados continuamente.

Os círculos com a numeração romana indicam as atividades, de acordo com cada processo. As setas indicam o fluxo de informação entre as atividades. Uma linha continua significa um fluxo que sempre existe. Já uma linha pontilhada refere-se a um fluxo que existe em algumas situações apenas. Este é o caso, por exemplo, dos problemas que surgem em algum processo e não são resolvidos dentro do escopo de tal processo, devendo ser encaminhados ao processo de *Resolução de Problemas*. Como isso não ocorre sempre, tal fluxo deve ser representado por uma seta com linha pontilhada. A numeração arábica que acompanha cada seta diz respeito aos produtos contidos no fluxo e essa associação pode ser verificada na tabela do **Apêndice B**. Setas numeradas devem ser entendidas, então, como fluxos de dados. O produto pode ser gerado pela atividade, no caso de um fluxo de saída ou utilizado por ela, no caso de um fluxo de entrada. O fluxo que não se origina de uma atividade ou que a ela não é enviado representa um produto cuja origem ou o destino é o ambiente externo à organização. Foram utilizadas, ainda, cores diferentes para a representação das categorias de processos. Variações de azul representam os **processos primários**. Os **processos de apoio** foram representados utilizando-se as variações de verde. Por fim, aos **processos organizacionais** foram atribuídas as variações da cor vermelha.

Outras considerações sobre a representação encontram-se listadas a seguir.

- O Plano de Projeto de Software é composto por várias seções que podem ser utilizadas individualmente por atividades da abordagem. Dessa forma, cada seção é produzida por uma atividade específica, dentro do processo de *Planejamento*. Para a união dessas seções, gerando-se assim o Plano de Projeto de Software, foi criada uma atividade-fantasma (pois ela não existe na prática) composta por uma tarefa apenas: agrupar essas seções. Tal atividade é indicada pela sigla **A.F.** e está representada no diagrama por um círculo sem cor de preenchimento.
- Os balões à direita de uma atividade mostram que ela está intimamente ligada àquelas cuja referência (o código das atividades) é feita nesses balões. Duas situações podem ocorrer aqui: (i) a atividade representada pelo círculo numerado e aquela indicada no balão encontram-se na mesma coluna temporal. Nesse caso, a atividade mostrada no círculo **deve** ser realizada em conjunto com a listada no balão; (ii) a atividade representada pelo círculo numerado encontra-se num instante posterior ao da atividade indicada no balão. Nesse caso, a atividade mostrada no

círculo **deve** ser realizada em conformidade com o estabelecido pela atividade listada no balão.

- Existe uma estrutura retangular empregada para representar bases de dados. Para a abordagem desenvolvida, duas bases foram definidas. Uma delas, a “base de dados históricos” é usada para se armazenar documentos relacionados à empresa e/ou aos projetos (pelos processos de *Documentação* e *Gerenciamento de Configuração*) e para conseqüentes estudos empíricos (pelo processo de *Melhoria*). A outra, o “repositório de software”, é usado para o controle dos itens de software que já passaram por uma *linha de referência* (de acordo com o processo de *Gerenciamento de Configuração*).
- Uma outra estrutura, na forma retangular e de cor preta, foi empregada para indicar que a atividade dos círculos deve ser realizada seguindo-se o processo (representado por essa estrutura) como um todo. Nesse caso, não se pode precisar que atividade do processo deve ser seguida (se isso fosse possível, a representação seria feita por meio de balões). Usou-se tal artifício para cobrir as situações em que a atividade representada pelos círculos numerados é realizada, ora seguindo-se uma(s), ora seguindo-se outra(s) atividade(s) contida(s) no processo.

3.6 PAPÉIS NA ABORDAGEM

No desenvolvimento da abordagem, optou-se por considerar os profissionais como desempenhando um papel e não um cargo. Os papéis são designados de acordo com cada projeto. Assim, pode haver uma rotatividade dos profissionais, o que, por conseguinte, irá permitir a institucionalização de um processo. É imperativo ressaltar que, observadas as restrições quanto ao quadro de profissionais de uma empresa de pequeno porte, deverá haver um acúmulo de papéis por parte desses profissionais. Não raro, ter-se-á um profissional desempenhando certo papel em um projeto e um papel diferente em outro projeto. Pode, ainda, no mesmo projeto, desempenhar dois ou mais papéis. O importante é que se tenha em mente o alcance da função desempenhada, de maneira a não interferir no perfeito andamento dos processos. Os profissionais necessários, de acordo com a abordagem *QJ-AE*, podem ser associados aos seguintes papéis:

- **Gerente da empresa:** responsável pela organização. Para o bom andamento de todos os processos, é necessário que ele seja uma entidade onipresente e onisciente.

- **Gerente de projeto:** responsável por um projeto específico. Deve cuidar da execução dos processos envolvidos no projeto, bem como da distribuição das tarefas e da aprovação dos resultados.
- **Administrador dos recursos:** responsável pela gerência dos recursos humanos e computacionais disponíveis na empresa e, mais especificamente, para um dado projeto.
- **Analista:** responsável pela especificação do sistema, pela determinação dos requisitos e pela validação do software.
- **Projetista:** responsável pelo projeto e pela coordenação da implementação do software.
- **Programador:** responsável pela implementação do software, cuidando das atividades de codificação e teste do sistema.
- **Operador:** responsável pela assistência aos usuários, durante a operação do sistema.
- **Mantenedor:** Responsável pela manutenção (correção de falhas e implementação de melhorias) do sistema.
- **Gerente de versões:** responsável pelo gerenciamento da configuração, tanto do sistema quanto dos componentes de software.
- **Coordenador de times²²:** responsável por coordenar um time que desempenha determinada atividade. Esse papel não é obrigatório, principalmente nos casos em que os times são compostos por um número bastante reduzido de profissionais.

A *Figura 11* apresenta uma tabela em que são mostrados os relacionamentos existentes entre os papéis definidos e cada uma das atividades da abordagem. As celas de cor azul indicam que o profissional é um dos responsáveis pela atividade. Por outro lado, uma cela preenchida com a cor amarela indica que esse profissional apenas colabora na execução da atividade, não sendo responsável por ela (esse é o caso das atividades que se encontram destacadas nos balões).

²² Um *time* supõe um nível maior de colaboração entre seus membros do que aquele encontrado entre os membros de uma *equipe*.

3.7 OS PROCESSOS PRIMÁRIOS

Os processos primários definem as atividades indispensáveis ao desenvolvimento de software. Estão organizadas nos processos de *Definição*, *Planejamento*, *Desenvolvimento*, *Operação e Manutenção*. As atividades que asseguram a realização dos processos primários são determinadas pelos processos organizacionais.

3.7.1 DEFINIÇÃO

O processo de *Definição* inicia com a identificação das necessidades iniciais. Aqui, procura-se obter uma visão preliminar das necessidades do cliente, as quais deverão ser atendidas pelo software ou sistema que se vai construir, com base numa definição informal, tanto do problema quanto dos requisitos do sistema a ser desenvolvido, fornecida pelo próprio cliente. A seguir, deve-se fazer um levantamento sobre a viabilidade do projeto. Informações sobre os recursos humanos, técnicos e econômicos disponíveis na empresa, a serem consultadas nesse levantamento, são fornecidas pelo processo de *Capacitação*. Métricas envolvidas no processo de *Definição* incluem o tempo gasto na identificação das necessidades iniciais do cliente e a porcentagem dos recursos necessários ao projeto e que estão disponíveis na empresa. As atividades que constituem o processo de *Definição* são:

Def.I – identificar as necessidades iniciais (pág.96). Com essa atividade procura-se definir, mais formalmente, o problema, fazendo-se um levantamento inicial dos requisitos do sistema, o qual deve incluir aspectos de segurança, confiabilidade e desempenho. O escopo do sistema é determinado e as características do ambiente de operação, identificadas. Essas tarefas são de responsabilidade dos analistas alocados para o projeto, podendo ser realizadas em conjunto com o gerente de projeto. A este cabe, de maneira exclusiva, aprovar o levantamento inicial desenvolvido.

Def.II – avaliar a viabilidade do projeto (pág.96). O gerente de projeto deve identificar, dentre os recursos disponíveis, aqueles necessários ao projeto, analisando os riscos, custos e benefícios de possíveis opções de desenvolvimento, de acordo com os recursos disponíveis. Deve, ainda, celebrar o contrato com o cliente, expressando claramente as expectativas e as responsabilidades do cliente e da organização. Todos os resultados devem ser

documentados²³, seguindo-se o estabelecido no processo de *Documentação*. Eventuais problemas e/ou não-conformidades que não puderem ser resolvidos no escopo do processo de *Definição* devem ser encaminhados ao processo de *Resolução de Problemas*²⁴.

3.7.2 PLANEJAMENTO

A primeira atividade a ser realizada na execução do processo de *Planejamento* é a de preparação, cujo objetivo é estabelecer a estrutura adequada à realização das demais atividades. Segue-se um refinamento do levantamento inicial feito pelo processo de *Definição*, com vistas ao desenvolvimento do Plano de Projeto de Software. O estabelecimento do modelo de desenvolvimento de software adequado ao projeto (Modelo Cascata, Espiral, Prototipação, Evolutivos, dentre outros), a identificação dos riscos do projeto e a preparação das estimativas para o projeto são as próximas atividades. Por fim, devem-se planejar as atividades dos processos de apoio a serem empregadas no projeto. Métricas envolvidas no processo de *Planejamento* incluem o tempo gasto em cada tarefa de preparação, o tempo gasto no refinamento do levantamento inicial e o número de consultas ao cliente, até a completa aprovação da especificação preliminar dos requisitos. Todos os resultados do processo de *Planejamento* devem ser documentados, seguindo-se o estabelecido no processo de *Documentação*. As atividades que constituem o processo de *Planejamento* podem ser assim divididas:

Plan.I – preparar para o processo de *Planejamento*²⁵(pág.97). É tarefa do gerente de projeto determinar a metodologia de trabalho a ser seguida, identificando ferramentas de apoio às tarefas de planejamento, de acordo com o processo de *Infra-Estrutura*. Ele também identifica, em conformidade com o estabelecido no processo de *Capacitação*, a necessidade de treinamento para as atividades de *Planejamento*, preparando as diretrizes para o mesmo, se

²³ Sempre que um relatório, parecer ou avaliação **documentada** forem gerados, recomenda-se que todos os profissionais envolvidos com tais documentos (as partes diretamente implicadas) tenham ciência formal de seu conteúdo. Alternativamente, nos casos em que existe uma coordenação de time, basta que o coordenador tome o conhecimento formal do documento e cuide de repassar as informações aos membros do time.

²⁴ Os profissionais alocados para os processos que não estão ligados ao processo de *Resolução de Problemas* devem ser capazes de solucionar os problemas que ocorrerem, única e exclusivamente, dentro do escopo do processo em que se inserem tais problemas. Isso é necessário para se minimizar o tempo despendido com comunicação desnecessária ou supérflua entre os profissionais de processos diversos. Para assegurar que isso seja levado a efeito, atividades específicas foram inseridas nos processos de *Capacitação* e de *Treinamento*.

²⁵ A maioria dos processos requer, como primeira atividade a ser executada, uma atividade de preparação. Nesses casos, é forçosamente necessário que isso seja seguido, para se evitar que o processo seja executado de maneira desordenada, descontrolada ou caótica. Isso, certamente, frustraria os objetivos de garantia e de melhoria da qualidade do software produzido.

ele for necessário. Essas diretrizes deverão ser encaminhadas ao processo de *Treinamento*. Como definido no processo de *Infra-Estrutura*, fica a cargo do gerente de projeto, em conjunto com o administrador dos recursos, a disponibilização da “base de dados históricos” para uso no projeto. Essa “base” deverá conter informações relevantes sobre o projeto, possibilitando a identificação e a padronização de um glossário sobre o domínio da aplicação e sobre esse projeto, capaz de garantir a compreensão única dos termos pelos vários times. Dados históricos para melhoria de processo também devem ser armazenados na “base”, seguindo-se as atividades correlatas estabelecidas. O armazenamento/rearmazenamento ou exclusão de documentos da “base de dados históricos” fica a cargo do processo de *Documentação*.

Plan.II – refinar o levantamento inicial (pág.97). Primeiramente, o gerente de projeto deve alocar as atividades do processo de *Planejamento* aos profissionais, de acordo com os resultados dos processos de *Capacitação* e *Treinamento*. Feito isso, aquele profissional, em conjunto com os analistas, analisam o problema a ser tratado e os requisitos iniciais captados pelo processo de *Definição*. As características, os usuários e as interfaces do sistema devem ser definidos. Através do método de desenvolvimento adequado, faz-se a modelagem inicial do sistema, avaliando-se a consistência e a conformidade com o levantamento inicial. Os resultados são documentados e o gerente de projeto deve aprovar tais resultados.

Plan.III – estabelecer o modelo de desenvolvimento de software adequado ao projeto (pág.98). Considerando-se as necessidades do cliente, um modelo de desenvolvimento de software adequado é definido. Para isso, determinam-se os componentes do sistema. Tais componentes devem ser classificados de acordo com o conhecimento e recursos técnicos necessários para sua construção. Com isso, identifica-se o modelo de desenvolvimento apropriado. Mais uma vez, cabe ao gerente de projeto aprovar os resultados.

Plan.IV – identificar os riscos do projeto (pág.98). Com base no levantamento inicial dos requisitos e no relatório da viabilidade do projeto, provenientes do processo de *Definição*, o gerente de projeto e os analistas identificam os riscos envolvidos no projeto, fazendo uma estimativa destes e definindo opções para que sejam evitados. Procedimentos para a monitoração desses riscos também necessitam ser definidos. Cabe ao gerente de projeto aprovar os resultados obtidos.

Plan.V – preparar as estimativas para o projeto (pág.99). Mais uma vez, analistas e gerente de projeto reúnem-se para elaborar as estimativas do projeto, tanto de prazo quanto de custo e esforço. Primeiramente, um método adequado de estimativas deve ser definido. A seguir, o grupo elabora as estimativas necessárias, documentando-as. O gerente de projeto, então, aprova os resultados.

Plan.VI – realizar o planejamento das atividades de apoio ao desenvolvimento do projeto (pág.99). Inicialmente, um levantamento dos produtos de software necessários para o estabelecimento e manutenção do controle do projeto é feito. Devem ser definidos os pontos de revisão e os dados a serem coletados em tais pontos. Essas duas tarefas são realizadas em conjunto com o processo de *Acompanhamento*. Com base no estabelecido pelo processo de *Controle de Qualidade*, o planejamento da qualidade deve, também, ser realizado aqui. Cabe, ainda, aos analistas, planejar a operação e a manutenção do sistema. Todas as tarefas desta atividade são realizadas tendo-se, em mãos, a especificação preliminar dos requisitos. Os resultados devem ser aprovados pelo gerente de projeto.

3.7.3. DESENVOLVIMENTO

Este processo determina as atividades dos desenvolvedores do software relativas à especificação, projeto, codificação, integração, testes, instalação e entrega, de acordo com o que foi determinado no processo de *Planejamento*. As atividades dos processos de apoio são realizadas na medida em que forem necessárias. Portanto, em cada atividade de *Desenvolvimento*, o desenvolvedor deve: produzir a documentação do software, como definido pelo processo de *Documentação*; controlar as mudanças e administrar as versões do sistema, de acordo com o processo de *Gerência da Configuração*; notificar os problemas encontrados e solucioná-los, fazendo uso, nos casos necessários, do processo de *Resolução de Problemas*; efetuar as verificações e validações necessárias, bem como o controle da qualidade, seguindo-se os processos de *Verificação*, *Validação* e *Controle de Qualidade*. Cabe ressaltar, mais uma vez, que a abordagem é independente do modelo de desenvolvimento adotado na empresa. Dessa forma, as atividades do processo de *Desenvolvimento* foram definidas considerando-se as fases genéricas de produção de um software: definição, desenvolvimento (projeto e implementação) e manutenção. Métricas envolvidas no processo de *Desenvolvimento* incluem o número total de componentes do projeto, a densidade de comentários, *fan-in* e *fan-out* dos módulos, o número de *drivers* e *stubs* utilizados nos testes, bem como o número de falhas encontradas, o tempo necessário para a instalação do sistema e o número de problemas encontrados nessa instalação. As atividades que constituem o processo de *Desenvolvimento* podem ser assim descritas:

Desenv.I – iniciar o processo de *Desenvolvimento* (pág.100). O gerente de projeto, juntamente com analistas e projetistas, deve preparar o plano para a condução das atividades de desenvolvimento, de acordo com as diretrizes estabelecidas para os requisitos de software

e com base em informações provenientes do processo de *Capacitação*. O plano deve incluir, sobretudo, um cronograma para a realização das atividades. O gerente de projeto identifica ferramentas de apoio às tarefas de desenvolvimento, com base no processo de *Infra-Estrutura*. Ele ainda deve identificar, em conformidade com o estabelecido no processo de *Capacitação*, a necessidade de treinamento para as atividades do processo de *Desenvolvimento*, preparando as diretrizes para o mesmo, se ele for necessário. Essas diretrizes deverão ser encaminhadas ao processo de *Treinamento*.

Desenv.II – especificar os requisitos (pág.100). Esta atividade deve ser realizada em conformidade com o estabelecido no processo de *Gerenciamento de Configuração*. O objetivo é definir, de forma mais completa e menos ambígua possível, os requisitos do sistema, funcionais e não-funcionais. Devem, inicialmente, ser alocadas, aos profissionais, as atividades do processo de Desenvolvimento, com base nos resultados dos processos de *Capacitação* e *Treinamento*. As demais tarefas desta atividade compreendem: estabelecer os requisitos do projeto e do sistema; identificar os requisitos do sistema que devem ser alocados ao projeto, bem como as restrições de projeto; estabelecer, analisar e refinar os requisitos do software (descrição funcional, de controle e comportamental do sistema); estabelecer critérios de validação, em conjunto com o processo de *Validação*; explicitar os requisitos de qualidade, de acordo com o processo de *Controle de Qualidade*. É necessário, por fim, que os resultados sejam aprovados pelo gerente de projeto.

Desenv.III – desenvolver o projeto de software (pág.101). Em conformidade com o estabelecido no processo de *Gerência da Configuração*, os projetistas devem definir a estrutura de cada componente, identificando os itens de hardware e software necessários, de maneira a possibilitar reúso. Definido o projeto arquitetural, é necessário que ele esteja adequado à Especificação de Requisitos, fazendo-se uso do estabelecido no processo de *Verificação*. Os componentes individuais de software devem, então, ser refinados até o nível de interfaces, classes ou componentes pré-existentis. Aos projetistas cabe ainda definir bases de dados, linguagens de programação, padrões de integração, estilos de interface e requisitos de testes de integração e de unidade. Para que essas tarefas sejam realizadas, é necessário que estejam disponíveis a Especificação de Requisitos e o Plano de Projeto de Software. Mais uma vez, a aprovação do gerente de projeto faz-se necessária. O resultado desta atividade é o projeto arquitetural e o projeto detalhado do software.

Desenv.IV – implementar o sistema (pág.101). Aqui os projetistas e programadores têm a tarefa de transformar os componentes do sistema em elementos executáveis. Para tanto, em conformidade com o estabelecido pelo processo de *Gerência da Configuração*, devem desenvolver o código para cada componente projetado, documentando interna e externamente

esse código, seguindo-se os padrões de documentação definidos pelo processo de *Documentação*. A elaboração do Manual do Usuário é uma tarefa que tem início com esta atividade.

Desenv.V – realizar os testes individuais (pág.102). O objetivo principal desta atividade é o de detectar e corrigir falhas nos módulos e nos procedimentos, de modo a verificar se os requisitos especificados para o software estão sendo corretamente implementados. Os programadores e projetistas devem, em conformidade com o estabelecido no processo de *Gerenciamento de Configuração*, estabelecer, primeiramente, os planos e procedimentos, casos e critérios de teste. Feito isso, cada componente deve ser testado, corrigindo-se as falhas evidenciadas pelos testes individuais. Os problemas e/ou não-conformidades que não puderem ser resolvidos no escopo do processo de *Desenvolvimento*, relacionados apenas aos testes individuais, devem ser encaminhados, respeitando-se a formalidade exigida, ao processo de *Resolução de Problemas*. Ao final desta atividade, obtém-se o código-fonte do sistema, com os testes de unidade devidamente realizados e pronto para que sejam realizados os testes de integração.

Desenv.VI – integrar e testar o sistema (pág.102). Outra vez, projetistas e programadores devem, respeitando-se o processo de *Gerência da Configuração*: elaborar um plano para integrar os componentes de software, definindo casos e critérios de teste, bem como os procedimentos necessários; integrar e verificar o comportamento do software integrado; corrigir falhas evidenciadas pelos testes de integração; finalizar o Manual do Usuário; validar o sistema, com base no que foi definido pelo processo de *Validação* e gerar o código executável do sistema. Os problemas e/ou não-conformidades que não puderem ser resolvidos no escopo do processo de *Desenvolvimento*, relacionados aos testes de integração e à validação do sistema, devem ser encaminhados, respeitando-se a formalidade exigida, ao processo de *Resolução de Problemas*. Futuras melhorias, para futuras versões podem ser propostas pelos profissionais. Esse relatório de melhorias deverá ser encaminhado ao processo de *Manutenção*.

Desenv.VII – instalar e entregar o sistema (pág.103). É tarefa dos operados preparar o sistema para ser utilizado pelos usuários, no ambiente de operação. Para tanto, um plano de instalação deve ser confeccionado. A documentação necessária para a instalação e a utilização do sistema, sobretudo o Manual do Usuário, deve estar disponível. Os operados ainda introduzem os dados iniciais necessários à execução do sistema. Todos os resultados devem ter a aprovação do gerente de projeto.

3.7.4 OPERAÇÃO

O processo de operação contém as atividades dos operados, conforme estabelecidas no plano de operação e suporte. Refere-se à operação do produto de software e ao suporte operacional aos usuários. Métricas envolvidas no processo de *Operação* incluem o número de testes operacionais realizados até a liberação do sistema para uso, o intervalo médio de tempo entre os testes operacionais, o número de visitas ao ambiente de operação para assistência ou suporte e a evolução do nível de satisfação do usuário. Todos os resultados do processo de *Operação* devem ser documentados, seguindo-se o estabelecido no processo de *Documentação*. As atividades que constituem o processo de *Operação* são:

Oper.I – preparar para o processo de *Operação* (pág.103). Primeiramente, as atividades do processo de *Operação* devem ser alocadas aos profissionais, com base nos resultados dos processos de *Capacitação* e *Treinamento*. Feito isso, identificam-se ferramentas de apoio às atividades de operação, de acordo com o processo de *Infra-Estrutura*, e os operadores desenvolvem um plano para a condução das atividades relacionadas à operação e ao suporte aos usuários. Devem ser estabelecidos, também, critérios para o teste operacional, para inserir pedidos de modificação no processo de *Manutenção* e para liberar o produto para uso operacional. É necessária a aprovação do gerente de projeto, o qual também é responsável por identificar, em conformidade com os processos de *Capacitação*, a necessidade de treinamento para as atividades do processo de *Operação*, preparando as diretrizes para o mesmo, se ele for necessário.

Oper.II – cuidar dos testes operacionais e da operação do sistema (pág.104). As tarefas que compõem esta atividade são de responsabilidade dos operadores, que devem seguir o plano de operação e suporte desenvolvido. Primeiramente, o gerente de projeto deve alocar as atividades envolvidas no processo de *Operação* aos profissionais, com base nos resultados dos processos de *Capacitação* e *Treinamento*. Definidas as responsabilidades, os operadores executam os testes operacionais, respeitando-se o disposto no Manual do Usuário; avaliam os resultados e, tendo sido, os testes, satisfatórios, o sistema é liberado para uso. Essas tarefas são realizadas juntamente com o processo de *Validação*. Na companhia do usuário, os operadores devem operar o sistema no ambiente para o qual este foi desenvolvido, fornecendo o auxílio necessário nas atividades iniciais de operação. Os problemas e/ou não-conformidades que não puderem ser resolvidos no escopo do processo de *Operação* devem ser encaminhados ao processo de *Resolução de Problemas*.

Oper.III – fornecer assistência e/ou suporte aos usuários (pág.104). Nos casos em que um pedido (informal) de assistência e/ou suporte é realizado, operadores e mantenedores

devem, primeiramente, formalizar esse pedido, seguindo-se o estabelecido no processo de *Documentação*. Fornecem assistência, suporte e consultoria aos usuários, documentando suas dúvidas e dificuldades, e solucionando os problemas. Nos casos necessários, pedidos de manutenção ou possíveis melhorias são encaminhados ao processo de *Manutenção*. Em qualquer dos casos, retorno ao solicitante sempre é necessário, monitorando-se a execução dos pedidos de manutenção. Os instrumentos de avaliação do nível de satisfação dos usuários (com relação ao sistema e aos serviços) definidos no processo de *Melhoria* devem ser aplicados. Todos os resultados devem ser aprovados pelo gerente de projeto.

