

UNIVERSIDADE FEDERAL DE PERNAMBUCO
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA

MONOGRAFIA DA DISCIPLINA QUALIDADE DE SOFTWARE

**USABILIDADE NO PROCESSO DE
DESENVOLVIMENTO DE SOFTWARE**

Alessandro Vasconcelos
Alexandra Barreto Assad de Barros
Cynthia Pimentel Belleza Bernardino
Filipe Levi Barros de Azevedo
Rafael Machado Duarte

Recife, 3 de Setembro de 2004

RESUMO

Neste trabalho abordamos a área de usabilidade no processo de desenvolvimento de software, para isso iniciamos o trabalho falando das fases e os processos de desenvolvimento de software, os processos mais usados atualmente o Rational Unified Process (RUP) e o eXtreme Programming (XP) são explorados e detalhados, sobre o RUP são descritas suas boas práticas, a estrutura do processo e suas fases, sobre o XP seus valores e princípios e o seu ciclo de vida, depois é feita uma comparação entre os dois visando destacar suas aplicações e objetivos. Em seguida é destacada a questão da qualidade do processo de software, e são descritos os principais modelos de qualidade existentes: ISO 9000-3; ISO/IEC 12207-1; SEI SW-CMM e SPICE, por fim é feita uma comparação entre os modelos destacando a forma que cada modelo atua para atingir seu objetivo. Na última parte é discutida e detalhada a Usabilidade no Processo de Desenvolvimento de Software, é explicada a importância e os ganhos de se considerar a usabilidade como um fator chave no processo de desenvolvimento, e depois são apresentados e discutidos os modelos centrados nos usuários e como estes se relacionam com os principais modelos de qualidade existentes, destacamos como relacionar a usabilidade com projetos, ciclos de vida e metodologias de processo de desenvolvimento de software em geral com o projeto centrado no usuário.

Palavras-chave: usabilidade, processo de software, ISO 13407, ISO TR 18529, qualidade de software.

SUMÁRIO

Capítulo 1—Introdução	1
Capítulo 2—Processo de Desenvolvimento de Software	3
2.1 Introdução	3
2.1.1 Fases do Processo de Desenvolvimento de Software . .	3
2.2 Rational Unified Process (RUP)	5
2.2.1 Boas Práticas	5
2.2.2 Estrutura do Processo	6
2.2.3 Fases do RUP	8
2.3 eXtreme Programming (XP)	8
2.3.1 Introdução	8
2.3.2 Valores e Princípios	8
2.3.3 O Ciclo de Vida	10
2.4 Comparando RUP e XP	11
Capítulo 3—Qualidade de Processo de Software	13
3.1 Introdução	13
3.2 Modelos de Qualidade	13
3.2.1 ISO 9000-3	13
3.2.2 ISO/IEC 12207-1	14
3.2.3 SEI SW-CMM	15
3.2.4 SPICE	16
3.2.5 Comparação entre os Modelos Apresentados	19
Capítulo 4—Usabilidade no Processo de Desenvolvimento de Software	21
4.1 Introdução	21
4.2 Garantia de Usabilidade	21
4.2.1 Determinantes de Usabilidade	21
4.2.2 Combinando Componentes para Garantir Usabilidade .	22
4.3 História do Modelo de Design Centrado no Usuário	25

SUMÁRIO	iii
4.4 ISO 13407 e ISO TR 18529	27
4.4.1 Processo Centrado no Usuário: ISO 13407	27
4.4.2 Descrição do Processo de Ciclo de Vida Centrado no Humano: ISO TR 18529	28
4.5 Relação com Projetos, Ciclos de Vida e Metodologias	29
Capítulo 5—Conclusões	31
Apêndice A—Terminologia de Descrição, Avaliação e Melhoria de Processo	37
Apêndice B—ISO 13407	41
Apêndice C—ISO TR 18529 - Processos para o Projeto Centrado no Humano	43
Apêndice D—Exemplos de Melhoria em Processos Centrados no Usuário	51
D.1 Avaliação formal frente o modelo de maturidade de usabilidade	51
D.1.1 Procedimento	51
D.1.2 Resultado	51
D.2 Avaliação informal frente o modelo de maturidade de usabilidade	52
D.2.1 Procedimento	52
D.2.2 Resultado	52
D.3 Avaliação informal frente o modelo ciente do humano	52
D.3.1 Procedimento	53
D.3.2 Resultado	53

LISTA DE FIGURAS

1.1	Métodos para obtenção de qualidade em uso.	1
2.1	Fases do Processo de Desenvolvimento de Software e suas tarefas.	4
2.2	Estrutura do RUP	6
2.3	Elementos primários do RUP	7
2.4	Comparando método em cascata com o iterativo	7
2.5	Ciclo de vida no XP	10
2.6	Comparando XP e RUP	12
3.1	Visão geral dos Processos - ISO/IEC 12207-1.	15
3.2	Níveis de maturidade e áreas chave de processo - SW-CMM versão 1.1	16
3.3	Avaliação de Processo de Software - SPICE	17
3.4	Modelo de Referência de Processos do SPICE	18
3.5	Níveis de Capacitação	19
3.6	Comparação entre os modelos apresentados	20
4.1	A interdependência das atividades do projeto centrado no usuário.	28
A.1	O processo de melhoramento do processo	39

CAPÍTULO 1

INTRODUÇÃO

Com o visível crescimento no mercado de Tecnologia da Informação e, conseqüentemente, de empresas de desenvolvimento de software, uma maior atenção tem sido dada à garantia de qualidade sob o produto final. Há duas formas principais de verificação do nível de qualidade: através do produto ou do processo. A última mostra-se mais interessante no caso da maioria das empresas de software, pois supõe melhorias a longo prazo e que se estendem a uma gama de produtos e/ou projetos.

Uma maior consciência para com os usuários finais dos sistemas tem despertado empresas para o termo 'qualidade no uso', que seria o efeito do produto no contexto de uso, indo de encontro com as necessidades dos usuários. Ao todo, são três métodos de melhoria de qualidade complementares (Figura 1.1).

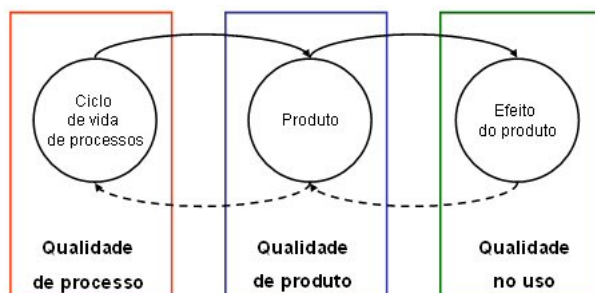


Figura 1.1. Métodos para obtenção de qualidade em uso.

A qualidade do processo de desenvolvimento de software pode ser melhorada a através da definição de atividades centradas no usuário.

Neste trabalho, exploraremos a área de usabilidade no processo de desenvolvimento de software, uma área com muito potencial de crescimento e ainda pouco explorada. No Capítulo 2, conheceremos as metodologias de desenvolvimento de software mais utilizados atualmente, tais como o Rational Unified Process (RUP) e eXtreme Programming (XP). No Capítulo 3, veremos a importância da qualidade em processos de software e modelos e normas reconhecidas no mundo. O Capítulo 4 trata da usabilidade como fator mister de qualidade, abordando sua utilização em processos de desen-

volvimento de software. O Capítulo 5 versa sobre as conclusões gerais deste trabalho.

CAPÍTULO 2

PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

2.1 INTRODUÇÃO

O processo de desenvolvimento de software pode ser definido como um conjunto de atividades, métodos, práticas e transformações que as pessoas usam para desenvolver e manter software e os produtos associados. [Paulk, 1993] Seu objetivo é garantir a qualidade do produto desenvolvido e alcançar boa produtividade de desenvolvimento. A forma como medir essa produtividade será vista no próximo capítulo, que trata de qualidade de software.

Sete questões devem ser respondidas por qualquer projeto de sistemas de informação (W5H2)[Boehm, 1996]:

Why. Porque é que o sistema vai ser desenvolvido?

What. O que vai / deve ser feito?

When. Quando é que vai ser feito?

Who. Quem é o responsável?

Where. Onde é que as responsabilidades estão localizadas?

How. Como é que vai ser feito?

How much. Quanto vai custar em termos de recursos?

2.1.1 Fases do Processo de Desenvolvimento de Software

Podemos considerar que um processo pode ser dividido em três grandes fases:

Concepção. Tem como objetivo identificar o que é que o sistema deve fazer (a informação a processar, as funcionalidades a implementar, as restrições existentes, os critérios que determinam o sucesso e a aceitação.)

Implementação. O *como fazer o sistema* e construí-lo de fato, sendo definidas e construídas as estruturas de dados, os programas, os módulos, as interfaces (internas e externas) e os testes a realizar.

Manutenção. Inclui todas as alterações posteriores à aceitação do produto pelo cliente final, a correção de erros e a introdução de melhorias e/ou de novas funcionalidades.

Essas fases podem ser subdivididas em tarefas mais específicas, como ilustrado na Figura 2.1.

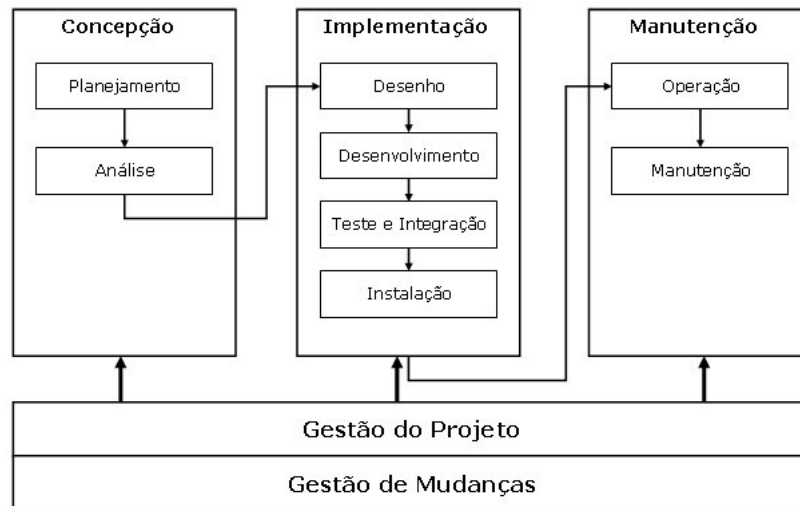


Figura 2.1. Fases do Processo de Desenvolvimento de Software e suas tarefas.

Sendo sete essas tarefas:

Planejamento. Corresponde a uma identificação geral das necessidades, identificação e seleção de alternativas e definição de plano do trabalho.

Análise. Identificação detalhada das funcionalidades do sistema (Levantamento de Requisitos) e a respectiva descrição (Especificação do Sistema) de modo a que os mesmos requisitos possam ser validados pelos usuários finais do sistema.

Desenho. Definição detalhada da arquitetura global da solução (módulos, tabelas, interface, máquinas, etc.).

Desenvolvimento. Programação dos diversos componentes do sistema.

Testes/Integração. O sistema é verificado com o objetivo de obter a aceitação do usuário.

Instalação. Atividades relacionadas com a disponibilização do sistema para os seus usuários finais.

Manutenção. Corresponde ao tempo de vida útil do sistema e durante o qual serão efetuadas todas as alterações posteriores à entrada em funcionamento do produto.

2.2 RATIONAL UNIFIED PROCESS (RUP)

O *Rational Unified Process* (RUP) é um processo de engenharia de software desenvolvido pela *Rational Software Corporation*, cujas principais características são o desenvolvimento iterativo e incremental, orientação a objetos, foco na criação de uma arquitetura robusta, análise de riscos e utilização de casos de uso para o desenvolvimento [Alcântara et al., 2002].

O RUP é um framework para gerar processos que sejam utilizados no desenvolvimento de sistemas dos mais variados tipos. Este processo pode ser perfeitamente configurável de maneira a se adequar a diversas realidades.

É também um processo de engenharia de software disciplinado que possibilita o gerenciamento de atividades e responsabilidades em uma organização de desenvolvimento de software. O objetivo deste processo é produzir, dentro do prazo e custo estimados, um software de alta qualidade que atenda às necessidades dos usuários finais. O RUP é baseado nas melhores práticas utilizadas no desenvolvimento de software moderno [Kruchten, 2000].

Provê informações e facilidades na utilização de ferramentas da *Rational* para o desenvolvimento de software, mas não é necessária a utilização destas para sua efetiva implantação em uma organização [Extreme, 2001]. Pelo fato de ser um framework para gerar processos, o RUP descreve métodos sem a imposição de ferramentas específicas. Por outro lado, a utilização de ferramentas da *Rational* traz ganhos pelo fato de possuírem qualidade, trabalharem dentro dos conceitos do RUP e, principalmente, por trabalharem de forma integrada. O grande problema destas ferramentas é o elevado custo que as tornam inviáveis para boa parte dos projetos.

2.2.1 Boas Práticas

O RUP possui várias boas práticas que buscam maior eficiência ao longo da construção do software. São elas [Extreme, 2001]:

- desenvolvimento de software de forma iterativa;
- gerenciamento de requisitos;
- utilização de arquitetura baseada em componentes;
- modelagem de software visual;

- verificação contínua da qualidade do software;
- controle de mudanças no software.

