

UNICAMP  
BIBLIOTECA CENTRAL  
SEÇÃO CIRCULANTE

Este exemplar corresponde à redação final da Tese/Dissertação devidamente corrigida e defendida por: <u>Ademir Morgenstern Padilha</u>
e aprovada pela Banca Examinadora. Campinas, <u>06</u> de <u>Junho</u> de <u>2001</u>
 COORDENADOR DE PÓS-GRADUAÇÃO CPG-IC

UNICAMP  
BIBLIOTECA CENTRAL  
SEÇÃO CIRCULANTE

**MAPS-15504**

Uma metodologia de avaliação de processo de software para  
o ambiente ExpSEE baseada no ISO/IEC TR 15504

Ademir Morgenstern Padilha

Dissertação de Mestrado

# MAPS-15504

## Uma metodologia de avaliação de processo de *software* para o ambiente ExpSEE baseada no ISO/IEC TR 15504

Ademir Morgenstern Padilha

setembro de 2000

### **Banca examinadora**

Prof. Dr. Jacques Wainer – (Orientador)

Prfa. Dra. Sílvia Regina Vergilio – UFPR

Prof. Dr. Mário Lúcio Cortes – UNICAMP

Prof. Dra. Eliane Martins - UNICAMP

**FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DO IMECC DA UNICAMP**

Padilha, Ademir Morgenstern

P134m        MAPS 15504: uma metodologia de avaliação de processo de software  
para o ExpSEE baseada no ISO/IEC TR 15504 / Ademir Morgenstern Padilha -  
- Campinas, [S.P. :s.n.], 2000.

Orientadores : Jacques Wainer; Itana Maria Souza Gimenez

Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de  
Computação.

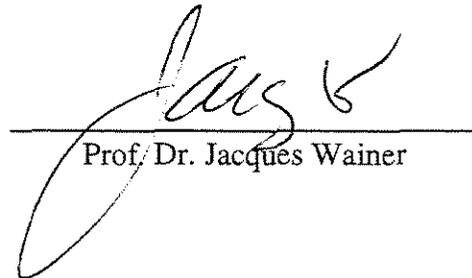
1. Engenharia de software. 2. Controle de qualidade – Manuais, guias,  
etc. I. Wainer, Jacques. II. Gimenez, Itana Maria Souza III. Universidade  
Estadual de Campinas. Instituto de Computação. IV. Título.

MAPS-15504

Uma metodologia de avaliação de processo de software para o ambiente ExpSEE baseada no ISO/IEC TR 15504

Este exemplar corresponde à redação final da dissertação devidamente corrigida e defendida por Ademir Morgenstern Padilha e aprovada pela banca.

Campinas, 23 de outubro de 2000.

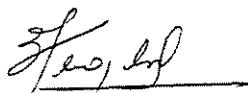


Prof. Dr. Jacques Wainer

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

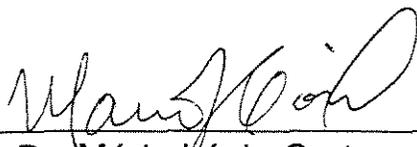
## TERMO DE APROVAÇÃO

Tese defendida e aprovada em 25 de setembro de 2000, pela Banca Examinadora composta pelos Professores Doutores:



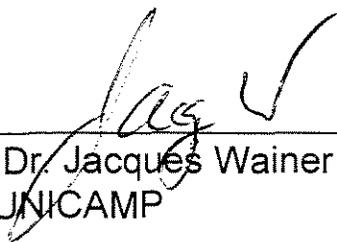
---

Prof. Dra. Silvia Regina Vergilio  
UFPR



---

Prof. Dr. Mário Lúcio Cortes  
IC – UNICAMP



---

Prof. Dr. Jacques Wainer  
IC – UNICAMP

© Ademir Morgenstern Padilha, 2000.  
Todos os direitos reservados.

## Resumo

Devido às crescentes exigências por qualidade de *software* a comunidade de engenharia de *software* tem produzido diversas normas e apresentado diversas abordagens sobre a qualidade de *software*. Grande parte destas normas são aplicadas ao processo de *software*, dentre elas destacam-se pela larga utilização a ISO 9000-3, a ISO 12207, o CMM e o ISO/IEC TR 15504 como resultado dos trabalhos do projeto SPICE. Outro resultado das pesquisas da comunidade de engenharia de *software* são os ambientes de engenharia de *software* centrados no processo (PSEE), os quais visam à automação do processo de *software*.

Estas duas idéias, normatização e automação do processo de *software*, compõem a idéia central deste trabalho: avaliação da qualidade do processo de *software* através de um ambiente automatizado. Para isto foi desenvolvida uma metodologia de avaliação de processo de *software* com base no ISO/IEC TR 15504, denominada MAPS-15504. A metodologia de avaliação de processo de *software* foi implementada no ambiente do ExpSEE, um ambiente experimental desenvolvido no Departamento de Informática (DIN) da Universidade Estadual de Maringá (UEM), e utilizada na avaliação do processo de *software* da Empresa X Ltda, uma empresa fictícia criada com base em um padrão de empresas de médio porte do relatório do Programa Brasileiro de Qualidade e Produtividade e dos *trials* do projeto SPICE.

A metodologia proposta MAPS-15504 traz grandes benefícios para o processo de avaliação, pois: define um processo de avaliação claro e passível de automação, introduz tarefas específicas para a estruturação da avaliação, introduz uma tarefa para coleta de documentos, introduz o gerenciamento automatizado do processo através do controle sobre o andamento e o estado de cada tarefa a qualquer tempo, permite a reutilização do processo de avaliação, garante a execução da tarefa somente se ela puder ser executada através de pré e pós-condições, automatiza os *checklists* que podem ser programados na tarefa, gerando automaticamente os resultados da avaliação com controle de acesso aos resultados.

## Abstract

The increasing requirements for *software* quality have lead the *software* engineering community to produce several standards to guide the quality of *software* products and processes. Amongst the process standards, it can be highlighted the following ones due to their wide use: ISO 9000-3, ISO 12207, CMM and ISO/IEC TR 15504. An additional important research result of the *software* engineering community is Process-centred *software* engineering environment (PSEE). These environments aim at providing integrated *software* process automation.

Process standards and PSEE compose the central subject of the work presented in this dissertation that is the evaluation of the quality of *software* processes through an automated environment. In order to achieve this a method of *software* process evaluation, called MAPS-15504, based on the ISO/IEC TR 15504, was developed. MAPS-15504 was applied to the hypothetical Company X and implemented in the ExpPSEE environment. ExpPSEE is an experimental PSEE developed in the Informatics Department (DIN) of the Universidade Estadual de Maringá (UEM). Company X was idealized based on medium sized companies as established in the Brazilian Program for Quality and Productivity and the trials of the SPICE Project.

MAPS-15504, the proposed methodology, brought several benefits for the evaluation process which include: definition of a clear and executable process; introduction of specific tasks for the evaluation structure, introduction of a well-defined task for document collection, management of process through the control of status of each task at any time; reuse of evaluation processes; guarantee that tasks satisfy their pre and post-conditions, and implementation of checklists, thus automatically generating the evaluation results.

## **Agradecimentos**

Primeiramente agradeço a Deus, por estar sempre presente, principalmente nas horas mais difíceis e solitárias desta jornada.

Agradeço a Profa. Dra. Itana Maria Souza Gimenes, pela orientação, dedicação e o muito que me ensinou.

Agradeço ao Prof. Dr. Jacques Wainer pela orientação e pelo empenho em viabilizar a conclusão deste trabalho.

Agradeço a minha esposa Silvana e minha filha Vitória que compreenderam minhas ausências e sempre me apoiaram neste trabalho.

Agradeço ao CESULON pela compreensão neste ano e em anos anteriores.

Agradeço ao projeto Genesis / GeNorP, na figura de sua coordenadora Prof. Cleusa Rocha Asanome que sempre compreendeu a importância deste trabalho.

Agradeço ao grupo de engenharia de *software* da Universidade Estadual de Maringá - UEM, pela ajuda que dispensaram quando do trabalho junto ao ExpSEE.

Agradeço ao CNPq, que viabilizou este estudo.

Agradeço aos amigos e parentes, em especial ao "tio Arildo" e família, pelo apoio, compreensão e pelo carinho que me dispensaram.

E a todos que direta ou indiretamente contribuíram para a conclusão deste trabalho.

## Lista de Figuras

Figura 1.1 -	Avaliação de processo de <i>software</i> [ISO99-1].....	2
Figura 2.1 -	Fatores de Qualidade [GILL97].....	7
Figura 2.2 -	Modelo hierárquico de Boehm [GILL97].....	8
Figura 2.3 -	Ciclo PDCA [GILL97].....	13
Figura 3.1 -	Estrutura da norma ISO 9000-3 [CABR98].....	16
Figura 3.2 -	Visões da ISO/IEC 12207 [SALV96].....	18
Figura 4.1 -	Arquitetura de um PSEE [BOOC99].....	32
Figura 4.2 -	Diagrama de Pacotes do Ambiente ExpPSEE [CARN99].....	36
Figura 4.3 -	Diagrama de classes do gerenciador de processos do ExpPSEE [CARN99].....	38
Figura 4.4 -	Diagrama de Estados das Tarefas [CARN99].....	41
Figura 5.1 -	Arquitetura geral do processo de avaliação .....	45
Figura 5.2 -	Tarefas que compõem a tarefa TA1 Definir entradas.....	46
Figura 5.3 -	Tarefas que compõem a tarefa TA2 Processo de avaliação... ..	49
Figura 5.4 -	Relacionamento entre os indicadores da avaliação e a capacidade do processo [ISO99-5].....	63
Figura 6.1 -	Organograma da divisão de informática da Empresa X.....	80
Figura 6.2 -	Ciclo de vida de prototipação [PRES95].....	82
Figura 6.3 -	Documento AR2 Descrição inicial.....	84
Figura 6.4a -	Documento AR3 Entradas da avaliação.....	85
Figura 6.4b -	Documento AR3 Entradas da avaliação.....	87
Figura 6.5 -	Documento AR4 Instâncias de processos de <i>software</i> .....	89
Figura 6.6 -	Documento AR5 Avaliadores.....	91
Figura 6.7 -	Documento AR6 Registro de treinamento.....	93
Figura 6.8a -	Documento AR7 Planejamento da avaliação.....	94
Figura 6.8b -	Documento AR7 Planejamento da avaliação.....	95
Figura 6.9a -	Documento AR8 Checklist – práticas de gerenciamento.....	96
Figura 6.9b -	Documento AR8 Checklist – práticas básicas.....	97
Figura 6.10a -	Documento AR9 Lista de documentos.....	99
Figura 6.10b -	Documento AR9 Lista de documentos.....	100
Figura 6.10c -	Documento AR9 Lista de documentos.....	101
Figura 6.10d -	Documento AR9 Lista de documentos.....	102
Figura 6.10e -	Documento AR9 Lista de documentos.....	103
Figura 6.10f -	Documento AR9 Lista de documentos.....	104

Figura 6.10g-	Documento AR9 Lista de documentos.....	105
Figura 6.11a -	Documento AR8 Checklist – práticas de gerenciamento (respondido).....	107
Figura 6.11b -	Documento AR8 Checklist – práticas básicas (respondido)..	108
Figura 6.12 -	Documento AR10 Medidas das instâncias para práticas básicas.....	110
Figura 6.13 -	Documento AR10 Medidas das instância para práticas de gerenciamento.....	111
Figura 6.14 -	Documento AR11 Perfil do processo de <i>software</i> .....	112
Figura 6.15 -	Arquitetura do processo de avaliação do processo de <i>software</i> .....	114
Figura 6.16 -	Instanciação da arquitetura do processo de <i>software</i> .....	115
Figura 6.17 -	Janela Start up.....	116
Figura 6.18 -	Instância de processo instanciada, programada e alocada.....	117
Figura 6.19 -	Agenda de tarefas.....	118
Figura A.1 -	Modelo do documento AR2 Descrição inicial.....	130
Figura A.2 -	Modelo do documento AR3 Entradas da avaliação.....	131
Figura A.3 -	Modelo do documento AR3 Entradas da avaliação.....	132
Figura A.4 -	Modelo do documento AR4 Instâncias dos processos de <i>software</i> .....	133
Figura A.5 -	Modelo do documento AR5 Avaliadores.....	134
Figura A.6 -	Modelo do documento AR6 Registro de treinamento.....	135
Figura A.7a -	Modelo do documento AR7 Planejamento da avaliação.....	136
Figura A.7b -	Modelo do documento AR7 Planejamento da avaliação.....	137
Figura A.8 -	Modelo do documento AR8 Checklist – práticas de gerenciamento.....	138
Figura A.9 -	Modelo do documento AR8 Checklist – práticas básicas.....	139
Figura A.10a -	Modelo do documento AR9 Lista de documentos.....	140
Figura A.10b -	Modelo do documento AR9 Lista de documentos.....	141
Figura A.11a -	Modelo do documento AR8 Checklist – práticas de gerenciamento (respondido).....	142
Figura A.11b -	Modelo do documento AR8 Checklist – práticas básicas (respondido).....	143
Figura A.12 -	Modelo do documento AR10 Medidas das instâncias para práticas básicas.....	144
Figura A.13 -	Modelo do documento AR10 Medidas das instâncias para práticas de gerenciamento.....	145
Figura A.14 -	Modelo do documento AR11 Medidas do processo de	146

	<i>software</i> .....	
Figura B.1 -	Janela Work context para tarefa TA2.1 Definir instâncias.....	147
Figura B.2 -	Janela Work context para tarefa TA2.2 Definir avaliadores..	150
Figura B.3 -	Janela Work context para tarefa TA2.3 Treinar avaliadores..	153
Figura B.4 -	Janela Work context para tarefa TA2.4 Planejar avaliação....	156

## Lista de Tabelas

Tabela 2.1 - Atributos e sub atributos da qualidade propostos por Gilb [GILL97].....	9
Tabela 2.2 - Comparação das principais idéias de Crosby, Deming e Juran [GILL97].....	11
Tabela 3.1 - Certificação do sistema da qualidade [MCT00].....	17
Tabela 3.2 - Indicadores e Metas da Qualidade e Produtividade em Software [MCT99].....	17
Tabela 3.3 - Conhecimento de normas para qualidade [MCT00].....	19
Tabela 3.4 - níveis de maturidade e áreas chave de processos do CMM [SALV96].....	20
Tabela 3.5 - Conhecimento dos modelos CMM e SPICE [MCT00].....	21
Tabela 3.6 - Modelo de Referência de Processos do SPICE [ISO99-2].....	23
Tabela 3.7 - Níveis de Capacitação e Atributos de processos do SPICE [ISO99-2].....	24
Tabela 3.8 - Comparativo entre os modelos [SALV96].....	25
Tabela 4.1 - Classificação de linguagens de processos de <i>software</i> [FINK94]....	34
Tabela 7.1 - Características de uma ferramenta para automação do processo de <i>software</i> .....	124

## Lista de Abreviaturas.

CASE	<i>Computer Aided Software Engineering</i> – Engenharia de Software Apoiada por Computador
CITS	Centro Internacional de Tecnologias de <i>Software</i>
COCOMO	<i>CO</i> nstructive <i>CO</i> ast <i>MO</i> del – projeto baseado nos trabalhos de Bohem, 1981
COQUAMO	<i>CO</i> nstructive <i>QU</i> ality <i>MO</i> del – projeto executado por Kitchenham em 1989 no Reino Unido
DoD	<i>Department of Defense</i> – Departamento de Defesa
ExpPSEE.	<i>Experimental PSEE</i> – PSEE Experimental.
ISO	<i>International Organization for Standardization</i> – Organização Internacional para Padronização.
JUSE	<i>Union of Japanese Scientists and Engineers</i> - União Japonesa de Cientistas e Engenheiros.
PBQP	Programa Brasileiro de Qualidade e Produtividade.
PDCA	<i>Plan-Do-Check-Act</i> – Planejar-Executar-Verificar-Agir
PQPS	Programa de Qualidade e produtividade em <i>Software</i> .
PSEE	<i>Process Software Engineering Environment</i> – Ambiente de Engenharia de <i>Software</i> Voltado ao Processo.
QIP	<i>Quality Improvement Program</i> – Programa de Melhoria da Qualidade
QMS	<i>Quality Management System</i> – Sistema de Gerenciamento da Qualidade
SPC	<i>Statistical Process Control</i> - Controle Estatístico do Processo
SPICE	<i>Software Process Improvement and Capability dEtermination</i> – Determinação de Capacidade e Melhoria do Processo de <i>Software</i>
SQA	<i>Software Quality Assurance</i> - Garantia de Qualidade de <i>Software</i>
TQM	<i>Total Quality Management</i> – Gerenciamento da Qualidade Total
SEAL	<i>Software Engineering Applications Laboratory</i> – Laboratório de Aplicações em Engenharia de <i>Software</i> .
CTI	Fundação Centro Tecnológico para Informática.
CMM	<i>Capability Marurity Model</i> – Modelo de Capacidade e Maturidade.
SEI	<i>Software Engineering Institute</i> – Instituto de Engenharia de <i>Software</i> .

# Conteúdo

RESUMO.....	VII
ABSTRACT .....	VIII
AGRADECIMENTOS .....	IX
LISTA DE FIGURAS.....	X
LISTA DE TABELAS.....	XIII
LISTA DE ABREVIATURAS.....	XIV
CONTEÚDO.....	XV
CAPÍTULO 1.....	1
INTRODUÇÃO.....	1
CAPÍTULO 2.....	6
QUALIDADE DE <i>SOFTWARE</i> .....	6
2.1 – CONCEITO DE QUALIDADE .....	6
2.1.1 - O trabalho de McCall.....	6
2.1.2 - O trabalho de Boehm .....	7
2.1.3 - O trabalho de Gilb .....	8
2.2 – CONCEITO DE GERENCIAMENTO DA QUALIDADE .....	10
2.2.1 - Os trabalhos de Deming, Juran e Crosby.....	10
2.2.2 - Os principais termos do gerenciamento da qualidade.....	12
2.3 - CONSIDERAÇÕES FINAIS.....	13
CAPÍTULO 3.....	15
MODELOS DE QUALIDADE DE <i>SOFTWARE</i> .....	15
3.1 – A ISO 9000-3.....	15
3.2 – A NORMA ISO/IEC 12207 .....	18
3.3 – O MODELO CMM: <i>CAPABILITY MATURITY MODEL</i> .....	19
3.4 – O PROJETO SPICE.....	21

3.5 – COMPARAÇÃO ENTRE OS MODELOS.....	24
3.6 - CONSIDERAÇÕES FINAIS.....	27
<b>CAPÍTULO 4.....</b>	<b>28</b>
<b>O PROCESSO DE SOFTWARE.....</b>	<b>28</b>
4.1 – CONCEITOS BÁSICOS .....	28
4.1.1 – Modelos de ciclo de vida versus modelos de processo de <i>software</i> .....	29
4.1.2 – Requisitos para Modelagem do Processo de Software.....	29
4.2 – AMBIENTES DE ENGENHARIA DE <i>SOFTWARE</i> .....	31
4.3 – O AMBIENTE EXPSEE.....	34
4.3.1 – Arquitetura do Ambiente ExpSEE .....	35
4.3.2 – O Gerenciador de Processos .....	36
4.3.3 – Definição de Arquiteturas e Instanciação de Processos .....	39
4.3.4 – Diagrama de Estados das Tarefas .....	40
4.4 – COMENTÁRIOS FINAIS .....	41
<b>CAPÍTULO 5.....</b>	<b>43</b>
<b>MAPS-15504: UMA METODOLOGIA DE AVALIAÇÃO DE PROCESSO DE SOFTWARE</b>	
<b>BASEADA NO ISO/IEC TR 15504.....</b>	<b>43</b>
5.1 O PROCESSO DE AVALIAÇÃO.....	43
5.1.1 Detalhamento da tarefa Definir entradas .....	46
5.1.2 Explosão da tarefa Processo de avaliação .....	47
5.1.3 Tarefa TA3 Apresentar resultados.....	51
5.2 DESCRIÇÃO DAS TAREFAS .....	52
5.2.1 TA1.1 Descrever avaliação .....	52
5.2.2 TA1.2 Definir entradas da avaliação .....	54
5.2.3 TA2.1 Definir instâncias .....	57
5.2.4 TA2.2 Definir avaliadores.....	59
5.2.5 TA2.3 Treinar avaliadores.....	60
5.2.6 TA2.4 Planejar avaliação .....	62
5.2.7 TA2.5 Coletar documentos .....	65
5.2.8 TA2.6 Coletar informações.....	67
5.2.9 TA 2.7 Verificar informações .....	70
5.2.10 TA2.8 Determinar medidas.....	72
5.2.11 TA 2.9 Consolidar medidas .....	73
5.2.12 TA2.10 Validar medidas .....	75

5.2.13 TA3 Apresentar resultados.....	77
5.3 CONSIDERAÇÕES FINAIS .....	79
<b>CAPÍTULO 6.....</b>	<b>80</b>
<b>ESTUDO DE CASO .....</b>	<b>80</b>
6.1 APRESENTAÇÃO DO CONTEXTO .....	80
6.1.1 A Empresa X Ltda .....	80
6.1.2 Tecnologias utilizadas .....	81
6.2 APLICAÇÃO DA METODOLOGIA MAPS-15504 .....	83
6.2.1 TA1.1 Descrever avaliação .....	83
6.2.2 TA1.2 Definir entradas da avaliação .....	84
6.3 A AVALIAÇÃO UTILIZANDO O EXPSEE.....	113
6.3.1 Definição da arquitetura .....	113
6.3.2 Instanciação.....	114
6.3.3 Programação.....	115
6.3.4 Alocação .....	116
6.3.5 Tarefa no estado pronta .....	117
6.3.6 Tarefa no estado executando .....	118
6.3.7 Tarefa no estado finalizada .....	118
6.4 CONSIDERAÇÕES FINAIS .....	118
6.4.1 Vantagens da metodologia MAPS-15504.....	119
6.4.2 Limitações na utilização da metodologia MAPS-15504.....	119
6.4.3 Vantagens da automação da metodologia MAPS-15504 no ambiente do ExpPSEE.....	120
6.4.4 Limitações da automação da metodologia MAPS-15504 no ambiente do ExpPSEE.....	121
<b>CAPÍTULO 7.....</b>	<b>122</b>
<b>CONCLUSÕES E TRABALHOS FUTUROS .....</b>	<b>122</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>126</b>
<b>ANEXO A - MODELOS DE ARTEFATOS.....</b>	<b>130</b>
A.1 TA1.1 DESCREVER AVALIAÇÃO .....	130
A.2 TA1.2 DEFINIR ENTRADAS DA AVALIAÇÃO .....	131
A.4 TA2.2 DEFINIR AVALIADORES.....	134
A.5 TA2.3 TREINAR AVALIADORES .....	135
A.6 TA2.4 PLANEJAR AVALIAÇÃO .....	136
A.7 TA2.5 COLETAR DOCUMENTOS .....	140

A.8 TA2.6 COLETAR INFORMAÇÕES .....	142
A.9 TA2.8 DETERMINAR MEDIDAS.....	144
A.10 TA2.9 CONSOLIDAR MEDIDAS.....	146
<b>ANEXO B – PROGRAMAÇÃO DAS TAREFAS.....</b>	<b>147</b>
B.1 PROGRAMAÇÃO DA TAREFA TA2.1 DEFINIR INSTÂNCIAS .....	147
B.2 PROGRAMAÇÃO DA TAREFA TA2.2 DEFINIR AVALIADORES .....	150
B.3 PROGRAMAÇÃO DA TAREFA TA2.3 TREINAR AVALIADORES.....	152
B.4 PROGRAMAÇÃO DA TAREFA TA2.4 PLANEJAR AVALIAÇÃO.....	155
<b>ANEXO C – DESCRIÇÃO DA EXECUÇÃO NO EXPSEE .....</b>	<b>159</b>
C.1 TA1.1 DESCREVER AVALIAÇÃO.....	159
C.2 TA1.2 DEFINIR ENTRADAS DA AVALIAÇÃO.....	159
C.3 TA2.1 DEFINIR INSTÂNCIAS .....	160
C.4 TA2.2 DEFINIR AVALIADORES.....	161
C.5 TA2.3 TREINAR AVALIADORES.....	161
C.6 TA2.4 PLANEJAR AVALIAÇÃO .....	162
C.7 TA2.5 COLETAR DOCUMENTOS .....	162
C.8 TA2.6 COLETAR INFORMAÇÕES.....	162
C.9 TA2.7 VERIFICAR INFORMAÇÕES .....	163
C.10 TA2.8 DETERMINAR MEDIDAS.....	163
C.11 TA2.9 CONSOLIDAR MEDIDAS .....	164
C.12 TA2.10 VALIDAR MEDIDAS .....	164
C.13 TA3 APRESENTAR RESULTADOS.....	164
<b>ANEXO D – ARTIGO ACEITO NO WQS2000.....</b>	<b>166</b>
<b>ABSTRACT.....</b>	<b>166</b>
<b>RESUMO.....</b>	<b>166</b>
<b>1. INTRODUÇÃO.....</b>	<b>167</b>
<b>2. DESCRIÇÃO DA MAPS-15504.....</b>	<b>168</b>
<b>2.1 A TAREFA TA1 DEFINIR ENTRADAS.....</b>	<b>169</b>
<b>2.2 A TAREFA TA2 DEFINIR PROCESSO DE AVALIAÇÃO.....</b>	<b>171</b>
<b>2.3 A TAREFA TA3 APRESENTAR RESULTADOS.....</b>	<b>175</b>

<b>3. CONCLUSÕES.....</b>	<b>175</b>
<b>4. BIBLIOGRAFIA.....</b>	<b>178</b>

# Capítulo 1.

## Introdução

A busca pela qualidade tem sido objeto de intensas pesquisas pela comunidade de engenharia de *software*. O termo qualidade possui várias definições na literatura as quais variam de acordo com a abordagem [GILL97]. Segundo a ISO<sup>1</sup> qualidade é a totalidade dos atributos e características de um produto ou serviço que lhe confere a capacidade de satisfazer as suas necessidades explícitas e implícitas [ISO86].

Os padrões para garantia de qualidade surgiram na década de 70 com a formação das equipes de SQA<sup>2</sup>. No início, certificava-se a qualidade do produto a um custo elevadíssimo, seja pelo trabalho de verificação, seja pelo desperdício representado pela detecção e eliminação dos defeitos ou partes defeituosas. No final dessa década, Bohem e McCall propuseram uma classificação para os fatores que afetam a qualidade de um produto de *software*, descrito em por [GILL97]. O trabalho destes autores fundamentou outros modelos como o projeto COQUAMO<sup>3</sup> e suas ferramentas COQUAMO-1 (predição), COQUAMO-2 (monitoramento) e COQUAMO-3 (avaliação), chamadas anteriormente de COCOMO<sup>4</sup>. Os modelos da ISO também utilizaram os conceitos de Bohem e McCall como base para a norma de qualidade de produto de *software*, a ISO 9126 [ABNT94a].

A partir da definição de critérios de qualidade para o produto de *software*, evoluiu-se para a noção de um sistema da qualidade, envolvendo toda a organização no esforço pela qualidade com forte ênfase em normas (padronização). É neste contexto que nasceu a série de normas ISO 9000 [ABNT94b, ABNT94c, ABNT94d, ABNT94e, ABNT94f]. Esta série é composta de um conjunto de normas internacionais para sistemas de gerenciamento da qualidade e foi o primeiro acordo sobre os requisitos básicos para sistemas da qualidade com alcance mundial. A série de normas ISO 9000 possui diretrizes genéricas que aplicam-se a qualquer produto ou serviço, porém algumas adaptações foram necessárias para os produtos de *software*, o que resultou na criação da norma ISO 9000-3 [ABNT96], um guia para aplicação da norma ISO 9001 [ABNT94c] para produtos de *software*.

Através do conceito de sistema da qualidade, procurou-se dar ênfase tanto ao produto de *software* quanto ao processo de desenvolvimento. Com a necessidade de um maior conhecimento e controle do processo de *software*, pela sofisticação dos meios de produção (ex: metodologias, ambientes e ferramentas), em 1991, o comitê de engenharia de *software* da ISO aprovou a realização de estudos para analisar as necessidades de um padrão para avaliação do processo de desenvolvimento de *software* [ISO99]. Foi a partir

---

<sup>1</sup> ISO - International Organization for Standardization – Organização Internacional para Padronização.

<sup>2</sup> SQA - Software Quality Assurance - Garantia de Qualidade de Software

<sup>3</sup> COQUAMO - CONstrutive QUALity MOdel – projeto executado por Kitchenham em 1989 no Reino Unido.

<sup>4</sup> COCOMO - CONstructive COst MOdel – projeto baseado nos trabalhos de Bohem, 1981.

deste estudo que criou-se, em 1993, o projeto SPICE<sup>5</sup>. O objetivo principal deste projeto é produzir normas que orientem a avaliação de processos de *software* visando a melhoria contínua do processo e a determinação da sua capacidade. Para cumprir esta diretriz, o projeto SPICE baseou-se no que havia de melhor em modelos pré-existentes voltados para avaliação de processos, dentre eles: Trillium, STD (*Software Technology Diagnostic*), Bootstrap, TQM (*Total Quality Management*), Malcolm Baldrige e CMM (*Capability Maturity Model*) [ISO99-1]. O CMM [HUMP89], modelo do Instituto de Engenharia de Software (SEI) da Carnegie de Melon University, que é o mais utilizado pela indústria de *software* dos Estados Unidos, foi o que mais influenciou no projeto SPICE.

O resultado da conciliação dessas normas gerou uma estrutura para avaliação de processos de *software* com os seguintes propósitos:

- a) entender o estado dos processos da organização para a melhoria de processo;
- b) determinar a adequação dos processos da organização a um requisito particular ou classe de requisitos;
- c) determinar a adequação dos processos de outra organização para um contrato particular ou classe de contratos.

Esses propósitos estão representados na Figura 1.1, na qual podemos observar que a avaliação do processo de *software* conduz a determinação da capacidade ou a melhoria do processo. Uma vez determinada a capacidade do processo, esta pode motivar a tomada de ações para melhoria do processo. A determinação da capacidade identifica riscos, enquanto a melhoria do processo conduz a identificação de mudanças para o mesmo.

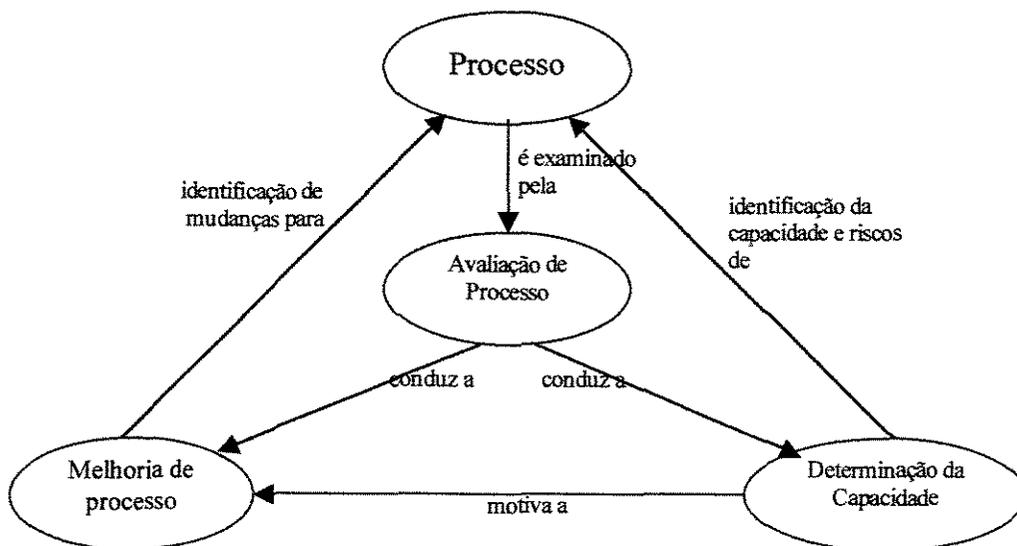


Figura 1.1: Avaliação de processo de *software* [ISO99-1].

<sup>5</sup> SPICE - Software Process Improvement and Capability dEtermination – Determinação de Capacidade e

Nos últimos trinta anos, a indústria de *software* voltou seus esforços para a melhoria da qualidade dos produtos resultantes do processo de desenvolvimento de *software* e para o aumento da produtividade. Destes esforços de melhoria, nasceram modelos formais para o desenvolvimento de *software*. O primeiro deles foi o ciclo de vida clássico (*waterfall model*) em 1970 e a partir deste, vários outros tais como o modelo de prototipação, o modelo espiral e técnicas de quarta geração [PRES95]. Com a consolidação e o amadurecimento desses paradigmas, conheceu-se o termo *software life cycle models* (modelos de ciclo de vida de *software*), mas só na década de 80 é que o termo *Software Process* (processo de *software*) foi reconhecido como uma disciplina independente.

O processo de engenharia de *software* (processo de *software*) pode ser definido como sendo o conjunto de todas as atividades relacionadas ao desenvolvimento, controle, validação e manutenção de um *software* operacional. Desta forma, as pesquisas relativas ao processo de *software* tem como objetivo consolidar o entendimento do processo de *software*, desenvolver modelos e formalismos adequados para representá-lo, bem como produzir estratégias, metodologias e ferramentas para apoiar sua programação, execução e manutenção. Assim, o processo de *software* diz respeito à uma seqüência de passos ordenados, envolvendo conjuntos relacionados de artefatos, recursos humanos e computacionais, estruturas organizacionais e restrições, com o objetivo de produzir e manter um sistema através da execução de determinadas atividades [GIME94].

A idéia de execução de atividades de processo de *software* foi proposta por Osterweil em [OSTE87]. Na última década, essas idéias se disseminaram, estabelecendo-se a necessidade do suporte não só aos objetos gerados durante o desenvolvimento de *software* através de ferramentas CASE<sup>6</sup>, mas também à definição e controle do próprio processo de *software*. Surgiram, então os ambientes de engenharia de *software* orientados a processo PSEE<sup>7</sup>. Esses ambientes constituem uma visão automatizada e integrada dos ambientes de engenharia de *software* [GIME94].

Um PSEE consiste de um conjunto de mecanismos para apoiar a definição e execução de processos de *software*. Esses ambientes envolvem o conceito de programação de processos e caracterizam-se por suportar a descrição e execução de processos de modo a auxiliar e controlar todas as atividades envolvidas na produção e manutenção de um produto de *software* [GIME94].

A maneira pela qual um engenheiro de *software* utiliza uma linguagem rigorosamente definida como um veículo para descrever complexos e intrincados processos de criação e gerenciamento de informação é o que chama-se de programação [OSTE87]. As linguagens que suportam este tipo de programação são denominadas de linguagens de propósito geral [BAND92]. A atividade de especificar o processo de *software* apoiado por técnicas de programação é chamada de programação de processo [OSTE87]. Uma

---

Melhoria do processo de software

<sup>6</sup> CASE – Computer Aided Software Engineering – Engenharia de software Suportada por Computador.

<sup>7</sup> PSEE – Process Software Engineering Environment – Ambiente de Engenharia de Software Voltado ao Processo.

linguagem que suporte a programação do processo de *software* deve apresentar algumas características essenciais, conforme descritas em [BAND92]:

- características herdadas das linguagens de programação de propósito geral seqüenciais, tais como: abstração, modularidade, generalização e não-determinismo;
- características herdadas de linguagens para sistemas concorrentes, reativos e tempo real, tais como: paralelismo, especificações de restrições de tempo e descrição de interação com o ambiente;
- características herdadas de linguagens de banco de dados, tais como: prover um modelo de dados conceitual, suporte eficiente a objetos persistentes de diferentes granularidades, suporte a transações muito longas e suporte a versões;
- características específicas do processo de *software*, tais como:
  - permitir coexistência de partes formalmente descritas e partes informais, incompletas e ambíguas na mesma descrição do processo;
  - suportar a ligação da execução de diferentes partes de um processo com dispositivos computacionais, seres humanos ou ferramentas pré-definidas;
  - suportar controlando a modificação dos processos durante a execução;
  - representar atividades técnicas e não técnicas;
  - oferecer ferramentas de análise, incluindo facilidades para medir processos e produtos, ferramentas para suporte a cronograma e planejamento;
  - representar o que é para ser feito quando uma certa atividade falha.

Existem diversos PSEEs desenvolvidos como resultados de pesquisa, dentre os quais podemos citar: EPOS, SOCCA, MERLIN, OIKOS, ALF, ADELE-TEMPO, SPADE, PEACE E<sup>3</sup> e PADM [CHRI95,FINK94]. É neste contexto que se insere o ExPSEE<sup>8</sup> [GIME99], o ambiente no qual é desenvolvido este trabalho.

As duas áreas descritas acima, a automação de processo de *software* (Ex: PSEE) e normalização da avaliação do processo de *software* (Ex: ISO/IEC TR 15504), tem propósitos complementares na obtenção da qualidade de *software*. A motivação para este trabalho advém do fato de que os padrões e modelos desenvolvidos para estas duas áreas são considerados recentes dentro da engenharia de *software*, portanto constituir-se tanto uma necessidade quanto um desafio para a indústria e para a comunidade científica.

O objetivo deste trabalho é elaborar uma metodologia de avaliação baseada no ISO/IEC TR 15504 e verificar como a seqüência de suas tarefas evolutivas pode ser automatizada com o apoio de um PSEE. Utilizamos como estudo de caso o ambiente ExPSEE, um ambiente experimental desenvolvido na Universidade Estadual de Maringá – UEM, sob a coordenação da Prfa. Dra. Itana Maria de Souza Gimenes, a qual co-orientou este trabalho.

No Capítulo 2 são apresentados os conceitos que originaram a base da qualidade de *software*. No Capítulo 3 são abordados os modelos de qualidade, dentre outros o modelo produzido pelo projeto SPICE, o ISO/IEC TR 15504, tratado como *framework* de avaliação

---

<sup>8</sup> ExPSEE - Experimental PSEE.

de processo de *software*. No Capítulo 4 são discutidos os conceitos relacionados ao processo de *software* e os PSEEs. Nesse Capítulo é descrito com mais detalhes o ExpPSEE. No Capítulo 5 é apresentada a metodologia proposta para avaliação do processo de *software*. No Capítulo 6, é apresentado um estudo de caso no qual esta metodologia é aplicada e seus resultados são discutidos. No Capítulo 7, são apresentados conclusões e os trabalhos futuros.

## Capítulo 2

### Qualidade de *software*

Como descrito no Capítulo 1, este trabalho envolve duas áreas: a automação de processo de *software* e a normalização da avaliação do processo de *software*. Estas duas áreas possuem um único propósito, o de garantir a qualidade do produto obtido pelo processo de *software*. Portanto, é importante neste capítulo apresentar os conceitos relacionados a qualidade de *software* no contexto atual. Para tanto, são apresentados os trabalhos de McCall, Boehm, Gilb, os quais são voltados a medições da qualidade e os trabalhos de Crosby, Deming e Juran, os quais são voltados ao gerenciamento da qualidade. Estes trabalhos foram descritos por [GILL97].

Esses conceitos formam a base para os modelos de qualidade atualmente utilizados os quais são apresentados nos capítulos 3 e 4.

#### 2.1 – Conceito de qualidade

Existem várias definições sobre o termo qualidade na literatura, uma discussão sobre elas pode ser encontrada em [GILL97]. Segundo a definição da ISO, a qualidade dos produtos e serviços está relacionada ao atendimento de necessidades explicitamente declaradas pelo cliente, ou implicitamente subtendidas para o produto que está sendo adquirido [ISO86]. Esta definição pode ser resumida como a resposta a duas perguntas: O produto é uma boa solução ?, A solução oferecida pelo produto visa ao problema correto ? A idéia de conformidade com a especificação e aptidão para o propósito é encontrada na maioria das definições [GILL97].

Nesta seção são apresentadas as idéias centrais dos trabalhos de McCall, Boehm e Gilb, os quais são voltados à medição da qualidade.

##### 2.1.1 - O trabalho de McCall

Em 1977, McCall propôs um modelo baseado em três áreas do processo de *software*: operação do produto, revisão do produto e transição do produto. A operação do produto requer que ele seja entendido e operado facilmente. A revisão do produto preocupa-se com a correção de erros e suas adaptações. A transição do produto para outras máquinas, outros ambientes operacionais ou a interação com outros sistemas pode não ser aplicada em todos os casos, mas com as constantes mudanças de hardware e *software*, e processamento distribuído, cada vez mais vem ganhando importância. O trabalho McCall foi descrito por [GILL97]. A Figura 2.1 representa estas três áreas com seus critérios de qualidade.

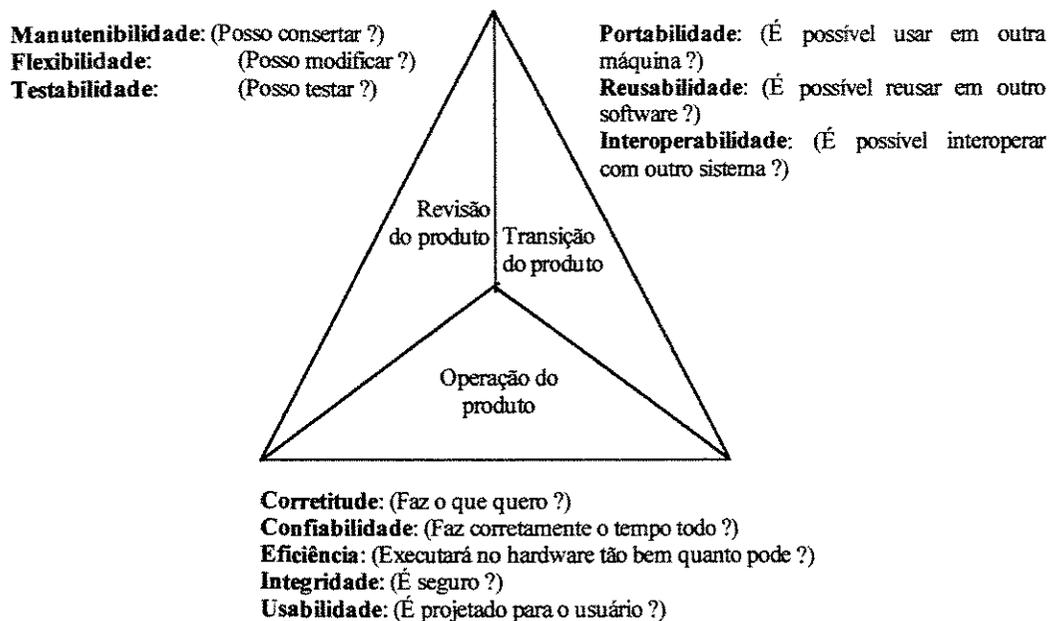


Figura 2.1: Fatores de Qualidade [GILL97].

Este trabalho apresentado por McCall formou a base da norma ISO 9126 [ABNT94a]. Dele e dos trabalhos de Gilb foram retiradas as características e subcaracterísticas para avaliação de produtos de *software*.

### 2.1.2 - O trabalho de Boehm

O modelo proposto por Boehm (1978) provê um conjunto de características de qualidade de *software* organizadas de maneira hierárquica, descrito por [GILL97]. As características são organizadas de maneira que os critérios de qualidade são divididos em dois níveis intermediários de acordo com o seu uso dentro do processo de *software*. Após estes dois níveis, os critérios chegam a características primitivas as quais são facilmente mensuráveis. A Figura 2.2 mostra o modelo hierárquico de Boehm.