3.7.5 MANUTENÇÃO

O processo de *Manutenção* contém as atividades dos mantenedores, de acordo com o que foi estabelecido no plano e nos procedimentos de manutenção. A fase de manutenção consiste na correção de erros e implementação de melhorias no software, assegurando-se sua integridade. As mudanças a serem realizadas podem ser de três tipos: (i) de caráter corretivo (manutenção efetuada em resposta às falhas de implementação), (ii) de caráter preventivo (manutenção efetuada em resposta às alterações no ambiente de dados e/ou no ambiente de processamento) e (iii) de caráter perfectivo (manutenção efetuada por conta da inclusão de novas capacidades, modificações em funções existentes ou ampliações gerais). A manutenção perfectiva é a responsável pela maior parte de todo o esforço despendido em manutenção de software (PRESSMAN, 2002). De qualquer forma, independentemente do tipo das mudanças, elas devem ser sempre documentadas, de forma que os documentos gerados durante o processo de *Desenvolvimento* sejam atualizados. O processo de *Manutenção* é iniciado quando o software é modificado e termina quando este não mais é utilizado. Métricas envolvidas no processo incluem a razão entre o número de pedidos e o tipo de manutenção a ser efetuada, o número de alterações efetuadas no sistema, o tempo médio de implementação de uma modificação, assim como seu custo envolvido. Todos os resultados do processo de *Manutenção* devem ser documentados, seguindo-se o estabelecido no processo de *Documentação*. As atividades envolvidas na manutenção de um sistema podem ser assim divididas:

Manut.I – preparar para o processo de *Manutenção* (pág.105). Devem ser identificadas ferramentas de apoio às tarefas de manutenção, de acordo com o processo de *Infra-Estrutura*. Depois disso, os mantenedores desenvolvem planos e procedimentos para conduzir as atividades e tarefas de manutenção, incluindo-se os casos de migração e

descontinuação, com base no estabelecido no processo de *Planejamento*. Procedimentos para receber, armazenar e acompanhar os pedidos de modificação e os relatórios de problemas dos usuários são estabelecidos. Os planos e todos os procedimentos necessitam da aprovação do gerente de projeto, que também deve identificar, em conformidade com o estabelecido no processo de *Capacitação*, a necessidade de treinamento para as atividades do processo de *Manutenção*, preparando as diretrizes para o mesmo, se ele for necessário.

Manut.II – fazer a análise do problema (pág.105). O pedido de modificação deve, aqui, ser avaliado, considerando-se o tipo de manutenção a ser realizada, o alcance da alteração e as conseqüências de tal mudança. Entretanto, antes disso, é necessário que as atividades do processo de *Manutenção* sejam alocadas aos profissionais, com base nos resultados dos processos de *Capacitação* e *Treinamento*. Opções para a implementação da modificação devem ser desenvolvidas. O gerente de projeto tem a tarefa de decidir, com base na análise realizada, sobre o pedido de alteração. Nos casos em que esse pedido receber a aprovação, deve-se informar a prioridade de execução da alteração e emitir uma “ordem de serviço”.

Manut.III – implementar a modificação (pág.106). Os mantenedores determinam e documentam quais unidades de software, versões e documentação relacionada precisam ser modificadas. Seguindo-se os processos de *Desenvolvimento* e *Operação*, as modificações são implementadas. Testes de regressão nas partes já existentes são necessários, para revalidar o sistema, de acordo com o processo de *Validação*. Nos casos de migração e/ou descontinuação do software, elas devem ser executadas de acordo com o que foi estabelecido nos planos e procedimentos para manutenção. Em qualquer dos casos, recomenda-se que o cliente seja notificado. Ao gerente de projeto cabe a aprovação de todos os resultados.

3.8 OS PROCESSOS DE APOIO

Os processos de apoio definem as atividades de suporte aos outros processos do ciclo de vida do software. Compreendem os processos de *Documentação*, *Acompanhamento*, *Gerenciamento de Configuração*, *Controle de Qualidade*, *Verificação*, *Validação* e *Resolução de Problemas*. As atividades e tarefas dos processos de apoio são de responsabilidade da organização de software (processos organizacionais), que deve estabelecer os pré-requisitos necessários para sua existência, garantindo sua funcionalidade.

3.8.1 DOCUMENTAÇÃO

O processo de *Documentação* reúne as atividades de registro de informações produzidas ao longo do ciclo de desenvolvimento do software. Este processo contém um conjunto de atividades que planejam, projetam e mantêm os documentos necessários para todos os profissionais do software, tais como gerentes, programadores e usuários. Métricas envolvidas no processo incluem o número de documentos identificados e a profundidade de documentação do projeto e de cada processo. As atividades que constituem o processo de *Documentação* são:

Doc.I – preparar para o processo de *Documentação* (pág.106). Analistas e projetistas, sob a supervisão do gerente de projeto, devem identificar os requisitos de documentação do projeto, em conjunto com o processo de *Definição*. Identificados os requisitos, com base no conteúdo técnico, são identificados os documentos a serem produzidos. Estabelece-se sua forma de organização, sobretudo quanto à hierarquia, e de identificação. Os resultados devem ser aprovados pelo gerente de projeto, e compilados em diretrizes para produção dos documentos necessários ao longo do projeto. É recomendável que o gerente de projeto também cuide para que todos os demais processos da empresa tenham conhecimento dessas diretrizes e possam utilizá-las.

Doc.II – gerenciar a produção dos documentos (pág.107). Pretende-se, com esta atividade, gerenciar a produção e cuidar do armazenamento dos documentos a serem produzidos e utilizados ao longo do projeto. Para isso, analistas e projetistas, ainda sob a supervisão do gerente de projeto, devem identificar os documentos a serem armazenados na “base de dados históricos”, estabelecendo procedimentos para seu armazenamento. Os documentos a serem armazenados/rearmazenados (provenientes de alguma atividade que os gerou) são, portanto, colocados na “base de dados históricos” ou eliminados dela, de acordo com o processo de *Verificação* e seguindo-se os procedimentos definidos. Nesta atividade também deve ser identificado o padrão de documentação interna e externa do código-fonte, de acordo com a linguagem de programação a ser executada.

3.8.2 ACOMPANHAMENTO

O objetivo do processo de *Acompanhamento* é fornecer a visibilidade adequada do progresso do projeto de software, de forma que se possam tomar medidas efetivas quando

esse desenvolvimento desvia-se significativamente do que estabelecem os planos. Métricas envolvidas no processo incluem a porcentagem de tempo gasto em cada tarefa de preparação, a porcentagem de desvio nas estimativas elaboradas e a evolução dos riscos. Todos os resultados devem ser documentados seguindo-se o estabelecido pelo processo de *Documentação*. As atividades que constituem o processo de *Acompanhamento* podem ser assim divididas:

Acomp.I – preparar para o processo de Acompanhamento (pág.107). O gerente de projeto, em conjunto com os projetistas, elabora um plano de acompanhamento que identifique itens a serem revistos, pontos de revisão e recursos necessários para se realizar as revisões, sobretudo os técnicos. Devem ser definidos procedimentos, associados aos produtos e às atividades, para o estabelecimento e manutenção do controle do projeto. Ferramentas de apoio às tarefas de acompanhamento devem ser identificadas, de acordo com o processo de *Infra-Estrutura*. Ao gerente de projeto cabe, então, aprovar os resultados.

Acomp.II – acompanhar o projeto de software (pág.108). Seguindo-se as atividades do processo de *Desenvolvimento* e com base no Plano de Projeto de Software, devem ser coletadas as métricas necessárias, em conjunto com o processo de *Controle de Qualidade*. São necessárias revisões gerenciais (que avaliam os custos e riscos envolvidos no projeto, os recursos e esforços necessários ao seu desenvolvimento, e o seu cronograma) e revisões técnicas (que avaliam os produtos e serviços, a fim de verificar se estão completos, respeitam os padrões e especificações e estão prontos para se poder executar a próxima atividade para a qual servem como entrada). Estas últimas devem ser conduzidas com o auxílio do processo de *Controle de Qualidade*. Atitudes corretivas devem ser levadas a efeito, quando necessário e possível, alterando a forma como o trabalho está sendo feito, sem prejuízo para o projeto e para o curso de qualquer outro processo, e desde que com a ciência do processo de *Gerência*.

3.8.3 GERENCIAMENTO DE CONFIGURAÇÃO

Este processo está relacionado aos procedimentos administrativos e técnicos para identificar os itens de software, controlar modificações e versões dos mesmos, registrar o estado de cada um, garantindo a consistência e a correção dos itens e controlando seu armazenamento, manipulação e entrega. Métricas envolvidas no processo incluem a razão entre o número de itens colocados sob controle e o número de itens possíveis, o número de pedidos de alteração de configuração aprovados e o número total de alterações da nova versão do sistema. Todos os resultados devem ser documentados seguindo-se o estabelecido pelo

processo de *Documentação*. As atividades que constituem o processo de *Gerenciamento de Configuração* podem ser assim descritas:

GConfig.I – preparar para o processo de Gerenciamento de Configuração (pág.108).

Determina-se que ferramentas, padrões e procedimentos serão usados, sobretudo quanto ao acesso e ao funcionamento do repositório de software, de acordo com o processo de *Infra-Estrutura*. Caso haja necessidade de treinamento em *Gerência da Configuração*, as diretrizes para esse treinamento são formuladas e devem ser encaminhadas ao processo de *Treinamento*. Os resultados são compilados num plano de gerência da configuração, a ser aprovado pelo gerente de projeto.

GConfig.II – selecionar os itens de configuração a serem controlados (pág.109).

A primeira tarefa desta atividade é alocar as atividades do processo aos profissionais, com base no estabelecido pelos processos de *Capacitação e Treinamento*. Feito isso, seguindo-se o processo de *Desenvolvimento*, definem-se os itens de configuração relevantes ao projeto, que serão efetivamente colocados sob controle de configuração. As relações entre esses itens devem ser explicitadas. É necessário, ainda, definir uma forma (ferramenta, por exemplo) para identificar, unicamente, cada item de configuração. Estabelecem-se as *linhas de referência (baselines)*. Cada item de configuração deve estar documentado (de acordo com o estabelecido pelo processo de *Documentação*), descrevendo-se a maneira como eles são arquivados e recuperados do repositório de software. Por fim, é necessária a aprovação do gerente de projeto.

GConfig.III – aprovar/rejeitar um pedido de alteração (pág.109). Esta atividade é realizada quando existe um pedido de alteração de um item que já passou por uma *linha de referência*, devidamente justificado, proveniente de alguma atividade do processo de *Desenvolvimento*. Assim, é necessário que o gerente de projeto avalie o pedido e, com base na avaliação, tome a decisão de aprová-lo ou rejeitá-lo. Em caso positivo, ele deve informar a prioridade de execução do pedido, emitindo a “ordem de serviço” associada. Eventuais problemas e/ou não-conformidades que não puderem ser resolvidos no escopo do processo de *Gerenciamento de Configuração* devem ser encaminhados ao processo de *Resolução de Problemas*.

GConfig.IV – implementar as alterações (pág.110). De acordo com a “ordem de serviço”, o gerente de versões retira do repositório de software os itens de configuração necessários. Tais itens são colocados na área de trabalho dos projetistas/programadores. Foi, então, efetuado o que se chama de *check out*. Os projetistas/programadores efetuam as alterações necessárias, seguindo-se o processo de *Desenvolvimento*. O aspecto funcional de cada item modificado é avaliado, na tentativa de se descobrirem omissões ou erros de

configuração que degradem os padrões de construção do software. Essa avaliação ainda verifica se as versões respeitam os requisitos definidos anteriormente e deve ser conduzida com base no estabelecido pelo processo de *Controle de Qualidade*. Uma verificação para se assegurar que a nova configuração a ser ‘congelada’ pela *linha de referência* está composta das versões mais recente dos itens de configuração é a próxima tarefa. É necessário verificar, ainda, se o projeto e a documentação estão atualizados, respeitando-se os procedimentos e padrões definidos, de acordo com o processo de *Verificação*.

GConfig.V – relatar a situação da configuração (pág.110). Pretende-se, com isso, manter a ordem e a unidade do projeto sendo desenvolvido. Os itens são armazenados no repositório de software, visto que foram aprovados na revisão. É o que se chama de *check in*. O gerente de versões documenta todas as alterações ocorridas, tais como o que aconteceu, quem o fez, quando ocorreu, o que mais será afetado, etc, confeccionando um relatório da situação atual da configuração do sistema.

3.8.4 CONTROLE DE QUALIDADE

O processo de *Controle de Qualidade* reúne atividades para garantir que os produtos e os processos de software (primários e de apoio) respeitam os requisitos e são coerentes com os respectivos planos, utilizando resultados de outros processos de apoio. Métricas envolvidas neste processo incluem a porcentagem de tempo gasto em cada tarefa de preparação, o número total de problemas e/ou não-conformidades encontrados, resolvidos no escopo do processo ou enviados ao processo de *Resolução de Problemas*, etc. Todos os resultados devem ser documentados, seguindo-se o estabelecido pelo processo de *Documentação*. As atividades que constituem o processo de *Controle de Qualidade* podem ser assim descritas:

CQual.I – preparar para o processo de Controle de Qualidade (pág.111). Primeiramente, os analistas, em conjunto com o gerente de projeto, devem identificar as relações existentes entre o processo de *Controle de Qualidade* e os processos de *Verificação* e *Validação*. Os requisitos e padrões de qualidade devem ser definidos, bem como os procedimentos para identificação, coleta, arquivamento, manutenção e disponibilização de tais requisitos. Ferramentas para auxiliarem nas tarefas de controle da qualidade devem ser identificadas, com base no processo de *Infra-Estrutura*. Os resultados são compilados num plano de controle da qualidade. Por fim, o gerente de projeto deve aprovar o plano, que servirá de base para as demais atividades de controle da qualidade dos produtos e dos processos envolvidos no desenvolvimento do software.

CQual.II – controlar a qualidade dos produtos (pág.111). Com base nos resultados da verificação dos produtos, esta atividade deve garantir que todos os planos exigidos no projeto estejam documentados, sejam mutuamente consistentes e executados quando exigidos. Deve assegurar, ainda, que os produtos de software – sobretudo o produto final – e a respectiva documentação seguiram os planos e satisfazem os requisitos especificados. Por fim, deve garantir que as medições dos produtos de software estejam de acordo com os padrões e procedimentos estabelecidos, com base no estabelecido pelo processo de *Acompanhamento*. Relatórios periódicos dos resultados do controle de qualidade necessitam ser confeccionados. O controle da qualidade dos produtos é feito pelo gerente de projeto, analistas, projetistas, programadores, operadores, mantenedores e gerentes de versões, na medida em que os produtos são gerados. Eventuais problemas e/ou não-conformidades que não puderem ser resolvidas no escopo do processo de *Controle de Qualidade* devem ser encaminhadas ao processo de *Resolução de Problemas*.

CQual.III – controlar a qualidade dos processos (pág.112). Com base nos resultados da verificação dos processos, esta atividade deve garantir que todos os processos de software (primários e de apoio) tenham seguido os planos e respeitado os padrões. Deve assegurar, ainda, que todas as práticas de Engenharia de Software, o ambiente de desenvolvimento e o ambiente de testes estejam em conformidade com os planos estabelecidos. É necessário também garantir que as medições dos processos de software estejam de acordo com os padrões e procedimentos definidos, seguindo-se o processo de *Acompanhamento*. Além disso, faz-se necessário garantir que os times tenham recebido treinamento adequado para a realização das atividades, se necessário, e que tenham habilidade e conhecimento suficientes para a realização das tarefas. Isso deve ser feito com base nos processos de *Capacitação e Treinamento*. O controle da qualidade dos processos também é feito pelo gerente de projeto, analistas, projetistas, programadores, operadores, mantenedores e gerentes de versões, na medida em que os processos são executados. Entretanto, competem ao administrador dos recursos as tarefas de garantia de qualidade associadas aos times de projeto. Eventuais problemas e/ou não-conformidades que não puderem ser resolvidas no escopo do processo de *Controle de Qualidade* devem ser encaminhadas ao processo de *Resolução de Problemas*.

3.8.5 VERIFICAÇÃO

O processo de *Verificação* determina se os produtos e as atividades (incluindo-se tarefas) de um processo respeitam os requisitos ou condições impostas para esse processo. É

necessário, pois, que sejam executadas análises, revisões e auditorias. Métricas que envolvem as atividades do processo incluem o tempo gasto em cada tarefa de verificação tanto para os produtos quanto para os processos, o número total de problemas e/ou não-conformidades encontrados, dentre outras. Todos os resultados produzidos devem ser documentados seguindo-se o estabelecido pelo processo de *Documentação*. As atividades que constituem o processo de *Verificação* são as seguintes:

Verif.I – preparar para o processo de *Verificação* (pág.112). Primeiramente, devem ser identificadas as relações entre o processo de *Verificação* e o de *Controle de Qualidade*. Feito isso, os analistas devem analisar os requisitos do projeto, a fim de que riscos envolvidos, caso haja erros, sejam identificados. As restrições das tecnologias a serem usadas e a disponibilidade dos recursos necessários precisam ser verificadas. Identificam-se as ferramentas para auxiliar nas atividades que compõem o processo, com base no processo de *Infra-Estrutura*. Um plano de verificação, a ser aprovado pelo gerente de projeto, é, então, elaborado.

Verif.II – verificar os produtos (pág.113). A esta atividade cabe revisar os requisitos do sistema e do software, o projeto, o código, a documentação e qualquer outro produto de trabalho, de acordo com o estabelecido no plano de verificação. Eventuais problemas e/ou não-conformidades que não puderem ser resolvidas no escopo do processo de *Verificação* devem ser encaminhadas ao processo de *Resolução de Problemas*. O gerente de projeto deve aprovar os resultados obtidos nas tarefas de verificação dos produtos.

Verif.III – verificar os processos (pág.113). É necessário revisar os procedimentos descritos nos planos para verificar se eles estão adequados à implementação dos processos. Além disso, a definição e a utilização dos padrões devem ser analisadas. É necessário, também, verificar a condição dos times para a realização das atividades, o que cabe ao administrador dos recursos e deve ser feito com base em resultados dos processos de *Capacitação e Treinamento*. Qualquer outra atividade ou tarefa deve ser analisada, de acordo com o descrito no plano de verificação. Eventuais problemas e/ou não-conformidades que não puderem ser resolvidas no escopo do processo de *Controle de Qualidade* devem ser encaminhadas ao processo de *Resolução de Problemas*. O gerente de projeto deve, por fim, aprovar os resultados obtidos nas tarefas de verificação dos processos.

3.8.6 VALIDAÇÃO

O processo de *Validação* determina se os requisitos e o produto final (software ou sistema) atendem ao uso específico proposto. Métricas que envolvem as atividades do processo incluem o tempo gasto em cada tarefa de validação, o número total de problemas e/ou não-conformidades encontrados e o número de testes conduzidos até a validação do produto. Todos os resultados obtidos devem ser documentados de acordo com o estabelecido pelo processo de *Documentação*. Para o perfeito andamento do processo, são necessárias as atividades descritas a seguir:

Valid.I – preparar para o processo de *Validação* (pág.114). Os analistas devem identificar as relações existentes entre o processo de *Validação* e o processo de *Controle de Qualidade*, bem como as ferramentas para auxiliar nas atividades que constituem o processo, de acordo com o estabelecido no processo de *Infra-Estrutura*. Um plano de validação, a ser aprovado pelo gerente de projeto, deve também ser elaborado.

Valid.II – validar um produto de software (pág.114). Em conjunto com o processo de *Operação*, é necessário que se preparem os requisitos, os casos e as especificações de teste para análise dos resultados. Os profissionais responsáveis devem, então, assegurar que o produto reflete os requisitos particulares para um uso específico do software e realizar testes de estresse, de limite, de entradas específicas, além dos testes junto ao usuário representativo, no seu ambiente de operação. Eventuais problemas e/ou não-conformidades que não puderem ser resolvidas no escopo do processo de *Validação* devem ser encaminhadas ao processo de *Resolução de Problemas*. Todos os resultados dos testes envolvidos nesta atividade devem ser aprovados pelo gerente de projeto.

3.8.7 RESOLUÇÃO DE PROBLEMAS

O processo de *Resolução de Problemas* pretende sistematizar a análise e a resolução dos problemas (incluindo não-conformidades), de qualquer natureza ou fonte, detectados durante o desenvolvimento, a operação, a gerência da configuração, o controle da qualidade, a verificação e a validação. O objetivo é fornecer meios que garantam, em tempo adequado e de maneira responsável e documentada, a análise e resolução de todos os problemas encontrados e a identificação das tendências de novas ocorrências. Métricas envolvendo as atividades do processo incluem a porcentagem de tempo gasto nas tarefas de análise e solução, o tempo

médio de resolução de um problema e/ou não conformidade e o número total de problemas e/ou não-conformidades resolvidos no projeto. O processo de *Resolução de Problemas* está intimamente ligado ao processo de *Gerência*. Para sua execução, são necessárias as seguintes atividades:

RProb.I – preparar para o processo de *Resolução de Problemas* (pág.115). A primeira tarefa desta atividade é a definição de mecanismos para categorizar e priorizar os problemas e/ou não-conformidades. Um plano de resolução deve ser elaborado, de forma a garantir que, após a detecção dos problemas e/ou não-conformidades, eles sejam rapidamente notificados e suas causas, identificadas, analisadas e, quando possível, eliminadas. Tal plano deve ser aprovado pelo gerente de projeto.

RProb.II – resolver os problemas e/ou não-conformidades (pág.115). Um relatório sobre cada problema e/ou não-conformidade deve ser elaborado. Tal relatório deve descrever cada problema e/ou não-conformidade e, fazendo uso dos mecanismos previamente definidos, categorizá-los e priorizá-los. Seguindo-se a prioridade estabelecida, fazendo-se uso de pessoal capacitado e em conjunto com o processo de *Gerência*, o problema e/ou não-conformidade deve ser analisado e solucionado. É necessário que seja atualizado o relatório e que ele constitua-se em parte integrante da documentação do processo que originou o problema e/ou a não-conformidade.

3.9 OS PROCESSOS ORGANIZACIONAIS

Os processos organizacionais estão relacionados às características, capacitação, recursos e infra-estrutura das empresas ou instituições que desenvolvem software. Estabelecem a estrutura necessária à realização das atividades e tarefas dos outros processos e definem atividades para a melhoria dos recursos humanos, através de treinamentos e do próprio ciclo de desenvolvimento de software. São organizados nos processos de *Gerência*, *Capacitação*, *Melhoria*, *Infra-Estrutura* e *Treinamento*.

3.9.1 GERÊNCIA

O processo de *Gerência* estabelece uma estrutura organizacional adequada para a realização das atividades relacionadas a esse processo, tornando possível a execução de todos

os processos praticados na empresa. O processo de *Gerência* tem participação direta no processo de *Resolução de Problemas*. Métricas envolvendo as atividades do processo incluem a porcentagem de tempo gasto em cada tarefa de preparação, o número de processos a serem executados na empresa/no projeto e o número médio de requisitos por processo identificado. As atividades que constituem o processo de *Gerência* são:

Gerem.I – preparar para o processo de *Gerência* (pág.116). Ao gerente da empresa cabe estabelecer a visão e cultura da organização, estabelecendo o negócio que ela deverá seguir. Os papéis iniciais para os processos organizacionais são estabelecidos. Determinam-se os requisitos dos processos a serem executados no ambiente da empresa, bem como seu escopo e as inter-relações existentes entre tais processos. Em conjunto com os processos de *Capacitação*, *Melhoria* e *Infra-Estrutura*, um roteiro de execução dos processos deve ser preparado, com base no escopo de cada um deles, contendo cronogramas, estimativas de esforço, recursos necessários, alocação de tarefas, determinação de responsabilidades, quantificação dos riscos associados a cada processo, medições e custos.

Gerem.II – administrar a execução de um processo (pág.116). A execução dos processos deve ser monitorada, permitindo conhecer-se o progresso das atividades. Além disso, em conjunto com o processo de *Resolução de Problemas*, deve analisar e solucionar os problemas propostos. Assegura, ainda, que as melhorias propostas pelo processo de *Melhoria* tenham sido colocadas em prática, favorecendo-as e monitorando sua execução, a fim de garantir que não sejam introduzidas inconsistências em outras atividades. O roteiro de execução dos processos, que serve de base para esta atividade, deve ser atualizado, sempre que melhorias forem incorporadas aos processos do ciclo de desenvolvimento de software.

3.9.2 CAPACITAÇÃO

O processo de *Capacitação* permite que sejam avaliados os times de trabalho, pela identificação de seus recursos humanos. Sob outra perspectiva, procura fornecer aos indivíduos da organização visão e cultura do processo de produção de software. Métricas envolvendo as atividades do processo incluem o número de palestras, discussões, encontros, etc. promovidos, a porcentagem de profissionais participantes, o número de profissionais por perfil identificado, dentre outras. O processo de *Capacitação* compreende as seguintes atividades:

Prep.I – iniciar o processo de *Capacitação* (pág.117). O administrador dos recursos, em conjunto com o gerente de projeto e ambos sob a supervisão do gerente da empresa, procura

selecionar e adequar normas, métodos, padrões, procedimentos técnicos, linguagens de programação e ferramentas, com base no negócio da empresa e em conformidade com o estabelecido no processo de *Gerência*. Por meio de cartilhas, palestras, discussões, encontros, fornecem aos indivíduos da empresa a visão e cultura do processo de produção de software, de forma a capacitá-los a trabalhar cooperativa e eficientemente. Um plano para obter informações periódicas sobre os times de trabalho deve ser elaborado, em conjunto com os processos de *Melhoria* e *Gerência*. Essas informações devem incluir, sobretudo, as capacidades técnica e gerencial de cada profissional. Ferramentas e/ou métodos para a coleta dos dados especificados no plano devem ser definidas.