2.2.2 Estrutura do Processo

O processo é dividido em duas estruturas ou dimensões: a horizontal, que representa o tempo e mostra o ciclo de vida do processo apresentando aspectos dinâmicos expressados em ciclos, fases, iterações e marcos, e a vertical, que representa o núcleo do fluxo de trabalho com os grupos lógicos de atividades, apresentando os aspectos estáticos, como componentes, atividades, fluxos de trabalho, artefatos e papéis [Kruchten, 2000].

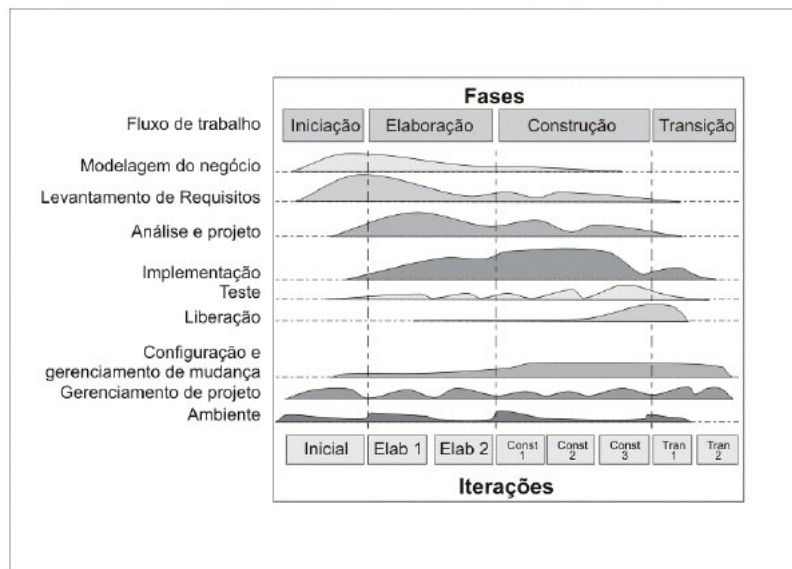


Figura 2.2. Estrutura do RUP

São apresentadas na parte inferior da figura as iterações que ocorrem em cada uma das fases dispostas no topo. No lado esquerdo estão os fluxos de trabalho. Fica, desta forma, bastante evidente o envolvimento em cada uma das fases; a modelagem de negócio, por exemplo, que é muito mais intensa no início do projeto, vai diminuindo durante seu desenvolvimento.

Estrutura Estática Um processo descreve quem está fazendo o que, como e quando. O RUP é representado utilizando quatro elementos primários apresentados pela Figura 2.3 [Kruchten, 2000]:

Papéis. Quem.

Atividades. Como.

Artefatos. O quê.

Fluxos de Trabalho. Quando.

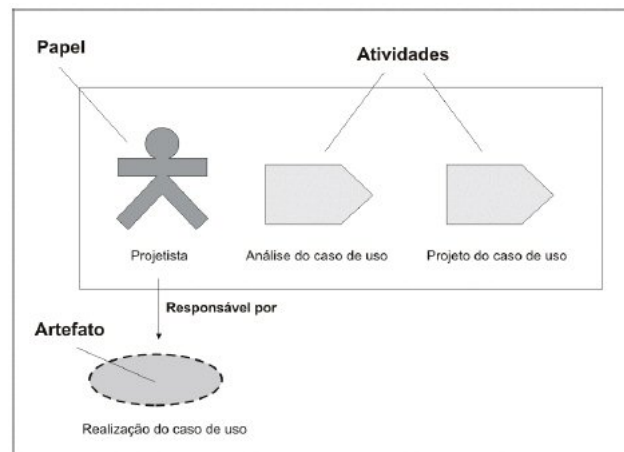


Figura 2.3. Elementos primários do RUP

Estrutura Dinâmica O processo circular e iterativo especificado pelo RUP segue a filosofia *dividir para conquistar*. Desta forma, grandes projetos são quebrados em sucessivos e pequenos projetos em cascata. A Figura 2.4 apresenta a diferença existente entre o método em cascata e o método iterativo.

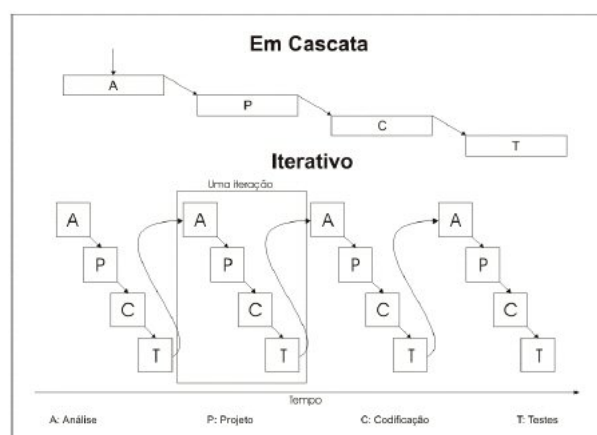


Figura 2.4. Comparando método em cascata com o iterativo

2.2.3 Fases do RUP

O desenvolvimento de software efetuado através do RUP é incremental, e cada incremento é desenvolvido utilizando-se quatro fases: iniciação, elaboração, construção e transição [Calsavara et al., 2002]:

Iniciação. Fase de compreensão do problema e definição da tecnologia a ser utilizada. No final desta fase, é necessário que seja conhecido o escopo do produto, levantados os riscos e concluído que o projeto é viável do ponto de vista do negócio da organização.

Elaboração. Fase de descrição da arquitetura do software, em que os requisitos que mais possuem impacto na arquitetura são levantados em forma de casos de uso. No final da fase de elaboração, deve ser possível prever custos, elaborar o cronograma e detalhar o plano de construção.

Construção. Fase na qual se desenvolve o produto. Além da codificação, também são produzidos os casos de teste e é criada a documentação.

Transição. Fase de treinamento dos usuários e manutenções realizadas no produto.

2.3 EXTREME PROGRAMMING (XP)

2.3.1 Introdução

Desde sua origem, a indústria de software é uma ferramenta poderosa em um cenário corporativo cada vez mais competitivo. Desta forma, existe uma grande pressão para a apresentação de resultados rápidos e com qualidade. Diante dessa realidade, em 1996, no departamento de computação da montadora de automóveis *DaimlerChrysler*, Kent Beck iniciou o desenvolvimento do *Extreme Programming* (XP), um processo de desenvolvimento peso leve (*lightweight*) [Wells, 2003] cujo objetivo era oferecer uma alternativa de desenvolvimento rápido, consistente, que facilitasse a manutenibilidade e se tornasse totalmente aderente a realidades em que os requisitos são instáveis.

2.3.2 Valores e Princípios

Comunicação, simplicidade, *feedback* e coragem são os valores que direcionam o XP e trazem uma mudança cultural significativa para gerentes que estão acostumados a contar com um conjunto de especificações antes do início da fase de construção do sistema. Existem ainda doze princípios adotados:

Planejamento do Processo. O cliente XP define os recursos necessários e utiliza estimativa fornecida pelos desenvolvedores para priorizar funcionalidades, bem como identificar o que pode ser adiado.

Pequenas Liberações. Os ciclos de desenvolvimento são muito curtos. Partes ainda bem simples do sistema são colocadas em produção para validação rápida de todo o processo, minimizando riscos.

Metáforas do Sistema. Para facilitar a comunicação entre cliente e desenvolvedores, é utilizada uma descrição do sistema sem a utilização de termos técnicos.

Projeto Simples. O projeto de software deve ser simples. Devem ser levados em consideração os requisitos sobre os quais há maior visibilidade. Outros requisitos que possam surgir deverão ser atendidos no futuro.

Teste. Existe uma preocupação com a validação de qualidade durante todo o processo. Os testes são escritos e somente depois é construído o componente de software a ser testado. Esta estratégia será mais bem detalhada na sequência deste documento.

Reconstrução. Durante a construção, são realizadas análises e aperfeiçoamento dos produtos já desenvolvidos.

Programação em Dupla. Seguindo alguns estudos que mostram vantagens nesta forma de trabalho, o desenvolvimento é realizado por dois programadores em uma mesma estação de trabalho.

Propriedade Coletiva. Todo código gerado ao longo da construção pertence a todos os programadores. Assim, alterações podem ser realizadas por todos os membros da equipe.

Integração Contínua. Existem várias integrações diárias que geram uma versão do sistema. Esta integração é realizada gradativamente pelo próprio desenvolvedor durante a construção do software.

Quarenta Horas de Trabalho Semanal. As equipes não trabalham excessivamente, minimizando assim as falhas relacionadas à exaustão.

Cliente Dedicado. Existe um cliente dedicado ao projeto. Este determina requisitos, atribui prioridades e responde a dúvidas existentes. Desta forma, é minimizada a necessidade de documentos.

Código Padrão. São seguidos padrões de codificação que asseguram a clareza do mesmo.

2.3.3 O Ciclo de Vida

O ciclo de vida de um projeto que utiliza XP possui as fases de Planejamento, Teste, Codificação e Projeto [Oshiro et al, (s.d)], ilustradas na Figura 2.5.

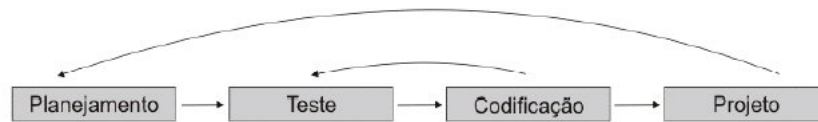


Figura 2.5. Ciclo de vida no XP

.Em um primeiro momento, os requisitos são cuidadosamente coletados, em seguida são elaborados os teste. Após isto, é realizada a etapa de construção e, por fim, o sistema é novamente projetado, ou reconstruído.

Planejamento Para oferecer o dinamismo necessário em um ambiente em que os requisitos são freqüentemente alterados, o planejamento em um projeto guiado pelo XP é constantemente modificado e, por isso, segue como premissa a simplicidade. Os seguintes artefatos são criados nesta fase:

User Stories. Descrição dos requisitos, feito pelo usuário. Também são especificados os cenários de testes para validação dos requisitos.

Release Plan. Planejamento das versões do sistema que serão liberadas. Deve prever versões menores do sistema.

Ao longo do desenvolvimento, deve haver uma verificação constante do andamento das atividades. O desenvolvimento é realizado de forma iterativa, e os programadores não devem atuar em um único ponto do projeto, mas transitar por várias partes dele para que todos conheçam todo o sistema. Reuniões rápidas devem ser realizadas para sincronizar as informações.

As *user stories* são agrupadas e priorizadas. Normalmente, são utilizadas três semanas como base de tempo para formação de um grupo. Os desenvolvedores são responsáveis por informar o tempo de desenvolvimento, cuja precisão é de vital importância para um planejamento bem-feito. Este grupo de *user stories* é colocado no *release plan* levando em consideração os recursos disponíveis. No início do desenvolvimento de cada iteração, deve ser realizada uma reunião para planejar a melhor forma de trabalhar nesta iteração e quais erros existentes no sistema serão consertados. Ao final de cada iteração ou liberação de uma versão, deve ser analisada a velocidade da equipe verificando o desempenho obtido.

Teste Os dois tipos de testes de execução mais comuns são os **unitários**, em que o programador escreve os códigos para verificar se os componentes de um sistema funcionam corretamente, e os de **sistema** (ou testes funcionais), em que um testador opera o sistema para verificar se ele se comporta como esperado. Testes unitários são um ingrediente essencial para a prática de integração contínua e para o próprio XP.

Codificação Neste momento, é realizada a tradução das necessidades anteriormente traçadas nas *user stories* para codificação do software que será gerado. Para que esta atividade seja realizada com qualidade, três práticas importantes devem ser obedecidas: programação em dupla, estratégia de integração contínua e código coletivo.

Projeto A etapa de projeto para a construção de sistemas pode ser realizada seguindo duas abordagens: de forma **planejada** ou de forma **evolutiva** [Fowler, (s.d.)]. A primeira tem como princípio a evolução à medida que a construção avança. Tradicionalmente, esta abordagem transforma o projeto em uma série de decisões tomadas para problemas quando estes aparecem, tornando-as muito reativas. Os problemas aumentam com o passar do tempo e o crescimento do projeto. Na forma planejada, são empregados conceitos de engenharia de software para que as principais funcionalidades sejam previstas e devidamente tratadas. Geralmente, é utilizada uma notação padrão como, por exemplo, UML, para determinar e documentar de forma padronizada o que foi ou será implementado. Trabalhando desta forma, quando existir a necessidade de alterações, existirá maior impacto, ou seja, os artefatos de projetos devem ser alterados.

O XP utiliza o processo evolutivo para projeto. Esta é colocada como a melhor alternativa onde os requisitos são instáveis; Práticas como integração contínua e testes são apontadas como práticas para reduzir os problemas desta forma de trabalho, uma vez que oferece sincronização da equipe e segurança de funcionamento do software produzido.

2.4 COMPARANDO RUP E XP

A principal diferença existente entre RUP e XP é a natureza de aplicação. O primeiro é chamado de processo pesado (*heavy weight*), oferece maior controle, exige a criação de mais artefatos, o que o torna mais burocrático. Já o XP é um caminho inverso. Chamado de processo leve (*light weight*), oferece menor controle, maior velocidade e maior simplicidade. Tradicionalmente, a necessidade de controle em um projeto de software é diretamente propor-

cional ao seu tamanho e, por conseqüência, ao número de pessoas envolvidas. Outro fator de influência é a estabilidade dos recursos. Uma mudança de requisito terá impacto maior em um ambiente em que o número de artefatos é maior. Desta forma, XP propõe-se a ser mais aderente a projetos menores e de requisitos instáveis, porém não se aplica a projetos de grande porte.