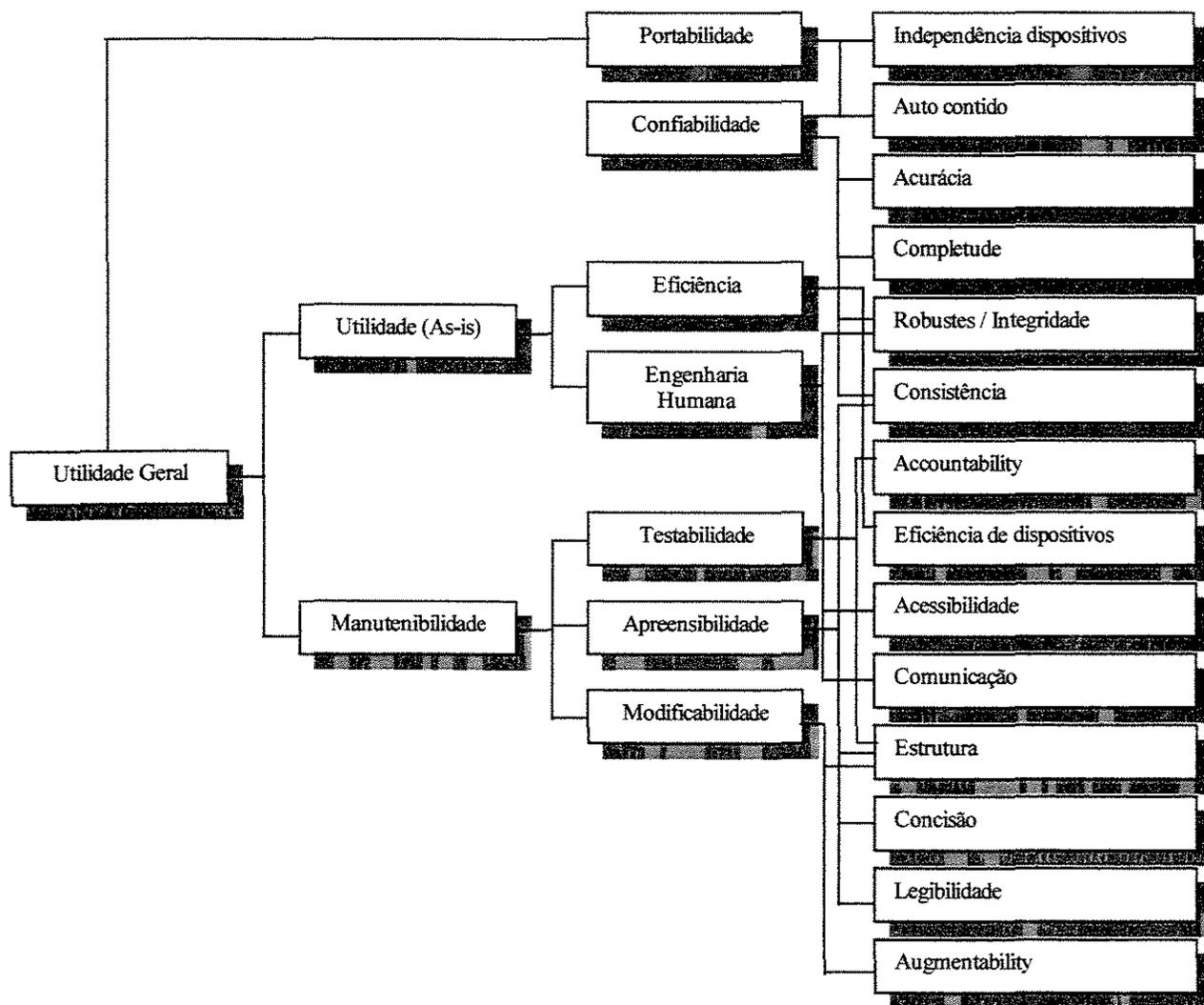


Figura 2.2: Modelo hierárquico de Boehm [GILL97].

A estrutura hierárquica do trabalho de Boehm formou a estrutura da norma ISO 9126 [ABNT94a]. As características de qualidade dos trabalhos de Gilb e McCall, foram arranjadas de maneira hierárquica em dois níveis e são utilizadas desta forma atualmente.

### 2.1.3 - O trabalho de Gilb

Uma das principais contribuições foram os trabalhos desenvolvidos por Tom Gilb, descrito por [GILL97], contemporaneamente a Boehm e MacCall. Gilb adotou um modelo baseado em duas linhas: uma evolucionária, na qual o processo de *software* se auto desenvolve e outra baseada em *templates*.

A idéia de um processo evolucionário é baseada em uma abordagem iterativa dos estágios de desenvolvimento de *software*. Em cada estágio, o desenvolvedor tentará maximizar a distância percorrida em direção ao produto de *software*, minimizando os recursos gastos. Ao invés de entender e especificar um projeto como um todo, logo no início, o desenvolvedor poderá fazê-lo por partes, com maiores chances de sucesso, descrito por [GILL97]. Esta visão é complementar ao uso de *templates*, que, ao invés de um modelo hierárquico rígido, pode atender aos requisitos específicos de cada produto de *software*, os quais podem variar de um produto para outro. Os *templates* são representados por 4 atributos e 15 sub-atributos. Estes atributos de qualidade sofrem restrições de 4 outros atributos relacionados aos recursos: pessoas, tempo, custo e ferramentas. A Tabela 2.1 mostra os atributos, sub-atributos e um exemplo de medição.

Estes trabalhos formaram a base para uma série de pesquisadores como os trabalhos do *National Computer Centre* (NCC) da Alemanha, o modelo MQ publicado por Watts em 1987 no Reino Unido, os trabalhos de Kitchenham através do projeto COCOMO em 1984, resultando no conjunto de ferramentas do modelo COQUAMO em 1989, descritos por [GILL97]. O modelo mais difundido, originado destes trabalhos, foi publicado pela primeira vez em 1986 e é utilizado até os dias de hoje: a ISO 9126, a qual está baseada nas características e os atributos de qualidade para padronizar a avaliação de produtos de *software* [ABNT94a].

Tabela 2.1: Atributos e sub atributos da qualidade propostos por Gilb [GILL97].

Atributo	Sub-atributo	Medida geral	Exemplo específico
Facilidade de trabalho	Capacidade do processo	Unidades por tempo	Transações por segundo
	Tempo de resposta	Ações por tempo	Tempo de resposta em segundos
	Capacidade de armazenamento	Unidades armazenadas	Bytes por registro
Disponibilidade	Geral	Probabilidade de disponibilidade	Tempo disponível / tempo total
	Confiabilidade	Tempo médio entre falhas	Tempo total / número de falhas.
	Manutenibilidade	Tempo médio para consertar	Tempo médio para consertar 90% dos erros do teste
	Integridade	Totalidade	Grau do produto de <i>software</i> intacto
Adaptabilidade	Melhorabilidade	Menor tempo de mudança	Tempo para adicionar um conjunto de testes
	Extensibilidade	Tempo para adicionar uma função	Tempo para adicionar 10% de lógica
	Portabilidade	Esforço para transferir	Porcentagem de esforço para portar

Usabilidade	Geral	Grau de produtividade	Tempo para obter níveis básicos de habilidade
	Nível de entrada	Nível de qualificação	Confiabilidade
	Requisitos de aprendizado	Tempo para aprender	Quantidade do treinamento necessário em dias
	Operação	Produtividade	Tarefas por hora
	Agradabilidade	Extensão de atitude positiva	Porcentagem de avaliação

Gilb utiliza o termo atributo como uma característica local, o que é aplicável àquele processo de *software*, não necessariamente transferível para outro contexto. O termo sub-atributo é equivalente a um critério mensurável, também levando em consideração o aspecto abrangência.

## 2.2 – Conceito de gerenciamento da qualidade

Os conceitos citados no item 2.1 deste capítulo, apresentados por McCall, Boehm, Gilb e os modelos baseados nestes, têm foco na medição de características e critérios do processo e do produto de *software* propriamente ditos. Outros pesquisadores preocuparam-se com o gerenciamento da qualidade. Os três principais foram Deming, Juran e Crosby, os quais foram descritos por [GILL97] e são apresentados a seguir.

### 2.2.1 - Os trabalhos de Deming, Juran e Crosby

O principal responsável pela criação do termo *quality management* foi Edward Demming, descrito por [GILL97]. Após a segunda guerra mundial, a convite da JUSE<sup>9</sup>, Deming colocou em prática suas idéias, as quais formaram a base da recuperação da economia japonesa no pós-guerra. Com a grande prova de que as idéias de Deming, compartilhadas e apoiadas por Juran, estavam certas, estas foram rapidamente adotadas nos Estados Unidos juntamente com as idéias de Phil Crosby e sendo em seguida disseminadas para vários outros países.

Mesmo formando a base dos princípios de gerenciamento da qualidade utilizada hoje, os três autores tinham posturas diferentes e algumas divergências de opiniões e abordagens em relação à qualidade. Deming acreditava em uma única fonte de

<sup>9</sup> JUSE - Union of Japanese Scientists and Engineers - União Japonesa de Cientistas e Engenheiros.

fornecimento de insumos para o processo, argumentando que os benefícios de um forte relacionamento de cooperação com fornecedores poderia ser maior do que os ganhos com compras de baixo custo. Este relacionamento é avaliado pelo SPC<sup>10</sup> [GILL97]. Juran é conhecido pela sua frase aptidão para o propósito que influenciou fortemente as idéias de qualidade de *software*. Juran argumenta fortemente que as definições de qualidade baseadas em conformidade com as especificações são inadequadas. Esta abordagem não é contrária a abordagem de Deming, mas sim complementar. Isto não é sempre verdade quando comparadas com as idéias de Crosby. Por exemplo, Deming e Juran argumentam contra campanhas exaltando a equipe à perfeição, eles preferem aplicar técnicas de SPC. Já as principais abordagens de Crosby relativas a qualidade são zero defeitos e conformidade com a especificação. A Tabela 2.2 compara as principais idéias destes autores.

Tabela 2.2: Comparação das principais idéias de Crosby, Deming e Juran [GILL97].

	<b>Crosby</b>	<b>Deming</b>	<b>Juran</b>
Definição	Conformidade com os requisitos	Grau de uniformidade e dependência previsíveis a um baixo custo.	Aptidão para o propósito.
Responsabilidade do gerenciamento sênior	Responsáveis pela qualidade	Responsáveis por 94% dos problemas	Responsáveis por mais de 80% dos problemas
Padrão de desempenho	Defeitos zero	Muitas escalas: uso de SPC	Evitar campanhas exaltando a perfeição
Abordagem geral	Prevenção	Reduzir a variabilidade: melhoria contínua	Ênfase no gerenciamento dos aspectos humanos
Estrutura	14 passos	14 pontos	10 passos
SPC	Rejeita níveis de qualidade estatisticamente aceitáveis	SPC deve ser usado	Recomenda SPC, mas cautela contra abordagens baseadas em ferramentas
Base para melhora	Um processo, não um programa	Contínua: eliminação de objetivos	Abordagem baseada em projetos: conjunto de objetivos
Equipe de trabalho	Equipes para melhoria da	Empregados participam nas	Abordagem cíclica para a equipe e

<sup>10</sup> SPC - Statistical Process Control - Controle Estatístico do Processo.

	qualidade: conselho da qualidade	decisões	qualidade
Custo da qualidade	Qualidade é de graça	Não existe o ótimo: melhoria contínua	Existe o ótimo, qualidade não é de graça
Compras	Fornecedores são uma extensão do negócio	Uso de SPC através de forte cooperação	Problemas complexos, uso de formas de pesquisa
Comissão do vendedor	Sim	Não	Sim, mas trabalhar com fornecedores
Uma única fonte de fornecimento		Sim, número pequeno	Não

## 2.2.2 - Os principais termos do gerenciamento da qualidade

Com a popularização e o uso freqüente de termos relativos à qualidade, alguns deles tornaram-se bastante populares e conhecidos apenas pelas suas abreviações. Para melhor compreensão dos modelos que se seguem, é necessário definir os principais termos que são: QMS, TQM e QIP.

### 2.2.2.1 – QMS: *Quality Management System*

A ISO define o sistema de gerenciamento da qualidade como a estrutura organizacional, responsabilidades, procedimentos, processos e recursos para implementação do gerenciamento da qualidade [ISO86]. O objetivo principal do QMS é prover uma estrutura para assegurar que os processos organizacionais sejam executados de uma maneira formal e sistemática. Uma parte essencial de qualquer QMS é o processo de melhoria contínua a partir da documentação de erros encontrados no processo e o mecanismo de *feedback loop* popularizado por Deming como o PDCA<sup>11</sup> conforme mostra a Figura 2.3.

---

<sup>11</sup> PDCA - Plan-Do-Check-Act – Planejar-Executar-Verificar-Agir

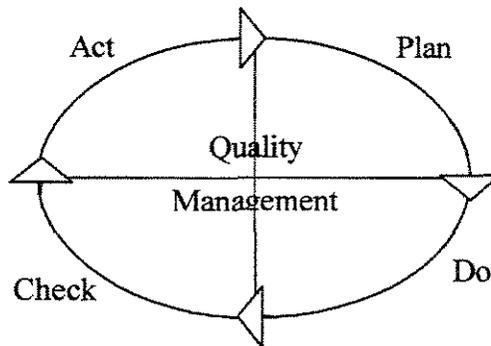


Figura 2.3: Ciclo PDCA [GILL97].

#### 2.2.2.2 – TQM: *Total Quality Management*

Gerenciamento da qualidade total é um método para resgatar a vida das pessoas do esforço desperdiçado, envolvendo cada um no esforço de melhoria efetiva do trabalho, de maneira que resultados sejam obtidos em menos tempo [GILL97]. TQM às vezes é mal interpretado em virtude da associação com a idéia de Crosby: qualidade é defeito zero. O princípio básico do TQM é o envolvimento de todas as pessoas e todos os processos dentro de um esforço de gerenciamento da qualidade, o que não implica em promessa de perfeição.

#### 2.2.2.3 – QIP – *Quality Improvement Program*

Este termo, introduzido por Crosby, refere-se aos programas projetados para melhorar a qualidade os quais são baseados na introdução ou no refinamento de um QMS. O ponto forte deste termo é sua ênfase de que melhorar é preferível a monitorar o estado dos negócios. A desvantagem é que a qualidade pode ser vista como um programa de curta duração, ao invés de um processo contínuo.

### 2.3 - Considerações finais

Neste capítulo foi apresentada uma definição para qualidade e os principais conceitos relacionados a qualidade de *software*, os quais foram tratados sob duas abordagens: medição e gerenciamento. A abordagem de medição apresenta os modelos de McCall, Boehm e Gilb e está ligada atualmente a qualidade de produto de *software*. A abordagem de gerenciamento mostra os trabalhos de Deming, Juran e Crosby, e os termos freqüentemente utilizados no dia a dia da qualidade: TQM, QMS e QIP, com suas respectivas definições e está ligada atualmente a qualidade do processo de *software*.

A principal contribuição deste capítulo para a metodologia proposta é formar uma base de conceitos relativos a qualidade, os quais serão utilizados nos capítulos seguintes, principalmente nos Capítulos 3 e 4, nos quais são apresentados, discutidos e comparados os padrões e normas de qualidade de *software* mais utilizados atualmente, elaborados a partir destes conceitos.

## Capítulo 3

### Modelos de Qualidade de *software*

Este trabalho abrange duas áreas: a automação de processo de *software* e normalização da avaliação do processo de *software*, ambos objetivando garantir a qualidade do produto obtido em um processo de *software*. Portanto, neste capítulo serão apresentados os principais modelos de qualidade de *software* e uma comparação entre eles, com um enfoque voltado ao processo de *software*.

É importante mostrar que esses modelos são baseados nos conceitos apresentados no Capítulo 2, por meio dos trabalhos de McCall, Boehm, Gilb, Crosby, Demin, Juran e outros que nestes desenvolveram outras experiências. O entendimento das origens dos modelos, é fundamental para seu entendimento e comparação. A partir dessa comparação é mostrado o porque da escolha em se trabalhar com o modelo proposto pelo projeto SPICE, a norma para avaliação de processo de *software* ISO/IEC TR 15504.

#### 3.1 – A ISO 9000-3

A ISO 9000-3 é um guia que define diretrizes para facilitar a aplicação da norma ISO 9001 nas organizações que desenvolvem, fornecem e mantêm *software* [ABNT96]. Além de orientar contratos entre duas partes possibilitando uma análise da capacidade de desenvolvimento e manutenção de produtos de *software*, esta norma destina-se a descrever métodos sugeridos para a produção de *software* que atendam aos requisitos do comprador.

Um sistema de qualidade em conformidade com os requisitos da ISO 9000-3 deve seguir as diretrizes constantes na política de qualidade ditadas pela alta administração. Os responsáveis pela gestão da qualidade na empresa devem, ao estruturar o sistema de qualidade, definir responsabilidades e delegar autoridade (tanto do cliente quanto do fornecedor), definir a capacitação para o exercício de cada função, bem como viabilizar recursos e treinamento para a execução do sistema de qualidade [ABNT96].

O sistema de qualidade deve ser um processo integrado ao longo de todo o ciclo de desenvolvimento de *software*, assegurando assim que a qualidade dos produtos desenvolvidos ao longo do processo e do produto final, atinja aos requisitos determinados. A prevenção de problemas deve ser mais enfatizada do que a correção, por isso, além de exigir a documentação de todo o processo de *software* e assegurar o cumprimento do plano de qualidade elaborado pelo fornecedor, um sistema de auditorias internas deve ser efetivamente implementado.

A ISO 9000-3 está dividida em 3 grandes partes, conforme mostra a Figura 3.1. Estas partes definem os procedimentos para processos organizacionais, de produção e de apoio [ABNT96]. As atividades organizacionais são:

- 4.1 – Responsabilidade da administração
- 4.2 – Sistema da qualidade
- 4.3 – Auditorias internas do sistema da qualidade
- 4.4 – Ações corretivas

Para as atividades de produção, a ISO 9000-3 não define ou beneficia a aplicação de qualquer modelo de ciclo de vida, mas exige que um modelo seja definido e implementado. São definidas nove atividades relacionadas a produção de *software*:

- 5.2 – Análise crítica de contrato.
- 5.3 – Especificação dos requisitos do comprador.
- 5.4 – Planejamento do desenvolvimento.
- 5.5 – Planejamento da qualidade.
- 5.6 – Projeto e implementação.
- 5.7 – Ensaio e validação.
- 5.8 – Aceitação.
- 5.9 – Cópia, entrega e instalação.
- 5.10 – Manutenção.

As atividades de apoio são independentes de fase, ou seja, aplicam-se a qualquer fase do ciclo de vida definido. São elas:

- 6.1 – Gerenciamento de configuração.
- 6.2 – Controle de documentos.
- 6.3 – Registros da qualidade.
- 6.4 – Medição.
- 6.5 - Regras, práticas e convenções.
- 6.6 – Ferramentas e técnicas
- 6.7 – Aquisição.
- 6.8 – Produto de *software* incluído.
- 6.9 – Treinamento.

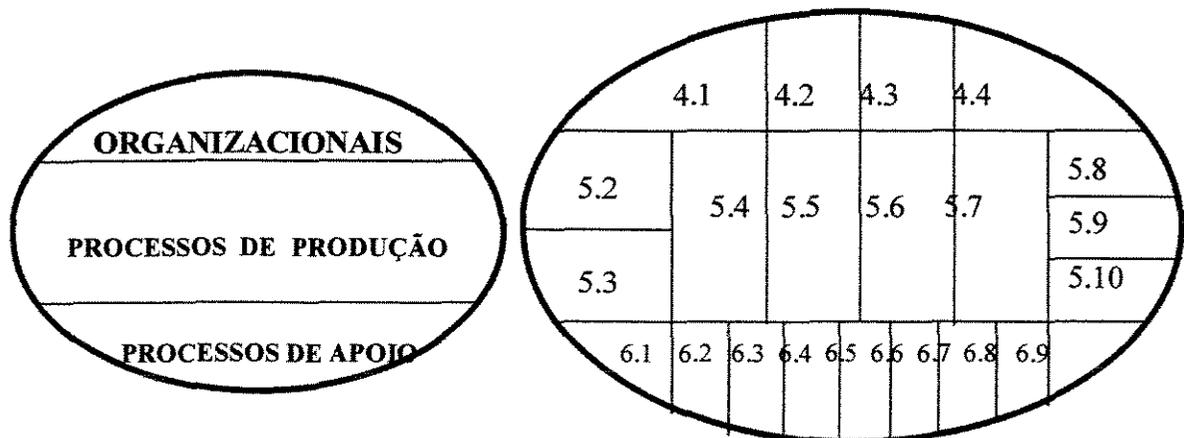


Figura 3.1: estrutura da ISO 9000-3 [CABR98].

No Brasil existem 268 organizações do setor de informática com algum tipo de certificado oficial segundo a ABNT/CB25<sup>12</sup>, conforme mostra a Tabela 3.1. Utilizando algum padrão da série ISO 9000, 74 organizações o que é um número bastante pequeno se comparado a padrões europeus. Deve ser observado que, mesmo que as organizações de *software* utilizem a ISO 9000-3 como padrão para adequação de seus processos, a certificação obtida será ISO 9001 ou ISO 9002 de acordo com o escopo da certificação.

Tabela 3.1: Certificação do sistema da qualidade [MCT00].

<b>Empresas Certificadas</b>	<b>1999</b>
Informática	268
Pesquisa da Qualidade em Software	74
Certificação ISO 9001	63
Certificação ISO 9002	16
Certificado CMM	5
Software explicitado no escopo do certificado	39

Nota: A fonte para empresas certificadas de todos os setores foi a ABNT/CB-25.

Porém a tendência deste quadro é que ele melhore graças à iniciativas do PBQP conforme mostra a Tabela 3.2. As ações do PBQP são para que o número de empresas de informática certificadas cresça. Apesar da meta de 120 empresas em 1999 não ter sido atingida, houve um crescimento de 60%, que é algo muito significativo.

Tabela 3.2: Indicadores e Metas da Qualidade e Produtividade em Software [MCT99].

<b>Métodos de Gestão</b>	<b>Fonte</b>	<b>1995</b>	<b>1997</b>	<b>1999</b>	<b>2001</b>
Percentual de empresas com programa da Qualidade total, sistema da qualidade ou similar implantado sobre o total de empresas	SEPIN	11%	18%	30%	50%
Percentual de empresas com sistema da Qualidade certificado (ISO 9001 e ISO 9002) sobre o total de empresas	SEPIN	2%	8%	20%	35%
Número de empresas com sistema da qualidade certificado (ISO 9001 e ISO 9002)	SEPIN	8	45	120	200
Número de empresas com <i>software</i> explicitado no escopo do certificado de qualidade (ISO 9001 e ISO 9002)	SEPIN	-	16	50	100

<sup>12</sup> ABNT – Associação Brasileira de Normas Técnicas – Comissão Brasileira de qualidade.

Percentual de empresas que conhecem e usam o modelo CMM sobre o total de empresas	SEPIN	3%	5%	10%	20%
Número de empresas com modelo CMM implantado, por nível	SEPIN	-	1 (nível 2)	Acompanhamento	
Percentual de empresas que fazem, de forma sistemática, medições usando a técnica de pontos por função sobre o total de empresas	SEPIN	-	4%	Acompanhamento	
Percentual de empresas que mantém, de forma sistemática, contabilidade de custos da Qualidade sobre o total de empresas	SEPIN	4%	6%	Acompanhamento	

**Nota:** Valores observados para 1995 e 1997 e metas estabelecidas para 1999 e 2001.

### 3.2 – A norma ISO/IEC 12207

A norma ISO/IEC 12207 estabelece os processos, atividades e tarefas a serem aplicados durante a aquisição, fornecimento, desenvolvimento, operação e manutenção de *software*. Com uma definição abrangente em relação aos processos, a norma ISO/IEC 12207 orienta a adaptação para a utilização destes processos nos projetos de *software* implementados numa organização [ISO94].

A norma ISO/IEC 12207 define dezessete processos do ciclo de vida de *software* e os organiza em três classes: processos primários, processos de apoio e processos organizacionais, conforme mostra Figura a 3.2.

A importância desta norma é o estabelecimento de uma estrutura de classificação de processos e normalização da terminologia [SALV96].

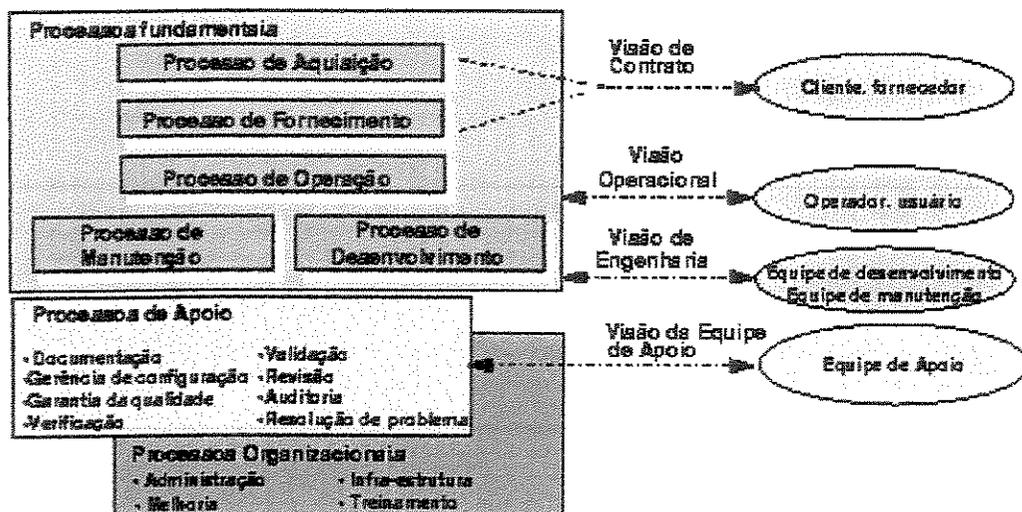


Figura 3.2: Visões da ISO/IEC 12207 [SALV96].

A norma ISO/IEC 12207 é importante também para padronizar a nomenclatura e os processos junto a outras normas. Ela é largamente utilizada no ISO/IEC TR 15504 e na ISO 9000-3, dentre outras, o que torna possível um mapeamento entre os modelos.

No Brasil, a norma ISO/IEC 12207 é pouco conhecida e utilizada pelas organizações, conforme mostra a Tabela 3.3. As normas ISO 9126 e a ISO 12119 [OLIV97], utilizadas para avaliação de produto e de pacote de *software* respectivamente, não fogem a esses padrões. Um trabalho muito importante é realizado pela fundação CTI – Campinas com o método MEDE-PROS, o qual é baseado nestas duas normas. Este método realiza a avaliação de produtos de *software* no Brasil a 8 anos e é utilizado como referência internacional [OLIV97].

Tabela 3.3: Conhecimento de normas para qualidade [MCT00].

Categorias	ISO / IEC 12207		ISO / IEC 12119		ISO / IEC 9126	
	Nº	%	Nº	%	Nº	%
Conhece e usa	16	3,6	4	0,9	15	3,4
Conhece e começa a usar	53	11,9	23	5,2	27	6,1
Conhece, mas não usa	121	27,2	114	25,6	119	26,7
Não conhece	255	57,3	304	68,3	284	63,8
<b>Base</b>	<b>445</b>	<b>100</b>	<b>445</b>	<b>100</b>	<b>445</b>	<b>100</b>

**Nota:** No Brasil, a Norma NBR 13596 corresponde à ISO/IEC 9126.

### 3.3 – O modelo CMM: *Capability Maturity Model*

O modelo CMM foi desenvolvido pelo SEI da *Carnegie Mellon University*, e tem por objetivo a avaliação da capacidade e maturidade de uma organização, indicando diretrizes para melhoria do processo de *software*. Este modelo foi desenvolvido segundo uma solicitação do DoD<sup>13</sup> americano sendo divulgado desde 1991. O modelo rapidamente tornou-se conhecido pela riqueza de detalhes e a completa descrição do processo de *software* [SALV96]. O CMM passou a influenciar outros modelos que se seguiram, inclusive o modelo proposto pelo projeto SPICE, o ISO/IEC TR 15504.

No modelo CMM as organizações de *software* são enquadradas em um dos cinco níveis de maturidade definidos. Estes cinco níveis de maturidade, estão baseados nos princípios de qualidade de processo propostos por Deming, Juran e Crosby, com contribuições de Shewart [SALV96], conforme apresentado na Seção 2.2.1 desta dissertação.

<sup>13</sup> DoD – Department of Defense – Departamento de Defesa

Os níveis de maturidade representam o perfil de uma organização, a capacidade e maturidade da mesma. Para se atingir um nível, deve-se satisfazer completamente as práticas definidas para cada nível. Estas práticas são mostradas na Tabela 3.4.

Tabela 3.4: níveis de maturidade e áreas chave de processos do CMM [SALV96].

<b>Nível de maturidade</b>	<b>Áreas chave de processos</b>
<b>Nível 1 – Inicial</b>	
O processo de <i>software</i> é caracterizado como <i>ad hoc</i> , ocasionalmente até caótico, poucos processos definidos, o sucesso depende dos esforços individuais e heroísmo.	Não existem áreas chave para este nível.
<b>Nível 2 – Repetitivo</b>	
Processos básicos de gerenciamento de projeto são estabelecidos para controlar custos, cronogramas e funcionalidades. A disciplina de processo permite repetir sucessos anteriores em projetos de aplicação similar.	<ul style="list-style-type: none"> <li>– Gerenciamento de requisitos</li> <li>– Planejamento de projeto de <i>software</i></li> <li>– Acompanhamento de projeto de <i>software</i></li> <li>– Gerenciamento de sub-contratos</li> <li>– Qualidade assegurada de <i>software</i></li> <li>– Gerenciamento de configuração</li> </ul>
<b>Nível 3 – Definido</b>	
O processo de <i>software</i> em relação tanto às atividades de gerenciamento como às atividades de engenharia é documentado, padronizado e integrado aos processos padrões da organização. Todos os projetos usam uma versão aprovada e adaptada do processo padrão de <i>software</i> da organização para o desenvolvimento e manutenção.	<ul style="list-style-type: none"> <li>– Foco no processo da organização</li> <li>– Definição do processo da organização</li> <li>– Programa de treinamento</li> <li>– Engenharia de produto de <i>software</i></li> <li>– Gerenciamento integrado do <i>software</i></li> <li>– Coordenação entre grupos</li> <li>– Revisões</li> </ul>
<b>Nível 4 – Gerenciado</b>	
São efetuadas medições detalhadas do processo de <i>software</i> e qualidade do produto. Tanto o processo como o produto são entendidos e controlados quantitativamente.	<ul style="list-style-type: none"> <li>– Gerenciamento Quantitativo do processo</li> <li>– Gerenciamento da qualidade de <i>software</i></li> </ul>
<b>Nível 5 – Em Otimização</b>	
Melhoria contínua do processo é possibilitada pela realimentação quantitativa do processo e conduzida a partir de idéias e tecnologias inovadoras.	<ul style="list-style-type: none"> <li>– Prevenção de defeitos</li> <li>– Gerenciamento de mudanças tecnológicas</li> <li>– Gerenciamento de mudanças de processo</li> </ul>

Quanto a utilização deste modelo no Brasil, nota-se que o CMM é um modelo cujo uso vem crescendo nos últimos anos, conforme mostra a Tabela 3.5, obtida pela pesquisa bianual do PBQP<sup>14</sup>. No Brasil algumas instituições já estão divulgando e implantando este modelo em outras organizações, pode-se citar como exemplo o CITS<sup>15</sup> em Curitiba e a Fundação VANZOLIN em São Paulo.

Tabela 3.5: Conhecimento dos modelos CMM e SPICE [MCT00].

Categorias	CMM		ISO/IEC TR 15504	
	Nº	%	Nº	%
Conhece e usa	8	1,8	1	0,2
Conhece e começa a usar	36	8,1	14	3,2
Conhece, mas não usa	165	37,2	121	27,3
Não conhece	234	52,8	308	69,4
<b>Base</b>	<b>443</b>	<b>100</b>	<b>444</b>	<b>100</b>

### 3.4 – O projeto SPICE

O Comitê de Engenharia de Software da ISO aprovou em junho de 1991, a realização de estudos para analisar as necessidades de um padrão para avaliação do processo de *software*. Este estudo apontou um consenso internacional sobre a necessidade deste padrão e a importância de se implementar melhorias no processo de *software* [ISO99-1].

Para atender este consenso, foi criado em 1993 o projeto SPICE, um acordo internacional que determinou a criação de quatro centros de coordenação mundial: Austrália, EUA, Inglaterra e Canadá, com pesquisadores de vários países, dentre eles o Brasil. O objetivo principal do projeto SPICE é produzir normas que orientem a avaliação de processo de *software* visando melhoria contínua e determinação da capacidade de processos de *software*.

O projeto SPICE propôs um documento que está tramitando na ISO para ser publicado como a norma ISO 15504. Atualmente, esta norma é denominada ISO/IEC TR 15504<sup>16</sup>, por estar em fase de elaboração pelo grupo de trabalho ISO/IEC SC7/WG10. Esta norma prevê uma estrutura para a avaliação do processo de *software*. Esta estrutura pode

<sup>14</sup> PBQP – Programa Brasileiro de Qualidade e Produtividade.

<sup>15</sup> CITS – Centro Internacional de Tecnologias de Software

<sup>16</sup> ISO/IEC TR 15504 – esta denominação refere-se ao estágio de TR - Technical Report (Relatório Técnico) e é um dos estágios finais para que um documento seja publicado como norma. Este documento é público e já está a vesnda.

ser usada por organizações envolvidas em planejamento, gerenciamento, monitoramento, controle e melhoria da aquisição, fornecimento, desenvolvimento, operação, evolução e suporte de produtos de *software*. O documento que descreve a norma está estruturado em 9 partes como segue [ISO99]:

- Parte 1: conceitos e guia introdutório.
- Parte 2: um modelo de referência para processo e capacitação de processo.
- Parte 3: execução de avaliação.
- Parte 4: guia para execução de avaliação.
- Parte 5: um modelo de avaliação e guia de indicadores.
- Parte 6: guia para qualificação de avaliadores.
- Parte 7: guia para uso na melhoria do processo.
- Parte 8: guia para uso na determinação da capacitação de processos dos fornecedores.
- Parte 9: vocabulário.

A avaliação dos processos de *software* pelo ISO/IEC TR 15504 examina os processos utilizados por uma organização para determinar se são eficazes para atingir seus objetivos. A avaliação caracteriza a prática corrente dentro de uma unidade organizacional em termos da capacitação dos processos de *software* selecionados. Os resultados podem ser utilizados para direcionar as atividades de melhoria de processo ou para determinar a capacitação de processo. Através da análise dos resultados no contexto das necessidades dos negócios da organização, identificam-se pontos fortes, pontos fracos e riscos inerentes aos processos [ISO99-1]. Para os fins deste trabalho, utilizamos a abordagem de determinação da capacidade do processo descrita no [ISO99-8].

A seleção dos processos que serão analisados em um organização deve ser feita dentro do conjunto de processos definidos pelo Modelo de Referência [ISO99-2], conforme apresentado na Tabela 3.6. O Modelo de Referência descreve os processos que uma organização pode executar para adquirir, desenvolver, operar, manter e apoiar o desenvolvimento de *software* e os atributos de processos que caracterizam a capacidade destes processos na execução de uma avaliação de processo.

A definição do Modelo de Referência está quase 100% em acordo com a norma ISO/IEC 12207, a qual padroniza os processos de ciclo de vida de *software*, conforme descrito na Seção 3.2 deste capítulo.

Tabela 3.6: Modelo de Referência de Processos do SPICE [ISO99-2].

<b>Categoria de Processo</b>	<b>Sigla</b>	<b>Nome do Processo</b>
Cliente – Fornecedor	CUS.1	adquirir <i>software</i>
	CUS.2	gerenciar necessidades do cliente
	CUS.3	fornecer <i>software</i>
	CUS.4	operar <i>software</i>
	CUS.5	prover serviço ao cliente
Engenharia	ENG.1	desenvolver requisitos e projeto de sistema
	ENG.2	desenvolver requisitos de <i>software</i>
	ENG.3	desenvolver projeto de <i>software</i>
	ENG.4	implementar projeto de <i>software</i>
	ENG.5	Integrar e testar <i>software</i>
	ENG.6	Integrar e testar sistema
	ENG.7	manter sistema e <i>software</i>
Suporte	SUP.1	desenvolver documentação
	SUP.2	executar gerência de configuração
	SUP.3	executar garantia da qualidade
	SUP.4	executar verificação dos produtos de trabalho
	SUP.5	executar validação dos produtos de trabalho
	SUP.6	executar revisões conjuntas
	SUP.7	executar auditorias
	SUP.8	executar resolução de problemas
Gerência	MAN.1	gerenciar projeto
	MAN.2	gerenciar qualidade
	MAN.3	gerenciar riscos
	MAN.4	gerenciar subcontratados
Organização	ORG.1	construir o negócio
	ORG.2	definir processo
	ORG.3	melhorar processo
	ORG.4	prover recursos humanos treinados
	ORG.5	prover infra-estrutura de engenharia de <i>software</i>

No Modelo de Referência a medida de capacidade está baseada em um conjunto de nove atributos de processo (A 1.1 à A 5.2), conforme mostra a Tabela 3.7. Cada atributo mede um aspecto particular da capacidade do processo de *software*. Os atributos por sua vez são medidos em uma escala de 4 pontos: *F - Full*, *L - Large*, *P - Partial* e *N - No*. A composição destes atributos produz o mapa geral dos processos da organização.

Tabela 3.7: Níveis de Capacitação e Atributos de processos do SPICE. [ISO99-2].

Nível	Nome	Descrição	Atributos de processo
Nível 0	Processo Incompleto	O processo não está implementado e falha na tentativa de atender seus objetivos	não há atributos neste nível.
Nível 1	Processo Executado	O processo implementado atinge o objetivo definido	A 1.1 – Atributo de gerenciamento de desempenho do processo.
Nível 2	Processo Gerenciado	O processo executado entrega produtos de trabalho de qualidade definida dentro de cronogramas e recursos definidos	A 2.1 – Atributo de gerenciamento de desempenho. A 2.2 - Atributo de gerenciamento de Produto de trabalho.
Nível 3	Processo Estabelecido	O processo gerenciado é executado usando um processo definido baseado em bons princípios de engenharia de <i>software</i>	A 3.1 - Atributo de Definição de Processo. A 3.2 - Atributo de recursos do processo.
Nível 4	Processo Previsível	O processo estabelecido é executado consistentemente dentro de limites definidos de controle para atingir seus objetivos	A 4.1 - Atributo de medição do processo. A 4.2 - Atributo de controle de processo
Nível 5	Processo Otimizado	O processo previsível otimiza o seu desempenho para atender as necessidades de negócio atuais e futuras e atinge repetibilidade em atender aos objetivos definidos de negócios	A 5.1 - Atributo de mudança de processo. A 5.2 - Atributo de melhoria contínua.

No início de 1998 o projeto SPICE realizou pela segunda vez, testes de campo (*trials*) com o objetivo de validar seu modelo. Pela avaliação dos resultados obtidos este modelo tem se mostrado bastante flexível, podendo ser direcionado para determinar o nível de capacitação dos processos selecionados pela organização e fornecer indicadores de melhoria.

### 3.5 – Comparação entre os modelos

A Tabela 3.8 apresenta uma visão comparativa das principais características dos modelos discutidos: ISO 9000-3, ISO/IEC 12207, CMM e ISO/IEC TR 15504. Este estudo foi desenvolvido pelo PQPS<sup>17</sup> do CTI – Campinas [SALV96], o qual é o representante oficial do projeto SPICE no Brasil.

<sup>17</sup> PQPS – Programa de Qualidade e produtividade em Software.

Tabela 3.8: Comparativo entre os modelos [SALV96].

Característica	ISO 9000-3	ISO/IEC 12207	CMM	ISO/IEC TR 15504
Objetivo	Certificar a organização de acordo com padrões estabelecidos em situações de contrato de fornecimento de <i>software</i> .	Estabelecer uma terminologia e um entendimento comum para os processos entre todos os envolvidos com <i>software</i> .	Determinar a capacitação da organização e apoiar a sua evolução de acordo com os níveis estabelecidos.	Conhecer e avaliar os processos da organização para determinar a capacitação e promover a melhoria.
Abordagem	Verificação de conformidade de processos e padrões documentados.	Definição dos processos para aquisição, fornecimento, desenvolvimento, operação e manutenção de <i>software</i> .	Avaliação dos processos e enquadramento da organização em um dos níveis de maturidade.	Avaliação dos processos da organização em relação aos níveis de capacitação.
Organizações Alvo	Organizações que necessitam de uma certificação.	Organizações em geral.	Organizações que necessitam de comprovação formal de sua capacitação.	Organizações em geral.
Definição de processos	Não estabelece processos, estabelece atividades a serem cumpridas, com visão de estrutura, ciclo de vida e suporte.	Estabelece 17 processos, organizados em 3 categorias.	Estabelece 18 áreas chaves de processos organizados em 5 níveis crescentes de maturidade.	Estabelece 35 processos organizados em 5 categorias.
Flexibilidade nos aspectos definidos pelo modelo	Não admite adaptação nos aspectos abordados.	Classificação de processos pode ser utilizada conforme os objetivos da organização.	Níveis e áreas chave de processo são a base do modelo e não podem ser alterados.	Permite a definição de perfis de processo e práticas de acordo com os objetivos da organização.
Instrumento de avaliação	Lista de verificação.	Não se aplica.	Questionário.	Fornecer orientações para montar questionários.
Inspiração e influência	Normas militares americanas, canadenses, sistemas de qualidade do Reino Unido.	TQM, PDCA	Princípios de Shewart, Deming, Juran, Crosby.	TQM, PDCA, CMM, STD, Trillium, Malcolm Baldrige, Bootstrap.
Aspectos positivos	Norma Internacional; Difusão extensa;	Norma Internacional. Definição de uma	Estabelecimento de diretrizes para melhoria	Norma Internacional em elaboração; Expansão e

	Reconhecimento do valor da certificação.	taxonomia para processos úteis para qualquer organização.	contínua; Difusão extensa nos EUA.	flexibilização dos modelos citados.
Limitações	Risco de se colocar a certificação como objetivo principal. Ausência de apoio à melhoria contínua. Falta abordagem de produto.	Apenas uma definição de taxonomia de processos.	Pouca consideração da diversidade das organizações. Dificuldade de aplicação em pequenas organizações. Falta abordagem de produto.	Dificuldade de aplicação devido à grande quantidade de informações. Falta abordagem de produto.

Nota-se que a ISO 9000-3 e o CMM possuem como alvo organizações que precisam comprovação de capacitação ou atendimento a um padrão. Por isto uma abordagem de verificação de conformidade a padrões ou de determinação de um nível de capacidade o qual possa ser comparado a outras organizações. Os resultado das avaliações são utilizados como ferramenta de marketing pelas organizações e isto tem motivado à utilização destes modelos, mais do que a melhoria do processo propriamente dita.

A necessidade de comparação entre organizações ou de comprovação de atendimento a um padrão definido torna os modelos ISO 9000-3 e CMM menos flexíveis do que a ISO/IEC 12207 e o ISO/IEC TR 15504, ou seja, em uma avaliação, estes modelos podem ser definidos de acordo com os objetivos da organização. Estes é um dos principais motivos da utilização opção em se utilizar o ISO/IEC TR 15504 neste trabalho, já que o foco é na organização e seus processos de *software*.

Além do critério flexibilidade, outros motivos levaram a opção por trabalhar com o ISO/IEC TR 15504 nesta dissertação, a saber:

- será uma norma internacional.
- seu objetivo é conhecer e avaliar os processos da organização.
- é baseado nos principais modelos de avaliação de processo de *software* existentes e compatível com eles.
- é bastante completo.

### **3.6 - Considerações finais**

Neste capítulo apresentou-se os principais modelos de qualidade para processos de *software*: as normas ISO 9000-3 e ISO/IEC 12207, o projeto SPICE e sua futura norma o ISO/IEC TR 15504, e o modelo CMM. Os modelos foram discutidos quanto a sua estrutura e apresentado o contexto atual no Brasil em relação a cada um dos modelos. Estes modelos foram comparados segundo 9 características: objetivo, abordagem, organizações alvo, definição de processos, flexibilidade nos aspectos definidos pelo modelo, instrumento de avaliação, inspiração e influência, aspectos positivos e limitações. A partir deste comparativo justificamos a opção em utilizar o ISO/IEC TR 15504 neste trabalho.

Pode-se observar que a questão da qualidade de *software* no Brasil está sendo tratada com seriedade, o número de empresas certificadas cresce a cada ano e o MCT, através do PBQP possui metas claras de crescimento. Isto é importante para que se possa afirmar que este trabalho está alinhado com as expectativas e metas governamentais e da indústria de *software* no Brasil, que deseja conhecer seus processos de produção, adequá-los às normas internacionais e estabelecer mecanismos de monitoramento e melhoria contínua.