Prep.II – determinar as capacidades dos profissionais (pág.117). Os dados identificados anteriormente devem ser coletados. As capacidades dos profissionais devem ser determinadas, com base nesses dados e segundo mecanismos de quantificação e qualificação definidos no plano. Um perfil deve ser associado a cada um deles. Por fim, um relatório de capacitação dos membros/times deve ser produzido.

3.9.3 MELHORIA

O processo de *Melhoria* define as atividades básicas que uma organização executa para estabelecer, avaliar, medir, controlar e melhorar cada processo envolvido no ciclo de desenvolvimento de seu software. Métricas para as atividades envolvidas no processo incluem a porcentagem de tempo gasto com cada tarefa de preparação, o número de métricas selecionadas ou definidas, o número total de melhorias efetuadas em cada processo, a evolução dos níveis de satisfação dos profissionais da empresa e dos clientes, etc. As atividades do processo de *Melhoria* podem ser assim descritas:

Melh.I – preparar para o processo de *Melhoria* (pág.118). Primeiramente devem ser definidos possíveis objetivos específicos de melhoria para cada processo (sobretudo quanto aos recursos humanos e computacionais), que possam refletir a melhoria do processo global da empresa, em conjunto com o estabelecido no processo de *Gerência*. Métricas²⁶ (objetivas e subjetivas) devem ser selecionadas ou definidas, não se perdendo de vista os objetivos pré-definidos. Procedimentos e ferramentas para coletar as métricas – sem sobrecarregar os times

²⁶ As medições encontradas no processo de *Melhoria* têm como objetivo melhorar os processos globais da empresa, num nível gerencial. Já as medições dos processos de *Verificação*, *Validação* e *Controle de Qualidade* são aplicadas apenas aos produtos e atividades do processo de *Desenvolvimento*. De qualquer forma, para serem efetivas, tais medições devem: (i) ter o foco em objetivos específicos; (ii) ser realizadas sobre todos os produtos, processos e recursos do ciclo de vida; e (iii) ter seus resultados interpretados com base em características do contexto organizacional e do ambiente (BASILI *et al*, 1994b).

em suas tarefas e sem afastá-los de suas atividades principais – devem ser identificados. Instrumentos de avaliação do nível de satisfação tanto dos indivíduos da empresa quanto dos clientes precisam ser estabelecidos, assim como mecanismos para se computar, qualitativa e quantitativamente, todos os resultados obtidos. Por fim, uma estratégia para melhoria de processo, com as conclusões obtidas, deve ser definida.

Melh.II – melhorar os processos (pág.118). As métricas indicadas anteriormente devem ser coletadas. Deve ser realizada, ainda, uma análise dos resultados obtidos, tanto das métricas coletadas, quanto dos relatórios (de capacitação dos profissionais, de resultados de treinamento, quando existirem, e de situação da infra-estrutura) que servem de entrada para esta atividade. É necessário documentar e armazenar cada medição, desde a coleta da métrica até a realização da modificação, nos casos em que ela ocorrer, sempre com a ciência e com a garantia do processo de *Gerência*. Além disso, estudos empíricos envolvendo as medições dos processos de software e os dados armazenados na “base de dados históricos” devem ser feitos. Por fim, deve-se elaborar um plano de melhoria, para cada processo, caracterizando sua situação atual e sugerindo possíveis mudanças (inclusive na infra-estrutura ou baseadas em treinamento), com base na análise dos resultados obtidos e considerando o impacto da mudança nos demais processos e para o negócio da empresa.

3.9.4 INFRA-ESTRUTURA

O processo de *Infra-Estrutura* define as atividades para estabelecer e manter a infra-estrutura necessária para qualquer outro processo. A infra-estrutura pode incluir hardware, software, ferramentas, técnicas, padrões e recursos necessários à realização das atividades e tarefas que compõem um dado processo. Métricas para as atividades envolvidas no processo incluem o número de recursos identificados, o número de modificações na infra-estrutura por unidade de tempo e o tempo gasto na instalação da infra-estrutura. As atividades que constituem o processo de *Infra-Estrutura* são:

IEstrut.I – definir a infra-estrutura e sua instalação (pág.119). O administrador dos recursos deve definir a infra-estrutura necessária, em conjunto com o estabelecido no processo de *Gerência*. Deve haver planejamento para aquisição (quando necessária), instalação e funcionamento da infra-estrutura. Além disso, a estrutura e o acesso à “base de dados históricos” da empresa, que serve para o registro das terminologias e dos eventos relacionados ao projeto, armazenando, assim, documentos importantes para estimativas e melhorias, também são definidos por esta atividade.

IEstrut.II – instalar e manter a infra-estrutura (pág.119). Esta atividade envolve a aquisição (quando necessária), a instalação e a renovação da infra-estrutura, seguindo-se o estabelecido no plano gerado pela atividade anterior. É preciso monitorar a utilização da infra-estrutura, emitindo-se relatórios periódicos que propõem mudança, quando elas se fazem necessárias. As mudanças, quando ocorrem, devem ser realizadas de acordo com o processo de *Melhoria*.

3.9.5 TREINAMENTO

O processo de *Treinamento* refere-se à melhoria do conhecimento dos times de trabalho, com relação às habilidades gerenciais e técnicas. É essencial que o treinamento de pessoal seja planejado e implementado com antecedência, para se conseguir um melhor rendimento dos profissionais no desempenho de suas tarefas. Métricas para as atividades deste processo incluem a porcentagem de profissionais com necessidade de treinamento para cada processo, o tempo total gasto no treinamento, o nível de aproveitamento, por profissional e globalmente, dentre outras. Compõem o processo de *Treinamento* as atividades:

Trein.I – preparar para o processo de *Treinamento* (pág.120). É necessário identificar, em conformidade com o estabelecido nos processos de *Capacitação* e *Infra-Estrutura*, os recursos técnicos e humanos necessários para a realização das atividades. Um plano de treinamento, estabelecendo cronogramas, metodologias e procedimentos para avaliação do aproveitamento deve ser gerado, tomando por base as diretrizes de treinamento, provenientes de algum processo para o qual esse treinamento seja necessário.

Trein.II – realizar o treinamento (pág.120). O material de treinamento é elaborado, seguindo-se o plano. Os profissionais são treinados e, ao final do treinamento, uma avaliação do aproveitamento deve ser feita. O perfil de cada profissional deve ser atualizado, em conjunto com o processo de *Capacitação*. Todos os resultados devem ser compilados num relatório do treinamento, a ser encaminhado ao processo de *Melhoria* para estudos empíricos. Se necessário, tal relatório deve sugerir – fundamentando-se nos resultados – modificações a serem realizadas.

3.10 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentados os aspectos constituintes da abordagem *GQS-AE* desenvolvida. Foram descritos seus processos e as atividades que os constituem. O **Apêndice** fornece, em conjunto com as informações expostas aqui, uma melhor compreensão da estrutura e do funcionamento da abordagem.

O capítulo seguinte apresenta uma instanciação da abordagem *GQS-AE*, tomando por base a maneira de se desenvolver software de uma determinada empresa de pequeno porte, a ser caracterizada no capítulo seguinte. Tal instanciação tem o propósito único de verificar a aplicabilidade do que foi desenvolvido.

CAPÍTULO 4: APLICABILIDADE DA ABORDAGEM *GQS-AE*

4.1 CONSIDERAÇÕES INICIAIS

Neste capítulo são descritos os resultados encontrados quando da instanciação da abordagem *GQS-AE* na empresa LinkWay. Escolheu-se a referida empresa pelo fato de ela possuir as características de uma pequena empresa de software e também porque a mesma já havia participado de um projeto anterior no qual foi documentado seu processo de desenvolvimento de software (TAVARES, 2002), o qual foi tomado como base para a instanciação.

Essa instanciação visava a determinar a aplicabilidade da abordagem, ou seja, esperava-se verificar se o que foi definido única e exclusivamente sobre uma base teórica representaria as características de um cenário real e conteria as condições para que fosse utilizado em uma situação prática.

Antes dos resultados obtidos, porém, uma breve descrição da empresa LinkWay é apresentada, assim como uma, também breve, descrição sobre o processo que serviu de base para a referida instanciação.

4.2 A EMPRESA LINKWAY

O texto descrito abaixo foi definido pela empresa, na intenção de caracterizar seus produtos e apresentar seus objetivos em relação ao trabalho realizado e consta do trabalho de TAVARES (2002).

“A LinkWay é uma empresa de Prestação de Serviços que atua no segmento de conexão a Internet, desenvolvimento de aplicações para Web e serviços de Telecomunicações, autorizada pela ANATEL (Agência Nacional de Telecomunicações).

A Matriz administrativa da LinWay é localizada em Rio Claro/SP e suas unidades comerciais estão localizadas em oito cidades do interior de São Paulo. O departamento de pesquisa e desenvolvimento de novas tecnologias fica em São Carlos/SP, no qual a empresa mantém uma equipe de circuito e redes de Telecomunicações e o Centro de Desenvolvimento de Aplicações Web, que são compostos por profissionais qualificados, atualmente responsáveis por Projetos de Conectividade e Soluções de Aplicações na Internet para empresas de pequeno, médio e grande porte de todo o Brasil.

A LinkWay investe continuamente na melhoria de processos de desenvolvimento, buscando com isso um melhor gerenciamento e controle das atividades internas, o que resulta em maior satisfação dos clientes, melhor qualidade dos produtos e aprimoramento profissional de seus funcionários. A busca pela certificação CMM – nível 2 é a garantia de que a LinkWay é uma empresa empenhada em superar as expectativas de seus clientes satisfazendo assim o crescente mercado da Internet.”

4.3 O PROCESSO-BASE PARA A INSTANCIAÇÃO

O trabalho de TAVARES (2002) teve como objetivo criar uma estratégia para diagnosticar a maneira como uma empresa de pequeno porte desenvolve software. Essa maneira foi definida formalmente e sua melhoria, planejada. Para tanto, foi realizado um estudo empírico na empresa LinkWay, a partir do qual foi gerada uma representação gráfica do processo da empresa, apresentada na *Figura 12*. Ela caracteriza o fluxo de dados que

ocorre quando se desenvolve um *site* para comércio eletrônico (denominado, na empresa, como *Shopping*).

Além dessa representação gráfica, foram gerados, ainda, *templates* que correspondem às atividades envolvidas no processo, representadas na figura por meio de retângulos com cantos arredondados. Tais *templates* possuem estrutura semelhante à dos utilizados neste trabalho.

Figura 12: Processo melhorado de desenvolvimento de software - Shopping.

Referência: Extraído de TAVARES (2002).

4.4 A INSTANCIAÇÃO

O objetivo da instanciação realizada foi verificar se a abordagem *GQS-AE*, baseada exclusivamente nos conceitos teóricos para se alcançar objetivos de melhoria de qualidade de software, abrangeria as características reais do ambiente de uma pequena empresa e contemplaria os aspectos necessários para que fosse empregada em uma situação prática.

Uma analogia para se entender o que vem a ser a instanciação é pensar na abordagem como uma roupa. Instanciá-la, então, para a empresa, significa verificar se a roupa pode ser usada. Isso é diferente do *uso* em si, que demandaria, por exemplo, ajustes de tamanho.

Os ajustes seriam, pois, para este trabalho, as modificações a serem realizadas na abordagem (estruturais e/ou funcionais). É importante ressaltar que o caráter evolucionista da abordagem *GQS-AE* permite que ela seja “ajustada”, na medida em que surjam as necessidades de modificações. Os ajustes, porém, não foram realizados como parte deste trabalho devido a algumas limitações existentes. A abordagem foi apenas *instanciada* na empresa, ou seja, não foi *utilizada* pela mesma seguindo-se um estudo de caso específico.

Feitas essas considerações, descreve-se a seguir como a instanciação foi conduzida.

- Passo 1: elaboração de um questionário baseado no campo **Tarefas** de cada *template* utilizado na abordagem *GQS-AE*. O objetivo do questionário foi o de identificar tarefas propostas na abordagem que não são realizadas na empresa ou o são de forma parcial. A execução das tarefas propostas foi graduada em três níveis, a saber: **não cumpre**, **cumpr parcialmente** e **cumpr totalmente**. Tal questionário – sem as respostas fornecidas – é apresentado no **Apêndice D**.
- Passo 2: aplicação do questionário, o que feito por meio de entrevistas semi-estruturadas com profissionais da empresa alocados para a instanciação.
- Passo 3: estudo do processo de desenvolvimento de *site* de comércio eletrônico praticado pela empresa LinkWay, definido no trabalho de TAVARES (2002), utilizando-se a representação gráfica desse processo e os *templates* das atividades correspondentes.

- Passo 4: comparação entre o que foi assinalado no questionário e o processo apresentado por TAVARES (2002), a fim de que, de certa forma, fossem “validadas” as respostas fornecidas no questionário.
- Passo 5: discussão, por meio de reuniões, de divergências e/ou dúvidas que vieram surgiram.
- Passo 6: apresentação das tarefas que não foram totalmente cumpridas na empresa, considerando-se cada processo proposto na abordagem *GQS- $\mathcal{AE}$* . O nível de refinamento para se fornecer tais resultados chegou à menor unidade necessária para se executar um determinado processo – uma tarefa. Resultados por atividades não foram apresentados. Além disso, os resultados a que se faz alusão aqui foram baseados apenas no número de tarefas que constituem o processo, não se associando níveis diferentes de importância (peso) a cada uma delas.
- Passo 7: discussão com os profissionais da empresa para identificar a viabilidade das tarefas não cumpridas integralmente.

4.5 RESULTADOS QUANTITATIVOS

A seguir, são apresentados 17 gráficos, um para cada processo da abordagem *GQS- $\mathcal{AE}$* , indicando a porcentagem de cumprimento, pela empresa LinkWay, das tarefas envolvidas nos processos. Tais gráficos foram gerados a partir dos dados coletados no questionário.

(03) Processo de *Desenvolvimento*

(04) Processo de *Operação*

(05) Processo de *Manutenção*

(06) Processo de *Documentação*

(07) Processo de *Acompanhamento*

(08) Processo de *Gerenciamento de Configuração*

(09) Processo de *Controle de Qualidade*

(10) Processo de *Verificação*

Figura 13: Gráficos do cumprimento das tarefas descritas na abordagem *GQS-AE*.

CAPÍTULO 5: CONCLUSÕES E TRABALHOS FUTUROS

5.1 CONCLUSÕES

Neste trabalho, a abordagem *GQS-AE*, para se garantir a qualidade do software produzido por uma pequena empresa, foi desenvolvida, tomando como princípio o gerenciamento dos processos envolvidos no desenvolvimento desse software. A elaboração da abordagem procurou se manter fiel a duas premissas básicas que, quando não seguidas, dificultam e, por vezes, inviabilizam o desenvolvimento de um produto:

- Um processo de software bem **definido e documentado** – utilizado para integrar pessoas, tarefas, ferramentas e métodos – pode prover a base essencial para garantir a qualidade do produto final.
- Um processo de software **gerenciado** propicia segurança frente às variações que o produto possa sofrer em relação às suas especificações iniciais.

Assim, uma contribuição importante do trabalho desenvolvido é tornar transparente, às pequenas empresas, atividades de engenharia e de gerência que, na grande maioria dos casos, não são desenvolvidas por tais empresas pela falta de entendimento que as mesmas têm dessas atividades.

Sobre o desenvolvimento da abordagem, há que se registrar a dificuldade na estruturação da mesma, dadas as relações de interdependência funcionais e até estruturais existentes entre os processos que a compõem. Identificou-se, assim, uma necessidade premente de empenho e comprometimento por parte das empresas que buscam implantar atividades para a garantia da qualidade de seu software.

A abordagem, porém, e a despeito do conhecimento e dos recursos utilizados na sua elaboração, apresenta um caráter deficiente que lhe é inerente. Isso porque ela é baseada na *evolução*, devendo ser, assim, entendida como uma primeira versão ou versão inicial de uma estratégia para se melhorar a qualidade do software produzido por uma organização de pequeno porte. Ela apresenta, seguramente, um número reduzido de métricas, talvez um relacionamento falho entre as tarefas existentes ou ainda tarefas incompletas ou baseadas em critérios de inicialização e finalização não muito exatos. E tais deficiências não podem ser encobertas nem ignoradas. É preciso que sejam propostas modificações tanto na estrutura da abordagem quanto no seu funcionamento, para assegurar que os níveis de qualidade dos produtos de software sejam sempre crescentes e que a abordagem esteja sempre adaptada à realidade da empresa.

Ressalta-se, então, que, nas condições em que foi concebida a abordagem *GQS-AE*, uma certa insegurança quanto a sua aplicação veio à tona. Com o intuito de eliminá-la, um estudo da aplicabilidade da abordagem foi conduzido, instanciando-a para a realidade de uma empresa de pequeno porte que atua no segmento de conexão à Internet, desenvolvimento de aplicações para Web e telecomunicações – chamada LinkWay.

Ao se realizar a instancição, muitos dos conceitos propostos e das noções exigidas pela abordagem já estavam bastante claros para os profissionais que participaram dessa fase, sobretudo para o gerente da empresa. Eles procuraram estar bastante comprometidos com as discussões que surgiram ao longo do processo, demonstrando, por vezes, anseio de uma padronização em seu ambiente de trabalho.

Nesse sentido, merece destaque o trabalho de TAVARES (2002), que serviu de base para a instancição da abordagem *GQS-AE* e deu passos significativos com a empresa LinkWay, auxiliando-a na definição de seu processo de desenvolvimento de software.

A conclusão a que se chegou com a instanciación da abordagem foi a de que a empresa está empenhada em definir bem e ter um controle mais rigoroso de todos os processos envolvidos no desenvolvimento de seus produtos, tendo, inclusive, diretrizes já definidas para que isso ocorra, impreterivelmente, a partir do início do próximo ano. Mais do que isso, a abordagem provou-se ser aplicável em uma situação prática, o que dá força a uma sua efetiva aplicação, não apenas em uma, mas em várias empresas que se mostrarem interessadas.

Destaca-se, entretanto, o que se inferiu com a instanciación: teoria e prática costumam não “caminhar no mesmo passo”. E prover-lhes sincronismo não é tarefa fácil porquanto não haja comprometimento necessário por parte de cada profissional, não importando o fato de ele estar mais comprometido com um ou com outro aspecto.

5.2 TRABALHOS FUTUROS

Indica-se, como trabalhos futuros para que sejam continuadas as pesquisas relacionadas à qualidade dos processos (e, por implicação, dos produtos) de software em empresas de pequeno porte, sobretudo com base neste desenvolvido aqui, os seguintes trabalhos:

- Uma vez que a instanciación conduzida não propôs sugestões de alteração para nenhum dos documentos produzidos, um estudo de caso, com base em tudo o que foi definido, deveria ser conduzido. Nesse estudo, seriam, então, coletadas as métricas definidas e feitas análises sobre todos os dados (objetivos e subjetivos encontrados ao longo do estudo), o que culminaria numa avaliação bastante significativa do uso efetivo da abordagem num ambiente real de desenvolvimento de software e, sobretudo, na garantia de qualidade no escopo de um projeto ou conjunto de projetos.
- Poderiam ser definidas, com base nos resultados encontrados quando da aplicação da abordagem *GQS-AE*, diretrizes para estabelecer prioridades e planejar ações de melhoria, semelhante à estratégia proposta no trabalho “Diretrizes para o estabelecimento de melhoria de processos adequada a organizações de pequeno porte” (FIATS, 2000).

- Para a aplicação da abordagem, um planejamento deve existir, contendo planos e procedimentos para execução das atividades, propondo, inclusive, treinamento para os casos em que ele se fizer necessário. Ferramentas de apoio a essa atividade poderiam, então, ser utilizadas ou mesmo desenvolvidas.
- É recomendado um relatório que contenha estimativas para cumprimento das tarefas que foram identificadas como não sendo cumpridas. Para tanto, a observação participativa no ambiente da empresa poderia ser útil, bem como entrevistas semi-estruturadas a respeito do andamento das atividades relacionadas a cada projeto.
- A abordagem *GQS-AE* poderia ser modificada e estruturada apenas com base no CMMI (para empresas que procuram a certificação CMM), de maneira a acomodar apenas as KPAs necessárias para o nível que se pretende alcançar, no caso de se utilizar o “modelo em estágios”, ou todas as KPAs, no caso de se utilizar o “modelo contínuo”.
- O questionário para avaliação da aplicabilidade da abordagem poderia ser modificado, tornando-se mais simples, porém não menos robusto, facilitando a identificação, por parte dos profissionais entrevistados, das tarefas não executadas dentro da empresa. A graduação das respostas para cada uma das tarefas poderia ser feita em quatro níveis, em vez de três, sendo: **0 – Não cumpre, 1 – Cumpre fracamente, 2 – Cumpre fortemente e 3 – Cumpre totalmente**. Níveis adicionais poderiam ser criados, dependendo da necessidade. Esses níveis poderiam, ainda, variar de acordo com cada processo.
- A abordagem *GQS-AE* pode ser entendida como uma rede de trabalho, com diversos fluxos de dados de entrada e saída, controlados por uma linha temporal. Sob essa perspectiva, um ambiente de simulação do seu comportamento seria bastante interessante. Isso poderia minimizar os riscos e diminuir sensivelmente as perdas geradas pelas modificações necessárias num ambiente real de desenvolvimento de software, ou seja, numa empresa de pequeno porte.

REFERÊNCIAS BIBLIOGRÁFICAS

(ANDRADE <i>et al</i> , 1996)	Andrade, A. L. <i>et al. Aplicação da norma ISO 12119 na avaliação da qualidade de produtos de software</i> . VII CITS, Curitiba, 1996.
(BARROS, 1991)	Barros, C. D. C. <i>Qualidade e participação: o caminho para o êxito</i> . São Paulo:Nobel, 1991.
(BASILI <i>et al</i> , 1994a)	Basili, V. <i>et al. Measurement</i> . In: The Encyclopedia of Software Engineering. [s.l.]:John Wiley & Sons, 1994.
(BASILI <i>et al</i> , 1994b)	Basili, V. <i>et al. Goal Question Metric paradigm</i> . In: The Encyclopedia of Software Engineering. [s.l.]:John Wiley & Sons, 1994.
(BELT, 1995)	Belt, A. V. <i>Representation of workflow design knowledge for the Scarabaus Project</i> . 1995. Dissertação (Mestrado em Ciências da Computação). Department of Computer Science, University of Twente, Twente.
(BELLOQUIM, 2002)	Belloquim, A. <i>CMMI: o futuro do CMM</i> . Disponível em: < http://www.choose.com.br/infochoose/artigos/viewer.asp?n=42&a=1 >. Acesso em 04/04/2002. Website desenvolvido em 1995.

(CASATI et al, 1995)	Casati, F. et al. <i>Conceptual Modeling of workflows</i> . In: International Conference on Object-Oriented and Entity-Relationship Modeling – OOER’95, Gold Coast, Austrália, 1995. Proceedings... Gold Coast:[s.c.], 1995.
(CROSBY, 1979)	Crosby, P. B. <i>Quality is free – the art of making quality certain</i> . Nova York:McGraw-Hill, 1979.
(FALLAH, 1994)	Fallah, M. H.; Jrad, A. M. <i>SQA – a proactive approach to assuring software quality</i> . AT&T Technical Journal , v.73, n.1, p.26-33, jan-fev., 1994.
(FIATS, 2000)	Fiats, M. I. F. <i>Diretrizes para o estabelecimento de melhoria de processos adequada a organizações de pequeno porte</i> . 2000. Dissertação (Mestrado em Ciência da Computação). Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos/SP.
(HUMPHREY, 1989)	Humphrey, W. <i>Managing the software process</i> . Reading:Addison-Wesley, 1998. SEI Series in Software Engineering.
(IEEE, 1983)	IEEE. <i>Standard glossary of Software Engineering terminology</i> . Software Engineering Technical Comittee of the IEEE Computer Society, 1983.
(IEEE, 1989)	IEEE. <i>Software Engineering Standards</i> . 3 rd ed. IEEE, 1989.
(ISO 12207, 1995)	ISSO/IEC 12207. <i>Information Technology – software life cycle process</i> , 1995.
(ISO 9001, 1994)	ISSO/IEC 9001:1994. <i>Quality systems. Model for quality assurance in: design, development, production, installation and servicing</i> , 1994.
(ISO 9001, 2000)	ISSO/IEC 9001:2000. <i>Quality management systems</i> . Requirements, 2000.
(ISO TR 15504, 1998)	ISSO/IEC TR 15504. <i>Information Technology – software process assessment – parts 1-9</i> , 1998.
(JONES, 1981)	Jones, T. C. <i>Programming productivity: issues for the 80s</i> . [S.l.]:IEEE Computer Society Press, 1981. p.13-20.
(JONES, 1992)	Jones, C. <i>Critical problems in software measurement – version 1.0</i> . Burlington:Software Productivity Research (SPR), 1992.