Existem opiniões divergentes com relação ao RUP. Pelo fato de ser um *framework* que pode ser estendido para a criação de processos para várias realidades, Smith [Smith, 2003] defende que o RUP pode ser customizado para a utilização ágil em pequenos projetos. Já Skaskiw [Skaskiw, 2001] apresenta o RUP como um processo a ser utilizado em projetos de grande porte. A natureza do RUP, seus conceitos e preocupações, realmente o tornam um processo pesado. Isto faz com que sua utilização em projetos pequenos não se justifique.

O RUP reforça a redução de riscos e assume que o custo de mudanças é muito alto e cresce exponencialmente, apresentando práticas para evitá-las. O XP considera que as alterações serão necessárias e possuirão baixo custo. Existe uma preparação para estar constantemente modificando o sistema.

Utilizando-se o XP, não existe muita documentação para manter atualizada. Já no RUP, uma modificação gera alterações em documentos, diagramas e fontes.

A Figura 2.6 apresenta uma comparação considerando vários pontos específicos dos dois processos [Rawsthorne, 2003].

Prática ou Conceito	RUP	XP
Iterativo ou incremental	X	X
Centrado na arquitetura	X	
Centrado na validação		X
Iteração do cliente com o desenvolvedor		X
Gerenciamento de riscos	X	X
Qualidade de código		X
Equipes grandes	X	
Equipes pequenas		X
Projetos complexos	X	

Figura 2.6. Comparando XP e RUP

O XP apresenta um número muito menor de artefatos e papéis em sua especificação. Segundo Smith [Smith, 2003], o RUP apresenta mais de 100 artefatos, e o XP, por volta de 30.

CAPÍTULO 3

QUALIDADE DE PROCESSO DE SOFTWARE

3.1 INTRODUÇÃO

A qualidade de software é largamente determinada pela qualidade dos processos utilizados para o desenvolvimento. Deste modo, a melhoria da qualidade de software é obtida pela melhoria da qualidade dos processos. Esta visão orientou a elaboração de modelos de definição, avaliação e melhoria de processos de software. Os modelos mais significativos estão apresentados nos subitens a seguir. Também é apresentado um quadro comparativo entre eles.

3.2 MODELOS DE QUALIDADE

3.2.1 ISO 9000-3

A ISO 9000-3 é um guia de aplicação da ISO 9001 [NBR, 9001] para o desenvolvimento, fornecimento e manutenção de software. A Norma ISO 9001 faz parte da série de normas ISO 9000, voltadas para a gestão e garantia da qualidade. Estas normas especificam os requisitos mínimos para que as empresas possam assegurar a qualidade de seus produtos e serviços, não definindo modelos ou impondo sistemas de qualidade a serem implementados nas organizações. As empresas definem seus próprios modelos de gestão da qualidade, dependendo do seu tipo de negócio e suas características.

A ISO 9001 é aplicável a empresas em geral que atuam em projeto, desenvolvimento, produção, instalação e assistência técnica. Como esses documentos da Série ISO 9000 são genéricos, foi necessária a elaboração de um documento complementar onde fossem abordados alguns aspectos importantes e específicos de software. Em junho de 1993 foi criada a Norma ISO 9000-3 com diretrizes para aplicação da ISO 9001 ao desenvolvimento, fornecimento e manutenção de software. Esta norma se espelha nos itens da ISO 9001; fazendo a necessária adaptação. Para cada item da ISO 9001 existe um correspondente na ISO 9000-3 que o detalha e o adequa ao software.

As diretrizes propostas na ISO 9000-3 cobrem questões como o entendimento comum entre as partes (contratante e contratado) de requisitos funcionais e o uso de metodologias consistentes para o desenvolvimento de software e gerenciamento de projeto como um todo, da concepção até a

manutenção. É dividida em três partes principais.

Estrutura. Descreve aspectos organizacionais, relacionados ao sistema de qualidade. São detalhadas as responsabilidades e ações relacionadas à qualidade que devem ser tomadas tanto pelo fornecedor como pelo comprador. Os pontos abordados são: responsabilidades da administração, sistema de qualidade, auditorias internas do sistema de qualidade e ação corretiva.

Atividades do Ciclo de Vida. Descreve as atividades de desenvolvimento de software. A Norma define que o desenvolvimento de software deve ser feito segundo um determinado modelo de ciclo de vida, e as atividades relacionadas à qualidade devem ser planejadas e implementadas de acordo com a natureza deste modelo. Independentemente do modelo de ciclo de vida estabelecido pela organização, a Norma define que as atividades do ciclo de vida devem ser agrupadas em nove categorias: análise crítica do contrato; especificação dos requisitos do comprador; planejamento da qualidade; projeto e implementação; ensaios e validação; aceitação; cópia, entrega e instalação; e manutenção.

Atividades de Suporte. Descreve as atividades que apoiam as atividades do ciclo de vida de desenvolvimento. Estão organizadas em nove itens: gestão de configuração; controle de desenvolvimento; registros da qualidade, medição; regras, práticas e convenções; ferramentas e técnicas; aquisição; produto de software incluído; e treinamento.

É importante salientar que a ISO 9001 somente indica o controle da não conformidade de um produto e recomenda ações corretivas e preventivas. A melhoria contínua do processo não é abordada por ela tal como é abordada explicitamente no modelo SW-CMM ou em outros [Coallier, 1994].

A procura pela certificação da série ISO 9000 é um dos grandes motivadores do atual movimento em relação à qualidade em todas as áreas de atividades econômicas, incluindo o software. Ela influenciou praticamente todas as iniciativas em qualidade de software. Em particular, o TickIT [TIC, 1992], desenvolvido no Reino Unido, que provê um esquema uniforme para estender a certificação ao campo de software, é baseado na ISO 9001 e ISO 9000-3 [ISO9000-3, 1993].

3.2.2 ISO/IEC 12207-1

Esta Norma começou a ser elaborada em junho de 1989, no grupo de trabalho ISO/IEC JTC1/SC7/WG7 (Comitê de Engenharia de Software) e foi

aprovada em agosto de 1995. Ela estabelece os processos, atividades e tarefas a serem aplicados durante a aquisição, fornecimento, desenvolvimento, operação e manutenção de software. A Norma apresenta uma definição abrangente em relação aos processos e orienta a adaptação para sua utilização nos projetos de software implementados em uma organização.

A Norma define dezessete processos do ciclo de vida de software e os organiza em três classes: processos fundamentais, processos de apoio e processos organizacionais. Cada classe contém os processos definidos e os possíveis usuários, como apresentado na Figura 3.1.

A importância desta Norma é o estabelecimento de uma estrutura de classificação de processos normalizando a terminologia.

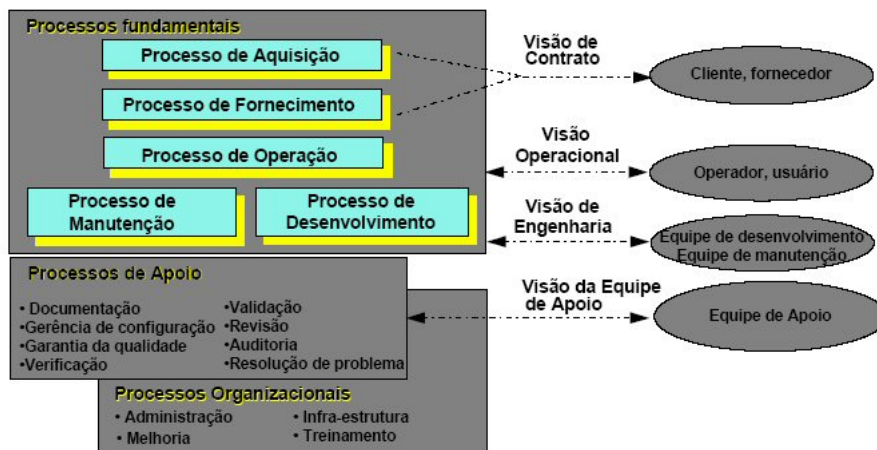


Figura 3.1. Visão geral dos Processos - ISO/IEC 12207-1.

3.2.3 SEI SW-CMM

O modelo *Capability Maturity Model* (SW-CMM) propõe a avaliação da capacidade e maturidade de uma organização e indica diretrizes para a melhoria. Foi desenvolvida pelo *Software Engineering Institute* (SEI) da *Carnegie Mellon University*, atendendo a clientes como o *Department of Defense* (DoD) dos EUA. Divulgado a partir de 1991, tornou-se um dos modelos de maior prestígio.

Neste modelo, as organizações de software são enquadradas em um dos cinco níveis de maturidade definidos. A Figura 3.2 [Paulk, 1995] mostra as características da organização e as áreas chaves de processo necessárias para que ela seja classificada em cada um dos níveis, conforme a versão 1.1 do SW-CMM. Essa estrutura em níveis do SW-CMM está baseada nos princípios de

qualidade de produto propostos por Walter Shewart, W. Edwards Deming, Joseph Juran e Philip Crosby.

O modelo SW-CMM teve tal repercussão, que diversos outros se baseiam diretamente nele, como é o caso do Bootstrap e do Trillium [TRI, 1994] e influenciou fortemente outros como o SPICE.

NÍVEIS E CARACTERÍSTICAS	ÁREAS CHAVES DE PROCESSO
Nível 5 - Otimizado Melhoria contínua do processo é possibilitada pela realimentação quantitativa do processo e conduzida a partir de idéias e tecnologias inovativas.	Prevenção de defeitos Gerenciamento de mudanças tecnológicas Gerenciamento de mudanças de processo
Nível 4 - Gerenciado São efetuadas medições detalhadas do processo de software e qualidade do produto. Tanto o processo como o produto são entendidos e controlados quantitativamente.	Gerenciamento quantitativo do processo Gerenciamento da Qualidade de Software
Nível 3 - Definido O processo de software em relação tanto às atividades de gerenciamento, como de engenharia, são documentadas, padronizadas e integradas em processos padrão para a organização. Todos os projetos usam uma versão aprovada e adaptada do processo padrão de software da organização para o desenvolvimento e manutenção	Foco no processo da organização Definição do processo da organização Programa de Treinamento Engenharia de produto de software Gerenciamento integrado do software Coordenação entre grupos Revisões
Nível 2 - Repetitivo Processos básicos de gerenciamento de projeto são estabelecidos para controlar custos, cronogramas e funcionalidade. A disciplina necessária de processo permite repetir sucessos anteriores em projetos de aplicação similar.	Gerenciamento de requisitos Planejamento de projeto de software Acompanhamento de projeto de software Gerenciamento de subcontratos Qualidade assegurada de software Gerenciamento de Configuração
Nível 1 - Inicial O processo de software é caracterizado como ad hoc, ocasionalmente até caótico. Poucos processos definidos. Sucesso depende dos esforços individuais e heroísmo.	

Figura 3.2. Níveis de maturidade e áreas chave de processo - SW-CMM versão 1.1

3.2.4 SPICE

Em junho de 1991, o comitê de Engenharia de Software da ISO aprovou a realização de estudos para analisar as necessidades e os requisitos de um padrão para avaliação do processo de software.

Como resultado desse estudo, chegou-se à conclusão que havia um consenso internacional sobre a necessidade de requisitos para um padrão de avaliação de processo, e a importância de se adotar uma forma de desenvolvimento em que resultados pudessem ser utilizados o mais breve possível, garantindo que o padrão elaborado atendesse completamente aos seus usuários.

Firmou-se então, um compromisso internacional de iniciar um projeto com uma equipe com dedicação exclusiva, coordenada por quatro centros técnicos de desenvolvimentos: Inglaterra, Austrália, EUA e Canadá.

Dentro dessa visão, em 1993 foi lançado o projeto SPICE, com o objetivo de gerar normas para avaliação de processos, visando a melhoria contínua do processo e a determinação da sua capacitação. O modelo de referência do SPICE (Software Process Improvement and Capability Determination) é, na verdade, um *framework* para valiação de processos de software que harmoniza os diversos modelos nos quais ele se baseia, como: SW-CMM, Trillium, *Software Technology Diagnostic* (STD) e Bootstrap. O objetivo é que cada um destes modelos, e outros que venham a ser criados, possam ser definidos como modelos compatíveis com este *framework*, possibilitando que os resultados de avaliações, segundo cada um destes modelos, possam ser comparados. O resultado do projeto tornou-se a norma ISO/IEC 15504.

O SPICE pode ser utilizado por organizações envolvidas em planejar, gerenciar, monitorar, controlar e melhorar a aquisição, fornecimento, desenvolvimento, operação, evolução e suporte de software.

Dentro da visão do SPICE (Figura 3.3), a avaliação de processos de software tem como propósito:

- entender o estado dos processos de uma organização para a sua melhoria;
- determinar a adequação dos processos de uma organização para um requisito particular ou uma classe de requisitos;
- determinar a adequação dos processos de uma outra organização para um determinado contrato ou para uma classe de contratos.