## Capítulo 4

### O processo de *software*

No capítulo anterior foram apresentados e comparados os principais modelos de qualidade de *software*, os quais tem como objetivo avaliar o processo de *software* para certifi-cá-lo, determinar sua capacidade ou padronizá-lo. Neste capítulo serão apresentados os principais ambientes de engenharia de *software*, que tem por objetivo o apoio a definição e execução do processo de *software*. Esses ambientes servem de apoio à programas de melhoria de processos de *software*, constituindo-se em mecanismos eficazes para utilização das normas e modelos de qualidade.

Para apresentar esses ambientes, é necessário discutir os conceitos relacionados a processo de *software* e apresentar uma arquitetura genérica para eles, o que será feito na Seção 4.2.1, deste capítulo.

É importante apresentar esses ambientes de engenharia de *software*, pois um dos objetivos deste trabalho é investigar como e o quanto da seqüência de tarefas da metodologia proposta no Capítulo 5 pode ser automatizada com o apoio de um ambiente de engenharia de *software*. O ambiente utilizado para este trabalho é o ExpPSEE, o qual será detalhado neste capítulo.

#### 4.1 – Conceitos básicos

No ano de 1969 em uma conferência de grande importância que reuniu cientistas e pesquisadores de renome, foi reconhecida a necessidade de uma disciplina que estudasse o controle da produção de *software* com fundamentos práticos e teóricos, surgiu então o termo *software engineering*, conforme descrito por [GIME94]. Outro resultado importante desta conferência expressava as dificuldades em se lidar com os crescentes requisitos colocados ao *software*: redução da taxa de falhas, melhoramento das técnicas para desenvolvimento de sistemas de grande porte e cada vez mais complexos. A partir da necessidade de se produzir *software* com baixos custos e de gerenciar grandes projetos, surgiu o termo *software crisis*, conforme descrito por [GIME94]. Os esforços realizados para vencer estas dificuldades consolidaram os modelos de ciclo de vida na década de 70 [PRES95] e o termo processo de *software* na década de 80 [OSTE87].

Um processo de *software* pode ser visto sob dois aspectos: o processo em si e a sua descrição. O processo é o veículo utilizado para fazer um trabalho, enquanto a descrição do processo é a especificação de como o trabalho deve ser feito [OSTE87]. Um programa de computador é um exemplo, no qual pode-se considerar a declaração de uma classe (ou de um tipo) e a instanciação dela como exemplos de descrição de processo e o processo gerado pelo programa como o processo em si.

A organização do processo de *software* em um conjunto de atividades ordenadas de acordo com um ciclo de vida adotado e o paradigma descrição / instanciação / execução dessas atividades, constituem a idéia central do gerenciamento do processo de *software* [OSTE87].

As subseções a seguir apresentam importantes discussões sobre o processo de *software*. Essas discussões são fundamentadas por Gimenes [GIME94].

#### **4.1.1 – Modelos de ciclo de vida versus modelos de processo de *software***

Um modelo de ciclo de vida é uma estrutura contendo processos, atividades e tarefas envolvidas no desenvolvimento, operação e manutenção de um produto de *software*, abrangendo a vida do sistema desde a definição de seus requisitos até o término de seu uso [ISO94]. As noções iniciais de modelagem de processo de *software* originaram-se dos modelos de ciclo de vida de *software*. Porém estes termos, que são muitas vezes utilizados como sinônimos, têm significados distintos e uma distinção entre eles faz-se necessário no contexto desta dissertação [GIME94].

Considera-se como o conjunto de estágios canônicos do desenvolvimento de *software*: a análise de requisitos, a especificação do sistema, o projeto da arquitetura do sistema, o projeto detalhado e a implementação [PRES95]. Em geral, decisões gerenciais adicionam estruturação e procedimentos aos estágios técnicos. Finalizadas as decisões gerenciais, o projeto pode ser estruturado em fases gerenciáveis e os planos e estratégias de desenvolvimento serem elaborados. Essas fases terminam em pontos específicos no tempo, produzindo um conjunto pré-definido de itens. Esses itens devem passar por procedimentos de revisão, os quais podem liberar os itens para as próximas fases ou requerer acertos nos mesmos.

Modelos de ciclo de vida de *software* enfatizam a definição de um conjunto ordenado de fases e as estratégias para administrar o seu desenvolvimento. Em contraste, os modelos de processo de *software* concentram-se na representação dos elementos do processo de *software* e sua dinâmica. Um modelo de processo de *software* não estabelece metodologias de desenvolvimento pois trabalha num nível mais alto de abstração do que os modelos de ciclo de vida [OSTE87].

Modelos de ciclo de vida de *software* são, normalmente, utilizados para instanciar modelos de processo de *software*. Esses estão mais próximos de uma metodologia de desenvolvimento, sendo algumas vezes, referenciados dessa forma [OSTE87]. Exemplos clássicos de modelos de ciclo de vida são: cascata, espiral, prototipação, dentre outros [PRES95].

#### **4.1.2 – Requisitos para Modelagem do Processo de Software**

O primeiro passo para a melhoria do processo de *software* é sua compreensão [ISO99]. O domínio de aplicações voltadas ao processo de *software* não é tão bem

compreendido quanto o domínio de aplicações convencionais, assim é importante analisar os requisitos de modelagem e automação do processo de *software*. Os requisitos aqui apresentados estão divididos em duas partes: a primeira envolve as características inerentes ao processo de *software*, a segunda envolve características importantes para que abordagens específicas ofereçam boa cobertura e suporte ao processo de *software* [GIME94].

#### Requisitos inerentes ao processo de *software*:

Critérios de transição e relações de precedência: um processo se caracteriza por uma série de ações e um conjunto de estados de seus componentes. Assim, torna-se necessário garantir que o estado de um componente só possa ser modificado se as condições para esta transição forem satisfeitas. Também é necessário que seja possível definir relações de precedência, para se estabelecer prioridades de ações e critérios gerenciais.

Iteração e *feedback loop*: o processo de *software* não é monotônico e, usualmente, requer repetição de tarefas. Iteração em processo de *software* significa a repetição de trabalho necessária para garantir a correta evolução das atividades e o acerto de inconsistências encontradas na avaliação destas. *Feedback loop* representa a repetição de trabalho que deve ser realizado para acertar as inconsistências detectadas entre diferentes atividades ou fases, tais como inconsistência entre especificação e implementação.

Concorrência e não-determinismo: muitas atividades de desenvolvimento de *software* podem ser realizadas concorrentemente. Também um certo grau de não-determinismo é necessário para representar atividades que não demandam uma estrita ordem de execução.

Comunicação entre os elementos do processo: o processo de *software* usualmente envolve grandes grupos de projeto. A cooperação entre os membros do projeto exige um alto grau de comunicação.

#### Requisitos adicionais para a modelagem do processo de *software*:

Oferecer uma representação sintática e semântica para o processo de *software*: processos de *software* são difíceis de se representar informalmente ou através de representação puramente visual. Detalhes sobre os tipos de objetos, métodos e restrições precisam ser especificados. Dessa forma, uma linguagem de programação de processos é necessária para que se possa especificar, interpretar e analisar processos de *software*.

Permitir a construção de mecanismos para automação do processo: um modelo de processo de *software* deve permitir a construção de mecanismos automatizados para definição, validação e simulação de processos. Idealmente o modelo deve estar integrado em um ambiente de suporte, onde maior controle da execução do processo possa ser exercido.

Oferecer mecanismos de análise do processo: é importante que o modelo do processo de *software* seja baseado em um formalismo com base matemática que permita análise das propriedades do processo, tais como conectividade, tempo de duração e dependência das tarefas.

Permitir integração com gestão de projetos: existe uma necessidade de integração da gestão de projetos com a gestão de processos, pois as funções inerentes a essas áreas são fortemente relacionadas.

Suportar múltiplas visões do processo: é necessário que se possa acessar as informações do processo em níveis diferentes de granularidade e pontos de vista. O modelo deve possibilitar diferentes visões do processo, tais como: visão da atividade, visão do produto e visão da organização.

Permitir registro da história do processo: os componentes do processo de *software* evoluem ao longo da execução do processo. Dessa forma, é importante manter um registro dos objetos e atividades durante a execução do processo. As informações a serem registradas incluem tempo de duração das atividades, grupo real de desenvolvimento, métodos e ferramentas utilizadas. Estas informações permitem auditoria e determinação de responsabilidades.

## **4.2 – Ambientes de engenharia de *software***

Existem muitos métodos voltados a construção de produtos de *software* que atendem aos princípios da engenharia de *software*. Esses princípios de engenharia fundamentam o desenvolvimento de sistemas que sejam confiáveis e que funcionem eficientemente em máquinas reais. Esses princípios englobam as tarefas desde o planejamento e estimativa do projeto, passando pela construção do produto de *software*, até sua manutenção durante o uso. As ferramentas de engenharia de *software* proporcionam suporte semi-automatizado ou automatizado a estes métodos. Essas ferramentas que apoiam o desenvolvimento de *software* são conhecidas como ferramentas CASE [PRES95].

Na última década, notou-se que, além da necessidade de apoiar os artefatos gerados durante o desenvolvimento de *software* através de ferramentas CASE, também há necessidade de se definir e controlar o próprio processo de *software*. Surgiram, então, os ambientes de engenharia de *software* orientados a processo PSEE, os quais consistem em adicionar uma visão automatizada e integrada dos processos de *software* aos ambientes de engenharia de *software* [OSTE87].

Um PSEE é um conjunto de mecanismos para apoiar o gerenciamento de processos de *software*. Esses ambientes envolvem o conceito de programação de processos e caracterizam-se por suportar a descrição e execução de processos de modo a auxiliar e controlar todas as atividades envolvidas na produção e manutenção de um produto de *software* [GIME94].

Um dos principais requisitos para um PSEE, conforme apresentado na Seção 4.1, é a definição clara e explícita do processo através da qual os itens de *software* serão concebidos, projetados, desenvolvidos e distribuídos. Esta definição explícita do processo de *software* é representada por um modelo de processo [CARN99]. Um modelo de processo inclui a descrição dos recursos usados no processo de *software*, das regras aplicadas a ele,

das atividades e tarefas que estruturam o processo de *software* e qualquer outra informação útil para descrevê-lo [OSTE87].

Um PSEE também deve possuir facilidades para apoiar tanto a definição quanto a avaliação dos modelos de processos e suas interações com ferramentas de desenvolvimento de *software* [CARN99].

Geralmente, um PSEE tem como componentes básicos um repositório central onde todos os objetos produzidos durante o desenvolvimento de *software* são armazenados e uma interface pública para ferramentas que facilita a integração de ferramentas no ambiente, conforme mostra a Figura 4.1.

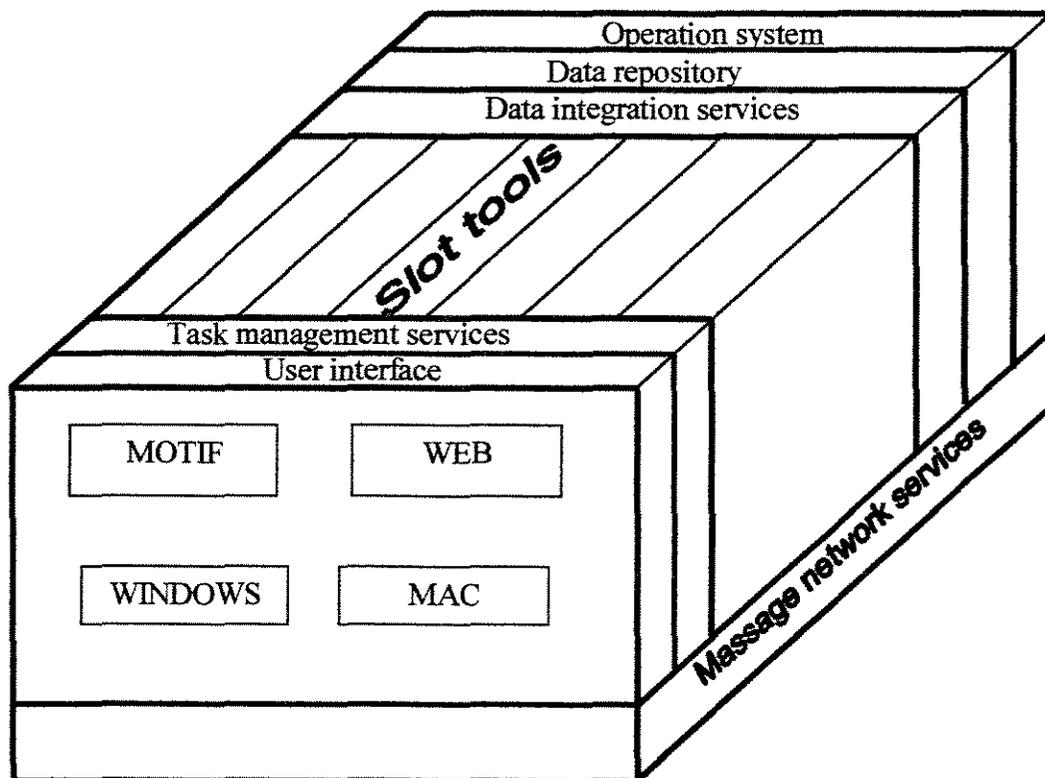


Figura 4.1: Arquitetura de um PSEE [BOOC99].

Um PSEE deve possuir algumas características básicas, a saber [GIME94]:

- permitir adaptações que o tornem mais adequado a um determinado projeto e às preferências de seus usuários;
- poder reutilizar em outros processos os componentes que determinam a estrutura do ambiente;
- prover o controle de acesso e a evolução de objetos compartilhados;
- apoiar o controle de versões e configurações;

- prover acesso a mecanismos de comunicação, como mensagens e conferências eletrônicas ao pessoal envolvido no desenvolvimento de *software*;
- poder ser utilizado no desenvolvimento de um vasto domínio de aplicações e apoiar o desenvolvimento de *software* em diferentes escalas;
- possuir uma interface gráfica consistente, com apresentação uniforme e utilização de técnicas e recursos avançados. Deve também possuir uma interface interna consistente de forma a permitir que uma ferramenta possa se comunicar perfeitamente com outra, possibilitando a passagem de informações entre elas, além de permitir que ferramentas possam ser facilmente introduzidas e/ou substituídas no ambiente;
- ser aberto, permitindo que novas ferramentas possam ser incorporadas à medida que forem requisitadas;
- apoiar o desenvolvimento do produto ao longo de todo o processo de *software* e não apenas em determinadas etapas;

Um PSEE é baseado em um modelo de processo de *software* que representa a abstração escolhida do processo de *software*. Esses ambientes utilizam um modelo de processo que permite a descrição de regras, procedimentos, interações entre usuários, artefatos e requisitos. Além disso, uma linguagem para programação de processos é necessária para definição do processo, de acordo com o modelo de processo. Através desta linguagem o processo pode ser especificado e executado [GIME94].

Existem diversos PSEEs focados na modelagem do processo de *software* e tecnologias relacionadas. Finkelstein em [FINK94] e Bandinelli em [BAND92] descrevem os principais para modelagem do processo de *software*. A Tabela 4.1 apresenta uma comparação desses ambientes segundo a abordagem da linguagem de programação de processos utilizada em cada ambiente. Esta comparação utiliza os seguintes critérios:

Ru – *logic rules* – denota abordagem baseada em regras e baseada em lógica.

Gr – *grammars attribute*.

At – *automata* – inclui autômatos de estado finito e redes de Petri (*Petri nets*)

Ip – *imperative programming languages*

AI – *artificial Intelligence* – refere-se àquelas linguagens que utilizam técnicas de Inteligência artificial ou sistemas baseados em conhecimento.

Et – *event-triggering* – formalismos baseados em eventos.

Adt – *abstract data types* – os tipos de dados abstratos incluem abordagem orientada a objetos para estruturação dos dados.

Observando-se a Tabela 4.1, nota-se que a maioria dos ambientes suporta mais de um estilo de programação do processo de *software*. Os estudos de caso utilizados para esta comparação [FINK94], mostraram que as linguagens baseadas em redes de Petri são capazes de modelar paralelismo e não determinismo natural e explicitamente. Isso não é verdade para as linguagens baseadas em regras, onde o controle da execução é implícito. Por outro lado, em um sistema baseado em regras, a modificação dinâmica do processo é permitida de maneira natural e homogênea.

Tabela 4.1: Classificação de linguagens de processos de *software* [FINK94].

Ambientes	Estilo de Linguagem						
	Ru	Gr	At	Ip	AI	Et	adt
Adele-Tempo						X	X
ALF	X					X	X
APPL/A				X			X
Entity			X				
EPOS	X			X	X		X
FUNSOFT			X	X			X
HFSP		X					
Marvel	X				X		
Merlin	X						
Oikos	X				X		
PADM	X				X		
SMART	X			X	X		X
SPADE	X		X				X
Process Weaver			X	X			

Segundo esses estudos, nenhum ambiente fornece uma solução satisfatória para todas as características requeridas para uma linguagem de especificação do processo de *software*. Essa imaturidade das linguagens de modelagem de processos deve-se a ser uma área bastante jovem e Bandinelli em [BAND92] descreve algumas questões que permanecem abertas:

- representar a interação com humanos;
- gerenciar falhas e eventos inesperados;
- garantir que as modificações do processo em tempo de execução (*on the fly*) sejam feitas de uma maneira disciplinada e controlada;
- integrar modelagem de processos e gerenciamento de processos (planejamento, monitoramento, etc.);
- integrar processos e modelagem de dados (banco de dados, transações longas, etc.).

### 4.3 – O ambiente ExpPSEE

O projeto ExpPSEE é desenvolvido pelo grupo de engenharia de *software* da UEM<sup>18</sup> o qual produziu um ambiente experimental de engenharia de *software* orientado a processo. Este projeto estabeleceu uma base de pesquisa na área de ambientes de

<sup>18</sup> UEM – Universidade Estadual de Maringá

engenharia de *software* que permite tanto investigações sobre os mecanismos do ambiente quanto sobre métodos e ferramentas para o desenvolvimento de *software*. A sua estrutura atual foi concebida com base em três projetos de pesquisa anteriores:

- construção de um Protótipo de um Ambiente de Engenharia de Software Orientado a Processos[GIME99];
- FORMLAB - Formal Laboratory [GIME98];
- AIGLE – um Ambiente de Engenharia de Software Evolutivo.

O ExpSEE é um ambiente de engenharia de *software* orientado a processos que possui uma arquitetura aberta e flexível. Suas características principais são:

- uma interface amigável orientada para definição e execução de processos de *software*;
- uma arquitetura que leva em conta a separação de domínios de gerenciamento;
- um modelo de processo de *software* que permite a definição de arquiteturas de processos que podem ser instanciadas e executadas;
- um mecanismo de apoio à construção, verificação e instanciação de arquiteturas de processo de *software* que facilita a reutilização de processos;
- um esquema de transição de estados de tarefas que leva em consideração os estados significativos das tarefas de engenharia de *software*;
- um ambiente de programação de processos de *software* cooperativos;
- um mecanismo de apoio à execução de processos de *software* auxiliado por agendas e
- uma base de dados orientada a objetos.

#### 4.3.1 – Arquitetura do Ambiente ExpSEE

Para especificação do ambiente, foi utilizada a linguagem de modelagem UML<sup>19</sup> [BOOC99]. A Figura 4.2 representa o diagrama de pacotes do ExpSEE. Os pacotes representam os responsáveis pelo gerenciamento de processos, gerenciamento de banco de dados e gerenciamento da interface com o usuário [CARN99].

---

<sup>19</sup> UML – Unified Modeling Language – Linguagem de Modelagem Unificada

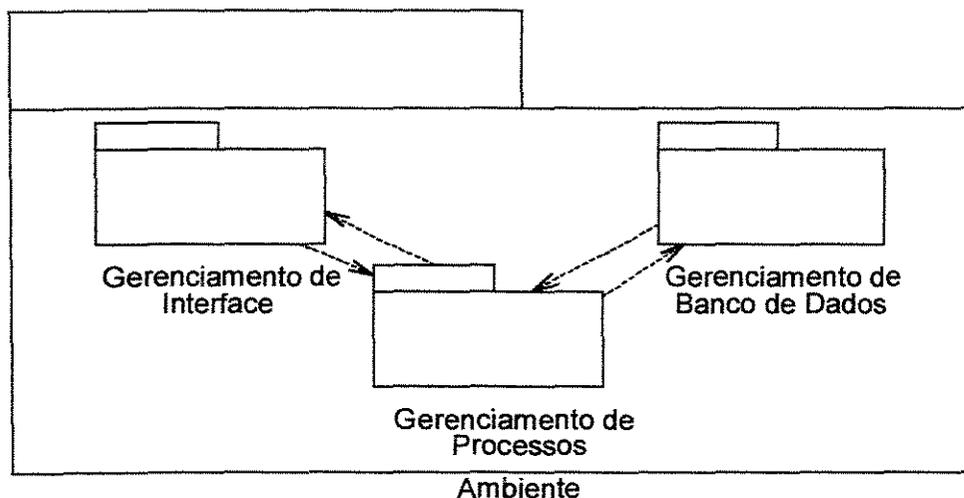


Figura 4.2: Diagrama de Pacotes do Ambiente ExpPSEE [CARN99].

Gerenciador de processos: é responsável por controlar a criação, a modificação e a execução dos processos de *software*. Também é responsável por controlar a disponibilidade e alocação dos recursos utilizados nas tarefas, bem como por controlar os cargos<sup>20</sup> humanos do processo, assegurando que nenhum direito seja violado [CARN99].

Gerenciador de banco de dados: corresponde aos serviços realizados pelo sistema gerenciador de banco de dados orientado a objetos *ObjectStore* [OBJE95], o qual é empregado no controle persistente dos artefatos gerados ao longo do processo de *software*.

Gerenciador da interface: é utilizado pelo gerenciador de processos para fazer a comunicação com o usuário através de uma interface.

### 4.3.2 – O Gerenciador de Processos

A execução das atividades pelo gerenciador de processos é baseada no conceito de separação de domínios de gerenciamento, sendo assim, o gerenciador de processos é composto por um conjunto de módulos gerenciadores, que interagem entre si, comunicam-se com o sistema gerenciador de banco de dados *ObjectStore* e com a interface [CARN99], conforme apresentado na seção anterior. Os módulos gerenciadores do gerenciador de processos são responsáveis por controlar e gerenciar os elementos envolvidos no processo e por registrar e manter as relações entre os objetos existentes no ambiente.

A Figura 4.3 representa o diagrama de classes do gerenciador de processos, o qual pode ser visto como composto de três partes [CARN99]:

<sup>20</sup> Utilizamos cargo como tradução do inglês *role*.

- parte superior: representa as classes referentes aos módulos do gerenciador de processos;
- parte central: representa as classes responsáveis pela definição de arquiteturas de processos de *software*;
- parte inferior: representa as classes responsáveis pela instanciação de arquiteturas em processos de *software*.

Os módulos do gerenciador de processos são descritos a seguir:

- Gerenciador de Projetos - responsável pelo controle e gerenciamento da execução de projetos de *software*;
- Gerenciador de Arquiteturas de Processo - responsável pelo controle e gerenciamento da construção e manutenção de arquiteturas de processo de *software* e sua instanciação através da definição de processos de *software*;
- Gerenciador de Tarefas - responsável pelo controle e gerenciamento das tarefas a serem realizadas no processo de *software*;
- Gerenciador de Artefatos - responsável pelo controle e gerenciamento dos artefatos utilizados e produzidos pelas tarefas através das ferramentas;
- Gerenciador de Ferramentas - responsável pelo controle e gerenciamento das ferramentas utilizadas pelas tarefas no processo de *software*;
- Gerenciador de Cargos - responsável pelo controle e gerenciamento dos cargos existentes no processo de *software* a serem ocupados pelos atores presentes no processo;
- Gerenciador de Ações - responsável pelo controle e gerenciamento das ações que as ferramentas desempenham sobre os artefatos do processo de *software*;
- Gerenciador de Atores - responsável pelo controle e gerenciamento dos atores, e suas agendas, envolvidos no processo de *software*.

O diagrama de classes do gerenciador, conforme mostra Figura 4.3, propõe que uma arquitetura de processo pode possuir um ou mais tipos de tarefas, artefatos, ferramentas, cargos, ações ou direitos. Cada arquitetura tem um relacionamento direto com os tipos de tarefas e indireto, através dos tipos de tarefas, com os demais tipos apresentados no diagrama, que são recursos e restrições para a sua realização.

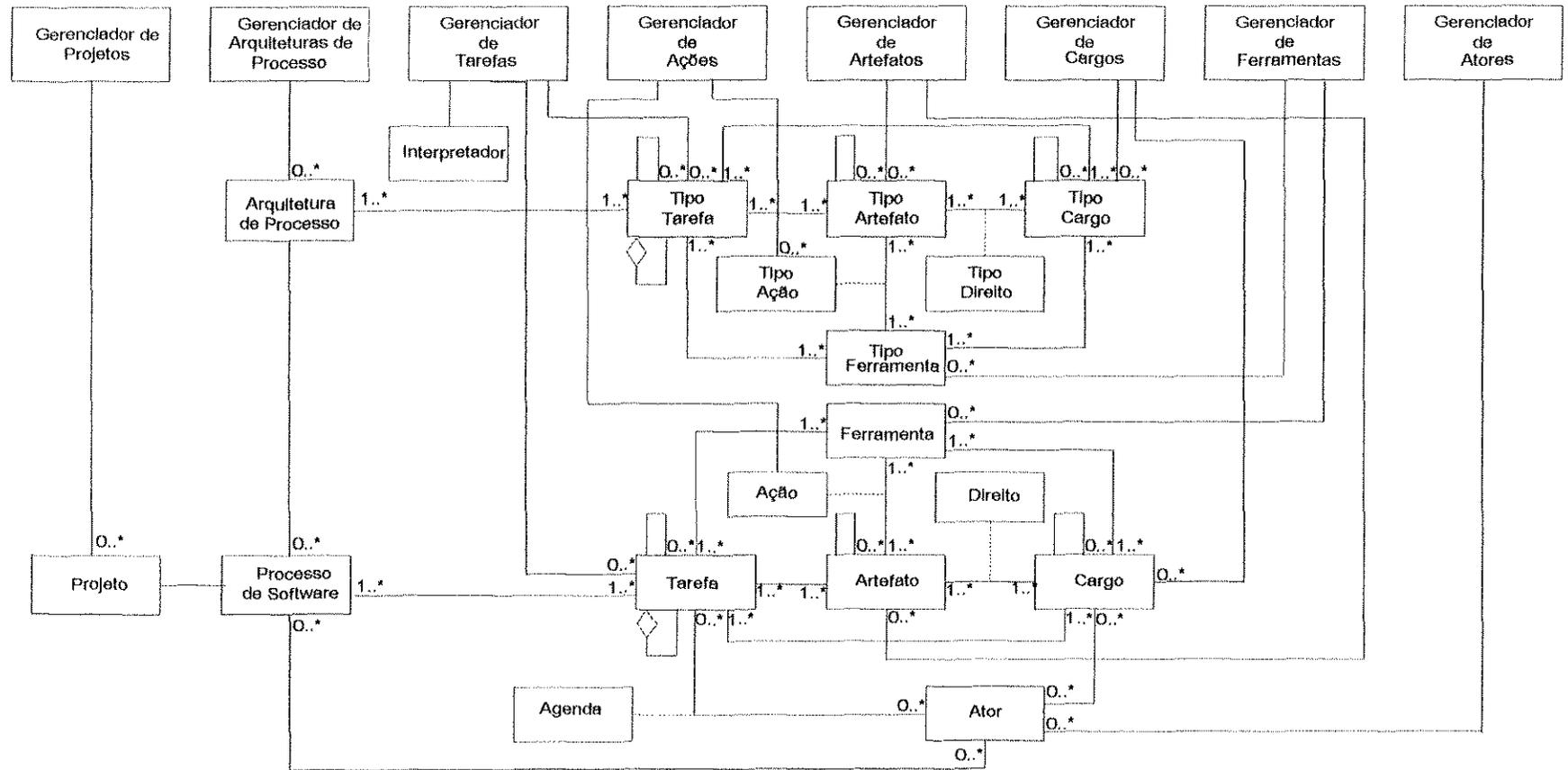


Figura 4.3: Diagrama de classes do gerenciador de processos do ExPSEE [CARN99].

Os tipos de objetos envolvidos na definição de arquiteturas de processos possuem relações entre si. Um tipo de tarefa pode ser precedido e/ou composto por zero ou mais tipos de tarefas. Esses tipos de tarefas devem utilizar um ou mais tipos de artefatos para a produção de um ou mais tipos de artefatos, sendo assim, cada tipo de artefato pode depender de zero ou mais tipos de artefatos. Para esta ação de transformação, os tipos de tarefas devem utilizar um ou mais tipos de ferramentas e requerer a utilização de um ou mais tipos de cargos. Um tipo de cargo pode ser liderado por um ou mais outros tipos de cargos. Além disso, os tipos de cargos devem manipular um ou mais tipos de ferramentas e acessar um ou mais tipos de artefatos, segundo um conjunto de tipos de direitos (leitura, escrita e/ou execução). Os tipos de ferramentas devem manusear um ou mais tipos de artefatos segundo um conjunto de tipos de ações válidas sobre estes tipos de artefatos.

Sobre essas arquiteturas de processos definidas segundo a parte central do diagrama de classes do gerenciador, vários processos de *software* podem ser instanciados. Dessa forma, todos os tipos de objetos existentes na arquitetura são instanciados em objetos que comporão o processo de *software*. Esses objetos são: tarefa, artefato, ferramenta, cargo, ação e direito, os quais relacionam-se entre si do mesmo modo que seus respectivos tipos relacionam-se na arquitetura do processo. Além desses objetos, os processos de *software* instanciados devem possuir dois outros objetos necessários para a sua execução. Eles são: ator e agenda, os quais também possuem relações entre si e com os demais objetos existentes no processo. Analogamente, o diagrama de classes do gerenciador propõe que um processo de *software* deve possuir um(a) ou mais tarefas, artefatos, ferramentas, cargos, ações, direitos, atores e agendas.

Para cada tarefa, deve ser alocado um ator responsável pela sua execução. Os atores devem ocupar um ou mais cargos, enquanto que os cargos devem ser ocupados por um ou mais atores, definindo assim quais são os direitos e deveres que cada ator possui. Cada ator possui associado a si uma agenda onde estão relacionadas as tarefas destinadas a ele e os dados mais importantes relacionados a estas tarefas, tais como: nome, *status*, relacionamentos, o procedimento a ser executado, datas de início e término e carga horária destinada para sua execução.

Além de todas essas classes, existe ainda a classe referente ao interpretador, a qual está relacionada diretamente com a classe do gerenciador de tarefas. O interpretador é responsável pela execução dos programas de processos, escritos na linguagem de programação de processos utilizada no ExPSEE, um subconjunto da linguagem TCL/TK.

#### **4.3.3 – Definição de Arquiteturas e Instanciação de Processos**

O controle oferecido pelo gerenciador de arquiteturas de processo permite a definição de arquiteturas de processo de *software* e, com base nestas arquiteturas, a instanciação e a execução de processos de *software*. No Capítulo 5 é apresentado um estudo de caso, os passos necessários e telas exemplo para definição da arquitetura, instanciação do processo e execução das tarefas.

O gerenciador de arquiteturas de processo possui duas interfaces com o usuário. O *Process Architecture Builder*, responsável pela definição de arquiteturas de processo e o *Project Manager*, responsável pela criação e gerenciamento de projetos de *software*, através da instanciação e execução desses processos.

Através do *Process Architecture Builder*, arquiteturas de processo, compostas por tipos de objetos e seus relacionamentos podem ser definidas. Para que estas arquiteturas possam ser instanciadas em processos de *software*, elas devem estar completas. Uma arquitetura de processos estará completa quando possuir pelo menos um relacionamento referente a cada um dos relacionamentos definidos para o seu tipo no diagrama de classes do gerenciador. Essa restrição é verificada por um *Process Inspector*. Sem a aprovação desse inspetor a arquitetura não poderá ser instanciada.

A instanciação de processos de *software* é feita pelo *Project Manager* que é utilizado para criar e gerenciar projetos de *software*. Ao se iniciar um novo projeto é necessário, primeiramente, escolher uma arquitetura de processo completa já existente ou criar uma nova arquitetura, para que ela possa ser instanciada em um processo de *software*.

Esse processo de instanciação é feito de modo automático pelo gerenciador de arquiteturas de processo, que para cada tipo de objeto existente na arquitetura, instancia um objeto no processo de *software*. O mesmo ocorre com cada relacionamento existente entre dois tipos de objetos que passará a existir entre os dois objetos instanciados. Cada objeto instanciado, recebe por padrão o mesmo nome, versão e descrição de seu tipo, podendo ser mudado antes de iniciar a sua execução.

Após o processo de *software* ser instanciado, as tarefas deste processo já podem ser programadas.

#### 4.3.4 – Diagrama de Estados das Tarefas

Cada tarefa do processo de *software* definida no ambiente ExpPSEE, passa por um conjunto de estados conforme mostra a Figura 4.4. A seqüência de transições é controlada pelo escalonador de tarefas do ExpPSEE.

Assim que um processo é instanciado, suas tarefas estão no estado instanciada. Neste estado inicial a tarefa tem sua estrutura definida através de seus relacionamentos com outros objetos do processo: artefatos necessários e a serem produzidos, ferramentas a serem utilizadas e cargos requeridos, de acordo com a arquitetura do processo. Porém, a tarefa ainda não tem o seu procedimento programado para que possa ser executada.

Para programar a tarefa, o gerente de projeto, que é o responsável por esta atividade, invoca o ambiente de programação de processos do ExpPSEE. Os dados necessários para a programação das tarefas estão definidos no processo a partir de suas arquiteturas. Esses dados são os cargos, ferramentas e artefatos relacionados à tarefa. A tarefa é, então, implementada no ambiente de programação e vai para o estado codificada quando o seu procedimento estiver programado. Porém, ela ainda não tem todos os recursos

necessários para ser executada, e portanto, permanecerá neste estado até que tenha estes recursos alocados.

Quando a tarefa tiver todos os recursos (ex. artefatos, ferramentas e atores) necessários à sua execução alocados, ela muda para o estado alocada. Neste estado, a tarefa é incluída na agenda dos atores que devem executá-la. No entanto, para uma tarefa estar no estado pronta é necessário que ela tenha os artefatos de entrada disponíveis, e se existirem, suas tarefas precedentes prontas. Sua permanência neste estado depende dos atores que devem executá-la. Quando um ator escolhe uma tarefa de sua agenda para executar, esta tarefa passa ao estado executando. Então, o gerenciador de tarefas realiza uma chamada ao interpretador para que este possa executar o procedimento da tarefa.

Para que as transições possam ocorrer, um conjunto de pré-condições definido para cada tarefa, deve ser satisfeito. Uma descrição mais detalhada do diagrama de estados das tarefas da Figura 4.4, bem como do conjunto de pré-condições necessários para se alcançar cada um dos estados podem ser encontrados são descritas por Gimenes em [CARN99].

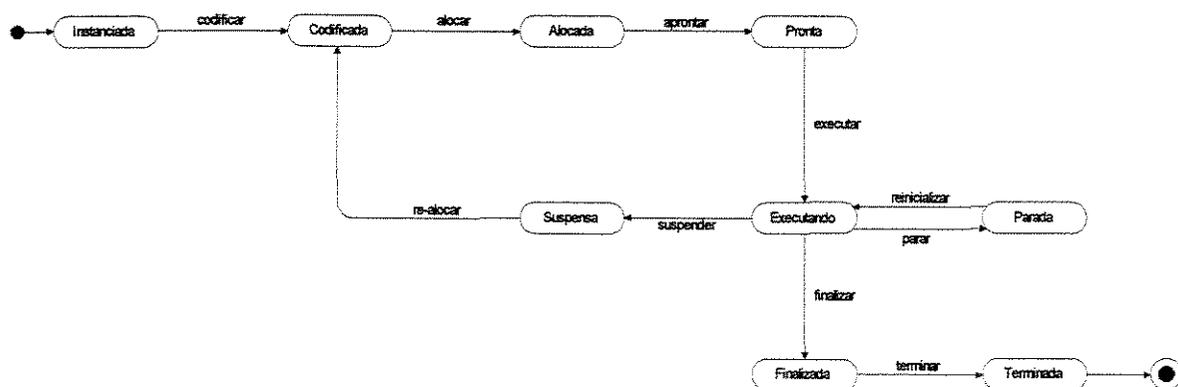


Figura 4.4: Diagrama de Estados das Tarefas [CARN99].

#### 4.4 – Comentários finais

Neste capítulo foram apresentados os conceitos relacionados ao processo de *software* e introduzido o conceito de ambiente de engenharia de *software* centrado no processo, o PSEE. Foi apresentado um comparativo entre os principais PSEEs que atualmente estão sendo pesquisados.

Finalmente, foi apresentado o ExpPSEE, o PSEE que será utilizado neste trabalho. Foram listadas suas principais características, apresentada sua arquitetura e descritos seus pacotes.

O que há de comum entre os ambientes é que nenhum deles atende por completo as necessidades de programação do processo de *software* [BAND92]. De acordo com o estilo de construção os PSEEs fortalecem uma ou outra funcionalidade, porém deixam a desejar em outra. Portanto a utilização do ExpPSEE neste trabalho é perfeitamente adequada e as dificuldades porventura encontradas serão descritas no estudo de caso do Capítulo 6, em outro ambiente poderiam não existir, mas com certeza outras apareceriam.

## Capítulo 5

### MAPS-15504: Uma metodologia de avaliação de processo de *software* baseada no ISO/IEC TR 15504

Neste capítulo é apresentada uma metodologia de avaliação de processo de *software*. Esta apresentação está dividida em duas partes. A primeira parte consiste de uma descrição do processo de avaliação baseada nos requisitos da parte 3 do ISO/IEC TR 15504, apresentando uma arquitetura de processos e suas tarefas compatível com esta parte 3. A segunda parte apresenta uma descrição detalhada de cada tarefa sob três enfoques: as definições segundo o ISO/IEC TR 15504, as definições que não estão no ISO/IEC TR 15504, mas que são necessárias para que a metodologia esteja completa e as definições necessárias para execução da metodologia no ambiente do ExpPSEE. Embora repetitivo, fazem-se necessárias estas definições para cada tarefa da metodologia pois estão diretamente relacionadas com a implementação no ExpPSEE. A partir das necessidades oriundas destas definições, juntamente com as definições do ISO/IEC TR 15504 são retiradas importantes conclusões. Um modelo de cada documento definido pela metodologia é apresentado no anexo A.

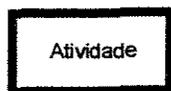
A metodologia é a base para que se possa automatizar o ISO/IEC TR 15504 por meio de um PSEE, neste caso o ExpPSEE conforme apresentado no Capítulo 4.

#### 5.1 O processo de avaliação

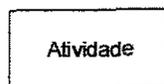
Antes da descrição das tarefas do processo de avaliação, é apresentada a notação da linguagem gráfica escolhida para representação das tarefas. Esta notação foi a utilizada no projeto PRONET [CHRI95], pois:

- seus elementos são definidos para apoiar a especificação do processo de *software*, mas podem ser utilizados para outras especificações que não são *software*;
- sua notação gráfica é de fácil entendimento por pessoas não técnicas;
- um gerenciamento de versões dentro de um contexto definição / execução é fornecido, além de suportar persistência de entidades geradas durante a execução;
- notações diretas dentro da execução e verificação do processo são disponibilizadas;
- notação é precisa e não ambígua.

Os elementos básicos que compõem a notação PRONET são:



**Atividade:** representa uma atividade a qual é composta de outras atividades (subatividades).



**Atividade:** representa uma atividade para o ambiente PRONET. No contexto deste trabalho será equivalente a uma tarefa.



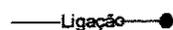
**Artefato:** um artefato (por exemplo um documento) pode ser produzido por uma atividade ou utilizado por ela.



**Agente:** um agente pode ser uma pessoa, um ator humano, ou uma ferramenta que deva ser inserida no processo.



**Condição:** pode ser necessária para iniciar ou ser o resultado de uma atividade. Uma condição retorna verdadeiro ou falso.



**Relacionamento:** relacionamentos ligam as entidades às atividades. Podem ser de dois tipos: entrada e saída.



**Composições:** são combinações booleanas de condições, artefatos e agentes. São usados como elementos de sincronização. Podem ser:

- CA - convergente and.
- CO - convergente or.
- DA - divergente and.
- DO - divergente or.

Será adotada a seguinte nomenclatura para melhor referenciar os elementos que compõem a metodologia:

- TA1 - representa a tarefa de número 1. Caso ela seja composta de outras tarefas, serão acrescentados tantos subitens quantos sejam os níveis que ela representa. Por exemplo: TA1.1 significa a primeira tarefa que compõe a tarefa TA1.
- AR1 - representa o artefato de número 1. Um modelo de cada artefato referenciado na metodologia será apresentado no anexo 1.
- AG1 - representa o agente número 1.
- CO1 - representa a condição número 1.

O processo de avaliação de processo de *software* proposto na Figura 5.1, é composto por três tarefas distintas: TA1 Definir entradas, TA2 Definir processo de avaliação e TA3 Apresentar resultados. A metodologia proposta neste trabalho está centrada nas subtarefas da tarefa TA2 Definir processo de avaliação. As tarefas TA1 Definir entradas e TA2 Apresentar resultados estão descritas para consistência da metodologia.

Está fora do escopo do ISO/IEC TR 15504 definir uma metodologia de avaliação de processos de *software* [ISO99-5]. Esta norma define apenas requisitos para a avaliação de processos de *software* [ISO99-3]. É com base nestes requisitos que as tarefas que compõem a tarefa TA2 Definir processo de avaliação foram elaboradas. As seções deste capítulo tratam de detalhar cada uma das tarefas da Figura 5.1.

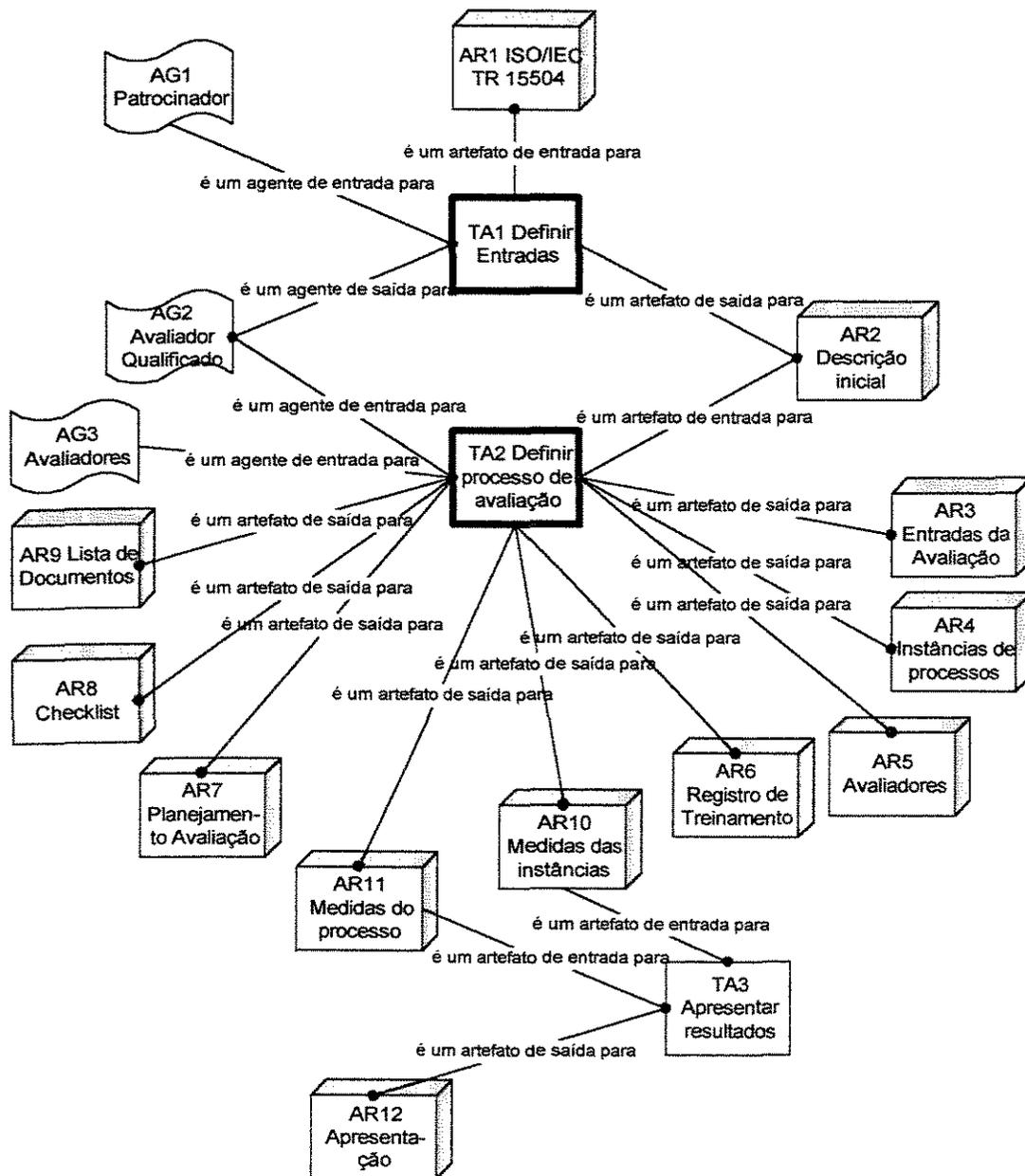


Figura 5.1: Arquitetura geral do processo de avaliação.