(KAN, 1995)	Kan, S. H. <i>Metrics and models in software quality engineering</i> . [s.l.]:Addison-Wesley, 1995.
(LAMPRECHT, 1994)	Lamprecht, J. L. <i>ISO-9000 e o setor de serviços: uma interpretação crítica das revisões</i> . Rio de Janeiro:Qualitymark, 1994. p.237-253.
(MAIDANTCHIK, 1999)	Maidantchik, C. L. L. <i>Gerência de processos de software para equipes geograficamente dispersas</i> . 1999. Dissertação (Doutorado em Ciência da Computação). COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro/RJ.
(MCT/SEPIN, 2002)	Ministério da Ciência e Tecnologia, Secretaria de Política de Informática e Automação, <i>Qualidade e produtividade no setor de software brasileiro</i> . Disponível em < http://www.mct.gov.br/sepin/Dsi/Software/Menu_Qualidade.htm >. Acesso em 12/03/2002.
(MOREAU, 1998)	Moreau, B.; Martin, J. M. <i>Introduction au referential ISO/SPICE (ISO/CEI TR 15504) et à son utilisation pour le management de la qualité des processus du logiciel</i> . Relatório Técnico, Association Française de Normalisation (AFNOR), FD Z 67-910, 1998.
(NBR 13596, 1996)	NBR 13596:1996. <i>Tecnologia de informação – avaliação de produto de software – características de qualidade e diretrizes para o seu uso</i> , 1996.
(NBR 9000-3, 1997)	NBR ISO 9000-3:1997. <i>Normas de gestão e garantia da qualidade – parte 3: diretrizes para a aplicação da NBR ISO 9001 ao desenvolvimento, fornecimento e manutenção de software</i> , 1997.
(OLIVEIRA, 1995)	Oliveira, A. L. M. <i>Processo de software – visão global</i> . Campinas:[s.e.], 1995. (Relatório Técnico – Fundação CTI – Brasil).
(OMAN, 1997)	Oman, P.; Pflieger, S. L. <i>Applying software metrics</i> . IEEE Computer Society, Los Alamitos, CA, 1997.
(PAULK et al, 1993)	Paulk, M. C. et al. <i>Key Practices of the Capability Maturity Model, version 1.1</i> . Pittsburgh:SEI, 1993. (Technical Report CMU/SEI-93-TR-025).

-
- (PAULK *et al*, 1995) Paulk, M. C. *et al*. *The Capability Maturity Model: guidelines for improving the software process*. Reading:Addison-Wesley, 1995. (SEI Series in Software Engineering).
-
- (PRESSMAN, 2002) Pressman, R. S. *Engenharia de software, 5ª edição*. Rio de Janeiro:McGraw-Hill, 2002.
-
- (ROCHA *et al*, 2001) Rocha, A. R. C.; Maldonado, J. C.; Weber, K. C. *Qualidade de software – teoria e prática*. São Paulo:Prentice Hall, 2001.
-
- (ROTHERY, 1993) Rothery, B. *ISO 9000*. Rio de Janeiro:Makron Books, 1993.
-
- (SMITH, 1989) Smith, D.J.; Wood, K. B. *Engineering quality software – a review of current practices, standards and guidelines including new methods and development tool*. [s.l.]:Elsevier Science Publishers Ltd., 1989.
-
- (TAVARES, 2002) Tavares, D. P. D. *Melhoria de processo de pequena empresa: um estudo de caso*. 2002. Dissertação (Mestrado em Ciência da Computação). Centro de Ciências Exatas e de Tecnologia, Universidade Federal de São Carlos, São Carlos/SP.
-
- (TOLEDO, 1987) Toledo, J. C. *Qualidade industrial: conceitos, sistemas e estratégias*. São Paulo:Atlas, 1987.
-
- (VERA, 1983) Vera, A. A. *Metodologia de pesquisa científica*. Porto Alegre: Globo, 1983.
-
- (WERNECK, 1994) Werneck, D. *O movimento da qualidade*. Folha de São Paulo (Fascículo SEBRAE – Qualidade Total), São Paulo, março, 1994.
-

BIBLIOGRAFIA COMPLEMENTAR

-
- (ARES *et al*, 2000) Arcs, J. *et al*. *A more rigorous and comprehensive approach to software process assessment*. **Software Process Improvement and Practice**, v.5, n.1, mar., 2000.
-
- (ARNOLD, 1995) Arnold, J. E. *Toward collaborative software process*. In: 17th International Conference on Software Engineering, Seattle, Washington, EUA, 1995. **Proceedings...**
-
- (ARRUDA *et al*, 1998) Arruda, M. N. *et al*. *Avaliação CMM – A experiência CPqD Telebrás*. In: Workshop de Qualidade de Software, Maringá, Paraná, Brasil, 1998. **Proceedings...**
-
- (ARTHUR, 1993) Arthur, L. J. *et al*. *Avaliação CMM – A experiência CPqD Telebrás*. In: Workshop de Qualidade de Software, Maringá, Paraná, Brasil, 1998. **Proceedings...**
-
- (BACH, 1989) Bach, J. *Microdynamics of process evolution*. **IEE Computer**, fev., 1998.
-
- (BANDINELLI *et al*, 1995) Bandinelli, S. *et al*. *Modeling and improving an industrial software process*. **IEEE Transactions on Software Engineering**, v.21, n.5, pp.440-454, maio, 1995.
-

(BELLOQUIM, 1994)	Belloquim, A. <i>CMM em pequenas organizações: seria mesmo possível?</i> Developer Magazine , jan.1994. p.26-28.
(BOEHM, 1996)	Boehm, B. <i>Anchoring the software process</i> . IEE Software , pp.73-82, jul., 1996.
(BOLCER, 1996)	Bolcer, G. A. <i>Endeavors: a process system integration infrastructure</i> . In: 4 th International Conference on Software Process – ICSP’4, Brighton, UK, 1996. Proceedings...
(BRÖCKERS, 1996)	Bröckers, A.; Differding, C.; Threin, G. <i>The role of software process modelling in planning industrial measurement programs</i> . In: 3 rd International Software Metrics Symposium, Berlim, Alemanha, 1996. Proceedings...
(BROWNSWORD et al, 2000)	Brownsword, L. <i>et al. Applying CMM Project planning practices to diverse environments</i> . IEEE Software . jul-ago., 2000.
(CARVALHO, 1997)	Carvalho, M. B.; Frossard, R. S.; Padilha, A. M. <i>Certificação X Melhoria</i> . In: XI Simpósio Brasileiro de Engenharia de Software, Fortaleza, Ceará, Brasil, 1997. Proceedings of the Workshop de Qualidade de Software...
(CURTIS, 1992)	Curtis, B. <i>Maintaining the software process</i> . In: IEEE Conference on Software Maintenance , Orlando, 1996. Proceedings... Orlando:[s.e.], 1996. p.2-8.
(CURTIS, 1992)	Curtis, B.; Kellner, M. I.; Over, J. <i>Process modeling</i> . Communications of the ACM , v.35, n.9, pp.75-90, set., 1992.
(CURTIS, 1997)	Curtis, B. <i>Software process improvement: methods and lessons learned</i> . In: 19 th International Conference on Software Engineering, Boston, Massachusetts, EUA, 1997. Proceedings...
(CURTIS, 1998)	Curtis, B. <i>Which comes first, the organisation or its process?</i> IEEE Software , v.15, n.6, pp.10-13, nov-dez., 1998.
(CURTIS, 2000)	Curtis, B. <i>The global pursuit of process maturity</i> . IEEE Software . jul-ago., 2000.

(DEMIRÖRS <i>et al</i> , 1998)	Demirörs, E. <i>et al. Process improvement towards ISO 9001 Certification in a small organisation</i> . In: 20 th International Conference on Software Engineering, Kyoto, Japão, 1998. Proceedings...
(ELLMER, 1995)	Ellmer, E. <i>Improving software process</i> . In: Software Engineering Environments Conference, Nordwijkerhout, Holanda, 1995. Proceedings...
(EMAM, 1996)	Emam, K. E.; Goldenson, D. R. <i>Some initial results from the international SPICE trials</i> . Software Process Newsletter , v.spring, n.6, pp.1-5, 1996.
(EMAM, 1998)	Emam, K. E.; Drouin, J.; Melo, W. <i>SPICE – The theory and practice of Software Process Improvement and Capability dEtermination</i> . IEEE Computer , Society Press, 1998.
(FIORINI <i>et al</i> , 1998)	Fiorini, S. T. <i>et al. Engenharia de software com CMM</i> , Rio de Janeiro: Brasport, 1998.
(FOX, 1997)	Fox, C.; Frakes, W. <i>The quality approach: is it delivering?</i> Communications of the ACM , v.40, n.6, pp.24-29, jun., 1997.
(FUGGETTA, 2000)	Fuggetta, A. <i>Software process: a roadmap</i> . In: The future of software engineering, A. Finkelstein:[s.l.], 2000.
(GENTLEMAN, 2000)	Gentleman, W. M. <i>If software quality is a perception, how do we measure it?</i> . Disponível em < http://seg.iit.nrc.ca/papers/NRC40149.pdf >. Acesso em 28/07/2000.
(GIBSON, 1997)	Gibson, R. <i>Software process modeling and assessment</i> . In: 9 th International Conference on Software Engineering & Knowledge – SE-KE'97, Madri, Espanha, 1997. Proceedings...
(HAREL, 1987)	Harel, D. <i>Statecharts: a visual formalism for complex systems</i> . Science of Computer Programming , n.8, 1987.
(HERBSLEB <i>et al</i> , 1997)	Herbsleb, J. <i>et al. Software quality and the Capability Maturity Model</i> . Communications of the ACM , v.40, n.6, pp.30-40, jun., 1997.
(HOLLENBACH <i>et al</i> , 1997)	Hollenbach, C. <i>et al. Combining quality and software improvement</i> . Communications of the ACM , v.40, n.6, pp.41-45, jun., 1997.

(HUMPHREY, 1990)	Humphrey, W. S. <i>Managing the software process</i> . Addison-Wesley Publishing Co.:[s.l.], 1990.
(HUMPHREY, 2001)	Humphrey, W. <i>Pathways to process maturity: the Personal Software Process and Team Software Process</i> . Disponível em < http://interactive.sei.cmu.edu/Features/1999/June/Background/Background.jun99.htm >. Acesso em 18/04/2001.
(HUMPREY, 1996)	Humphrey, W. S. <i>Using a defined and measurable personal software process</i> . IEEE Software , pp.77-88, maio, 1996.
(KRASNER et al, 1992)	Krasner, H. et al. <i>Lessons learned from a software process modelling system</i> . Communications of the ACM , v.35, n.9, pp.91-100, set., 1992.
(LAI, 1993)	Lai, R. <i>The move to maturity process</i> . IEEE Software , pp.14-17, jul., 1993.
(LEHMAN, 1995)	Lehman, M. M. <i>Software process improvement – the way forward</i> . In: VII International Conference on Advanced Information Systems Engineering – CaiSE'95, Jyväskylä, Finlândia, 1995. Proceedings...
(LEHMAN, 1997)	Lehman, M. M. <i>Process modelling – where next?</i> In: 19 th International Conference on Software Engineering, Boston, Massachusetts, EUA, 1997. Proceedings...
(LIGHTFOOT, 2002)	Lightfoot, T. <i>The Software Process Improvement and Capability Determination</i> . Disponível em: < http://www-sqi.cit.gu.edu.au/spice >. Acesso em 04/04/2002.
(MACHADO, 2000)	Machado, L. F. D. C.; Oliveira, K. M.; Rocha, A. R. C. <i>Using standards and maturity models for the software process definition</i> . In: 4 th International Software Quality Week Europe – QWE 2000, Bélgica, 2000.
(MADHAVJI, 1993)	Madhavji, N. H.; Pencdo, M. H. <i>Special section on the evolution of software process</i> . IEEE Transactions on Software Engineering , v.19, n.12, pp.1125-1127, dez., 1993.
(MI, 1991)	Mi, P.; Scacchi, W. <i>Modeling articulation work in software engineering process</i> . In: 1 st International Conference on Software Process – ICSP'1, Redonto Beach, California, EUA, 1991. Proceedings...

(OSKARSSON & GRASS, 1996)	Oskarsson, O.; Grass, R. L. <i>An ISO 9000 approach to build quality software</i> . New Jersey:Prentice Hall, 1996.
(OSTERWEIL <i>et al</i> , 1996)	Osterweil, L. J. <i>et al</i> . <i>Strategic directions in software quality</i> . ACM Computing Surveys , v.28, n.4, pp.738-750, dez., 1996.
(PAULA FILHO, 2001)	Paula Filho, W. P. <i>Engenharia de software: fundamentos, métodos e padrões</i> . Rio de Janeiro:LTC, 2001
(PAULK, 1994)	Paulk, M. C. <i>How ISO 9001 compares with the CMM</i> . IEEE Software , pp.74-83, jan., 1994.
(PERRY <i>et al</i> , 1994)	Perry, D. <i>et al</i> . <i>People, organisations and process improvement</i> . IEEE Software , pp.36-45, jul., 1994.
(SAKAMOTO <i>et al</i> , 1998)	Sakamoto, K. <i>et al</i> . <i>Toward computational support for software process improvement activities</i> . In: 20 th International Conference on Software Engineering, Kyoto, Japão, 1998. Proceedings...
(SCHNEIDEWIND, 1996)	Schneidewind, N. F.; Fenton, N. <i>Do standards improve quality?</i> IEEE Software , pp.22-24, jan., 1996.
(SEI, 2002)	SEI. <i>Capability Maturity Model Integration - CMMI</i> . Disponível em: < http://www.sei.cmu.edu/cmmi >. Acesso em 23/09/2002.
(SOBEY, 2000)	Sobey, A. <i>Project management – software quality assurance</i> . Disponível em < http://louisa.levels.unisa.edu.au/sc1/management-sqa/wcek7_02.htm >. Acesso em 19/11/2000.
(SOMMERVILLE, 1996)	Sommerville, I. <i>Software process models</i> . ACM Computing Surveys , v.28, n.1, pp.269-271, mar.,1996.
(TANTARA INC., 2001)	Tantara Inc. <i>Software process improvement and related standards/models</i> . Disponível em < http://www.tantara.ab.ca/astds.htm >. Acesso em 19/11/2001.
(WALLACE, 2000)	Wallace, D. R.; Peng, W. W.; Ippolito, L. M. <i>Software quality assurance: documentation and reviews</i> . Disponível em < http://hiss.ncsl.nist.gov/publications/nistir4909/ >. Acesso em 19/11/2000.
(WEBER, 2001)	Weber, K. C.; Rocha, A. R. C.; Nascimento, C. J. <i>Qualidade e produtividade em software</i> , 4 ^a edição renovada. São Paulo: Makron Books, 2001.

APÊNDICE A

Atividade:	identificar as necessidades iniciais (Def.I)
Objetivo:	definir o problema e a necessidade do cliente em adquirir, desenvolver ou aprimorar um sistema, produto ou serviço de software
Responsáveis:	gerente de projeto e analistas
Critério de inicialização:	o cliente apresentou um problema para o qual deseja uma solução computacional
Elementos de entrada:	descrição informal, tanto do problema quanto dos requisitos do sistema a ser desenvolvido (informação externa) (28)
Tarefas:	<ol style="list-style-type: none"> 1. definir, mais formalmente, o problema 2. fazer um levantamento inicial dos requisitos do sistema, incluindo-se aspectos de segurança, confiabilidade e desempenho 3. determinar o escopo do sistema e as características do ambiente de operação 4. aprovar esse levantamento inicial – tarefa do gerente de projeto
Elementos de saída:	levantamento inicial dos requisitos (17)
Critério de finalização:	o levantamento foi estruturado em conformidade com a vontade do cliente, obtendo a aprovação do gerente de projeto
Métricas registradas:	<ul style="list-style-type: none"> • tempo gasto na identificação das necessidades iniciais

Atividade:	avaliar a viabilidade do projeto (Def.II)
Objetivo:	fazer um levantamento da viabilidade do projeto a ser desenvolvido, através da identificação dos recursos humanos, técnicos e econômicos necessários
Responsáveis:	gerente de projeto
Critério de inicialização:	as necessidades iniciais do cliente e do sistema foram captadas
Elementos de entrada:	levantamento inicial dos requisitos (Def.I) (17) + informações sobre os recursos humanos, técnicos e econômicos disponíveis na empresa (Capac.I) (14) + relatórios sobre os problemas e/ou não-conformidades resolvidos, quando existirem (RProb.II) (29)
Tarefas:	<ol style="list-style-type: none"> 1. identificar, dentre os recursos disponíveis, aqueles necessários ao projeto 2. analisar os riscos, custos e benefícios de possíveis opções de desenvolvimento, de acordo com os recursos disponíveis 3. encaminhar, ao processo de <i>Resolução de Problemas</i>, problemas e/ou não-conformidades que não puderem ser resolvidos no escopo do processo de <i>Definição</i> 4. celebrar o contrato com o cliente 5. documentar os resultados
Elementos de saída:	relatório da viabilidade do projeto (18) + problemas e/ou não-conformidades a serem tratados, quando existirem (RProb.II) (27)
Critério de finalização:	a viabilidade do projeto foi avaliada
Métricas registradas:	<ul style="list-style-type: none"> • % dos recursos necessários ao projeto e que estão disponíveis na empresa

Atividade:	preparar para o processo de <i>Planejamento</i> (Plan.I)
Objetivo:	estabelecer uma estrutura adequada para a realização das atividades relacionadas ao processo de <i>Planejamento</i>
Responsável:	gerente de projeto e administrador de recursos
Critério de inicialização:	reuniões para conscientização e esclarecimento sobre a importância do processo de <i>Planejamento</i> já foram efetuadas, segundo o processo de <i>Capacitação</i>
Elementos de entrada:	levantamento inicial dos requisitos (Def.I) (17) + relatório da viabilidade do projeto (Def.II) (18)
Tarefas:	<ol style="list-style-type: none"> 1. determinar a metodologia de trabalho a ser seguida 2. identificar ferramentas de apoio às tarefas de planejamento, com base no processo de <i>Infra-Estrutura</i> 3. identificar a necessidade de treinamento para as atividades do processo de <i>Planejamento</i>, em conformidade com o estabelecido no processo de <i>Capacitação</i>, preparando as diretrizes para o mesmo, se ele for necessário 4. disponibilizar a "base de dados históricos", como definido no processo de <i>Infra-Estrutura</i> 5. documentar os resultados
Elementos de saída:	Plano de Projeto de Software – seção inicial (31A) + diretrizes de treinamento para as atividades do processo de <i>Planejamento</i> , quando necessário (08)
Critério de finalização:	os resultados foram devidamente documentados
Métricas registradas:	<ul style="list-style-type: none"> • % tempo gasto em cada tarefa de preparação

Atividade:	refinar o levantamento inicial (Plan.II)
Objetivo:	refinar o levantamento inicial dos requisitos, para o desenvolvimento do Plano de Projeto de Software
Responsáveis:	gerente de projeto e analistas
Critério de inicialização:	o levantamento inicial dos requisitos e o relatório da viabilidade do projeto já foram concluídos
Elementos de entrada:	levantamento inicial dos requisitos (Def.I) (17) + relatório da viabilidade do projeto (Def.II) (18)
Tarefas:	<ol style="list-style-type: none"> 1. alocar as atividades do processo aos profissionais, com base nos resultados dos processos de <i>Capacitação</i> e <i>Treinamento</i> – tarefa do gerente de projeto 2. analisar o problema a ser tratado e os requisitos iniciais 3. definir as características, os usuários e as interfaces do sistema proposto 4. fazer a modelagem inicial do sistema, utilizando o método de desenvolvimento adequado 5. avaliar a consistência e a conformidade com o levantamento inicial 6. documentar os resultados 7. aprovar os resultados – tarefa do gerente de projeto
Elementos de saída:	especificação preliminar dos requisitos (19)
Critério de finalização:	os refinamentos foram aprovados pelo gerente de projeto e pelo cliente
Métricas registradas:	<ul style="list-style-type: none"> • tempo gasto no refinamento do levantamento inicial • # consultas ao cliente até a completa aprovação da especificação preliminar dos requisitos

Atividade:	estabelecer o modelo de desenvolvimento de software adequado ao projeto (Plan.III)
Objetivo:	definir o modelo de desenvolvimento de software adequado ao projeto, de acordo com as necessidades do cliente
Responsáveis:	gerente de projeto e analistas
Critério de inicialização:	o levantamento inicial dos requisitos e o relatório da viabilidade do projeto foram concluídos
Elementos de entrada:	levantamento inicial dos requisitos (Def.I) (17) + relatório da viabilidade do projeto (Def.II) (18)
Tarefas:	<ol style="list-style-type: none"> 1. determinar os componentes do sistema 2. classificar os componentes de acordo com o conhecimento e recursos técnicos necessários para sua construção; 3. identificar um modelo de desenvolvimento apropriado para o projeto, segundo as características identificadas anteriormente 4. documentar os resultados 5. aprovar os resultados – tarefa do gerente de projeto
Elementos de saída:	Plano de Projeto de Software – seção <u>Modelo de desenvolvimento de software</u> (31B)
Critério de finalização:	o modelo de desenvolvimento foi proposto e aprovado pelo gerente de projeto
Métricas registradas:	não identificado

Atividade:	identificar os riscos do projeto (Plan.IV)
Objetivo:	fazer uma análise dos riscos envolvidos no projeto
Responsáveis:	gerente de projeto e analistas
Critério de inicialização:	o levantamento inicial dos requisitos e o relatório de viabilidade do projeto foram concluídos
Elementos de entrada:	levantamento inicial dos requisitos (Def.I) (17) + relatório da viabilidade do projeto (Def.II) (18)
Tarefas:	<ol style="list-style-type: none"> 1. identificar os riscos envolvidos no projeto 2. fazer uma estimativa dos riscos 3. definir opções para evitar os riscos 4. definir procedimentos para a monitoração dos riscos 5. documentar os resultados 6. aprovar os resultados – tarefa do gerente de projeto
Elementos de saída:	Plano de Projeto de Software – seção <u>Análise dos riscos do projeto</u> (31C)
Critério de finalização:	a análise dos riscos do projeto foi devidamente aprovada pelo gerente de projeto
Métricas registradas:	não identificado

Atividade:	preparar as estimativas para o projeto (Plan.V)
Objetivo:	elaborar as estimativas do projeto, tanto de prazo quanto de custo e esforço
Responsáveis:	gerente de projeto e analistas
Critério de inicialização:	a especificação preliminar dos requisitos já foi concluída
Elementos de entrada:	especificação preliminar dos requisitos (Plan.II) (19) + seção <u>Modelo de desenvolvimento de software</u> do Plano de Projeto de Software (Plan.III) (31B)
Tarefas:	<ol style="list-style-type: none"> 1. definir o método de estimativas adequado 2. elaborar as estimativas de prazo 3. elaborar as estimativas de esforço e custo 4. documentar as estimativas 5. aprovar as estimativas – tarefa do gerente de projeto
Elementos de saída:	Plano de Projeto de Software – seção <u>Estimativas de projeto</u> (31D)
Critério de finalização:	as estimativas de projeto foram devidamente aprovadas pelo gerente de projeto
Métricas registradas:	não identificado

Atividade:	realizar o planejamento das atividades de apoio ao desenvolvimento do projeto (Plan.VI)
Objetivo:	planejar as atividades dos processos de apoio a serem empregadas no projeto
Responsáveis:	gerente de projeto e analistas
Critério de inicialização:	o levantamento inicial dos requisitos e o relatório de viabilidade do projeto já foram concluídos
Elementos de entrada:	especificação preliminar dos requisitos (Plan.II) (19) + seção <u>Modelo de desenvolvimento de software</u> do Plano de Projeto de Software (Plan.III) (31B)
Tarefas:	<ol style="list-style-type: none"> 1. fazer um levantamento dos produtos de software e das atividades necessárias para o estabelecimento e manutenção do controle do projeto, em conjunto com o processo de <i>Acompanhamento</i> 2. estabelecer os pontos de revisão e os dados a serem coletados nesses pontos de revisão, também em conjunto com o processo de <i>Acompanhamento</i> 3. fazer o planejamento da qualidade, com base no estabelecido pelo processo de <i>Controle de Qualidade</i> 4. planejar a operação e a manutenção do sistema 5. documentar os resultados 6. aprovar os resultados – tarefa do gerente de projeto
Elementos de saída:	Plano de Projeto de Software – seção final (31E)
Critério de finalização:	os resultados dessa atividade foram devidamente aprovados pelo gerente de projeto
Métricas registradas:	não identificado

Atividade:	iniciar o processo de <i>Desenvolvimento</i> (Desenv.I)
Objetivo:	fornecer um ambiente integrado de métodos, recursos e ferramentas, essencial ao desenvolvimento dos projetos da empresa
Responsáveis:	gerente de projeto, analistas e projetistas
Critério de inicialização:	a especificação preliminar dos requisitos já foi concluída
Elementos de entrada:	especificação preliminar dos requisitos (Plan.II) (19) + seção <u>Modelo de desenvolvimento de software</u> do Plano de Projeto de Software (Plan.III) (31B) + informações sobre as normas, métodos, padrões, procedimentos técnicos, linguagens de programação e ferramentas, com base no negócio da empresa (Capac.I) (05)
Tarefas:	<ol style="list-style-type: none"> 1. identificar ferramentas de apoio às tarefas de desenvolvimento, com base no processo de <i>Infra-Estrutura</i> 2. desenvolver um plano para a condução das atividades, de acordo com as diretrizes estabelecidas para os requisitos de software e com base nas informações de entrada da atividade, que deve incluir, sobretudo, um cronograma de desenvolvimento 3. identificar a necessidade de treinamento para as atividades do processo, em conformidade com o estabelecido no processo de <i>Capacitação</i>, preparando as diretrizes para o mesmo, se ele for necessário 4. documentar os resultados
Elementos de saída:	plano para a condução das atividades de desenvolvimento do sistema (32) + diretrizes de treinamento para as atividades do processo de <i>Desenvolvimento</i> , quando necessário (08)
Critério de finalização:	os planos para a condução das atividades de desenvolvimento do sistema foram iniciados
Métricas registradas:	não identificado