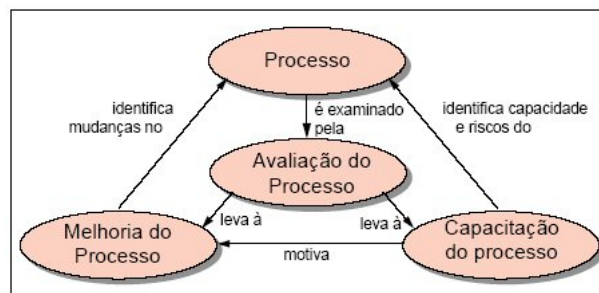


Figura 3.3. Avaliação de Processo de Software - SPICE

Dentro do contexto de melhoria de processos, a avaliação significa a caracterização das práticas correntes de uma organização, unidade organizacional ou projeto em termos da capacidade dos processos selecionados. A análise dos resultados é feita em relação às necessidades de negócio da organização,

identificando os aspectos positivos e negativos, e os riscos associados aos processos. Isto leva a determinar se os processos estão atingindo efetivamente seus objetivos e identificar causas da baixa qualidade, alto custo ou tempo excessivo, indicando a priorização na melhoria dos processos.

O SPICE estabelece um Modelo de Referência de processos e define um conjunto universal de processos que são fundamentais para uma boa engenharia de software, cobrindo as melhores práticas. A estrutura desse modelo é apresentada na Figura 3.4.

CATEGORIAS DE PROCESSO	SIGLA	PROCESSOS
Cliente-Fornecedor (Customer-Supplier)	CUS.1	Adquirir software
	CUS.2	Gerenciar necessidades do cliente
	CUS.3	Fornecer software
	CUS.4	Operar o software
	CUS.5	Prover serviço ao cliente
Engenharia	ENG.1	Desenvolver requisitos e projeto do sistema
	ENG.2	Desenvolver requisitos de software
	ENG.3	Desenvolver projeto do software
	ENG.4	Implementar o projeto do software
	ENG.5	Integrar e testar o software
	ENG.6	Integrar e testar o sistema
	ENG.7	Manter o sistema e o software
Apoio (Support)	SUP.1	Desenvolver documentação
	SUP.2	Desempenhar a gerência de configuração
	SUP.3	Executar a garantia da qualidade
	SUP.4	Executar a verificação dos produtos de trabalho
	SUP.5	Executar a validação dos produtos de trabalho
	SUP.6	Executar revisões conjuntas
	SUP.7	Executar auditorias
	SUP.8	Executar resolução do problema
Gerência (Management)	MAN.1	Gerenciar o projeto
	MAN.2	Gerenciar a qualidade
	MAN.3	Gerenciar riscos
	MAN.4	Gerenciar subcontratantes
Organização	ORG.1	Construir o negócio
	ORG.2	Definir o processo
	ORG.3	Melhorar o processo
	ORG.4	Prover recursos treinados
	ORG.5	Prover infra-estrutura organizacional

Figura 3.4. Modelo de Referência de Processos do SPICE

A determinação da capacidade dos processos de uma organização é feita através da comparação das capacidades de suas práticas contra o modelo para gerenciamento de processos que engloba atividades. Essas atividades são estruturadas de modo a proporcionar um modelo lógico do processo de software, identificando práticas que permitem o gerenciamento e melhoria de qualquer parte do processo ou do processo como um todo.

No Modelo de Referência do SPICE, são definidos ainda seis níveis de capacitação conforme a Figura 3.5. Na avaliação de uma organização, são

selecionados os processos relevantes e, para cada um deles, é atribuído um perfil composto pela porcentagem de adequação a cada um dos níveis de capacitação.

Nível 0	Processo incompleto	o processo não está implementado o processo falha na tentativa de atingir os seus objetivos
Nível 1	Processo executado	o processo implementado atinge o seu objetivo definido
Nível 2	Processo gerenciado	o processo executado entrega produtos de trabalho de definida qualidade dentro de cronogramas e recursos definidos
Nível 3	Processo estabelecido	o processo gerenciado é executado usando um processo definido baseado em bons princípios de engenharia de software
Nível 4	Processo previsível	o processo estabelecido é executado consistentemente dentro de limites definidos de controle para atingir seus objetivos
Nível 5	Processo otimizado	o processo previsível otimiza o seu desempenho para atender às necessidades de negócio atuais e futuras, e atinge repetibilidade em atender seus objetivos definidos de negócios

Figura 3.5. Níveis de Capacitação

Este projeto é interessante pelo seu direcionamento e flexibilidade. Está disponível para que as organizações o utilizem conforme suas necessidades e planos de negócios, medindo a capacitação de cada um de seus processos com o objetivo de promover melhorias contínuas nos mesmos. Deste modo, obtém-se uma avaliação mais detalhada do estado da organização, permitindo a comparação de resultados de avaliações por outros modelos compatíveis.

3.2.5 Comparação entre os Modelos Apresentados

A Figura 3.6 descreve uma visão conjunta dos principais aspectos dos modelos apresentados. Nessa Figura, o aspecto **Abordagem** caracteriza, resumidamente, como e sobre o que cada modelo atua para atingir seu objetivo. **Organizações Alvo** caracteriza a que tipo/porte de organização melhor se aplica o modelo; **Definição de Processos** quantifica o desmembramento e a classificação de processos; **Flexibilidade** indica a possibilidade de adaptação dos aspectos definidos pelo método. Os nomes dos demais aspectos da Figura são auto explicativos e dispensam comentários.

Aspectos abordados	ISO 9000-3	ISO/IEC 12207-1	SW-CMM	SPICE
Objetivo	Certificar a organização de acordo com padrões estabelecidos em situações de contrato de fornecimento de software.	Estabelecer uma terminologia e um entendimento comum para os processos entre todos os envolvidos com software.	Determinar a capacitação da organização e apoiar a sua evolução de acordo com os níveis estabelecidos	Conhecer e avaliar os processos da organização, determinar a capacitação e promover a melhoria.
Abordagem	Verificação de conformidade de processos a padrões documentados.	Definição dos processos para aquisição, fornecimento, desenvolvimento, operação e manutenção de software.	Avaliação dos processos e enquadramento da organização em um dos níveis de maturidade.	Avaliação dos processos da organização em relação a níveis de capacitação.
Organizações Alvo	Organizações que necessitam de uma certificação.	Organizações em geral	Organizações que necessitam de comprovação formal de sua capacidade	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 de processos organizados em 5 níveis crescentes de maturidade.	Estabelece 29 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 e entrevistas	Fornece orientações para definição dos instrumentos.
Inspiração e Influência	Normas militares americanas, canadenses, Sistemas de qualidade do Reino Unido.	TQM, PDCA	Princípios de Shewart, Deming, Juran, Crosby.	TQM, PDCA, SW-CMM, STD, Trillium, Malcolm Baldrige, Bootstrap.
Aspectos Positivos	Norma Internacional; Difusão extensa; Reconhecimento do valor da certificação.	Norma Internacional; Definição de uma taxonomia para processos útil para qualquer organização.	Estabelecimento de diretrizes para a melhoria contínua. Difusão extensa nos EUA.	Norma Internacional em elaboração; Expansão e 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 à diversidade das organizações. Dificuldade de aplicação em pequenas organizações. Falta abordagem de produto.	Devido à grande quantidade de informações, exige treinamento para sua aplicação. Falta abordagem de produto.

Figura 3.6. Comparação entre os modelos apresentados

CAPÍTULO 4

USABILIDADE NO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

4.1 INTRODUÇÃO

Para o usuário, a interface é o sistema. Naturalmente, a funcionalidade por trás deve ser relevante e trabalhar apropriadamente. Entretanto, se o sistema é difícil de usar ou esteticamente não-prazeroso, satisfação e aceitação irão sofrer em proporção direta ao nível de frustração. Clientes confusos recorrem a serviços de suporte técnico, incorrendo em custos adicionais para o fornecedor, presumindo que eles têm paciência para completar suas tarefas. Usabilidade pobre custa dinheiro.

A proposta ao desenvolver um sistema interativo é descobrir as necessidades dos usuários: para prover qualidade no uso [Bevan, 1999]. O termo qualidade em uso significando que o software não existe isoladamente, mas deve adequar-se a um ambiente de trabalho socio-tecnológico se criado para funcionar na prática.

Com o objetivo de aumentar diretamente a qualidade de produtos e sistemas, métodos de usabilidade são integrados aos processos de desenvolvimento de sistemas existentes.

4.2 GARANTIA DE USABILIDADE

4.2.1 Determinantes de Usabilidade

A utilização de uma abordagem centrada no usuário para o design pode reduzir o tempo de desenvolvimento e retrabalho para novas versões, melhorando a produtividade dos usuários e reduzindo treinamento, documentação e custos de suporte [Bevan, 2001a]. A ISO 9241-11 define usabilidade em termos de efetividade, eficiência e satisfação [ISO 9241, 1998], enquanto a ISO 9126-1 considera efetividade, eficiência, segurança e satisfação como principais objetivos do desenvolvimento de sistemas [ISO 9126, 2000]. Atributos do produto, tais como capacidade de compreensão, aprendizado e operação, contribuem para a usabilidade. Como usabilidade vem sendo reconhecida como um requisito para produtos, clientes e usuários necessitam, cada vez mais, de alguma garantia sobre esse fator.

Seis métodos para garantia de usabilidade foram descritos por Daly-Jones, Bevan e Thomas [Daly et al., 1997], Bevan, Claridge, Earthy e Kirakowski [Bevan et al., 1998], Earthy [Earthy, 1998a,b] e EUSC [EUSC, 1998]. Para qualquer negócio, é necessária uma combinação de componentes para dar o requerido nível de garantia de que a usabilidade do produto seja válida e realizável. A Tabela 4.1 lista cada componente, sua finalidade e comentários sobre sua aplicação prática.

4.2.2 Combinando Componentes para Garantir Usabilidade

A tradicional garantia de **atributos do produto** pega um produto e testa-o baseado em um padrão aceito. Validações baseadas em performance fixa ou atributos característicos tornam-se menos úteis com o aumento da complexidade dos produtos. Isso é algo extremamente limitante com sistemas de tecnologia da informação (TI) e atributos de qualidade tão complexos como usabilidade. A inspeção do produto pode encontrar algumas possíveis causas de problemas de usabilidade, mas não pode avaliar suas consequências.

A garantia de **performance e satisfação do usuário** forma um elemento necessário na garantia de usabilidade. Entretanto, a especificação de métricas de performance e condições de teste válidas pode não ser possível até bem depois do início do projeto.

A **certificação de processo** pode ser utilizada apenas retrospectivamente. Por essa razão, é de grande valor (A) para potenciais clientes e usuários de um produto certificado e (B) como evidência de que um requisito de contrato de aplicar um padrão de processo foi satisfeito.

Como os componentes mais tradicionais de garantia (atributos de produto, performance do usuário e certificação do processo) são voltados para projetos mais estáveis, houve a necessidade de encontrar caminhos para prever a qualidade de produtos em projetos com mudanças nos requisitos do usuário, contexto de uso e tecnologia-base durante o projeto e o desenvolvimento. Esses são discutidos abaixo.

Métodos genéricos para maturidade baseados em Crosby [Crosby, 1978], como **organização centrada no humano** [Ehrlich e Rohn, 1994] [Flanagan, 1996] [Earthy, 1998b], identificam riscos potenciais, mas não são precisos sobre soluções específicas. **Competência técnica** é relevante (ex: MErgS - Member of the Ergonomics Society), mas fatores como dinâmica da equipe e contexto de projeto terão um impacto mais significativo nos atributos do produto do que contribuições individuais apenas.

A **capacidade do processo** [Humphrey, 1989] é única em oferecer (A) uma predição da habilidade de um negócio de entregar um produto que satisfaz um nível requerido de performance, (B) uma indicação dos fatores que

Componente	O que é	Finalidade(s)	Comentários
Atributos do Produto	Atesta se um produto segue os guidelines ergonômicos	Usado no final da fase de desenvolvimento para verificar uma conclusão bem-sucedida; Usado no final do projeto como parte da certificação; Pode ser diagnóstico (ex: prover orientação no re-design do produto); Conformidade com os princípios da ISO 9241.	Análises retrospectivas; Específico para um produto ou versão de um produto; Difícil de relacionar ao contexto de uso; Não avalia as consequências das falhas.
Performance e Satisfação do Usuário	Medindo a usabilidade de um produto (ex: ISO 9241-11)	Usado no final do ciclo de desenvolvimento para validação; Usado no final do projeto como parte da aceitação; Pode ser usado para atestar a adequação de um produto a ser adquirido; Pode ser combinado com diagnóstico de retorno (orientação no re-design)	Análise retrospectiva; Específico para uma versão do produto; Válido apenas para o contexto usado na medição; Não diagnostica a causa das falhas; Avaliação completa pode ser custosa.
Certificação do Processo	Atestando se um processo de desenvolvimento centrado no usuário foi utilizado.	Aumenta a confiança do cliente; Retorno para o desenvolvedor; Interpretação da ISO 13407 para produtos genéricos; Definição de processos do projeto; Aplicado a um produto ou uma linha de produtos; Depende do desenvolvedor possuir um processo reconhecido	Análise retrospectiva; Específico para um produto ou linha de produtos.
Organização Centrada no Humano	Atesta a maturidade da centralização no humano de uma organização	Avaliação inicial do cliente; Levantamento da maturidade e usabilidade; Modelo para mudança de gerenciamento/atitude; Extra-escala para avaliação de processos de software/sistemas; Fonte de práticas para avaliações de software/sistemas.	Não preciso sobre como conseguir melhorias

Tabela 4.1. Componentes de um ambiente de gerência de garantia [Earthy et al., 2001].