### 5.1.1 Detalhamento da tarefa Definir entradas

A tarefa TA1 Definir entradas é composta de duas tarefas, conforme mostra a Figura 5.2. A tarefa TA1.1 Descrever avaliação é de responsabilidade da organização no papel do AG1 Patrocinador e precede o processo de avaliação propriamente dito. Nessa tarefa deve-se descrever no documento AR2 Descrição inicial o propósito da avaliação, as unidades organizacionais a serem avaliadas (setores ou departamentos) e contratar o AG2 Avaliador qualificado. O avaliador qualificado pode ser uma segunda parte contratada ou um funcionário treinado pertencente a organização especificamente designado para esta função. A parte 6 do ISO/IEC TR 15504 [ISO99-6] define os requisitos para um avaliador qualificado.

A partir das informações da tarefa TA1.1 Descrever avaliação, o avaliador qualificado fará a descrição do processo de avaliação conforme descrito na tarefa TA1.2 Definir entradas da avaliação. O propósito da avaliação descrito no documento AR2 Descrição inicial deve ser revisado e transcrito no documento AR3 Entradas da avaliação.

Deve ser elaborado entre o patrocinador e o avaliador qualificado, um acordo de confidencialidade no qual os direitos sobre a avaliação e seus dados estão descritos e assinados por ambas as partes. Este acordo faz parte do documento AR2 Descrição inicial.

Após a tarefa TA1.2 Definir entradas é executado uma composição DA (*divergente and*). Significa que assim que a tarefa TA1.2 Definir entradas termine, outras tarefas se iniciarão simultaneamente, que são as tarefas TA2.1 Definir instâncias e TA 2.2 Definir avaliadores.

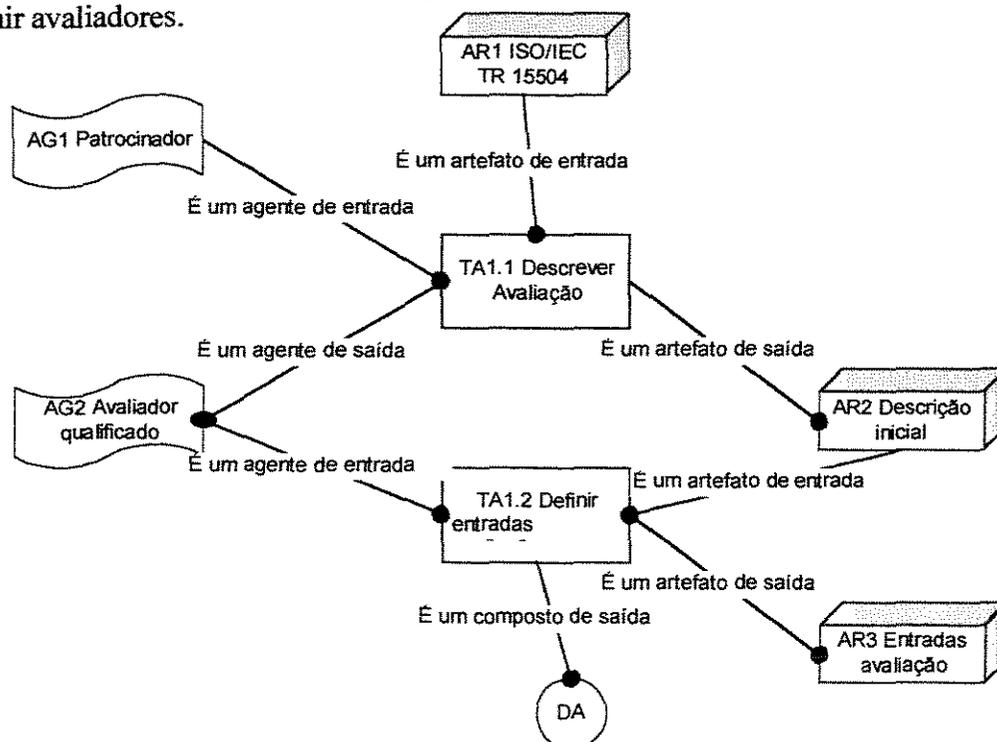


Figura 5.2 - Tarefas que compõem a tarefa TA1 Definir entradas.

### **5.1.2 Explosão da tarefa Processo de avaliação**

A Figura 5.3 representa graficamente o processo de avaliação de processo de *software*.

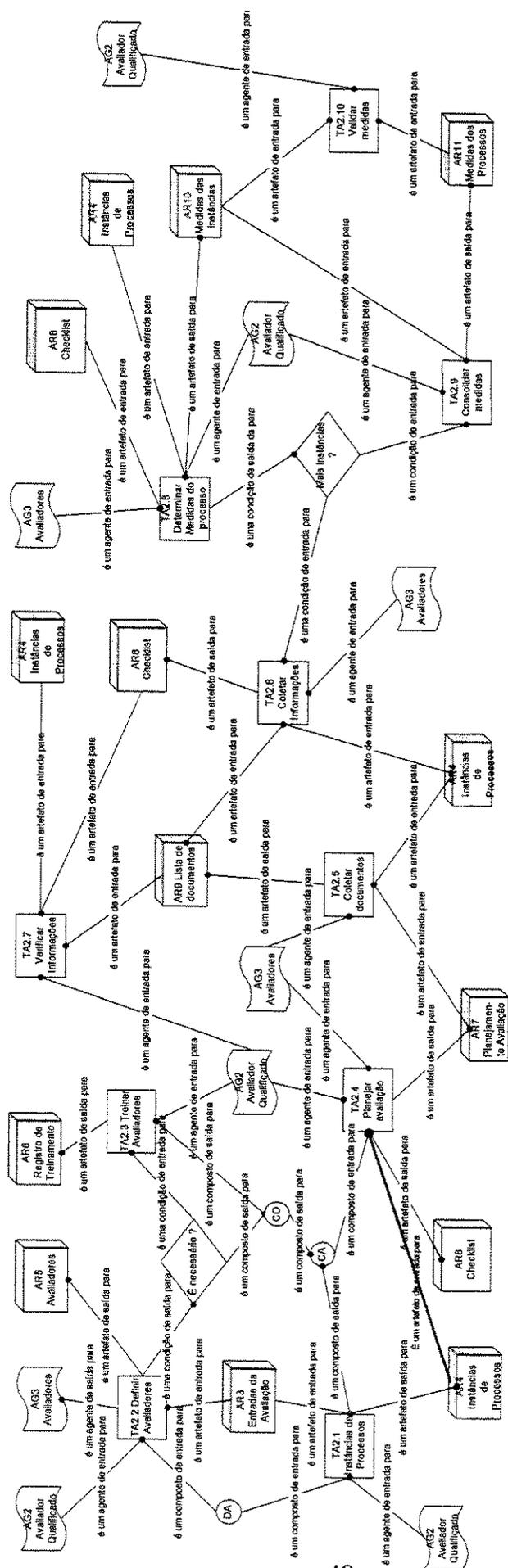


Figura 5.3 – Tarefas que compõem a tarefa TA2 Processo de avaliação.

A partir das entradas da avaliação, através da composição (elemento de sincronização) DA - *Divergente and*, são iniciadas as tarefas TA2.1 Definir instâncias e TA2.2 Definir avaliadores. A tarefa TA2.1 Definir instâncias deve escolher os projetos da organização a partir dos quais possa-se ter uma visão completa do processo de *software* desta, respeitados os limites definidos no documento AR3 Entradas da Avaliação. Estas instâncias de processo devem ser registradas em um documento denominado AR4 Instâncias de processo. O ISO/IEC TR 15504 não define parâmetros para esta seleção, apenas define que as instâncias devem ser selecionadas.

A tarefa TA2.2 Definir avaliadores, deve ser executada independentemente da tarefa TA2.1 Definir instâncias. O ISO/IEC TR 15504 define um mínimo de 2 e um máximo de 6 pessoas na equipe de avaliação, mais o avaliador qualificado. Esses avaliadores podem ser externos a organização a exemplo do avaliador qualificado, porém avaliadores internos também devem ser selecionados pois os mesmos conhecem a organização (setores, departamentos, chefias, cultura) e essas informações facilitarão a condução do processo de avaliação. A descrição desses avaliadores, seus papéis e sua formação devem ser registrados em um documento denominado AR5 Avaliadores. Caso o grupo de avaliadores não tenha conhecimento do ISO/IEC TR 15504, treinamento adequado deve ser ministrado. O AG2 Avaliador qualificado deve estar capacitado para ministrar este treinamento. O treinamento deve ser registrado em um documento denominado AR6 Registro de treinamento.

A tarefa TA2.4 Planejar avaliação só poderá ser iniciada uma vez terminadas as duas tarefas TA2.1 Definir instâncias e TA2.2 Definir avaliadores. Essa condição é definida pela composição CA – *convergente and*. Toda a equipe deve reunir-se sob a coordenação do avaliador qualificado e elaborar o planejamento da avaliação conforme definido pela tarefa TA2.4 Planejar avaliação. Esse planejamento deve ser registrado em um documento denominado AR7 Planejamento da avaliação e consiste basicamente de cronogramas de reuniões: data, hora, local, participantes e a pauta das reuniões. Nessa atividade também será elaborado o documento AR8 Checklist que será utilizado para coleta de dados. Para as avaliações que utilizam apenas o ISO/IEC TR 15504 as práticas base e as práticas de gerenciamento são utilizadas como *checklist*. Outros modelos compatíveis podem ser utilizados como guias auxiliares na avaliação, conforme definido na parte 4 do ISO/IEC TR 15504.

A tarefa TA2.5 Coletar documentos deve seguir o documento AR7 Planejamento da avaliação, com base no documento AR4 Instâncias de processo. Os documentos gerados ou utilizados nas instâncias de processo selecionados, devem ser coletados de maneira a gerar evidências que serão avaliados segundo as práticas base descritas no modelo de referência, parte 2, do ISO/IEC TR 15504. As práticas de gerenciamento serão coletadas nas entrevistas e obtidas através das impressões dos avaliadores coletadas ao longo do processo de avaliação. Os documentos selecionados devem ser registrados em um documento denominado AR9 Lista de documentos. Essa lista de documentos deve conter também um mapeamento mostrando os documentos equivalentes definidos no anexo C da parte 5 do ISO/IEC TR 15504. Esses documentos equivalentes fornecem uma série de

características desejáveis, mas não obrigatórias, que podem ser utilizadas como indicadores para o avaliador.

De posse do documento AR4 Instâncias de processo e do documento AR9 Lista de documentos, deve-se coletar informações juntamente aos responsáveis pelos projetos, nas unidades organizacionais onde são desenvolvidos. Este é um processo que se repete até que todas as instâncias de processos sejam verificadas e todas as dúvidas dos avaliadores sejam esclarecidas. Durante a execução da avaliação o AR8 Checklist é preenchido e as evidências são verificadas junto aos documentos coletados. As práticas de gerenciamento são confirmadas pelas impressões dos avaliadores coletadas durante a avaliação. Uma vez verificadas as informações, as instâncias de processos são medidas segundo as definições do modelo de referência, parte 2 do ISO/IEC TR 15504 e registradas em um documento denominado AR10 Medidas das instâncias. Caso hajam mais instâncias a serem avaliadas, a decisão deve seguir o caminho de volta a tarefa TA2.6 Coletar informações. Caso não hajam mais informações a serem coletadas, a tarefa TA2.9 Consolidar medidas deve ser executada.

Na tarefa TA2.9 Consolidar medidas, as medidas de todas as instâncias de processo devem ser reunidas dando o perfil do processo da organização. As medidas dos processos devem ser registradas no documento AR11 Medidas do processo. Estas medidas são validadas pelo avaliador qualificado e uma vez confirmadas finalizarão o processo de avaliação, obtendo-se os perfis que devem atender ao propósito da avaliação contido no documento AR3 Entradas da avaliação. Após essa tarefa, o processo de avaliação é finalizado e segue-se a tarefa TA3 apresentar resultados.

### 5.1.3 Tarefa TA3 Apresentar resultados

Esta tarefa sucede todo o processo de avaliação. De acordo com o ISO/IEC TR 15504, os resultados obtidos devem ser apresentados, respeitando-se o acordo de confidencialidade. Os resultados devem apresentar o perfil da avaliação de acordo com a determinação dos objetivos da avaliação e com o acordo de confidencialidade descritos no documento AR2 Descrição inicial. Os requisitos para esta tarefa estão descritos na Seção 4.5 da parte 3 do ISO/IEC TR 15504, devendo conter no mínimo:

- a data da avaliação;
- as entradas da avaliação;
- as instâncias de processo de *software* selecionadas e a justificativa da seleção;
- a abordagem utilizada na avaliação, incluindo a identificação do modelo de avaliação, método e qualquer instrumento de suporte ou ferramentas para suportar reclamações de discordância.
- os nomes dos membros da equipe que conduziram a avaliação, incluindo o avaliador qualificado responsável pela avaliação e a equipe de avaliadores;
- o conjunto de perfis de processo de *software* resultantes da avaliação;
- a localização de registros que suportam evidências objetivas e as medidas dos atributos de processo nos perfis de processo;

- as informações adicionais coletadas durante a avaliação que foram identificadas nas entradas da avaliação.

## 5.2 Descrição das tarefas

A seguir serão descritas em detalhes todas as tarefas que compõem o processo de avaliação. Este trabalho concentra-se no processo de avaliação, as tarefas TA1.1 Descrever avaliação, TA1.2 Definir entradas da avaliação e TA3 Apresentar resultados, descritas nas seções 5.2.1, 5.2.2 e 5.1.13 respectivamente, estão fora do escopo do processo de avaliação, conforme mostra a Figura 5.1. Elas estão aqui descritas para consistência na apresentação do modelo.

A descrição de cada tarefa está dividida em três partes:

- a) uma descrição segundo o ISO/IEC TR 15504. Para cada item foi acrescentado entre parênteses, negrito e itálico o item da norma ISO/IEC TR 15504 ao qual se refere a descrição, por exemplo *(4.2.a)*<sup>21</sup>.
- b) uma definição dos aspectos que faltam ao ISO/IEC TR 15504 para que a tarefa possa ser executada, e
- c) uma descrição segundo o ExpPSEE para que a tarefa possa ser programada e executada nesse ambiente. A execução das tarefas no ExpPSEE é realizada através do preenchimento de formulários utilizando-se de recursos do ambiente. Por isso, estão definidos neste item apenas as pré e pós-condições que devem ser verificadas, pois a execução da tarefa propriamente dita dar-se-á fora do ambiente ExpPSEE, ele trata apenas de fazer a invocação da ferramenta com o artefato alocado. A execução se dá fora do ambiente porque a estrutura dos documentos não estão definidas dentro do banco de dados do ExpPSEE, o Object Store, não sendo possível portanto, nenhuma inferência sobre os mesmos. As conseqüências desta execução estão descritas na análise do estudo de caso do Capítulo 6.

### 5.2.1 TA1.1 Descrever avaliação

Esta tarefa representa a decisão da organização em executar uma avaliação de seus processos segundo o ISO/IEC TR 15504. Essa decisão consiste em definir o patrocinador, o propósito e o escopo da avaliação conforme mostram os subitens abaixo. O documento de entrada para esta tarefa é a própria norma definida como o artefato AR1 ISO/IEC TR 15504, o qual será utilizado ao longo de toda a avaliação.

---

<sup>21</sup> (4.2.a) significa o Capítulo 4, Seção 2, item a da parte 3 do ISO/IEC TR 15504 [10 parte 3], o qual define os requisitos para execução de avaliação.

### 5.2.1.1 Definições segundo o ISO/IEC TR 15504

Esta é a tarefa inicial a organização deve tomar a decisão de realizar uma avaliação de seus processos e definir:

- o patrocinador da avaliação (4.2.a);
- o propósito da avaliação (4.2.b);
- o avaliador qualificado responsável pela avaliação, de acordo com os critérios definidos na parte 6 do ISO/IEC TR 15504, indicado pelo patrocinador (4.2.e, 4.3.P.1)<sup>22</sup>;

#### **Pré-condições**

Não estão definidas pré-condições para esta tarefa.

#### **Pós-condições**

Como pós-condições, devem estar definidos e documentados no artefato AR2 Descrição inicial, os seguintes itens:

- as Unidades organizacionais envolvidas na avaliação;
- o Propósito da avaliação;
- o AG1 Patrocinador da avaliação;
- o AG2 Avaliador qualificado;
- o contrato de confidencialidade assinado entre a organização, representada pelo patrocinador e o avaliador qualificado.

### 5.2.1.2 Definições que não estão no ISO/IEC TR 15504

Esta tarefa não faz parte do processo de avaliação propriamente dito. Ela faz parte do processo de preparação para avaliação. A abrangência do ISO/IEC TR 15504 não inclui esta tarefa.

### 5.2.1.3 Definições para o ExpPSEE:

A execução desta tarefa no ExpPSEE corresponde a edição do documento AR2 Descrição inicial pelo editor de textos.

#### **Pré-condições**

Não estão definidas pré-condições para esta tarefa.

---

<sup>22</sup> P indica o parágrafo (neste caso primeiro parágrafo) dentro da Seção 3 do parágrafo 4.

## **Pós-condições**

- o artefato AR2 Descrição inicial deve estar concluído.

## **Artefatos**

- AR2 Descrição inicial;

## **Papel**

- AG1 Patrocinador;

## **Ferramenta**

- Editor de textos.

## **Relacionamentos**

Ferramenta – Artefato:

Editor com AR2 Descrição inicial – ler, gravar, excluir;

Papel – Artefato:

Patrocinador com AR2 Descrição inicial – ler, gravar, excluir;

Papel – Ferramenta:

Patrocinador com Editor – executar;

### **5.2.2 TA1.2 Definir entradas da avaliação**

Esta tarefa será executada pelo avaliador qualificado definido e contratado na tarefa TA1.1 Descrever avaliação. O documento AR2 Descrição inicial elaborado pela organização deve ser revisto pelo AG2 Avaliador qualificado e adequado às definições do ISO/IEC TR 15504. O documento AR3 Entradas da Avaliação gerado nesta tarefa guiará todo o processo de avaliação.

#### **5.2.2.1 Definições segundo o ISO/IEC TR 15504**

Esta tarefa deve ser executada juntamente com o avaliador qualificado e descreverá todo o processo de avaliação. As entradas do processo de avaliação são definidas claramente no ISO/IEC TR 15504 e formam o documento AR3 Entradas da avaliação, o qual guiará todo o processo de avaliação. Este documento é elaborado tendo como base as definições elaboradas pela organização no documento AR2 Descrição inicial. O documento AR3 Entradas da avaliação deve ter as informações descritas a seguir.

- Definição da abrangência da avaliação incluindo (4.2.c):
  - os processos de *software* (categorias de processo) a serem investigados dentro da unidade organizacional (4.2.c.1);
  - o mais alto nível de capacidade a ser investigado para cada processo de *software* (4.2.c.2);
  - a unidade organizacional que executa estes processos de *software* (4.2.c.3);
  - o contexto do processo de *software* que deve conter pelo menos (4.2.c.4):
    - o tamanho da unidade organizacional;
    - o domínio de aplicação dos produtos ou serviços da unidade organizacional;
    - o tamanho, o quanto são críticos e a complexidade dos produtos e serviços;
    - as características de qualidade dos produtos ou serviços (exemplo: ISO 9126).
- Definição das restrições, as quais podem incluir (4.2.d):
  - a disponibilidade de recursos chaves (4.2.d.1);
  - a quantidade máxima de tempo a ser usada na avaliação (4.2.d.2);
  - o tamanho mínimo, o tamanho máximo ou um exemplo de tamanho específico de processo de *software* a ser selecionado, ou ainda o grau de cobertura que é desejada para avaliação (4.2.d.4);
  - o dono dos resultados da avaliação e as restrições em seu uso (4.2.d.5);
  - o controle sobre as informações resultantes do acordo de confidencialidade (4.2.d.6);
  - qualquer informação adicional resultante do acordo de confidencialidade (4.2.d.7).
- Informações adicionais que devam ser coletadas durante o processo de avaliação (4.2.6).

### **Pré-condições**

- A tarefa TA1.1 Descrever avaliação deve estar concluída.
- O documento AR2 Descrição inicial deve estar disponível.

### **Pós-condições**

- O documento AR3 Entradas da avaliação deve estar concluído.

#### 5.2.2.2 Definições que não estão no ISO/IEC TR 15504

Está fora da abrangência do ISO/IEC TR 15504 auxiliar na montagem do perfil da avaliação, esta é uma percepção que dependerá da experiência do avaliador qualificado, o qual poderá utilizar outras metodologias para fazê-lo.

### 5.2.2.3 Definições para o ExpPSEE

A execução desta tarefa no ExpPSEE corresponde a edição do documento AR3 Entradas da avaliação pelo editor de textos.

#### **Pré-condições**

- A tarefa TA1.1 Descrever avaliação deve estar concluída;
- O documento AR2 Descrição inicial deve estar disponível.

#### **Pós-condições**

- O artefato AR3 Entradas da avaliação deve estar concluído.

#### **Artefatos**

- AR2 Descrição inicial;
- AR3 Entradas da avaliação.

#### **Papel**

- AG1 Patrocinador;
- AG2 Avaliador Qualificado;

#### **Ferramenta**

- Editor de textos.

#### **Relacionamentos**

##### Ferramenta – Artefato

Editor de textos com AR3 Entradas da avaliação – ler, gravar, excluir;  
Editor de textos com AR2 Descrição inicial – ler, gravar

##### Papel – Artefato:

AG1 Patrocinador com AR3 Entradas da avaliação – ler, gravar, excluir;  
AG2 Avaliador Qualificado com AR3 Entradas da avaliação – ler, gravar,  
excluir

AG1 Patrocinador com AR2 Descrição inicial – ler, gravar, excluir;  
AG2 Avaliador Qualificado com AR2 Descrição inicial – ler, gravar;

##### Papel – Ferramenta:

AG1 Patrocinador com Editor de textos – executar;  
AG2 Avaliador Qualificado com Editor de textos – executar;

### 5.2.3 TA2.1 Definir instâncias

Esta tarefa, com base no documento AR2 Entradas da avaliação, deve definir quais as instâncias de processo de *software* (projetos) da organização devem fazer parte do processo de avaliação. As instâncias de processos devem representar os processos da organização de maneira que os objetivos da avaliação possam ser alcançados.

#### 5.2.3.1 Definições segundo ISO/IEC TR 15504

O ISO/IEC TR 15504 define o conteúdo do documento AR3 Instâncias de processos, conforme descrito abaixo:

- as instâncias de processo de *software* específicas que devam ser excluídas no processo de avaliação (4.2.d.3);
- quaisquer outras pessoas com responsabilidades específicas de apoiar o processo de avaliação (4.2.5, 4.4.P.2);
- as instâncias de processos de *software* escolhidas dentro da unidade organizacional, incluindo justificativa para a escolha destas (4.4.P.5);
- o perfil da avaliação para cada processo de *software* escolhido, de acordo com o modelo definido na parte 2 do ISO/IEC TR 15504 (4.4.P.3);
- a correspondência entre os processos organizacionais e o perfil da avaliação definido (4.4.P.4);
- os responsáveis em cada unidade organizacional que possuam instâncias de processos de *software* selecionados para avaliação (4.4.P.9.b).

#### Pré-condições

- A tarefa TA2.1 Definir entradas avaliação deve estar concluída.
- O documento AR3 Entradas da avaliação deve estar disponível.

#### Pós-condições

- O artefato AR4 Instâncias de processos concluído.

#### 5.2.3.2 Definições que não estão no ISO/IEC TR 15504

O ISO/IEC TR 15504 não fornece indicações sobre a seleção dos processos de *software* e das unidades organizacionais de maneira que melhor representem o perfil da avaliação de maneira a atender aos requisitos do documento AR3 Entradas da avaliação. Também não é dada nenhuma indicação sobre o contexto dos processos (ex: domínio, tamanho e a influência do quanto são críticos) ou unidades organizacionais que melhor se

adeqüem a avaliação desejada. A exemplo do item 5.2.2.2, outras metodologias devem ser utilizadas para esta definição.

### 5.2.3.3 Definições para o EXPSEE

A execução desta tarefa no ExpPSEE corresponde a edição do documento AR4 Instâncias de processos pelo editor de textos.

#### **Pré-condições**

- A tarefa TA1.2 Definir entradas da avaliação deve estar concluída.
- O documento AR3 Entradas da avaliação deve estar disponível.

#### **Pós-condições**

- O artefato AR4 Instâncias de processos deve estar concluído.

#### **Artefatos**

- AR3 Entradas da avaliação.
- AR4 Instâncias de processos.

#### **Papel**

- AG2 Avaliador qualificado.

#### **Ferramenta**

- Editor de textos.

#### **Relacionamentos**

Ferramenta – Artefato:

Editor de textos com AR4 Instâncias de processos – ler, gravar, excluir.  
Editor de textos com AR3 Entradas da avaliação – ler.

Papel – Artefato:

AG2 Avaliador qualificado com AR4 Instâncias de processos – ler, gravar, excluir;  
AG2 Avaliador qualificado com AR3 Entradas da avaliação – ler, gravar;

Papel – Ferramenta:

AG2 Avaliador qualificado com Editor de textos – executar.

## 5.2.4 TA2.2 Definir avaliadores

Esta tarefa deve compor a equipe de avaliação. Esta equipe é formada pelo avaliador qualificado e pelos avaliadores, documentados em AG2 Avaliador qualificado e AG3 Avaliadores, respectivamente. A equipe pode ser contratada externamente à organização ou ser treinada internamente para desenvolver um processo de melhoria contínua, tomando como base a avaliação dos processos de *software*. Mesmo que uma equipe externa seja contratada, recomenda-se que alguns membros da organização façam parte da mesma, pois o conhecimento de setores, departamentos, chefias e da cultura organizacional facilitará o trabalho de avaliação e coleta de evidências, sejam elas documentos ou impressões durante as entrevistas. O número de membros da equipe dependerá da abrangência da avaliação definida no documento AR3 Entradas da avaliação.

### 5.2.4.1 Definições segundo o ISO/IEC TR 15504

O ISO/IEC TR 15504 define um mínimo de 2 e um máximo de 6 pessoas na equipe de avaliação. Além de definir um perfil completo do avaliador qualificado em [ISO99-6], estabelece que o documento avaliadores seja elaborado com base no documento AR2 Entradas da avaliação e deve conter as definições de :

- responsabilidades específicas de cada membro da equipe de avaliação abrangendo:
  - conhecimentos necessários para a execução da avaliação (4.4.P.2);
  - necessidades de treinamento (4.4.P.2);

#### Pré-condições

- A tarefa da norma TA2.1 Definir entradas da avaliação deve estar concluída.
- O artefato AR3 Entradas da avaliação deve estar disponível.

#### Pós-condições

- O artefato AR5 Avaliadores deve estar concluído.

### 5.2.4.2 Definições que não estão no ISO/IEC TR 15504

O ISO/IEC TR 15504 na sua parte 6, define todos os requisitos necessários para um avaliador qualificado, bem como uma recomendação sobre o tamanho da equipe de avaliação. Porém o ISO/IEC TR 15504 não possui nenhum requisito sobre os outros membros da equipe, qual o nível de conhecimento ou experiência desejáveis (ou mínimos) para a execução de uma avaliação satisfatória.

#### 5.2.4.3 Definições para o ExpPSEE:

A execução desta tarefa no ExpPSEE corresponde à edição do documento AR5 Avaliadores pelo editor de textos.

#### **Pré-condições**

- A tarefa TA1.2 Definir entradas da avaliação deve estar concluída.
- O documento AR3 Entradas da avaliação deve estar disponível.

#### **Pós-condições**

- O artefato AR5 Avaliadores deve estar concluído.

#### **Artefatos**

- AR3 Entradas da avaliação.
- AR5 Avaliadores.

#### **Papel**

- AG2 Avaliador qualificado.

#### **Ferramenta**

- Editor de textos.

#### **Relacionamentos**

Ferramenta – Artefato:

Editor de textos com AG2 Avaliador qualificado – ler, gravar, excluir.

Papel – Artefato:

AG2 Avaliador qualificado com AR5 Avaliadores – ler, gravar, excluir;

Papel – Ferramenta:

AG2 Avaliador qualificado com Editor de textos – executar;

#### **5.2.5 TA2.3 Treinar avaliadores**

Esta tarefa será executada condicionalmente ao conteúdo do documento AR5 Avaliadores. Caso a equipe de avaliadores não possua o conhecimento da metodologia de avaliação de processos de *software* e do ISO/IEC TR 15504 necessários para a execução da

avaliação esta tarefa deverá ser executada, caso contrário não. Esta é uma decisão que cabe ao avaliador qualificado o qual deve estar capacitado para ministrar este treinamento.

#### 5.2.5.1 Definições segundo o ISO/IEC TR 15504

Esta tarefa deve ser registrada no documento AR6 Registro de treinamento, conforme modelo no anexo A, cujas definições segundo o ISO/IEC TR 15504 são:

- Planejar o treinamento dos avaliadores (4.4.P.2).
- Executar o treinamento planejado para os avaliadores (4.4.P.2).

#### **Pré-condições**

- A tarefa TA2.2 Definir avaliadores deve estar concluída.
- O artefato AR5 Avaliadores deve estar disponível.

#### **Pós-condições**

- Artefato AR6 Registro de treinamento deve estar concluído.

#### 5.2.5.2 Definições que não estão no ISO/IEC TR 15504

A exemplo do item 5.2.4.2, o qual mostra que o ISO/IEC TR 15504 possui definições apenas para o avaliador qualificado, nenhuma definição sobre o conteúdo que deve ser abordado no treinamento dos avaliadores é fornecida.

#### 5.2.5.3 Definições para o ExpPSEE

A execução desta tarefa no ExpPSEE corresponde a edição do documento AR6 Registro de treinamento pelo editor de textos.

#### **Pré-condições**

- A tarefa TA2.2 Definir avaliadores deve estar concluída.
- O documento AR5 Avaliadores deve estar disponível.

#### **Pós-condições**

- O artefato AR6 Registro de treinamento deve estar concluído.

#### **Artefatos:**

- AR5 Avaliadores.
- AR6 Registro de treinamento.

## **Papel**

- AG2 Avaliador qualificado.
- AG3 Avaliadores.

## **Ferramenta**

- Editor de textos.

## **Relacionamentos**

Ferramenta – Artefato:

Editor de textos com AR5 Avaliadores – ler.

Editor de textos com AR6 Registro de treinamento – ler, gravar, excluir.

Papel – Artefato:

AG2 Avaliador qualificado com AR6 Registro de treinamento – ler, gravar, excluir.

AG2 Avaliador qualificado com AR5 Avaliadores – ler.

AG3 Avaliadores com AR6 Registro de treinamento – ler.

AG3 Avaliadores com AR5 Avaliadores – ler.

Papel – Ferramenta:

AG2 Avaliador qualificado com Editor de textos – executar.

AG3 Avaliadores com Editor de textos – executar.

### **5.2.6 TA2.4 Planejar avaliação**

Esta tarefa consiste na elaboração de um cronograma detalhado de todo processo de avaliação e do *checklist* que será utilizado para coleta de evidências das instâncias de processos a serem avaliados. O cronograma, registrado no documento AR7 Planejamento da avaliação deve conter a agenda de reuniões com os responsáveis pelas instâncias de processo de *software* selecionadas no documento AR3 Instâncias de processos, devendo prever espaço para revisão das evidências coletadas e possíveis reuniões extras para esclarecimento de dúvidas.

O *checklist*, que é documentado no AR8 Checklist, é composto pelas práticas base e pelas práticas de gerenciamento, descritas em [ISO99-5].

### 5.2.6.1 Definições segundo ISO/IEC TR 15504

Para elaboração do cronograma, o qual será documentado no documento AR7 Planejamento da avaliação, o ISO/IEC TR 15504 especifica que esse planejamento deve definir:

- o cronograma detalhado de reuniões com os supervisores de cada unidade organizacional, juntamente com um cronograma detalhado para todo o processo de avaliação (4.4.P.9.b);
- um conjunto de indicadores de processo de *software* para ajudar no julgamento dos avaliadores (4.4.P.8).

Para o *checklist*, documentado no AR8 Checklist, o ISO/IEC TR 15504 define práticas segundo as quais os processos da avaliação devem ser comparados. Estas práticas dividem-se em dois grandes grupos: práticas base e práticas de gerenciamento, conforme a Figura 5.4, a qual mostra o relacionamento entre os indicadores da avaliação e a capacidade do processo. A dimensão de processo possui um conjunto de indicadores baseados nos produtos de trabalho (documentos) para os quais o anexo C da parte 5 do ISO/IEC TR 15504 possui um conjunto de características desejáveis. Já a dimensão de capacidade trata das práticas de gerenciamento que devem ser observadas durante o processo de avaliação. Apesar de não possuírem evidência física claramente definida, são observações durante o processo de avaliação, devem estar presentes através do comportamento ou práticas executadas pela organização, mesmo que não documentadas. Estes elementos constituem o *checklist* que deverá ser utilizado para a avaliação. Este *checklist* é elaborado nesta atividade no documento AR8 Checklist e respondido na tarefa TA2.6 Coletar informações.

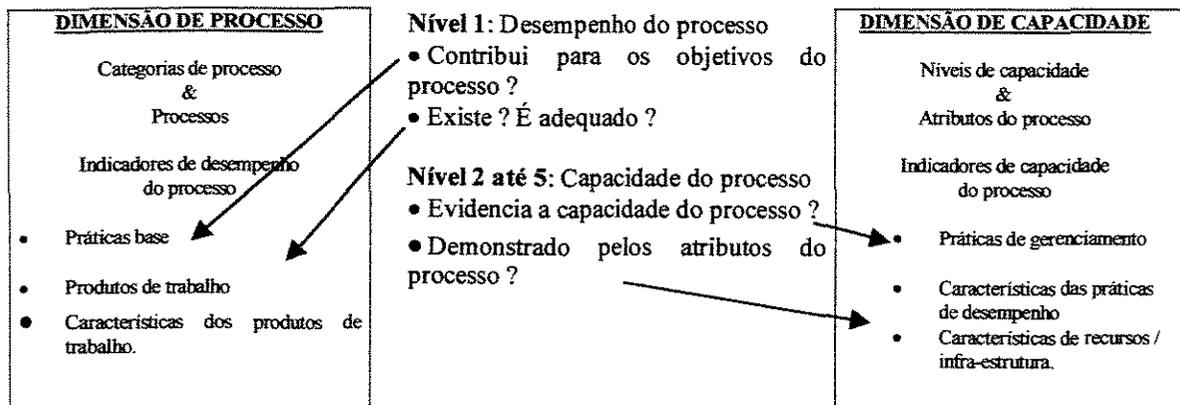


Figura 5.4: Relacionamento entre os indicadores da avaliação e a capacidade do processo [ISO99-5].

Este conjunto de práticas conforme descrito em [ISO99-5], mostrou-se bastante prático para execução do processo de avaliação, segundo o relato dos avaliadores durante a execução da fase 2 dos *trials* do projeto SPICE, descrito em [ISO98].

### **Pré-condições**

- A tarefa TA2.1 Definir Instâncias deve estar concluída.
- A tarefa TA2.2 Definir avaliadores deve estar concluída.
- A tarefa TA2.3 Treinar avaliadores (se necessária) deve estar concluída.
- O artefato AR4 Instâncias de processo deve estar disponível.

### **Pós-condições**

- O artefato AR7 Planejamento da avaliação deve estar concluído.
- O artefato AR8 Checklist deve estar concluído.

#### 5.2.6.2 Definições que não estão no ISO/IEC TR 15504

O ISO/IEC TR 15504 não define parâmetros para o planejamento da avaliação, como por exemplo a duração média de uma avaliação ou das entrevistas de acordo com as práticas que estão sendo avaliadas, ficando a cargo da experiência do avaliador qualificado este planejamento. Algumas estatísticas relacionadas a este planejamento podem ser encontradas em [ISO98], o qual documenta a fase 2 dos *trials* do projeto SPICE.

#### 5.2.6.3 Definições para o ExpPSEE

A execução desta tarefa no ExpPSEE corresponde a edição do documento AR7 Planejamento da avaliação e do documento AR8 Checklist pelo editor de texto.

### **Pré-condições**

- A tarefa TA2.1 Definir Instâncias deve estar concluída.
- A tarefa TA2.2 Definir avaliadores deve estar concluída.
- A tarefa TA2.3 Treinar avaliadores (se necessária) deve estar concluída.
- O artefato AR4 Instâncias de processo deve estar disponível.

### **Pós-condições**

- O artefato AR7 Planejamento da avaliação deve estar concluído.
- O artefato AR8 Checklist deve estar concluído.

### **Artefatos**

- AR4 Instâncias de processos.

- AR7 Planejamento da avaliação.
- AR8 Checklist.

### **Papel**

- AG2 Avaliador qualificado.
- AG3 Avaliadores.

### **Ferramenta**

- Editor de textos.

### **Relacionamentos**

#### Ferramenta – Artefato:

- Editor de textos com AR4 Instâncias de processos – ler.
- Editor de textos com AR7 Planejamento da avaliação – ler, gravar, excluir.
- Editor com AR8 Checklist – ler, gravar, excluir.

#### Papel – Artefato:

- AG2 Avaliador qualificado com AR4 Instância de processos – ler, gravar.
- AG2 Avaliador qualificado com AR7 Planejamento da avaliação – ler, gravar, excluir.
- AG2 Avaliador qualificado com AR8 Checklist – ler, gravar, excluir.
- AG3 Avaliadores com AR4 Instância de processos – ler.
- AG3 Avaliadores com AR7 Planejamento da avaliação – ler, gravar.
- AG3 Avaliadores com AR8 Checklist – ler, gravar.

#### Papel – Ferramenta:

- AG2 Avaliador qualificado com Editor de textos – executar;
- AG3 Avaliadores com Editor de textos – executar;

## **5.2.7 TA2.5 Coletar documentos**

Esta tarefa consiste na coleta de evidências através de documentos gerados durante a execução do processo de *software*. Dependendo do grau de controle de configuração disponível na organização, esta tarefa pode ser simples ou muito complexa. A lista de documentos, documento AR9 Lista de documentos, deve ser elaborada para as instâncias de processos definidas no documento AR4 Instâncias de processos, seguindo o planejamento da avaliação descrito no documento AR7 Planejamento da avaliação.

### 5.2.7.1 Definições segundo o ISO/IEC TR 15504

O ISO/IEC TR 15504 apenas indica que os documentos gerados pelas instâncias de processo de *software* selecionados devem ser reunidos para serem avaliados (4.4.P.5, 4.4.P.9.a).

#### **Pré-condições**

- A tarefa TA2.4 Planejar avaliação deve estar concluída.
- O artefato AR7 Planejamento da avaliação deve estar disponível.
- O artefato AR4 Instâncias de processos deve estar disponível.

#### **Pós-condições**

- O artefato AR9 Lista de documentos deve estar concluído.

### 5.2.7.2 Definições que não estão no ISO/IEC TR 15504

O ISO/IEC TR 15504 traz um conjunto de documentos padrões para que os mesmos sejam utilizados como indicadores na avaliação, porém não referencia a coleta de documentos como sendo um passo importante do processo de avaliação. Como nada se sabe sobre a gestão de configuração dentro da organização que está sendo avaliada, é importante reunir estes documentos pois é com base nestas evidências que a avaliação será conduzida e muitas medições (práticas base) serão realizadas.

### 5.2.7.3 Definições para o ExpPSEE

A execução desta tarefa no ExpPSEE corresponde a edição do documento AR9 Lista de documentos pelo editor de textos.

#### **Pré-condições**

- A tarefa TA2.4 Planejar avaliação deve estar concluída.
- O artefato AR4 Instâncias de processo deve estar disponível.
- O artefato AR7 Planejamento da avaliação deve estar disponível.

#### **Pós-condições**

- O artefato AR9 Lista de documentos deve estar concluído.

## Artefatos

- AR4 Instâncias de processos.
- AR7 Planejamento da avaliação.
- AR9 Lista de documentos.

## Papel

- AG2 Avaliador qualificado.
- AG3 Avaliadores.

## Ferramenta

- Editor de textos.

## Relacionamentos

### Ferramenta – Artefato:

Editor de textos com AR9 Lista de documentos – ler, gravar, excluir;  
excluir;  
Editor de textos com AR7 Planejamento da avaliação – ler, gravar,  
excluir;  
Editor de textos com AR4 Instâncias de processos – ler.

### Papel – Artefato:

AG2 Avaliador qualificado com AR9 Lista de documentos – ler, gravar,  
excluir.  
AG2 Avaliador qualificado com AR7 Planejamento da avaliação – ler.  
AG2 Avaliador qualificado com AR4 Instâncias de processos – ler.  
AG3 Avaliadores com AR9 Lista de documentos – ler, gravar.  
AG3 Avaliadores com AR7 Planejamento da avaliação – ler.  
AG3 Avaliadores com AR4 Instâncias de processos – ler.

### Papel – Ferramenta:

AG2 Avaliador qualificado com Editor de textos – executar.  
AG3 Avaliadores com Editor de textos – executar.

## 5.2.8 TA2.6 Coletar informações

De acordo com o planejamento da avaliação, o documento AR7 Planejamento da avaliação, de posse dos documentos coletados na tarefa TA2.5 Coletar documentos, cuja lista de documentos está no documento AR9 Lista de documentos, responder o *checklist* preenchendo o documento AR8 Checklist. Esse documento será preenchido em duas etapas, uma comparando-se os produtos de trabalho com os indicadores contidos no anexo C da parte 5 do ISO/IEC TR 15504 e a outra coletando-se as evidências e esclarecendo-se dúvidas através de entrevistas com os responsáveis e/ou participantes das instâncias de

processos selecionadas. Nestas entrevistas e visitas as unidades organizacionais, as práticas de gerenciamento devem ser observadas e o AR8 Checklist deve ser respondido.

#### 5.2.8.1 Definições segundo o ISO/IEC TR 15504

O ISO/IEC TR 15504 define os seguintes requisitos para execução desta tarefa:

- registrar evidências sobre o processo que está sendo avaliado, através da documentação apresentada (4.4.P.5, Nota 2);
- entrevistar os Supervisores levantando informações, confirmando evidências e descobrindo novas, sobre as instâncias dos processos selecionadas (4.4.P.5);
- registrar as evidências coletadas nas entrevistas e na documentação (4.4.P.5);
- entrevistar os supervisores das unidades organizacionais com as quais as instâncias de processos de *software* selecionadas estejam envolvidas (4.4.P.5, 4.4.P.9.b).
- justificar cada medida coletada de acordo com as evidências apontadas (4.4.P.9.c).

#### Pré-condições

- A tarefa TA2.5 Coletar documentos deve estar concluída.
- O artefato AR8 Checklist deve estar disponível.
- O artefato AR9 Lista de documentos deve estar disponível.
- O artefato AR4 Instâncias de processos deve estar disponível.

#### Pós-condições

- O artefato AR8 Checklist deve estar respondido e todas as dúvidas devem ter sido esclarecidas junto as unidades organizacionais envolvidas com as instâncias de processos selecionadas no documento AR4 Instâncias de processos.