Atividade:	especificar os requisitos (Desenv.II)
Objetivo:	definir, da forma mais completa e menos ambígua possível, os requisitos do sistema, funcionais e não-funcionais
Responsáveis:	gerente de projeto e analistas
Critério de inicialização:	a especificação preliminar dos requisitos já foi concluída
Elementos de entrada:	especificação preliminar dos requisitos (Plan.II) (19)
Tarefas:	<ol style="list-style-type: none"> 1. alocar as atividades do processo aos profissionais, com base nos resultados dos processos de <i>Capacitação</i> e <i>Treinamento</i> – tarefa do gerente de projeto <p>Em conformidade com o estabelecido no processo de <i>Gerenciamento de Configuração</i>:</p> <ol style="list-style-type: none"> 2. estabelecer os requisitos do projeto e do sistema 3. identificar os requisitos do sistema que devem ser alocados ao projeto, bem como as restrições de projeto 4. estabelecer, analisar e refinar os requisitos do software (descrição funcional, de controle e comportamental do sistema), de acordo com o processo de <i>Verificação</i> 5. estabelecer os critérios de validação, em conjunto com o processo de <i>Validação</i> 6. explicitar os requisitos de qualidade, respeitando-se o estabelecido no processo de <i>Controle de Qualidade</i> 7. documentar a especificação 8. aprovar os resultados – tarefa do gerente de projeto
Elementos de saída:	Especificação de Requisitos (33)
Critério de finalização:	a Especificação de Requisitos foi concluída e aprovada pelo gerente de projeto
Métricas registradas:	não identificado

Atividade:	desenvolver o projeto de software (Desenv.III)
Objetivo:	estabelecer um projeto de software adequado aos requisitos especificados, de maneira a poder transformar esses requisitos em componentes e esses em unidades mais baixas, possíveis de serem codificadas, compiladas e testadas
Responsáveis:	gerente de projeto e projetistas
Critério de inicialização:	a Especificação de Requisitos já foi concluída
Elementos de entrada:	Especificação de Requisitos (Desenv.II) (33) + Plano de Projeto de Software (A.F.) (31)
Tarefas:	<p>Em conformidade com o estabelecido no processo de <i>Gerenciamento de Configuração</i>:</p> <ol style="list-style-type: none"> 1. definir a estrutura de cada componente, identificando os itens de hardware e software necessários 2. possibilitar reuso 3. verificar se o projeto arquitetural definido atende à Especificação de Requisitos previamente estabelecida, de acordo com o processo de <i>Verificação</i> 4. refinar os componentes individuais de software até o nível de interfaces, classes ou componentes pré-existent 5. definir bases de dados, linguagens de programação, padrões de integração e estilos de interface 6. definir os requisitos de teste de unidade e de integração 7. documentar os resultados 8. aprovar o projeto – tarefa do gerente de projeto
Elementos de saída:	projeto arquitetural do software e projeto detalhado do software (34)
Critério de finalização:	a fase de projeto foi devidamente concluída
Métricas registradas:	<ul style="list-style-type: none"> • # arquivos de banco de dados • # total de componentes do projeto • # componentes de interface / # total de componentes do sistema • # componentes projetados para serem reutilizados / # total de componentes do sistema • # componentes dependentes de hardware / # total de componentes do sistema • % cobertura dos requisitos alocados nos itens presentes na arquitetura de cada componente

Atividade:	implementar o sistema (Desenv.IV)
Objetivo:	transformar os componentes do sistema em elementos executáveis
Responsáveis:	gerente de projeto, projetistas e programadores
Critério de inicialização:	os componentes foram projetados e já podem ser codificados
Elementos de entrada:	projeto arquitetural do software e projeto detalhado do software (Desenv.III) (34)
Tarefas:	<p>Em conformidade com o estabelecido pelo processo de <i>Gerenciamento de Configuração</i>:</p> <ol style="list-style-type: none"> 1. desenvolver o código para cada componente projetado 2. documentar o código, com base no estabelecido pelo processo de <i>Documentação</i> 3. iniciar a elaboração do Manual do Usuário
Elementos de saída:	código-fonte do sistema (35A)
Critério de finalização:	os componentes do sistema foram implementados
Métricas registradas:	<ul style="list-style-type: none"> • densidade de comentários de cada módulo (# linhas de comentário / # total de linhas do módulo) • densidade de comentários do sistema (# linhas de comentário do sistema / # total de linhas do sistema) • fan-in e fan-out (módulos e/ou procedimentos) • # procedimentos de interface / # total procedimentos do sistema • # linhas de comentário de cada módulo • # linhas de comentário do sistema • # módulos dependentes de hardware • # total de módulos do sistema • # total de opções do menu do sistema • # procedimentos de interface

Atividade:	realizar os testes individuais (Desenv.V)
Objetivo:	detectar e corrigir falhas nos módulos e nos procedimentos, de modo a verificar se os requisitos especificados para o software estão sendo corretamente implementados
Responsáveis:	gerente de projeto, projetistas e programadores
Critério de inicialização:	os componentes do sistema já foram implementados
Elementos de entrada:	código-fonte do sistema (Desenv.IV) (35A) + relatórios sobre os problemas e/ou não-conformidades resolvidos, quando existirem (RProb.II) (29)
Tarefas:	<p>Em conformidade com o estabelecido pelo processo de <i>Gerenciamento de Configuração</i>:</p> <ol style="list-style-type: none"> 1. estabelecer planos e procedimentos, casos e critérios de teste, a fim de que seja testado cada um dos componentes do sistema 2. realizar os testes, verificando se cada componente testado tem satisfeitos os requisitos de teste definidos anteriormente e com base no processo de <i>Verificação</i> 3. corrigir eventuais falhas evidenciadas pelos testes individuais 4. encaminhar, ao processo de <i>Resolução de Problemas</i>, problemas e/ou não-conformidades que não puderem ser resolvidos no escopo do processo de <i>Desenvolvimento</i> 5. documentar os resultados
Elementos de saída:	código-fonte do sistema (com os testes de unidade devidamente realizados) (35A*) + problemas e/ou não-conformidades a serem tratados, quando existirem (27)
Critério de finalização:	os componentes do sistema foram devidamente testados e falhas eventuais, corrigidas
Métricas registradas:	<ul style="list-style-type: none"> • # drivers utilizados • # stubs utilizados • # falhas encontradas / # total de componentes do sistema • # casos de teste utilizados / # total de componentes do sistema

Atividade:	integrar e testar o sistema (Desenv.VI)
Objetivo:	detectar e corrigir falhas resultantes da integração dos módulos, garantindo que os requisitos especificados para o software tenham sido corretamente implementados
Responsáveis:	gerente de projeto, projetistas e programadores
Critério de inicialização:	os testes individuais foram concluídos e os componentes estão prontos para serem integrados
Elementos de entrada:	código-fonte do sistema (com os testes de unidade devidamente realizados) (Desenv.V) (35A*) + relatórios sobre os problemas e/ou não-conformidades resolvidos, quando existirem (RProb.II) (29)
Tarefas:	<p>Em conformidade com o estabelecido pelo processo de <i>Gerenciamento de Configuração</i>:</p> <ol style="list-style-type: none"> 1. elaborar um plano para integrar os componentes de software, definindo casos e critérios de teste, bem como os procedimentos necessários 2. integrar os componentes, de acordo com o plano 3. testar o comportamento do software integrado, verificando se os requisitos definidos anteriormente são satisfeitos, com base no processo de <i>Verificação</i> 4. corrigir eventuais falhas evidenciadas pelos testes de integração 5. finalizar o Manual do Usuário, também verificando-o com base no processo de <i>Verificação</i> 6. validar o sistema, avaliando o projeto, código, testes, resultados dos testes e documentação, de acordo com o que foi definido no processo de <i>Validação</i> 7. gerar o código executável do sistema 8. gerar um relatório propondo futuras melhorias, para futuras versões 9. encaminhar, ao processo de <i>Resolução de Problemas</i>, problemas e/ou não-conformidades que não puderem ser resolvidos no escopo do processo de <i>Desenvolvimento</i> 10. documentar os resultados
Elementos de saída:	código-fonte do sistema (com os testes de integração devidamente realizados) (35A*) + código executável do sistema (35B) + Manual do Usuário (36) + relatório de futuras melhorias do sistema, quando existir (37) + problemas e/ou não-conformidades a serem tratados, quando existirem (27)
Critério de finalização:	o sistema foi devidamente integrado e testado e o Manual do Usuário está concluído
Métricas registradas:	<ul style="list-style-type: none"> • # falhas encontradas • # casos de teste utilizados • # falhas encontradas / # casos de teste utilizados • # atualizações na documentação do software

Atividade:	instalar e entregar o sistema (Desenv.VII)
Objetivo:	preparar o sistema para ser utilizado pelos usuários, no ambiente de operação
Responsáveis:	gerente de projeto e operadores
Critério de inicialização:	o sistema já foi liberado para ser instalado no ambiente de operação
Elementos de entrada:	código executável do sistema (Desenv.VI) (35B) + código fonte, quando especificado no contrato (Desenv.VI) (35A**) + Manual do Usuário (Desenv.VI) (36)
Tarefas:	<ol style="list-style-type: none"> 1. elaborar um plano de instalação do sistema no ambiente de operação 2. disponibilizar toda a documentação necessária para instalação e utilização do sistema, sobretudo o Manual do Usuário 3. introduzir os dados iniciais necessários à execução do sistema 4. documentar os resultados 5. aprovar os resultados – tarefa do gerente de projeto
Elementos de saída:	sistema instalado (38)
Critério de finalização:	o sistema foi devidamente instalado no ambiente de operação
Métricas registradas:	<ul style="list-style-type: none"> • tempo necessário para a instalação do sistema • # problemas encontrados na instalação

Atividade:	preparar para o processo de <i>Operação</i> (Oper.I)
Objetivo:	estabelecer uma estrutura organizacional para a realização das tarefas associadas ao processo de <i>Operação</i>
Responsáveis:	gerente de projeto e operadores
Critério de inicialização:	o sistema já foi instalado no ambiente de operação
Elementos de entrada:	Manual do Usuário (Desenv.VI) (36) + características do ambiente de operação, constantes do levantamento inicial dos requisitos (Def.I) (17)
Tarefas:	<ol style="list-style-type: none"> 1. identificar ferramentas de apoio às tarefas de operação, de acordo com o processo de <i>Infra-Estrutura</i> 2. desenvolver um plano para a condução das atividades relacionadas à operação e ao suporte aos usuários, com base no estabelecido pelo processo de <i>Planejamento</i> 3. estabelecer critérios para o teste operacional, para inserir pedidos de modificação no processo de <i>Manutenção</i> e para liberar o produto para uso operacional 4. identificar a necessidade de treinamento para as atividades do processo, preparando as diretrizes para o mesmo, se ele for necessário, com conformidade com o estabelecido no processo de <i>Capacitação</i> 5. documentar os resultados 6. aprovar os resultados – tarefa do gerente de projeto
Elementos de saída:	plano de operação e suporte aos usuários (41) + diretrizes de treinamento para as atividades do processo de <i>Operação</i> , quando necessário (08)
Critério de finalização:	as diretrizes para as tarefas de operação e suporte foram estabelecidas
Métricas registradas:	<ul style="list-style-type: none"> • % tempo gasto em cada tarefa de preparação

Atividade:	cuidar dos testes operacionais e da operação do sistema (Oper.II)
Objetivo:	liberar o sistema para uso, após ter passado pelos testes operacionais e fornecer a assistência necessária ao usuário para operação do sistema
Responsáveis:	operadores
Critério de inicialização:	as diretrizes de operação e suporte (critérios, procedimentos e padrões) já foram definidas
Elementos de entrada:	sistema instalado (Desenv.VII) (38) + plano de operação e suporte aos usuários (Oper.I) (41) + relatório sobre os problemas e/ou não-conformidades resolvidos, quando existirem (RProb.II) (29)
Tarefas:	<ol style="list-style-type: none"> 1. alocar as atividades do processo aos profissionais, com base nos resultados dos processos de <i>Capacitação e Treinamento</i> – tarefa do gerente de projeto 2. executar o teste operacional, respeitando-se o disposto no Manual do Usuário, em conjunto com o processo de <i>Validação</i> 3. avaliar os testes (resultados obtidos X resultados esperados), também em conjunto com o processo de <i>Validação</i> 4. liberar o sistema para uso, tendo sido, os testes, satisfatórios 5. operar o sistema no ambiente para o qual foi desenvolvido, em conjunto com o usuário, fornecendo a ele o auxílio necessário nas atividades iniciais de operação 6. documentar os resultados 7. encaminhar para o processo de <i>Resolução de Problemas</i> os problemas e/ou não-conformidades que não puderem ser resolvidos no escopo do processo de <i>Operação</i>
Elementos de saída:	sistema liberado para uso (39) + problemas e/ou não-conformidades a serem tratados, quando existirem (27)
Critério de finalização:	o sistema foi liberado para utilização dos usuários
Métricas registradas:	<ul style="list-style-type: none"> • # testes operacionais realizados até a liberação do sistema para uso • intervalo médio de tempo entre os testes operacionais

Atividade:	fornecer assistência e/ou suporte aos usuários (Oper.III)
Objetivo:	garantir assistência técnica e/ou suporte aos usuários, encaminhando pedidos de manutenção ao processo de <i>Manutenção</i> , nos casos em que ela for necessária
Responsáveis:	gerente de projeto, operadores e mantenedores
Critério de inicialização:	uma solicitação de assistência e/ou suporte foi efetuada pelo usuário
Elementos de entrada:	pedido (informal) de assistência e/ou suporte (informação externa) (40)
Tarefas:	<ol style="list-style-type: none"> 1. formalizar o pedido, seguindo-se o estabelecido no processo de <i>Documentação</i> 2. fornecer assistência, suporte e consultoria aos usuários, documentando suas dúvidas e dificuldades, e solucionando os problemas 3. encaminhar, ao processo de <i>Manutenção</i>, pedidos de manutenção, nos casos necessários, ou possíveis melhorias a serem realizadas 4. monitorar a execução dos pedidos de manutenção 5. fornecer retorno aos solicitantes 6. aplicar os instrumentos de avaliação do nível de satisfação dos usuários (com relação ao sistema e aos serviços da empresa) definidos no processo de <i>Melhoria</i> 7. documentar as tarefas 8. aprovar os resultados – tarefa do gerente de projeto
Elementos de saída:	relatório referente ao pedido de assistência e/ou suporte, a ser encaminhado ao usuário (42) + pedido de manutenção, nos casos em que este for necessário (43)
Critério de finalização:	a dificuldade do usuário foi sanada de forma satisfatória
Métricas registradas:	<ul style="list-style-type: none"> • tempo médio de atendimento dos pedidos de assistência e/ou suporte • # visitas ao ambiente de operação para assistência e/ou suporte • # contatos com o usuário (desconsiderando-se visitas ao ambiente de operação) até a resolução do problema • # pedidos de assistência e/ou suporte efetuados • # pedidos de assistência e/ou suporte encaminhados ao processo de <i>Manutenção</i> • # pedidos de assistência e/ou suporte efetuados / unidade de tempo • evolução do nível de satisfação do usuário

Atividade:	preparar para o processo de <i>Manutenção</i> (Manut.I)
Objetivo:	estabelecer uma estrutura organizacional adequada para a realização das atividades relacionadas ao processo de <i>Manutenção</i>
Responsáveis:	gerente de projeto e mantenedores
Critério de inicialização:	o sistema está pronto para ser instalado e entregue
Elementos de entrada:	relatório de futuras melhorias no sistema, se existir (Desenv.VI) (41)
Tarefas:	<ol style="list-style-type: none"> 1. identificar ferramentas de apoio às tarefas de manutenção, de acordo com o processo de <i>Infra-Estrutura</i> 2. desenvolver planos e procedimentos para conduzir as atividades e tarefas de manutenção, incluindo-se os casos de migração e descontinuação, com base no estabelecido pelo processo de <i>Planejamento</i> 3. estabelecer procedimentos para receber, armazenar e acompanhar os pedidos de modificação e os relatórios de problemas dos usuários 4. identificar a necessidade de treinamento para as atividades do processo, preparando as diretrizes para o mesmo, se ele for necessário, com conformidade com o estabelecido no processo de <i>Capacitação</i> 5. documentar os planos e procedimentos 6. aprovar os resultados – tarefa do gerente de projeto
Elementos de saída:	planos e procedimentos para manutenção (44) + diretrizes de treinamento para as atividades do processo de <i>Manutenção</i> , quando necessário (08)
Critério de finalização:	os planos e procedimentos foram concluídos e aprovados pelo gerente de projeto
Métricas registradas:	<ul style="list-style-type: none"> • % tempo gasto em cada tarefa de preparação

Atividade:	fazer a análise do problema (Manut.II)
Objetivo:	analisar o pedido de manutenção proveniente do processo de <i>Operação</i> , tomando as medidas necessárias
Responsáveis:	gerente de projeto e mantenedores
Critério de inicialização:	o sistema apresentou um problema, o qual gerou um pedido de manutenção, proveniente do processo de <i>Operação</i>
Elementos de entrada:	planos e procedimentos para manutenção (Manut.I) (44) + pedido de manutenção (Op.III) (43)
Tarefas:	<ol style="list-style-type: none"> 1. alocar as atividades do processo aos profissionais, com base nos resultados dos processos de <i>Capacitação</i> e <i>Treinamento</i> – tarefa do gerente de projeto 2. avaliar o pedido de modificação, considerando o tipo de manutenção a ser realizada, o alcance da alteração e as consequências de tal mudança 3. desenvolver opções para a implementação da modificação 4. documentar os resultados da análise 5. tomar a decisão sobre o pedido de alteração – tarefa do gerente de projeto <p>Para pedidos aprovados:</p> <ul style="list-style-type: none"> • informar a prioridade de execução da alteração • emitir a "ordem de serviço"
Elementos de saída:	"ordem de serviço" referente à manutenção solicitada (45)
Critério de finalização:	o parecer do gerente de projeto, a respeito do pedido de manutenção, foi expedido
Métricas registradas:	<ul style="list-style-type: none"> • # pedidos aprovados / # pedidos solicitados • # pedidos / tipo de manutenção • tempo médio de avaliação de um pedido

Atividade:	implementar a modificação (Manut.III)
Objetivo:	realizar a(s) modificação(ões) necessária(s) no software, de acordo com a "ordem de serviço" recebida
Responsáveis:	gerente de projeto e mantenedores
Critério de inicialização:	uma "ordem de serviço" para manutenção do sistema foi expedida pelo gerente de projeto
Elementos de entrada:	sistema liberado para uso (Oper.II) (39) + "ordem de serviço", referente à manutenção solicitada (Manut.II) (45) + planos e procedimentos para manutenção (44)
Tarefas:	<ol style="list-style-type: none"> 1. determinar e documentar quais unidades de software, versões e documentação relacionada precisam ser modificadas 2. implementar a modificação, respeitando-se os processos de <i>Desenvolvimento</i> e <i>Operação</i> 3. efetuar testes de regressão nas partes já existentes, para revalidar o sistema, de acordo com o processo de <i>Validação</i> 4. executar, quando necessário e seguindo-se o estabelecido nos planos e procedimentos para manutenção, a migração e/ou a descontinuação do software 5. documentar os resultados 6. aprovar as modificações – tarefa do gerente de projeto
Elementos de saída:	Sistema liberado para uso (39)
Critério de finalização:	as modificações foram efetuadas e aprovadas pelo gerente de projeto
Métricas registradas:	<ul style="list-style-type: none"> • # alterações efetuadas no sistema • # módulos do sistema • # total de procedimentos do sistema • tempo médio de implementação de uma modificação • custo envolvido na modificação • tamanho atualizado do sistema (em linhas de código e/ou linhas de código executável)

Atividade:	preparar para o processo de <i>Documentação</i> (Doc.I)
Objetivo:	estabelecer uma estrutura organizacional adequada para a realização das atividades relacionadas ao processo de <i>Documentação</i>
Responsáveis:	gerente de projeto, analistas e projetistas
Critério de inicialização:	um projeto de software foi iniciado, ou seja, começou uma instância do processo de <i>Definição</i>
Elementos de entrada:	informações sobre normas, métodos, padrões, procedimentos técnicos, linguagens de programação e ferramentas, com base no negócio da empresa (Capac.I) (05)
Tarefas:	<ol style="list-style-type: none"> 1. identificar os requisitos de documentação do projeto, em conformidade com o processo de <i>Definição</i> 2. identificar os documentos a serem produzidos, de acordo com seu conteúdo técnico 3. estabelecer a forma de organização – sobretudo com relação à hierarquia – e de identificação dos documentos 4. projetar a estrutura e a apresentação dos documentos 5. aprovar os resultados, compilando-os em diretrizes para produção dos documentos necessários ao longo do projeto – tarefa do gerente de projeto 6. cuidar para que todos os demais processos da empresa tenham conhecimento dessas diretrizes e possam utilizá-las
Elementos de saída:	diretrizes de produção dos documentos necessários ao longo do projeto (15)
Critério de finalização:	as diretrizes preliminares de produção dos documentos foram aprovadas pelo gerente de projeto
Métricas registradas:	<ul style="list-style-type: none"> • % tempo gasto em cada tarefa de preparação • # documentos identificados

Atividade:	gerenciar a produção dos documentos (Doc.II)
Objetivo:	gerenciar a produção e cuidar do armazenamento dos documentos a serem produzidos e utilizados ao longo do projeto
Responsáveis:	gerente de projeto, analistas e projetistas
Critério de inicialização:	as diretrizes de produção e distribuição dos documentos foram aprovadas pelo gerente de projeto
Elementos de entrada:	documentos a serem armazenados/rearmazenados (provenientes de alguma atividade que os gerou) (16)
Tarefas:	<ol style="list-style-type: none"> 1. identificar os documentos que devem ser armazenados na "base de dados históricos" 2. estabelecer os procedimentos para o armazenamento desses documentos 3. identificar o padrão de documentação interna e externa do código-fonte, de acordo com a linguagem de programação a ser utilizada 4. armazenar/rearmazenar na "base de dados históricos" ou eliminar dela os documentos apropriados, de acordo com os procedimentos definidos e com base no processo de <i>Verificação</i>
Elementos de saída:	documentos a serem armazenados/rearmazenados (provenientes de alguma atividade que os gerou) (16)
Critério de finalização:	um projeto foi concluído
Métricas registradas:	<ul style="list-style-type: none"> • profundidade de documentação do projeto (número máximo de documentos diretamente originados de um documento) • profundidade de documentação de cada processo • # documentos produzidos no projeto • # documentos modificados no projeto • # atual de documentos produzidos por processo • # atual de documentos modificados por processo

Atividade:	preparar para o processo de <i>Acompanhamento</i> (Acomp.I)
Objetivo:	estabelecer uma estrutura organizacional adequada para a realização das atividades relacionadas ao processo de <i>Acompanhamento</i>
Responsáveis:	gerente de projeto e projetistas
Critério de inicialização:	uma instância do processo de <i>Planejamento</i> teve início
Elementos de entrada:	Plano de Projeto de Software (A.F.I) (31) + plano para a condução das atividades de desenvolvimento do sistema (Desenv.I) (32)
Tarefas:	<ol style="list-style-type: none"> 1. elaborar um plano de acompanhamento que identifique itens a serem revistos, pontos de revisão e recursos necessários para se realizar as revisões, sobretudo os técnicos 2. definir procedimentos, associadas aos produtos e às atividades, para o estabelecimento e manutenção do controle do projeto 3. identificar ferramentas de apoio às tarefas de acompanhamento, com base no processo de <i>Infra-Estrutura</i> 4. documentar os planos e procedimentos 5. aprovar os resultados – tarefa do gerente de projeto
Elementos de saída:	plano de acompanhamento (30)
Critério de finalização:	o plano de acompanhamento foi concluído
Métricas registradas:	<ul style="list-style-type: none"> • % tempo gasto em cada tarefa de preparação

Atividade:	acompanhar o projeto de software (Acomp.II)
Objetivo:	garantir que o projeto de software em desenvolvimento está de acordo com o que foi inicialmente planejado, tomando as atitudes corretivas nos casos em que haja algum desvio significativo dos planos
Responsáveis:	gerente de projeto e projetistas
Critério de inicialização:	o plano de acompanhamento já foi concluído
Elementos de entrada:	plano de acompanhamento (Acomp.I) (30) + Plano de Projeto de Software (A.F.) (31)
Tarefas:	<p>Seguindo-se as atividades do processo de <i>Desenvolvimento</i>:</p> <ol style="list-style-type: none"> 1. coletar as métricas necessárias, em conjunto com o processo de <i>Controle de Qualidade</i> 2. avaliar os custos e riscos envolvidos no projeto, os recursos e esforços necessários ao seu desenvolvimento, e o seu cronograma (revisões gerenciais) 3. avaliar os produtos e serviços, a fim de verificar se estão completos, respeitam os padrões e especificações e estão prontos para se poder executar a próxima atividade para a qual servem de entrada (revisões técnicas), com o auxílio do processo de <i>Controle de Qualidade</i> 4. tomar atitudes corretivas, quando necessário e possível, alterando a forma como o trabalho está sendo feito, sem prejuízo para o projeto e para o curso de qualquer outro processo, e desde que com a ciência do processo de <i>Gerência</i>
Elementos de saída:	não identificado
Critério de finalização:	atitudes corretivas foram tomadas, com a ciência do processo de <i>Gerência</i>
Métricas registradas:	<ul style="list-style-type: none"> • # ações corretivas tomadas • % de desvio nas estimativas de custo e cronograma (cumulativo) • % de desvio nas estimativas de tamanho (cumulativo) • % de desvio nas estimativas de esforço (cumulativo) • evolução dos riscos • # relatórios de progresso / processo • # total de relatórios de progresso gerados no projeto