Componente	O que é	Finalidade(s)	Comentários
Competência Técnica	Reconhece a habilidade de uma organização como um fornecedor de serviços da usabilidade	Reconhecimento de consultantes da usabilidade; Qualificação do subcontratante; Projeto do treinamento; Desenvolvimento organizacional; Desenvolvimento da equipe; Qualificação profissional.	Sensível à equipe; Aplicação limitada; Não se dirige ao desenvolvimento de produto.
Capacidade do Processo	Avaliar o potencial de uma organização para executar atividades centradas no usuário.	Avaliação de uma linha de base de processos centrados no usuário; Avaliação para o reconhecimento de processos centrados no usuário; Elaboração da ISO 13407; Instrução sobre quais necessidades devem ser feitas no projeto centrado no usuário; Modelo para a melhoria de processos; Modelo da referência da indústria.	Requiere compromisso organizacional; Nova extensão do processo de avaliação.

Tabela 4.2. Continuação.

impedem essa habilidade, e (C) os meios de tratar tais falhas. Essas forças têm levado à crescente adoção de capacidade do processo como um elemento na garantia da entrega de um software efetivo e ideal. Essa monografia descreve, a seguir, o desenvolvimento e a aplicação de capacidade do processo na garantia de usabilidade.

4.3 HISTÓRIA DO MODELO DE DESIGN CENTRADO NO USUÁRIO

Próximo ao final do desenvolvimento da ISO 9241 - Requisitos Ergonômicos para Área de Trabalho com Terminais de Dispositivo Visual, os grupos envolvidos nesse padrão de produto identificaram a necessidade de um documento separado e de alto-nível que abordasse as atividades de fatores humanos executadas durante o projeto de sistemas interativos. Utilizando o método para projeto centrado no usuário originado de Norman e Draper [Norman et al., 1986], a ISO TC 159/SC4 Ergonomia da Interação Homem-Sistema formou um grupo de pesquisa sob a liderança de Tom Stewart para o desenvolvimento de uma guia para gerentes de projeto. Ela foi publicada como ISO 13407 - Projeto de Processos Centrados no Humano para Sistemas Interativos [ISO 13407, 1999]. A ISO 13407 lista os seguintes princípios para o projeto centrado no humano:

- O ativo envolvimento de usuários e uma clara compreensão dos requisitos do usuário e da tarefa;
- Uma alocação apropriada em função dos usuários e da tecnologia;
- A iteração de soluções de projeto;
- Projeto multi-disciplinar;

Isso especifica as seguintes atividades:

- Planejamento do processo centrado no humano;
- Especificação dos requisitos do usuário e da organização;
- Compreensão e especificação do contexto de uso;
- Produção de soluções de projeto;
- Validação dos projetos baseada nos requisitos.

Como o trabalho na ISO 13407 progrediu, uma conexão formal foi estabelecida com a ISO/IEC JTC1/SC7 - Ergonomia de Software, o comitê

condutor do desenvolvimento de modelos de processo e métodos de avaliação de processo para ciclos de vida de software e sistemas. Essa conexão, e o trabalho no projeto EC INUSE [Earthy, 1996], identificou a necessidade de uma apresentação estruturada das atividades de projeto centradas no usuário para o uso na avaliação de processos.

Vários modelos de avaliação têm sido criados para o projeto centrado no usuário, incluindo *HumanWare* [Taylor et al, 1998], *Maturidade de Gerência em Usabilidade* [Flanagan, 1996], *Maturidade do Projeto Centrado no Usuário* [Eason e Harker, 1997] e *Maturidade de Sistemas Totais* [Sherwood-Jones, 1995]. Os processos desses modelos formaram entradas para o *Modelo de Maturidade em Usabilidade* [Earthy, 1998a,b]. O modelo de maturidade em usabilidade, por sua vez, formou a base para a ISO TR 18529 - Descrições do Ciclo de Vida de Processos Centrados no Humano [ISO TR 18529, 2000], destinados a ser usados em conjunção com a ISO 13407 (vide Apêndice B). A ISO TC 18529 contém descrições de processos centrados no humano estruturados para uso em avaliação de processo (vide Apêndice C).

A ISO TR 18529 obedece aos requisitos da ISO TR 15504 - Avaliação de Processos de Software para modelos de avaliação de processos. Ela pode ser mapeada no modelo de capacidade e maturidade (Capability Maturity Model - CMM; vide Apêndice A) via modelo de maturidade em usabilidade. Essa compatibilidade que permite que resultados de uma avaliação de capacidade de um processo centrado no humano sejam integrados com outras avaliações vem emergindo como um fator importante na avaliação de processos [Hamilton e Cutler, 2000]. A razão é, parcialmente, relacionada à credibilidade, mas é largamente financeira. Nenhum negócio possui orçamento ilimitado para melhoria de processos e avaliações em separado aliadas a uma variedade de modelos incompatíveis parece desorganizado e irá dividir o orçamento da avaliação, criando um problema em priorizar os achados e fazendo o projeto de um programa de melhoria harmônica mais difícil.

Kuuti, Jokela, Nieminen e Jokela [Kuuti et al., 1998] descreveram uma bem-sucedida experiência com o modelo de maturidade em usabilidade em produtos do setor de escritórios. O Apêndice D lista a aplicação da melhoria de processos baseada em modelos por Taylor et al [Taylor et al., 1998] para o projeto centrado no usuário de produtos de alta tecnologia, e esboça avaliações de capacidade em usabilidade por Bevan e Ryan [Bevan e Ryan, 2000] e Bevan e Bogomolni [Bevan e Bogomolni, 2000].

Jokela [Jokela, 2000] examinou alguns modelos de capacidade em usabilidade. Ele concluiu que o modelo de maturidade em usabilidade era a melhor escolha, mas notou que os aspectos gerenciais deveriam ser expandidos. Enquanto destacou a utilidade da avaliação, ele chamou a atenção para a necessidade por outros elementos de garantia - particularmente, competência

técnica, usabilidade na estratégia do negócio e cultura organizacional no projeto centrado no usuário.

No Japão, a usabilidade e a acessibilidade são reconhecidos como atributos de qualidade do produto que definem a imagem da organização [National/Panasonic, 1999]. ISO 13407 é vista como uma ferramenta de campo industrial para a melhoria do produto [Hirasawa, 2000], onde ela complementa o requisito para operar processos de forma transparente para a organização, o cliente e qualquer usuário de um produto. Realmente, o Japão é o primeiro país a produzir livros didáticos que explicam a ISO 13407 [Kurosu et al., 2001].

Subseqüente à publicação da ISO 13407 e ISO TR 18529, a SC7 - Engenharia de Software padronizou a ISO 12207 - Processos do Ciclo de Vida de Software para que incluísse um processo de usabilidade baseado na ISO TR 18529, e a ISO 15288 - Processos de Ciclo de Vida de Sistemas foi conectada à TC159 para levar em conta questões relacionadas a fatores humanos no desenvolvimento e operação de sistemas.

4.4 ISO 13407 E ISO TR 18529

4.4.1 Processo Centrado no Usuário: ISO 13407

A ISO 13407 provê instruções sobre como conseguir qualidade no uso incorporando atividades de projeto centrado no usuário através do ciclo de vida de sistemas interativos computacionais. Ela descreve o projeto centrado no usuário como uma atividade multi-disciplinar, a qual incorpora fatores humanos e conhecimento ergonômico e técnicas com o objetivo de acentuar a efetividade e a produtividade, melhorando as condições de trabalho humano, e contra-atacando possíveis efeitos adversos do uso na saúde, segurança e performance humana.

Nela existem quatro atividades de projeto centrado no usuário que precisam ser iniciadas nos estágios iniciais de um projeto. São elas:

- entender e especificar o contexto de uso;
- especificar os requisitos do usuário e da organização;
- produção de soluções de projeto;
- validação dos projetos baseada nos requisitos.

A natureza iterativa dessas atividades é ilustrada na Figura 4.1. O processo envolve iterações até que os objetivos sejam satisfeitos.

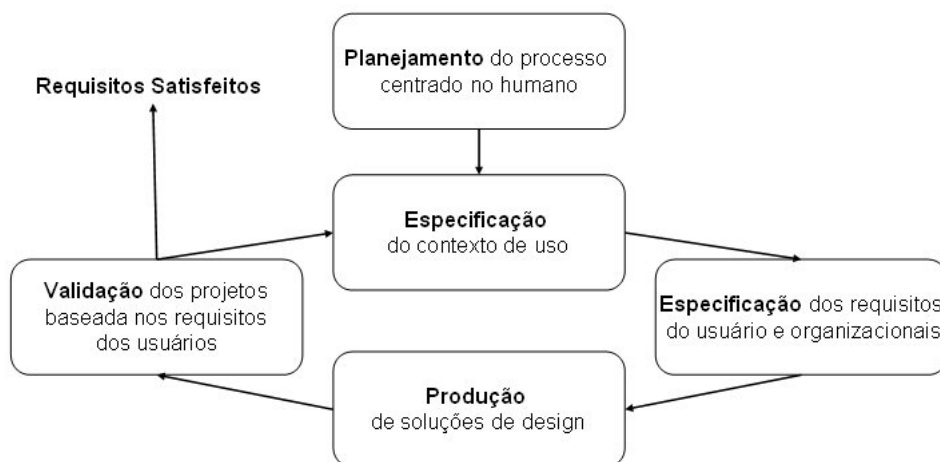


Figura 4.1. A interdependência das atividades do projeto centrado no usuário.

A seqüência em que elas são executadas e o nível de esforço e detalhe apropriado varia de acordo com o ambiente do projeto e o estágio do processo de projeto.

4.4.2 Descrição do Processo de Ciclo de Vida Centrado no Humano: ISO TR 18529

A INUSE desenvolveu uma definição estruturada e formalizada de processos centrados no humano descritos na ISO 13407 [Earthy, 1998a]. Uma versão melhorada foi, subsequente, publicada como ISO TR 18529. Sua intenção é tornar o conteúdo da ISO 13407 acessível a especialistas em avaliação de processos de software e aos familiarizados com ou envolvidos na modelagem de processos. A ISO TR 18529 pode ser usada na especificação, avaliação e melhoria de processos centrados no humano no desenvolvimento e operação de sistemas.

O modelo consiste de sete conjuntos de práticas básicas (vide Apêndice C). Essas práticas descrevem o que deve ser feito em ordem para representar e incluir os usuários de um sistema durante o ciclo de vida. O modelo usa o formato comum de modelos de avaliação de processos. Esses modelos descrevem os processos que devem ser executados por uma organização para alcançar objetivos técnicos definidos. Os processos nesse modelo são descritos no formato definido pela ISO 15504. Embora o uso primário de um modelo de avaliação de processo seja para a medição de quão bem uma organização cumpre com os processos incluídos no modelo, tais modelos podem ser usados como uma descrição do que é requerido com o intuito de projetar e desenvolver processos de projeto e organizacionais efetivos.

4.5 RELAÇÃO COM PROJETOS, CICLOS DE VIDA E METODOLOGIAS

As atividades de projeto centrado no usuário podem ser especificadas ou descritas através de um número de níveis; cada nível possuindo suas próprias propostas e necessidades para relacionar-se com outros níveis com o objetivo de gerenciar, executar e revisar um projeto. A ISO TR 18529 é um exemplo de conjunto de **declarações de processo**. Ela descreve o que deve ser feito para tornar o ciclo de vida de um sistema centrado no humano. Declarações de processo são, tipicamente, concisas, rigorosas e independentes de implementação. Elas visam a uma audiência especializada de avaliadores e implementadores de processos, mas são amplamente usadas no planejamento de um projeto. Os detalhes de como isso é feito dependem do negócio e do contexto do projeto. **Declaração contextualizada de processo** é baseada em metodologias e ciclos de vida e apresenta uma guia detalhada de como fazer projeto centrado no usuário, normalmente na forma de um livro-texto, sendo divididos em métodos genéricos (na forma de livros para consulta) e métodos operacionais (manual da empresa). **Instanciação de projeto**, na forma de plano de projeto, firma as atividades a serem executadas em um projeto em particular. **Representação de projeto** engloba a aplicação de recursos específicos, como ferramentas e técnicas. A Tabela 4.3 lista essas diferentes formas de especificação com exemplos.