#### 5.2.8.2 Definições que não estão no ISO/IEC TR 15504

O ISO/IEC TR 15504 possui um conjunto completo de requisitos para coleta de dados, ajudando na montagem do *checklist*, conforme mostra a tarefa TA2.4 Planejar avaliação, através das práticas básicas, de gerenciamento e de um conjunto de indicadores. A parte 5 do ISO/IEC TR 15504 foi considerada durante os *trials* [ISO98] do projeto SPICE como o guia que mais ajudou na execução desta tarefa.

#### 5.2.8.3 Definições para o ExpPSEE

A execução desta tarefa no ExpPSEE corresponde a edição do documento AR8 Checklist pelo editor de textos. Este documento já existe pois foi gerado na tarefa TA2.4 Planejar avaliação, deve ser editado e respondido.

## **Pré-condições**

- A tarefa TA2.5 Coletar documentos deve estar concluída.
- O artefato AR8 Checklist deve estar disponível.
- O artefato AR9 Lista de documentos deve estar disponível.
- O artefato AR4 Instâncias de processos deve estar disponível.

## **Pós-condições**

- O artefato AR8 Checklist deve estar respondido.

## **Artefatos**

- AR4 Instâncias de processos.
- AR8 Checklist.
- AR9 Lista de documentos.

## **Papel**

- AG2 Avaliador qualificado.
- AG3 Avaliadores.

## **Ferramenta**

- Editor de textos.

## **Relacionamentos**

### Ferramenta – Artefato:

Editor de textos com AR8 Checklist – ler, gravar.

Editor de textos com AR9 Lista de documentos – ler.

Editor de textos com AR4 Instâncias de processos – ler.

### Papel – Artefato:

AG2 Avaliador qualificado com AR4 Instâncias de processos – ler.

AG2 Avaliador qualificado com AR9 Lista de documentos – ler.

AG2 Avaliador qualificado com AR8 Checklist – ler, gravar.

AG3 Avaliadores com AR4 Instâncias de processos – ler.

AG3 Avaliadores com AR9 Lista de documentos – ler.

AG3 Avaliadores com AR8 Checklist – ler, gravar.

### Papel – Ferramenta:

AG2 Avaliador Qualificado com Editor de textos – executar;

AG3 Avaliadores com Editor de textos – executar;

## 5.2.9 TA 2.7 Verificar informações

Esta tarefa deve ser executada com objetivo de verificar a consistência da avaliação executada, ou seja, todas as respostas dadas no documento AR8 Checklist devem ter evidências que as comprovem (justificativas). Deve ser verificado se existem ainda questões (perguntas) em aberto que ainda não foram esclarecidas e se os avaliadores estão seguros das respostas anotadas no *checklist*. Outro item a ser verificado é se todas as instâncias de processos foram avaliadas conforme definido no documento AR4 Instâncias de processo.

### 5.2.9.1 Definições segundo o ISO/IEC TR 15504

O ISO/IEC TR 15504 define uma série de requisitos para que a avaliação seja consistente e repetível. A estrutura do *checklist* definido segundo as práticas base e de gerenciamento, conforme mostra a tarefa TA2.4 Planejar avaliação mostra isto. O cruzamento das evidências encontradas nos documentos coletados (práticas base) com as anotações das entrevistas pode facilmente revelar inconsistências na avaliação propriamente dita, ou na organização. Os requisitos definidos pelo ISO/IEC TR 15504 que devem ser checados no documento AR8 Checklist são:

- checar as evidências coletadas na tarefa anterior, procurando inconsistências entre o que foi documentado e o que foi afirmado nas entrevistas (4.4.P.9);
- confirmar evidências encontradas na documentação (4.4.P.9);
- confirmar práticas que estão sendo executadas, mas não possuem evidências que as comprovem (4.4.P.9);
- marcar novas entrevistas com os supervisores para esclarecer dúvidas, Se necessário (4.4.P.9).

### Artefatos

- AR4 Instâncias de processo.
- AR8 Checklist.
- AR9 Lista de documentos.

### Pré-condições

- A tarefa TA2.6 Coletar informações deve estar concluída.
- O artefato AR4 Instâncias de processo deve estar disponível.
- O artefato AR8 Checklist o qual registra todas as evidências levantadas de acordo com os documentos analisados (práticas base) e as evidências levantadas nas entrevistas (práticas de gerenciamento), deve estar disponível.
- O artefato AR9 Lista de documentos deve estar disponível.

### Pós-condições

- AR8 Checklist verificado e aprovado.

### 5.2.9.2 Definições que não estão no ISO/IEC TR 15504

O ISO/IEC TR 15504 não traz nenhuma referência cruzada entre os atributos de processo ou entre as práticas base e de gerenciamento, apesar de saber-se que eles não são independentes. Uma referência cruzada ajudaria em muito a validação das informações coletadas, diminuindo as inconsistências e melhorando a repetibilidade da avaliação.

### 5.2.9.3 Definições para o ExpPSEE

A execução desta tarefa no ExpPSEE corresponde à edição do documento AR8 Checklist pelo editor de textos.

#### **Pré-condições**

- A tarefa TA2.6 Coletar informações deve estar concluída.
- O artefato AR4 Instâncias de processo deve estar disponível.
- O artefato AR8 Checklist deve estar disponível.
- O artefato AR9 Lista de documentos deve estar disponível.

#### **Pós-condições**

- O artefato AR8 Checklist deve ter sido verificado e aprovado.

#### **Artefatos**

- AR4 Instâncias de processos.
- AR8 Checklist.
- AR9 Lista de documentos.

#### **Papel**

- AG2 Avaliador qualificado.

#### **Ferramenta**

- Editor de textos.

#### **Relacionamentos**

Ferramenta – Artefato:

Editor de textos com AR8 Checklist – ler, gravar;  
Editor de textos com AR4 Instâncias de processos – ler;  
Editor de textos com AR9 Lista de documentos – ler;

Papel – Artefato:

- AG2 Avaliador qualificado com AR8 Checklist – ler, gravar.
- AG2 Avaliador com AR4 Instância de processos – ler.
- AG2 Avaliador com AR9 Lista de documentos – ler.

Papel – Ferramenta:

- AR2 Avaliador qualificado com Editor de textos – executar;

### **5.2.10 TA2.8 Determinar medidas**

Nesta tarefa, é definido para cada instância de processo de *software* avaliada o perfil conforme definido no documento AR3 Entradas da avaliação para cada instância de processo definida no documento AR4 Instância de processo e avaliado no documento AR8 Checklist. A forma de determinação de tais medidas está definida em [ISO99-2].

#### **5.2.10.1 Definições segundo o ISO/IEC TR 15504**

O ISO/IEC TR 15504 define em sua parte 2 que o perfil de um processo de *software* deve ser definido de acordo com as evidências registradas e deve-se estabelecer o perfil de cada instância de processo de *software* conforme modelo ali apresentado (**4.4.P.6**);

#### **Pré-condições**

- A tarefa TA2.7 Verificar informações deve estar concluída.
- O artefato AR8 Checklist deve estar disponível.
- O artefato AR4 Instâncias de processos deve estar disponível.

#### **Pós-condições**

- Artefato AR10 Medidas das instâncias de processo deve estar concluído.

#### **5.2.10.2 Definições que não estão no ISO/IEC TR 15504**

O ISO/IEC TR 15504 separa práticas bases das práticas de gerenciamento, apesar de bem explicar como representá-las, deixa-as em separado quando demonstra o perfil de cada instância de processo de *software*.

#### **5.2.10.3 Definições para o ExpSSEE**

A execução desta tarefa no ExpSSEE corresponde à edição do documento AR8 Checklist pelo editor de textos.

### **Pré-condições**

- A tarefa TA2.7 Verificar informações deve estar concluída.
- O artefato AR8 Checklist deve estar disponível.
- O artefato AR4 Instâncias de processo deve estar disponível.

### **Pós-condições**

- O artefato AR10 Medidas das instâncias de processo deve estar concluído.

### **Artefatos**

- AR4 Instâncias de processos.
- AR8 Checklist.
- AR10 Medidas das instâncias.

### **Papel**

- AG2 Avaliador qualificado.

### **Ferramenta**

- Editor de textos.

### **Relacionamentos**

Ferramenta – Artefato:

Editor de textos com AR10 Medidas das instâncias de processo – ler, gravar, excluir;

Editor de textos com AR4 Instâncias dos processos – ler;

Editor de textos com AR8 Checklist – ler;

Papel – Artefato:

AG2 Avaliador qualificado com AR10 Medidas das instâncias – ler, gravar, excluir.

AG2 Avaliador qualificado com AR4 Instâncias dos processos – ler.

AG2 Avaliador qualificado com AR8 Checklist – ler.

Papel – Ferramenta:

AG2 Avaliador qualificado com Editor de textos – executar.

#### **5.2.11 TA 2.9 Consolidar medidas**

Esta tarefa consiste na determinação do perfil do processo de *software*, constante no documento AR11 Medidas do processo de *software*, a partir do documento AR10

Medidas das instâncias de processo. Este é o documento que consolida todo o processo de avaliação.

#### 5.2.11.1 Definições segundo o ISO/IEC TR 15504:

A partir das medições contidas no documento AR9 Medidas das instâncias, elaborar o perfil da avaliação, reunindo todas as medições estabelecidas para cada perfil de processo e unificá-las resultando no perfil da avaliação, conforme modelo definido na parte 2 do ISO/IEC TR 15504 (**4.4.P.7**);

##### **Pré-condições**

- A tarefa TA2.8 Determinar medidas deve estar concluída.
- O artefato AR10 Medidas das instâncias de processo deve estar disponível.

##### **Pós-condições**

- Artefato AR11 Medidas do processo de *software* deve estar concluído.

#### 5.2.11.2 Definições que não estão no ISO/IEC TR 15504

Esta tarefa está bem descrita na parte 2 do ISO/IEC TR 15504, mostrando claramente como se representam as várias instâncias de processos, os gráficos que podem ser gerados, auxiliando na apresentação dos resultados. Um exemplo destes resultados pode ser encontrado na execução desta tarefa no estudo de caso do Capítulo 6.

#### 5.2.11.3 Definições para o ExpPSEE

A execução desta tarefa no ExpPSEE corresponde a edição do documento AR11 Medidas do processo de *software* pelo editor de textos.

##### **Pré-condições**

- A tarefa TA2.8 Determinar medidas deve estar concluída.
- O artefato AR10 Medidas das instâncias de processo deve estar disponível.

##### **Pós-condições**

- Artefato AR11 Medidas do processo de *software* deve estar concluído.

##### **Artefatos**

- AR10 Medidas das instâncias de processos.

- AR11 Medidas do processo de *software*.

### **Papel**

- AG2 Avaliador qualificado.

### **Ferramenta**

- Editor de textos.

### **Relacionamentos**

Ferramenta – Artefato:

Editor de textos com AR11 Medidas do processo de *software* – ler, gravar.

Editor de textos com AR10 Medidas das instâncias de processo – ler.

Papel – Artefato:

AG2 Avaliador qualificado com AR11 Medidas do processo de *software* – ler, gravar.

AG2 Avaliador qualificado com AR10 Medidas das instâncias de processo – ler.

Papel – Ferramenta:

AG2 Avaliador qualificado com Editor de textos – executar.

## **5.2.12 TA2.10 Validar medidas**

Esta tarefa deve verificar se o resultado final da avaliação possui alguma inconsistência por erros gerados durante a tabulação dos dados e a adequação do resultado final ao perfil do processo de *software* descrito no documento AR3 Entradas da avaliação.

### **5.2.12.1 Definições segundo o ISO/IEC TR 15504**

Esta tarefa deve reunir todas as medições estabelecidas para cada perfil de processo de *software* documentadas no artefato AR10 Medidas das instâncias de processos e verificar se o perfil da avaliação está de acordo com o modelo definido em [ISO99-2]. Para isto, deve-se verificar se o perfil da avaliação atende o que foi determinado no objetivo da avaliação descrito no documento AR2 Descrição da avaliação (4.3.P.2, 4.4.P.1).

### **Artefatos**

- AR11 Medidas do processo de *software*.
- AR10 Medidas das instâncias de processo.

### **Pré-condições**

- A tarefa TA2.9 Consolidar medidas deve estar concluída.
- O artefato AR11 Medidas do processo de *software* deve estar disponível.
- O artefato AR10 Medidas das instâncias de processo deve estar disponível.

### **Pós-condições**

- Artefato AR11 Medidas do processo de *software* validado e aprovado.

#### 5.2.12.2 Definições que não estão no ISO/IEC TR 15504

Novamente uma referência cruzada entre os perfis obtidos para cada atributo daria mais confiabilidade aos resultados da avaliação.

#### 5.2.12.3 Definições para o ExpSSEE

A execução desta tarefa no ExpSSEE corresponde a edição do documento AR11 Medidas do processo de *software* pelo editor de textos. Este documento já existe e foi criado na tarefa TA2.9 Consolidar medidas.

### **Pré-condições:**

- A tarefa TA2.9 Consolidar medidas deve estar concluída.
- O artefato AR11 Medidas do processo de *software* deve estar disponível.
- O artefato AR10 Medidas das instâncias de processo deve estar disponível.

### **Pós-condições:**

- Artefato AR11 Medidas do processo de *software* deve estar concluído.

### **Artefatos:**

- AR11 Medidas do processo de *software*.
- AR10 Medidas das instâncias de processo.

### **Papel:**

- AG2 Avaliador qualificado.

### **Ferramenta:**

- Editor de textos.

### **Relacionamentos:**

Ferramenta – Artefato:

Editor de textos com AR11 Medidas do processo de *software* – ler, gravar.

Editor de textos com AR3 Entradas da avaliação – ler.

Papel – Artefato:

AG2 Avaliador qualificado com AR11 Medidas do processo de *software* – ler, gravar.

AG2 Avaliador qualificado com AR3 Entradas da avaliação – ler.

Papel – Ferramenta:

AG2 Avaliador qualificado com Editor de textos – executar.

### 5.2.13 TA3 Apresentar resultados

Esta tarefa está fora do processo de avaliação e tem como objetivo apresentar os resultados da avaliação conforme o perfil da avaliação e do acordo de confidencialidade documentados no artefato AR3 Entradas da avaliação. Esta apresentação deve estar registrada em um documento denominado AR12 Apresentação.

#### 5.2.13.1 Definições segundo ISO/IEC TR 15504

O ISO/IEC TR 15504 define claramente como os resultados devem ser apresentados. Os critérios estão listados abaixo:

- apresentar o perfil da avaliação de acordo com a determinação do objetivo da avaliação descrito no documento AR2 Descrição inicial e do acordo de confidencialidade contido no documento AR3 Entradas da avaliação, devendo conter no mínimo (4.5):
  - a data da avaliação;
  - as entradas da avaliação;
  - as instâncias de processo de *software* selecionadas e a justificativa da seleção;
  - a abordagem utilizada na avaliação, incluindo a identificação do modelo de avaliação, método e qualquer instrumento de suporte ou ferramentas para suportar reclamações de adequação.
  - os nomes dos membros da equipe que conduziram a avaliação, incluindo o avaliador qualificado responsável pela avaliação e a equipe de avaliadores.
  - o conjunto de perfis de processo de *software* resultantes da avaliação.
  - a localização de registros que suportam evidências objetivas, as medidas dos atributos de processo nos perfis de processo.
  - informações adicionais coletadas durante a avaliação que foram identificadas nas entradas da avaliação.

#### Pré-condições

- A tarefa TA2.10 Validar medidas deve estar concluída.

- O artefato AR11 Medidas do processo de *software* deve estar disponível.
- O artefato AR3 Entradas da avaliação deve estar disponível.

### **Pós-condições**

- Deve-se realizar a apresentação dos resultados de acordo com o acordado no documento AR3 Entradas da avaliação atendendo aos requisitos listados acima.
- O artefato AR12 Apresentação deve estar finalizado.

### 5.2.13.2 Definições que não estão no ISO/IEC TR 15504

O ISO/IEC TR 15504 define com clareza o que deve ser apresentado, bem como regulamenta através do acordo de confidencialidade definido no documento AR3 Entradas da avaliação.

### 5.2.13.3 Definições para o ExpSEE:

A execução desta tarefa no ExpSEE corresponde a edição do documento AR11 Medidas do processo de *software* pelo editor de textos do sistema operacional. Este documento já existe e foi criado na tarefa TA2.9 Consolidar medidas.

### Pré-condições:

- A tarefa TA2.10 Validar medidas deve estar concluída.
- O artefato AR11 Medidas do processo de *software* deve estar disponível.
- O artefato AR3 Entradas da avaliação deve estar disponível.

### Pós-condições:

- O artefato AR12 Apresentação deve estar finalizado.

### Artefatos:

- AR11 Medidas do processo de *software*.
- AR3 Entradas da avaliação.
- AR12 Apresentação.

### Papel:

- AG1 Patrocinador.
- AG2 Avaliador qualificado.
- AG3 Avaliadores.

### Ferramenta:

- Editor de textos.

#### Relacionamentos:

##### Ferramenta – Artefato:

- Editor de textos com AR11 Medidas do processo de *software* – ler.
- Editor de textos com AR3 Entradas da avaliação – ler.
- Editor de textos com AR12 Entradas da avaliação – ler, gravar.

##### Papel – Artefato:

- AG1 Patrocinador com AR12 Apresentação– ler, gravar.
- AG1 Patrocinador com AR11 Medidas do processo de *software* – ler.
- AG1 Patrocinador com AR3 Entradas da avaliação – ler.
- AG2 Avaliador qualificado com AR11 Medidas do processo de *software* – ler.
- AG2 Avaliador qualificado com AR13 Entradas da avaliação – ler.
- AG2 Avaliador qualificado com AR12 Apresentação– ler, gravar.
- AG3 Avaliadores com AR12 Apresentação– ler.
- AG3 Avaliadores com AR11 Medidas do processo de *software* – ler.
- AG3 Avaliadores com AR3 Entradas da avaliação – ler.

##### Papel – Ferramenta:

- AG1 Patrocinador com Editor de textos – executar;
- AG2 Avaliador qualificado com Editor de textos – executar;
- AG3 Avaliadores com Editor de textos – executar;

### 5.3 Considerações finais

Neste capítulo descrevemos uma metodologia de avaliação de processos de *software* utilizando a linguagem do ambiente PRONET. A metodologia foi descrita seguindo-se três enfoques principais:

- quais as definições segundo a norma ISO/IEC TR 15504,
- quais as definições necessárias para que ela execute em uma ambiente automatizado de processo de *software*, o ExpPSEE, e
- os limites nas definições do ISO/IEC TR 15504, ou seja, o que é necessário acrescentar às definições da norma para que a metodologia possa ser executada.

No capítulo seguinte será apresentado um estudo de caso para a metodologia apresentada neste capítulo com sua implementação (programação) no ExpPSEE. Este estudo de caso, juntamente com as definições deste capítulo nos permitirão formular as conclusões e trabalhos futuros apresentados no Capítulo 7.

## Capítulo 6

### Estudo de caso

Este estudo de caso visa a validação da metodologia proposta e é composto de um conjunto de documentos e atividades, de acordo com a metodologia MAPS-15504 apresentada no Capítulo 5. Este conjunto de documentos é gerado seguindo-se cada fase da metodologia, a partir do contexto de uma empresa fictícia descrita na Seção 6.1 deste capítulo. Esta empresa fictícia foi baseada em um perfil médio de empresas obtido a partir do relatório dos *trials* do ISO/IEC TR 15504 [MCT97] e do relatório sobre o estado da qualidade de *software* no Brasil editado a cada dois anos pelo PBQP<sup>23</sup> descrito em [ISO98].

#### 6.1 Apresentação do Contexto

A seguir apresentaremos a descrição da divisão de informática da Empresa X Ltda. a qual será utilizada para o estudo de caso da metodologia de avaliação de processo de *software* baseada no projeto ISO/IEC TR 15504.

##### 6.1.1 A Empresa X Ltda

Este estudo de caso refere-se a Empresa X Ltda que desenvolve e mantém produtos de *software* para aplicações comerciais com tecnologia de ponta e fatura anualmente R\$ 900.000,00 (Novecentos mil reais). A unidade organizacional responsável pela área de informática da empresa é composta por 22 pessoas e possui uma estrutura organizacional conforme o organograma da Figura 6.1. Está fora do escopo deste estudo de caso, outras unidades organizacionais da Empresa X Ltda.



Figura 6.1: Organograma da divisão de informática da empresa X.

A seguir é apresentada uma descrição sumária dos departamentos da divisão de informática da Empresa X Ltda.

Diretoria de informática: órgão composto por um (1) diretor responsável pela unidade organizacional que coordena a área de informática da empresa.

Garantia de qualidade: órgão composto por três (3) pessoas, as quais executam testes, verificações e validações, controlando toda documentação gerada para os produtos de *software*.

Prospecção tecnológica: órgão composto por três (3) pessoas, as quais executam atividades de pesquisa e testes de novas tecnologias, repassando-as aos outros órgãos da divisão de informática em forma de treinamento.

Desenvolvimento: órgão composto por oito (8) pessoas, das quais duas são analistas e projetistas seniores. Seis trabalham com atividades ligadas diretamente ao desenvolvimento de *software* (análise e programação).

Suporte e treinamento: órgão composto por três (4) pessoas, as quais dão suporte e treinamento aos clientes, usuários dos produtos da empresa.

Redes: órgão composto por três (3) pessoas, as quais dão suporte aos usuários da rede da empresa.

### **6.1.2 Tecnologias utilizadas**

As tecnologias utilizadas para desenvolvimento de *software* na organização estão baseadas na abordagem orientada a objetos, utilizando UML (*Unified Modeling Language*) [BOOC99] como linguagem para documentação da análise e projeto dos sistemas. O Ciclo de vida adotado é o de prototipação evolutiva [PRES95].

---

<sup>23</sup>PBQP – Programa Brasileiro de Qualidade e Produtividade – vinculado ao MCT – Ministério da Ciência e Tecnologia.

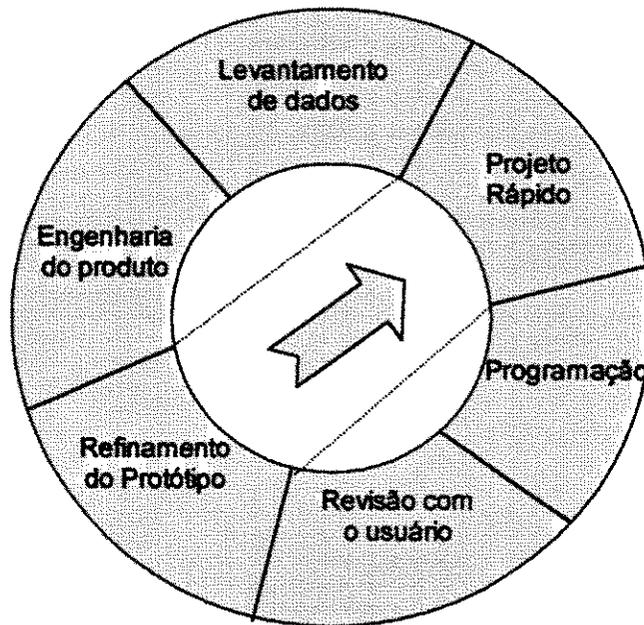


Figura 6.2: Ciclo de vida de prototipação [PRES95].

Os seguintes documentos são gerados ao longo do processo de desenvolvimento conforme representado na Figura 6.2.

Levantamento de dados: nesta fase são realizadas reuniões com o usuário para que se conheçam todos os requisitos do *software* a ser desenvolvido. Estes requisitos primeiramente são documentados em atas de reuniões. Estas atas são aprovadas e resumidas em um documento que contém uma descrição do sistema proposto. Após esta descrição são gerados os diagramas de Caso de uso, Classe e Seqüência. O resultado final, após documentação do levantamento de dados nos diagramas citados, é um contrato e um cronograma para o desenvolvimento do produto de *software*.

Projeto rápido: nesta fase são realizados tantos projetos quanto necessários até a aprovação do sistema pelo usuário. Com base nos Diagramas de caso de uso, seqüência e classes são elaborados os primeiros protótipos, os quais mostram as funcionalidades do sistema através do modelo navegacional obtido pelo projeto da interface. O projeto rápido segue refinando o protótipo e gerando novas versões da documentação até que uma versão do sistema seja aceita pelo usuário. Os requisitos para o refinamento são obtidos através dos documentos Refinamento do protótipo e Resultados de testes. Quando necessário, de acordo com o estágio do protótipo, com base no diagrama de classes far-se-á o Projeto de Banco de Dados.

Programação: nesta fase é realizada a implementação (codificação) do projeto desenvolvido na fase anterior. Todos os documentos serão entradas para a programação. O resultado desta fase são os módulos programados ou código fonte dos programas.

Revisão com o usuário: a equipe de garantia da qualidade deve organizar as revisões juntamente com o usuário e com o analista sênior responsável pelo projeto. O protótipo deverá ser testado de acordo com os requisitos documentados através do Contrato, Documento de alterações, Resultados de testes (outras revisões com o usuário). Podem ser utilizados, caso necessário, os diagramas de Use Case, Seqüência e Classes, e os projetos de Banco de Dados e Interface para validar a implementação. Esta revisão deve ser documentada através de um *checklist* para verificação e validação.

Refinamento do protótipo: uma vez documentadas as impressões do usuário sobre o protótipo, deve ser gerado um documento contendo o Resultado dos testes. Este resultado será utilizado para a elaboração do novo projeto o qual trará um protótipo mais refinado e assim sucessivamente até a aceitação do protótipo pelo usuário.

Engenharia do produto: nesta fase, o produto de *software* é empacotado, sendo gerados os discos de instalação, elaborado o manual do usuário, folhetos e quaisquer outros documentos que acompanhem o pacote que será enviado ao cliente. Na entrega do pacote de *software*, deve ser documentado o aceite do usuário através do documento de aceitação pelo cliente.

## **6.2 Aplicação da Metodologia MAPS-15504**

A seguir é apresentada a descrição da aplicação de cada tarefa da metodologia descrita no Capítulo 5, no contexto descrito na Seção 6.1, bem como um exemplo de documentação gerado pela aplicação da metodologia durante a avaliação. As tarefas executadas foram: TA1.1 Descrever avaliação, TA1.2 Definir entradas da avaliação, TA2.1 Definir instâncias, TA2.2 Definir avaliadores, TA2.3 Treinar avaliadores, TA2.4 Planejar avaliação, TA2.5 Coletar documentos, TA2.6 Coletar informações, TA2.7 Validar informações, TA2.8 Determinar medidas das instâncias de processo, TA2.9 Determinar medidas do processo, TA2.10 Validar medidas e TA3 Apresentar resultados.

### **6.2.1 TA1.1 Descrever avaliação**

A descrição da avaliação por parte da organização é feita através da emissão de um documento oficial da empresa dando autorização para início das tarefas relativas a avaliação do processo de *software*. Nesta tarefa é gerada a descrição inicial conforme mostra o documento AR2 Descrição (Figura 6.3), gerado segundo as definições descritas na Seção 5.2.1 do Capítulo 5.

**EMPRESA X**

AR2 Descrição inicial

Maringá 14 de janeiro de 2000.

## DESCRIÇÃO INICIAL

**Organização:** Empresa X Ltda.

**Patrocinador:** Patrocinador da Cunha

**Propósito:** O propósito desta avaliação é ter um perfil do setor de desenvolvimento de software da divisão de informática, através do qual seja possível conhecer o perfil do setor e prever potenciais riscos dos processos.

---

A diretoria

Figura 6.3: Documento AR2 Descrição inicial.

### 6.2.2 TA1.2 Definir entradas da avaliação

Esta tarefa consiste na estruturação do processo de avaliação, conforme mostram os documentos AR2 Descrição inicial e AR3 Entradas da avaliação o qual contém o acordo de confidencialidade, descritos nas figuras 6.4a e 6.4b respectivamente. A execução da mesma é de responsabilidade do avaliador qualificado e suas definições guiarão todo o processo de avaliação.

**EMPRESA X**

AR3 Entradas da avaliação

Maringá 14 de janeiro de 2000.

## ENTRADAS DA AVALIAÇÃO

**Organização:** Empresa X Ltda.

**Patrocinador:** Patrocinador da Cunha.

**Avaliador Qualificado:** Avaliador Qualificado da Silva

**Propósito:** O propósito desta avaliação é ter um perfil do setor de desenvolvimento de software da divisão de informática, através do qual seja possível conhecer o perfil do setor e prever potenciais riscos dos processos.

**Avaliadores:** Analista Sênior dos Santos

Analista Programador da Rosa

Analista Programador de Oliveira

**Processos de Software:**

<input checked="" type="checkbox"/> Cliente-Fornecedor	CUS 2	Nível 3
<input checked="" type="checkbox"/> Engenharia	ENG 2, ENG 3	Nível 3
<input checked="" type="checkbox"/> Suporte	SUP 2	Nível 2
<input checked="" type="checkbox"/> Gerenciamento	MAN 1, MAN 2	Nível 3
<input checked="" type="checkbox"/> Organizacional	ORG 2	Nível 4

**Metodologia:** Será utilizada a metodologia MAPS-15504.

**Modelo Compatível:** Como modelo compatível será utilizada a parte 5 da ISO/IEC TR 15504.

**Data de Início:** 01/02/2000.

**Data de Término:** 31/05/2000.

**Previsão de horas:** 450 horas.

---

Patrocinador

---

Avaliador Qualificado

Figura 6.4a: Documento AR3 Entradas da avaliação.

## ACORDO DE CONFIDENCIALIDADE

**Organização:** Empresa X Ltda.

**Patrocinador:** Patrocinador da Cunha

**Avaliador Qualificado:** Avaliador Qualificado da Silva

**Parágrafo 1:** Do objeto deste acordo de confidencialidade :

Este acordo de confidencialidade está restrito a avaliação do processo de software conforme descrita no documento DTA2.2 Descrição do processo de avaliação.

**Parágrafo 2:** Dos direitos sobre os resultados:

Fica estabelecido que todos os direitos sobre os resultados da avaliação são de propriedade da Empresa X Ltda.

**Parágrafo 3:** Dos direitos sobre o uso dos resultados:

Fica estabelecido que a Empresa X Ltda possui todos os direitos sobre a utilização dos resultados da análise, incluindo publicação de artigos e estatísticas, não podendo os mesmos serem utilizados sem prévia autorização por escrito.

**Parágrafo 4:** Do acesso aos dados e resultados:

Fica estabelecido que somente as pessoas envolvidas no processo de avaliação terão acesso aos dados e somente a diretoria da Empresa X Ltda, juntamente com a equipe de avaliação terá acesso aos resultados, devendo esta decidir o que deverá ser repassados as demais áreas da empresa, porém sem prejuízo ou identificação da fonte das informações preservando a identidade dos colaboradores.

**Parágrafo 5:** Demais condições:

Fica estabelecido que quaisquer questões não regulamentadas neste acordo de confidencialidade devem ser tratadas entre a diretoria da Empresa X Ltda e o Avaliador Qualificado, as resoluções documentadas e adendadas a este documento.

---

Patrocinador

---

Avaliador Qualificado

Figura 6.4b: Documento AR3 Entradas da avaliação.

### **6.2.3 TA2.1 Definir instâncias**

Nesta tarefa, as instâncias de processos de *software* (projetos) devem ser selecionadas e a partir da sua avaliação definirão o perfil desejado de acordo com o propósito da avaliação, conforme a Figura 6.5 que mostra o documento AR4 Instâncias de processos de *software* com a instância de projeto Sistema de controle de pedidos pela Internet para a Empresa Z Ltda.

Maringá 14 de janeiro de 2000.

## INSTÂNCIAS DE PROCESSOS DE SOFTWARE

**Organização:** Empresa X Ltda.

**Avaliador Qualificado:** Avaliador Qualificado da Silva

**Projeto:** Sistema de Controle de Pedidos pela Internet para Empresa Z Ltda.

**Unidade Organizacional:** Divisão de Informática.

**Responsável:** Analista Sênior Francisco

**Tamanho:** Médio

**Domínio da Aplicação:** Aplicação comercial, utilizando tecnologia de internet integrada com banco de dados.

**Tamanho e complexidade dos produtos:** Produto de software de porte médio, com alta complexidade por utilizar tecnologia de internet, atender requisitos de segurança e possuir um banco de dados robusto.

**Características de Qualidade:** Usabilidade, portabilidade, manutenibilidade e reutilização.

**Recursos:** Neste projeto foram utilizados um analista sênior, dois analistas programadores, um técnico em redes com conhecimento em *internet* para suporte, dois computadores e ferramentas de software de desenvolvimento para *internet*.

**Justificativa:** Este projeto possui alta complexidade e é um projeto típico de desenvolvimento de software. Para este projeto foram gerados um conjunto de documentos que pode ser analisado.

**Correspondência com o perfil da avaliação:** Para este projeto são analisados todos os processos descritos no documento AR3 Descrição inicial.

---

Avaliador Qualificado

Figura 6.5: Documento AR4 Instâncias dos processos de *software*.

#### **6.2.4 TA2.2 Definir avaliadores**

Esta tarefa definirá os papéis da equipe de avaliadores e analisar se possuem o conhecimento necessário para executar a avaliação, caso não possuam, deve ser providenciado treinamento adequado. A definição dos avaliadores está representada na Figura 6.6 a qual mostra o documento AR5 Avaliadores.

Maringá 14 de janeiro de 2000.

**AVALIADORES****Organização:** Empresa X Ltda.**Avaliador Qualificado:** Avaliador Qualificado da Silva**Avaliadores****Cargo**

Avaliador Qualificado da Silva	Avaliador qualificado
Analista Sênior dos Santos	Avaliador
Analista Programador da Rosa	Avaliador
Analista Programador de Oliveira	Avaliador

**Necessidades de treinamento**

Avaliador qualificado

O avaliador qualificado foi contratado externamente à organização, especialmente para esta avaliação, e após análise de *curriculum*, verificou-se que o mesmo não necessita de treinamento, preenchendo os requisitos descritos na Parte 6 do ISO/IEC TR 15504.

Avaliadores

Após análise da ficha funcional dos Avaliadores, todos internos a organização, verificou-se que os mesmos nunca participaram de processos de avaliação, especialmente baseados no ISO/IEC TR 15504.

---

Avaliador Qualificado

Figura 6.6: Documento AR5 Avaliadores.

### **6.2.5 TA2.3 Treinar avaliadores**

Esta tarefa somente será executada se houver necessidade de treinamento de acordo com o documento AR5 Avaliadores. Uma vez que o treinamento seja necessário, o conteúdo adequado deve ser preparado e os avaliadores devem ser treinados. O treinamento é registrado conforme documento AR6 Registro de treinamento, descrito na Figura 6.7.

## REGISTRO DE TREINAMENTO

**Organização:** Empresa X Ltda.

**Avaliador Qualificado:** Avaliador Qualificado da Silva

**Objetivo do treinamento:** O objetivo deste treinamento é prover conhecimento adequado aos avaliadores para a execução da avaliação dos processos de software da organização.

- Conteúdo:**
1. O que é o ISO/IEC TR 15504
  2. Introdução ao processo de avaliação
  3. Introdução a determinação de capacidade
  4. O modelo contido na parte 5 do ISO/IEC TR 15504
  5. As dimensões de processo
  6. A metodologia de avaliação

**Instrutor:** Avaliador Qualificado da Silva

**Avaliadores:** Analista Sênior dos Santos  
Analista Programador da Rosa  
Analista Programador de Oliveira

**Data de Início:** 01/02/2000

**Data de término:** 05/02/2000

**Carga horária:** 40 horas.

### Controle de frequência

<b>Avaliador</b>	01/02	02/02	03/02	04/02	05/02
Analista Sênior dos Santos					
Analista Programador da Rosa					
Analista Programador de Oliveira					
Comentários sobre o curso:					

---

Avaliador Qualificado

Figura 6.7: Documento AR6 Registro de treinamento.

## 6.2.6 TA2.4 Planejar avaliação

A partir do conhecimento sobre o processo de avaliação, quando forem definidos o perfil da avaliação, a equipe (já treinada), os processos e as instâncias de processos, é necessário realizar um planejamento das tarefas que se seguem, gerando um cronograma de reuniões para coleta dos dados geral e um detalhado, conforme mostram as Figura 6.8a e 6.8b. É necessário também preparar um *checklist* das informações a serem coletadas. Este *checklist* é baseado no modelo de referência do ISO/IEC TR 15504 – Parte 2, seguindo-se o exemplo descrito na Parte 5, o qual deve ser adaptado de acordo com o perfil da avaliação. O *checklist* é dividido em duas partes: práticas de gerenciamento e práticas básicas, conforme mostram as Figuras 6.9a e 6.9b respectivamente. Ambas as partes devem ser respondidas para cada instância de processo selecionada.

<b>EMPRESA X</b>	AR7 Planejamento da avaliação		
	Maringá 14 de janeiro de 2000.		
<b>PLANEJAMENTO DA AVALIAÇÃO</b>			
<b>Organização:</b> Empresa X Ltda.			
<b>Avaliador Qualificado:</b> Avaliador Qualificado da Silva			
<b>Avaliadores:</b> Analista Sênior dos Santos			
Analista Programador da Rosa			
Analista Programador de Oliveira			
<b>Cronograma Geral</b>			
<b>Tarefa</b>	<b>Início</b>	<b>Término</b>	<b>Responsável</b>
Coletar documentos	10/02/00	16/02/00	Avaliadores
Coletar informações	17/02/00	09/03/00	Avaliadores
Verificar informações	10/03/00	30/03/00	Avaliadores
Determinar medidas	31/03/00	13/04/00	Avaliadores
Consolidar medidas	14/04/00	20/04/00	Avaliador Qualificado
Validar medidas	21/04/00	30/05/00	Avaliador Qualificado
Apresentar resultados	31/05/00	31/05/00	Avaliador Qualificado
Página 1 de 2			

Figura 6.8a: Documento AR7 Planejamento da Avaliação.

**PLANEJAMENTO DA AVALIAÇÃO****Organização:** Empresa X Ltda.**Avaliador Qualificado:** Avaliador Qualificado da Silva**Avaliadores:** Analista Sênior dos Santos

Analista Programador da Rosa

Analista Programador de Oliveira

**Cronograma Detalhado**

<b>Assunto</b>	<b>Data / hora</b>	<b>Responsável</b>	<b>Avaliador</b>
Coletar informações	10/02/00 : 09:00	Analista Sênior	Analista Sênior dos Santos
Coletar informações	10/02/00 : 14:00	Analista Programador	Analista Programador da Rosa
Coletar informações	11/02/00 : 09:00	Analista Programador	Analista Sênior dos Santos
Coletar informações	11/02/00 : 14:00	Analista Programador	Analista Programador de Oliveira
Coletar informações	12/02/00 : 09:00	Analista Sênior	Analista Sênior dos Santos
Coletar informações	13/02/00 : 14:00	Analista Programador	Analista Programador da Rosa
Coletar informações	14/02/00 : 09:00	Analista Programador	Analista Sênior dos Santos

Obs: As entrevistas devem ser planejadas para durarem 02 horas no máximo.

Página 2 de 2

---

**Avaliador Qualificado**

---

**Avaliadores**

Figura 6.8b: AR7 Documento Planejamento da avaliação.

Maringá 14 de janeiro de 2000.

**CHECKLIST – PRÁTICAS DE GERENCIAMENTO****Organização:** Empresa X Ltda.**Avaliadores:** Analista Sênior dos Santos

Analista Programador da Rosa

Analista Programador de Oliveira

**Projeto:** Sistema de Controle de Pedidos pela Internet para Empresa Z Ltda.**Avaliação:** Dimensão de capacidade – atributos de processo**Processo:** CUS – Cliente Fornecedor**Categoria de processo:** CUS.2 – Gerenciamento do processo**Atributo:** 2.1 - Gerenciamento da execução

<b>Prática de Gerenciamento</b>	<b>Descrição</b>	<b>Avaliação</b>	
CUS.1.[1];2.1: Referência 2.1.1:	Identifica requisitos para possibilitar o planejamento e execução do processo.	<input type="radio"/> -F	<input type="radio"/> -L
		<input type="radio"/> -P	<input type="radio"/> -N
CUS.1.[1];2.1: Referência 2.1.2:	Planeja desempenho do processo identificando as atividades dos processos e a alocação dos recursos de acordo com os requisitos.	<input type="radio"/> -F	<input type="radio"/> -L
		<input type="radio"/> -P	<input type="radio"/> -N
CUS.1.[1];2.1: Referência 2.1.3:	Implementa as atividades definidas para atingir o propósito do processo.	<input type="radio"/> -F	<input type="radio"/> -L
		<input type="radio"/> -P	<input type="radio"/> -N
CUS.1.[1];2.1: Referência 2.1.4:	Gerencia a execução das atividades para produzir produtos de trabalho dentro do tempo declarado e requisitos de recursos.	<input type="radio"/> -F	<input type="radio"/> -L
		<input type="radio"/> -P	<input type="radio"/> -N

Evidências e observações:

---

 Avaliador

Figura 6.9a: Documento AR8 Checklist - práticas de gerenciamento.

Maringá 14 de janeiro de 2000.

**CHECKLIST – PRÁTICAS BÁSICAS****Organização:** Empresa X Ltda.**Avaliadores:** Analista Sênior dos Santos

Analista Programador da Rosa

Analista Programador de Oliveira

**Projeto:** Sistema de Controle de Pedidos pela Internet para Empresa Z Ltda.**Avaliação:** Dimensão de processo – produtos de trabalho**Processo:** CUS – Cliente Fornecedor**Categoria de processo:** CUS.1 – Aquisição de software

<b>Prática Básica</b>	<b>Descrição</b>	<b>Executa ?</b>
CUS.1.1[1]: Identifica necessidades:	Identifica uma necessidade para adquirir, desenvolver ou melhorar um produto de software.	<input type="radio"/> - Sim <input type="radio"/> - Não
CUS.1.2[1]: Define requisitos:	Identifica os requisitos para um sistema e/ou produto de software que satisfará as necessidades para um novo produto e/ou serviço.	<input type="radio"/> - Sim <input type="radio"/> - Não
CUS.1.3[1]: Prepara estratégia de aquisição:	Prepara uma requisição para proposta, incluindo requisitos para aquisição e calendário de projeto.	<input type="radio"/> - Sim <input type="radio"/> - Não
Evidências e observações:		

---

Avaliador

Figura 6.9b: Documento AR8 Checklist - práticas básicas.

### 6.2.7 TA2.5 Coletar documentos

A partir das instâncias de processos de *software* definidas no documento AR4 Instâncias de processo de *software*, deve-se coletar a documentação gerada para as instâncias de processos e documentá-la conforme o documento AR9 Lista de documentos, mostrado na Figura 6.10a. As impressões sobre estes documentos devem ser confirmadas nas reuniões com os responsáveis pelo projeto. Para ajudar na formação de julgamento sobre os documentos coletados, o ISO/IEC TR 15504 define em sua parte 5 um conjunto de indicadores os quais devem ser mapeados para os documentos, fazendo-se uma relação de equivalência entre os mesmos, conforme mostram as páginas de 2 à 7 do documento AR9 Lista de documentos nas Figuras 6.10a à 6.10g. Com esta relação de equivalência os avaliadores tem um conjunto de características desejáveis, não obrigatório, para a documentação que está sendo avaliada. Os correspondentes no ISO/IEC TR 15504 estão divididos em entrada e saída e preservam a numeração oriunda da norma para facilitar sua localização e acesso.

Maringá 14 de janeiro de 2000.

**LISTA DE DOCUMENTOS****Organização:** Empresa X Ltda.**Avaliador Qualificado:** Avaliador Qualificado da Silva**Avaliadores:** Analista Sênior dos Santos  
Analista Programador da Rosa  
Analista Programador de Oliveira

Sistema de Controle de Pedidos pela Internet para Empresa Z Ltda.