Atividade:	preparar para o processo de <i>Gerenciamento de Configuração</i> (GConfig.I)
Objetivo:	estabelecer uma estrutura organizacional adequada para a realização das atividades relacionadas ao processo de <i>Gerenciamento de Configuração</i>
Responsáveis:	gerente de projeto e gerente de versões
Critério de inicialização:	uma instância do processo de <i>Planejamento</i> teve início
Elementos de entrada:	Plano de Projeto de Software (A.F.) (31)
Tarefas:	<ol style="list-style-type: none"> 1. determinar ferramentas, padrões e procedimentos a serem utilizados, sobretudo quanto ao repositório de software (funcionamento e acesso), de acordo com o processo de <i>Infra-Estrutura</i> 2. identificar a necessidade de treinamento para as atividades do processo de <i>Gerenciamento de Configuração</i>, em conformidade com o processo de <i>Capacitação</i>, preparando as diretrizes para o mesmo, se ele for necessário 3. documentar os resultados, a serem reunidos num plano de gerenciamento de configuração 4. aprovar o plano – tarefa do gerente de projeto
Elementos de saída:	plano de gerenciamento de configuração de software (46) + diretrizes de treinamento para as atividades do processo de <i>Gerenciamento de Configuração</i> , quando necessário (08)
Critério de finalização:	o plano foi aprovado pelo gerente de projeto de software
Métricas registradas:	<ul style="list-style-type: none"> • % tempo gasto em cada tarefa de preparação

Atividade:	selecionar os itens de configuração a serem controlados (GConfig.II)
Objetivo:	definir os itens de software (e suas versões) a serem controlados durante o projeto
Responsáveis:	gerente de projeto e gerente de versões
Critério de inicialização:	tanto o Plano de Projeto de Software quanto o plano de gerenciamento de configuração de software foram concluídos e aprovados
Elementos de entrada:	plano de gerenciamento de configuração (GConfig.I) (46) + Plano de Projeto de Software (A.F.) (31)
Tarefas:	<ol style="list-style-type: none"> 1. alocar as atividades aos profissionais, de acordo com o estabelecido nos processos de <i>Capacitação e Treinamento</i> Seguindo-se o processo de <i>Desenvolvimento</i>; 2. definir os itens de configuração relevantes ao projeto que serão efetivamente colocados sob controle de configuração 3. descrever as relações entre os itens de configuração selecionados 4. definir uma forma (ferramenta, por exemplo) para identificar, unicamente, cada item de configuração (nome, tipo, versão, etc.) 5. estabelecer as <i>linhas de referência (baselines)</i> 6. documentar cada item de configuração, descrevendo a maneira como eles serão arquivados e recuperados do repositório de software, de acordo com o estabelecido no processo de <i>Documentação</i> 7. aprovar os resultados – tarefa do gerente de projeto
Elementos de saída:	não identificado
Critério de finalização:	a aprovação do gerente de projeto foi dada
Métricas registradas:	<ul style="list-style-type: none"> • # itens colocados sob controle / # itens possíveis

Atividade:	aprovar/rejeitar um pedido de alteração (GConfig.III)
Objetivo:	utilizar mecanismos e procedimentos já definidos para identificação e registro de pedidos de mudanças, e conseqüente análise e avaliação dessas mudanças
Responsáveis:	gerente de projeto
Critério de inicialização:	o item de configuração a ser modificado já passou por uma <i>linha de referência</i>
Elementos de entrada:	pedido de alteração de um item que já passou por uma <i>linha de referência (47)</i> , devidamente justificado, proveniente de alguma atividade do processo de <i>Desenvolvimento</i> + relatórios sobre os problemas e/ou não-conformidades resolvidos (RProb.II) (29)
Tarefas:	<ol style="list-style-type: none"> 1. avaliar o pedido de alteração 2. relatar os resultados da avaliação 3. tomar a decisão sobre o pedido de alteração Para pedidos aprovados: <ul style="list-style-type: none"> • informar a prioridade de execução do pedido • emitir a "ordem de serviço" <ol style="list-style-type: none"> 4. encaminhar eventuais problemas e/ou não-conformidades que não puderem ser resolvidos no escopo do processo de <i>Gerenciamento de Configuração</i> ao processo de <i>Resolução de Problemas</i>
Elementos de saída:	"ordem de serviço" referente à alteração de um item que já passou por uma <i>linha de referência (48)</i> + problemas e/ou não-conformidades a serem tratados, quando existirem (27)
Critério de finalização:	o parecer do gerente de projeto, a respeito do pedido de alteração, foi expedido
Métricas registradas:	<ul style="list-style-type: none"> • # pedidos aprovados / # pedidos feitos

Atividade:	implementar as alterações (GConfig.IV)
Objetivo:	executar as alterações dos itens de configuração, de acordo com a "ordem de serviço" expedida
Responsáveis:	gerente de versões, projetistas e programadores
Critério de inicialização:	uma "ordem de serviço" para alteração de item de configuração foi expedida pelo gerente de projeto
Elementos de entrada:	"ordem de serviço" referente à alteração de um item que já passou por uma <i>linha de referência (GConfig.V) (48)</i> + item, que já passou por uma <i>linha de referência (49)</i> , a ser alterado, proveniente de alguma atividade do processo de <i>Desenvolvimento</i>
Tarefas:	<ol style="list-style-type: none"> 1. retirar o(s) item(ns) de configuração necessário(s) do repositório e colocá-lo(s) na área de trabalho dos projetistas/programadores (<i>check out</i>) – tarefa do gerente de versões 2. efetuar as alterações, de acordo com a "ordem de serviço" – tarefa dos projetistas/programadores 3. avaliar o aspecto funcional de cada item modificado, tentando descobrir omissões ou erros na configuração que degradam os padrões de construção de software, de acordo com o processo de <i>Controle de Qualidade</i> 4. avaliar se a(s) nova(s) versão(ões) respeita(m) os requisitos definidos anteriormente, também com base no processo de <i>Controle de Qualidade</i> 5. verificar se a nova configuração a ser 'congelada' pela <i>linha de referência</i> está composta da versão mais recente dos itens de configuração determinados para a fase específica do ciclo de vida 6. verificar se o projeto e a documentação estão atualizados, respeitando-se os procedimentos e padrões definidos, de acordo com o processo de <i>Verificação</i>
Elementos de saída:	item, que já passou por uma <i>linha de referência (49)</i> , alterado
Critério de finalização:	as mudanças ocorridas foram devidamente registradas
Métricas registradas:	<ul style="list-style-type: none"> • # pedidos de mudanças processados / unidade de tempo • média de pedidos processados / item de configuração • # total de alterações da nova versão do sistema

Atividade:	relatar a situação da configuração (GConfig.V)
Objetivo:	manter a ordem e a unidade do projeto sendo desenvolvido
Responsáveis:	gerente de versões
Critério de inicialização:	o sistema sofreu alterações na sua configuração
Elementos de entrada:	item, que já passou por uma <i>linha de referência (49)</i> , alterado
Tarefas:	<ol style="list-style-type: none"> 1. armazenar o(s) item(ns) de configuração alterado(s) no repositório, uma vez aprovado(s) na revisão (<i>check in</i>) 2. documentar as alterações ocorridas (o que aconteceu, quem o fez, quando ocorreu, o que mais será afetado, etc.) 3. compilar os resultados num relatório da configuração atual do sistema, a ser armazenado na base de dados históricos, de acordo com o processo de <i>Documentação</i>
Elementos de saída:	item, que já passou por uma <i>linha de referência (49)</i> , alterado, a ser enviado para a atividade do processo de <i>Desenvolvimento</i> que solicitou a alteração + relatório da situação atual de configuração do sistema (50)
Critério de finalização:	o relatório sobre a nova configuração do sistema, após as alterações, foi concluído
Métricas registradas:	<ul style="list-style-type: none"> • # pareceres favoráveis / # total de pareceres

Atividade:	preparar para o processo de <i>Controle de Qualidade</i> (CQual.I)
Objetivo:	estabelecer uma estrutura organizacional capaz de garantir que os produtos e processos de software respeitem os requisitos especificados e são coerentes com os respectivos planos
Responsáveis:	gerente de projeto e analistas
Critério de inicialização:	um projeto de software foi iniciado, ou seja, uma instância do processo de <i>Definição</i> começou
Elementos de entrada:	especificação preliminar dos requisitos (Plan.II) (19) + informações sobre as normas, métodos, padrões, procedimentos técnicos, linguagens de programação e ferramentas, com base no negócio da empresa (Capac.I) (05)
Tarefas:	<ol style="list-style-type: none"> 1. identificar as relações existentes entre o processo de <i>Controle de Qualidade</i> e os processos de <i>Verificação</i> e <i>Validação</i> 2. definir os requisitos e padrões de qualidade 3. definir procedimentos para identificação, coleta, arquivamento, manutenção e disponibilização dos requisitos de qualidade, tanto dos produtos quanto dos processos envolvidos no projeto 4. identificar ferramentas de apoio às tarefas de controle da qualidade, de acordo com o processo de <i>Infra-Estrutura</i> 5. documentar os resultados num plano de controle da qualidade 6. aprovar o plano – tarefa do gerente de projeto
Elementos de saída:	plano de controle da qualidade (20)
Critério de finalização:	o plano de controle da qualidade foi concluído
Métricas registradas:	<ul style="list-style-type: none"> • % tempo gasto em cada tarefa de preparação

Atividade:	controlar a qualidade dos produtos (CQual.II)
Objetivo:	garantir que os produtos gerados ao longo do projeto apresentem os requisitos de qualidade estabelecidos
Responsáveis:	gerente de projeto, analistas, projetistas, programadores, operadores, mantenedores e gerentes de versões
Critério de inicialização:	o plano de controle da qualidade já foi aprovado
Elementos de entrada:	plano de controle da qualidade (CQual.I) (20) + resultados da verificação dos produtos (Verif.II) (21) + resultados da validação dos produtos (Valid.II) (22) + relatórios sobre os problemas e/ou não-conformidades resolvidos, quando existirem (RProb.II) (29)
Tarefas:	<ol style="list-style-type: none"> 1. garantir que todos os planos exigidos no projeto sejam documentados, mutuamente consistentes e executados quando exigidos 2. garantir que os produtos de software intermediários e a respectiva documentação seguiram os planos e satisfazem os requisitos especificados 3. garantir que o produto final, incluindo-se sua documentação, satisfaz todos os requisitos especificados pelo cliente, com base no estabelecido pelo processo de <i>Validação</i> 4. garantir que as medições dos produtos de software estejam de acordo com os padrões e procedimentos estabelecidos, de acordo com o processo de <i>Acompanhamento</i> 5. encaminhar eventuais problemas e/ou não-conformidades que não puderem ser resolvidas no escopo do processo de <i>Controle de Qualidade</i> ao processo de <i>Resolução de Problemas</i>
Elementos de saída:	problemas e/ou não-conformidades a serem tratados, quando existirem (27)
Critério de finalização:	a avaliação da qualidade dos produtos foi concluída
Métricas registradas:	<ul style="list-style-type: none"> • # total de problemas e/ou não-conformidades encontradas • # problemas e/ou não-conformidades encaminhadas ao processo de <i>Resolução de Problemas</i> • # problemas e/ou não-conformidades resolvidas no escopo do processo de <i>Controle de Qualidade</i> • # problemas e/ou não-conformidades encaminhadas / # total de não-conformidades encontradas • # problemas e/ou não-conformidades resolvidas / # total de não-conformidades encontradas

Atividade:	controlar a qualidade dos processos (CQual.III)
Objetivo:	garantir que os processos empregados ao longo do projeto apresentem os requisitos de qualidade estabelecidos
Responsáveis:	gerente de projeto, administrador dos recursos, analistas, projetistas, programadores, operadores, mantenedores e gerentes de versões
Critério de inicialização:	o plano de controle da qualidade já foi aprovado
Elementos de entrada:	plano de controle da qualidade (CQual.I) (20) + resultados da verificação dos processos (Verif.III) (24) + relatórios sobre os problemas e/ou não-conformidades resolvidos, quando existirem (RProb.II) (29)
Tarefas:	<ol style="list-style-type: none"> 1. garantir que todos os processos de software (primários e de apoio) seguiram os planos e respeitam os padrões 2. garantir que todas as práticas de Engenharia de Software, o ambiente de desenvolvimento e o ambiente de teste estejam em conformidade com os planos estabelecidos 3. garantir que as medições dos processos de software estejam de acordo com os padrões e procedimentos definidos, com base no estabelecido no processo de <i>Acompanhamento</i> 4. garantir que os times têm habilidade e conhecimento necessários para a realização das tarefas, com base nos resultados do processo de <i>Capacitação</i> – tarefa do administrador dos recursos 5. garantir que os times tenham recebido treinamentos adequados, com base nos resultados do processo de <i>Treinamento</i> – tarefa do administrador dos recursos 6. encaminhar eventuais problemas e/ou não-conformidades que não puderem ser resolvidos no escopo do processo de <i>Controle de Qualidade</i> ao processo de <i>Resolução de Problemas</i>
Elementos de saída:	problemas e/ou não-conformidades a serem tratados, quando existirem (27)
Critério de finalização:	a avaliação da qualidade dos processos foi devidamente concluída
Métricas registradas:	<ul style="list-style-type: none"> • # total de problemas e/ou não-conformidades encontradas • # problemas e/ou não-conformidades encaminhadas ao processo de <i>Resolução de Problemas</i> • # problemas e/ou não-conformidades resolvidas no escopo do processo de <i>Controle de Qualidade</i> • # problemas e/ou não-conformidades encaminhadas / # total de não-conformidades encontradas • # problemas e/ou não-conformidades resolvidas / # total de não-conformidades encontradas

Atividade:	preparar para o processo de Verificação (Verif.I)
Objetivo:	estabelecer uma estrutura organizacional adequada para a execução das atividades relacionadas ao processo de <i>Verificação</i>
Responsáveis:	gerente de projeto e analistas
Critério de inicialização:	um projeto de software foi iniciado, ou seja, uma instância do processo de <i>Definição</i> começou
Elementos de entrada:	especificação preliminar dos requisitos (Plan.II) (19) + informações sobre as normas, métodos, padrões, procedimentos técnicos, linguagens de programação e ferramentas, com base no negócio da empresa (Capac.I) (05)
Tarefas:	<ol style="list-style-type: none"> 1. identificar as relações existentes entre o processo de <i>Verificação</i> e o processo de <i>Controle de Qualidade</i> 2. analisar os requisitos do projeto, a fim de que riscos envolvidos, caso haja erros, sejam identificados 3. verificar as restrições das tecnologias a serem usadas e a disponibilidade dos recursos necessários 4. identificar ferramentas de apoio às tarefas de verificação, de acordo com o processo de <i>Infra-Estrutura</i> 5. elaborar um plano de verificação, identificando as tarefas e os recursos necessários ao processo 6. aprovar o plano – tarefa do gerente de projeto
Elementos de saída:	plano de verificação (23)
Critério de finalização:	o plano de verificação foi concluído
Métricas registradas:	<ul style="list-style-type: none"> • % tempo gasto em cada tarefa de preparação

Atividade:	verificar os produtos (Verif.II)
Objetivo:	avaliar as características dos produtos de software gerados pelos diversos processos envolvidos no projeto
Responsáveis:	gerente de projeto, analistas, projetistas, programadores, operadores, mantenedores e gerentes de versões
Critério de inicialização:	o plano de verificação já foi aprovado
Elementos de entrada:	plano de verificação (Verif.I) (23) + relatórios sobre os problemas e/ou não-conformidades resolvidos, quando existirem (RProb.II) (29)
Tarefas:	<ol style="list-style-type: none"> 1. revisar os requisitos do software e do sistema, a fim de verificar se são coerentes, factíveis e testáveis 2. revisar o projeto, a fim de verificar se ele está correto e coerente com os requisitos, implementando adequadamente as seqüências de eventos, entradas, saídas e requisitos de segurança 3. revisar o código, a fim de verificar se ele está de acordo com os requisitos e padrões de codificação e se é testável e correto, implementando os requisitos de segurança com métodos rigorosos 4. revisar os componentes e unidades de código, bem como itens de hardware e software, a fim de verificar se foram integrados completa e corretamente, de acordo com o plano de integração 5. revisar a documentação, a fim de verificar se ela está adequada, é completa, coerente e foi desenvolvida seguindo os procedimentos estabelecidos pelo processo de <i>Documentação</i> 6. revisar algum outro produto de trabalho, de acordo com o estabelecido no plano de verificação 7. encaminhar eventuais problemas e/ou não-conformidades que não puderem ser resolvidas no escopo do processo de <i>Verificação</i> ao processo de <i>Resolução de Problemas</i> 8. documentar os resultados 9. aprovar os resultados – tarefa do gerente de projeto
Elementos de saída:	resultados da verificação dos produtos (21) + problemas e/ou não-conformidades a serem tratados, quando existirem (27)
Critério de finalização:	os produtos foram devidamente verificados
Métricas registradas:	<ul style="list-style-type: none"> • tempo gasto em cada tarefa de verificação • # total de problemas e/ou não-conformidades encontrados • # problemas e/ou não-conformidades encaminhados / # problemas e/ou não-conformidades encontrados • # problemas e/ou não-conformidades resolvidos / # problemas e/ou não-conformidades encontrados • # total de problemas e/ou não-conformidades encontrados / produto • # problemas e/ou não-conformidades encaminhados ao processo de <i>Resolução de Problemas</i> / produto • # problemas e/ou não-conformidades resolvidos no escopo do processo de <i>Verificação</i> / produto

Atividade:	verificar os processos (Verif.III)
Objetivo:	avaliar as características dos processos (atividades e tarefas) de software envolvidos no projeto
Responsáveis:	gerente de projeto e administrador dos recursos
Critério de inicialização:	o plano de verificação já foi aprovado
Elementos de entrada:	plano de verificação (Verif.I) (23) + relatórios sobre os problemas e/ou não-conformidades resolvidos, quando existirem (RProb.II) (29)
Tarefas:	<ol style="list-style-type: none"> 1. revisar os procedimentos descritos nos planos, para verificar se estão adequados 2. verificar a implementação dos processos 3. verificar a definição e utilização dos padrões 4. verificar a condição dos profissionais para a realização das atividades, com base nos resultados dos processos de <i>Capacitação e Treinamento</i> 5. revisar alguma outra atividade ou tarefa, de acordo com o estabelecido no plano de verificação 6. encaminhar eventuais problemas e/ou não-conformidades que não puderem ser resolvidas no escopo do processo de <i>Verificação</i> ao processo de <i>Resolução de Problemas</i> 7. documentar os resultados 8. aprovar os resultados – tarefa do gerente de projeto
Elementos de saída:	resultados da verificação dos processos (24) + problemas e/ou não-conformidades a serem tratados, quando existirem (27)
Critério de finalização:	os processos foram devidamente verificados
Métricas registradas:	<ul style="list-style-type: none"> • tempo gasto em cada tarefa de verificação • # total de problemas e/ou não-conformidades encontrados • # problemas e/ou não-conformidades encaminhados / # problemas e/ou não-conformidades encontrados • # problemas e/ou não-conformidades resolvidos / # problemas e/ou não-conformidades encontrados • # total de problemas e/ou não-conformidades encontrados / processo • # problemas e/ou não-conformidades encaminhados ao processo de <i>Resolução de Problemas</i> / processo • # problemas e/ou não-conformidades resolvidos no escopo do processo de <i>Verificação</i> / processo

Atividade:	preparar para o processo de <i>Validação</i> (Valid.I)
Objetivo:	estabelecer uma estrutura organizacional adequada para a realização das atividades relacionadas ao processo de <i>Validação</i>
Responsáveis:	gerente de projeto e analistas
Critério de inicialização:	um projeto de software foi iniciado, ou seja, uma instância do processo de <i>Definição</i> começou
Elementos de entrada:	especificação preliminar dos requisitos (Plan.II) (19) + informações sobre as normas, métodos, padrões, procedimentos técnicos, linguagens de programação e ferramentas, com base no negócio da empresa (Capac.I) (05)
Tarefas:	<ol style="list-style-type: none"> 1. identificar as relações existentes entre o processo de <i>Validação</i> e o processo de <i>Controle de Qualidade</i> 2. identificar ferramentas de apoio às tarefas de validação, de acordo com o processo de <i>Infra-Estrutura</i> 3. elaborar um plano de validação, identificando as tarefas de validação a serem realizadas e os recursos necessários para o processo 4. aprovar o plano – tarefa do gerente de projeto
Elementos de saída:	plano de validação (25)
Critério de finalização:	o plano foi concluído
Métricas registradas:	<ul style="list-style-type: none"> • % tempo gasto em cada tarefa de preparação

Atividade:	validar o produto de software (Valid.II)
Objetivo:	determinar se o produto de software final atende aos requisitos especificados pelo usuário
Responsáveis:	gerente de projeto, analistas, projetistas, programadores, operadores, mantenedores e gerentes de versões
Critério de inicialização:	o produto de software final necessita de validação
Elementos de entrada:	plano de validação (Valid.I) (25) + relatórios sobre os problemas e/ou não-conformidades resolvidos, se existirem (RProb.II) (29)
Tarefas:	<p>Em conjunto com o processo de <i>Operação</i>:</p> <ol style="list-style-type: none"> 1. preparar os requisitos, os casos e as especificações de teste para análise dos resultados 2. assegurar que o produto reflete os requisitos particulares para um uso específico do software 3. conduzir testes considerando testes de estresse, de limite e de entradas específicas, avaliando também a habilidade do produto para isolar e minimizar os efeitos dos erros 4. conduzir testes com usuários representativos para avaliar se eles conseguem realizar suas tarefas usando o produto; 5. testar o produto apropriadamente no seu ambiente de operação 6. encaminhar eventuais problemas e/ou não-conformidades que não puderem ser resolvidas no escopo do processo de <i>Validação</i> ao processo de <i>Resolução de Problemas</i> 7. documentar os resultados 8. aprovar os resultados – tarefa do gerente de projeto
Elementos de saída:	resultados da validação dos produtos + não-conformidades a serem tratadas
Critério de finalização:	o item foi devidamente validado
Métricas registradas:	<ul style="list-style-type: none"> • tempo gasto em cada tarefa de validação • # total de problemas e/ou não-conformidades encontrados • # problemas e/ou não-conformidades encaminhados / # problemas e/ou não-conformidades encontrados • # problemas e/ou não-conformidades resolvidos / # problemas e/ou não-conformidades encontrados • # testes efetuados até a validação do produto • # tentativas de validação até o sucesso da atividade

Atividade:	preparar para o processo de <i>Resolução de Problemas</i> (RProb.I)
Objetivo:	estabelecer uma estrutura organizacional adequada para a realização das atividades relacionadas ao processo de <i>Resolução de Problemas</i>
Responsáveis:	gerente de projeto e analistas
Critério de inicialização:	um projeto de software foi iniciado, ou seja, uma instância do processo de <i>Definição</i> está sendo executada
Elementos de entrada:	informações sobre normas, métodos, padrões, procedimentos técnicos, linguagens de programação e ferramentas, com base no negócio da empresa (Capac.I) (05)
Tarefas:	<ol style="list-style-type: none"> 1. definir mecanismos para categorizar e priorizar os problemas 2. documentar os resultados num plano de resolução de problemas, garantindo que após sua detecção, os problemas sejam rapidamente notificados e as causas sejam identificadas, analisadas e, quando possível, eliminadas 3. aprovar o plano – tarefa do gerente de projeto
Elementos de saída:	plano de resolução dos problemas e/ou não-conformidades encontrados (26)
Critério de finalização:	o plano de resolução de problemas foi aprovado pelo gerente de projeto
Métricas registradas:	<ul style="list-style-type: none"> • % tempo gasto em cada tarefa de preparação

Atividade:	resolver os problemas e/ou não-conformidades (RProb.II)
Objetivo:	eliminar os problemas e/ou não-conformidades que tenham surgido em algum momento da execução de um processo
Responsáveis:	gerente de projeto
Critério de inicialização:	há um problema e/ou não-conformidade necessitando de solução
Elementos de entrada:	plano de resolução dos problemas e/ou não-conformidades encontrados (RProb.I) (26) + problemas e/ou não-conformidades a serem tratados (provenientes de algum dos processos de <i>Desenvolvimento, Operação, Gerenciamento de Configuração, Controle de Qualidade, Verificação ou Validação</i>) (27)
Tarefas:	<ol style="list-style-type: none"> 1. preparar um relatório sobre cada problema e/ou não-conformidade, que os descreva e que, fazendo uso dos mecanismos previamente definidos, possa categorizá-los e priorizá-los 2. de acordo com a prioridade estabelecida, fazendo uso de pessoal capacitado e em conjunto com o processo de <i>Gerência</i>: <ul style="list-style-type: none"> ▪ analisar o problema e/ou não-conformidade ▪ solucionar o problema e/ou não-conformidade 3. atualizar o relatório, na medida em que as tarefas anteriores forem sendo concluídas 4. encaminhar o relatório ao processo responsável pelo envio do problema e/ou não-conformidade ao processo de <i>Resolução de Problemas</i>
Elementos de saída:	relatório sobre os problemas e/ou não-conformidades resolvidos (29)
Critério de finalização:	os problemas e/ou não-conformidades que serviram de entrada ao processo foram tratados
Métricas registradas:	<ul style="list-style-type: none"> • % tempo gasto nas tarefas de análise e solução • tempo médio de resolução de um problema e/ou não-conformidade • tempo médio de resolução de um problema e/ou não-conformidade / prioridade • # total de problemas e/ou não-conformidades resolvidos no projeto • # problemas e/ou não-conformidades resolvidos / processo