A tabela 4.3 pode ser vista como uma hierarquia de abstração na descrição do projeto centrado no usuário. Deve ser percebido que apenas a nível de declaração de processo há uma declaração explícita do retorno desejado ou da proposta da atividade. A natureza de objetivos implícita na maioria das especificações das atividades dirige a equipe de projeto através de uma conduta **orientada a objetivo** [Roth, Bennet e Woods, 1987], onde seguir um curso de atividade é considerado como liderar a um retorno bem-sucedido, enquanto que a conduta **direcionada a objetivo** utiliza-se de procedimentos como recursos para conseguir um objetivo explícito. Esse problema é exacerbado dentro de atividades de especialistas em engenharia, como engenheiros de usabilidade, com o resultado de uma entrada inadequada para o projeto e desenvolvimento provida no tempo incorreto. Hakiel [Hakiel, 1997] atribui o problema a uma comunicação pobre surgida de um *mismatch* entre os objetivos dos engenheiros especialistas (aplicar e desenvolver sua perícia) e os objetivos dos gerentes de projeto (completar o projeto dentro dos recursos especificados). Ele concluiu que para garantir a entrada apropriada na usabilidade, o plano de projeto deve incluir, explicitamente, as requeridas atividades centradas no usuário ou humano. É argumentado que problemas surgem quando modelos de projeto centrado no usuário (e outras

Forma de Especificação	Formato dos Requisitos	Usuário	Exemplos de Usabilidade	Exemplos Gerais
Declaração de Processo	Resultado, objetivos, atividades, produtos de trabalho	Assessor, implementador de processo, planejador	ISO TR 18529	ISO 12207, 15288, EFQM
Declaração contextualizada de processo - genérica	Atividades, entradas, saídas	Gerente de projeto	UPA lifecycle, CCTA	Engenharia de software, SSADM
Declaração contextualizada de processo - corporação	Atividades, entradas, saídas	Gerente de projeto	Philips HumanWare	IBM's project lifecycle
Instanciação de projeto	Atividades, dependências, recursos, sincronismo	Gerente de projeto	Plano de Usabilidade	Plano de Projeto
Representação de processo	Atividades, ferramentas, recursos, procedimentos, entradas, saídas	Especialista	Stakeholder workshop, análise da tarefa, prototipação	Análise de sistemas, HAZOP, teste de fábrica

Tabela 4.3. Padrões de processo em relação a outras formas de especificar atividades de projeto.

atividades de negócio) são especificadas em um nível de abstração menor que a do modelo de processo.

CAPÍTULO 5

CONCLUSÕES

As inúmeras pesquisas na aplicação de usabilidade em processos de desenvolvimento de software somam-se como esforços na tentativa de expandir a cultura do design centrado no usuário, de lembrar a empresas e desenvolvedores de que o usuário é o seu motivo de ser.

A sugestão não é a de adotar um dos métodos relacionados anteriormente no presente documento, mas de adaptá-los ao contexto e às necessidades específicas da empresa ou do projeto. A forma como essa adaptação deve ser feita fica como tema para trabalhos futuros.

REFERÊNCIAS BIBLIOGRÁFICAS

[**Alcântara et al., 2002**] Alcântara, A.A.; Maciel, T.M.M.; Meira, S.L.; Silva, F.Q.B. Uso do Processo RUP na implantação da ISO9000-3. 2002.

[**Bevan, 1999**] Bevan, N. Quality in Use: meeting user needs for quality. In: Journal of Systems and Software, 49(1), 89-96.

[**Bevan, 2001a**] Bevan, N., Cost Benefit Analysis. TRUMP deliverable D3.0. Disponível em:

www.usability.serco.com/trump/methods/integration/costbenefits.htm.

Acesso em: 17 ago. 2004.

[**Bevan e Bogomolni, 2000**] Bevan, N. e Bogomolni, I. (2000). Incorporating user quality requirements in the software development process. In Proceedings of the 4th International Software e Internet Quality Week Conference [QWE2000], pp. 1192-1204. Brussels, Belgium, 20-24 November. San Francisco: Software Research Inc.

Disponível em: www.soft.com/QualWeek/QWE2K/Papers/11A.html ou

www.soft.com/QualWeek/QWE2K/Papers.pdf/Bevan.pdf

[**Bevan e Ryan, 2000**] Bevan, N. e Ryan, N. (2000). Introducing UCD methods at Inland Revenue.

www.usability.serco.com/trump/case-studies/irintroducing.htm

[**Bevan et al., 1998**] Bevan, N., Claridge, N., Earthy, J e Kirakowski, J. Proposed Usability Engineering Assurance Scheme. IE2016 INUSE Deliverable D5.2.3. Disponível em: www.lboro.ac.uk/eusc/guides/d523.2b.doc. Acesso em: 17 ago. 2004.

[**Boehm, 1996**] Boehm, B. and Port, D., Escaping the Software Tar Pit: Model Clashes and How to Avoid Them, ACM Software Engineering Notes, January 1999, pp. 36-48.

[**Booch, 1998**] The Unified Software Development Process, Ivar Jacobson, Grady Booch, James Rumbaugh. Addison Wesley, 1998.

[**Calsavara et al., 2002**] Calsavara, A.; Machado, C.A.F.; Reinehr, S.S; Burnett, R.C. Aderência do RUP à norma NBR ISO/IEC 12207. 2002.

[**Coallier, 1994**] Coallier, F.; How ISO 9001 fits into the software world, IEEE Software, Jan. 1994.

[**Crosby, 1978**] Crosby, P. B. (1978). Quality is Free: The Art of Making Quality Certain. New York: McGraw-Hill. ISBN 0-451-62585-4.

[**Daly et al., 1997**] Daly-Jones, O., Bevan, N e Thomas, N. Handbook of User Centred Design. IE2016 INUSE Deliverable D6.2.1., 1997. Disponível em: www.ejeisa.com/nectar/inuse/6.2/index.htm. Acesso em: 17 ago. 2004.

[**Earthy, 1996**] Earthy, J. V. (1996). Development of the usability maturity model. IE2016 INUSE Deliverable D5.1.1(t). www.lboro.ac.uk/eusc

[**Earthy, 1998a**] Earthy, J.V. Usability Maturity Model: processes. IE2016 INUSE Deliverable D5.1.4p., 1998. Disponível em: www.lboro.ac.uk/eusc. Acesso em: 17 ago. 2004.

[**Earthy, 1998b**] Earthy, J.V. Usability Maturity Model: human-centredness scale. IE2016 INUSE Deliverable D5.1.4s. Disponível em: www.lboro.ac.uk/eusc. Acesso em: 17 ago. 2004.

[**Earthy et al., 1998**] Earthy, J., Shackel, B., Courteney, H., Hakiel, S., Serrwood-Jones, B.e Taylor, B. (1998). Human-centred processes. In J. MAY, J. SIDDIQI, e J. WILKINSON, Eds. HCI'98 Conference Companion, Adjunct Proceedings of the 13th BCS Annual Conference on HCI, HCI'98. Sheffield Hallam University, Sheffield, September.

[**Eason e Harker, 1997**] Eason, K. e Harker, S. D. (1997). User centred design maturity. Internal working document, Department of Human Sciences, Loughborough University, Loughborough LE11 3TU.

[**Ehrlich e Rohn, 1994**] Ehrlich, K. e Rohn, J. A. (1994). Cost justification of usability engineering: a vendor's perspective. In R. G. BIAS, e D. J. MAYHEW, Eds. Cost-Justifying Usability, pp. 76-78. London: Academic Press.

[**Extreme, 2001**] Using the Rational Unified Process for Small Projects: Ex-

panding Upon eXtreme Programming. 2001. Rational Software Corporation.

[**EUSC, 1998**] European Usability Support Centres. Web Site Pages on Assurance of Usability, 1998. Disponível em: www.lboro.ac.uk/eusc/index-r-assurance.htm

[**Flanagan, 1996**] Flanagan, G. A. (1996). Usability management maturity. Part 1. Self assessment-how do you stack up? SIGCHI Bulletin, 28, 61-62. www1.acm.org:82/sigs/sigchi/bulletin/1996.4/flanagan.html.

[**Fowler, (s.d.)**] Fowler, Martin. Is Design Dead? [s.d.]. Disponível em: www.martinfowler.com/articles/designDead.html. Acesso em: 1 mai. 2004.

[**Hakiel, 1997**] Hakiel, S. R. (1997). Usability engineering and software engineering: how do they relate? Advances in Human Factors/Ergonomics 21B, Design of Computing Systems, pp. 521-524. Amsterdam: Elsevier.

[**Hamilton e Cutler, 2000**] Hamilton, J. M. e Cutler, A. (2000). Personal communication. Meeting to discuss the integration of TR 18529 and systems capability assessment, DERA SSCE, Great Malvern, July 2000.

[**Hirasawa, 2000**] Hirasawa, N. (2000). Personal communication following the Human Factors 2000 Symposium.

[**ISO9000-3, 1993**] NBR ISO 9000-3, Normas de Gestão da Qualidade e Garantia da Qualidade - Diretrizes para aplicação da NBR 90001 ao Desenvolvimento, Fornecimento e Manutenção de Software, Rio de Janeiro, 1993.

[**ISO 9126, 2000**] ISO 9126, Software Product Quality - quality model, 2000.

[**ISO 9241, 1998**] ISO 9241-11, Ergonomics of office work with VDTs - guidance on usability, 1998.

[**ISO 13407, 1999**] ISO 13407 (1999). Human-centred design processes for interactive systems.

[**ISO TR 18529, 2000**] ISO TR 18529 (2000). Ergonomics of human system interaction - human-centred lifecycle process descriptions.

[**Jacobson, 1994**] Jacobson, I., Object-Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley, 1994.

[**Jokela, 2000**] Jokela, T. (2000). Usability capability models - review and analysis. In S. McDonald, Y. Waern e G. Cockton, Eds. Human-Computer Interaction 2000 People and Computers XIV Usability or Else!, pp. 163-181. London: Springer-Verlag.

[**Kruchten, 2000**] Kruchten, P. The Rational Unified Process - an Introduction. 2^a ed. Addison Wesley, 2000.

[**Kurosu et al., 2001**] Kurosu, M., Horibe, Y., Miki, H., Hirasawa, N. e Horino, S. (2001). ISO13407. Tokyo: Ohm-sha, Ltd.

[**Kuuti et al., 1998**] Kuuti, K., Jokela, T., Nieminen, M. e Jokela, P. (1998). Assessing human-centred design processes in product development by using the INUSE maturity model. In S. NISHIDA e K. INOUE, Eds. Analysis, Design and Evaluation of Man-Machine Systems, pp. 89-94. (7th IFAC/IFIP/IFORS/IEA Symposium, Kyoto, Japan, September). Oxford: Pergamon Press.

[**National/Panasonic, 1999**] National/Panasonic. (1999), Matsushita Universal Design Guide. Matsushita Group Design Department. Internal publication, Matsushita Electrical Industrial Co., Ltd.

[**NBR9001, 1991**] NBR19001, Associação Brasileira de Normas Técnicas. Sistemas de qualidade - Modelo para garantia da qualidade em projetos/desenvolvimento, produção, instalação e assistência técnica, Rio de Janeiro, Brasil, 1991.

[**Norman et al., 1986**] Norman, D. A. e Draper, S. W. (1986). User Centered System Design: A New Perspective on Human-Computer Interaction. Hillsdale, NJ: Lawrence Earlbaum and Associates.

[**Oshiro et al, (s.d)**] Oshiro, A.K.; Novelli, A.D.P.; Caseli, H.M.; Lucena P. Extreme Programming, um novo modelo de processo para o desenvolvimento de software. [s.d.]. 2004.

[**Paulk, 1993**] M. C. Paulk, et. al., Capability Maturity Model for Software, Version 1.1, Software Engineering Institute, CMU/SEI-93-TR-024, February 1993.

[**Paulk, 1995**] Paulk, M.C., How ISO 9001 compares with the CMM. IEEE Software, Jan 1995.

[**Rawsthorne, 2003**] Rawsthorne, D. Comparing/Combining RUP, XP and Scrum - mixing the Process. Cocktail. 2003.

[**Roth, Bennet e Woods, 1987**] Roth, E. M., Bennett, K. e Woods, D. D. (1987). Human interaction with an 'intelligent' machine. International Journal of Man-Machine Studies, 27, 479-525.

[**Schneider, 1998**] Applying Use Cases: A Practical Guide, Geri Schneider, Jason P. Winters, Ivar Jacobson. Addison-Wesley, 1998.

[**Sherwood-Jones, 1995**] Sherwood-Jones, B. (1995). Total systems maturity. Internal report, version 2. BAeSEMA, 1 Atlantic Quay, Broomielaw, Glasgow G2 8JE, UK.

[**Smith, 2003**] Smith, J. A Comparison of RUP and XP. 2003

[**Taylor et al, 1998**] Taylor, B., Gupta, A., Hefley, W., McLelland, I. e Van Gelderen, T. (1998). HumanWare Process Improvement - institutionalising the principles of user centred design. In Tutorial PM14 Human-centred processes and their impact. Human-Computer Interaction Conference on People and Computers XIII, Sheffield Hallam University, September. Tutorial notes available from British HCI Group, www.bcs.org.uk/hci.

[**TIC, 1992**] TickIt Project. Guide to Software Quality System Construction and Certification using EN29001, 1992.

[**TRI, 1994**] Bell Canada Inc; Trillium: Model for Telecom Product Development and Support Process Capability, release 3.0, Dec. 1994

[**Wells, 2003**] Wells, Don. Extreme Programming.
Disponível em: www.extremeprogramming.org. Acesso em: 2 ago. 2004

APÊNDICE A

TERMINOLOGIA DE DESCRIÇÃO, AVALIAÇÃO E MELHORIA DE PROCESSO

Uma tarefa técnica ou de gerenciamento que contribua para a criação da saída, ou produtos, de um processo ou ainda, que intensifique as capacidades de um processo é chamada *atividade* ou *prática*. Os elementos do formato usado para descrever processos na ISO 15504 estão listados abaixo. Estes estão sendo seguidos na ISO TR 18529 *Descrição dos Processos de Ciclo de Vida Centrados no Humano*.