<b>Documento</b>	<b>Local</b>	<b>Meio de armazenamento</b>
Atas de reuniões	Pasta de clientes	Físico – papel
Descritivo do Sistema	f:\sistemas\empresa_z\scp\doc\	Eletrônico – Word 97
Diagramas de Caso de uso	f:\sistemas\empresa_z\scp\doc\	Eletrônico - Rational Rose
Diagrama de Classes	f:\sistemas\empresa_z\scp\doc\	Eletrônico - Rational Rose
Diagramas de Seqüência	f:\sistemas\empresa_z\scp\doc\	Eletrônico - Rational Rose
Contrato	Pasta de clientes	Físico – papel
Cronograma	f:\sistemas\empresa_z\scp\doc\	Eletrônico – Word 97
Projeto de Banco de Dados	f:\sistemas\empresa_z\scp\doc\	Eletrônico – Erwin 3.5.2
Projeto de Interface	f:\sistemas\empresa_z\scp\sis\	Eletrônico – VB 5
Refinamento do protótipo	f:\sistemas\empresa_z\scp\sis\	Eletrônico – Word 97
Cheklists para Verificação e Validação	f:\sistemas\empresa_z\scp\doc\	Eletrônico – Word 97
Resultados de testes	f:\sistemas\empresa_z\scp\doc\	Eletrônico – Word 97
Aceitação pelo cliente	Pasta de clientes	Físico – papel

Página 1 de 7

Figura 6.10a: Documento AR9 Lista de documentos.

Maringá 14 de janeiro de 2000.

**LISTA DE DOCUMENTOS****Organização:** Empresa X Ltda.**Avaliador Qualificado:** Avaliador Qualificado da Silva**Avaliadores:** Analista Sênior dos Santos

Analista Programador da Rosa

Analista Programador de Oliveira

**Conjunto de indicadores para a fase de levantamento de dados**

Documentos	Correspondentes no ISO/IEC TR 15504	
	Entrada ( <i>input</i> )	Saída ( <i>output</i> )
Atas de reuniões	44 Product needs assessment	21 Analysis results
Descritivo do sistema	46 Market analysis	52 System requirements
Diagrama de use case	52 Customer requirements	52 Maintenance requirements
Diagrama de seqüência	52 Maintenance requirements	53 System design/architecture
Diagrama de classes	83 Customer request	58 Trace ability record / mapping
Alterações	84 Problem reports	69 Release strategy/plan
Contrato	94 Change request	95 Change control record
Cronograma	95 Change Control	96 Change History
	30 Review Plan	16 Plan
	27 Quality criteria	51 Contract

Página 2 de 7

Figura 6.10b: Documento AR9 Lista de documentos.

Maringá 14 de janeiro de 2000.

**LISTA DE DOCUMENTOS****Organização:** Empresa X Ltda.**Avaliador Qualificado:** Avaliador Qualificado da Silva**Avaliadores:** Analista Sênior dos Santos

Analista Programador da Rosa

Analista Programador de Oliveira

<b>Conjunto de indicadores para a fase de projeto rápido</b>		
<b>Documentos</b>	<b>Correspondentes no ISO/IEC TR 15504</b>	
	<i>Entrada (input)</i>	<i>Saída (output)</i>
Diagrama de use case	52 Software Requirements	54 High Level Software
Diagrama de seqüência	53 System design/architecture	design
Diagrama de classes	57 Build lists	55 Low level software design
Projeto de banco de dados	83 Customer request	58 Trace ability record/map
Projeto de interface	84 Problem reports	101 Database design
Refinamento do protótipo	94 Change request	
Resultados de testes	95 Change control	

Página 3 de 7

Figura 6.10c: Documento AR9 Lista de documentos.

Maringá 14 de janeiro de 2000.

**LISTA DE DOCUMENTOS****Organização:** Empresa X Ltda.**Avaliador Qualificado:** Avaliador Qualificado da Silva**Avaliadores:** Analista Sênior dos Santos

Analista Programador da Rosa

Analista Programador de Oliveira

<b>Conjunto de indicadores para a fase de programação</b>		
<b>Documentos</b>	<b>Correspondentes no ISO/IEC TR 15504</b>	
	<b>Entrada (<i>input</i>)</b>	<b>Saída (<i>output</i>)</b>
Diagrama de seqüência	10 Coding standards	56 Software units (code)
Diagrama de classes	101 Database design	59 Test plan
Projeto de banco de dados	52 Software requirements	60 Unit test script
	52 System requirements	61 Test case
Projeto de interface	55 Low level software design	62 Test result

Página 4 de 7

Figura 6.10d: Documento AR9 Lista de documentos.

Maringá 14 de janeiro de 2000.

**LISTA DE DOCUMENTOS****Organização:** Empresa X Ltda.**Avaliador Qualificado:** Avaliador Qualificado da Silva**Avaliadores:** Analista Sênior dos Santos

Analista Programador da Rosa

Analista Programador de Oliveira

<b>Conjunto de indicadores para a fase de revisão com usuário</b>		
<b>Documentos</b>	<b>Correspondentes no ISO/IEC TR 15504</b>	
	<i>Entrada (input)</i>	<i>Saída (output)</i>
Contrato	52 Maintenance requirements	57 Build lists
Alterações	52 Software requirements	58 Trace ability
Diagrama de use case	52 System requirements	record/mapping
Diagrama de	53 System design/architecture	60 Integration test script
seqüência	54 High level software design	60 Software test script
Diagrama de classes	55 Low level software design	61 Test case
Projeto de banco de	56 Software units (code)	62 Test result
dados	69 Release strategy/plan	64 Software test plan
Projeto de interface	95 Change control	65 Integration test
Checklist para	51 Contract	strategy/plan
verificação e		67 Regression test strategy
validação		72 Integrated software
Resultados dos testes		

Figura 6.10e: Documento AR9 Lista de documentos.

**LISTA DE DOCUMENTOS****Organização:** Empresa X Ltda.**Avaliador Qualificado:** Avaliador Qualificado da Silva**Avaliadores:** Analista Sênior dos Santos

Analista Programador da Rosa

Analista Programador de Oliveira

<b>Conjunto de indicadores para a fase de refinamento do protótipo</b>		
<b>Documentos</b>	<b>Correspondentes no ISO/IEC TR 15504</b>	
	<i>Entrada (input)</i>	<i>Saída (output)</i>
Refinamento do protótipo	57 Build lists	57 Build lists
Checklist para verificação e validação	62 Test result	83 Customer request
Resultados dos testes	72 Integrated software	84 Problem reports
	95 Change control	94 Change request
		95 Change control

Figura 6.10f: Documento AR9 Lista de documentos.

Maringá 14 de janeiro de 2000.

**LISTA DE DOCUMENTOS****Organização:** Empresa X Ltda.**Avaliador Qualificado:** Avaliador Qualificado da Silva**Avaliadores:** Analista Sênior dos Santos

Analista Programador da Rosa

Analista Programador de Oliveira

<b>Conjunto de indicadores para a fase de engenharia do produto</b>		
<b>Documentos</b>	<b>Correspondentes no ISO/IEC TR 15504</b>	
	<i>Entrada (input)</i>	<i>Saída (output)</i>
Pacote de software	108 System components	31 Review records
Aceitação pelo cliente	73 System	74 Installation plan
	68 Acceptance test strategy / plan	75 Installation guide
	105 customer documentation	78 Delivery instructions
	52 Customer requirements	106 Instalation record
	51 Contract	76 Packaging record
		70 release package
	73 System	81 Acceptance record
		79 Delivery record
		51 Contract

Figura 6.10g: Documento AR9 Lista de documentos.

### **6.2.8 TA2.6 Coletar informações**

Deve-se proceder de acordo com a agenda de reuniões contida no documento AR7 Planejamento da avaliação, elaborada durante o planejamento da avaliação. Caso necessário, outras reuniões devem ser realizadas para dirimir dúvidas na avaliação de documentos e para confirmar impressões observadas durante as entrevistas. As informações são coletadas respondendo-se ao documento AR8 Checklist – práticas básicas e documento AR8 Checklist – práticas de gerenciamento. Estes documentos estão representados nas Figura 6.11a e 6.11b respectivamente.

Maringá 14 de janeiro de 2000.

**CHECKLIST – PRÁTICAS DE GERENCIAMENTO****Organização:** Empresa X Ltda.**Avaliadores:** Analista Sênior dos Santos

Analista Programador da Rosa

Analista Programador de Oliveira

**Projeto:** Sistema de Controle de Pedidos pela Internet para Empresa Z Ltda.**Avaliação:** Dimensão de capacidade – atributos de processo**Processo:** CUS – Cliente Fornecedor**Categoria de processo:** CUS.2 – Gerenciamento do processo**Atributo:** 2.1 – Gerenciamento da execução

<b>Prática de Gerenciamento</b>	<b>Descrição</b>	<b>Avaliação</b>	
		<b>Executado ?</b>	
CUS.1.[1];2.1: Referência 2.1.1:	Identifica requisitos de recursos para possibilitar o planejamento e execução do processo.	o-F	v-L
		o-P	o-N
CUS.1.[1];2.1: Referência 2.1.2:	Planeja a performance do processo identificando as atividades dos processos e a alocação dos recursos de acordo com os requisitos.	v-F	o-L
		o-P	o-N
CUS.1.[1];2.1: Referência 2.1.3:	Implementa as atividades definidas para atingir o propósito do processo.	o-F	v-L
		o-P	o-N
CUS.1.[1];2.1: Referência 2.1.4:	Gerencia a execução das atividades para produzir produtos de trabalho dentro do tempo declarado e requisitos de recursos.	o-F	v-L
		o-P	o-N

**Evidências e observações:** Demonstrou nas entrevistas que conhece e utiliza o planejamento dos processos.

Figura 6.11a: Documento AR8 Checklist – práticas de gerenciamento (respondido).

Maringá 14 de janeiro de 2000.

**CHECKLIST – PRÁTICAS BÁSICAS****Organização:** Empresa X Ltda.**Avaliadores:** Analista Sênior dos Santos  
Analista Programador da Rosa  
Analista Programador de Oliveira**Projeto:** Sistema de Controle de Pedidos pela Internet para Empresa Z Ltda.**Avaliação:** Dimensão de processo – produtos de trabalho**Processo:** CUS – Cliente Fornecedor**Categoria de processo:** CUS.1 – Aquisição de software

<b>Prática Básica</b>	<b>Descrição</b>	<b>Avaliação</b>	
CUS.1.1[1]: Identifica necessidades:	Identifica uma necessidade para adquirir, desenvolver ou melhorar um produto de software.	v-F o-P	o-L o-N
CUS.1.2[1]: Define requisitos:	Identifica os requisitos para um sistema e/ou produto de software que satisfará as necessidades para um novo produto e/ou serviço.	v-F o-P	o-L o-N
CUS.1.3[1]: Prepara estratégia de aquisição:	Prepara uma requisição para proposta, incluindo requisitos para aquisição e calendário de projeto.	o-F o-P	v-L o-N
Evidências e observações:			

---

 Avaliador

Figura 6.11 b: Documento AR8 Checklist – práticas básicas (respondido).

### **6.2.9 TA2.7 Verificar informações**

Os *checklists* preenchidos devem ser verificados quanto à contradições, como por exemplo, o que foi afirmado na entrevista e o que foi evidenciado na documentação, procurando justificar e dirimir qualquer dúvida sobre todas as medidas anotadas no *checklist*. Esta atividade não gera nenhum artefato, apenas valida o *checklist* já existente.

### **6.2.10 TA2.8 Determinar medidas**

A partir das anotações nos *checklists* já validadas, determinar o perfil de cada instância de processo de *software* para as práticas básicas e de gerenciamento, conforme mostram os documentos AR10 Medidas das instâncias de processo de *software* para práticas básicas e AR10 Medidas das instâncias de processo de *software* para práticas de gerenciamento. Estas medidas estão representados nas Figuras 6.12 e 6.13 respectivamente.

**EMPRESA X**

AR10 Medidas das instâncias

Maringá 14 de janeiro de 2000.

## MEDIDAS DAS INSTÂNCIAS

**Organização:** Empresa X Ltda.

**Avaliador qualificado:** Avaliador qualificado da Silva

**Projeto:** Sistema de Controle de Pedidos pela Internet para Empresa Z Ltda.

**Avaliação:** Dimensão de capacidade – atributos de processo

**Processo:** CUS – Cliente Fornecedor

**Categproa de processo** CUS.1 – Aquisição de software

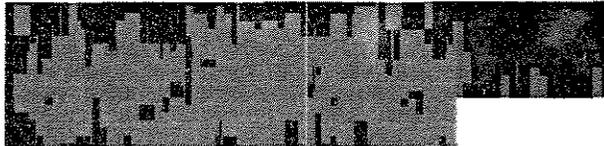
**No      Parcial      Large      Full**

### Práticas Básicas

CUS.1.1[1]: Identifica necessidades:

CUS.1.2[1]: Define requisitos:

CUS.1.3[1]: Prepara estratégia de aquisição:



---

Avaliador Qualificado

Figura 6.12: Documento AR10 Medidas das instâncias de processo de *software* para práticas básicas.

## MEDIDAS DAS INSTÂNCIAS

**Organização:** Empresa X Ltda.

**Avaliador qualificado:** Avaliador qualificado da Silva

**Projeto:** Sistema de Controle de Pedidos pela Internet para Empresa Z Ltda.

**Avaliação:** Dimensão de processo – produtos de trabalho

**Processo:** CUS – Cliente Fornecedor

**Categoria de processo:** CUS.2 – Gerenciamento do processo

	No	Parcial	Large	Full
<b>Práticas de gerenciamento</b>				
CUS.1.[1];2.1: Referência 2.1.1:				
CUS.1.[1];2.1: Referência 2.1.2:				
CUS.1.[1];2.1: Referência 2.1.3:				
CUS.1.[1];2.1: Referência 2.1.4:				

---

Avaliador Qualificado

Figura 6.13: Documento AR10 Medidas das instâncias de processo de *software* para práticas de gerenciamento.

### 6.2.11 TA2.9 Consolidar medidas

A partir das medidas de cada instância de processo de *software* conforme descrito no documento AR10 Medidas das instâncias de processo de *software*, gerar o perfil da avaliação, conforme mostra o documento AR11 Medidas do processo de *software* descrito na Figura 6.14.

<b>EMPRESA X</b>	AR11 Medidas do processo de software					
	Maringá 14 de janeiro de 2000.					
<b>MEDIDAS DO PROCESSO DE SOFTWARE</b>						
<b>Organização:</b> Empresa X Ltda.						
<b>Avaliador qualificado:</b> Avaliador qualificado da Silva						
<b>Projeto:</b> Sistema de Controle de Pedidos pela Internet para Empresa Z Ltda.						
<b>Avaliação:</b> Dimensão de processo						
<b>Processo:</b> CUS – Cliente Fornecedor						
	<b>Nível</b>					
<b>Instâncias de processo de software</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
Sistema de Controle de Pedidos pela Internet para Empresa Z Ltda						
_____ Avaliador Qualificado						

Figura 6.14: Documento AR11 Medidas do processo de *software*.

### 6.2.12 TA2.10 Validar medidas

A partir do perfil do processo consolidado conforme descrito no documento AR11 Medidas do processo de *software*, verificar as medidas de cada processo de *software* conforme descrito no documento AR10 Medidas das instâncias de processo de *software*. Verificar se o mesmo atende aos objetivos da avaliação, solicitados pelo patrocinador. Esta tarefa não gera artefatos, apenas atualiza o documento AR11 Medidas do processo de *software*, se necessário.

### 6.2.13 TA3 Apresentar resultados

A partir das medidas dos processo de *software* conforme descritos nas Seções 6.2.10 e 6.2.11, apresentar os resultados da avaliação de acordo com o documento AR3 Entradas da avaliação o qual contém o contrato de confidencialidade e conforme descrito na tarefa TA3 Apresentar resultados, descrita na Seção 5.2.13 no Capítulo 5.

## 6.3 A avaliação utilizando o ExpPSEE

Para análise da possibilidade de automação da metodologia apresentada no Capítulo 5 em um PSEE, foi aplicado o estudo de caso apresentado no item 6.2 ao ambiente ExpPSEE descrito na Seção 4.3. Para isto foram seguidas algumas etapas as quais estão descritas nas subseções seguintes.

### 6.3.1 Definição da arquitetura

Para construção da arquitetura do processo de avaliação de processo de *software*, são definidos para cada tarefa seus tipo papéis, artefatos e ferramentas, bem como os relacionamentos entre si. Esses relacionamentos suportam as definições apresentadas na Seção 5.2, referente às definições para o ExpPSEE. A ferramenta de definição da arquitetura de processo de *software* do ExpPSEE, o *Architecture builder*, permite a definição desses relacionamentos. A definição da arquitetura é apresentada na Figura 6.15, a qual mostra os tipos ferramenta (Editor), tipo tarefa (Definir instâncias, Definir Avaliadores e Planejar Avaliação), o tipo ator (Avaliadores) e os tipos artefatos (Entradas Avaliação, Instâncias de processo, Registro de treinamento, Avaliadores, Planejamento da avaliação e Checklists). As tarefas serão programadas e os tipos instanciados conforme mostra o diagrama de classes do gerenciador de processos, Figura 4.3 na Seção 4.3.4.

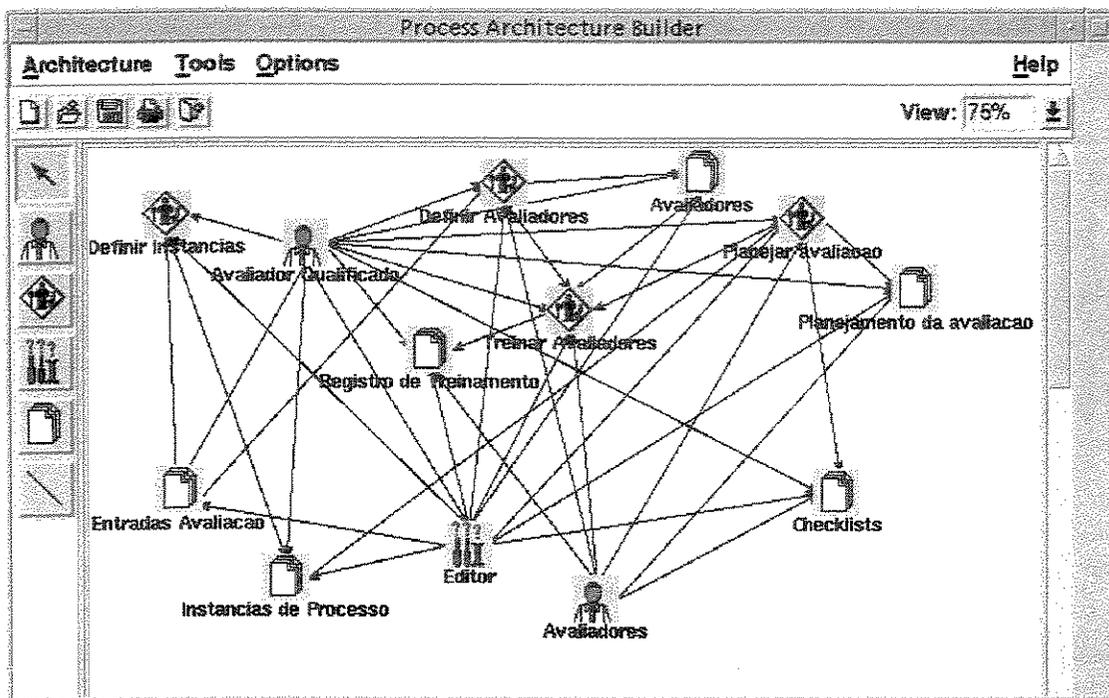


Figura 6.15 – Arquitetura do processo de avaliação do processo de *software*.

### 6.3.2 Instanciação

Uma vez definida a arquitetura do processo, ela é transformada em uma *baseline*, onde a arquitetura é validada por um *Object Inspector* (componente da ferramenta *Architecture builder* do ExpPSEE). Se a arquitetura estiver completa, poderá ser instanciada, senão os erros serão apontados e a operação abortada. Uma vez como *baseline*, a arquitetura poderá ser transformada em um projeto pela ferramenta *Project manager* do ExpPSEE, conforme mostra a Figura 6.16.



Figura 6.16 – Instanciação da arquitetura do processo de *software*.

### 6.3.3 Programação

Na fase de programação, será gerado um programa para a execução de cada tarefa. Essa programação executará comandos em TCL/TK contendo as pré-condições, o *script* da execução da tarefa propriamente dita e as pós-condições da tarefa. Na janela apresentada na Figura 6.17, é selecionado o projeto, a tarefa e o papel para o qual será feita a programação.

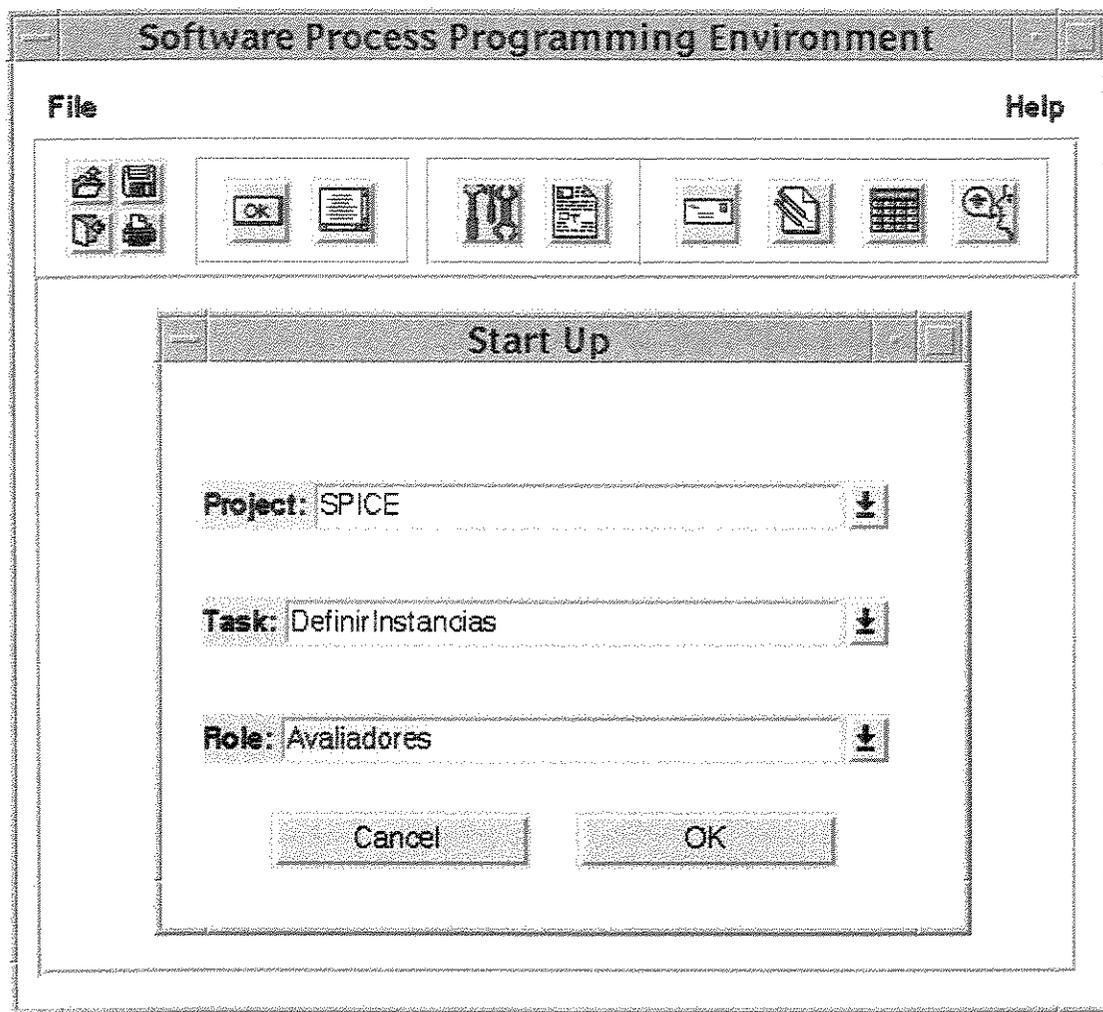


Figura 6.17: Janela Start up.

A partir da seleção do projeto Controle de pedidos pela Internet, da tarefa Definir instâncias e do papel Avaliador Qualificado na janela *Start up* conforme indicado na Figura 6.19, a programação da tarefa é realizada na janela *Workcontext*. Um exemplo de janela *Workcontext* para as atividades mostradas na arquitetura do processo de *software* definida na Seção 6.3.1, bem como o código fonte que deve ser gerado por cada janela é apresentado no anexo B.

#### 6.3.4 Alocação

Após realizada a programação, são alocados os elementos reais para cada tipo definido na arquitetura da Figura 6.15. Esses elementos reais constituem-se de documentos físicos para os artefatos e atores para os papéis selecionados, conforme mostra a Figura 6.18.

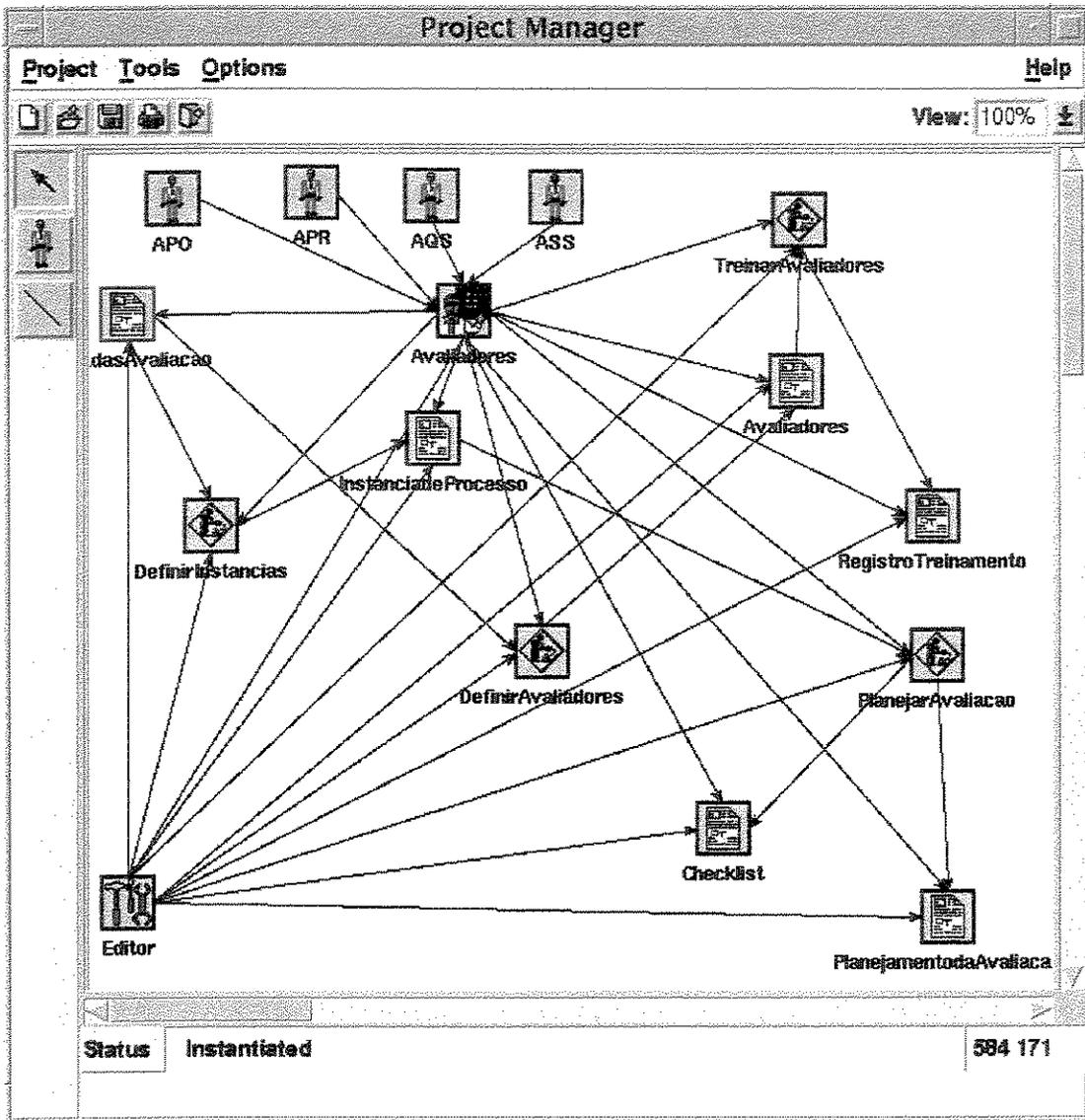


Figura 6.18 – Instância de processo instanciada, programada e alocada.

### 6.3.5 Tarefa no estado pronta

Uma vez alocada a tarefa está pronta para execução, ou seja, ela está na agenda do ator alocado, porém só poderá ser iniciada uma vez que todas as suas pré-condições sejam satisfeitas. A agenda é mostrada na Figura 6.19.

Project: The name of the Project  
 Actor: Actor's Name  
 Role: Avaliador      Order By: Task Name

Status	Task	Description	Start Date	End Date
<input checked="" type="radio"/>	Definir Instancias	Escolher as instancias de processo de		
<input type="radio"/>	Definir Avaliadores	Escolher a equipe de avaliadores e se		
<input type="radio"/>	Treinar Avaliadores	Treinar na metodologia e nos conceit		
<input type="radio"/>	Planejar Avaliacao	Elaborar cronograma detalhado da av		

Figura 6.19: Agenda de tarefas.

### 6.3.6 Tarefa no estado executando

Uma vez que todas as pré-condições da tarefa sejam satisfeitas, a execução poderá ser iniciada pelo ator designado. A execução é realizada através da interpretação do código TCL/TK gerado conforme mostra o anexo B, o qual contém chamadas a métodos do gerenciador de processos que farão o acesso ao banco de dados e aos recursos alocados disponíveis no sistema.

### 6.3.7 Tarefa no estado finalizada

A tarefa será considerada finalizada, quando todas as pós-condições forem satisfeitas. Estas pós-condições são testadas a cada execução da tarefa, através do código programado em TCL/TK para cada janela Work context constante no anexo B.

## 6.4 Considerações finais

Neste capítulo foi apresentado um estudo de caso. Todos os documentos definidos para a metodologia MAPS-15504 descrita no Capítulo 5 foram gerados e chegou-se ao perfil do processo de *software* da Empresa X Ltda. Após a realização do estudo de caso, foi descrita a automação da metodologia no ambiente ExpSEE, as ferramentas utilizadas na definição da arquitetura do processo de avaliação, na programação das tarefas e a instanciação da arquitetura definida.

Nas subseções seguintes, é apresentada uma avaliação da metodologia proposta destacando-se as vantagens e as limitações da metodologia e de sua automação no ambiente do ExpSEE. Além disso, no anexo C é apresentado um mapeamento entre os requisitos do ISO/IEC TR 15504 e como eles são atendidos no ExpSEE.

#### 6.4.1 Vantagens da metodologia MAPS-15504

A metodologia MAPS-15504 tornou possível a definição de um processo de avaliação completo centrado em tarefas, porque através dela foi possível contornar algumas limitações do ISO/IEC TR 15504, tais como:

- a) a metodologia possui duas tarefas bem definidas para a definição das entradas da avaliação: TA1.1 Descrição inicial e TA1.2 Definir entradas. Isto está fora do escopo do ISO/IEC TR 15504, conforme descrito na Seção 5.2.1.2 e 5.2.2.2. Estas tarefas deixam mais clara a estruturação do processo de avaliação, do que apenas apresentar os requisitos das entradas da avaliação como faz o ISO/IEC TR 15504.
- b) a metodologia possui a tarefa TA2.5 Coletar documentos a qual não é referenciada pelo ISO/IEC TR 15504 conforme descrito na Seção 5.2.7.2. Esta coleta de documentos traz ganhos de tempo e qualidade da avaliação, pois as entrevistas de avaliação são conduzidas de posse dos documentos com o objetivo de esclarecer dúvidas e confirmar impressões, assim, não se transformarão em uma grande busca por documentos;
- c) o ISO/IEC TR 15504 não apresenta nenhum requisito sobre o processo de avaliação, portanto o uso da metodologia proposta acrescenta um processo e um conjunto de documentos bem definidos para a realização da avaliação.

#### 6.4.2 Limitações na utilização da metodologia MAPS-15504

A metodologia MAPS-15504 ainda possui algumas limitações do ISO/IEC TR 15504. Essas limitações foram observadas durante o estudo de caso e para sua solução precisam ser melhor definidas no ISO/IEC TR 15504 ou precisam apoiar-se em outras metodologias. Estas limitações são:

a tarefa TA1.2 Definir Entradas precisa ser refinada com outras metodologias, conforme descrito na Seção 5.2.2.2, pois não há nenhuma indicação sobre a composição do perfil da avaliação.

a tarefa TA2.1 Definir instâncias, precisa ser detalhada pois é preciso garantir que o conjunto de instâncias de processo escolhido realmente representa os processos de *software* da empresa conforme descrito na Seção 5.2.3.2. Isto é apenas um requisito no ISO/IEC TR 15504 atualmente.

as tarefas TA2.2 Definir avaliadores e TA2.3 Treinar avaliadores ainda não possuem requisitos sobre os membros da equipe de avaliação, conforme descrito na Seção 5.2.4.2. A diferença de experiência da equipe de avaliação pode agravar o caráter subjetivo da avaliação prejudicando a repetibilidade da mesma.

a execução das tarefas TA2.7 Verificar informações e TA2.10 Validar medidas, carecem de referências cruzadas, conforme descrito na Seção 5.2.9.2 e 5.2.12.2,

respectivamente. Essas referências cruzadas darão suporte as tarefas de verificação e validação.

#### 6.4.3 Vantagens da automação da metodologia MAPS-15504 no ambiente do ExpPSEE

Dentre as vantagens de obter um processo de avaliação bem definido e automatizado, pode-se destacar as seguintes:

- a) a definição e validação do processo de avaliação: a definição da arquitetura do processo da MAPS-15504 no ExpPSEE força a definição do processo e permite a validação do mesmo antes de ser instanciado.
- b) a alocação de recursos: todos os recursos necessários para a execução da avaliação devem ser alocados antes do início da avaliação, na fase de alocação.
- c) o registro: todos os documentos gerados, todas as ações dos avaliadores, inclusive os resultados da avaliação são armazenados no repositório de dados do ambiente do ExpPSEE.
- d) o gerenciamento do processo: o gerenciamento do processo dá-se através do controle sobre o andamento e o estado de cada tarefa a qualquer tempo, bem como a automação do planejamento da avaliação. Estes recursos vem suprir uma limitação do ISO/IEC TR 15504, conforme descrito na Seção 5.2.6.2, pois o mesmo não define parâmetros para o planejamento da avaliação, ficando isto a cargo da experiência do avaliador qualificado.
- e) a reutilização do processo: uma vez que a arquitetura da metodologia está definida no ambiente do ExpPSEE, ela poderá ser reutilizada sempre que necessário.
- f) o controle de configuração: todas as ações sobre os documentos são definidas nas tarefas do processo, permitindo controlar todos os documentos gerados na avaliação.
- g) a validação automática das condições para execução de cada tarefa: os testes sobre as pré e pós-condições garantem que a tarefa só será executada caso ela possa ser executada, garantindo que a avaliação seguirá conforme as definições da metodologia.
- h) a automação dos checklists: os *checklists* podem ser programados na tarefa, gerando um repositório de dados, o qual pode ser considerado apenas mais um documento (artefato) para o ambiente do ExpPSEE.
- i) a geração automática dos resultados da avaliação: a partir deste repositório de dados, nas tarefas TA2.8 Determinar medidas do processo e TA2.9 Determinar medidas, pode-se gerar automaticamente os resultados da avaliação.
- j) o controle de acesso aos resultados da avaliação: como descrito no item f desta Seção, na definição da arquitetura são definidos os tipo papéis e suas ações sobre os documentos, portanto tem-se um acesso controlado aos documentos.

#### **6.4.4 Limitações da automação da metodologia MAPS-15504 no ambiente do ExpPSEE**

Durante a utilização da metodologia no ambiente do ExpPSEE, foram observadas algumas limitações, a saber:

a definição dos formulários: os formulários estão projetados como se fossem preenchidos no papel. Assim sua transposição para o ambiente de programação podem sofrer modificações para melhor aproveitamento dos recursos disponíveis na automação.

a modelagem de dados dos *checklists*: a metodologia não apresenta um modelo de dados que permita a automação dos *checklists* e geração automática das medidas.

Estas limitações apareceram somente no estudo de caso pois considerou-se que no ambiente do usuário não existe nenhuma ferramenta de automação de processo. Porém, a execução baseada em papel, sem a automação dos formulários implicou nas limitações enumeradas acima, portanto, esta condição deve ser repensada. Uma solução é disponibilizar a agenda de tarefas do ExpPSEE na Internet, o que permite a execução remota do processo de avaliação. Esta é uma nova linha de trabalhos que está sendo proposto para o ambiente do ExpPSEE.

## Capítulo 7

### Conclusões e trabalhos futuros

Devido a busca incessante por qualidade de *software* a comunidade de engenharia de *software* produziu diversas normas. As principais normas são apresentadas e comparadas neste trabalho, bem como as idéias centrais dos autores que as influenciaram. Outro resultado das pesquisas desta comunidade são os ambientes de engenharia de *software* centrados no processo (PSEE), os quais visam a automação do processo de *software*. Alguns dos principais ambientes foram apresentados e comparados neste trabalho e foi apresentado o ExpPSEE o qual foi utilizado como estudo de caso para automação da metodologia proposta.

Estas duas idéias, normatização e automação do processo de *software*, compõem o objetivo central deste trabalho: avaliação da qualidade do processo de *software* através de um processo automatizado.

Para isto foi desenvolvida uma metodologia de avaliação de processo de *software* com base no ISO/IEC TR 15504, denominada MAPS-15504. Esta metodologia foi implementada no ambiente do ExpPSEE para avaliação do processo de *software* da Empresa X Ltda.

As principais contribuições deste trabalho foram: a investigação criação de uma metodologia de avaliação de processo de *software* com base no ISO/IEC TR 15504 e a investigação da automação desta metodologia em PSEE.

A partir de um estudo mais detalhado do ISO/IEC TR 15504, concluiu-se que o mesmo não tem por objetivo apresentar uma metodologia de avaliação de processo, mas apresenta uma estrutura (*framework*) para sua avaliação. Portanto, o primeiro esforço deste trabalho foi definir um conjunto de tarefas e formulários para a avaliação de processos de *software* que atendessem aos critérios da parte 3 do ISO/IEC TR 15504. Durante esse trabalho de definição, notou-se que a estrutura da norma não é completa pois não define:

- a estruturação do empreendimento, pois não possui critérios bem definidos para a escolha dos processos de *software* que comporão o perfil da avaliação;
- o perfil da avaliação, pois não possui critérios claros para a escolha das instâncias de processos de *software* que serão avaliadas para determinar o perfil dos processos da organização;
- o perfil da equipe de avaliação, pois isto fica a cargo do responsável pela condução da avaliação, o avaliador qualificado, para o qual a parte 6 do ISO/IEC TR 15504 define todos os requisitos;
- os parâmetros para o planejamento da avaliação;
- a coleta de documentos como uma fase importante do processo de avaliação embora esses sejam a base para conduzir as medições das práticas básicas;
- as referências cruzadas entre os resultados obtidos e a composição do perfil de cada avaliação. As referências cruzadas podem apontar falhas na avaliação, ou pontos

que deveriam ser avaliados a partir de resultados obtidos. Com isto poderiam ser minimizadas falhas na avaliação principalmente por avaliadores inexperientes melhorando a repetibilidade.

A metodologia proposta além de ser uma contribuição que o ISO/IEC TR 15504 não se propõe a fornecer, contribui para o processo de avaliação completando algumas lacunas citadas acima, pois:

- introduz duas tarefas para a definição das entradas da avaliação: TA1.1 Descrição inicial e TA1.2 Definir entradas, contribuindo para a estruturação do processo de avaliação;
- introduz a tarefa TA2.5 Coletar documentos o que traz ganhos de tempo e qualidade da avaliação;
- define um processo de avaliação claro e passível de automação;
- introduz o gerenciamento automatizado do processo através do controle sobre o andamento e o estado de cada tarefa a qualquer tempo;
- permite a reutilização da arquitetura da avaliação;
- garante a execução da tarefa somente se ela puder ser executada através de pré e pós-condições;
- automatiza os checklists que podem ser programados na tarefa;
- gera automaticamente os resultados da avaliação;
- controla o acesso aos resultados da avaliação.

A partir da utilização da metodologia proposta no ambiente do ExpPSEE, podem ser apresentadas algumas considerações sobre:

- a) o estudo de caso: a utilização de um estudo de caso fictício não foi a situação ideal, pois já na definição dos documentos foi inevitavelmente induzido o padrão de qualidade desejado. Porém, as empresas que utilizaram o ISO/IEC TR 15504 para avaliação de seus processos não disponibilizam os documentos, apenas parte dos resultados da avaliação. Isto é compreensível, tal o grau de exposição que sofreriam seus processos organizacionais.
- b) as mudanças no ambiente: o ExpPSEE é um ambiente de pesquisa, isso significa que outros estudos estão sendo desenvolvidos paralelamente a este trabalho e algumas partes estão em fase de implementação enquanto novas ferramentas estão sendo acrescentadas e testadas. Em função disto, não foi possível obter uma execução completa da metodologia, faltou por exemplo a definição e alimentação da agenda de tarefas.
- c) a subutilização do ExpPSEE: muitas das funcionalidades do ambiente ExpPSEE não foram utilizadas neste trabalho. Por exemplo, a programação das tarefas, exceto a verificação das pré e pós-condições, limitou-se a edição de documentos via um editor de textos, logo os documentos para o ambiente são tratadas como “caixas pretas”. Assim nenhuma forma automática de tomada de decisão, avaliação ou qualquer outro tipo de inferência sobre o documento foi possível. Isso limitou em muito a utilização das funcionalidades do gerenciador de processos (*process manager*). O escalonador de tarefas do gerenciador de processos não foi utilizado pelo fato da agenda de tarefas estar

em fase de implementação, isto significa que o comportamento pró-ativo do ambiente não foi explorado na execução das tarefas de avaliação.

Além destas considerações, o estudo de caso mostrou que a automatização das tarefas traz grandes benefícios para o processo de avaliação. A definição e validação do processo antes de sua execução, permitindo a alocação de todos os recursos necessários para a execução da avaliação proporcionam maior entendimento, clareza e facilitam o planejamento da avaliação.

A definição do perfil da avaliação em formulários próprios de acordo com os requisitos do ISO/IEC TR 15504 e sua automação na programação das tarefas, permite não só o atendimento destes requisitos, como também a garantia de execução das tarefas através de pré e pós condições que são verificadas via programação. Uma vez as tarefas programadas, o ambiente passa a controlar a agenda dos avaliadores, com a possibilidade de agregação de outras ferramentas como vídeo conferência, *e-mail*, dentre outras.

A programação das tarefas, além da simples automação de formulários deve atender a requisitos específicos da avaliação de processos, como o suporte automatizado a *checklists* e geração automática dos perfis das instâncias de processo e do processo de *software* como um todo. O suporte automatizado poderá melhorar a qualidade das informações que serão analisadas, validando a entrada de dados.

A Tabela 7.1 apresenta uma comparação do ExpPSEE com a ferramenta SEAL<sup>24</sup>. A ferramenta SEAL foi escolhida por ser um produto de *software* específico para a avaliação segundo o ISO/IEC TR 15504. Esta ferramenta foi a mais utilizada durante a segunda fase dos *trials* do projeto SPICE, conforme descrito em [SEAL99].

Tabela 7.1: Características de uma ferramenta para automação do processo de *software*.

Característica	SEAL	ExpPSEE
Definição e validação do processo de avaliação segundo uma metodologia compatível com a ISO/IEC TR 15504.		X
Permitir a alocação dos recursos necessários para execução da metodologia.		X
Permitir a definição do perfil da avaliação segundo as definições da ISO/IEC TR 15504.	X	X
Permitir a definição da estrutura da documentação necessária (artefatos) para a execução da avaliação.		X
Fornecer suporte a uma agenda de tarefas para múltiplos avaliadores permitindo assim um melhor gerenciamento do processo.		X

<sup>24</sup> SEAL : Software Engineering Applications Laboratory – Laboratório de Aplicações em Engenharia de Software.