Atividade:	preparar para o processo de <i>Gerência</i> (Geren.I)
Objetivo:	estabelecer uma estrutura organizacional adequada para a realização das atividades relacionadas ao processo de <i>Gerência</i> , tornando possível a execução dos processos praticados na empresa
Responsáveis:	gerente da empresa
Critério de inicialização:	o negócio da empresa necessita ser estabelecido, a fim de que ela possa ser capaz de desenvolver bons projetos para se estabelecer e se manter no mercado
Elementos de entrada:	não identificado (Obs.: A base para a realização das tarefas constitui-se nos conhecimentos prévios e na capacitação do gerente da empresa)
Tarefas:	<ol style="list-style-type: none"> 1. estabelecer o negócio da empresa (visão e cultura da organização) e os papéis iniciais para os processos organizacionais 2. determinar os requisitos dos processos a serem executados no ambiente da empresa 3. estabelecer o escopo de cada processo e as inter-relações entre os processos 4. preparar, juntamente com os processos de <i>Capacitação</i>, <i>Melhoria</i> e <i>Infra-Estrutura</i>, um roteiro de execução dos processos, com base no escopo de cada um deles, que contenha cronogramas, estimativas de esforço, recursos necessários, alocação de tarefas, determinação de responsabilidades, quantificação dos riscos associados a cada processo, medições e custos
Elementos de saída:	roteiro de execução dos processos (01)
Critério de finalização:	a dinâmica dos processos foi identificada e um roteiro inicial para colocá-la em prática está concluído
Métricas registradas:	<ul style="list-style-type: none"> • % tempo gasto em cada tarefa de preparação • # processos a serem executados na empresa/no projeto • # médio de requisitos / processo identificado

Atividade:	administrar a execução de um processo (Geren.II)
Objetivo:	gerenciar a execução de um processo
Responsáveis:	gerente da empresa
Critério de inicialização:	o roteiro de execução dos processos foi definido e um determinado processo (primário ou de apoio) deverá ter início
Elementos de entrada:	roteiro de execução dos processos (Geren.I) (01) + plano para execução das melhorias a serem efetuadas, quando existir (Melh.II) (12)
Tarefas:	<ol style="list-style-type: none"> 1. monitorar a execução do processo, inteirando-se do progresso das atividades 2. analisar e solucionar problemas, em conjunto com o processo de <i>Resolução de Problemas</i> 3. assegurar que as melhorias propostas pelo processo de <i>Melhoria</i> tenham sido colocadas em prática, favorecendo-as e monitorando sua execução, a fim de garantir que não sejam introduzidas inconsistências em outras atividades 4. atualizar o roteiro de execução dos processos, a fim de que ele reflita a exata situação do que ocorre na empresa
Elementos de saída:	roteiro de execução dos processos (01) atualizado
Critério de finalização:	não identificado
Métricas registradas:	não identificado

Atividade:	iniciar o processo de <i>Capacitação</i> (Capac.I)
Objetivo:	estabelecer uma estrutura organizacional adequada para a realização das atividades relacionadas ao processo de <i>Capacitação</i>
Responsáveis:	gerente de projeto e administrador dos recursos
Critério de inicialização:	o negócio da empresa já está sendo estabelecido pelo processo de <i>Gerência</i>
Elementos de entrada:	informações sobre técnicas de preparação e capacitação de pessoal (informações externas) (02) + informações sobre normas, métodos, padrões, procedimentos técnicos, linguagens de programação e ferramentas (informações externas) (03)
Tarefas:	<ol style="list-style-type: none"> 1. selecionar e adequar as normas, os métodos, os padrões, os procedimentos técnicos, as linguagens de programação e as ferramentas, com base no negócio da empresa, em conjunto com o estabelecido no processo de <i>Gerência</i> 2. fornecer aos indivíduos da organização visão e cultura do processo de produção de software, de forma a capacitá-los a trabalhar cooperativa e eficientemente (cartilhas, palestras, discussões, encontros, etc.), também em conjunto com o processo de <i>Gerência</i> 3. elaborar um plano para obter informações periódicas sobre os times de trabalho, tais como: número de times de trabalho, número de membros em cada time, formação e experiência de cada membro, capacidades técnica e gerencial dos profissionais e dos times, etc., através de entrevistas, observações, medições, etc., estabelecendo, inclusive, mecanismos de graduação dos resultados, em conjunto com os processos de <i>Melhoria</i> e de <i>Gerência</i> 4. definir ferramentas e/ou métodos para a coleta de dados, de acordo com o especificado no plano e com base no processo de <i>Infra-Estrutura</i>
Elementos de saída:	plano para avaliação e determinação da capacidade dos profissionais (04) + informações sobre as normas, métodos, padrões, procedimentos técnicos, linguagens de programação e ferramentas, com base no negócio da empresa (05) + informações sobre os recursos humanos, técnicos e econômicos disponíveis na empresa (14)
Critério de finalização:	o plano foi aprovado pelo gerente da empresa e as informações sobre normas e padrões foram filtradas, com base no negócio da empresa
Métricas registradas:	<ul style="list-style-type: none"> • % tempo gasto em cada tarefa de preparação • # palestras, discussões, encontros, etc. promovidos • % profissionais participantes

Atividade:	determinar as capacidades dos profissionais (Capac.II)
Objetivo:	avaliar os times de trabalho pela identificação de seus recursos humanos
Responsáveis:	gerente de projeto e administrador dos recursos
Critério de inicialização:	o plano para avaliação e determinação da capacidade dos profissionais da empresa está pronto para ser seguido
Elementos de entrada:	plano para determinação e avaliação das capacidades dos profissionais (Capac.I) (04)
Tarefas:	<ol style="list-style-type: none"> 1. coletar os dados identificados no plano 2. determinar as capacidades dos profissionais, com base nos dados coletados e segundo mecanismos de quantificação e qualificação definidos no plano 3. associar um perfil a cada um 4. documentar os resultados num relatório de capacitação dos profissionais
Elementos de saída:	relatório de capacitação dos profissionais (10)
Critério de finalização:	o relatório da capacitação dos profissionais foi concluído
Métricas registradas:	<ul style="list-style-type: none"> • # total de profissionais participantes do projeto • # membros / perfil • # membros / time • tempo gasto na coleta dos dados

Atividade:	preparar para o processo de <i>Melhoria</i> (Melh.I)
Objetivo:	definir uma estrutura que possibilite a avaliação contínua de todos os processos da empresa e que, com base nessa avaliação, forneça diretrizes para melhoria
Responsáveis:	gerente de projeto
Critério de inicialização:	o negócio da empresa já está sendo estabelecido pelo processo de <i>Gerência</i>
Elementos de entrada:	não identificado
Tarefas:	<ol style="list-style-type: none"> 1. definir possíveis objetivos específicos de melhoria para cada processo (sobretudo quanto aos recursos humanos e computacionais), que possam refletir a melhoria do processo global da empresa, em conjunto com o estabelecido no processo de <i>Gerência</i> 2. selecionar/definir métricas (objetivas e subjetivas) adequadas para se realizar medições, com base nos objetivos estabelecidos e também em conjunto com o processo de <i>Gerência</i> 3. identificar procedimentos e ferramentas para coleta das métricas, sem sobrecarregar os times em suas tarefas e sem afastá-los de suas atividades principais 4. definir instrumentos de avaliação (entrevistas, observações, etc.) do nível de satisfação tanto dos indivíduos da empresa quanto dos clientes 5. definir mecanismos para avaliar, quantitativa e qualitativamente, todos os resultados obtidos 6. compilar os resultados numa estratégia para coleta e avaliação de métricas para melhoria de processo
Elementos de saída:	estratégia de coleta e avaliação de métricas para melhoria de processo (07)
Critério de finalização:	a estratégia para coleta e avaliação de métricas para melhoria dos processos foi definida
Métricas registradas:	<ul style="list-style-type: none"> • % tempo gasto em cada tarefa de preparação • # métricas selecionadas/definidas

Atividade:	melhorar os processos (Melh.II)
Objetivo:	melhorar continuamente a efetividade e a eficiência dos processos utilizados e empregados na empresa
Responsáveis:	gerente de projeto
Critério de inicialização:	a estratégia para coleta e avaliação de métricas para melhoria dos processos foi definida
Elementos de entrada:	estratégia de coleta e avaliação de métricas para melhoria de processo (Melh.I) (07) + relatório de capacitação dos profissionais (Capac.II) (10) + relatórios com os resultados de treinamento, se existirem (Trein.II) (11) + relatórios sobre a utilização e situação da infra-estrutura (IEstrut.II) (13)
Tarefas:	<ol style="list-style-type: none"> 1. coletar as métricas indicadas na estratégia de coleta e avaliação de métricas, como parte integrante do projeto 2. analisar os resultados obtidos, tanto das métricas coletadas quanto dos relatórios que servem de entrada para esta atividade 3. documentar e armazenar cada medição, desde a coleta da métrica até a realização da modificação, nos casos em que ela ocorrer, sempre com a ciência e com a garantia do processo de <i>Gerência</i> 4. realizar estudos empíricos envolvendo as medições dos processos de software e os dados armazenados na "base de dados históricos" 5. preparar um plano de melhoria, para cada processo, caracterizando sua situação atual e sugerindo possíveis mudanças (inclusive na infra-estrutura ou baseados em treinamento), com base na análise dos resultados obtidos e considerando o impacto da mudança nos demais processos e para o negócio da empresa
Elementos de saída:	plano para execução das melhorias a serem efetuadas (12)
Critério de finalização:	o plano foi concluído e está sendo colocado em prática, na realização das mudanças
Métricas registradas:	<ul style="list-style-type: none"> • # total de melhorias efetuadas • # melhorias efetuadas / processo • tempo gasto na coleta das métricas de melhoria • tempo gasto na avaliação das métricas de melhoria • evolução dos níveis de satisfação dos indivíduos • evolução dos níveis de satisfação dos clientes • evolução dos níveis de satisfação dos indivíduos / processo

Atividade:	definir a infra-estrutura e sua instalação (IEstrut.I)
Objetivo:	fazer o planejamento dos equipamentos, sistemas computacionais e ferramentas necessárias para a execução dos demais processos
Responsáveis:	gerente de projeto e administrador dos recursos
Critério de inicialização:	o negócio da empresa já está sendo estabelecido pelo processo de <i>Gerência</i>
Elementos de entrada:	não identificado
Tarefas:	<ol style="list-style-type: none"> 1. definir a infra-estrutura necessária, em conjunto com o estabelecido no processo de <i>Gerência</i> 2. planejar a aquisição (quando necessária), a instalação e o funcionamento da infra-estrutura 3. definir a estrutura e o acesso à "base de dados históricos" da empresa, a qual deverá servir para o registro das terminologias e dos eventos relacionados ao projeto, armazenando, assim, documentos importantes para estimativas e melhorias
Elementos de saída:	plano para instalação da infra-estrutura, incluindo diretrizes para aquisição (quando necessária) (06)
Critério de finalização:	o plano para aquisição (quando necessária), instalação e funcionamento da infra-estrutura foi concluído
Métricas registradas:	<ul style="list-style-type: none"> • % tempo gasto em cada tarefa de preparação • # recursos identificados

Atividade:	instalar e manter a infra-estrutura (IEstrut.II)
Objetivo:	fornecer e manter um ambiente de Engenharia de Software e facilidades de trabalho para a realização dos projetos da empresa
Responsáveis:	gerente de projeto e administrador dos recursos
Critério de inicialização:	a infra-estrutura para a empresa/o projeto já foi definida e o plano para sua instalação, concluído
Elementos de entrada:	plano para instalação da infra-estrutura, incluindo diretrizes para aquisição (quando necessária) (IEstrut.I) (06)
Tarefas:	<ol style="list-style-type: none"> 1. adquirir (quando preciso), instalar e/ou renovar a infra-estrutura 2. monitorar a utilização da infra-estrutura 3. emitir relatório periódico sobre a utilização e a situação da infra-estrutura, propondo mudanças, quando necessárias
Elementos de saída:	relatório sobre a utilização e situação da infra-estrutura (13)
Critério de finalização:	o relatório sobre a utilização e situação da infra-estrutura foi devidamente concluído
Métricas registradas:	<ul style="list-style-type: none"> • # modificações na infra-estrutura / unidade de tempo • # modificações na infra-estrutura / projeto • # relatórios emitidos • tempo gasto na instalação da infra-estrutura

Atividade:	preparar para o processo de <i>Treinamento</i> (Trein.I)
Objetivo:	estabelecer uma estrutura organizacional que possibilite a realização de treinamento para algum processo de um departamento específico
Responsáveis:	gerente de projeto e administrador dos recursos
Critério de inicialização:	uma solicitação por treinamento (proveniente de algum processo) foi efetuada
Elementos de entrada:	diretrizes de treinamento, provenientes de algum processo para o qual esse treinamento seja necessário (08)
Tarefas:	<ol style="list-style-type: none"> 1. identificar, em conformidade com o estabelecido nos processos de <i>Capacitação</i> e <i>Infra-Estrutura</i>, os recursos humanos necessários para a realização das atividades de treinamento 2. definir um plano de treinamento, estabelecendo cronogramas, metodologias e procedimentos para avaliação e aproveitamento
Elementos de saída:	plano de treinamento (09)
Critério de finalização:	o plano de treinamento foi concluído
Métricas registradas:	<ul style="list-style-type: none"> • % tempo gasto em cada tarefa de preparação • % profissionais com necessidade de treinamento • % profissionais com necessidade de treinamento / processo

Atividade:	realizar o treinamento (Trein.II)
Objetivo:	desenvolver a habilidade e o conhecimento dos indivíduos para que eles possam desempenhar suas funções efetivamente e de forma eficiente na empresa
Responsáveis:	gerente de projeto e administrador dos recursos
Critério de inicialização:	o plano de treinamento foi concluído
Elementos de entrada:	plano de treinamento (Trein.I) (09)
Tarefas:	<ol style="list-style-type: none"> 1. elaborar o material de treinamento, seguindo o plano 2. treinar os profissionais 3. avaliar o aproveitamento dos participantes, também seguindo os critérios de avaliação estabelecidos no plano 4. atualizar o perfil de cada profissional, em conjunto com o processo de <i>Capacitação</i> 5. preparar um relatório com os resultados do treinamento, descrevendo o aproveitamento e sugerindo modificações
Elementos de saída:	relatório com os resultados de treinamento (11)
Critério de finalização:	o relatório de treinamento foi devidamente concluído
Métricas registradas:	<ul style="list-style-type: none"> • tempo total gasto no treinamento • nível de aproveitamento / profissional • nível de aproveitamento / time • nível de aproveitamento global

APÊNDICE D

Questionário para Graduação dos Processos da Abordagem *GQS-AE*

Seção 1: processo de *Definição*
Esta seção compreende as atividades envolvidas no processo de *Definição*, a saber:

- **Def.I:** identificar as necessidades iniciais
- **Def.II:** avaliar a viabilidade do projeto

Atividades	Tarefas	Não cumpre	Cumpre parcialmente	Cumpre totalmente
Def.I	1. definir, mais formalmente, o problema			
	2. fazer um levantamento inicial dos requisitos do sistema, incluindo-se aspectos de segurança, confiabilidade e desempenho			
	3. determinar o escopo do sistema e as características do ambiente de operação			
	4. aprovar esse levantamento inicial			
Def.II	1. identificar, dentre os recursos disponíveis, aqueles necessários ao projeto			
	2. analisar os riscos, custos e benefícios de possíveis opções de desenvolvimento, de acordo com os recursos disponíveis			
	3. encaminhar, ao processo de <i>Resolução de Problemas</i> , problemas e/ou não-conformidades que não puderem ser resolvidos no escopo do processo de <i>Definição</i>			
	4. celebrar o contrato com o cliente			
	5. documentar os resultados			
Total (%)				

Seção 2: processo de *Planejamento*

Esta seção compreende as atividades envolvidas no processo de *Planejamento*, a saber:

- **Plan.I:** preparar para o processo de *Planejamento*
- **Plan.II:** refinar o levantamento inicial
- **Plan.III:** estabelecer o modelo de desenvolvimento de software adequado ao projeto
- **Plan.IV:** identificar os riscos do projeto
- **Plan.V:** preparar as estimativas para o projeto
- **Plan.VI:** realizar o planejamento das atividades de apoio ao desenvolvimento do projeto

Atividades	Tarefas	Não cumpre	Cumpre parcialmente	Cumpre totalmente
Plan.I	1. determinar a metodologia de trabalho a ser seguida			
	2. identificar ferramentas de apoio às tarefas de planejamento, com base no processo de <i>Infra-Estrutura</i>			
	3. identificar a necessidade de treinamento para as atividades do processo de <i>Planejamento</i> , em conformidade com o estabelecido no processo de <i>Capacitação</i> , preparando as diretrizes para o mesmo, se ele for necessário			
	4. disponibilizar a “base de dados históricos”, como definido no processo de <i>Infra-Estrutura</i>			
	5. documentar os resultados			
Plan.II	1. alocar as atividades do processo aos profissionais, com base nos resultados dos processos de <i>Capacitação e Treinamento</i>			
	2. analisar o problema a ser tratado e os requisitos iniciais			
	3. definir as características, os usuários e as interfaces do sistema proposto			
	4. fazer a modelagem inicial do sistema, utilizando o método de desenvolvimento adequado			
	5. avaliar a consistência e a conformidade com o levantamento inicial			
	6. documentar os resultados			
	7. aprovar os resultados			
Plan.III	1. determinar os componentes do sistema			
	2. classificar os componentes de acordo com o conhecimento e recursos técnicos necessários para sua construção;			
	3. identificar um modelo de desenvolvimento apropriado para o projeto, segundo as características identificadas anteriormente			
	4. documentar os resultados			
	5. aprovar os resultados			
Plan.IV	1. identificar os riscos envolvidos no projeto			
	2. fazer uma estimativa dos riscos			
	3. definir opções para evitar os riscos			
	4. definir procedimentos para a monitoração dos riscos			
	5. documentar os resultados			
	6. aprovar os resultados			
Plan.V	1. definir o método de estimativas adequado			
	2. elaborar as estimativas de prazo			
	3. elaborar as estimativas de esforço e custo			
	4. documentar as estimativas			
	5. aprovar as estimativas			

Plan.VI	1. fazer um levantamento dos produtos de software e das atividades necessárias para o estabelecimento e manutenção do controle do projeto, em conjunto com o processo de <i>Acompanhamento</i>			
	2. estabelecer os pontos de revisão e os dados a serem coletados nesses pontos de revisão, também em conjunto com o processo de <i>Acompanhamento</i>			
	3. fazer o planejamento da qualidade, com base no estabelecido pelo processo de <i>Controle de Qualidade</i>			
	4. planejar a operação e a manutenção do sistema			
	5. documentar os resultados			
	6. aprovar os resultados			
	Total (%)			

Seção 3: processo de *Desenvolvimento*

Esta seção compreende as atividades envolvidas no processo de *Desenvolvimento*, a saber:

- **Desenv.I:** iniciar o processo de *Desenvolvimento*
- **Desenv.II:** especificar os requisitos
- **Desenv.III:** desenvolver o projeto de software
- **Desenv.IV:** implementar o sistema
- **Desenv.V:** realizar os testes individuais
- **Desenv.VI:** integrar e testar o sistema
- **Desenv.VII:** instalar e entregar o sistema

Atividades	Tarefas	Não cumpre	Cumpre parcialmente	Cumpre totalmente
Desenv.I	1. identificar ferramentas de apoio às tarefas de desenvolvimento, com base no processo de <i>Infra-Estrutura</i>			
	2. desenvolver um plano para a condução das atividades, de acordo com as diretrizes estabelecidas para os requisitos de software e com base nas informações de entrada da atividade, que deve incluir, sobretudo, um cronograma de desenvolvimento			
	3. identificar a necessidade de treinamento para as atividades do processo, em conformidade com o estabelecido no processo de <i>Capacitação</i> , preparando as diretrizes para o mesmo, se necessário			
	4. documentar os resultados			
Desenv.II	1. alocar as atividades do processo aos profissionais, com base nos resultados dos processos de <i>Capacitação e Treinamento</i>			
	2. estabelecer os requisitos do projeto e do sistema			
	3. identificar os requisitos do sistema que devem ser alocados ao projeto, bem como as restrições de projeto			
	4. estabelecer, analisar e refinar os requisitos do software (descrição funcional, de controle e comportamental do sistema), de acordo com o processo de <i>Verificação</i>			
	5. estabelecer os critérios de validação, em conjunto com o processo de <i>Validação</i>			
	6. explicitar os requisitos de qualidade, respeitando-se o estabelecido no processo de <i>Controle de Qualidade</i>			
	7. documentar a especificação			
	8. aprovar os resultados			
Desenv.III	1. definir a estrutura de cada componente, identificando os itens de hardware e software necessários			
	2. possibilitar reúso			
	3. verificar se o projeto arquitetural definido atende à Especificação de Requisitos previamente estabelecida, de acordo com o processo de <i>Verificação</i>			
	4. refinar os componentes individuais de software até o nível de interfaces, classes ou componentes pré-existent			
	5. definir bases de dados, linguagens de programação, padrões de integração e estilos de interface			
	6. definir os requisitos de teste de unidade e de integração			
	7. documentar os resultados			
	8. aprovar o projeto			
Desenv.IV	1. desenvolver o código para cada componente projetado			
	2. documentar o código, com base no estabelecido pelo processo de <i>Documentação</i>			
	3. iniciar a elaboração do Manual do Usuário			
Desenv.V	1. estabelecer planos e procedimentos, casos e critérios de teste, a fim de que seja testado cada um dos componentes do sistema			
	2. realizar os testes, verificando se cada componente testado tem satisfeitos os requisitos de teste definidos anteriormente e com base no processo de <i>Verificação</i>			
	3. corrigir eventuais falhas evidenciadas pelos testes individuais			
	4. encaminhar, ao processo de <i>Resolução de Problemas</i> , problemas e/ou não-conformidades que não puderem ser resolvidos no escopo do processo de <i>Desenvolvimento</i>			
	5. documentar os resultados			

Desenv. VI	1. elaborar um plano para integrar os componentes de software, definindo casos e critérios de teste, bem como os procedimentos necessários			
	2. integrar os componentes, de acordo com o plano			
	3. testar o comportamento do software integrado, verificando se os requisitos definidos anteriormente são satisfeitos, com base no processo de <i>Verificação</i>			
	4. corrigir eventuais falhas evidenciadas pelos testes de integração			
	5. finalizar o Manual do Usuário, também verificando-o com base no processo de <i>Verificação</i>			
	6. validar o sistema, avaliando o projeto, código, testes, resultados dos testes e documentação, de acordo com o que foi definido no processo de <i>Validação</i>			
	7. gerar o código executável do sistema			
	8. gerar um relatório propondo futuras melhorias, para futuras versões			
	9. encaminhar, ao processo de <i>Resolução de Problemas</i> , problemas e/ou não-conformidades que não puderem ser resolvidos no escopo do processo de <i>Desenvolvimento</i>			
	10. documentar os resultados			
Desenv. VII	1. elaborar um plano de instalação do sistema no ambiente de operação			
	2. disponibilizar toda a documentação necessária para instalação e utilização do sistema, sobretudo o Manual do Usuário			
	3. introduzir os dados iniciais necessários à execução do sistema			
	4. documentar os resultados			
	5. aprovar os resultados			
Total (%)				

Seção 4: processo de *Operação*

Esta seção compreende as atividades envolvidas no processo de *Operação*, a saber:

- **Oper.I:** preparar para o processo de *Operação*
- **Oper.II:** cuidar dos testes operacionais e da operação do sistema
- **Oper.III:** fornecer assistência e/ou suporte aos usuários

Atividades	Tarefas	Não cumpre	Cumpre parcialmente	Cumpre totalmente
Oper.I	1. identificar ferramentas de apoio às tarefas de operação, de acordo com o processo de <i>Infra-Estrutura</i>			
	2. desenvolver um plano para a condução das atividades relacionadas à operação e ao suporte aos usuários, com base no estabelecido pelo processo de <i>Planejamento</i>			
	3. estabelecer critérios para o teste operacional, para inserir pedidos de modificação no processo de <i>Manutenção</i> e para liberar o produto para uso operacional			
	4. identificar a necessidade de treinamento para as atividades do processo, preparando as diretrizes para o mesmo, se ele for necessário, com conformidade com o estabelecido no processo de <i>Capacitação</i>			
	5. documentar os resultados			
	6. aprovar os resultados			
Oper.II	1. alocar as atividades do processo aos profissionais, com base nos resultados dos processos de <i>Capacitação e Treinamento</i>			
	2. executar o teste operacional, respeitando-se o disposto no Manual do Usuário, em conjunto com o processo de <i>Validação</i>			
	3. avaliar os testes (resultados obtidos X resultados esperados), também em conjunto com o processo de <i>Validação</i>			
	4. liberar o sistema para uso, tendo sido, os testes, satisfatórios			
	5. operar o sistema no ambiente para o qual foi desenvolvido, em conjunto com o usuário, fornecendo a ele o auxílio necessário nas atividades iniciais de operação			
	6. documentar os resultados			
	7. encaminhar para o processo de <i>Resolução de Problemas</i> os problemas e/ou não-conformidades que não puderem ser resolvidos no escopo do processo de <i>Operação</i>			
Oper.III	1. formalizar o pedido, seguindo-se o estabelecido no processo de <i>Documentação</i>			
	2. fornecer assistência, suporte e consultoria aos usuários, documentando suas dúvidas e dificuldades, e solucionando os problemas			
	3. encaminhar, ao processo de <i>Manutenção</i> , pedidos de manutenção, nos casos necessários, ou possíveis melhorias a serem realizadas			
	4. monitorar a execução dos pedidos de manutenção			
	5. fornecer retorno aos solicitantes			
	6. aplicar os instrumentos de avaliação do nível de satisfação dos usuários (com relação ao sistema e aos serviços da empresa) definidos no processo de <i>Melhoria</i>			
	7. documentar as tarefas			
	8. aprovar os resultados			
	Total (%)			