- *Número do processo* - para uma referência precisa.
- *Nome do processo* - resumo do processo.
- *Proposta do processo* - o que é realizado pelo processo.
- *Retorno* - porque ele é executado. O resultado da aplicação bem sucedida de um processo.
- *Práticas* - o que é feito para satisfazer o escopo.
 - Número da prática - para uma referência precisa.
 - Nome da prática - resumo da prática.
 - Descrição da prática - qual tarefa é executada.
- *Produtos do trabalho* - itens utilizados e produzidos por um processo incluindo: pedaços de informação, documentos, hardware, software, cursos de treinamento e conscientização de indivíduos.
- Uma validação disciplinada do processo de uma organização mediante um modelo é chamado *avaliação de processo*.

A avaliação de um processo geralmente foca em identificar prioridades de melhoria (i.e. validação formativa). Ações tomadas com o intuito de modificar os processos de uma organização a fim de que eles satisfaçam as necessidades de negócio de uma organização e atinjam suas metas de uma maneira mais eficiente são chamadas *melhorias de processo*. Avaliação e

melhoria de processo são descritas abaixo em mais detalhes. Avaliação de processo busca estabelecer, primeiramente, se os processos são executados com sucesso e, em segundo, o grau de controle para cada processo. Uma escala de *capacidade* é usada nessa avaliação. Uma escalada de capacidade é uma escala de tipos qualquer. A escala mais comum (usada em ambos: *Capability Maturity Model*[Paulk, 1993] e ISO 15504) utiliza seis níveis para avaliar a capacidade do processo.

- 1 Nenhum processo, ou processos, alcançou um resultado (nível 0).
- 2 Performance de maneira ad-hoc (nível 1).
- 3 Monitoramento de tempo e qualidade de produto (nível 2).
- 4 Uso de infra-estrutura e procedimentos corporativos definidos (nível 3).
- 5 Uso de controle estatístico (nível 4).
- 6 Otimização de cada processo para alcançar necessidades de negócio atuais e futuras (nível 5).

Uma avaliação de processo irá examinar a evidência para a performance das práticas e a existência e qualidade do produto de trabalho. No entanto, a decisão final sobre o grau de performance de um projeto é baseada no grau no qual os retornos são atingidos.

Melhoria de processo é uma atividade iterativa. A Figura A resume o processo implementado nos exemplos contidos no apêndice D e apresenta esses processos como um ciclo englobando os seguintes estágios: revisão das necessidades de negócio, seleção de relevantes processos para referência, avaliação da capacidade atual, definição da performance necessária, decidir como contornar pequenas falhas (e como preservar boas práticas) e mudança organizacional. A avaliação da capacidade atual é feita através do exame de um ou mais projetos, idealmente, cobrindo uma gama de estágios de ciclos de vida (da iniciação a conclusão).

O objetivo que guia a melhoria de processo é o benefício aos negócios. Fayad e Laitinen (1097) ilustram os riscos de uma melhoria de processo não planejada. Abordagem para uma melhoria de processo planejada são propostas por Zultner (1993), que descreve um procedimento para seleção dos processos a serem melhorados baseado nos atributos do produto, e por Keen (1997), que descreve um procedimento para seleção dos processos a serem melhorados baseado nos objetivos do negócio. Os desafios e oportunidades apresentados pela ISO 12207 e ISO 15288 dados na seção 6 e 7, ilustram a

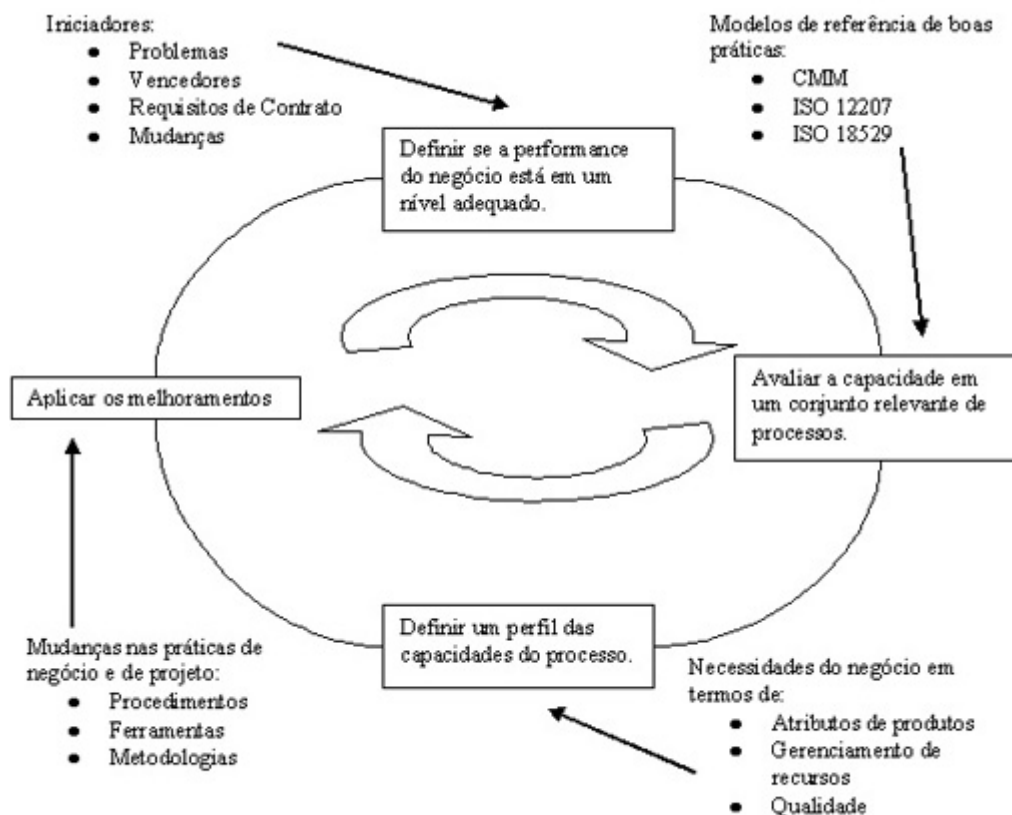


Figura A.1. O processo de melhoramento do processo

gama de possíveis diretrizes de negócio para melhoria de processo. Qualquer que seja a abordagem utilizada, o custo-benefício depende de ajustar a abordagem às necessidades dos processos de uma organização, e alguma forma de avaliação de práticas atuais é necessário a fim de identificar pontos fortes e fracos. Este assessment não precisa ser muito rigoroso ou cobrir uma vasta gama de processos. A Tabela A1 descreve abordagem para avaliação e os benefícios dados a tipos particulares de organizações.

Independente da abordagem utilizada para avaliação, ela precisa ser ajustada e focada para obter mais eficiência. Na prática, a eficiência de qualquer abordagem para avaliação de processo depende em projetar a avaliação a fim de que examine somente os processos que estão relacionados às diretrizes selecionadas para melhoria de processo. Exemplos de diretrizes são: riscos relacionados a um contrato particular, capacidade concorrente e defeitos que ocorrem ao longo das versões ou linhas de produtos. Com um assessment mais

formal esse ajuste ocorre primeiramente para a avaliação. Uma avaliação no estilo de workshop pode ser mais flexível e o ajuste pode ocorrer no próprio workshop.

Tabela A.1. Usos da avaliação de processos

Abordagem utilizada	Tipo da organização				Descrição da abordagem em uso pelas organizações
	Orientada a contrato	Orientada a serviço	Orientada a qualidade	Empreendimento/parcerias	
Certificado	Preferível	Utilizado		Utilizado	Alcance de um nível objetivado de capacidade organizacional para um conjunto genérico de processos. (A abordagem mais tradicional, ex: CMM)
Avaliação de riscos	Utilizado			Preferível	Capacidade em um conjunto de processos relevante para os riscos de uma missão ou projeto particular.
Perfil		Preferível	Utilizado		Classificação da capacidade em uma vasta gama de processo. Fornece uma visão geral sobre quais devem ser melhorados.
Workshop			Preferível		Exame informal de um projeto ou unidade organizacional perante os requisitos de um modelo de processo.
Razão para uso da abordagem.	Certificados provêem um testemunho de capacidade. A avaliação dos riscos é utilizada pelos clientes. Contratos são necessários para os melhoramentos especificados.	O uso mais comum é como benchmarking para outras organizações. Certificados são mais dígnos e derivados de objetivos do negócio.	O foco organizacional em melhorias de processos é comumente achado no desenvolvimento de produtos genéricos, especialmente no japão. A capacidade pode ser avaliada informalmente.	Organizações que pretendem desenvolver ou operar um sistema necessitam um claro entendimento das fraquezas nas áreas chaves. Isto inclui a organização do cliente e do usuário.	

APÊNDICE B

ISO 13407

Esse apêndice contém um resumo da ISO 13407:1999 *Processos de projeto centrado no humano para sistemas interativos* e apresenta seu sumário.

Tabela B.1. Visão geral da ISO 13407

Título	ISO 13407 Processos de projeto centrado no humano para sistemas interativos
Data	Julho de 1999
Escopo	Guia para atividades centradas no humano através do ciclo de vida de sistemas de computadores iterativos.
Conteúdo	Rasão para o processo centrado no usuário. Uma descrição dos quatro princípios essenciais de um projeto centrado no humano. Planejamento do projeto do processo centrado no usuário. Descrição das quatro atividades que devem ser realizadas durante o processo de desenvolvimento de um sistema. Uma lista de processos atuais e padrões de produto para projeto centrado no usuário.
Propósito	ISO 13407 objetiva ajudar aqueles responsáveis por gerenciar os processos de projeto de hardware e software a fim de identificar e planejar atividades de projeto centradas no usuários eficientes e precisas.
Audiência	Aqueles que gerenciam o processo de projeto. Todas as partes envolvidas do desenvolvimento de sistemas baseados no humano, incluindo os usuários finais dos sistemas, devem achar o padrão que seja relevante.
Requisitos	Qualquer processo de desenvolvimento que reivindique ter cumprido as recomendação na ISO 13407 deve especificar os procedimentos utilizados, informações coletadas e o uso feito dos resultados.

As cláusulas do padrão são as seguintes:

Introdução

- 1 Escopo
- 2 Termos e definições
- 3 Estrutura desse Padrão Internacional
- 4 Razão para adotar esse processo de projeto centrado no humano
- 5 Princípios de projeto centrado no humano
 - 5.1 Geral
 - 5.2 O envolvimento ativo dos usuários e um claro entendimento dos requisitos do usuário de das tarefas
 - 5.3 Uma apropriada alocação de tarefas entre usuário e tecnologia
 - 5.4 Iteração de soluções de projeto
 - 5.5 Projeto multidisciplinar
- 6 Planejando o processo de projeto centrado no humano
- 7 Atividades projeto centrado no humano
 - 7.1 Geral
 - 7.2 Entenda e especifique o contexto de uso
 - 7.3 Especifique os requisitos da organização e do usuário
 - 7.4 Produza soluções de projeto
 - 7.5 Avalie projetos mediante os requisitos
- 8 Compatibilidade

Apêndice A (informativo)	Guia para outros padrões relevantes
Apêndice B (informativo)	Exemplos de estruturas para relatório de testes de usabilidade
Apêndice C (informativo)	Exemplo de procedimento para demonstrar compatibilidade com esse Padrão Internacional.

Bibliografia

APÊNDICE C

ISO TR 18529 - PROCESSOS PARA O PROJETO CENTRADO NO HUMANO

Esse apêndice apresenta um resumo da ISO TR 18529:2000 *Descrição de processos de ciclo de vida de projeto centrado no humano* e descrições parciais de processos de projeto centrado no humano (HCD, *human-centered design*) contidas nesse padrão.

Processos HCD consideram os usuários finais e outros stakeholders na especificação, desenvolvimento e operação de um sistema. Os processos sempre remetem ao sistema funcional sendo desenvolvido, não apenas aos detalhes de hardware e software. Os processos levam em conta as atividades centradas no humano durante toda a vida do sistema. Cada descrição de processo é apresentada na forma: número do processo, título, propósito, lista de resultados, lista de títulos de práticas. O uso de cada parte da descrição está resumido no Apêndice A desse documento. Práticas são passadas através do uso de métodos, técnicas e ferramentas. Métodos, técnicas e ferramentas particulares centrados no humano não são descritos na ISO TR 18529. No entanto, algumas notas explicativas para as práticas são ilustram os requisitos de métodos, técnicas e ferramentas. Os próximos parágrafos resumem cada processo de projeto centrado no humano, seus resultados e práticas.

HCD 1: INSERIR NA ESTRATÉGIA DO SISTEMA OS CONCEITOS DO HCD

O propósito do processo HCD 1 é estabelecer e manter um foco nas questões dos stakeholders e do usuário nas partes da organização que lidam com marketing, conceitos, desenvolvimento e suporte de sistemas. Como resultado de uma bem sucedida implementação desse processo teremos que:

- O marketing levará em conta questões de usabilidade, ergonomia e sócio-técnicas.
- O sistema estará focado em satisfazer as necessidades e expectativas do usuário.
- Os planejadores irão considerar os stakeholders e os requisitos da organização quando forem definir a estratégia do sistema.

- O sistema responderá mais facilmente a mudanças em seus usuários (suas necessidades, tarefas, contexto, etc).
- O empreendimento responderá mais facilmente a mudanças em seus usuários.
- Será menos provável que o sistema seja rejeitado pelo mercado.

Esse processo compreende as seguintes práticas:

- Representar o usuário final.
- Coletar a inteligência do marketing.
- Definir e planejar uma estratégia de sistema.
- Coletar o feedback do marketing.
- Analisar as tendências do usuário.