Automatizar aos <i>checklists</i> utilizados durante a avaliação de acordo com o perfil da avaliação elaborado.	X	X
Verificar os formulários respondidos quanto a sua completitude.		X
Gerar de maneira automática os perfis das instâncias de processo de <i>software</i> avaliados.	X	X
Gerar de maneira automática o perfil do processo de <i>software</i> da organização.	X	X
Disponibilizar os resultados.	X	X

Atualmente o ExpPSEE suporta a execução de comandos TCL/TK juntamente com a chamada a métodos do gerenciador de processos do ExpPSEE através de um interpretador TCL/TK estendido [CARN99]. Isso permite a criação de formulários genéricos, bem como a programação de atividades com certo grau de complexidade. Quanto ao armazenamento e recuperação das informações, podem ser definidas classes no banco de dados *Object Store* que manipulem os dados de tais formulários.

Em relação ao ISO/IEC TR 15504, a metodologia proposta neste trabalho deixou algumas questões em aberto que merecem ser investigadas em trabalhos futuros: a composição do perfil da avaliação e como ele influencia a definição das instâncias de processo, o perfil da equipe de avaliação e um conjunto de referências cruzadas que auxilie na validação dos dados da avaliação.

Em relação ao ambiente ExpPSEE, conclui-se que é necessário conhecer a estrutura dos documentos manipulados pelo ambiente, para que a programação das tarefas possa utilizar-se destas informações. Portanto, uma ferramenta que suporte a edição de documentos deve ser integrada ao sistema e sua comunicação com o ambiente deve ser melhor investigada.

O Anexo D apresenta uma cópia do artigo MAPS-15504 - Uma Metodologia de Avaliação de Processo de Software para o Ambiente ExpPSEE baseado no ISO/IEC TR 15504, o qual foi aceito para publicação no WQS2000 – Workshop de Qualidade de Software – que será realizado no dia 06 de outubro de 2000 em João Pessoa na Paraíba.

## Referências Bibliográficas

- [ABNT94a] ABNT – Associação Brasileira de Normas Técnicas, **ISO 9126 - Tecnologia de informação – avaliação do produto de *software* – características de qualidade e diretrizes para o seu uso**, 1994.
- [ABNT94b] ABNT – Associação Brasileira de Normas Técnicas, **ISO 9000 - Normas de gestão da qualidade e garantia da qualidade – Diretrizes para seleção e uso**, 1994.
- [ABNT94c] ABNT – Associação Brasileira de Normas Técnicas, **ISO 9001 - Sistemas da qualidade - Modelo para garantia da qualidade em projetos, desenvolvimento, produção, instalação e assistência técnica**, 1994.
- [ABNT94d] ABNT – Associação Brasileira de Normas Técnicas, **ISO 9002 - Sistemas da qualidade – Modelo para garantia da qualidade em produção e instalação**, 1994.
- [ABNT94e] ABNT – Associação Brasileira de Normas Técnicas, **ISO 9003 - Sistemas da qualidade – Modelo para garantia da qualidade em inspeção e ensaios finais**, 1994.
- [ABNT94f] ABNT – Associação Brasileira de Normas Técnicas, **ISO 9004 – Gestão da qualidade e elementos do sistema da qualidade - Diretrizes**, 1994.
- [ABNT96] ABNT – Associação Brasileira de Normas Técnicas, **ISO 9000-3 - Diretrizes para a aplicação da norma NBR ISO 9001 ao desenvolvimento, fornecimento e manutenção de *software***, 1996.
- [BAND92] BANDINELLI, Sergio, Pasquale Armenise, Carlo Ghezzi, and Angelo Morzenti. *Software process representation languages: survey and assessment*. In Fourth International Conference on Software Engineering and Knowledge Engineering, pages 455-462, Capri, Italy, June 1992.
- [BOOC99] BOOCH, Grady, James rumbaugh, Ivar Jacobson, *Unified Modeling Language User Guide*, Addison Wesley, 1999.
- [CABR98] CABREIRA, Mário, **Apostilas do curso de Leitura e interpretação das normas ISO 9000**, Qualysul, 1998.
- [CARN99] CARNIELLO, Andréia, Itana M. S. Gimenes, **Um ambiente de programação de processos cooperativos de *software* para o ExpSEE**, Universidade Estadual de Maringá, Paraná, 1999.
- [CHRI95] CHRISTIE, Alan M., *Software Process Automation*, Springer-verlag: Berlin Heidelberg , New York, USA, 1995.

- [FINK94] FINKELSTEIN, Anthony, Jeff Kramer and Bashar Nuseibeh, *Software Process Modelling and Technology*, John Willey & Sons Inc.: New York , New York, 1994.
- [GILL97] GILLIES, Alann C. *Software Quality Theory and Management – International Thomsom computer press, second edition, London, UK, 1997.*
- [GIME94] GIMENES, Itana I. M. S. **Uma Introdução ao Processo de Engenharia de Software**, XIII Jornada de atualização em Informática: Caxambu – MG, 1994.
- [GIME98] GIMENES, Itana M. S., Sérgio R. P. da Silva, Sérgio Akira Ito, Armando Haerberer, **FORMLAB: Um Ambiente Integrado de Apoio aos Métodos Formais para o Desenvolvimento de Software**, Relatório Técnico, Universidade Estadual de Maringá, Departamento de Informática, 1998.
- [GIME99] GIMENES, I. M. de S., Huzita, E. H. M., Carniello, A., Fantinato, M., **ExpSEE – An Experimental Process Centred Software Engineering Environment**, Relatório Técnico, Universidade Estadual de Maringá, Paraná , 1999.
- [HUMP99] HUMPHREY Watts S. *Managing the software process*, Reading, Mass: Addison-Wesley, 1989.
- [ISO86] ISO 8402, *Quality vocabulary*, 1986.
- [ISO94] JTC1/SC7, **ISO/IEC 12207 - Software life-cycle process**, 1994.
- [ISO98] ISO/IEC/JTC1/SC7/WG10, *Phase 2 Trials Interim Report*, Versão 1.0, 1998.
- [ISO99] ISO/IEC/JTC1/SC7/WG10, **ISO/IEC TR 15504 - Software Process Improvement and Capability dEtermination - Technical Report**, 1999.
- [ISO99-1] ISO/IEC/JTC1/SC7/WG10, **ISO/IEC TR 15504 - Software Process Improvement and Capability dEtermination - Technical Report**, Parte 1 – Conceitos e guia introdutório, 1999.
- [ISO99-2] ISO/IEC/JTC1/SC7/WG10, **ISO/IEC TR 15504 - Software Process Improvement and Capability dEtermination - Technical Report**, Parte 2 – Um modelo de referência, 1999.
- [ISO99-3] ISO/IEC/JTC1/SC7/WG10, **ISO/IEC TR 15504 - Software Process Improvement and Capability dEtermination - Technical Report**, Parte 3 – Execução de avaliação, 1999.
- [ISO99-4] ISO/IEC/JTC1/SC7/WG10, **ISO/IEC TR 15504 - Software Process Improvement and Capability dEtermination - Technical Report**, Parte 4 – Guia para executar avaliações, 1999.

- [ISO99-5] ISO/IEC/JTC1/SC7/WG10, **ISO/IEC TR 15504 - Software Process Improvement and Capability dEtermination - Technical Report**, Parte 5 – Um modelo de avaliação e guia de indicadores, 1999.
- [ISO99-6] ISO/IEC/JTC1/SC7/WG10, **ISO/IEC TR 15504 - Software Process Improvement and Capability dEtermination - Technical Report**, Parte 6 – Guia para qualificação de avaliadores, 1999.
- [ISO99-7] ISO/IEC/JTC1/SC7/WG10, **ISO/IEC TR 15504 - Software Process Improvement and Capability dEtermination - Technical Report**, Parte 7 – Guia para uso em melhoria de processos, 1999.
- [ISO99-8] ISO/IEC/JTC1/SC7/WG10, **ISO/IEC TR 15504 - Software Process Improvement and Capability dEtermination - Technical Report**, Parte 8 – Guia para uso da determinação da capacidade dos processos do fornecedor, 1999.
- [ISO99-9] ISO/IEC/JTC1/SC7/WG10, **ISO/IEC TR 15504 - Software Process Improvement and Capability dEtermination - Technical Report**, Parte 9 – Vocabulário, 1999.
- [MCT00] MCT – Ministério da Ciência e Tecnologia, **Qualidade no setor de software brasileiro**, SEPIN - Secretaria de Política de Informática e Automação, 2000.
- [MCT97] MCT - Ministério da Ciência e Tecnologia, **Relatório do Programa Brasileiro de Qualidade e Produtividade - PBQP**, 1997.
- [MCT99] MCT – Ministério da Ciência e Tecnologia, **Qualidade no setor de software brasileiro**, SEPIN - Secretaria de Política de Informática e Automação, disponível em <http://www.mct.gov.br/sepin/dsi/qualidad/indic.htm>, 1999
- [OBJE95] OBJECT DESIGN, **ObjectStore C++ - API User Guide** – Release 4, 1995.
- [OLIV97] OLIVEIRA, Angelina A. P. de, Claudete M. R., Eduardo P. de Souza, Maria Tereza V. Aguayo, Regina M. t. Colombo, Sílvia Raymundo Lopes, Sônia T. Mainlinguer, **La Experiencia del Brasil en la aplicación de normas para evaluación de la calidad de producto de software**, 1º Simpósio latinoamericano de calidad y productividad en desarrollo de *software* – INTEC, Santiago, Chile, 1997.
- [OSTE87] OSTERWEIL, Leon, **Software Processes are Software Too**, University of Colorado boulder, Colorado, USA, ACM, 1987.
- [PRES95] PRESSMAN, Roger S. **Engenharia de Software**, Makron Books do Brasil Editora Ltda: São Paulo, 1995.
- [SALV96] SALVIANO, Clênio F., Alfredo N.; Claudete M. Rego, Gláucia F. Azevedo, Luciano K. Meneghetti, Márcia C. C. Costa, Mário Bento de Carvalho, Regina M.

T. Colombo, **Qualidade de software: visões de produto e processo de software**, disponível em [www.cti.br](http://www.cti.br), Fundação CTI: Campinas, 1996.

[SEAL99] SEAL - Software Engineering Applications Laboratory, *Learning to Use the SEAL Process Assessment Tool*, Revision 0.90, África do Sul, 1999.

[ZARE90] ZARELLA, Paul F., *CASE tool integration and Standarization, Technical Report*, CMU/SEI-90-TR-14, Carniagie Mellon University, Pittsburgh Pennsylvania, 1990

## Anexo A - Modelos de artefatos.

Neste anexo são apresentados os modelos de cada um dos documentos (artefatos) apresentados na metodologia descrita no Capítulo 5.

### A.1 TA1.1 Descrever avaliação

<b>EMPRESA</b>	AR2 Descrição inicial Local e data.
<b>DESCRIÇÃO INICIAL</b>	
<b>Organização:</b>	
<b>Patrocinador:</b>	
<b>Propósito:</b>	
_____ A diretoria	

Figura A.1: Modelo do documento AR2 Descrição inicial

## A.2 TA1.2 Definir entradas da avaliação

<b>EMPRESA</b>	AR3 Entradas da avaliação Local e data.
<b>ENTRADAS DA AVALIAÇÃO</b>	
<b>Organização:</b>	
<b>Patrocinador:</b>	
<b>Avaliador Qualificado:</b>	
<b>Propósito:</b>	
<b>Avaliadores:</b>	
<b>Processos de Software:</b>	
<b>Metodologia:</b>	
<b>Modelo Compatível:</b>	
<b>Data de Início:</b>	
<b>Data de Término:</b>	
<b>Previsão de horas:</b>	
_____	_____
Patrocinador	Avaliador Qualificado

Figura A.2: Modelo do documento AR3 Entradas da avaliação.

**EMPRESA**

AR3 Entradas da avaliação

Local e data.

## ACORDO DE CONFIDENCIALIDADE

<b>Organização:</b>	
<b>Patrocinador:</b>	
<b>Avaliador Qualificado:</b>	

**Parágrafo 1:** Do objeto deste acordo de confidencialidade :

Este acordo de confidencialidade está restrito a avaliação do processo de software conforme descrita no documento DTA2.2 Descrição do processo de avaliação.

**Parágrafo 2:** Dos direitos sobre os resultados:

Fica estabelecido que todos os direitos sobre os resultados da avaliação são de propriedade da Empresa ....

**Parágrafo 3:** Dos direitos sobre o uso dos resultados:

Fica estabelecido que a Empresa ... possui todos os direitos sobre a utilização dos resultados da análise, incluindo publicação de artigos e estatísticas, não podendo os mesmos serem utilizados sem prévia autorização por escrito.

**Parágrafo 4:** Do acesso aos dados e resultados:

Fica estabelecido que somente as pessoas envolvidas no processo de avaliação terão acesso aos dados e somente a diretoria da Empresa ..., juntamente com a equipe de avaliação terá acesso aos resultados, devendo esta decidir o que deverá ser repassados as demais áreas da empresa.

**Parágrafo 5:** Demais condições:

Fica estabelecido que quaisquer questões não regulamentadas neste acordo de confidencialidade devem ser tratadas entre a diretoria da Empresa ... e o Avaliador Qualificado, as resoluções documentadas e adendadas a este documento.

\_\_\_\_\_  
Patrocinador

\_\_\_\_\_  
Avaliador Qualificado

Figura A.3: Modelo do documento AR3 Entradas da avaliação.

Obs: O texto do documento deve ser alterado para cada avaliação.

A.3 TA2.1 Definir instâncias

<b>EMPRESA</b>	AR4 Instâncias de processos de software Local e data.
<b>INSTÂNCIAS DE PROCESSOS DE SOFTWARE</b>	
<b>Organização:</b>	
<b>Avaliador Qualificado:</b>	
<b>Projeto:</b>	
<b>Unidade Organizacional:</b>	
<b>Responsável:</b>	
<b>Tamanho:</b>	
<b>Domínio da Aplicação:</b>	
<b>Tamanho e complexidade dos produtos:</b>	
<b>Características de Qualidade:</b>	
<b>Recursos:</b>	
<b>Justificativa:</b>	
<b>Correspondência com o perfil da avaliação:</b>	
<hr/> <b>Avaliador Qualificado</b>	

Figura A.4: Modelo do documento AR4 Instâncias de processos de *software*.

#### A.4 TA2.2 Definir avaliadores

<b>EMPRESA</b>		AR5 Avaliadores
		Local e data.
<b>AVALIADORES</b>		
<b>Organização:</b>		
<b>Avaliador Qualificado:</b>		
<b>Avaliadores</b>	<b>Cargo</b>	
<b>Necessidades de treinamento</b>		
_____ Avaliador Qualificado		

Figura A.5: Modelo do documento AR5 Avaliadores.

## A.5 TA2.3 Treinar avaliadores

<b>EMPRESA</b>		AR6 Registro de treinamento			
		Local e data.			
<b>REGISTRO DE TREINAMENTO</b>					
<b>Organização:</b>					
<b>Avaliador Qualificado:</b>					
<b>Objetivo do treinamento:</b>					
<b>Conteúdo:</b>					
<b>Instrutor:</b>					
<b>Avaliadores:</b>					
<b>Data de Início:</b>					
<b>Data de término:</b>					
<b>Carga horária:</b>					
<b>Controle de frequência</b>					
<b>Avaliadores</b>					
<b>Comentários sobre o curso:</b>					
_____ Avaliador Qualificado					

Figura A.6: Modelo do documento AR6 Registro de treinamento.



**EMPRESA**

AR7 Planejamento da avaliação

Local e data

## PLANEJAMENTO DA AVALIAÇÃO

<b>Organização:</b>	
<b>Avaliador Qualificado:</b>	
<b>Avaliadores:</b>	

### Cronograma de Reuniões para Coletar Documentos

Assunto	Data / hora	Responsável	Avaliador

Página de

\_\_\_\_\_

Avaliador Qualificado

\_\_\_\_\_

Avaliadores

Figura A.7b: Modelo do documento AR7 Planejamento da avaliação.

**EMPRESA**

AR8 Checklist – práticas de gerenciamento

Local e data.

## CHECKLIST – PRÁTICAS DE GERENCIAMENTO

<b>Organização:</b>	
<b>Avaliadores:</b>	

<b>Projeto:</b>	
<b>Avaliação:</b>	
<b>Processo:</b>	
<b>Categoria de processo:</b>	
<b>Atributo:</b>	

<b>Prática de Gerenciamento</b>	<b>Descrição</b>	<b>Avaliação</b>	
		o-F   o-L	o-P   o-N
		o-F   o-L	o-P   o-N
		o-F   o-L	o-P   o-N
		o-F   o-L	o-P   o-N
<b>Evidências e observações:</b>			

\_\_\_\_\_  
Avaliador

Figura A.8: Modelo do documento AR8 Checklist - práticas de gerenciamento.

**EMPRESA**

AR8 Checklist – práticas básicas

Local e data

## CHECKLIST – PRÁTICAS BÁSICAS

<b>Organização:</b>	
<b>Avaliadores:</b>	

<b>Projeto:</b>	
<b>Avaliação:</b>	
<b>Processo:</b>	
<b>Categoria de processo:</b>	

<b>Prática Básica</b>	<b>Descrição</b>	<b>Avaliação</b>	
		o-F	o-L
		o-P	o-N
		o-F	o-L
		o-P	o-N
		o-F	o-L
		o-P	o-N
<b>Evidências e observações:</b>			

\_\_\_\_\_  
Avaliador

Figura A.9: Modelo do documento AR8 Checklist - práticas básicas.



**EMPRESA**

AR9 Lista de documentos

Local e data

**LISTA DE DOCUMENTOS**

<b>Organização:</b>	
<b>Avaliador Qualificado:</b>	
<b>Avaliadores:</b>	

**Conjunto de indicadores para a fase de levantamento de dados**

<b>Documentos</b>	<b>Correspondentes no ISO/IEC TR 15504</b>	
	<i>Entrada (input)</i>	<i>Saída (output)</i>

Página de

Figura A.10b: Modelo do documento AR9 Lista de documentos.

## A.8 TA2.6 Coletar informações

<b>EMPRESA</b>	AR8 Checklist – práticas de gerenciamento	Local e data
<b>CHECKLIST – PRÁTICAS DE GERENCIAMENTO</b>		
<b>Organização:</b>		
<b>Avaliadores:</b>		
<b>Projeto:</b>		
<b>Avaliação:</b>		
<b>Processo:</b>		
<b>Categoria de processo:</b>		
<b>Atributo:</b>		
<b>Prática de Gerenciamento</b>	Descrição	Avaliação
		v-F   o-L o-P   o-N
		o-F   v-L o-P   o-N
		o-F   v-L o-P   o-N
		v-F   o-L o-P   o-N
<b>Evidências e observações:</b>		

Figura A.11a: Modelo do documento AR8 Checklist – práticas de gerenciamento (respondido).

**EMPRESA**

AR8 Checklist – práticas básicas

Local e data.

## CHECKLIST – PRÁTICAS BÁSICAS

<b>Organização:</b>	
<b>Avaliadores:</b>	

<b>Projeto:</b>	
<b>Avaliação:</b>	
<b>Processo:</b>	
<b>Categoria de processo:</b>	

<b>Prática Básica</b>	<b>Descrição</b>	<b>Avaliação</b>
		o-F v-L o-P o-N
		o-F v-L o-P o-N
		v-F o-L o-P o-N
<b>Evidências e observações:</b>		

\_\_\_\_\_

Avaliador

Figura A.11b: Figura do documento AR8 Checklist – práticas básicas (respondido).

**A.9 TA2.8 Determinar medidas**

<b>EMPRESA</b>	AR10 Medidas das instâncias			
	Local e data.			
<b>MEDIDAS DAS INSTÂNCIAS</b>				
<b>Organização:</b>				
<b>Avaliador qualificado:</b>				
<b>Projeto:</b>				
<b>Avaliação:</b>				
<b>Processo:</b>				
<b>Categoria de processo:</b>				
	<b>No</b>	<b>Parcial</b>	<b>Large</b>	<b>Full</b>
<b>Práticas Básicas</b>				
<hr style="width: 30%; margin: 0 auto;"/> Avaliador Qualificado				

Figura A.12: modelo do documento AR10 Medidas das instâncias para práticas básicas.

**EMPRESA**

AR10 Medidas das instâncias

Local e data.

**MEDIDAS DAS INSTÂNCIAS**

<b>Organização:</b>	
<b>Avaliador qualificado:</b>	

<b>Projeto:</b>	
<b>Avaliação:</b>	
<b>Processo:</b>	
<b>Categoria de processo:</b>	

	<b>No</b>	<b>Parcial</b>	<b>Large</b>	<b>Full</b>
<b>Práticas de gerenciamento</b>				

\_\_\_\_\_  
Avaliador Qualificado

Figura A.13: Modelo do documento AR10 Medidas das instâncias para práticas de gerenciamento.

## A.10 TA2.9 Consolidar medidas

<b>EMPRESA</b>	AR11 Medidas do processo de software					
	Local e data.					
<b>MEDIDAS DO PROCESSO DE SOFTWARE</b>						
<b>Organização:</b>						
<b>Avaliador qualificado:</b>						
<b>Projeto:</b>						
<b>Avaliação:</b>						
<b>Processo:</b>						
	<b>Nível</b>					
<b>Instâncias de processo de software</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
_____ Avaliador Qualificado						

Figura A.14: Modelo do documento AR11 Medidas do processo de *software*.

## Anexo B – Programação das tarefas

Este anexo apresenta a programação em TCL/TK para as tarefas TA2.1 Definir instâncias, TA2.2 Definir avaliadores, TA2.3 Treinar avaliadores e TA2.4 Planejar avaliação, do estudo de caso apresentado no Capítulo 6.

### B.1 Programação da tarefa TA2.1 Definir instâncias

A programação da tarefa TA2.1 Definir instâncias (Definir Instâncias na Figura 6.17) representada pela janela work context Figura B.1.

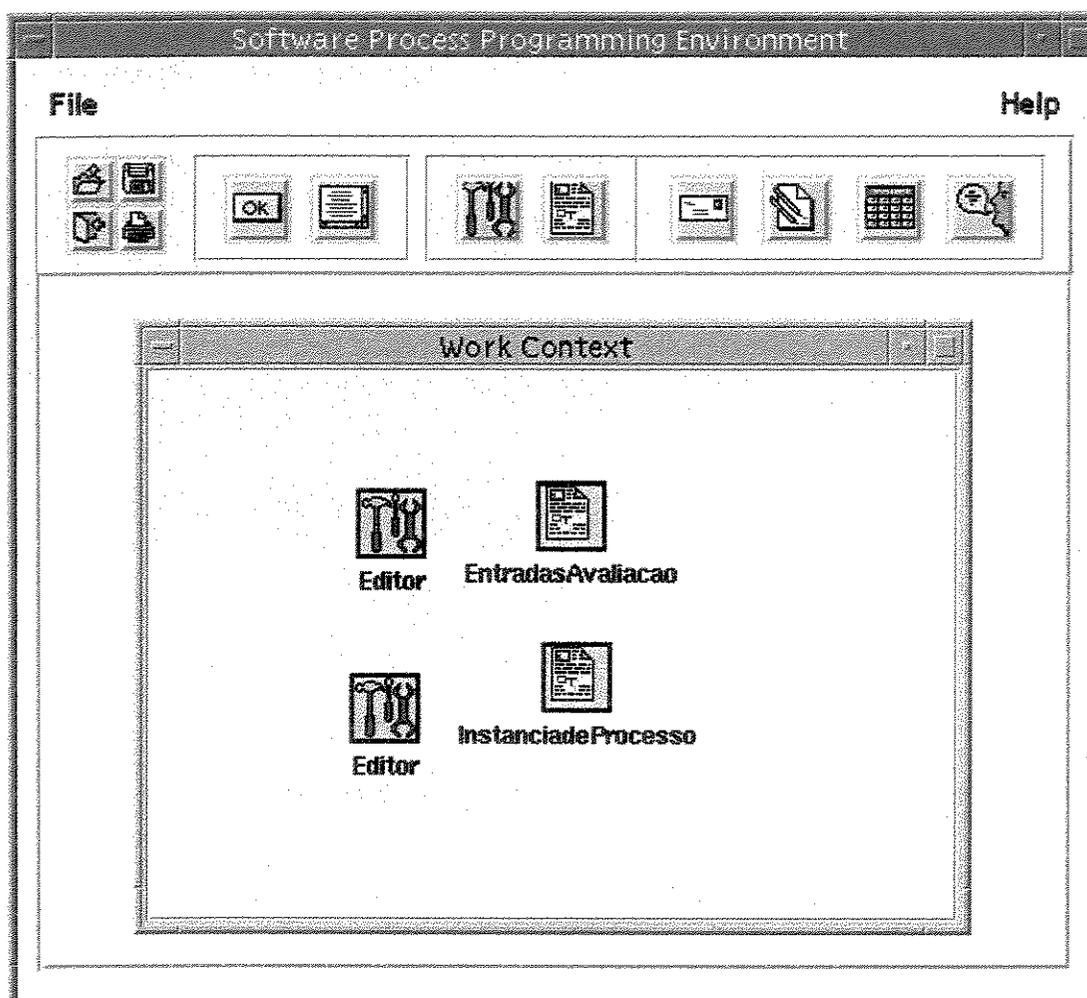


Figura B.1: Janela Work contex para tarefa TA2.1 Definir Instâncias.

A janela Work contex da Figura B.1 gerará o seguinte código fonte abaixo:

```

#!/bin/sh
#The next line restarts using tixshell \
exec /export/projetos/expsee/gerenciador/tixshell "$0" "$@"

#-----#
# ADEMIR MORGENSTERN PADILHA - 15/02/2000 - LES - UEM - MARINGA #
#-----#

#####
# METODOS DO GERENCIADOR A SEREM UTILIZADOS
#
#Estado do processo de software
#   int set_status_process_soft(char p[], int s);   " Define um
status para o processo de software
#   int get_status_process_soft(char p[]);         " Retorna o
status de um processo de software
#
#Estado de uma tarefa
#   int set_status_task(char p[], char t[], int s); " Define o
status para uma tarefa
#   int get_status_task(char p[], char t[]); " Retorna o status de
uma tarefa
#
#Estado do artefato
#   int set_status_artefact(char p[], char a[], int s);   " Define
o status para um artefato
#   int get_status_artefact(char p[], char a[]);         " Retorna
o status de um artefato

#-----Paths-----#

set ExpSEEHOME "/export/users/alunos/andreiac"

#####
# GLOBAL VARIABLES

global pre_con 0 # Pre-condicoes => 0 - falso 1 - verdadeiro
global pos_con 0 # Pos-condicoes => 0 - falso 1 - verdadeiro

#####
# USER DEFINED PROCEDURES

#-----inicializando variaveis

    set PSName "SPICE"
    set taskName "DefinirInstancias"
    set artefactNameIn "EntradasAvaliacao"
    set artefactNameOut "InstanciadeProcesso"

#-----testando as pre-condicoes

#-----do processo de software

    set status_Proc [get_status_process_soft $PSName]

```

```

if {$status_Proc == "0"} {
    puts stdout "\n Status do Processo de Software e "
    puts stdout $status_Proc
    exit
}

#-----da Tarefa

set status_Task [get_status_task $PSName $taskName]
if {$status_Task == "0"} {
    puts stdout "\n Status da Tarefa e "
    puts stdout $status_Task
    exit
}

#-----do Artefato de entrada

set status_Arte [get_status_artefact $PSName $artefactNameIn]
if {$status_Arte == "0"} {
    puts stdout "\n Status do Artefato de Entrada e "
    puts stdout $status_Arte
    exit
}

#-----do Artefato de saida

set status_Arte [get_status_artefact $PSName $artefactNameOut]
if {$status_Arte == "0"} {
    puts stdout "\n Status do Artefato de Saida e "
    puts stdout $status_Arte
    exit
}

#-----executando a tarefa

exec textedit
/export/users/alunos/andreiactcl/1999/tg/not_main_window/entradasavaliacao &
exec textedit
/export/users/alunos/andreiactcl/1999/tg/not_main_window/instanciadeproc
esso

set artefactStatus 1
set status_Arte [set_status_artefact $PSName $artefactNameOut
$artefactStatus]

#-----testando as pros-condicoes

#-----do Artefato

set status_Arte [get_status_artefact $PSName $artefactNameOut]
if {$status_Arte == "0"} {
    puts stdout "\n Execucao do processo ainda nao terminou \n"
}

```

## B.2 Programação da tarefa TA2.2 Definir avaliadores

A programação da tarefa TA2.2 Definir avaliadores (Definir Avaliadores na Figura 6.17) representada pela janela work context Figura B.2.

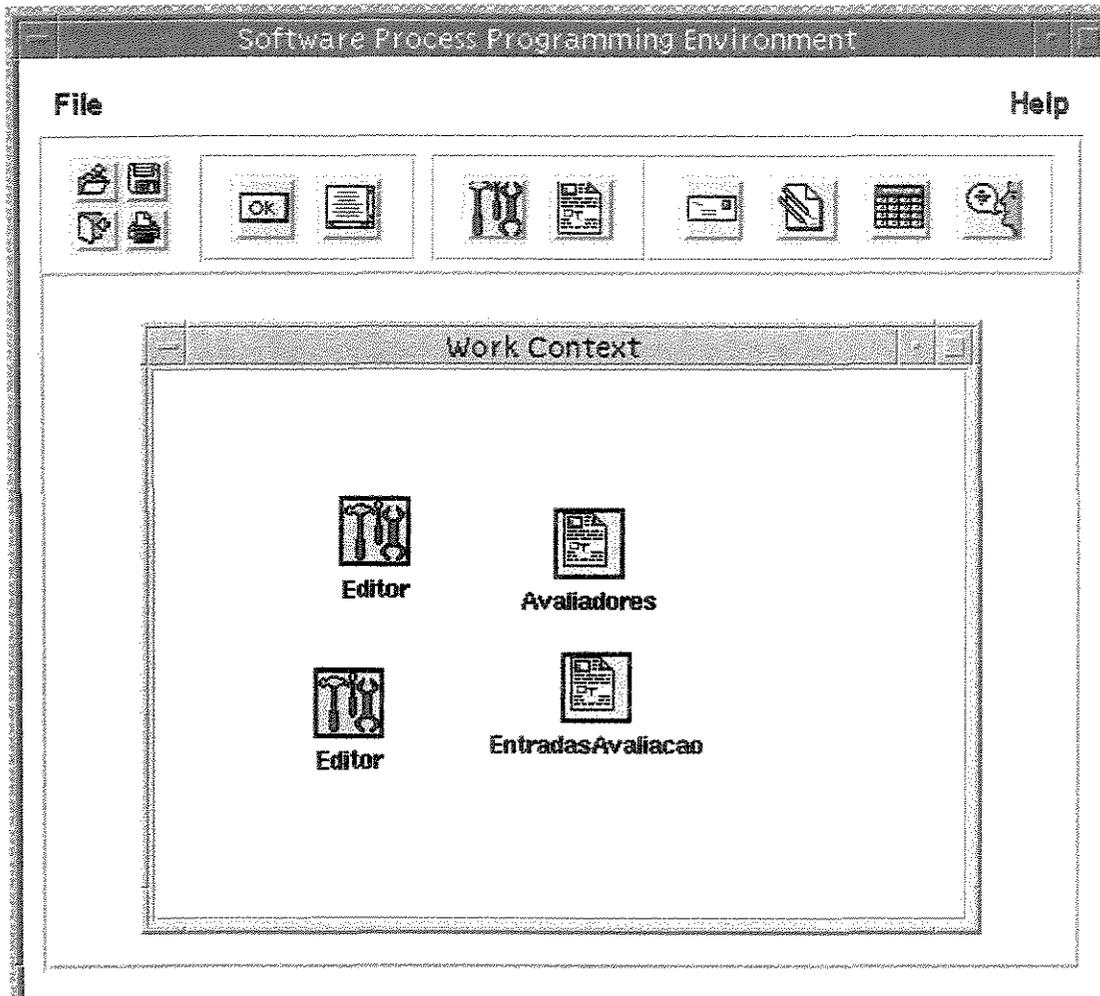


Figura B.2: Janela Work contex para tarefa TA2.2 Definir avaliadores.

A janela Work contex da Figura B.2 gerará o seguinte código fonte abaixo:

```
#!/bin/sh
#The next line restarts using tixshell \
exec /export/projetos/expsee/gerenciador/tixshell "$0" "$@"

#-----#
# ADEMIR MORGENSTERN PADILHA - 15/02/2000 - LES - UEM - MARINGA #
```

```

#-----#
#####
# METODOS DO GERENCIADOR A SEREM UTILIZADOS
#
#Estado do processo de software
#     int set_status_process_soft(char p[], int s);   " Define um
status para o processo de software
#     int get_status_process_soft(char p[]);         " Retorna o
status de um processo de software
#
#Estado de uma tarefa
#     int set_status_task(char p[], char t[], int s); " Define o
status para uma tarefa
#     int get_status_task(char p[], char t[]);       " Retorna o status de
uma tarefa
#
#Estado do artefato
#     int set_status_artefact(char p[], char a[], int s);   " Define
o status para um artefato
#     int get_status_artefact(char p[], char a[]);         " Retorna
o status de um artefato

#-----Paths-----#

set ExpPSEEHOME "/export/users/alunos/andreiac"

#####
# GLOBAL VARIABLES

global pre_con 0 # Pre-condicoes => 0 - falso 1 - verdadeiro
global pos_con 0 # Pos-condicoes => 0 - falso 1 - verdadeiro

#####
# USER DEFINED PROCEDURES

#-----inicializando variaveis

    set PSName "SPICE"
    set taskName "DefinirAvaliadores"
    set artefactNameIn "EntradasAvaliacao"
    set artefactNameOut "Avaliadores"

#-----testando as pre-condicoes

#-----do processo de software

    set status_Proc [get_status_process_soft $PSName]

    if {$status_Proc == "0"} {
        puts stdout "\n Status do Processo de Software e "
        puts stdout $status_Proc
        exit
    }

#-----da Tarefa

```

```

set status_Task [get_status_task $PSName $taskName]
if {$status_Task == "0"} {
    puts stdout "\n Status da Tarefa e "
    puts stdout $status_Task
    exit
}

#-----do Artefato de entrada

set status_Arte [get_status_artefact $PSName $artefactNameIn]
if {$status_Arte == "0"} {
    puts stdout "\n Status do Artefato de Entrada e "
    puts stdout $status_Arte
    exit
}

#-----do Artefato de saida

set status_Arte [get_status_artefact $PSName $artefactNameOut]
if {$status_Arte == "0"} {
    puts stdout "\n Status do Artefato de Saida e "
    puts stdout $status_Arte
    exit
}

#-----executando a tarefa

exec textedit
/export/users/alunos/andreiactcl/1999/tg/not_main_window/entradasavaliac
ao &
exec textedit
/export/users/alunos/andreiactcl/1999/tg/not_main_window/avaliadores

set artefactStatus 1
set status_Arte [set_status_artefact $PSName $artefactNameOut
$artefactStatus]

#-----testando as pros-condicoes

#-----do Artefato

set status_Arte [get_status_artefact $PSName $artefactNameOut]
if {$status_Arte == "0"} {
    puts stdout "\n Execucao do processo ainda nao terminou \n"
}

```

### **B.3 Programação da tarefa TA2.3 Treinar avaliadores**

A programação da tarefa TA2.3 Treinar avaliadores (Treinar Avaliadores na Figura 6.17) representada pela janela work context Figura B.3.

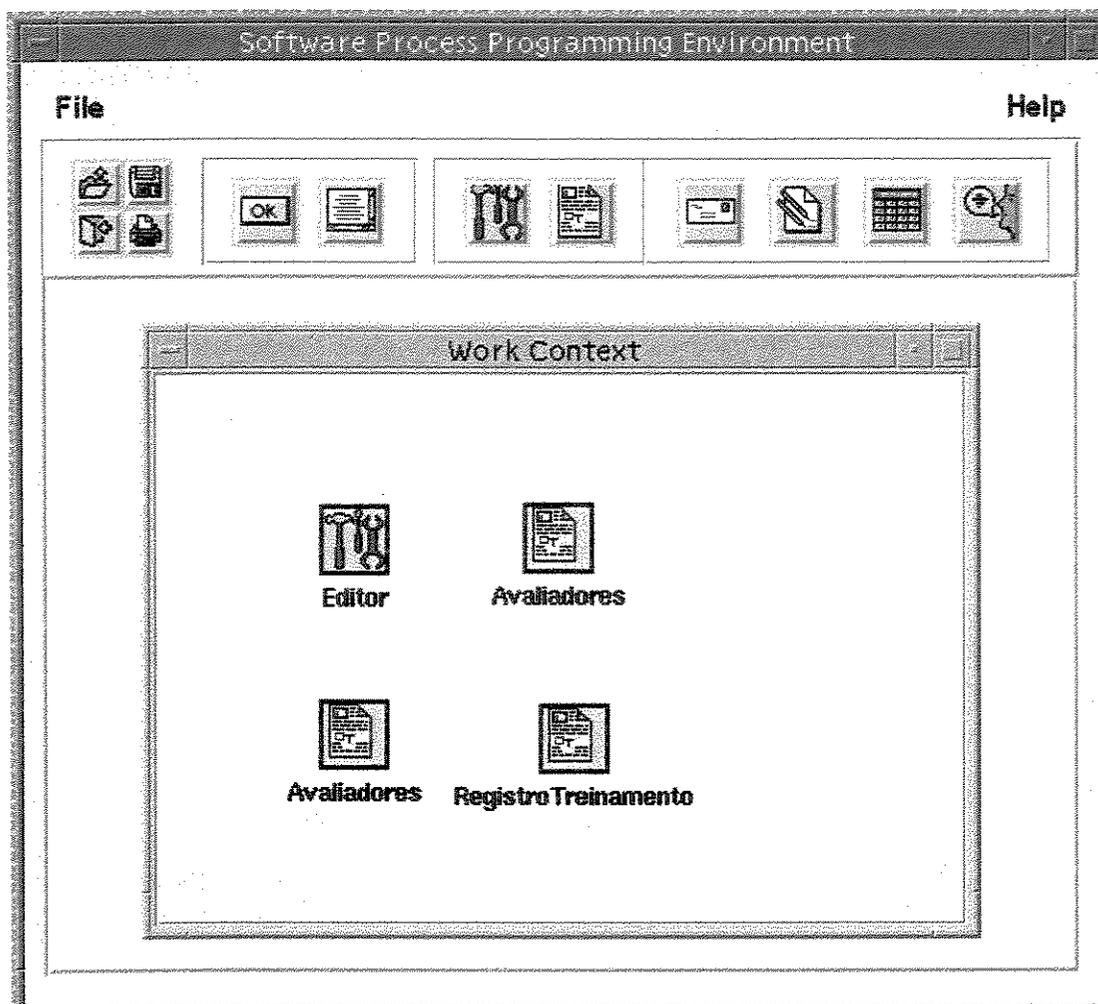


Figura B.3: Janela Work contex para tarefa TA2.3 Treinar avaliadores.

A janela Work contex da Figura B.3 gerará o seguinte código fonte abaixo:

```
#!/bin/sh
#The next line restarts using tixshell \
exec /export/projetos/expsee/gerenciador/tixshell "$@" "$@"

#-----#
# ADEMIR MORGENSTERN PADILHA - 15/02/2000 - LES - UEM - MARINGA #
#-----#

#####
# METODOS DO GERENCIADOR A SEREM UTILIZADOS
#
#Estado do processo de software
#   int set_status_process_soft(char p[], int s);   " Define um
status para o processo de software
#   int get_status_process_soft(char p[]);         " Retorna o
status de um processo de software
```

```

#
#Estado de uma tarefa
#   int set_status_task(char p[], char t[], int s); " Define o
status para uma tarefa
#   int get_status_task(char p[], char t[]); " Retorna o status de
uma tarefa
#
#Estado do artefato
#   int set_status_artefact(char p[], char a[], int s);   " Define
o status para um artefato
#   int get_status_artefact(char p[], char a[]);         " Retorna
o status de um artefato

#-----Paths-----#

set ExpPSEEHOME "/export/users/alunos/andreiac"

#####
# GLOBAL VARIABLES

global pre_con 0 # Pre-condicoes => 0 - falso 1 - verdadeiro
global pos_con 0 # Pos-condicoes => 0 - falso 1 - verdadeiro

#####
# USER DEFINED PROCEDURES

#-----inicializando variaveis

    set PSName "SPICE"
    set taskName "TreinarAvaliadores"
    set artefactNameIn "Avaliadores"
    set artefactNameOut "RegistroTreinamento"

#-----testando as pre-condicoes

#-----do processo de software

    set status_Proc [get_status_process_soft $PSName]

    if {$status_Proc == "0"} {
        puts stdout "\n Status do Processo de Software e "
        puts stdout $status_Proc
        exit
    }

#-----da Tarefa

    set status_Task [get_status_task $PSName $taskName]
    if {$status_Task == "0"} {
        puts stdout "\n Status da Tarefa e "
        puts stdout $status_Task
        exit
    }

#-----do Artefato de entrada

    set status_Arte [get_status_artefact $PSName $artefactNameIn]

```

```

if {$status_Arte == "0"} {
    puts stdout "\n Status do Artefato de Entrada e "
    puts stdout $status_Arte
    exit
}

#-----do Artefato de saida

    set status_Arte [get_status_artefact $PSName $artefactNameOut]
    if {$status_Arte == "0"} {
        puts stdout "\n Status do Artefato de Saida e "
        puts stdout $status_Arte
        exit
    }

#-----executando a tarefa

    exec textedit
/export/users/alunos/andreiad/tcl/1999/tg/not_main_window/avaliadores &
    exec textedit
/export/users/alunos/andreiad/tcl/1999/tg/not_main_window/registrodetrein
amento

    set artefactStatus 1
    set status_Arte [set_status_artefact $PSName $artefactNameOut
$artefactStatus]

#-----testando as pos-condicoes

#-----do Artefato

    set status_Arte [get_status_artefact $PSName $artefactNameOut]
    if {$status_Arte == "0"} {
        puts stdout "\n Execucao do processo ainda nao terminou \n"
    }

```

## **B.4 Programação da tarefa TA2.4 Planejar avaliação**

A programação da tarefa TA2.4 Planejar avaliação (Planejar Avaliação na Figura 6.17) é representada pela janela work context Figura B.4.

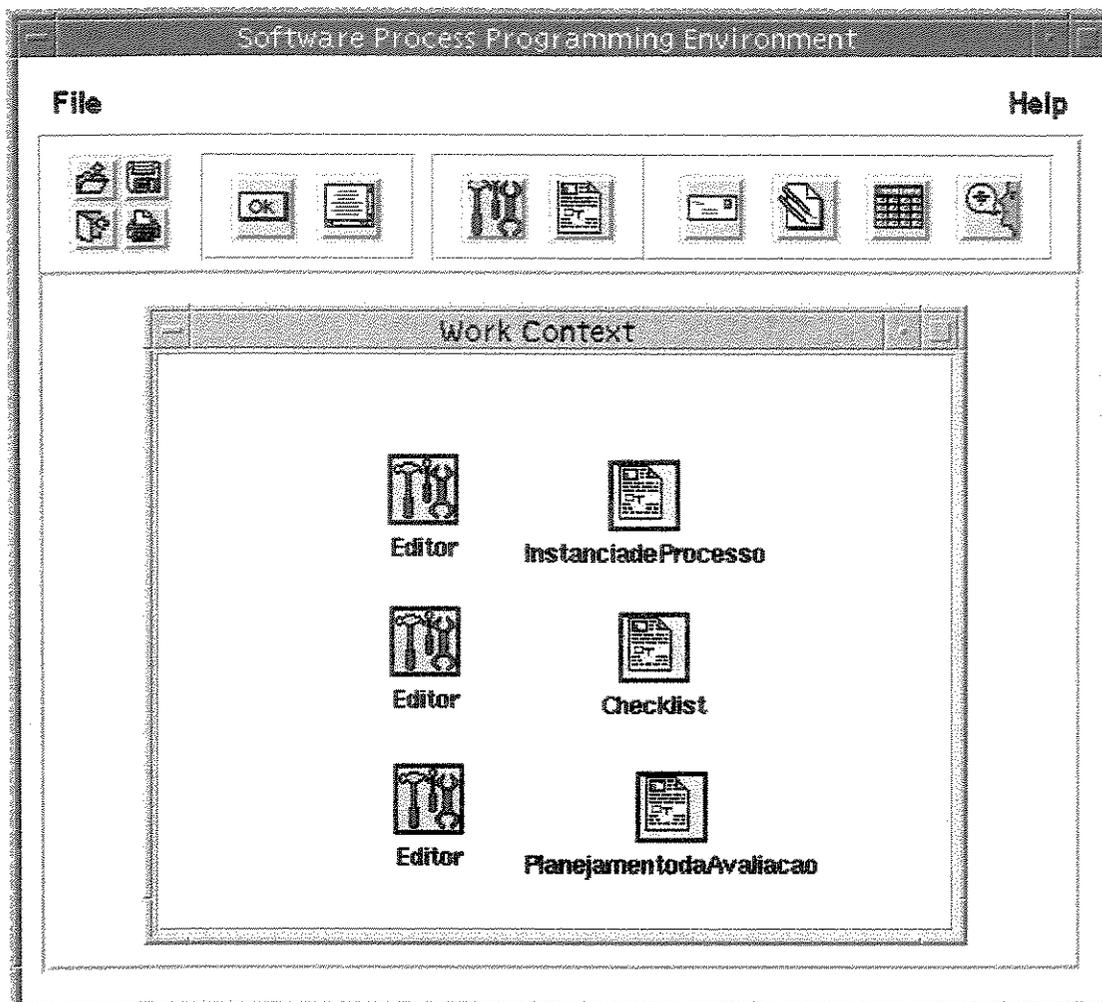


Figura B.4: Janela Work Contex para tarefa TA2.4 Planejar avaliação.