Seção 5: processo de *Manutenção*

Esta seção compreende as atividades envolvidas no processo de *Manutenção*, a saber:

- **Manut.I:** preparar para o processo de *Manutenção*
- **Manut.II:** fazer a análise do problema
- **Manut.III:** implementar a modificação

Atividades	Tarefas	Não cumpre	Cumpre parcialmente	Cumpre totalmente
Manut.I	1. identificar ferramentas de apoio às tarefas de manutenção, de acordo com o processo de <i>Infra-Estrutura</i>			
	2. desenvolver planos e procedimentos para conduzir as atividades e tarefas de manutenção, incluindo-se os casos de migração e descontinuação, com base no estabelecido pelo processo de <i>Planejamento</i>			
	3. estabelecer procedimentos para receber, armazenar e acompanhar os pedidos de modificação e os relatórios de problemas dos usuários			
	4. identificar a necessidade de treinamento para as atividades do processo, preparando as diretrizes para o mesmo, se ele for necessário, com conformidade com o estabelecido no processo de <i>Capacitação</i>			
	5. documentar os planos e procedimentos			
	6. aprovar os resultados			
Manut.II	1. alocar as atividades do processo aos profissionais, com base nos resultados dos processos de <i>Capacitação e Treinamento</i>			
	2. avaliar o pedido de modificação, considerando o tipo de manutenção a ser realizada, o alcance da alteração e as consequências de tal mudança			
	3. desenvolver opções para a implementação da modificação			
	4. documentar os resultados da análise			
	5. tomar a decisão sobre o pedido de alteração. Para pedidos aprovados: <ul style="list-style-type: none"> ▪ informar a prioridade de execução da alteração • emitir a "ordem de serviço" 			
Manut.III	1. determinar e documentar quais unidades de software, versões e documentação relacionada precisam ser modificadas			
	2. implementar a modificação, respeitando-se os processos de <i>Desenvolvimento e Operação</i>			
	3. efetuar testes de regressão nas partes já existentes, para revalidar o sistema, de acordo com o processo de <i>Validação</i>			
	4. executar, quando necessário e seguindo-se o estabelecido nos planos e procedimentos para manutenção, a migração e/ou a descontinuação do software			
	5. documentar os resultados			
	6. aprovar as modificações			
Total (%)				

Seção 6: processo de *Documentação*
 Esta seção compreende as atividades envolvidas no processo de *Documentação*, a saber:

- **Doc.I:** preparar para o processo de *Documentação*
- **Doc.II:** gerenciar a produção dos documentos

Atividades	Tarefas	Não cumpre	Cumpre parcialmente	Cumpre totalmente
Doc.I	1. identificar os requisitos de documentação do projeto, em conformidade com o processo de <i>Definição</i>		.	
	2. identificar os documentos a serem produzidos, de acordo com seu conteúdo técnico			
	3. estabelecer a forma de organização – sobretudo com relação à hierarquia – e de identificação dos documentos			
	4. projetar a estrutura e a apresentação dos documentos			
	5. aprovar os resultados, compilando-os em diretrizes para produção dos documentos necessários ao longo do projeto			
	6. cuidar para que todos os demais processos da empresa tenham conhecimento dessas diretrizes e possam utilizá-las			
Doc.II	1. identificar os documentos que devem ser armazenados na “base de dados históricos”			
	2. estabelecer os procedimentos para o armazenamento desses documentos			
	3. identificar o padrão de documentação interna e externa do código-fonte, de acordo com a linguagem de programação a ser utilizada			
	4. armazenar/rearmazenar na “base de dados históricos” ou eliminar dela os documentos apropriados, de acordo com os procedimentos definidos e com base no processo de <i>Verificação</i>			
Total (%)				

Seção 7: processo de *Acompanhamento*
Esta seção compreende as atividades envolvidas no processo de *Acompanhamento*, a saber:

- **Acomp.I:** preparar para o processo de *Acompanhamento*
- **Acomp.II:** acompanhar o projeto de software

Atividades	Tarefas	Não cumpre	Cumpre parcialmente	Cumpre totalmente
Acomp.I	1. elaborar um plano de acompanhamento que identifique itens a serem revistos, pontos de revisão e recursos necessários para se realizar as revisões, sobretudo os técnicos			
	2. definir procedimentos, associadas aos produtos e às atividades, para o estabelecimento e manutenção do controle do projeto			
	3. identificar ferramentas de apoio às tarefas de acompanhamento, com base no processo de <i>Infra-Estrutura</i>			
	4. documentar os planos e procedimentos			
	5. aprovar os resultados			
Acomp.II	1. coletar as métricas necessárias, em conjunto com o processo de <i>Controle de Qualidade</i>			
	2. avaliar os custos e riscos envolvidos no projeto, os recursos e esforços necessários ao seu desenvolvimento, e o seu cronograma (revisões gerenciais)			
	3. avaliar os produtos e serviços, a fim de verificar se estão completos, respeitam os padrões e especificações e estão prontos para se poder executar a próxima atividade para a qual servem de entrada (revisões técnicas), com o auxílio do processo de <i>Controle de Qualidade</i>			
	4. tomar atitudes corretivas, quando necessário e possível, alterando a forma como o trabalho está sendo feito, sem prejuízo para o projeto e para o curso de qualquer outro processo, e desde que com a ciência do processo de <i>Gerência</i>			
Total (%)				

Seção 8: processo de *Gerenciamento de Configuração*

Esta seção compreende as atividades envolvidas no processo de *Gerenciamento de Configuração*, a saber:

- **GConfig.I:** preparar para o processo de *Gerenciamento de Configuração*
- **GConfig.II:** selecionar os itens de configuração a serem controlados
- **GConfig.III:** aprovar/rejeitar um pedido de alteração
- **GConfig.IV:** implementar as alterações
- **GConfig.V:** relatar a situação da configuração

Atividades	Tarefas	Não cumpre	Cumpre parcialmente	Cumpre totalmente
GConfig.I	1. determinar ferramentas, padrões e procedimentos a serem utilizados, sobretudo quanto ao repositório de software (funcionamento e acesso), de acordo com o processo de <i>Infra-Estrutura</i>			
	2. identificar a necessidade de treinamento para as atividades do processo de <i>Gerenciamento de Configuração</i> , em conformidade com o processo de <i>Capacitação</i> , preparando as diretrizes para o mesmo, se ele for necessário			
	3. documentar os resultados, a serem reunidos num plano de gerenciamento de configuração			
	4. aprovar o plano			
GConfig.II	1. alocar as atividades aos profissionais, de acordo com o estabelecido nos processos de <i>Capacitação e Treinamento</i>			
	2. definir os itens de configuração relevantes ao projeto que serão efetivamente colocados sob controle de configuração			
	3. descrever as relações entre os itens de configuração selecionados			
	4. definir uma forma (ferramenta, por exemplo) para identificar, unicamente, cada item de configuração (nome, tipo, versão, etc.)			
	5. estabelecer as <i>linhas de referência (baselines)</i>			
	6. documentar cada item de configuração, descrevendo a maneira como eles serão arquivados e recuperados do repositório de software, de acordo com o estabelecido no processo de <i>Documentação</i>			
	7. aprovar os resultados			
GConfig.III	1. avaliar o pedido de alteração			
	2. relatar os resultados da avaliação			
	3. tomar a decisão sobre o pedido de alteração. Para pedidos aprovados: <ul style="list-style-type: none"> • informar a prioridade de execução do pedido • emitir a "ordem de serviço" 			
	4. encaminhar eventuais problemas e/ou não-conformidades que não puderem ser resolvidos no escopo do processo de <i>Gerenciamento de Configuração</i> ao processo de <i>Resolução de Problemas</i>			
GConfig.IV	1. retirar o(s) item(ns) de configuração necessário(s) do repositório e colocá-lo(s) na área de trabalho dos projetistas/programadores (<i>check out</i>)			
	2. efetuar as alterações, de acordo com a "ordem de serviço"			
	3. avaliar o aspecto funcional de cada item modificado, tentando descobrir omissões ou erros na configuração que degradam os padrões de construção de software, de acordo com o processo de <i>Controle de Qualidade</i>			
	4. avaliar se a(s) nova(s) versão(ões) respeita(m) os requisitos definidos anteriormente, também com base no processo de <i>Controle de Qualidade</i>			
	5. verificar se a nova configuração a ser 'congelada' pela <i>linha de referência</i> está composta da versão mais recente dos itens de configuração determinados para a fase específica do ciclo de vida			
	6. verificar se o projeto e a documentação estão atualizados, respeitando-se os procedimentos e padrões definidos, de acordo com o processo de <i>Verificação</i>			
GConfig.V	1. armazenar o(s) item(ns) de configuração alterado(s) no repositório, uma vez aprovado(s) na revisão (<i>check in</i>)			
	2. documentar as alterações ocorridas (o que aconteceu, quem o fez, quando ocorreu, o que mais será afetado, etc.)			
	3. compilar os resultados num relatório da configuração atual do sistema, a ser armazenado na base de dados históricos, de acordo com o processo de <i>Documentação</i>			
Total (%)				

Seção 9: processo de *Controle de Qualidade*

Esta seção compreende as atividades envolvidas no processo de *Controle de Qualidade*, a saber:

- **CQual.I:** preparar para o processo de *Controle de Qualidade*
- **CQual.II:** controlar a qualidade dos produtos
- **CQual.III:** controlar a qualidade dos processos

Atividades	Tarefas	Não cumpre	Cumpre parcialmente	Cumpre totalmente
CQual.I	1. identificar as relações existentes entre o processo de <i>Controle de Qualidade</i> e os processos de <i>Verificação e Validação</i>			
	2. definir os requisitos e padrões de qualidade			
	3. definir procedimentos para identificação, coleta, arquivamento, manutenção e disponibilização dos requisitos de qualidade, tanto dos produtos quanto dos processos envolvidos no projeto			
	4. identificar ferramentas de apoio às tarefas de controle da qualidade, de acordo com o processo de <i>Infra-Estrutura</i>			
	5. documentar os resultados num plano de controle da qualidade			
	6. aprovar o plano			
CQual.II	1. garantir que todos os planos exigidos no projeto sejam documentados, mutuamente consistentes e executados quando exigidos			
	2. garantir que os produtos de software intermediários e a respectiva documentação seguiram os planos e satisfazem os requisitos especificados			
	3. garantir que o produto final, incluindo-se sua documentação, satisfaz todos os requisitos especificados pelo cliente, com base no estabelecido pelo processo de <i>Validação</i>			
	4. garantir que as medições dos produtos de software estejam de acordo com os padrões e procedimentos estabelecidos, de acordo com o processo de <i>Acompanhamento</i>			
	5. encaminhar eventuais problemas e/ou não-conformidades que não puderem ser resolvidas no escopo do processo de <i>Controle de Qualidade</i> ao processo de <i>Resolução de Problemas</i>			
CQual.III	1. garantir que todos os processos de software (primários e de apoio) seguiram os planos e respeitam os padrões			
	2. garantir que todas as práticas de Engenharia de Software, o ambiente de desenvolvimento e o ambiente de teste estejam em conformidade com os planos estabelecidos			
	3. garantir que as medições dos processos de software estejam de acordo com os padrões e procedimentos definidos, com base no estabelecido no processo de <i>Acompanhamento</i>			
	4. garantir que os times têm habilidade e conhecimento necessários para a realização das tarefas, com base nos resultados do processo de <i>Capacitação</i>			
	5. garantir que os times tenham recebido treinamentos adequados, com base nos resultados do processo de <i>Treinamento</i>			
	6. encaminhar eventuais problemas e/ou não-conformidades que não puderem ser resolvidos no escopo do processo de <i>Controle de Qualidade</i> ao processo de <i>Resolução de Problemas</i>			
Total (%)				

Seção 10: processo de *Verificação*

Esta seção compreende as atividades envolvidas no processo de *Verificação*, a saber:

- **Verif.I:** preparar para o processo de *Verificação*
- **Verif.II:** verificar os produtos
- **Verif.III:** verificar os processos

Atividades	Tarefas	Não cumpre	Cumpre parcialmente	Cumpre totalmente
Verif.I	1. identificar as relações existentes entre o processo de <i>Verificação</i> e o processo de <i>Controle de Qualidade</i>			
	2. analisar os requisitos do projeto, a fim de que riscos envolvidos, caso haja erros, sejam identificados			
	3. verificar as restrições das tecnologias a serem usadas e a disponibilidade dos recursos necessários			
	4. identificar ferramentas de apoio às tarefas de verificação, de acordo com o processo de <i>Infra-Estrutura</i>			
	5. elaborar um plano de verificação, identificando as tarefas e os recursos necessários ao processo			
	6. aprovar o plano			
Verif.II	1. revisar os requisitos do software e do sistema, a fim de verificar se são coerentes, factíveis e testáveis			
	2. revisar o projeto, a fim de verificar se ele está correto e coerente com os requisitos, implementando adequadamente as seqüências de eventos, entradas, saídas e requisitos de segurança			
	3. revisar o código, a fim de verificar se ele está de acordo com os requisitos e padrões de codificação e se é testável e correto, implementando os requisitos de segurança com métodos rigorosos			
	4. revisar os componentes e unidades de código, bem como itens de hardware e software, a fim de verificar se foram integrados completa e corretamente, de acordo com o plano de integração			
	5. revisar a documentação, a fim de verificar se ela está adequada, é completa, coerente e foi desenvolvida seguindo os procedimentos estabelecidos pelo processo de <i>Documentação</i>			
	6. revisar algum outro produto de trabalho, de acordo com o estabelecido no plano de verificação			
	7. encaminhar eventuais problemas e/ou não-conformidades que não puderem ser resolvidas no escopo do processo de <i>Verificação</i> ao processo de <i>Resolução de Problemas</i>			
	8. documentar os resultados			
Verif.III	1. revisar os procedimentos descritos nos planos, para verificar se estão adequados			
	2. verificar a implementação dos processos			
	3. verificar a definição e utilização dos padrões			
	4. verificar a condição dos profissionais para a realização das atividades, com base nos resultados dos processos de <i>Capacitação e Treinamento</i>			
	5. revisar alguma outra atividade ou tarefa, de acordo com o estabelecido no plano de verificação			
	6. encaminhar eventuais problemas e/ou não-conformidades que não puderem ser resolvidas no escopo do processo de <i>Verificação</i> ao processo de <i>Resolução de Problemas</i>			
	7. documentar os resultados			
	8. aprovar os resultados			
Total (%)				

Seção 11: processo de *Validação*
Esta seção compreende as atividades envolvidas no processo de *Validação*, a saber:

- **Valid.I:** preparar para o processo de *Validação*
- **Valid.II:** validar o produto de software

Atividades	Tarefas	Não cumpre	Cumpre parcialmente	Cumpre totalmente
Valid.I	1. identificar as relações existentes entre o processo de <i>Validação</i> e o processo de <i>Controle de Qualidade</i>			
	2. identificar ferramentas de apoio às tarefas de validação, de acordo com o processo de <i>Infra-Estrutura</i>			
	3. elaborar um plano de validação, identificando as tarefas de validação a serem realizadas e os recursos necessários para o processo			
	4. aprovar o plano			
Valid.II	1. preparar os requisitos, os casos e as especificações de teste para análise dos resultados			
	2. assegurar que o produto reflete os requisitos particulares para um uso específico do software			
	3. conduzir testes considerando testes de estresse, de limite e de entradas específicas, avaliando também a habilidade do produto para isolar e minimizar os efeitos dos erros			
	4. conduzir testes com usuários representativos para avaliar se eles conseguem realizar suas tarefas usando o produto;			
	5. testar o produto apropriadamente no seu ambiente de operação			
	6. encaminhar eventuais problemas e/ou não-conformidades que não puderem ser resolvidas no escopo do processo de <i>Validação</i> ao processo de <i>Resolução de Problemas</i>			
	7. documentar os resultados			
	6. aprovar os resultados			
Total (%)				

Seção 12: processo de *Resolução de Problemas*

Esta seção compreende as atividades envolvidas no processo de *Resolução de Problemas*, a saber:

- **RProb.I:** preparar para o processo de *Resolução de Problemas*
- **RProb.II:** resolver os problemas e/ou não-conformidades

Atividades	Tarefas	Não cumpre	Cumpre parcialmente	Cumpre totalmente
RProb.I	1. definir mecanismos para categorizar e priorizar os problemas			
	2. documentar os resultados num plano de resolução de problemas, garantindo que após sua detecção, os problemas sejam rapidamente notificados e as causas sejam identificadas, analisadas e, quando possível, eliminadas			
	3. aprovar o plano			
RProb.II	1. preparar um relatório sobre cada problema e/ou não-conformidade, que os descreva e que, fazendo uso dos mecanismos previamente definidos, possa categorizá-los e priorizá-los			
	2. de acordo com a prioridade estabelecida, fazendo uso de pessoal capacitado e em conjunto com o processo de <i>Gerência</i> : <ul style="list-style-type: none">• analisar o problema e/ou não-conformidade• solucionar o problema e/ou não-conformidade			
	3. atualizar o relatório, na medida em que as tarefas anteriores forem sendo concluídas			
	4. encaminhar o relatório ao processo responsável pelo envio do problema e/ou não-conformidade ao processo de <i>Resolução de Problemas</i>			
Total (%)				

Seção 13: processo de *Gerência*

Esta seção compreende as atividades envolvidas no processo de *Gerência*, a saber:

- **Gerem.I:** preparar para o processo de *Gerência*
- **Gerem.II:** administrar a execução de um processo

Atividades	Tarefas	Não cumpre	Cumpre parcialmente	Cumpre totalmente
Gerem.I	1. estabelecer o negócio da empresa (visão e cultura da organização) e os papéis iniciais para os processos organizacionais			
	2. determinar os requisitos dos processos a serem executados no ambiente da empresa			
	3. estabelecer o escopo de cada processo e as inter-relações entre os processos			
	4. preparar, juntamente com os processos de <i>Capacitação</i> , <i>Melhoria</i> e <i>Infra-Estrutura</i> , um roteiro de execução dos processos, com base no escopo de cada um deles, que contenha cronogramas, estimativas de esforço, recursos necessários, alocação de tarefas, determinação de responsabilidades, quantificação dos riscos associados a cada processo, medições e custos			
Gerem.II	1. monitorar a execução do processo, inteirando-se do progresso das atividades			
	2. analisar e solucionar problemas, em conjunto com o processo de <i>Resolução de Problemas</i>			
	3. assegurar que as melhorias propostas pelo processo de <i>Melhoria</i> tenham sido colocadas em prática, favorecendo-as e monitorando sua execução, a fim de garantir que não sejam introduzidas inconsistências em outras atividades			
	4. atualizar o roteiro de execução dos processos, a fim de que ele reflita a exata situação do que ocorre na empresa			
Total (%)				

Seção 14: processo de *Capacitação*

Esta seção compreende as atividades envolvidas no processo de *Capacitação*, a saber:

- **Capac.I:** iniciar o processo de *Capacitação*
- **Capac.II:** determinar as capacidades dos profissionais

Atividades	Tarefas	Não cumpre	Cumpre parcialmente	Cumpre totalmente
Capac.I	1. selecionar e adequar as normas, os métodos, os padrões, os procedimentos técnicos, as linguagens de programação e as ferramentas, com base no negócio da empresa, em conjunto com o estabelecido no processo de <i>Gerência</i>			
	2. fornecer aos indivíduos da organização visão e cultura do processo de produção de software, de forma a capacitá-los a trabalhar cooperativa e eficientemente (cartilhas, palestras, discussões, encontros, etc.), também em conjunto com o processo de <i>Gerência</i>			
	3. elaborar um plano para obter informações periódicas sobre os times de trabalho, tais como: número de times de trabalho, número de membros em cada time, formação e experiência de cada membro, capacidades técnica e gerencial dos profissionais e dos times, etc., através de entrevistas, observações, medições, etc., estabelecendo, inclusive, mecanismos de graduação dos resultados, em conjunto com os processos de <i>Melhoria</i> e de <i>Gerência</i>			
	4. definir ferramentas e/ou métodos para a coleta de dados, de acordo com o especificado no plano e com base no processo de <i>Infra-Estrutura</i>			
Capac.II	1. coletar os dados identificados no plano			
	2. determinar as capacidades dos profissionais, com base nos dados coletados e segundo mecanismos de quantificação e qualificação definidos no plano			
	3. associar um perfil a cada um			
	4. documentar os resultados num relatório de capacitação dos profissionais			
Total (%)				

Seção 15: processo de *Melhoria*

Esta seção compreende as atividades envolvidas no processo de *Melhoria*, a saber:

- **Melh.I:** preparar para o processo de *Melhoria*
- **Melh.II:** melhorar os processos

Atividades	Tarefas	Não cumpre	Cumpre parcialmente	Cumpre totalmente
Melh.I	1. definir possíveis objetivos específicos de melhoria para cada processo (sobretudo quanto aos recursos humanos e computacionais), que possam refletir a melhoria do processo global da empresa, em conjunto com o estabelecido no processo de <i>Gerência</i>			
	2. selecionar/definir métricas (objetivas e subjetivas) adequadas para se realizar medições, com base nos objetivos estabelecidos e também em conjunto com o processo de <i>Gerência</i>			
	3. identificar procedimentos e ferramentas para coleta das métricas, sem sobrecarregar os times em suas tarefas e sem afastá-los de suas atividades principais			
	4. definir instrumentos de avaliação (entrevistas, observações, etc.) do nível de satisfação tanto dos indivíduos da empresa quanto dos clientes			
	5. definir mecanismos para avaliar, quantitativa e qualitativamente, todos os resultados obtidos			
	6. compilar os resultados numa estratégia para coleta e avaliação de métricas para melhoria de processo			
Melh.II	1. coletar as métricas indicadas na estratégia de coleta e avaliação de métricas, como parte integrante do projeto			
	2. analisar os resultados obtidos, tanto das métricas coletadas quanto dos relatórios que servem de entrada para esta atividade			
	3. documentar e armazenar cada medição, desde a coleta da métrica até a realização da modificação, nos casos em que ela ocorrer, sempre com a ciência e com a garantia do processo de <i>Gerência</i>			
	4. realizar estudos empíricos envolvendo as medições dos processos de software e os dados armazenados na “base de dados históricos”			
	5. preparar um plano de melhoria, para cada processo, caracterizando sua situação atual e sugerindo possíveis mudanças (inclusive na infra-estrutura ou baseados em treinamento), com base na análise dos resultados obtidos e considerando o impacto da mudança nos demais processos e para o negócio da empresa			
Total (%)				

Seção 16: processo de *Infra-Estrutura*

Esta seção compreende as atividades envolvidas no processo de *Infra-Estrutura*, a saber:

- **IEstrut.I:** definir a infra-estrutura e sua instalação
- **IEstrut.II:** instalar e manter a infra-estrutura

Atividades	Tarefas	Não cumpre	Cumpre parcialmente	Cumpre totalmente
IEstrut.I	1. definir a infra-estrutura necessária, em conjunto com o estabelecido no processo de <i>Gerência</i>			
	2. planejar a aquisição (quando necessária), a instalação e o funcionamento da infra-estrutura			
	3. definir a estrutura e o acesso à “base de dados históricos” da empresa, a qual deverá servir para o registro das terminologias e dos eventos relacionados ao projeto, armazenando, assim, documentos importantes para estimativas e melhorias			
IEstrut.II	1. adquirir (quando preciso), instalar e/ou renovar a infra-estrutura			
	2. monitorar a utilização da infra-estrutura			
	3. emitir relatório periódico sobre a utilização e a situação da infra-estrutura, propondo mudanças, quando necessárias			
Total (%)				

Seção 17: processo de *Treinamento*

Esta seção compreende as atividades envolvidas no processo de *Treinamento*, a saber:

- **Trein.I:** preparar para o processo de *Treinamento*
- **Trein.II:** realizar o treinamento

Atividades	Tarefas	Não cumpre	Cumpre parcialmente	Cumpre totalmente
Trein.I	1. identificar, em conformidade com o estabelecido nos processos de <i>Capacitação</i> e <i>Infra-Estrutura</i> , os recursos técnicos e humanos necessários para a realização das atividades de treinamento			
	2. definir um plano de treinamento, estabelecendo cronogramas, metodologias e procedimentos para avaliação do aproveitamento			
Trein.II	1. elaborar o material de treinamento, seguindo o plano			
	2. treinar os profissionais			
	3. avaliar o aproveitamento dos participantes, também seguindo os critérios de avaliação estabelecidos no plano			
	4. atualizar o perfil de cada profissional, em conjunto com o processo de <i>Capacitação</i>			
	5. preparar um relatório com os resultados do treinamento, descrevendo o aproveitamento e sugerindo modificações			
Total (%)				