HCD 2: PLANEJAR E GERENCIAR O PROCESSO HCD

O propósito do processo HCD 2 é especificar com as atividades centradas no humano se encaixam no empreendimento e em todo o processo de ciclo-de-vida do sistema. Como resultado de uma bem sucedida implementação desse processo teremos que:

- O plano de projeto permitirá a interação e incorporação do feedback do usuário.
- Recursos serão alocados para comunicação eficiente entre os participantes do time de projeto.
- Potenciais conflitos e trade-offs entre questões centradas no humano e outras serão reconciliados.
- Processos centrados no humano serão incorporados em sistemas de qualidade, procedimentos e padrões.

Esse processo compreende as seguintes práticas:

- Consultar stakeholders.
- Identificar e planejar o envolvimento do usuário.

- Selecionar métodos e técnicas centrados no usuário.
- Assegurar que o time de projeto segue uma abordagem centrada no humano.
- Planejar as atividades centradas no humano.
- Gerenciar as atividades centradas no humano.
- Premiar a abordagem centrada no humano.
- Prover suporte ao projeto centrado no humano.

HCD 3: ESPECIFICAR OS STAKEHOLDERS E OS REQUISITOS ORGANIZACIONAIS

O propósito do processo HCD 3 é estabelecer os requisitos da organização e outros grupos de interesse do sistema. Esse processo leva em conta completamente as necessidades, competências e o ambiente de trabalhos de cada stakeholder relevante do sistema. Como resultado de uma bem sucedida implementação desse processo teremos:

- A necessidade de performance do novo sistema mediante objetivos operacionais e funcionais.
- Requisitos legais ou legislativos relevantes.
- Cooperação e comunicação entre usuários e outros grupos relevantes.
- Os trabalhos do usuário (incluindo alocação de tarefas, conforto do usuário, segurança, saúde e motivação).
- Tarefa sobre a performance do usuário ao utilizar sistema.
- Projeto de trabalho e práticas e estruturas organizacionais.
- Viabilidade de operação e manutenção.
- Objetivos para a operação e/ou uso dos componentes de software e hardware do sistema.

Esse processo compreende as seguintes práticas:

- Esclarecer e documentar as metas do sistema.
- Definir stakeholders.

- Impor riscos aos stakeholders.
- Definir o sistema.
- Gerar os requisitos dos stakeholders e da organização.
- Definir a qualidade do uso.

HCD 4: ENTENDER E ESPECIFICAR O CONTEXTO DE USO

O propósito do processo HCD 4 é identificar esclarecer e armazenar as características dos stakeholders, suas tarefas e o ambiente físico organizacional no qual o sistema irá operar. Como resultado de uma bem sucedida implementação desse processo teremos:

- As características do usuário final.
- As tarefas que os usuários irão realizar.
- A organização e o ambiente no qual o sistema é usado.

Esse processo compreende as seguintes práticas:

- Identificar e documentar as tarefas do usuário.
- Identificar e documentar atributos significantes do usuário.
- Identificar e documentar o ambiente organizacional.
- Identificar e documentar o ambiente técnico.
- Identificar e documentar o ambiente físico.

HCD 5: PRODUZIR SOLUÇÕES DE PROJETO

O propósito do processo HCD 5 é criar potenciais soluções de projeto utilizando como fonte de suporte práticas já estabelecidas do estado-da-arte, a experiências e conhecimento dos participantes e os resultados do contexto da análise do uso. Como resultado de uma bem sucedida implementação desse processo teremos que:

- Todo o sistema social-técnico no qual qualquer componente técnico opere será considerado no projeto.

- Características e necessidades do usuário serão levadas em consideração na compra de componentes do sistema.
- Características e necessidades do usuário serão levadas em consideração no projeto do sistema.
- Conhecimento existente sobre as melhores práticas da engenharia de sistemas social-técnica, ergonomia, psicologia, ciência cognitiva e outras disciplinas relevantes serão integrados aos sistema.
- A comunicação entre os stakeholders do sistema será melhorada porque as decisões de projeto serão mais explícitas.
- O tipo de desenvolvimento será capaz de explorar vários conceitos de projeto antes de se decidir por um.
- Feedback do stakeholder e do usuário final serão incorporados no projeto mais cedo no processo de desenvolvimento.
- Será possível validar várias iterações de um projeto e de projetos alternativos.
- A interface entre o usuário e os componentes de software, hardware e organizacionais do sistema será projetada.
- O treinamento e o suporte dos usuários serão desenvolvidos.

Esse processo compreende as seguintes práticas:

- Alocar funções.
- Produzir um modelo de tarefas composto.
- Explorar o projeto do sistema
- Use conhecimento existente para desenvolver soluções de projeto.
- Especificar o sistema.
- Desenvolver protótipos.
- Desenvolver o treinamento do usuário.
- Desenvolver o suporte ao usuário.

HCD 6: VALIDAR O PROJETO MEDIANTE OS REQUISITOS

O propósito do processo HCD 6 é coletar o feedback do projeto em desenvolvimento. Esse feedback será coletado de usuários finais e outras fontes representativas. Como resultado de uma bem sucedida implementação desse processo teremos que:

- O feedback será provido para melhorar o projeto.
- Será avaliado se os objetivos dos stakeholders e organizacionais foram atingidos.
- Uso prolongado do sistema será monitorado.

No caso de validações para identificar melhoramentos no sistema (validação formativa), uma bem sucedida implementação do processo refletirá:

- Problemas e escopos potenciais para o melhoramento: na tecnologia, no suporte de material, no ambiente físicos ou organizacional e no treinamento.
- Que opção de projeto melhor se encaixa nos requisitos funcionais e do usuário.
- Feedback e requisitos posteriores do usuário.

No caso de validações para avaliar se os objetivos foram atingidos (validação aditiva), uma bem sucedida implementação do processo demonstrará:

- Quão bem o sistema atinge suas metas organizacionais
- Que um projeto particular satisfaz requisitos centrados no humano.
- Conformidade com requisitos internacionais, nacionais e/ou legais.

Esse processo compreende as seguintes práticas:

- Especificar e validar o contexto de validação.
- Validar protótipos recentes a fim de definir requisitos para o sistema.
- Validar protótipos a fim de melhorar o projeto.
- Validar o sistema a fim de checar se os requisitos do sistema foram satisfeitos.

- Validar o sistema a fim de checar se as práticas necessárias foram seguidas.
- Validar o sistema em uso a fim de assegurar que ele continuar a satisfazer necessidades organizacionais e do usuário.

HCD 7: INTRODUIZIR E OPERAR O SISTEMA

O propósito do processo HCD 7 é estabelecer os aspectos humanos do suporte a da implementação do sistema. Como resultado de uma bem sucedida implementação desse processo teremos que:

- As necessidades dos stakeholders do sistema serão comunicadas ao projeto.
- O controle de mudanças, incluindo as responsabilidades dos usuários e desenvolvedores, será especificado.
- Os requisitos de suporte a usuários finais, mantedores e outros stakeholders serão endereçados.
- Haverá obediência a procedimentos de saúde e segurança.
- Haverá suporte a customização local do sistema.
- As reações do usuário serão coletadas e as mudanças resultantes do sistema serão reportadas aos stakeholders.

Esse processo compreende as seguintes práticas:

- Controle de mudanças
- Determinar o impacto na organização e nos stakeholders.
- Projeto local e customizado.
- Entregar o treinamento do usuário.
- Suporte aos usuários nas atividades planejadas.
- Garantir conformidade com a legislação sobre a ergonomia do ambiente de trabalho.

Tabela C.1. Visão geral da ISO 18529

Título	ISO TR 18529 Descrição de processos de ciclo de vida de projeto centrado no humano
Data	Abril de 2000
Escopo	Um modelo formalizado baseado no processo centrado no usuário descrito na ISO 13407 <i>Processos de projeto centrado no humano para sistemas interativos</i> .
Conteúdo	Descrição estruturada dos processo que compreendem uma abordagem centrada no humano e listagem de seus componentes, retornos e a informações usadas e produzidas. Os processos do modelo são descritos no formato definido na ISO TR 15504 <i>Avaliação de processos de software</i> .
Propósito	O modelo é direcionado para o uso na especificação, avaliação e melhoramentos dos processos centrados no usuário no desenvolvimento do sistema e na operação do sistema. O Relatório Técnico é direcionado para agir como um anexo da ISO 13407 para uso dos avaliadores de processos.
Audiência	Aqueles que estão envolvidos no projeto, uso e avaliação de processos de ciclo de vida para o hardware e software do sistema. Os Fatores Humanos e os praticantes de HCD envolvidos no desenvolvimento de produtos ou sistemas são identificados como usuário e beneficiadores secundários desse documento.
Requisitos	O TR não é normativo, mas contém um anexo que descreve os requisitos situados em modelos de avaliação derivados da ISO 15504, o equivalente para software da ISO TR 18529.

APÊNDICE D

EXEMPLOS DE MELHORIA EM PROCESSOS CENTRADOS NO USUÁRIO

D.1 AVALIAÇÃO FORMAL FRENTE O MODELO DE MATURIDADE DE USABILIDADE

O UK Inland Revenue em conjunto com fornecedores de software [Bevan e Ryan, 2000] usaram uma avaliação formal frente o modelo de maturidade de usabilidade para obter uma avaliação completa sobre a extensão na qual os parceiros estavam fazendo uso de técnicas centradas no usuário e prover uma base para melhoramentos.

D.1.1 Procedimento

Um procedimento convencional de um processo de avaliação de software baseado na ISO TR 15504 *Avaliação do processo de software* foi usado para avaliar a extensão na qual (a) os objetivos no modelo de maturidade de usabilidade foram atingidos (b) as práticas definidas foram utilizadas. Uma avaliação formal com duração de 1 semana foi realizada por dois assessores, assistidos por dois especialistas em usabilidade que identificaram oportunidades para o melhoramento de processos. Um total de 13 stakeholders associados ao projeto experimental em diferentes níveis foram entrevistados em sessões de três horas.

Doze meses depois, quando os melhoramentos foram aplicados, uma segunda avaliação similar foi realizada para ver se os objetivos foram atingidos.

D.1.2 Resultado

A avaliação produziu um perfil detalhado da capacidade de processo centrado no humano e uma rica informação sobre onde os melhoramentos trariam benefícios. Em uma reunião de feedback no último dia de avaliação, é estabelecida uma base para um conjunto de comum acordo de atividades de melhoramento.

Quando os resultados da segunda avaliação foram apresentados em uma reunião de stakeholders sênior, os benefícios foram suficientes para a reunião autorizar a incorporação da maioria dos métodos centrados no usuário dentro

dos processos documentados da organização. A reunião também sugeriu que avaliações regulares sobre a maturidade da usabilidade deveriam ser agendadas a fim de monitorar o progresso.

D.2 AVALIAÇÃO INFORMAL FRENTE O MODELO DE MATURIDADE DE USABILIDADE

O modelo de maturidade de usabilidade foi utilizado pela Indústria de Aviões Israel (a) para obter um entendimento da extensão na qual a organização estava fazendo uso de técnicas centradas no usuário e (b) para prover uma base para o progresso [Bevan & Bogomolni, 2000].

D.2.1 Procedimento

As atividades do modelo de maturidade de software foram usadas como uma checklist de boas práticas para uma dia de entrevistas. O especialista em melhoramento de processo classificou cada atividade como não-realizada, largamente realizada ou gerenciada, baseado em uma breve discussão com um ou dois desenvolvedores ou gerentes que tivessem mais conhecimento em cada área.

Adiante, pequenas entrevistas foram realizadas dezesseis meses depois, quando os melhoramentos foram aplicados.

D.2.2 Resultado

Embora algumas classificações não tenham sido inteiramente representativas, foram suficientes para prover uma base para um programa de melhoramento de comum acordo. Como o especialista em melhoramento de processo, que havia posto em execução os melhoramentos, realizou a segunda entrevista, foi fácil chegar num comum acordo com a organização sobre o grau de melhoramentos em menos de duas horas.

D.3 AVALIAÇÃO INFORMAL FRENTE O MODELO CIENTE DO HUMANO

O Modelo Ciente do Humano foi usado pela Philips [Taylor et al., 1998] quando essa grande consumidora e produtora de tecnologia identificou a necessidade de elevar o perfil corporativo de questões do usuário e integrar o projeto centrado no usuário, especialmente nas funções de marketing da organização.

D.3.1 Procedimento

Projeto Cooperativo Philips trabalhou junto ao Instituto de Engenharia de Software da Universidade de Carnegie Mellon (criadores do CMM) para definir um conjunto de processos para o projeto centrado no usuário de produtos genéricos, para uma abordagem centrada no usuário para o marketing, e para o gerenciamento de processos específicos para o uso de informação sobre usuários dentro do projeto. O modelo é usado como base para um workshop anual de cada projeto. O workshop é facilitado por alguém familiarizado com Fatores Humanos e o modelo de processo. Não é uma avaliação, embora o projeto possa tomar nota de pontos bons e ruins sobre a performance do processo. O resultado são as ações tomadas pelos participantes para o próximo ano. Muitas ações estão relacionadas a linhas de tempo ou eficiência das comunicações do projeto, referentes a requisitos do usuário e questões de usabilidade.

D.3.2 Resultado

O resultado é a inserção do design centrado no usuário nas atividades do projeto com uma melhora nas experiências do usuário para a compra e/ou uso de produtos da Philips. Existe um padrão corporativo para design centrado no usuário que é moldado e possuído por cada projeto, sendo revisado em uma base regular.