A janela *Workcontex* da Figura B.4 gerará o seguinte código fonte abaixo:

```
#!/bin/sh
#The next line restarts using tixshell \
exec /export/projetos/expsee/gerenciador/tixshell "$@"

#-----#
# ADEMIR MORGENSTERN PADILHA - 15/02/2000 - LES - UEM - MARINGA #
#-----#

#####
# METODOS DO GERENCIADOR A SEREM UTILIZADOS
#
#Estado do processo de software
#   int set_status_process_soft(char p[], int s);   " Define um
status para o processo de software
#   int get_status_process_soft(char p[]);         " Retorna o
status de um processo de software
```

```

#
#Estado de uma tarefa
#   int set_status_task(char p[], char t[], int s); " Define o
status para uma tarefa
#   int get_status_task(char p[], char t[]); " Retorna o status de
uma tarefa
#
#Estado do artefato
#   int set_status_artefact(char p[], char a[], int s);   " Define
o status para um artefato
#   int get_status_artefact(char p[], char a[]);         " Retorna
o status de um artefato

#-----Paths-----#

set ExpSEEHOME "/export/users/alunos/andreiac"

#####
# GLOBAL VARIABLES

global pre_con 0 # Pre-condicoes => 0 - falso 1 - verdadeiro
global pos_con 0 # Pos-condicoes => 0 - falso 1 - verdadeiro

#####
# USER DEFINED PROCEDURES

#-----inicializando variaveis

    set PSName "SPICE"
    set taskName "PlanejarAvaliacao"
    set artefactNameIn "InstanciadeProcesso"
    set artefactNameOut1 "PlanejamentodaAvaliacao"
    set artefactNameOut2 "Cheklist"

#-----testando as pre-condicoes

#-----do processo de software

    set status_Proc [get_status_process_soft $PSName]

    if {$status_Proc == "0"} {
        puts stdout "\n Status do Processo de Software e "
        puts stdout $status_Proc
        exit
    }

#-----da Tarefa

    set status_Task [get_status_task $PSName $taskName]
    if {$status_Task == "0"} {
        puts stdout "\n Status da Tarefa e "
        puts stdout $status_Task
        exit
    }

#-----do Artefato de entrada

```

```

set status_Arte [get_status_artefact $PSName $artefactNameIn]
if {$status_Arte == "0"} {
    puts stdout "\n Status do Artefato de Entrada e "
    puts stdout $status_Arte
    exit
}

#-----do Artefato de saida 1

set status_Arte [get_status_artefact $PSName $artefactNameOut1]
if {$status_Arte == "0"} {
    puts stdout "\n Status do Artefato de Saida e "
    puts stdout $status_Arte
    exit
}

#-----do Artefato de saida 2

set status_Arte [get_status_artefact $PSName $artefactNameOut2]
if {$status_Arte == "0"} {
    puts stdout "\n Status do Artefato de Saida e "
    puts stdout $status_Arte
    exit
}

#-----executando a tarefa

exec textedit
/export/users/alunos/andriac/tcl/1999/tg/not_main_window/entradasavaliac
ao &
exec textedit
/export/users/alunos/andriac/tcl/1999/tg/not_main_window/planejamentodaa
valiacao &
exec textedit
/export/users/alunos/andriac/tcl/1999/tg/not_main_window/checklist

set artefactStatus 1
set status_Arte [set_status_artefact $PSName $artefactNameOut1
$artefactStatus]
set status_Arte [set_status_artefact $PSName $artefactNameOut2
$artefactStatus]

#-----testando as pros-condicoes

#-----do Artefato

set status_Arte [get_status_artefact $PSName $artefactNameOut]
if {$status_Arte == "0"} {
    puts stdout "\n Execucao do processo ainda nao terminou \n"
}

```

## Anexo C – Descrição da execução no ExpPSEE

Este anexo detalha a aplicação do estudo de caso ao ambiente do ExpPSEE, conforme descrito no item 6.3. São apresentados nas seções abaixo os requisitos de cada tarefa segundo o ISO/IEC TR 15504 e sua execução no ambiente do ExpPSEE, de acordo com a definição da arquitetura do processo de avaliação (Seção 6.3.1), conforme a programação das tarefas (Seção 6.3.3 e Anexo C) e instanciação da arquitetura de processo (Seção 6.3.2).

### C.1 TA1.1 Descrever avaliação

Requisitos (ISO/IEC TR 15504):

- Patrocinador da avaliação (4.2.a);
- Propósito da avaliação (4.2.b);
- avaliador qualificado responsável pela avaliação, de acordo com os critérios definidos na parte 6 do ISO/IEC TR 15504, indicado pelo Sponsor (4.2.e, 4.3.P.1);

Execução no ExpPSEE:

- Através do gerenciador de cargos os tipos papel Patrocinador e Avaliador qualificado foram criados na ferramenta actor manager , inseridos na arquitetura do processo de avaliação pela ferramenta architecture builder e um ator alocado para cada um na ferramenta project manager.
- O propósito da avaliação, foi especificado no documento AR3 Entradas da Avaliação.
- Não foram gerados a janela WorkContext e sua programação para esta tarefa.

### C.2 TA1.2 Definir entradas da avaliação

Requisitos (ISO/IEC TR 15504):

- Definir a abrangência da avaliação incluindo (4.2.c):
- Os processos de *software* (categorias de processo) a serem investigados dentro da unidade organizacional (4.2.c.1);
- mais alto nível de capacidade a ser investigado para cada processo de *software* (4.2.c.2);
- A unidade organizacional que executa estes processos de *software* (4.2.c.3);
- contexto do processo de *software* que deve conter pelo menos (4.2.c.4):
- tamanho da unidade organizacional;
- Domínio de aplicação dos produtos ou serviços da unidade organizacional;
- tamanho, o quanto são críticos e a complexidade dos produtos e serviços;

- As características de qualidade dos produtos ou serviços (exemplo: ISO 9126).
- Definir as restrições, as quais podem incluir (4.2.d):
- Disponibilidade de recursos chaves (4.2.d.1);
- A quantidade máxima de tempo a ser usada na avaliação (4.2.d.2);
- tamanho mínimo, o tamanho máximo ou um exemplo de tamanho específico ou ainda o grau de cobertura que é desejada para avaliação (4.2.d.4);
- dono dos resultados da avaliação e as restrições em seu uso (4.2.d.5);
- Controle sobre as informações resultantes do acordo de confidencialidade (4.2.d.6);
- Qualquer informação adicional resultante do acordo de confidencialidade (4.2.d.7);
- Informações adicionais que devam ser coletadas durante o processo de avaliação (4.2.6).

Mapeamento para o ExpSSEE.:

- O propósito da avaliação, os processos de *software*, a metodologia e informações adicionais foram especificados no documento AR3 Entradas da Avaliação.
- O documento AR3 Entradas da avaliação contém o acordo de confidencialidade, o qual foi elaborado nesta tarefa.
- Não foram gerados a janela WorkContext e sua programação para esta tarefa.

### C.3 TA2.1 Definir instâncias

Requisitos (ISO/IEC TR 15504):

- As instâncias de processo de *software* específicos que devam ser excluídos do processo de avaliação (4.2.d.3);
- Quaisquer outras pessoas com responsabilidades específicas para suportar o processo de avaliação (4.2.5, 4.4.P.2);
- As instâncias de processos de *software* escolhidas dentro da unidade organizacional (4.4.P.5);
- Justificar a escolha (4.4.P.5);
- O perfil da avaliação para cada processo de *software* escolhido, de acordo com o modelo definido na parte 2 do ISO/IEC TR 15504 (4.4.P.3);
- A correspondência entre os processos organizacionais e o perfil da avaliação definido (4.4.P.4);
- Os responsáveis em cada unidade organizacional que possuam instâncias de processos de *software* selecionados para avaliação (4.4.P.9.b);

Mapeamento para o ExpSSEE:

- As instâncias de processo de *software* (projetos), seus responsáveis, abrangência e justificativas foram definidos no documento AR4 Instâncias de processos de *software*. Este documento (artefato) é editado pela ferramenta textedit na programação da tarefa conforme mostra o código fonte no anexo B.

- Na execução deste código fonte as pré-condições são testadas, ou seja, o processo de avaliação e a tarefa devem estar ativos, os artefatos de entrada finalizados e o ator disponível. Esses estados são checados através de chamadas ao gerenciador (Process Manager).

## **C.4 TA2.2 Definir avaliadores**

Requisitos (ISO/IEC TR 15504):

- Definir as responsabilidades específicas de cada membro da equipe de avaliação:
  - Definir conhecimentos necessários para a execução da avaliação (4.4.P.2);
  - Identificar necessidades de treinamento (4.4.P.2);

Mapeamento para o ExpSSEE:

- A equipe de avaliação é definida no documento AR5 Avaliadores. Este documento (artefato) é editado pela ferramenta textedit na programação da tarefa conforme mostra o código fonte no anexo B.
- Na execução deste código fonte as pré-condições são testadas, ou seja, o processo de avaliação e a tarefa devem estar ativos, os artefatos de entrada finalizados e o ator disponível. Esses estados são checados através de chamadas ao gerenciador (Process Manager).

## **C.5 TA2.3 Treinar avaliadores**

Requisitos (ISO/IEC TR 15504):

- Planejar o treinamento dos avaliadores (4.4.P.2).
- Executar o treinamento planejado para os avaliadores (4.4.P.2).

Mapeamento para o ExpSSEE:

- O registro do treinamento da equipe de avaliadores é realizado no documento AR6 Registro de treinamento. Este documento (artefato) é editado pela ferramenta textedit na programação da tarefa conforme mostra o código fonte no anexo B.
- Na execução deste código fonte as pré-condições são testadas, ou seja, o processo de avaliação e a tarefa devem estar ativos, os artefatos de entrada finalizados e o ator disponível. Esses estados são checados através de chamadas ao gerenciador (Process Manager).

## C.6 TA2.4 Planejar avaliação

Requisitos (ISO/IEC TR 15504):

- Definir cronograma detalhado de reuniões com os supervisores de cada unidade organizacional, juntamente com um cronograma detalhado para todo o processo de avaliação (4.4.P.9.b).
- Definir um conjunto de indicadores de processo de *software* para ajudar no julgamento dos avaliadores (4.4.P.8).

Mapeamento para o ExpSSEE:

- O cronograma geral do processo de avaliação e o cronograma detalhado são registrados no documento AR7 Planejamento do treinamento. Este documento (artefato) é editado pela ferramenta textedit na programação da tarefa conforme mostra o código fonte no anexo B.
- Os indicadores são definidos no documento AR8 Checklist. Este documento é editado pela ferramenta textedit na programação da tarefa conforme mostra o código fonte no anexo B.
- Na execução deste código fonte as pré-condições são testadas, ou seja, o processo de avaliação e a tarefa devem estar ativos, os artefatos de entrada finalizados e o ator disponível. Esses estados são checados através de chamadas ao gerenciador (Process Manager).

## C.7 TA2.5 Coletar documentos

Requisitos (ISO/IEC TR 15504):

- Reunir os documentos das instâncias de processos de *software* selecionados (4.4.P.5, 4.4.P.9.a);

Mapeamento para o ExpSSEE:

- A lista de documentos é registrada no documento AR9 Lista de documentos.
- Não foram gerados a janela WorkContext e sua programação para esta tarefa.

## C.8 TA2.6 Coletar informações

Requisitos (ISO/IEC TR 15504):

- Através da documentação apresentada, registrar evidências sobre o processo que está sendo avaliado (4.4.P.5, Nota 2);

- Entrevistar os Supervisores levantando informações, confirmando evidências e descobrindo novas, sobre as instâncias dos processos selecionadas (4.4.P.5);
- Registrar as evidências coletadas nas entrevistas e na documentação (4.4.P.5);
- Entrevistar os supervisores das unidades organizacionais com as quais as instâncias de processos de *software* selecionadas estejam envolvidas (4.4.P.5, 4.4.P.9.b).
- Justificar cada medida coletada de acordo com as evidências apontadas (4.4.P.9.c).

Mapeamento para o ExpSEE:

- O registro dos dados coletados é realizado no documento AR8 Checklist.
- Não foram gerados a janela WorkContext e sua programação para esta tarefa.

## C.9 TA2.7 Verificar informações

Requisitos (ISO/IEC TR 15504):

- Checar as evidências coletadas na tarefa anterior, procurando inconsistências entre o que foi documentado e o que foi afirmado nas entrevistas (4.4.P.9);
- Confirmar evidências encontradas na documentação (4.4.P.9);
- Confirmar práticas que estão sendo executadas, mas não possuem evidências que as comprovem (4.4.P.9);
- Se necessário, marcar novas entrevistas com os supervisores para esclarecer dúvidas (4.4.P.9).

Mapeamento para o ExpSEE:

- O registro da verificação dos dados coletados é realizado no documento AR8 Checklist.
- Não foram gerados a janela WorkContext e sua programação para esta tarefa.

## C.10 TA2.8 Determinar medidas

Requisitos (ISO/IEC TR 15504):

- De acordo com as evidências registradas, estabelecer o perfil de cada instância de processo de *software* conforme modelo definido na parte 2 do ISO/IEC TR 15504 (4.4.P.6);

Mapeamento para o ExpSEE:

- O perfil de cada instância de processo é realizado no documento AR10 Medidas das instâncias de processo de *software*.
- Não foram gerados a janela WorkContext e sua programação para esta tarefa.

## C.11 TA2.9 Consolidar medidas

Requisitos (ISO/IEC TR 15504):

- Reunir todas as medições estabelecidas para cada perfil de processo e unificá-las resultando no perfil da avaliação, conforme modelo definido na parte 2 do ISO/IEC TR 15504 (4.4.P.7);

Mapeamento para o ExpPSEE:

- A consolidação das medidas dos processos de *software* é realizada no documento AR11 Medidas do processo de *software*.
- Não foram gerados a janela WorkContext e sua programação para esta tarefa.

## C.12 TA2.10 Validar medidas

Requisitos (ISO/IEC TR 15504):

- Verificar se o perfil da avaliação atende o que foi determinado no objetivo da avaliação descrito no documento Descrição do processo de avaliação (4.3.P.2, 4.4.P.1).

Mapeamento para o ExpPSEE:

- A validação das medidas dos processos de *software* é realizada no documento AR11 Medidas do processo de *software*.
- Não foram gerados a janela WorkContext e sua programação para esta tarefa.

## C.13 TA3 Apresentar resultados

Requisitos (ISO/IEC TR 15504):

- Apresentar o perfil da avaliação de acordo com a determinação do objetivo da avaliação descrito no documento descrição do processo de avaliação e com o Acordo de confidencialidade, devendo conter no mínimo (4.5):
  - A data da avaliação;
  - As entradas da avaliação;
  - As instâncias de processo de *software* selecionadas e a justificativa da seleção;

- A abordagem utilizada na avaliação, incluindo a identificação do modelo de avaliação, método e qualquer instrumento de suporte ou ferramentas para suportar reclamações.
- Os nomes dos membros da equipe que conduziram a avaliação, incluindo o avaliador qualificado responsável pela avaliação e a equipe de avaliadores.
- O conjunto de perfis de processo de *software* resultantes da avaliação.
- A localização de registros que suportam evidências objetivas, as medidas dos atributos de processo nos perfis de processo.
- Informações adicionais coletadas durante a avaliação que foram identificadas nas entradas da avaliação.

#### Mapeamento para o ExpSEE:

- Como foi explanado na Seção 5.2 do Capítulo 5, esta tarefa está fora do escopo do processo de avaliação.
- Não foram gerados a janela WorkContext e sua programação para esta tarefa.

## Anexo D – Artigo aceito no WQS2000

### MAPS-15504

#### Uma metodologia de avaliação de processo de *software* para o ambiente ExpSEE baseada no ISO/IEC TR 15504

Itana M. S. Gimenes  
([itana@din.uem.br](mailto:itana@din.uem.br))

Departamento de Informática  
UEM

Ademir Morgenstem Padilha  
([mpadilha@onda.com.br](mailto:mpadilha@onda.com.br))

Departamento de Informática  
CESULON

Jacques Wainer  
([wainer@ic.unicamp.br](mailto:wainer@ic.unicamp.br))

Instituto de Computação  
UNICAMP

### Abstract

The increasing requirements for *software* quality has lead the *software* engineering community to produce several standards and norms to guide the quality of *software* products and processes. Amongst the process standards, it can be highlighted the following ones due to their wide use: ISO 9000-3, ISSO 12207, CMM and ISO/IEC TR 15504. An additional important research results of the *software* engineering community are Process-centred *software* engineering environments (PSEE). These environments aim at providing integrated *software* process automation. Process standards and PSEE compose the central subject of the work presented in this paper that is the evaluation of the quality of *software* processes through an automated environment. In order to achieve this a method of *software* process evaluation, called MAPS-15504, based on the ISO/IEC TR 15504, was developed. The MAPS-15504 was applied to a case study and implemented in the ExpSEE environment. ExpSEE is an experimental PSEE developed in the Informatics Department (DIN) of the Universidade Estadual de Maringá (UEM).

**Word keys:** SPICE, Software process, evaluation and *software* process improvement, methods and tools for quality control, Quality evaluation techniques.

### Resumo

Devido as crescentes exigências por qualidade de *software* a comunidade de engenharia de *software* tem produzido diversas normas e apresentado diversas abordagens sobre a qualidade de *software*. Grande parte destas normas são aplicadas ao processo de *software*, dentre elas destacam-se pela larga utilização a ISO 9000-3, a ISO 12207, o CMM

e o ISO/IEC TR 15504 como resultado dos trabalhos do projeto SPICE. Outro resultado das pesquisas da comunidade de engenharia de *software* são os ambientes de engenharia de *software* centrados no processo (PSEE), os quais visam a automação do processo de *software*. Estas duas idéias, normatização e automação do processo de *software*, compõem a idéia central deste trabalho: avaliação da qualidade do processo de *software* através de um ambiente automatizado. Para isto foi desenvolvida uma metodologia de avaliação de processo de *software* com base no ISO/IEC TR 15504, denominada MAPS-15504. A metodologia de avaliação de processo de *software* foi implementada no ambiente do ExPSEE, um ambiente experimental desenvolvido no Departamento de Informática (DIN) da Universidade Estadual de Maringá (UEM).

**Palavras chaves:** SPICE, Processo de *software*, Avaliação e Melhorias em Processos, Métodos e Ferramentas de Controle de Qualidade, Técnicas de Avaliação da Qualidade.

## 1. Introdução

A busca pela qualidade tem sido objeto de intensas pesquisas pela comunidade de engenharia de *software*. Essas pesquisas resultaram em um conjunto de padrões, os quais possuem dois enfoques distintos: o produto e o processo de *software*. Neste trabalho, enfocaremos o processo de *software*.

Com a necessidade de um maior conhecimento e controle do processo de *software*, pela sofisticação dos meios de produção (ex.: metodologias, ambientes e ferramentas), em 1991, o comitê de engenharia de *software* da ISO<sup>25</sup> aprovou a realização de estudos para analisar as necessidades de um padrão para avaliação do processo de desenvolvimento de *software* [1]. Foi a partir deste estudo que criou-se em 1993, o projeto SPICE<sup>26</sup>. O objetivo principal deste projeto é produzir normas que orientem a avaliação de processos de *software* visando a melhoria contínua do processo e a determinação da sua capacidade.

O processo de *software* pode ser definido como sendo o conjunto de todas as atividades relacionadas ao desenvolvimento, controle, validação e manutenção de um *software* operacional [4].

---

<sup>25</sup> ISO - International Organization for Standardization

<sup>26</sup> SPICE - Software Process Improvement and Capability dEtermination

A idéia de definição e execução das atividades do processo de *software* foi proposta por Osterweil em [5]. Na última década, essas idéias se disseminaram, estabelecendo-se a necessidade do suporte não só aos objetos gerados durante o desenvolvimento de *software* através de ferramentas CASE<sup>27</sup>, mas também à definição e controle do próprio processo de *software*. Surgiram, então os ambientes de engenharia de *software* orientados a processo (PSEE<sup>28</sup>). Esses ambientes constituem-se de mecanismos que oferecem suporte automatizado e integrado as atividades de engenharia de *software* [4].

As duas áreas descritas acima, a automação de processo de *software* (ex.: PSEE) e normalização da avaliação do processo de *software* (ex.: ISO/IEC TR 15504), tem propósitos complementares na obtenção da qualidade de *software*.

Este artigo apresenta uma metodologia de avaliação de processo de *software*, denominada MAPS-15504. O ExpPSEE, um PSEE experimental [6], foi utilizado para apoiar a automação da metodologia proposta Neste artigo apresentamos na Seção 2 uma descrição da metodologia proposta, a MAPS 15504, e nas subseções um detalhamento destas atividades. Após esta descrição, apresentamos as conclusões.

## 2. Descrição da MAPS-15504

Antes da descrição das tarefas que compõem a Metodologia de Avaliação de Processo de Software – MAPS-15504, é apresentada a notação escolhida para representação da nomenclatura das tarefas, artefatos, agentes e condições. Esses elementos representam o processo de *software* e a notação gráfica utilizada foi baseada no projeto PRONET descrito em [7]:

- TA1 - representa a tarefa de número 1. Caso ela seja composta de outras tarefas, serão acrescentados tantos subitens quantos sejam os níveis que ela representa. Por exemplo: TA1.1 significa a primeira tarefa que compõe a tarefa TA1.
- AR1 - representa o artefato de número 1.
- AG1 - representa o agente número 1.
- CO1 - representa a condição número 1.

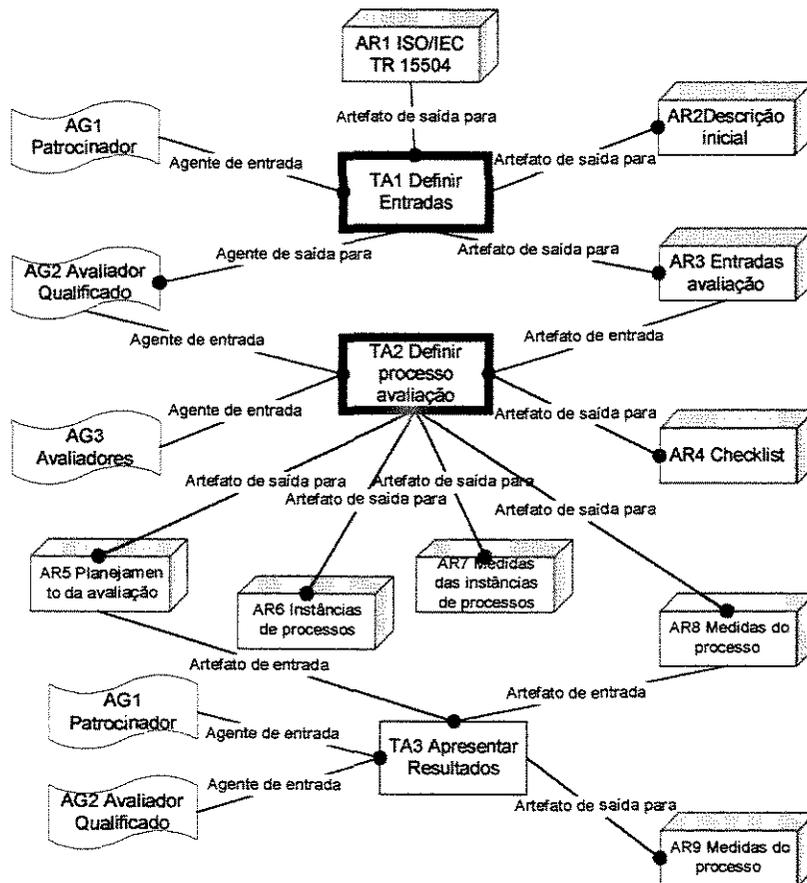
---

<sup>27</sup> CASE – Computer Aided Software Engineering

<sup>28</sup> PSEE – Process Software Engineering Environment

O processo de avaliação de processo de *software* proposto na Figura 1, é composto por três tarefas distintas: TA1 Definir Entradas, TA2 Definir Processo de Avaliação e TA3 Apresentar Resultados. A metodologia proposta neste trabalho está centrada nas tarefas que compõem a tarefa TA2 Definir Processo de Avaliação. As tarefas TA1 Definir Entradas e TA2 Apresentar Resultados estão descritas para consistência da metodologia.

Figura 1: Arquitetura geral do processo de avaliação.



## 2.1 A tarefa TA1 Definir Entradas

A tarefa TA1 Definir Entradas é composta de duas tarefas, conforme mostra a Figura 2. A tarefa TA1.1 Descrever Avaliação é de responsabilidade da organização no papel do AG1 Patrocinador e precede o processo de avaliação propriamente dito. Nessa tarefa deve-se descrever no documento AR2 Descrição Inicial, o propósito da avaliação, as unidades organizacionais a serem avaliadas (setores ou departamentos) e contratar o AG2 Avaliador Qualificado. O avaliador qualificado pode ser uma segunda parte contratada ou

um funcionário treinado pertencente a organização especificamente designado para esta função. A parte 6 do ISO/IEC TR 15504 [1] define os requisitos para um avaliador qualificado.

A partir das informações da tarefa TA1.1 Descrever Avaliação, o avaliador qualificado fará a descrição do processo de avaliação conforme descrito na tarefa TA1.2 Definir Entradas. O propósito da avaliação descrito no documento AR2 Descrição Inicial deve ser revisado e transcrito no documento AR3 Entradas da Avaliação.

Deve ser elaborado entre o patrocinador e o avaliador qualificado, um acordo de confidencialidade no qual os direitos sobre a avaliação e seus dados estão descritos e assinados por ambas as partes. Este acordo faz parte do documento AR2 Descrição Inicial.

Após a tarefa TA1.2 Definir Entradas é executado uma composição DA (*Divergente and*). Significa que assim que a tarefa TA1.2 Definir Entradas termine, outras tarefas iniciarão simultaneamente, que são as tarefas TA2.1 Definir Instâncias e TA 2.2 Definir Avaliadores.

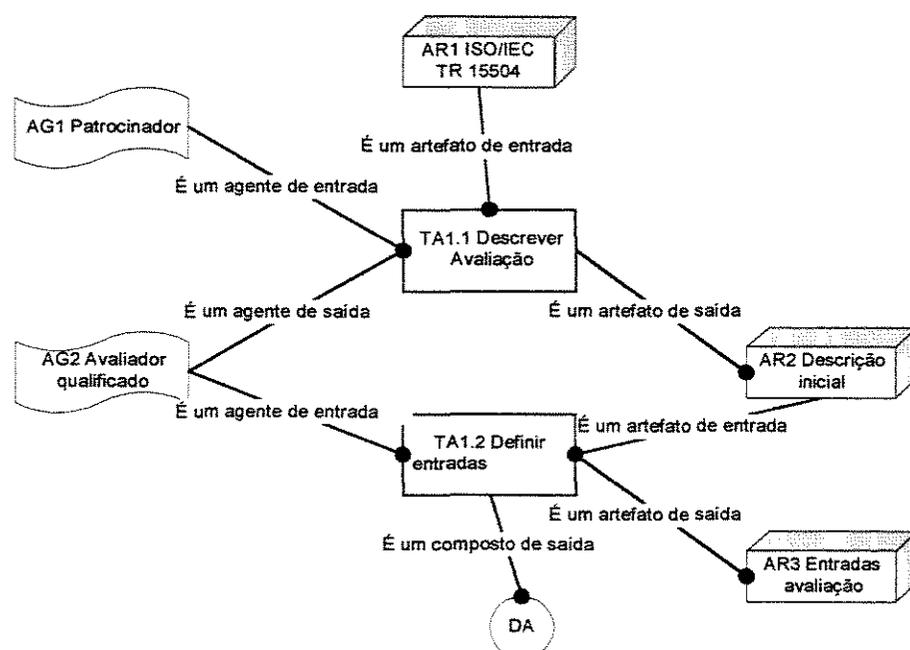


Figura 2 - Tarefas que compõem a tarefa TA1 Definir Entradas

## **2.2 A tarefa TA2 Definir Processo de Avaliação**

A Figura 3 representa graficamente o processo de avaliação de processo de *software*.

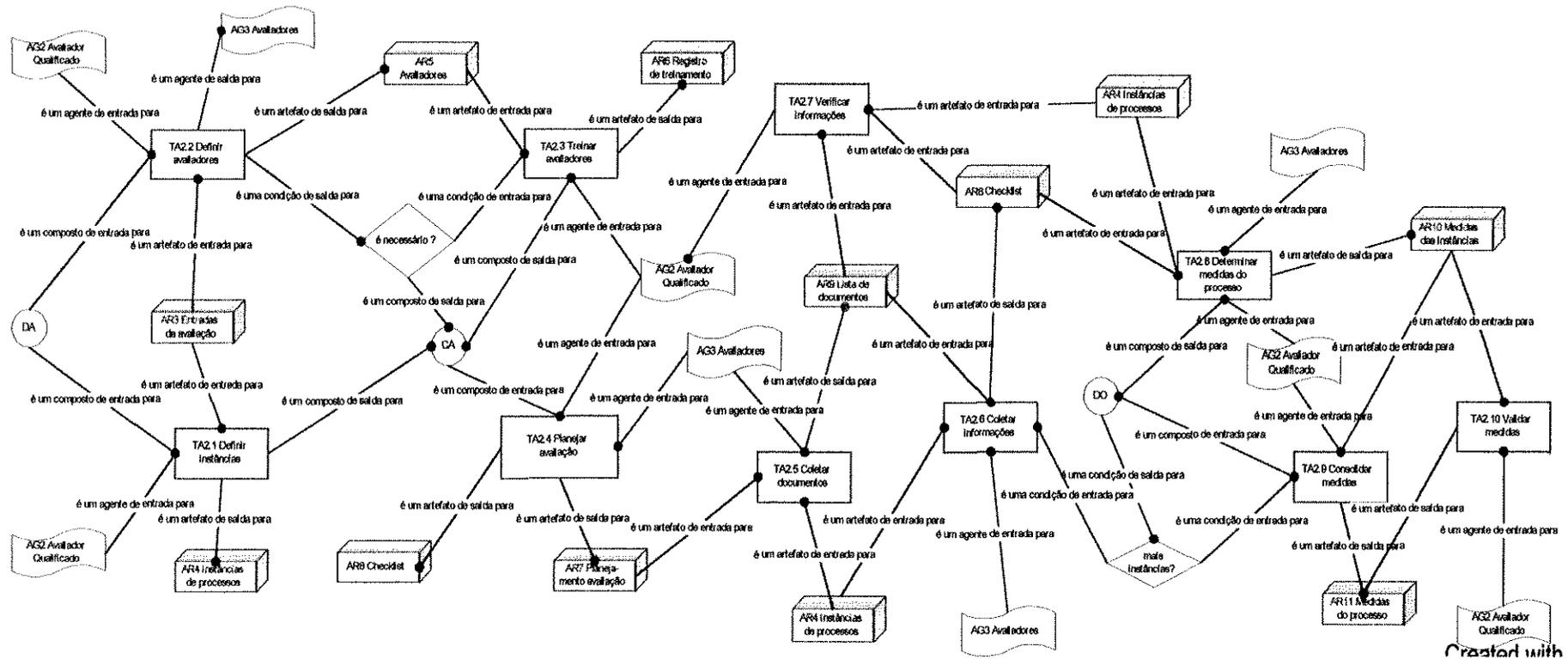


Figura 3 – Tarefas que compõem a tarefa TA2 Processo de avaliação

A partir das entradas da avaliação, através da composição (elemento de sincronização) DA - *Divergente and*, são iniciadas as tarefas TA2.1 Definir Instâncias e TA2.2 Definir Avaliadores. A tarefa TA2.1 Definir Instâncias deve escolher os projetos da organização a partir dos quais possa-se ter uma visão completa do processo de *software*, respeitados os limites definidos no documento AR2 Entradas da Avaliação. Estas instâncias de processo devem ser registradas em um documento denominado AR3 Instâncias de processo.

A tarefa TA2.2 Definir Avaliadores, deve ser executada independentemente da tarefa TA2.1 Definir Instâncias. Esses avaliadores podem ser externos a organização a exemplo do avaliador qualificado, porém avaliadores internos também devem ser selecionados pois os mesmos conhecem a organização (setores, departamentos, chefias, cultura) e essas informações facilitarão a condução do processo de avaliação. A descrição desses avaliadores, seus papéis e sua formação devem ser registrados em um documento denominado AR4 Avaliadores. Caso o grupo de avaliadores não tenha conhecimento do ISO/IEC TR 15504 e da metodologia de avaliação a ser empregada, treinamento adequado deve ser ministrado. O AG2 Avaliador Qualificado deve estar capacitado para ministrar este treinamento, o qual deve ser registrado em um documento denominado AR5 Registro de Treinamento.

A tarefa TA2.4 Planejar Avaliação só poderá ser iniciada uma vez terminadas as duas tarefas TA2.1 Definir Instâncias e TA2.2 Definir Avaliadores. Essa condição é definida pela composição CA – *Convergente and*. Toda a equipe deve reunir-se sob a coordenação do avaliador qualificado e elaborar o planejamento da avaliação conforme definido pela tarefa TA2.4 Planejar Avaliação. Esse planejamento deve ser registrado em um documento denominado AR6 Planejamento da Avaliação e consiste basicamente de cronogramas de reuniões: data, hora, local, participantes e a pauta das reuniões. Nessa atividade também será elaborado o documento AR7 Checklist que será utilizado para coleta de dados. Para as avaliações que utilizam apenas o ISO/IEC TR 15504 as práticas base e as práticas de gerenciamento são utilizadas como *checklist*. Outros modelos compatíveis podem ser utilizados como guias auxiliares na avaliação, conforme definido na parte 4 do ISO/IEC TR 15504.

A tarefa TA2.5 Coletar Documentos deve seguir o documento AR6 Planejamento da Avaliação, com base no documento AR3 Instâncias de processo. Os documentos gerados ou utilizados nas instâncias de processo selecionados, devem ser coletados de maneira a gerar evidências que serão avaliados segundo as práticas base descritas no modelo de referência, Parte 2 do ISO/IEC TR 15504. As práticas de gerenciamento serão coletadas nas entrevistas e obtidas através das impressões dos avaliadores coletadas ao longo do processo de avaliação. Os documentos selecionados devem ser registrados em um documento denominado AR9 Lista de Documentos. Essa lista de documentos deve conter também um mapeamento mostrando os documentos equivalentes definidos no anexo C da parte 5 do ISO/IEC TR 15504. Esses documentos equivalentes fornecem uma série de características desejáveis, mas não obrigatórias, que podem ser utilizadas como indicadores para o avaliador.

De posse do documento AR3 Instâncias de Processo e do documento AR9 Lista de Documentos, deve-se coletar informações juntamente aos responsáveis pelos projetos, nas unidades organizacionais onde são desenvolvidos. Este é um processo que se repete até que todas as instâncias de processos sejam verificadas e todas as dúvidas dos avaliadores sejam esclarecidas. Durante a execução da avaliação o AR7 Checklist é preenchido e as evidências são verificadas junto aos documentos coletados. As práticas de gerenciamento são confirmadas pelas impressões dos avaliadores coletadas durante a avaliação. Uma vez verificadas as informações, as instâncias de processos são medidas segundo as definições do modelo de referência na parte 2 do ISO/IEC TR 15504 e registradas em um documento denominado AR10 Medidas das Instâncias. Caso haja mais instâncias a serem avaliadas, a decisão deve seguir o caminho de volta a tarefa TA2.6 Coletar Informações. Caso não haja mais informações a serem coletadas, a tarefa TA2.9 Consolidar Medidas deve ser executada.

Na tarefa TA2.9 Consolidar Medidas, as medidas de todas as instâncias de processo devem ser reunidas dando o perfil do processo da organização. As medidas dos processos devem ser registradas no documento AR11 Medidas do Processo. Estas medidas são validadas pelo avaliador qualificado e uma vez confirmadas finalizarão o processo de avaliação, obtendo-se os perfis que devem atender ao propósito da avaliação contido no

documento AR2 Entradas da Avaliação. Após essa tarefa, o processo de avaliação é finalizado e segue-se a tarefa TA3 Apresentar Resultados.

## 2.3 A tarefa TA3 Apresentar Resultados

Esta tarefa sucede todo o processo de avaliação. De acordo com o ISO/IEC TR 15504, os resultados obtidos devem ser apresentados, respeitando-se o acordo de confidencialidade. Os resultados devem apresentar o perfil da avaliação de acordo com a determinação dos objetivos da avaliação e com o acordo de confidencialidade descritos no documento AR2 Descrição Inicial.

## 3. Conclusões

A partir de um estudo mais detalhado do ISO/IEC TR 15504, concluiu-se que o mesmo não tem por objetivo apresentar uma metodologia de avaliação de processo, mas apresenta uma estrutura (*framework*) para sua avaliação. Portanto, este trabalho definiu um conjunto de tarefas e formulários para a avaliação de processos de *software* que atendessem aos critérios da parte 3 do ISO/IEC TR 15504.

A metodologia proposta além de ser uma contribuição que o ISO/IEC TR 15504 não se propõe a fornecer, contribui para o processo de avaliação completando algumas lacunas as quais não estão definidas no *framework* do ISO/IEC TR 15504:

- introduz duas tarefas para a definição das entradas da avaliação: TA1.1 Descrição inicial e TA1.2 Definir entradas, contribuindo para a estruturação do processo de avaliação;
- introduz a tarefa TA2.5 Coletar documentos o que traz ganhos de tempo e qualidade da avaliação;
- define um processo de avaliação claro e passível de automação;
- introduz o gerenciamento automatizado do processo através do controle sobre o andamento e o estado de cada tarefa a qualquer tempo;
- permite a reutilização da arquitetura da avaliação;
- garante a execução da tarefa somente se ela puder ser executada através de pré e pós-condições;

- automatiza os *checklists* que podem ser programados na tarefa;
- gera automaticamente os resultados da avaliação;
- controla o acesso aos resultados da avaliação.

A automatização das tarefas traz grandes benefícios para o processo de avaliação. A definição e validação do processo antes de sua execução, permitindo a alocação de todos os recursos necessários para a execução da avaliação, proporcionam maior entendimento, clareza e facilitam o planejamento da avaliação.

A definição do perfil da avaliação em formulários próprios de acordo com os requisitos do ISO/IEC TR 15504 e sua automação na programação das tarefas, permite não só o atendimento destes requisitos, como também a garantia de execução das tarefas através de pré e pós condições que são verificadas via programação. Uma vez as tarefas programadas, o ambiente passa a controlar a agenda dos avaliadores, com a possibilidade de agregação de outras ferramentas como vídeo conferência, *e-mail*, dentre outras.

A programação das tarefas, além da simples automação de formulários deve atender a requisitos específicos da avaliação de processos, como o suporte automatizado a *checklists* e geração automática dos perfis das instâncias de processo e do processo de *software* como um todo. O suporte automatizado poderá melhorar a qualidade das informações que serão analisadas, validando a entrada de dados.

A Tabela 1 apresenta uma comparação da MAPS-15504 programada no ambiente do ExpSEE com a ferramenta SEAL<sup>29</sup>. [8]. A ferramenta SEAL foi escolhida por ser um produto de *software* específico para a avaliação segundo o ISO/IEC TR 15504. Esta ferramenta foi a mais utilizada durante a segunda fase dos *trials* do projeto SPICE, conforme descrito em [9].

Tabela 1: Características de uma ferramenta para automação do processo de *software*.

Característica	SEAL	ExpSEE
Definição e validação do processo de avaliação segundo uma metodologia compatível com a ISO/IEC TR 15504.		X
Permitir a alocação dos recursos necessários para execução da metodologia.		X

<sup>29</sup> SEAL : Software Engineering Applications Laboratory – Laboratório de Aplicações em Engenharia de Software.

Permitir a definição do perfil da avaliação segundo as definições da ISO/IEC TR 15504.	X	X
Permitir a definição da estrutura da documentação necessária (artefatos) para a execução da avaliação.		X
Fornecer suporte a uma agenda de tarefas para múltiplos avaliadores permitindo assim um melhor gerenciamento do processo.		X
Automatizar aos <i>checklists</i> utilizados durante a avaliação de acordo com o perfil da avaliação elaborado.	X	X
Verificar os formulários respondidos quanto a sua completitude.		X
Gerar de maneira automática os perfis das instâncias de processo de <i>software</i> avaliados.	X	X
Gerar de maneira automática o perfil do processo de <i>software</i> da organização.	X	X
Disponibilizar os resultados.	X	X

#### 4. Bibliografia

- [1] ISO/IEC/JTC1/SC7/WG10, **ISO/IEC TR 15504 - Software Process Improvement and Capability dEtermination - Technical Report**, 1999.
- [2] ISO/IEC/JTC1/SC7/WG10, **ISO/IEC TR 15504 - Software Process Improvement and Capability dEtermination - Technical Report**, Parte 2 – Um modelo de referência para processos e capacidade de processos, 1999.
- [3] ISO/IEC/JTC1/SC7/WG10, **ISO/IEC TR 15504 - Software Process Improvement and Capability dEtermination - Technical Report**, Parte 3 – Execução de avaliação, 1999.
- [4] GIMENES, Itana M. S. **Uma Introdução ao Processo de Engenharia de Software**, XIII Jornada de atualização em Informática: Caxambu – MG, 1994.
- [5] OSTERWEIL, Leon, **Software Processes are Software T $\hat{o}$ o**, University of Colorado boulder, Colorado, USA, ACM, 1987.
- [6] GIMENES, I. M. de S., Huzita, E. H. M., Carniello, A., Fantinato, M., **ExpPSEE – An Experimental Process Centred Software Engineering Environment**, Relatório Técnico, Universidade Estadual de Maringá, Paraná, 1999.
- [7] CHRISTIE, Alan M., **Software Process Automation**, Springer-verlag: Berlin Heidelberg , New York, USA, 1995.
- [8] SEAL - Software Engineering Applications Laboratory, **Learning to Use the SEAL Process Assessment Tool**, Revision 0.90, África do Sul, 1999.
- [9] ISO/IEC/JTC1/SC7/WG10, **Phase 2 Trials Ínterim Report**, Versão 1.0, 1998.
- [10] HUMPHREY Watts S. **Managing the *software* process**, Reading, Mass: Addison-Wesley, 1989.
- [11] PRESSMAN, Roger S. **Engenharia de Software**, Makron Books do Brasil Editora Ltda: São Paulo, 1995.
- [12] FINKELSTEIN, Anthony, Jeff Kramer and Bashar Nuseibeh, **Software Process Modelling and Technology**, John Willey & Sons Inc.: New York , New York, 1994.
- [13] SALVIANO, Clênio F., Alfredo N.; Claudete M. Rego, Gláucia F. Azevedo, Luciano K. Meneghetti, Márcia C. C. Costa, Mário Bento de Carvalho, Regina M. T. Colombo, **Qualidade de *software*: visões de produto e processo de *software***, Fundação CTI: Campinas, 1996.
- [14] CARNIELLO, Andréia, Itana M. S. Gimenes, **Um ambiente de programação de processos cooperativos de *software* para o ExpPSEE**, Universidade Estadual de Maringá, Paraná, 1999.